Preliminary Results of DSN Performance for Convolutional Codes With a Viterbi Decoder

J. M. Urech
Station Director, Cebreros, Spain

L. D. Vit
Operations Manager, Robledo Station

B. D. L. Mulhall
DSN Systems Engineering Office

To determine DSN Telemetry System performance when maximum-likelihood convolutional decoding is employed, testing is being undertaken at DSS 62, Madrid, Spain. Testing hardware and software have been developed to evaluate the performance of the DSN in the Viterbi mode with the LV7015 model. Since the bit errors at the decoder output occur in bursts, the test program includes a series of statistical analyses in runs of correct bits and bursts of bits in error.

I. Introduction

This is the first report on the Madrid DSN engineering task "DSN Performance for Convolutional Decoding," undertaken by DSSs 61/62 in a joint effort with the JPL DSN Systems Engineering Office. The objectives of this task are to determine telemetry system performance with maximum-likelihood convolutional decoding using the Viterbi algorithm. Specifically, bit error rate, burst error statistics, and performance monitor capability are objectives.

The task requires the integration of a maximum-likelihood convolutional Viterbi encoder/decoder (model LINKABIT LV7015) into the DSN Telemetry and Command Data Subsystem (TCD) as well as the evaluation of decoder performance. Since the LV7015 is functionally equivalent to the Maximum Likelihood Convolutional Decoder (MCD) being implemented in the Network (with two small exceptions which will be discussed later), performance of the LV7015 can be used to predict DSN Telemetry System performance for flight operations. It also includes the development of the appropriate testing software and a real-time performance evaluator algorithm.

The task began with the receipt of the LV7015 unit at the end of June. This article covers the integration phase and the preliminary results that have been obtained thus far. For clarification purposes, it has been divided into the following sections:

Hardware integration. Describing the hardware changes that were required to interface the decoder with the system and the implementation of the normalization rate counter and sync status bit.

Software integration. Describing the operational software modifications required for operating in the Viterbi mode, formatting of data flow, and processing of sampled parameters (normalization rate and sync status).

System calibration and preliminary evaluation. Describing the system parameters which affect decoder performance and the *theoretical evaluation of the normalization rate values.*

Testing software. Describing the test software development to analyze and evaluate the tests to be performed.

II. Hardware Integration

The LV7015 has been installed within the Data Decoder Assembly (DDA), as shown in Fig. 1. The idea behind this integration scheme has been to minimize the amount of modifications or additions required to incorporate the high-rate convolutional decoder into any of the existing Multimission Telemetry (MMT) strings. Furthermore, the modifications introduced in the operational equipment have been made, seeking a total compatibility with both operational software and hardware.

All the hardware modifications have been incorporated into a single DDA interface board, known as the Symbol Synchronizer Assembly–Block Decoder Assembly/DDA (SSA–BDA/DDA) coupler. The modifications are described in four different sections, according to the following criteria:

(1) Quantization scheme modification.

(2) Viterbi serial data input to DDA, and formatting modification.

(3) Normalization rate counter implementation.

(4) Viterbi decoder sync condition monitoring.

A. Quantization Scheme Modification

There exists an 8-level quantization scheme, already implemented within the SSA–DDA coupler, which is used by the firmwired sequential convolutional decoder of the DDA (Fano algorithm). The 8-level quantization scheme required by the Viterbi decoder is slightly different, as depicted in Fig. 2.

The conversion from the DDA 8-level scheme into the LV7015 scheme is accomplished by exclusive-oring the sign bit with the two magnitude bits to obtain the new magnitude bits. The sign bit remains unchanged.

Example: Convert DDA level 5 (binary 101) into LV7015 equivalent level.

	Fano's				Viterbi's
Sign bit	1	→	Unchanged	→	1
Magnitude 1	0	→	$0 \oplus 1$	→	1
Magnitude 2	1	→	$1 \oplus 1$	→	0
			Mag $\oplus$ Sign		
Octal	5		$\xrightarrow{\hspace{3em}}$		6

(Symbol $\oplus$ denotes exclusive-or operation.)

B. Viterbi Serial Data Input to DDA and Formatting Modification

To accept the Viterbi decoded data into the SSA–DDA coupler, a new operation mode of this coupler has been devised, maintaining some of the peculiarities of the convolutional decode mode (such as the 8-level quantization scheme). The new operation includes a full uncoded mode using as a serial data input the decoded data output from the Viterbi decoder.

The new operating mode is characterized by a simultaneous setting of both the uncoded mode (UCM1) and the convolutional mode (CCM1) in the corresponding initialization command memory. This is accomplished by modifying the SSA coupler command in the operational software. At the same time, the old uncoded and convolutional code mode indicators have to be modified accordingly so as not to create confusion in the existing hardware. For this reason, three new modes of operation are formed with two command control logic signals, as follows (the dot represents a logic AND operation):

Mode	*Controlled by*
Uncoded	UCM1 · CCM0
Convolutional (not Viterbi)	UCM0 · CCM1
Viterbi	UCM1 · CCM1

The serial data input of the uncoded data formatting registers is modified to accept either uncoded data from the SSA (existing capability retained), or decoded data from the Viterbi decoder. The clock accompanying the data is modified in the same way.

The modification includes the generation of the new operating modes and their incorporation into the applicable parts of the existing circuitry.

C. Normalization Rate Counter Implementation

The behavior of the Viterbi decoder may be known dynamically by observing the so-called "normalization rate." A normalization process takes place whenever a path global metric exceeds a fixed threshold established in the decoder circuitry as part of its memory overflow-protection scheme. When this happens, a normalization pulse is generated and sent to the appropriate decoder section to subtract a fixed quantity and thus reduce the metric to a value below the threshold.

To take advantage of this mechanism, a copy of the normalization pulses is used to feed a counter, which is periodically read by the operational software. In this way, a normalization rate can be computed and further utilized as an estimator of the signal-to-noise ratio of the signal processed by the decoder (see Section III).

The implementation of this counter requires the deletion of a test input monitoring not used at present either in the operational or in the test software.

D. Viterbi Decoder Sync Condition Monitoring

When the normalization rate exceeds a hardwired limit (which corresponds roughly to a symbol error rate greater than 11 percent), the Viterbi decoder declares itself out-of-sync. The monitoring of a condition to determine the in- or out-of-sync was implemented.

III. System Calibration and Preliminary Evaluation

After the integration of the Linkabit LV7015 into the existing multimission telemetry equipment, a series of tests was performed to determine the optimum system calibration and to evaluate its performance.

Theoretical documents concerning the Viterbi decoding algorithm are numerous but frequently ignore the effect of elements such as the receiver or demodulator. In addition, the information provided by the decoder manufacturer was insufficient in this respect, and did not include areas of interest for this study. It was, therefore, necessary to deduce specific operating characteristics from direct observations and many hours of testing. To this end the following lines of action were established:

(1) The decoding function was verified at high signal-to-noise ratio (SNR) using a simple test pattern.

(2) The Viterbi mode of operation explained in Section II was checked and found to be compatible with other operational modes.

(3) As explained elsewhere, the normalization rate counter is periodically transferred to the operational software for further processing. The different normalization criteria presented in available literature did not precisely match the LV7015 normalization scheme. Therefore, the normalization mechanism was studied and evaluated by forcing specific error patterns into otherwise error-free data and correlating the expected values with the actual normalization counter readings. The use of the normalization counter as a system performance evaluator will be explained in Subsection B.

(4) The next testing was aimed at studying the basic system calibration as explained in Subsection A.

A. System Calibration

Among the known degradation factors that take place along the system configuration, the Subcarrier Demodulator Assembly–Symbol Synchronizer Assembly (SDA–SSA) interface calibration plays a most significant role. In the soft quantization mode, the SSA integrator output is converted into a 3-digit binary number that constitutes a fundamental element in the decoding process. This analog-to-digital correspondence at the SSA integrator output is influenced by the relationship

$$K = \frac{\text{integrator mean (mV)}}{\text{quantization interval (mV)}} \qquad (1)$$

The quantization intervals have been represented in Fig. 2, Subsection II-A, and the integrator mean is actually the average of the integrated symbols. In a standard calibration procedure, given an arbitrary telemetry modulation index, the SDA output is adjusted to provide 280 mV to the SSA input. This is done by means of a set of attenuators (modulation index (MI) attenuators that will compensate different modulation indexes to have a relatively constant integration mean at the SSA output).

