1989002831
618418
9p.

S16-61
168227

N89-10202

Node Synchronization of Viterbi Decoders Using State Metrics

U. Cheng
Communications Systems Research Section

This article investigates the concept of node synchronization using state metrics. In this study, the branch metrics are integrated over a fixed time interval and the results are compared to the detection threshold. If the threshold is exceeded, the out-of-sync hypothesis is accepted; otherwise, the in-sync hypothesis is accepted. It is shown that the detection threshold can be chosen independent of any particular convolutional code with fixed code rate and constraint length if the code has reasonably good bit error rate performance.

Three node synchronization schemes are compared in this article: (1) a scheme using the syndrome (Scheme 1); (2) a scheme using the frame-sync patterns (Scheme 2); and (3) a scheme using the state metrics (Scheme 3). At very low SNRs, Scheme 2 can be faster than Scheme 1. For Voyager's rate 1/2 and constraint length 7 convolutional code, this happens for SNRs of less than 0.75 dB. This result is obtained by assuming that the coded frame-sync pattern has good aperiodic autocorrelation properties. For a fixed false alarm probability, the sequential detection scheme based on the syndrome is faster than Scheme 3 with fixed integration time. A sequential detection technique is needed to improve the speed of Scheme 3.

I. Introduction

The NASA standard concatenated coding system uses a (7,1/2) convolutional code as its inner code and an 8-bit (255, 223) Reed–Solomon (RS) code as its outer code. This system achieves a bit error rate (BER) of 10^{-6} at a bit signal-to-noise ratio (SNR) of 2.53 dB. However, it is possible to improve this performance by 2 dB using the newly discovered (15,1/6) convolutional code concatenated with a 10-bit (1023,959) RS code [1], [2]. Toward the goal of demonstrating this performance improvement, a convolutional decoder is currently under development for codes with constraint lengths up to 15. Good node synchronization schemes are essential to the achievement of the aforementioned performance. Three node synchronization schemes are considered in this article. Emphasis is placed on the scheme using state metrics because this scheme can be integrated in the Viterbi algorithm. The (7,1/2) convolutional code is chosen as an example to illustrate the theory. The same principle is applicable to the convolutional codes of constraint length 15.

Many node synchronization schemes have been investigated [3], [4]. In [3], the syndrome, which is computed from the hard-quantized received sequence, is used to detect the node synchronization. In this case, the syndrome $\{x_k\}$ is designed so that the distributions of sequences of $\{0,1\}$ in $\{x_k\}$ are different for the in-sync and out-of-sync hypotheses. Thus, by observing the distribution of $\{0,1\}$ in $\{x_k\}$, it is possible to determine which hypothesis is true. A sequential detection procedure based on this concept was described in [3]. This scheme is referred to as Scheme 1 in the subsequent discussion.

Another way to perform node synchronization is to use frame synchronization patterns. In this case, a fixed sync pattern is periodically inserted in the data stream. If the length of the sync pattern is longer than the memory length of the convolutional code, a fixed symbol pattern caused by the sync pattern periodically appears in the coded symbol stream. Therefore, a straightforward correlation process can be used to detect this pattern, and node synchronization is accomplished. The aperiodic autocorrelation property of the coded sync pattern is essential to node synchronization performance in this method. This scheme is referred to as Scheme 2 in the subsequent discussion.

The third method involves the use of state metrics in the Viterbi decoder to perform node synchronization. This is based on the observation that the growth rates of the state metrics are different under the in-sync and out-of-sync hypotheses. Many node synchronization techniques can be derived based on this observation. A sequential detection scheme was presented in [4]. In this article, a different approach is addressed, wherein the integration time is kept fixed and the result at the end of each integration period is compared to a preselected threshold. If the threshold is exceeded, the out-of-sync hypothesis is accepted (here we assume that the Viterbi decoder searches for the path with the smallest metric; see Section II for details); otherwise, the in-sync hypothesis is accepted. This scheme is referred to as Scheme 3 in the subsequent discussion. For this simple scheme, the "false alarm" and "missing" probabilities can be determined.

It will also be shown that the state metric growth rate under the out-of-sync hypothesis is independent of the choice of convolutional code with fixed rate and constraint length. This implies not only that the detection threshold can be the same for all convolutional codes with the same rate and constraint length, but that the resulting false alarm probability is also independent of the code. It turns out that the convolutional code having the best bit error rate (BER) performance may also give the best node synchronization performance. Suppose the detection threshold is determined for a given convolutional code. If another convolutional code with better BER performance is used in the future, the same detection threshold can still be used, and the missing probability is also reduced. It is also worth mentioning that the required integration time for a given false alarm probability is closely related to the BER of the convolutional code.

