A unified table-based Viterbi subset decoder for high-speed voice-band modems

by R. A. Nobakht

A unified table-based subset decoder for the Viterbi algorithm has been designed which is capable of decoding any constellation size for any data rate scheme of the CCITT (Comité Consultatif International Télégraphique et Téléphonique) V.32, V.32bis, and IBM V.32ter high-speed, full-duplex voice-band modem implementations with a constant number of computations. In addition, no error is introduced as the result of this subset decoding. The data rates include trellis coded 7200-, 9600-, 12000-, 14400-, 16800-, and 19200-bps rates. The number of computations necessary to perform the decoding is the same for any given data rate, resulting in reductions in computational complexity up to a factor of 32 in comparison to existing direct methods. The memory consumption of the decoder is relatively small and increases proportionally with data rate.

Introduction

The Viterbi algorithm (VA) can be used to obtain a recursive optimal solution to the problem of estimating the state sequence of a discrete-time finite-state Markov process observed in memoryless noise. Many problems in areas such as digital communications can be cast as such a problem. The algorithm was proposed in 1967 [1] as a method of decoding convolutional codes. Since that time, it has been recognized as an attractive solution to a variety of digital estimation problems, in a manner somewhat analogous to the use of the Kalman filter to address a variety of analog estimation problems. Like the Kalman filter, the VA tracks the state of a stochastic process with a recursive method and lends itself readily to implementation and analysis. However, the underlying process is assumed to be finite-state Markov rather than Gaussian, which leads to marked differences in structure.

The algorithm can use the structure of the trellis diagram and the input data to determine the most likely path through the trellis diagram [2]. The output for time (t_0)

0018-8646/95/$3.00 © 1995 IBM

reflects a decision made by the decoder on data received up to N time periods in the future. This means that the output for time (t_0) is necessarily delayed by N time periods, or that the latency of the decoder is N time periods. N is determined by the constraint length of the code and, for near optimum decoding, is four or five times the constraint length [3]. The most likely path through the trellis diagram (or, simply, trellis) is one that is a minimum-distance path for the input data or the path closest to the received data in Euclidean distance [4]. In other words, the Viterbi algorithm minimizes the distance

$$d(r, v) = \sum_{i=0}^{N-1} d(r_i, v_i), \tag{1}$$

where r_i and v_i are respectively the received and the decoded signal sequences. At each time period, every state in the trellis can have several paths (defined by each trellis subset) going into it, but only one will be the minimum distance for that state. Thus, the state with the smallest accumulated distance is the starting point, at that time period, to trace the minimum-distance subset path through the previous $N - 1$ time periods of the trellis. The minimum-distance subset paths to the next state are then calculated by determining the point on the constellation in each subset to which the input symbol is closest, determining the Euclidean distance to each of those points, and recording the appropriate bit sequence associated with the points. This is referred to as *subset decoding.*

Subset decoding

A subset decoder should analyze the input data point and determine the bit sequence and the Euclidean distance to the closest point in each of the n subsets. The Euclidean distance is defined to be

$$d = \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2}, \tag{2}$$

where x_c and y_c are the x and y coordinates of the point on the constellation, and x_i and y_i are the coordinates of the input symbol. This calculation can be done by computing the Euclidean distance to each point (parallel transition) in each subset, retaining only the smallest computed distance for that subset. Obviously, as the number of parallel transition points in the constellation increases for higher data rates of the same scheme, many more computations are required for subset decoding. This is a critical problem in the implementation of high-speed data modems.

Next, a powerful and efficient method of error-free subset decoding is described. This method consumes a constant number of computations for any constellation size of the same trellis structure of the CCITT V.32, V.32bis, and IBM V.32ter constellations. For such constellations, the boundaries that equally separate the locations of the sixteen points for a given subset can be outlined. This outline can point out the constellation point for the subset that is closest to the received input coordinate. Superimposing the boundaries of all eight subsets creates the boundaries for the entire constellation. There are 232 bounded areas on the constellation. For every bounded area, there is a unique set of eight points corresponding to the closest point for every subset, should the input fall in the bounded area [5].

Once the bounded area of the input is determined, the smallest Euclidean distance for every subset is just the Euclidean distance to that point for each subset determined by the boundary condition, considerably reducing the computation time. The bit sequences related to the closest points in all subsets for every bounded area are stored in a table for direct lookup. As mentioned previously, using the direct approach, the distance to all points in the constellation would be required for every input symbol. This translates to a great amount of computation, which grows exponentially with data rate. The proposed technique avoids this problem; i.e., it makes it possible to perform subset decoding for any constellation size with a constant number of computations.