However, this standard procedure must be proven to be optimum for the Viterbi decoding algorithm at different SNRs. In other words, it is necessary to find an optimum K and its influence on the decoding behavior.

Although the specific purpose of the MI *attenuators is to compensate different modulation indexes, they will be used here as a way of controlling the integrator output relative to the quantization intervals.* (These quantization intervals are fixed and hardwired to 312 mV).

By definition, the optimum K will be such as to provide the minimum bit error rate at the decoder output as a function of the SDA output signal level. Consequently, the SDA output was varied in 2-dB steps and the bit error rate (BER) measured at each point. Figure 3 shows this relationship for three different energy per symbol-to-spectral noise density ratios. The reference value of 0 dB selected was that resulting from the standard setup procedure, i.e., 280 mV at the SDA output.

Figure 3 then shows that the bit error rate goes through a minimum for a determined SDA-SSA adjustment and this holds at different energy-per-symbol-to-noise-spectral density (ST_S/N_0) values for a specific test configuration. However, we are actually interested in the corresponding value of K which comes in terms of SSA output values, while Fig. 3 was obtained in terms of SDA output values. Since the value of K cannot be determined easily as a function of the SDA calibrations, it was evaluated by experimental testing and was found that:

$$\{\text{SDA output: 280 mV}\} \approx \{K: 2.5\} \tag{2}$$

The value of K may vary, however, from 2 to 3 depending upon the degree of accuracy in the setup calibrations.

In general then, it can be said that the standard setup procedure (280 mV into the SSA) is valid for the Viterbi decoding algorithm and holds for different signal-to-noise ratios. As the MI setting attenuators are used very frequently as an operational parameter in any system configuration, care must be taken to select the proper value to avoid the degradations that would result (Figs. 3, 4, and 5).

It must be noted that a change of 1 unit in the SDA attenuator corresponds to a change of 2 dB in the SSA output signal power for a constant signal-to-noise ratio. Therefore:

(1) If the MI attenuators are *decreased* by *one* step,

$$\Delta\,(\text{MI})\text{: } -1 = \Delta\,(M^2)\text{: } +2\text{ dB}$$

From Eq. (1),

$$\Delta\,(\text{MI}) = 10 \log \left(\frac{K_1 \cdot Q}{K_0 \cdot Q}\right)^2 = 2$$

Thus,

$$\Delta\,(\text{MI}) = \frac{K_1}{K_0}\text{ (ratio): }\log^{-1}(2/20)\text{: } 1.258$$

(2) If the MI attenuators are *increased* by *one* step, then

$$\Delta\,(\text{MI}) = \frac{K_1}{K_0}\text{: } 1/1.258 \text{ (see Figs. 4 and 5)}$$

where M^2 represents the signal power at the SSA output, Q is the quantization interval and the built-up MI steps of 2 dB are a system characteristic.

Expression (2) was found experimentally. However, it will be justified in Subsection III-B-3 when analyzing the normalization rate parameter. Parameters $\overline{N}_b$ ( ) in Figs. 4 and 5 will be explained in Subsection III-B-3.

B. System Performance Evaluator

It is convenient to derive the system performance evaluation from some parameter directly related to the decoder operation, and the only parameter that may be made available to the operational program is the normalization counter, as already explained. The following presentation justifies the use of the normalization values as a performance evaluator.

1. Normalization rate. The Viterbi decoder algorithm behaves basically as a progression along the trellis by pairwise comparisons of paths and the elimination of less probable paths, following a metric criterion. The pairwise comparisons are made at each bit time and the metric values are derived from the channel symbol quantizations provided by the Symbol Synchronizer Assembly and the branch symbols hypothesized by the so-called "normalization rate" mechanism. The normalization rate will then be used to evaluate decoder performance.

The normalization mechanism may be visualized as being composed of a set of 4-stage buffers that, at each bit time, are incremented by a metric value and then compared pairwise as per the trellis structure. To simplify the scheme it may be assumed that the path holding the lowest metric is considered to be the "best path." However, during the decoding process all paths including the "best path" will accumulate metric values so as to saturate their corresponding buffers (assuming a significant noise level). In the decoding range of operation the "wrong" paths will nevertheless accumulate at a much faster rate than the "best" path. Therefore, the decoder logic must detect the fact that the "wrong" paths are going to be saturated and that the "best" path is growing over a preestablished threshold. It must be noted that at low SNRs the estimated "best path" may actually be in error due to the channel noise, but as will be shown later, the fact that bit errors occur does not seem to disturb the metric accumulation rates. A *normalization* occurs when the logic detects that all the "wrong" paths have a high metric ($m >> 4$) and that the best path has just reached or surpassed a threshold of 4. At this time all the buffers are reduced (normalized), and this fact (normalization) is registered in a counter. The normalization counter is made available to the operational program through the SSA coupler.

2. Theoretical model. Figure 6 represents two superimposed Gaussian probability density curves $p(Z)$ that correspond to the integrations of the telemetry symbols performed by the SSA. These integrations will have averages of M_0 and M_1, respectively. The channel noise causes the integrated output to vary around M_0 or M_1 with a normal distribution. A positive integration corresponds to a logical "0" while a negative integration represents a logical "1." The dotted lines in Fig. 6 represent the quantization intervals Q_i and reproduce the intervals shown in Fig. 2. Therefore, any voltage at the integrator output, when sampled at the end of a symbol time, represents the integration of a symbol and this sampled value is converted into a binary number. A symbol error occurs when a "0" falls on the negative side; or alternatively, when a "1" falls on the positive side. In other words, if for instance a "1" is sent through the channel and is received as any of the quantized values that correspond to (Q_0, Q_1, Q_2, or Q_3), a symbol error has occurred.

Regarding Fig. 6, it must be noted that the curves are assumed to be normalized ($\sigma = 1$) and the random variables or output voltages are expressed as Z from the respective origins M_0 and M_1. From all these considerations we can express the symbol error probability as:

$$P_e(\text{"0"}) = \int_{-\infty}^{-M_0} p(Z')\, d_Z' \text{ for a "0"}$$

$$P_e(\text{"1"}) = \int_{M_1}^{\infty} p(Z)\, d_Z \text{ for a "1"}$$

Since the two superimposed curves are symmetrical and two consecutive symbols are independent of each other, our analysis can be focused on the equivalent representation shown in Fig. 7, which will be valid either for a "0" or a "1."

It was stated in Subsection III-B-1 that the normalization mechanism is mainly governed by the metric evolution along the "best path."

For a normalization to occur, metrics (positive values) must equal or be greater than 4 in the best path. But at the same time this metric accumulation must represent a symptom that there is a certain degree of difficulty when trying to decode the received data. In other words, the normalization rate must behave so as to give an estimate of the channel noise level, which is equivalent to estimating symbols in error. Intuitively, at each bit time the global metric along the best path must increase when the hypothesized branch symbols have corresponding symbol quantizations of opposite sign (see Fig. 7). The higher the excursion over the error side, the higher the metric increment. It follows then that the criteria for selecting the metric values introduce a weight for each quantization interval that is inversely proportional to its probability of occurrence. The metric criteria used in the LV7015 model are summarized in Table 1.

For instance, if (1, 1) is the hypothesized branch symbol pair and the received quantization pair is (5, 1) (see Fig. 2) then the metric increment in this bit time for all paths having this particular hypothesis will be (0 + 2): 2. In general, in terms of separate symbols, for a hypothetical symbol X the metric increment will be $m(Q_i/X)$ when Q_i is the corresponding quantized value. The values are tabulated in Table 1.

It should now be clear that from the normalization point of view we are interested only in those increments such that

$$m(Q_i/X) \neq 0$$

and which affect the "best path" (wrong paths are assumed to be saturated or nearly saturated). These increments are precisely the weights that have been assigned to the quantized symbol values (Fig. 7).

The above considerations were verified for the LV7015 unit through simulation by inserting specific and repetitive errors into an otherwise error-free data stream, and by observing the normalization counter.

As a summary it will thus be assumed that:

(1) The best path metric is a direct consequence of the channel symbol error rate.

(2) The wrong-path metrics build up very quickly (for a moderate error rate once the decoder is in sync) while the "best path" builds up its metric at a much lower rate.

(3) The normalization rate is due to the necessity of normalizing the best path global metric thus avoiding values over or equal to 4.

Important: The present approach is valid and intuitive for a moderate symbol error rate but must certainly be verified to hold for low values of ST_S/N_0 where the noisy channel causes bit errors at the decoder output and "wrong paths" become more "competitive."