The concept of node synchronization using state metrics is explained in Section II. A performance analysis is given in Section III. Implementation considerations regarding Scheme 3 and a comparison of the three schemes are given in Section IV.

II. Node Synchronization Using State Metrics

Without loss of generality, we assume that the branch metrics are the square of the Euclidean distance between the received code words and the branch code words. Thus, the task of the Viterbi decoder is to search for the path with the smallest metric. For the in-sync hypothesis there is always a correct path, while for the out-of-sync hypothesis there is no correct path. If the decoder follows the correct path, the state metrics grow only because of channel noise. If the decoder follows the incorrect path, the state metrics grow at a rate determined by channel noise as well as by the mismatch between the received code word and the local reference branch code word. Thus, the state metric growth rates under the out-of-sync hypothesis are greater than those under the in-sync hypothesis. As the decoder event error probability decreases, the difference between the in-sync and out-of-sync state metric growth rates increases. Therefore, the performance of node synchronization is closely related to the BER performance of the convolutional decoder.

Note that the convolutional code improves BER performance over the uncoded system if the bit SNR is above a certain threshold, denoted by SNR_r, where the convolutional coded system has the same performance as the uncoded system. Simulation results indicate that if $SNR > SNR_r + 0.5$ dB, the node synchronization can be performed rapidly. If $SNR_r - 0.5 \text{ dB} < SNR < SNR_r + 0.5$ dB, a long integration time may be needed. A straightforward way to use the state metrics for node synchronization is as follows: A fixed integration time is selected. The state metric growth at the end of each integration period is compared to the preselected detection threshold. If the threshold is exceeded, the out-of-sync hypothesis is accepted; otherwise, the in-sync hypothesis is accepted. The integration time and the detection threshold are selected to yield satisfactory false alarm and missing probabilities.

III. Performance Analysis

Let $D_{i\tau}$ (in-sync) and $D_{i\tau}$ (out-of-sync) denote the state metric growth of state i in τ branches under the in-sync and out-of-sync hypotheses, respectively. The false alarm and missing probabilities based on the observations of $D_{i\tau}$ are given by

$$P_{FA,i}(\tau) = \text{prob}\{D_{i\tau}\text{ (out-of-sync)} < \gamma\} \tag{1}$$

and

$$P_{M,i}(\tau) = \text{prob}\{D_{i\tau}\text{(in-sync)} > \gamma\} \tag{2}$$

where γ is the detection threshold. At this moment, we assume that the state metric accumulators have infinite dynamic range. If the accumulators have a finite dynamic range, the actual implementation of the node synchronization scheme may depend on the renormalization procedure. This is explored in Section IV. The determination of the appropriate τ and γ can be made by simulation; however, it may be too time-consuming to be practical. Another approach is to generate the first $2N-1$ moments of $D_{i\tau}$ by simulation. The false alarm and missing probabilities can then be upper bounded by the technique referred to as the moment method, described in [5] and [6].

A good choice of the detection threshold γ should balance the false alarm probability and the missing probability. A reasonable choice can be

$$\gamma = \frac{\sigma_1 E\{D_{i\tau}\text{ (out-of-sync)}\} + \sigma_2 E\{D_{i\tau}\text{ (in-sync)}\}}{\sigma_1 + \sigma_2} \tag{3}$$

where

$$\sigma_1 = [\text{var}\{D_{i\tau}\text{ (in-sync)}\}]^{1/2} \tag{4}$$

and

$$\sigma_2 = [\text{var}\{D_{i\tau}\text{ (out-of-sync)}\}]^{1/2} \tag{5}$$

The quantization scheme used in this simulation study is as follows: Let x be the demodulator output for each received coded symbol: namely, $x = s + n$ if "0" is sent and $x = -s + n$ if "1" is sent, where n is a normal random variable with zero mean and unit variance and

$$s = \left(\frac{2Sr}{N_0}\right)^{1/2} \tag{6}$$

In the above equation, S is the signal power, N_0 is the one-sided noise power spectrum density, and r is the code rate. The metric for each received coded symbol is given by

$$d = \begin{cases} (x-s)^2 & \text{if the local reference coded symbol is "0"} \\ (x+s)^2 & \text{if the local reference coded symbol is "1"} \end{cases} \tag{7}$$

