Effects of Quantization on Symbol Stream Combining in a Convolutionally Coded System

F. Pollara and L. Swanson
Communications Systems Research Section

Symbol stream combining has been proposed as a method for arraying signals at different antennas. If the received symbol streams are recorded on tape, it is desirable to limit the required storage without significantly affecting the performance. It is shown that 4-bit quantized symbols introduce an E_b/N_0 penalty of only 0.05 dB.

I. Introduction

Symbol stream combining (Ref. 1) has been shown to be an effective method to increase data return in critical missions (Ref. 2) by arraying possibly distant receiving stations. Received symbols are recorded on tape at the output of two or more Symbol Synchronizer Assemblies (SSAs) and then combined off-line by properly aligning the data on the tapes and forming a weighted sum.

While the SSA is capable of delivering 8-bit (246 levels) uniformly quantized symbols, it is desirable to reduce this information to only 3 to 5 bits per symbol in order to limit the storage required on tape.

This study considers the situation shown in Fig. 1, where two quantized symbol streams are optimally combined by using unquantized weight coefficients. The resulting combined signal is then delivered to the maximum-likelihood convolutional decoder (MCD), which uses a 3-bit quantized input. A complete simulation of the quantized combiner, MCD and additive Gaussian noise channel was developed to measure the bit error rate at the output of the MCD, with different numbers of quantization levels at the combiner input. It is assumed that the two symbol streams have independent additive Gaussian noise components and that they are combined with optimal weights, as described in Ref. 3. It is concluded that 4-bit quantization represents a good compromise between tape storage and performance.

II. Branch Metric Quantization

The existing simulation of the MCD (Viterbi algorithm) did not take into account any branch metric quantization effect. Because the performance of this system can depend heavily on the interaction of combiner and metric quantization, it was necessary to include metric quantization in the simulation. Therefore, before considering the effects of quantization at the input of the combiner, we briefly review the quantization of branch metrics and choose one quantization scheme which we believe to be similar to that used in the DSN decoders.

It is known that optimum 3-bit quantization with uniform step size requires only 0.2 to 0.25 dB more E_b/N_0 than the unquantized case (Ref. 4). Ideally, the branch metric is proportional to the logarithm of the probability that a specific

information bit was transmitted, given a particular pair of received soft symbols from the SSA.

In practice (Ref. 5), the branch metrics are assigned according to one of the tables in Fig. 2, where a small branch metric represents a highly probable event, while larger metrics represent less probable events. Figures 2(a) and 2(b) are two different examples of *linear metric* assignments, while Fig. 2(c) is the so called *square metric*, since it is just a quantized version of the square Euclidean distance of the received symbol pair from the hypothesized correct pair, which is optimum. Note that the metrics in Figs. 2(a) and 2(b) require only a 3-bit representation, while that in Fig. 2(c) requires 4 bits due to only one entry with value 8. The tables are used by selecting the value corresponding to the two quantized symbol values, as shown in Fig. 2. The bit error rate performance of the three schemes of Fig. 2 is shown in Fig. 3. Throughout this study it is assumed that the survivor path memory in the Viterbi decoder is 32 bits long. In the study of Section III, we use the method of Fig. 2(c).

III. Combiner Input Quantization

Consider for simplicity a combiner with two inputs:

$$s_1 = a + n_1$$

$$s_2 = a + n_2$$

where $a = \pm 1$ and n_1, n_2 are two independent, zero mean, Gaussian random sequences. In this model, a is the coded message stream and n_1 and n_2 are the noise sequences due to the respective receivers. We are assuming that there is no gain control error in adjusting the amplitude of the two signals, and thus the a's have no coefficients. Then the combiner output y is the weighted sum:

$$y = cf(s_1) + (1 - c) f(s_2)$$

where $f(\cdot)$ is a nonlinear function representing the quantization to a certain number of bits.