A mathematical model was elaborated from assumptions (1), (2), and (3), above and was then compared with the experimental results, especially at low ST_S/N_0 near the decoder threshold.

3. Mathematical model. In Fig. 7 it is seen that a metric increment $m(Q_i)$ will be added to the global metric with relative frequency p_i.

In general the normalization per symbol pair (i.e., per bit) for an arbitrary path length of n symbols can be expressed by:

$$\text{normalization} \times \text{bit}^{-1} = N_b = 1/4 \left(2 \times \frac{\text{global metric along best path}}{n \text{ symbols}}\right)$$

(where the numerator is the best path global metric that would be subsequently accumulated if no normalizations occurred). The average value of N_b is

$$\overline{N}_b = 1/2 \sum_j p_j \cdot m(Q_j) \qquad j = 1, 2, 3, 4 \qquad (3)$$

From Fig. 7,

$$p_1 = \int_{Z_0}^{Z_1} p(Z)\, d_Z = P_e(Z_0) - P_e(Z_1)$$

$$p_2 = \int_{Z_1}^{Z_2} p(Z)\, d_Z = P_e(Z_1) - P_e(Z_2)$$

$$p_3 = \int_{Z_2}^{Z_3} p(Z)\, d_Z = P_e(Z_2) - P_e(Z_3)$$

$$p_4 = \int_{Z_3}^{\infty} p(Z)\, d_Z = P_e(Z_3)$$

Substituting in Eq. (3):

$$\overline{N}_b = 1/2(P_e(Z_o) + P_e(Z_1) + P_e(Z_2) + P_e(Z_3))$$

where $P_e(Z_i)$ represents the error probability which corresponds to variable Z_i

$$P_e(Z_i) = \int_{Z_i}^{\infty} p(Z)\, dZ$$

An alternate expression for Eq. (3) is:

$$\overline{N}_b = 1/2 \sum_i \operatorname{erf} Z_i \qquad i = 0,1,2,3$$

For practical purposes it is interesting to express the above relationship in terms of ST_S/N_0 and also in terms of the parameter K, which was introduced in Subsection III-A.

$$K = \frac{M}{Q} \left\} \begin{array}{l} \text{M = integration mean} \\ \text{Q = quantization interval} \end{array} \right.$$

$$\text{But } Z_i = M + i.Q \text{ or } Z_i = M\left(1 + \frac{Q.i}{M}\right)$$

$$Z_i = M\left(1 + \frac{i}{K}\right) \quad \text{for } i = 0, 1, 2, 3$$

Since

$$\left(\frac{Z_i}{Z_0}\right)^2 = (1 + i/K)^2$$

it follows that

$$(ST_{Si}/N_0 - ST_0/N_0)\text{ dB} = 20 \log(1 + i/K);$$
$$i = 0, 1, 2, 3$$

Equation (3) can thus be written:

$$\overline{N}_b = 1/2 \Bigg[+ P_e(ST_{S0}/N_0) + P_e\left(ST_{S0}/N_0 + 20 \log \left(1 + \frac{1}{K}\right)\right) + P_e\left(\frac{ST_{S0}}{N_0} + 20 \log \left(1 + \frac{2}{K}\right)\right) + P_e\left(\frac{ST_{S0}}{N_0} + 20 \log \left(1 + \frac{3}{K}\right)\right)\Bigg] \qquad (4)$$

where $P_e(X)$ represents the probability of symbol error at an ST_{S0}/N_0 of X dB, which is the actual channel energy per symbol-to-spectral density, and corresponds to the value Z_0.

Thus, expression (4) relates the normalization rate per bit to the (ST_S/N_0)dB in the channel and also includes the parameter K, which will determine the optimum calibration point in the SSA integrator.

Equation (4) has been numerically determined for a wide range of (ST_S/N_0)dB and three values of K (see Table 2 and Fig. 8).

At this point several considerations arise:

(1) For an $ST_S/N_0 \geq 5$ dB, the mathematical mode may be approximated by:

$$\overline{N}_b \approx 1/2\, P_e$$

where P_e is the channel symbol error probability. This can be justified by the fact that at these values of ST_S/N_0 the excursions beyond Z_1 (Fig. 7) have very low probability of occurring.

(2) $\overline{N}_b$ is a monotonic function of K for constant ST_S/N_0 and is bounded below by 1/2 P_e as already stated in (1).

(3) There is no considerable influence of K on the normalization rate.

Now the question arises: will the model depart from experimental results at low ST_S/N_0 when the bit error rate is significant? How will the model be adapted to the system operating conditions? To answer these questions a series of tests was run with a standard configuration, sampling the normalization counter for different ST_S/N_0 and assuming a value of K corresponding to the optimum SDA/SSA setting of 280 mV.

4. An algorithm for performance evaluation. The mathematical model developed in the preceding section seems to be accurate enough to constitute the basis for a performance evaluator. The requirements may be summarized as a need to evaluate decoder performance (or the system performance) from the normalization rate. A convenient performance estimator could be one that would evaluate the bit error rate or, more simply, the energy per bit-to-spectral noise density. In our case the problem will be reduced to relating the normalization counter values to the bit energy-to-spectral noise density $(ST_b)/N_0$ dB.

Let expression (4) be represented by:

$$\overline{N}_b = f(ST_S/N_0)$$

where parameter K has been substituted by its optimum numerical value. Since $\overline{N}_b$ is the actual variable and the image should be ST_b/N_0 rather than ST_S/N_0, we first must find f^{-1} ( ) and then introduce the change

$$(ST_b/N_0)\text{ dB} = (ST_S/N_0)\text{ dB} + 3\text{ dB}$$

where it has been assumed that the code rate (symbols per bit) is 2 and that all the degradation effects will be reflected by the normalization rate with no impact on the symbol-to-bit relationship.

Therefore, f^{-1} ( ) was first determined by graphically finding the inverse of $f($ $)$, and then the corresponding values of ST_S/N_0 were incremented by 3 dB. However, since the normalization counter is transferred to the operational program every 192 bits, it was thought that it would be preferable to use normalization counts $(\overline{N}_c)$ instead of the normalization rate $(\overline{N}_b)$ as the variable. Thus, a final change was made where

$$\overline{N}_c = 192 \times \overline{N}_b$$

and finally

$$(ST_b/N_0)\text{ dB} = g(\overline{N}_c)$$

was obtained. As stated previously, f^{-1} ( ) was found graphically. For the practical purposes of using the relationship as a computerized algorithm, a polynomial was fitted to the numerical values of $g($ $)$ by using the least squares criterion. The final expression adopted for the algorithm was then:

$$(ST_b/N_0)\, dB = \frac{2.9664}{\overline{N}_c + 0.08} + 5.1218 - 0.2252\,\overline{N}_c \qquad (5)$$

This expression will, therefore, convert the normalization counts as transferred from the decoder into the corresponding channel (ST_b/N_0) dB.

Figure 9 is a plotting of expression (5) and is compared to the values of g( ). The fit has an error lower than 0.3 dB in the range $1 < \overline{N}_c < 15$.

Although g( ) could theoretically reach very high values for values of $\overline{N}_c$ in the vicinity of 0, this is an impractical approach since the corresponding high values of ST_b/N_0 are useless. The criterion was then adopted to saturate the conversion on the order of 35 dB. At the same time the fit accuracy is worse at that end because of the low ST_b/N_0 values. Note that the statistics become very poor at ST_b/N_0 over 7 dB since extremely few normalizations will occur and the conversion into ST_b/N_0 becomes less relevant.

Finally, Fig. 10 is a plotting of BER versus the corresponding values of ST_b/N_0 measured with the algorithm directly from the normalization counter. Also shown is $\overline{N}_b$ versus ST_b/N_0.

5. Conclusions regarding normalization rate. The normalization rate mechanism has been analyzed in terms of its capability to provide a system performance evaluation. In summation, it has been found that:

(1) *The normalization rate behavior is similar to channel symbol error behavior.* In fact, at values of ST_S/N_0 about or above 5 dB, both functions may be closely approximated and differ only by a constant.

(2) *$\overline{N}_b$ is quite insensitive to the presence of decoded bits in error and does not significantly depart from the theoretical model.*

(3) When compared to the bit error probability, the normalization function varies at a much lower rate. The $\overline{N}_b$ global derivative function is considerably lower than the BER global derivative and, therefore, a significant BER increment will not be reflected by a significant $\overline{N}_b$ change.