If the desired dynamic range is α^2 and the number of quantization levels is Q, the quantized metric is given by

$$y = \begin{cases} \left[\dfrac{Qd}{\alpha^2} + 0.5\right] & \text{if } d < \alpha^2\ [1-(1.5/Q)] \\ Q-1 & \text{if } d > \alpha^2\ [1-(1.5/Q)] \end{cases} \tag{8}$$

where $[\cdot]$ represents the integer part. For the out-of-sync hypothesis, if the received coded symbols and the local reference coded symbols have equal probability of matching or mismatching, all branch metrics are identically distributed random variables. This observation indicates that $D_{i\tau}$ (out-of-sync) are nearly identically distributed random variables for all convolutional codes with the same rate and constraint length. The observation is important because it implies that the false alarm probability performance is nearly independent of the selection of the convolutional code. We can see this by simulation. In Table 1, we compare $E\{D_{i\tau}\text{ (out-of-sync)}\}$ for two convolutional codes. The first convolutional code is Voyager's code with free distance 10. The second code was selected arbitrarily. It has free distance 7, and thus the BER performance may be much worse than that of the first code. We find that the first few moments of $D_{i\tau}$ (out-of-sync) are roughly the same. Suppose that the detection threshold is determined for a given convolutional code (code A) and another convolutional code with the better BER performance (code B) is used in the future. Since $E\{D_{i\tau}\text{ (out-of-sync)}\}$ for the two codes are roughly the same and code B has a larger metric growth rate difference under two hypotheses (because of the better BER performance), the same detection threshold can still be used for code B, and the missing probability is also reduced.

The state metric growth rate under the out-of-sync hypothesis depends on three factors: SNR, the integration time, and the quantization scheme. For the in-sync hypothesis, in addition to the three factors just mentioned, the state metric growth rate also depends on the convolutional code. The integration time and the quantization scheme are usually determined in advance. Therefore, they do not create a problem in the selection of γ. The SNR dependence of γ must be handled with care since SNR is usually unknown in advance and can be estimated only during operation. We know that the operation of the AGC/quantizer also needs a good SNR estimate. Therefore, a good estimate of SNR is essential to the operation of the Viterbi decoder.

Figures 1 and 2 show the state metric growth versus the integration time τ. It can be seen that the difference between $D_{i\tau}$ (out-of-sync) and $D_{i\tau}$ (in-sync) increases linearly with τ. In Fig. 2, $D_{i\tau}$ (in-sync) at an SNR of 2 dB is used as the reference in order to present all curves clearly. We see that although both $D_{i\tau}$ (in-sync) and $D_{i\tau}$ (out-of-sync) grow faster at the lower SNR, the separation between them grows more slowly. Figures 3 and 4 show the normalized metric growth difference versus the integration time. Note that the normalized metric growth difference determines the node synchronization performance. Figure 4 shows the need for long integration times at an SNR of 0.5 dB. In Figs. 5 and 6, the missing probability is shown. Figures 7 and 8 show the false alarm probability. All curves shown in Figs. 5 through 8 were computed using the upper bound based on the moment method, and in all cases the convolutional code generator is (171,133). The smoothness and accuracy of these curves depend on the amount of simulation data.

IV. Implementation and Comparison

The implementation of a node synchronization scheme based on state metrics depends to a large extent on the renormalization procedure. Suppose that the state metric accumulators have dynamic range 2^n, and 2^{n-1} is subtracted from all state metrics at each time of renormalization. If SNR = 2 dB, τ = 1600, and n = 18, the expected numbers of times of renormalization are $E\{D_{i\tau}$ (in-sync)$\}/2^{17}$ = 0.8468 and $E\{D_{i\tau}$ (out-of-sync)$\}/2^{17}$ = 0.9532. Clearly, just looking at the number of times of renormalization is not enough to determine which hypothesis is true; the actual contents of the accumulators must be examined. On the other hand, if n = 10, the expected numbers of times of renormalization are $E\{D_{i\tau}$ (in-sync)$\}/2^9$ = 216.78 and $E\{D_{i\tau}$ (out-of-sync)$\}/2^9$ = 244.02. In this case, the number of times of renormalization can provide useful information about which hypothesis is true.

Since the statistics of the state metrics are roughly the same for all states, the implementation can be based on the growth of the minimum state metric. The node synchronization scheme by state metrics can be implemented outside the Viterbi decoder if the minimum state metric is made available. The advantage of this external node synchronization installation is that many schemes can be implemented; the best scheme can be selected during the decoder operation.