Defining $\lfloor x \rfloor$ as the largest integer less than x, the function $f(x)$ for 8-bit quantization is given by

$$f(x) = \begin{cases} b(\lfloor x/q \rfloor + 1/2), & 0 \leq x \leq 2^7 q \\ b(2^7 - 1/2), & x > 2^7 q \\ -f(-x), & x < 0 \end{cases}$$

where $q = 0.0465$, and b is chosen so that $f(1) = 1$. Thus, uniform N-bit quantization, $2 \leq N \leq 8$, can be of the form

$$f(x) = \begin{cases} b(\lfloor x/Lq \rfloor + 1/2), & 0 \leq x \leq 2^{N-1} Lq \\ b(2^{N-1} - 1/2), & x > 2^{N-1} Lq \\ -f(-x), & x < 0 \end{cases}$$

for $1 \leq L \leq 2^{8-N}$. The parameter L can be chosen to trade large dynamic range (large L) for fine quantization (small L), and tells how many 8-bit quantized levels are to be combined into an N-bit quantized level. In practice, a quantizer is easier to implement if L is a power of 2.

The DSN's Viterbi decoders use 3-bit quantization with $L = 2^3$. The results described below were therefore obtained for 3-bit quantization with $L = 2^3$, and for 4-bit quantization with $L = 2^3$ and $L = 2^2$. Because it proved superior, $L = 2^3$ was used for 4-bit quantization. An example of $f(x)$ for 3-bit quantization is given in Fig. 4.

The effects of quantization at the inputs of the combiner are shown in Fig. 5 in terms of probability of bit error at the output of the Viterbi decoder versus E_b/N_{01} of signal s_1, assuming that $E_b/N_{01} = E_b/N_{02} = (1/2)(E_b/N_0)$, and that the branch metrics are computed according to Fig. 2(c).

The baseline curve is for no MCD input quantization and no combiner quantization. The remaining three curves show the performance for 3-bit, 4-bit quantization, and no quantization in the combiner, when the MCD uses a 3-bit input. All results are based on simulation and are accurate to ±3% with a 95% confidence interval. Figure 6 shows the 4-bit combiner performance when $E_b/N_{02} = (1/2)(E_b/N_{01})$.

These results led us to conclude that a combiner using 4-bit quantized recorded inputs may be the most reasonable compromise of storage and performance loss.

References

1. Divsalar, D., "Symbol Stream Combining Versus Baseband Combining for Telemetry Arraying," *TDA Progress Report 42-74*, Jet Propulsion Laboratory, Pasadena, Calif., pp. 13–28, Aug. 1983.

2. Hurd, W. J., Pollara, F., Russell, M. D., Siev, B., and Winter, P. U., "Intercontinental Antenna Arraying by Symbol Stream Combining at Giacobini-Zinner Encounter," *TDA Progress Report*, this issue.

3. Vo, Q. D., "Signal-to-Noise Ratio Combiner Weights Estimation for Symbol Stream Combining," *TDA Progress Report 42-76*, Jet Propulsion Laboratory, Pasadena, Calif., pp. 86–98, Feb. 1984.

4. Heller, J. A., and Jacobs, I. M., "Viterbi Decoding for Satellite and Space Communications," *IEEE Trans. Comm.*, Vol. COM-19, pp. 835–848, Oct. 1971.

5. Clark, G. C., and Cain, J. B., *Error Correction Coding for Digital Communications*, Plenum Press, 1981.

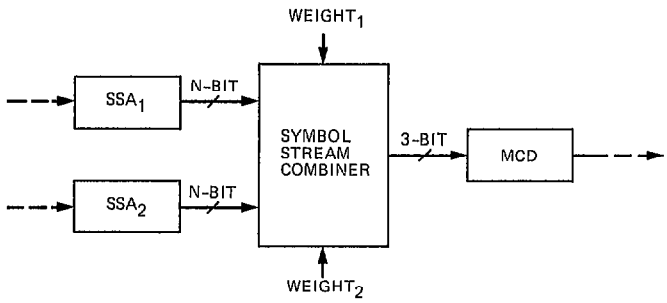

Fig. 1. Symbol stream combiner model

	0	1	2	3	4	5	6	7
0	0	0	1	1	2	3	4	
1	0	1	1	2	2	3	4	4
2	1	1	2	2	3	4	4	5
3	1	2	2	3	4	4	5	5
4	2	2	3	4	4	5	5	6
5	2	3	4	4	5	5	6	6
6	3	4	4	5	5	6	6	7
7	4	4	5	5	6	6	7	7