(4) $\overline{N}_b$ is a monotonic function of K where K was defined in Eq. (1) and represents a basic parameter in the SDA/SSA setup. The lower the value of K, the lower the normalization rate. However, it does not follow that the lower the K the lower the BER. *Consequently, $\overline{N}_b$ cannot be used to determine the optimum channel operating conditions.*

(5) A relationship between ST_b/N_0 and $\overline{N}_c$ was found from a theoretical model based on heuristic considerations. The mathematical model and the experimental results seem to match quite well. However, from (1) to (2) above it follows that the evaluation algorithm is useful only if the system has been properly calibrated or when any existing degradation directly affects the channel error probability by a measurable amount. For a given ST_b/N_0, an incorrect calibration (typically a higher MI setting) *may yield such a $\overline{N}_b$ as to induce an estimated BER better than the actual value.*

Therefore, it must be concluded that the system performance evaluator algorithm derived from the normalization rate is valid under certain conditions. It will be further analyzed and evaluated during the forthcoming testing phase as described in Section VIII.

IV. Differences Between the MCD and the LV7015

As previously stated, there are two minor differences in the functional capability between the DSN Maximum Likelihood Convolutional Decoder being implemented in the Network and the commercially available LV7015 Viterbi decoder. First, the MCD inverts alternate coded symbols to assure sufficient transition density, while the LV7015 does not; second, the MCD has a memory path length of 64 bits, while the LV7015 has a 36-bit memory path length.

The first difference is insignificant. The second difference results in an improvement in decoder performance for the longer memory path length. Based on Linkabit reports and a simulation in which over 200,000 bits were decoded, the difference in performance for the 64-bit memory over the 36-bit memory is undetectable at error rates of 10^{-5} (Ref. 1). Consequently, the results reported here are not affected since these tests are designed to investigate performance at this error rate.

Testing using the LV7015 to simulate the MCD at error rates of 10^{-3} will have to be adjusted by about 0.1 dB.

V. Test Plan

Table 3 lists the tests to be carried out. This general test plan includes Block III and Block IV receivers, different SDA bandwidths, and different modulation indexes. Those modulation indexes and the signal power levels included in the plan intend to cover a wide operating range to determine the optimum channel utilization of the MJS-77 mission.

The test objectives are aimed at the evaluation of the bit error characteristic at the decoder output. Namely:

(1) Estimate the bit error rate.

(2) Determine the statistical behavior of the error bursts.

(3) Evaluate the estimated channel SNR from the normalization rate scheme.

VI. Bit Error Characteristics

The Viterbi decoding algorithm does not proceed on a per block basis like the Fano algorithm and does not reconsider past bit decisions. The decoded bits may be in error in a certain path length and yet be able to remerge with the good path at a certain mode and remain correct thereafter. Therefore, the decoder always proceeds forward and may depart from the correct path occasionally, depending on channel noise characteristics. The symbol errors occur in bursts whose characteristics are to be determined for the testing conditions. The burst approach suggests two definitions:

(1) An "error burst" (burst) is defined as a sequence of decoded bits that begins with a bit in error, ends with a bit in error, and has fewer than 6 consecutive correct bits within the burst. In general for a constraint length of K, the number of consecutive correct bits within the burst must be K-2 at most.

(2) An "error free run" (runs) is a sequence of consecutive correct bits including correct bits within a burst. Even runs of length 0 (two consecutive errors) are considered.

The main point thereafter is to identify the bits in error, proceed with their classification into bursts and then analyze the clustering of errors and correct bits within the bursts.

To handle the previously exposed conditions, the following approach was chosen. A bit error pattern is obtained by direct comparison of the original data and the decoded bits. Therefore, the ones in this pattern represent bits in error in the decoded data. Instead of operating directly with this binary error pattern, a preprocessing step is first carried out. The number of consecutive ones and consecutive zeros in the error pattern are counted. The binary pattern is consequently converted into a sequence of integers where the terms of *odd* order correspond to consecutive *ones* in the error pattern, while terms of *even* order correspond to consecutive *zeros* in the error pattern. This preprocessing greatly reduces the data storage required, the search time, and also simplifies the statistical evaluation.

Given then an error pattern of the form:

$$\cdots \underbrace{1 \cdots 1}_{K_1} \quad \underbrace{0 \cdots 0}_{K_2} \quad \underbrace{1 \cdots 1}_{K_3} \quad \underbrace{0 \cdots 0}_{K_4}$$

the corresponding sequence of integers would be

$$\cdots K_1, K_2, K_3, K_4 \cdots \qquad K_i \epsilon N$$

In general then, the runs of ones and the runs of zeros are transformed into a sequence of integers

$$(a_i)_i \epsilon N$$

where the subsequence of odd terms

$$(a_i)_i = 2K - 1 \qquad K = 1, 2 \cdots$$

represents the respective lengths of the runs of *ones*, and the subsequence of even terms

$$(a_i)_i = 2K \qquad K = 1, 2 \cdots$$

represents the respective lengths of the runs of zeros.

All the considerations concerning error bursts or runs of correct bits (zeros in the errror pattern) are done directly from the sequence $(a_i)_{i\epsilon N}$. For instance, an error burst always starts with a term of subindex

$$i = (2K_0 - 1) \qquad \text{if } a_2(K_0 - 1) \geq 6$$

This would mean that $a_{2(K_0-1)}$ is not a run within a burst.

The burst will end when a run of zeros (from burst start) equals or exceeds 7; that is, the end of a burst occurs at certain K_1 for which

$$a_{2K_1} \geq 6$$

Then for a burst beginning with term

$$a_{(2K_0 - 1)}$$

and ending at term $a_{(2K_1-1)}$ with $K_0 < K < K_1$, those terms of the form a_{2K} for $K_0 < K < K_1$ must be

$$a_{2K} < 6$$

We may identify a burst by the pair

$$(K_1, K_2)$$

where $a_{2(K_1-1)}$ is the first term in the burst sequence, and $a_{2(K_2-1)}$ is the last term in the burst sequence. Note that the burst is 1 bit long, and $K_1 = K_2$. The number of ones in a burst (K_1, K_2) may be found by

$$\sum_{K_1}^{K_2} a_{2K-1}$$

For the number of zeros in a burst (K_1, K_2)

$$\sum_{K_1}^{K_2-1} a_{2K}$$

The total burst length is $(2K_2 - 1) - (2K_1 - 1) + 1 = 2(K_2 - K_1) + 1$. Any K defines a pair of terms of (a_i) where

$$a_{2K-1} \equiv \text{a run of ones}$$

$$a_{2K} \equiv \text{a run of zeros}$$

For any K such that $a_{2K} \geq 6$, a_{2K-1} is the end of a burst and a_{2K+1} is the beginning of a burst. For instance, for a pattern:

1,000000001101000000111110000000 • • •

$(a_i) \equiv$ 1, 8, 2, I, 1, 5, 5, 7 • • •

For $K = 1$, $a_2 > 6$ then a_1 is the end of a burst, a_2 is a run out of a burst and a_3 the beginning of a new run. Then

burst (1, 1) $\equiv$ 1, number of ones $=$ 1, number of zeros $=$ 0

burst (2, 4) $\equiv$ (2, 1, 1, 5, 5)

burst length $\equiv 2(4 - 2) + 1 = 5$ [in terms of elements of (a_i)]

$$\text{number of ones} = \sum_{2}^{4} a_{2K-1} = 2 + 1 + 5 = 8$$

$$\text{number of zeros} = \sum_{2}^{4-1} a_{2K} - 1 + 5 = 6$$

VII. Testing Program Algorithms

The program will analyze the previously presented sequence (a_i) and other complementary parameters within the high-speed data blocks to provide:

(1) The test signal level average and standard deviation in dBm, and the number of samples.

(2) The SSA ST_B/N_0 in dB (average and σ) and number of samples.

It must be noted that this is the standard value processed by the SSA and is biased for values below 4 dB. This should be taken into account when comparing this parameter to the ST_B/N_0 derived from $\overline{N}_c$.

(3) Normalization per bit value $\overline{N}_b$ (average and σ) and number of samples.

(4) The runs length average and σ, i.e., the statistic on the correct bits, and the total number of runs processed. (Note that correct bits within a burst are also considered.)

(5) The burst length average and σ, i.e., the statistic on the error bursts and the total number of bursts processed.