For Scheme 3 (node synchronization using state metrics), if $P_{FA} = 10^{-6}$, the integration time is about 1500 branches at an SNR of 2 dB, 2800 branches at an SNR of 1 dB, and 12,000 branches at an SNR of 0.5 dB. For Scheme 2 (node synchronization using the frame sync patterns), the worst case detection delay is the number of bits between two consecutive sync patterns if one sync pattern is sufficient for detection; it is between 3000 and 8000 bits in our case. Thus, roughly speaking, Scheme 2 is faster when the SNR is less than 0.75 dB for Voyager's rate 1/2 and constraint length 7 convolutional code. Note that more than one sync pattern may be needed to get the satisfactory performance for the convolutional codes of constraint length 15.

The comparison between Scheme 1 (node synchronization using the syndrome) and Scheme 3 is based on two parameters: the short ARL and the long ARL, defined as follows:

Short ARL = average number of branches needed to accept the out-of-sync hypothesis if the out-of-sync hypothesis is true from the beginning.

Long ARL = average number of branches needed to accept the out-of-sync hypothesis if the in-sync hypothesis is true from the beginning.

For Scheme 3, the short ARL is determined by the integration time, and the long ARL is determined by the missing probability. Explicitly, the long ARL is equal to the integration time divided by P_M. Thus, the long ARL = 1.3×10^9 branches at SNR = 2 dB, and the required integration time is about 1300 branches. Compared to the results in [3], at an SNR of 2.3 dB, if the long ARL is 7.9×10^9, the corresponding short ARL is 171.4 branches. Note that Figs. 5 and 6 are computed using the upper bound based on the moment method. This bound is usually not tight, although it does provide good insight into the performance. The actual integration time for Scheme 3 could be shorter than 1300 branches, and it may be determined by simulation. Based on the above results, we conclude that Scheme 1 could be faster than Scheme 3. To reduce the integration time of Scheme 3 further, a sequential detection procedure is needed [4].

Acknowledgment

The author would like to thank Dr. F. Pollara for his valuable comments during this study.

References

[1] J. H. Yuen and Q. D. Vo, "In Search of a 2 dB Coding Gain," *TDA Progress Report 42-83*, vol. July–September 1985, Jet Propulsion Laboratory, Pasadena, California, pp. 26–33, November 15, 1985.

[2] S. Dolinar, "A New Code for Galileo," *TDA Progress Report 42-93*, vol. January–March 1988, Jet Propulsion Laboratory, Pasadena, California, pp. 83–96, May 15, 1988.

[3] G. Lorden, R. J. McEliece, and L. Swanson, "Node Synchronization for the Viterbi Decoder," *IEEE Transactions on Communications*, vol. COM-32, no. 5, pp. 524–531, May 1984.

[4] G. K. Huth, *A Parametric Study of the Complexity of Sequential Decoders*, Axiomatrix Report No. R7201-1, Axiomatix Corp., Los Angeles, California, January 1972.

[5] J. K. Omura and M. K. Simon, *Satellite Communication Performance Evaluation: Computational Techniques Based on Moments*, JPL Publication 80-71, Jet Propulsion Laboratory, Pasadena, California, September 1980.

[6] R. A. Scholtz, *The Moment Problem*, Department of Electrical Engineering, University of Southern California, 1981.

Table 1. Typical $E\,[D_{i_\tau}$ (out-of-sync)] for two convolutional codes

τ	Code #1 generator (171,133)	Code #2 generator (101,133)	Difference/σ_2 (Code #1)
8000	695,476	697,437	0.76
16,000	1,391,302	1,394,790	1.02
24,000	2,086,506	2,090,037	0.83
32,000	2,781,027	2,787,637	1.15
40,000	3,477,955	3,482,137	0.82

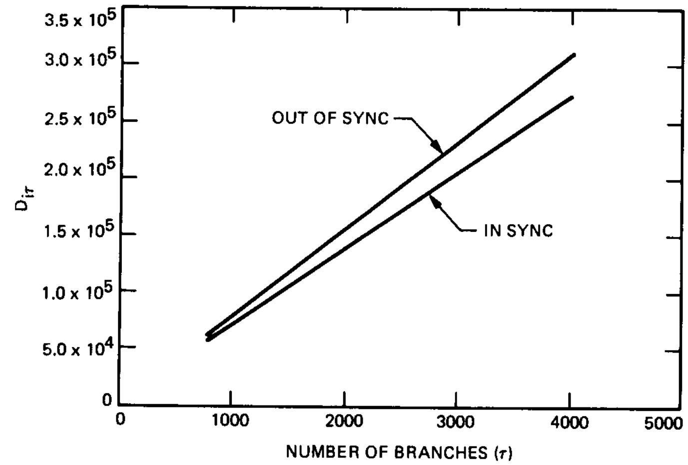

Fig. 1. Metric growth versus integration time; SNR = 2 dB, Q = 1024, and α = 5.52