(a)

	0	1	2	3	4	5	6	7
0	0	1	1	2	2	3	3	4
1	1	1	2	2	3	3	4	4
2	1	2	2	3	3	4	4	5
3	2	2	3	3	4	4	5	5
4	2	3	3	4	4	5	5	6
5	3	3	4	4	5	5	6	6
6	3	4	4	5	5	6	6	7
7	4	4	5	5	6	6	7	7

(b)

	0	1	2	3	4	5	6	7
0	0	0	0	0	1	2	3	4
1	0	0	0	0	1	2	3	4
2	0	0	0	0	1	2	3	4
3	0	0	0	0	1	2	3	4
4	1	1	1	1	2	3	4	5
5	2	2	2	2	3	4	5	6
6	3	3	3	3	4	5	6	7
7	4	4	4	4	5	6	7	8

(c)

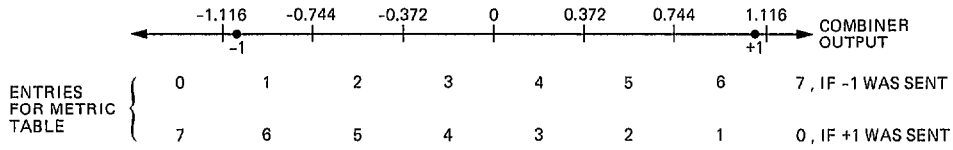