(6) The ST_b/N_0 (dB) estimation from the normalization count $\overline{N}_c$. The program computes:

$$\frac{ST_b}{N_0}\,\text{dB} = g(\overline{N}_c)$$

as

$$\frac{ST_b}{N_0}\,(\text{dB}) = \frac{a}{\overline{N}_c + K} + b + c\overline{N}_c$$

where

$\overline{N}_c = 192\ \overline{N}_b$ (from item 3 above) and constants a, b, c, specified in Subsection III-B-4.

An estimate of $\sigma(ST_b/N_0)$ is obtained by computing:

$$\left.\begin{array}{l} m_1 = (\overline{N}_b + \sigma_{N_b})\,192 \\ m_2 = (\overline{N}_b - \sigma_{N_b})\,192 \end{array}\right\} \frac{ST_b}{N_0}$$

$$= \frac{|g(\overline{N}_c) - g(m_1)| + |g(\overline{N}_c) - g(m_2)|}{2}$$

(7) The total number of bits processed.

(8) The total burst length i.e., $\sum$ (burst lengths) in bits.

(9) The bit error rate i.e., the relationship:

$$\text{BER} = \frac{\text{ones in the error pattern}}{\text{total no. of bits processed}}$$

(10) The probability that a bit within a burst is in error. Note that since runs of zeros in the error pattern of length $R < 6$ are possible, these bits are actually correct bits within a burst.

(11) The bit error in the average burst length, that is, average burst length × bit error in a burst.

(12) Finally the program tabulates the following statistics:

For bursts (B):

The density function $p(B = L) \equiv$ relative frequency of a burst of length $= L$ ($L = 1, 2, 3, \cdots, 100$), and for bursts of length L: the average number of error bits per burst and their standard deviation. The distribution function $p(B \geq L)$.

For runs (R):

The density function $p(R = L) \equiv$ relative frequency of a run $= L$ ($L = 0, 1, 2, \cdots$), and the distribution function $p(R \geq L)$.

VIII. General Conclusions

The statistical analysis of the bit error patterns is performed in terms of bursts and runs. The system characteristics will then be determined through the statistical behavior of these parameters. Also, the system performance is evaluated with the normalization rate parameter that has been developed in Section III.

This preliminary report does not consider actual results that will be included in a future document. Also, at the time of this report, the software program is already in use for test evaluation, but still requires modification in a few operating procedures. A full program description and handling procedure will also be presented in a future report.

Reference

1. Benjauthrit, B. D., Mulhall, D. L., and Wong, J. S., "A Viterbi Decoding Program for DSN Telemetry System Analysis," in *The Deep Space Network Progress Report 42-28*, pp. 5–10, Jet Propulsion Laboratory, Pasadena, Calif., Aug. 15, 1975.

Bibliography

Gilhousen, K. S., "Coding Systems Study For High Data Rate Telemetry Links," Linkabit Corp., NASA CR-114278, Jan. 1971.

Heller, J., and Jacobs, I., "Viterbi Decoding For Satellite And Space Communication," *IEEE Trans. Commun. Tech.*, Vol. COM-19, No. 5 Oct. 71.

"Instruction Manual LV7015 Convolutional Encoder Viterbi Decoder," Linkabit Corp., 1972.

Maximum Likelihood Convolutional Decoder MCD," Technical Requirement Document 338-256, 10 Jul 1974 Rev B. 12-13-74. Jet Propulsion Laboratory, Pasadena, Calif., July 10, 1974 (Rev. B, Dec. 13, 1974) (an internal document).

"Telemetry Coding Study for the International Magnetosphere Explorers," NASA-CR-132880, Vol. I, Sections 3.2.1 through 3.2.4.

Viterbi, A., "Convolutional Codes and Their Performance In Communications Systems." *IEEE Trans. Commun. Tech.*, Vol. COM-19, No. 5 Oct. 71.

Weber, C. L., "Notes On Convolutional Codes," Gaylord K. Huth, Axiomatic Corp.

Table 1. Metric criteria

Symbol	$m(Q_3)$	$m(Q_2)$	$m(Q_1)$	$m(Q_0)$	$m(Q_4)$	$m(Q_5)$	$m(Q_6)$	$m(Q_7)$
0	0	0	0	0	1	2	3	4
1	4	3	2	1	0	0	0	0

Table 2. Signal-to-noise ratio vs decoder normalization rate

dB	Z	$P_e(Z)$	N_b		
			$K=2$	$K=2.5$	$K=3.0$
−1.5	1.189912	0.117040	0.082154	0.092719	0.102841
−1.25	1.224658	0.110352	0.075857	0.085422	0.094688
−1.00	1.260419	0.103759	0.069882	0.078498	0.086935
−0.75	1.297224	0.097277	0.064221	0.071942	0.079585
−0.50	1.335104	0.090921	0.058870	0.065752	0.072636
−0.25	1.374089	0.084707	0.053822	0.059923	0.066088
0.00	1.414214	0.078650	0.049070	0.054447	0.059936
0.25	1.455509	0.072764	0.044605	0.049317	0.054175
0.50	1.498011	0.067065	0.040421	0.044524	0.048797
0.75	1.541754	0.061567	0.036510	0.040059	0.043794
1.00	1.586774	0.056282	0.032862	0.035913	0.039155
1.25	1.633108	0.051223	0.029470	0.032073	0.034868
1.50	1.680796	0.046401	0.026325	0.028530	0.030922
1.75	1.729876	0.041826	0.023418	0.025271	0.027303
2.00	1.780389	0.037506	0.020741	0.022284	0.023997
2.25	1.832378	0.033448	0.018283	0.019558	0.020990
2.50	1.885884	0.029655	0.016036	0.017079	0.018266
2.75	1.940953	0.026132	0.013989	0.014835	0.015810
3.00	1.997630	0.022878	0.012113	0.012814	0.013606
3.25	2.055962	0.019893	0.010439	0.011001	0.011639
3.50	2.115997	0.017173	0.008940	0.009385	0.009893
3.75	2.177785	0.014711	0.007603	0.007951	0.008352
4.00	2.241377	0.012501	0.006418	0.006680	0.007000
4.25	2.306827	0.010532	0.005374	0.005556	0.005822
4.50	2.374187	0.008794	0.004461	0.004596	0.004803
4.75	2.443515	0.007272	0.003665	0.003767	0.003923
5.00	2.514866	0.005953	0.003017	0.003086	0.003183
5.25	2.588302	0.004822	0.002437	0.002484	0.002554
5.50	2.663882	0.003862	0.001947	0.001979	0.002029
5.75	2.741668	0.003056	0.001538	0.001559	0.001593
6.00	2.821727	0.002388	0.001200	0.001213	0.001236
6.25	2.904122	0.001841	0.000924	0.000932	0.000948
6.50	2.988924	0.001399	0.000701	0.000707	0.000717
6.75	3.076203	0.001048	0.000525	0.000528	0.000534
7.00	3.166029	0.000772	0.000386	0.000388	0.000392
7.25	3.258479	0.000560	0.000280	0.000281	0.000283
7.50	3.353628	0.000398	0.000199	0.000200	0.000201
7.75	3.451556	0.000278	0.000139	0.000139	0.000140
8.00	3.552343	0.000190	0.000095	0.000095	0.000096
8.25	3.656074	0.000128	0.000064	0.000064	0.000064
8.50	3.762833	0.000083	0.000042	0.000042	0.000042
8.75	3.872710	0.000053	0.000027	0.000027	0.000027
9.00	3.985795	0.000033	0.000016	0.000016	0.000016
9.25	4.102182	0.000020	0.000010	0.000010	0.000010
9.50	4.221968	0.000012	0.000006	0.000006	0.000006
9.75	4.345252	0.000006	0.000003	0.000003	0.000003
10.00	4.472135	0.000003	0.000002	0.000002	0.000002