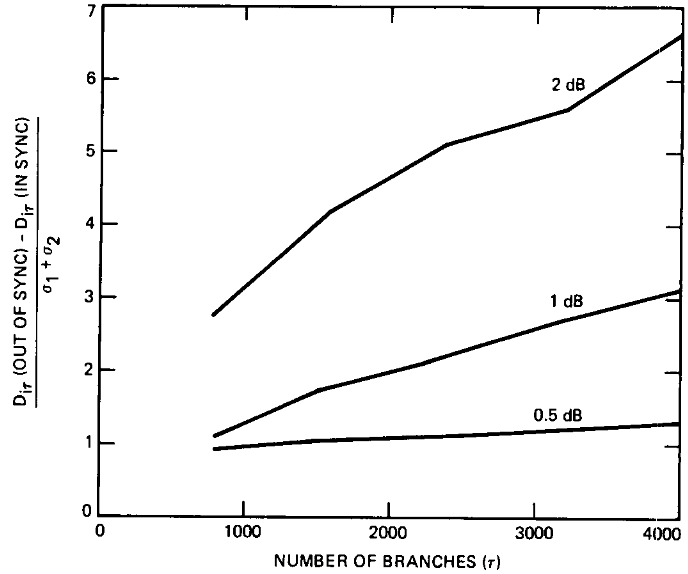

Fig. 3. Normalized metric growth difference versus integration time with SNR as the parameter. Q = 1024, α = 5.52 if SNR = 2 dB; α = 4.84 if SNR = 1 dB; α = 4.62 if SNR = 0.5 dB.

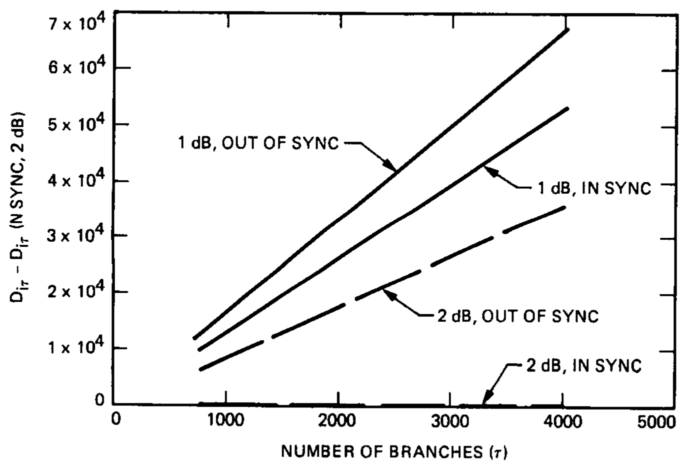

Fig. 2. Metric growth versus integration time with SNR as the parameter. Q = 1024, α = 5.52 if SNR = 2 dB; α = 4.84 if SNR = 1 dB.

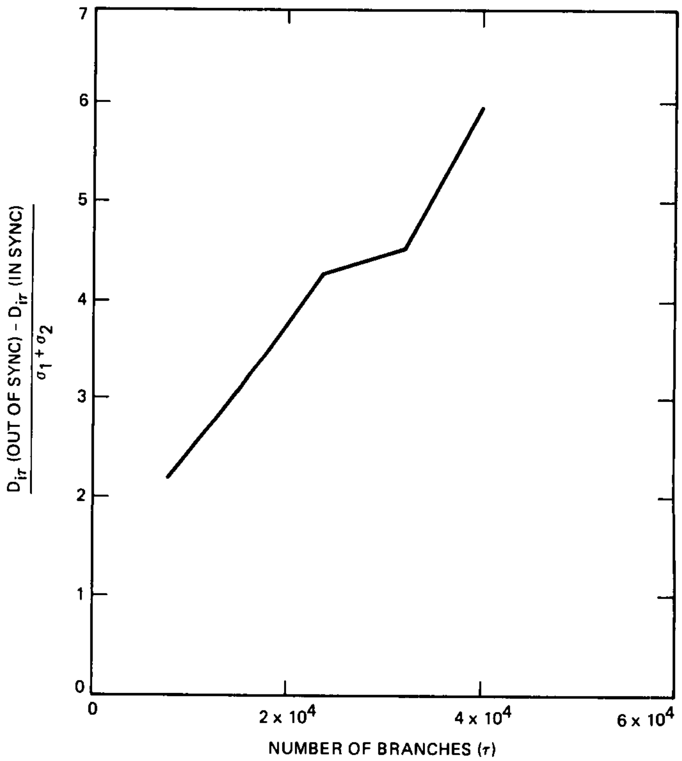

Fig. 4. Normalized metric growth difference versus integration time with SNR = 0.5 dB, Q = 1024, and α = 4.62