Fig. 2. Branch metric tables

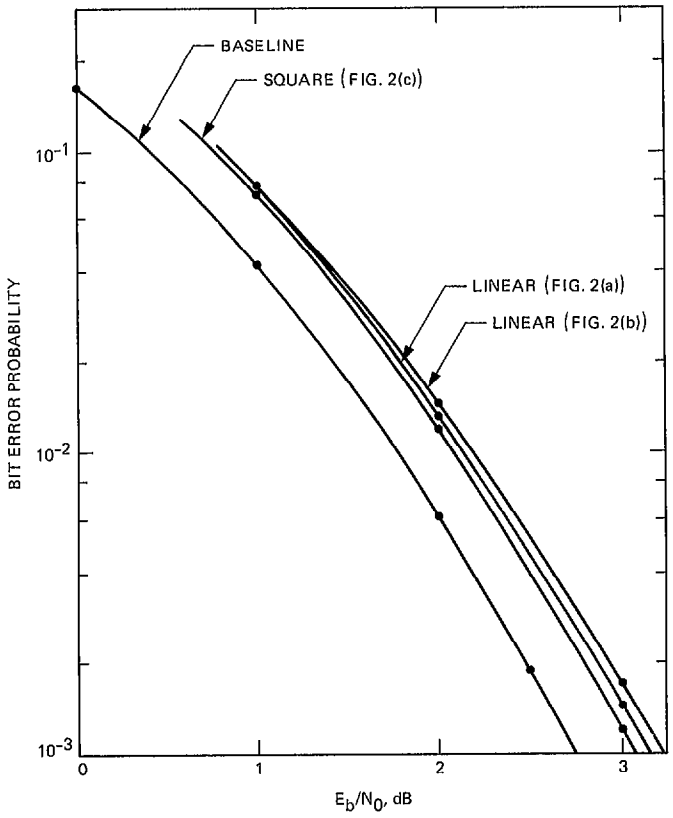

Fig. 3. Branch metric quantization effects

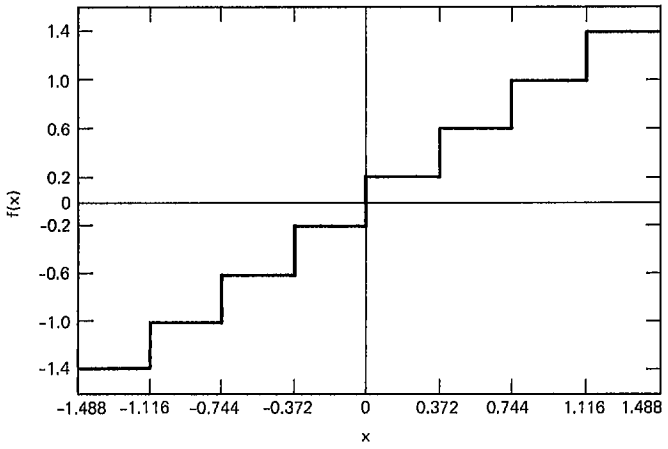

Fig. 4. Function f(x) for 3-bit quantization

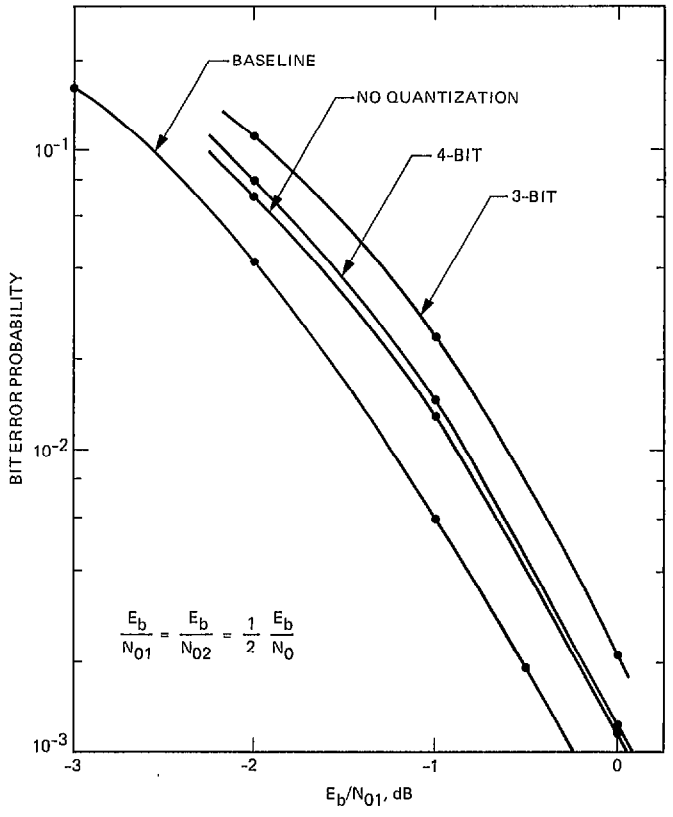

Fig. 5. Combiner quantization effect

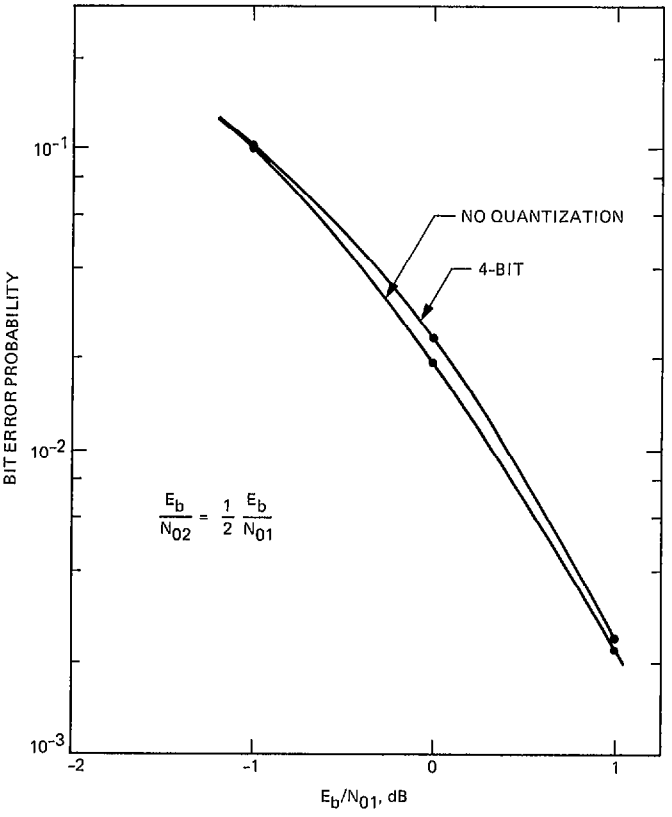

Fig. 6. Combiner with unequal strength inputs