Table 3. Test conditions

Run	Symbol/sec, SPS (BPS)	RCV BLK No., BW	SDA BLK No., BW	P_T, N_0 db	E_b/N_0 db	Mod index, deg	Carrier suppression, db	Duration of test, h	No. of bits × 10^6	No. of bit errors expected	Max. No. [a] of bit bursts expected
M1	5120 (2560)	III, N 12 Hz	III, N	39.67	3.8549	55	4.828	1.0	9.216	460.8	121.26
2					4.7331	65	7.481	1.0	9.216	460.8	121.26
3					4.9906	69	8.913	1.0	9.216	460.8	121.26
4					5.0473	70	9.319	2.0	18.432	921.6	242.42
5					5.0799	70.6[b]	9.573	2.0	18.432	921.6	242.42
6					5.1010	71	9.747	2.0	18.432	921.6	242.42
7					5.1517	72	10.200	1.0	9.216	460.8	121.26
8					5.1995	73	10.681				
9					5.2444	74	11.193				
10					5.3257	76	12.326				
11					5.3957	78	13.642				
12					5.4546	80	15.207			460.8	
N1	5120 (2560)	IV, N 3 Hz	IV, N	39.67	3.8549	55	4.828			921.6	
2					4.7331	65	7.481			921.6	
3					4.9906	69	8.913	1.0	9.216	921.6	121.26
4					5.0473	70	9.319	2.0	18.432	460.8	242.42
5					5.0799	70.6[b]	9.573	2.0	18.432	460.8	242.42
6					5.1010	71	9.747	2.0	18.432	460.8	242.42
7					5.1517	72	10.200	1.0	9.216	921.6	121.26
8					5.1995	73	10.681				
9					5.2444	74	11.193				
10					5.3257	76	12.326				
11					5.3957	78	13.642				
12					5.4546	80	15.207	1.0	9.216	921.6	121.26
G1	5120 (2560)	12 Hz	III, M	40.26	4.4449	55	4.828	1.0	9.216	460.8	121.26
2					5.3231	65	7.481				
3					5.5806	69	8.913				
4					5.6373	70	9.319	1.0	9.216	460.8	121.26
5					5.6910	71	9.747	5.0	46.08	2304	606.32
6					5.7417	72[b]	10.200	5.0	46.08	2304	606.32
7					5.7895	73	10.681	5.0	46.08	2304	606.32
8					5.8344	74	11.193	1.0	9.216	460.8	121.26
9					5.8765	75	11.740				
10					5.9857	78	13.642				
11					6.0446	80	15.207				
H1	5120 (2560)	12 Hz	III, N	39.67	3.8549	55	4.828		9.216		
2					4.7331	65	7.481		9.216		
3					4.9906	69	8.913	1.0	9.216	460.8	121.26
4					5.0473	70	9.319	5.0	46.08	2304	606.32
5					5.0799	70.6[b]	9.573	5.0	46.08	2304	606.32
6					5.1010	71	9.747	5.0	46.08	2304	606.32
7					5.1517	72	10.200	1.0	9.216	460.8	121.26
8					5.1995	73	10.681				
9					5.2444	74	11.193				
10					5.3257	76	12.326				
11					5.3957	78	13.642				
12					5.4546	80	15.207				
I1	5120 (2560)	12 Hz	III, M	39.77	3.9549	55	4.828	1.0	9.216	460.8	121.26

Table 3 (contd)

Run	Symbol/sec, SPS (BPS)	RCV BLK No., BW	SDA BLK No., BW	P_T, N_0 db	Eb/N_0 db	Mod index, deg	Carrier suppression, db	Duration of test, h	No. of bits × 10^6	No. of bit errors expected	Max. No. [a] of bit bursts expected
12	5120 (2560)	12 Hz			4.8331	65	7.481	1.0	9.216	460.8	121.26
3					5.0906	69	8.913	1.0	9.216	460.8	121.26
4					5.1473	70	9.319	5.0	46.08	2304	606.32
5					5.1852	70.7[b]	9.616	5.0	46.08	2304	606.32
6					5.2010	71	9.747	5.0	46.08	2304	606.32
7					5.2517	72	10.200	1.0	9.216	460.8	121.26
8					5.2995	73	10.661				
9					5.3444	74	11.193				
10					5.4257	76	12.326				
11					5.4957	78	13.642				
12					5.5546	80	15.207	1.0	9.216	460.8	121.26
J1	5120 (2560)	12 Hz	III, W	39.82	4.0049	55	4.828	1.0	9.216	460.8	121.26
2					4,8831	65	7.481	1.0	9.216	460.8	121.26
3					5.1406	69	8.913	5.0	46.08	2304	606.32
4					5.2405	70.8[b]	9.660	5.0	46.08	2304	606.32
5					5.2510	71	9.747	5.0	46.08	2304	606.32
6					5.3017	72	10.200	1.0	9.216	460.8	121.26
7					5.3495	73	10.681				
8					5.3944	74	11.193				
9					5.4757	76	12.326				
10					5.5457	78	13.642				
11					5.6046	80	15.207				
K1	5120 (2560)	12 Hz	III, M	41.26	5.4449	55	4.828				
2					6.3231	65	7.481	1.0	9.216	460.8	121.26
3					6.7417	72	10.200	5.0	46.08	2304	606.32
4					6.8123	73.5[b]	10.933		46.08	2304	606.32
5					6.8344	74	11.193		46.08	2304	606.32
6					6.8765	76	11.740	1.0	9.216	460.8	121.26
7					6.9157	76	12.326				
8					6.9521	77	12.958				
9					7.0446	80	15.207	1.0	9.216	460.8	121.26
L1	5120 (2560)	12 Hz	III, M	42.76	6.9449	55	4.828	1.0	9.216	460.8	121.26
2					7.8231	65	7.481				
3					8.2895	73	10.681				
4					9.3344	74	11.193	1.0	9.216	460.8	121.26
5					8.3765	75	11.740	5.0	46.08	2304	606.32
6					8.3845	75.2[b]	11.854	5.0	46.08	2304	606.32
7					8.4157	76	12.326	5.0	46.08	2304	606.32
8					8.4521	77	12.958	1.0	9.216	460.8	121.26
9					8.4857	78	13.642	1.0	9.216	460.8	121.26
10				42.76	8.5446	80	15.207	1.0	9.216	460.8	121.26

[a] Based on Linkabit's average number of errors per burst of 3.8 bits at about BER = 5×10^5.

[b] Theoretical optimal value.