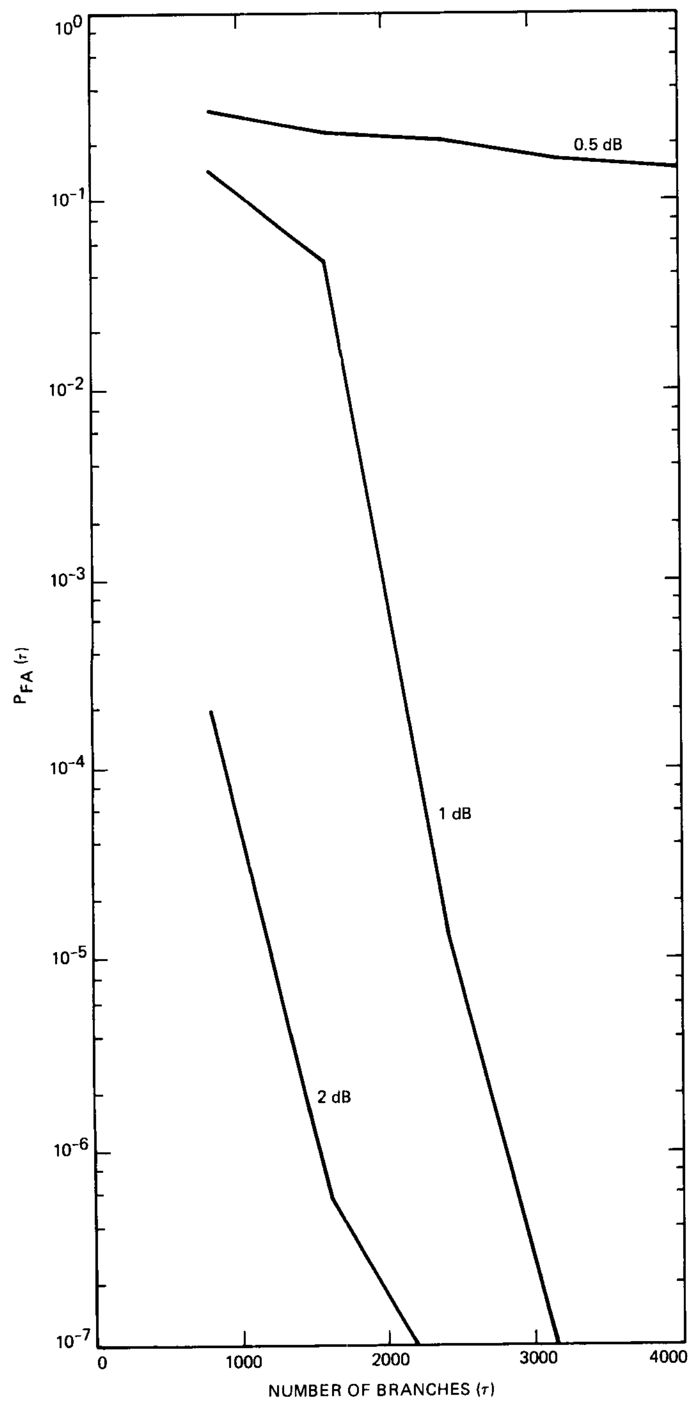

Fig. 5. Missing probability versus integration time with SNR as the parameter. $Q = 1024$; $\alpha = 5.52$ if SNR = 2 dB; $\alpha = 4.84$ if SNR = 1 dB; $\alpha = 4.62$ if SNR = 0.5 dB.