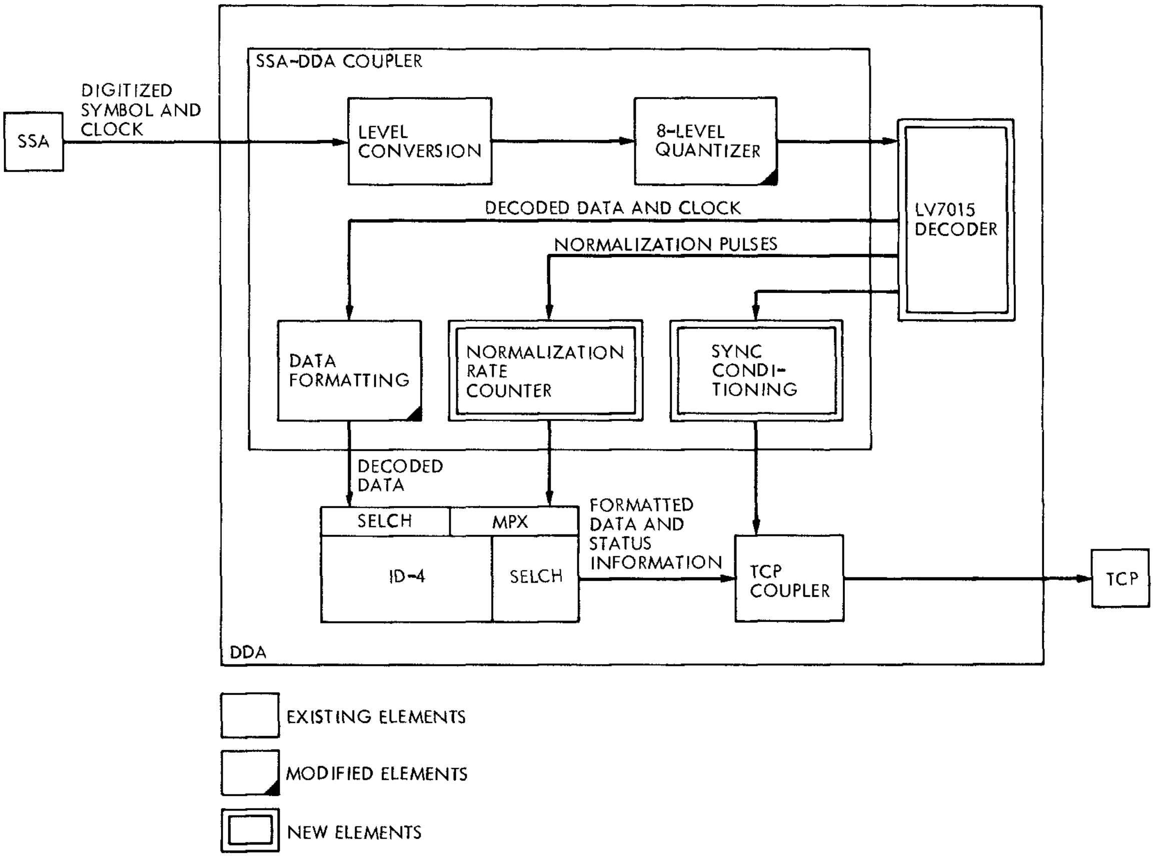

Fig. 1. LV7015 integration

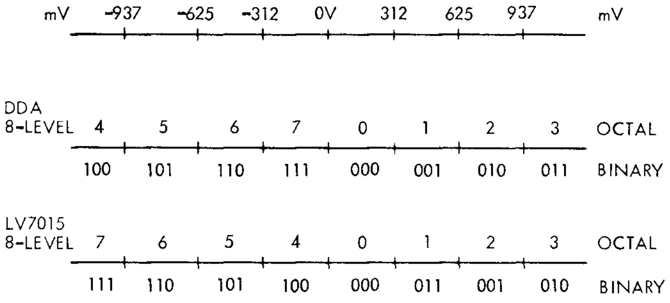
mV -937 -625 -312 0V 312 625 937 mV
DDA 8-LEVEL
4 5 6 7 0 1 2 3 OCTAL
100 101 110 111 000 001 010 011 BINARY
LV7015 8-LEVEL
7 6 5 4 0 1 2 3 OCTAL
111 110 101 100 000 011 001 010 BINARY

Fig. 2. SSA analog-to-digital conversion levels

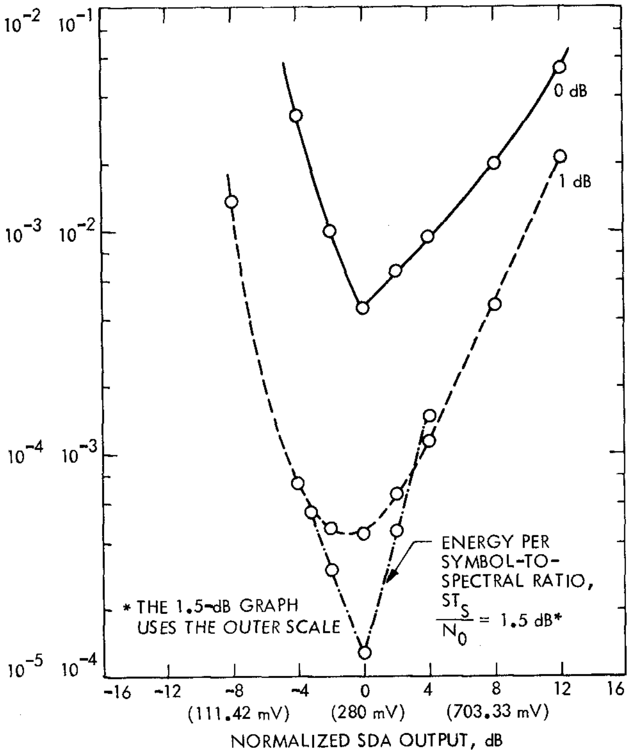

Fig. 3. Bit error rate vs SDA output for ST_S/N_0 = 0, 1, and 1.5 dB

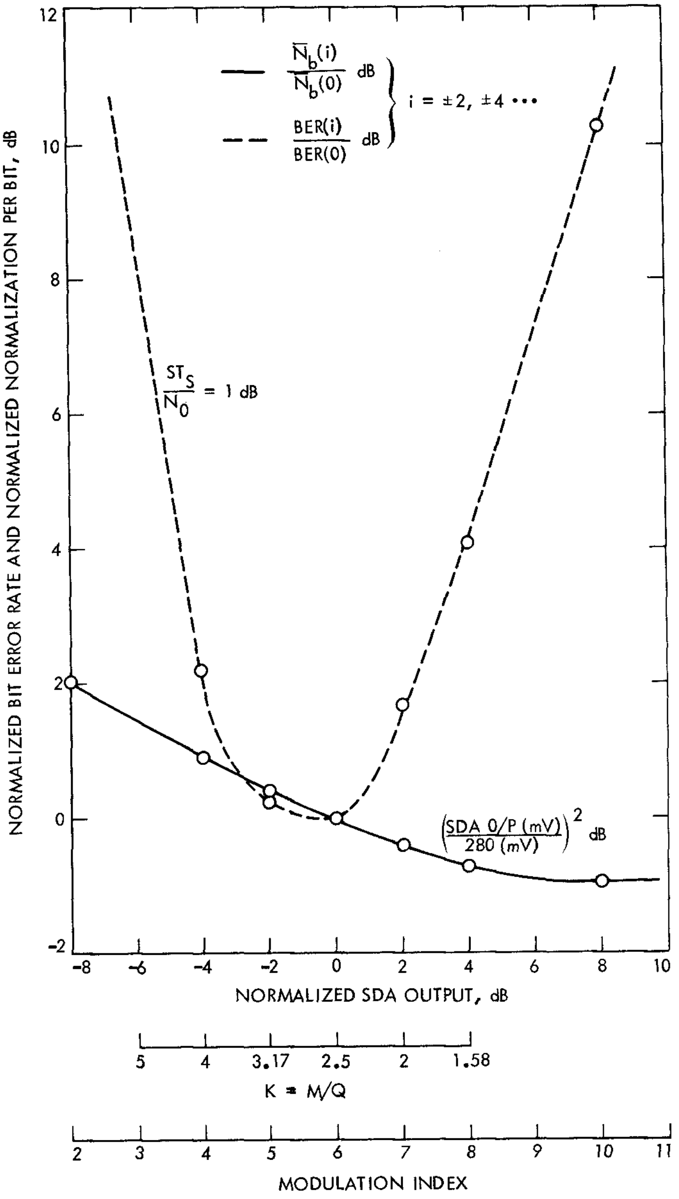

Fig. 4. Bit error rate vs SDA output and normalization rate vs modulation index (ST_S/N_0 = 1 dB)

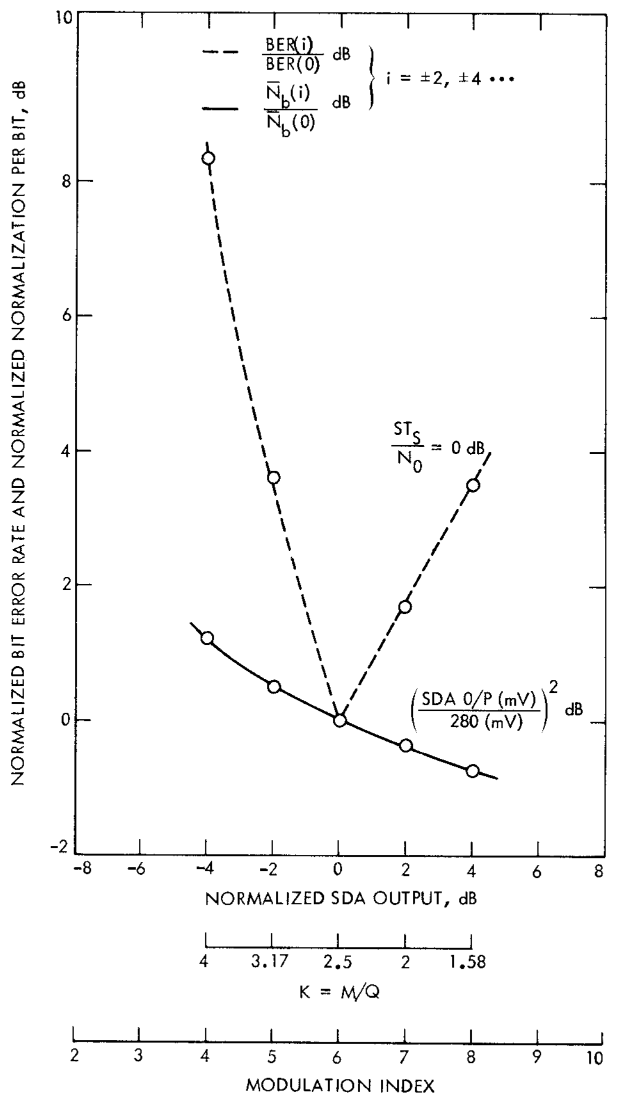

Fig. 5. Bit error rate vs SDA output and normalization rate vs modulation index (ST_S/N_0 = 0 dB)

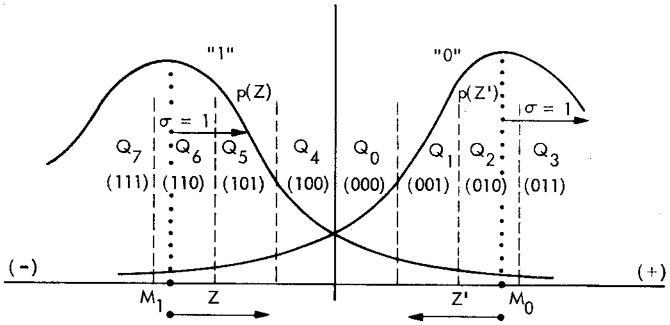

Fig. 6. Two superimposed gaussian probability density curves, p (Z)

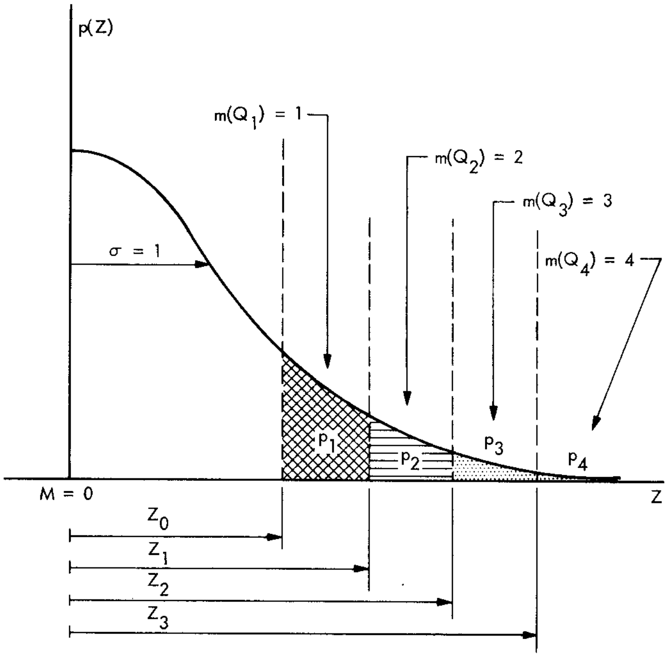

Fig. 7. An equivalent representation of two superimposed gaussian probability density curves

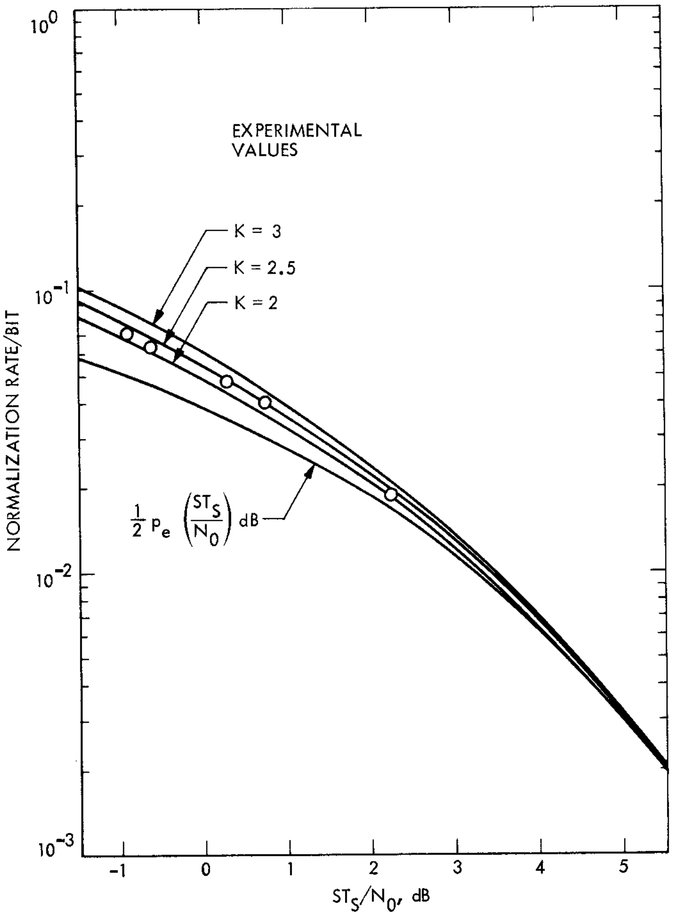

Fig. 8. Normalization rate per bit N_b vs energy per symbol-to-spectral noise density ratio ST_S/N_0, dB

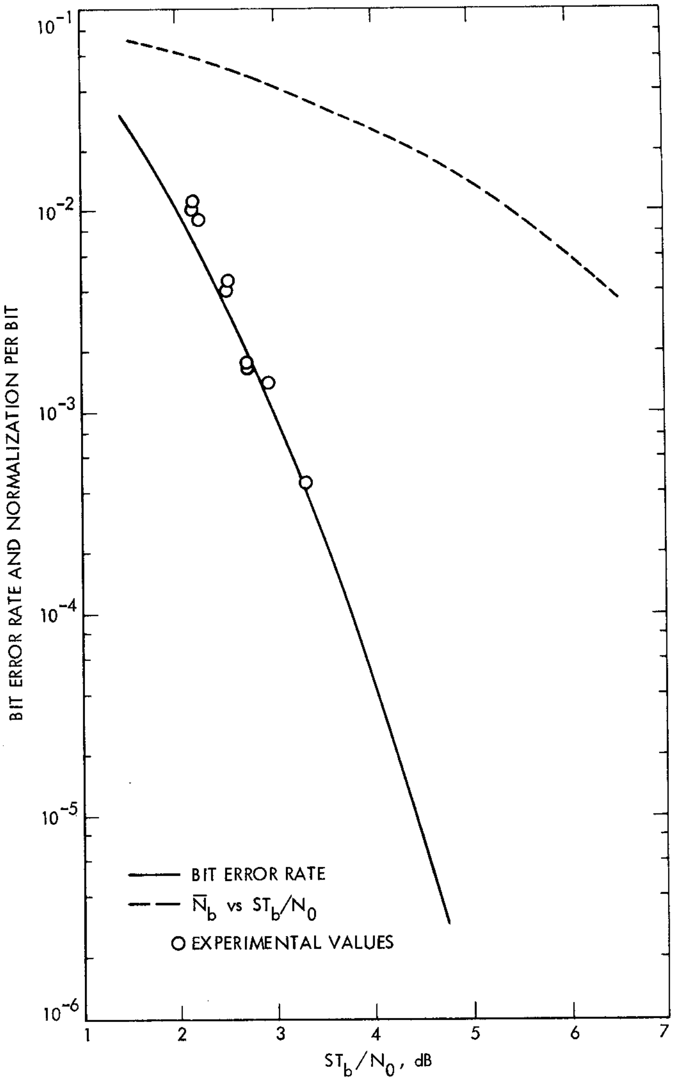

Fig. 10. Baseband performance characteristics of K = 7 rate 1/2 convolutional code, Viterbi coding, soft quantized, Q = 3

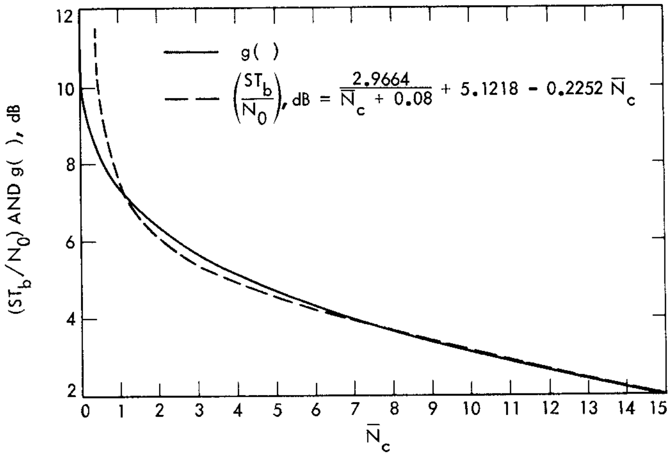

Fig. 9. ST_S/N_0 and g ( ) vs normalization counts N_c