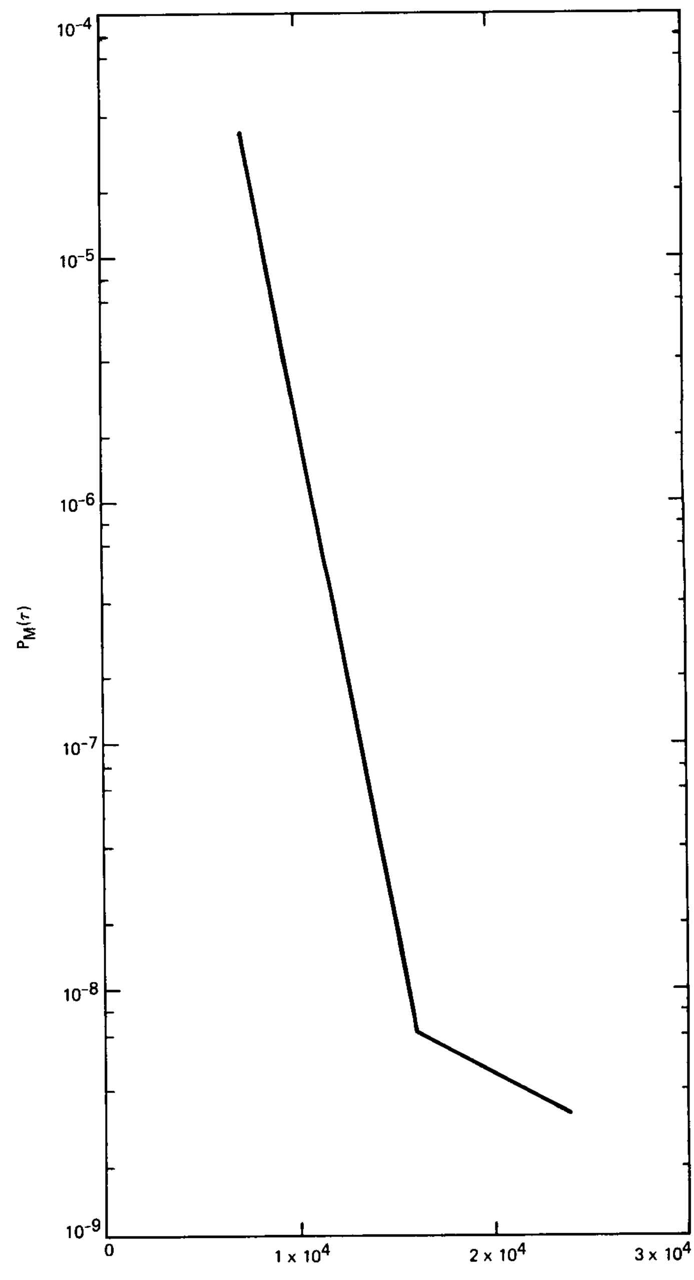

Fig. 6. Missing probability versus integration time with SNR = 0.5 dB. $Q = 1024$ and $\alpha = 4.62$.

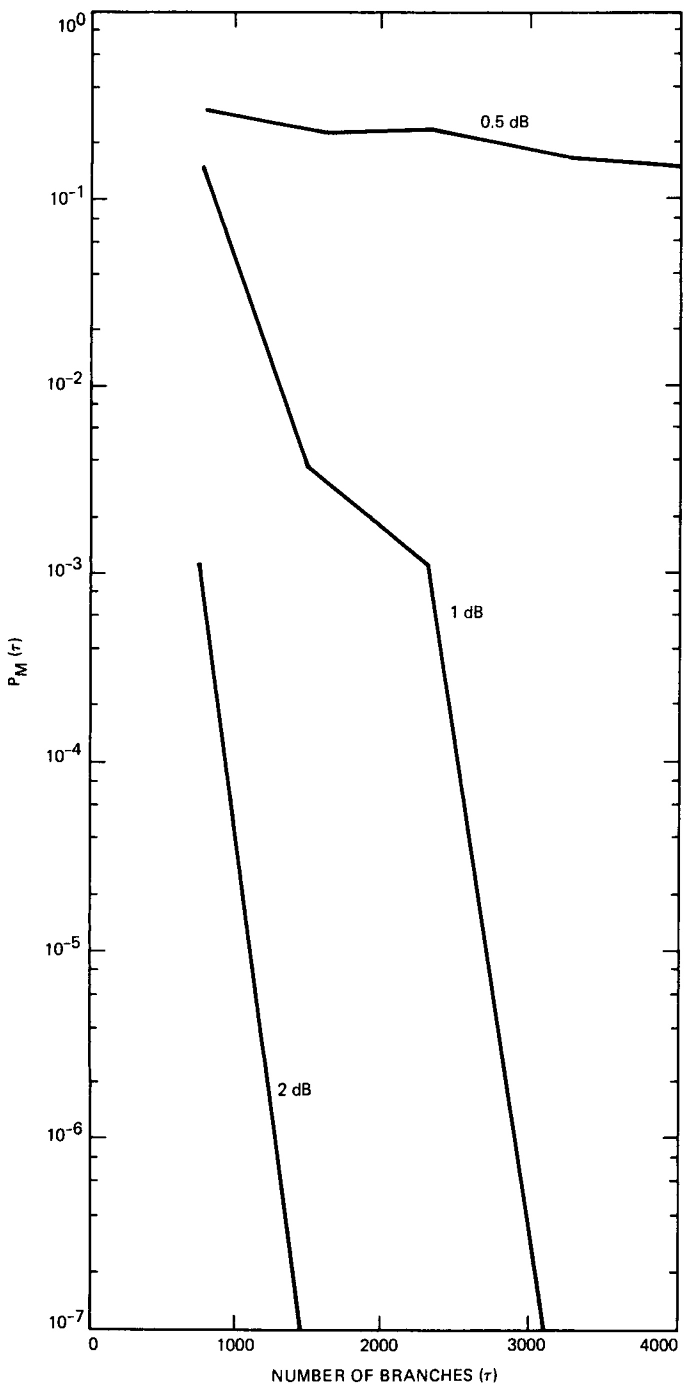

Fig. 7. False alarm probability versus integration time with SNR as the parameter. Q = 1024. α = 5.52 if SNR = 2 dB; α = 4.84 if SNR = 1 dB; α = 4.62 if SNR = 0.5 dB.

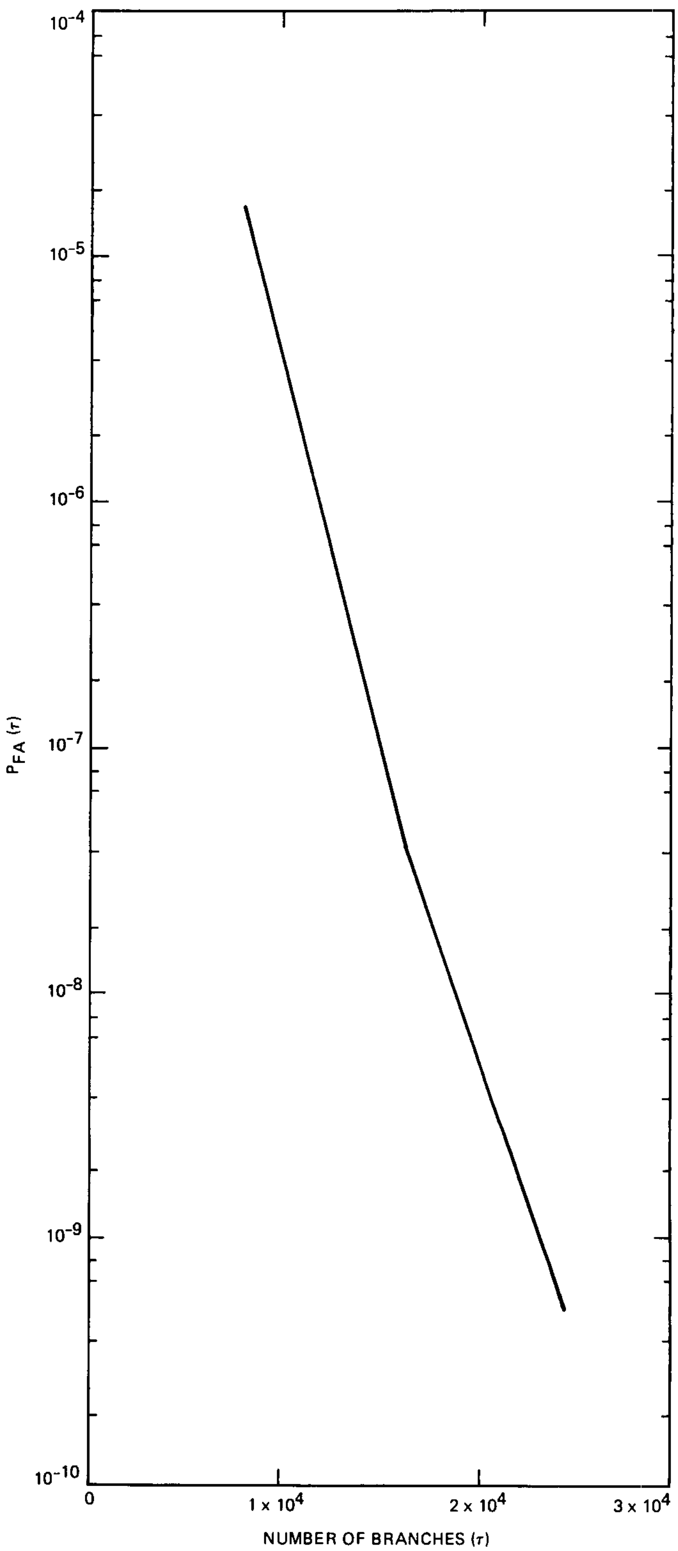

Fig. 8. False alarm probability versus integration time with SNR = 0.5 dB. Q = 1024 and α = 4.62.