Algorithmic methodology

Since the constellations for certain data rates can vary in phase by a $\pi/4$ rotation factor [6], one can obtain an appropriate constellation configuration by rotating a specific constellation by $\pi/4$ and normalizing the result by $\sqrt{2}$. The normalization is necessary in order to place the constellation points back to integer coordinates, since the grid size of the superimposed boundaries of all of the subsets is a 1×1 square. Therefore, for a given complex input symbol (x_i, y_i), where x_i and y_i are the in-phase and the quadrature components of the ith input symbol, and assuming an appropriate constellation having a $\pi/4$ phase rotation, it follows that

$$\begin{aligned}(x_i', y_i') &= (x_i, y_i)\frac{e^{j\pi/4}}{\sqrt{2}} \\ &= \frac{1}{2}(x_i + jy_i)(1 + j) \\ &= \frac{x_i - y_i}{2} + j\frac{x_i + y_i}{2}.\end{aligned} \tag{3}$$

Certain constellations, such as the CCITT V.32bis constellation for 12000-bit-per-second full-duplex modems, must be rotated by $\pi/4$ and normalized. It should be noted that not all constellations have to be rotated. Only specific constellations with a $\pi/4$ phase rotation need be considered.

The constellation bound (γ_x, γ_y) is enforced by requiring that the following boundary conditions be satisfied:

$$-\gamma_x < x_i' < \gamma_x,$$
$$-\gamma_y < y_i' < \gamma_y. \tag{4}$$

Next, we determine the axis quadrant index q_i of the input symbol:

$$q_i = \begin{cases} 0 & \text{if } x_i' \geq 0.0 \text{ and } y_i' \geq 0.0, \\ 1 & \text{if } x_i' < 0.0 \text{ and } y_i' > 0.0, \\ 2 & \text{if } x_i' > 0.0 \text{ and } y_i' < 0.0, \\ 3 & \text{if } x_i' \leq 0.0 \text{ and } y_i' \leq 0.0. \end{cases} \tag{5}$$

Since the constellations are symmetric about the axis [6], we map the symbol coordinates to the first quadrant $(q = 0)$:

$$x_i'' = |x_i'| \quad \text{and} \quad y_i'' = |y_i'|. \tag{6}$$

Then, we truncate the symbol coordinates to the nearest real and imaginary integer, namely

$$(\bar{x}_i, \bar{y}_i) = (\text{int})(x_i'', y_i'') \quad \text{s.t.} \quad (\bar{x}_i, \bar{y}_i) \in \{\Im\}, \tag{7}$$

where $\{\Im\}$ denotes the set of all positive integers. Referring back to the superimposed boundaries of all subsets, it is apparent that the overall grid structure is unique. This grid structure is composed of squares and triangles. This creates the common grid shape of a square with an upper and a lower triangle. Therefore, the location of the symbol coordinates relative to these triangles can be denoted by λ_i and determined as follows:

$$\lambda_i = \begin{cases} 0 & \text{if } y_i'' > x_i'' - \bar{x}_i + \bar{y}_i \quad \text{(upper triangle)}, \\ 1 & \text{if } y_i'' \leq x_i'' - \bar{x}_i + \bar{y}_i \quad \text{(lower triangle)}. \end{cases} \tag{8}$$

All of the necessary information is now available to determine the bounded area of the input in order to access two sets of tables, designated as Tables A and B. The former is composed of two region numbers (for the upper and the lower triangle regions) for all possible truncated input symbols. The indexes for this table are $\bar{x}_i + \gamma_x \bar{y}_i$ and λ_i for determining the appropriate region number. The maximum size of Table A can be mathematically represented in Equation (9). Since the number of triangles in the grid structure can vary from one constellation to the next, only an upper bound for the size of this table can be given:

$$A[\bar{x}_i + \gamma_x \bar{y}_i][\lambda_i] = 2(\bar{x}_i + \gamma_x \bar{y}_i) + \lambda_i. \tag{9}$$

Once a region number has been determined from Table A, it should be added to an offset, δ_i, in order to map it to its correct constellation coordinate. This offset is the product of the original quadrant index of the input symbol (q_i) and the value of the quarter of the total number of entries in Table B (the number of independent regions in the first quadrant). The upper bound of δ_i can be defined as

$$\delta_i = 2\gamma_x \gamma_y q_i. \tag{10}$$

After the index to Table B has been determined, it can be used to extract all n subset path values. The maximum size of Table B can now be defined as $B[\delta_i + 2(\bar{x}_i + \gamma_x \bar{y}_i) + \lambda_i][n]$. Table B is composed of the bit sequences related to the closest points in all subsets for every bounded area (subset path values). These bit sequences, which can be composed of only the Q bits, can be stored in the order of the state value of the corresponding subset. Thus, the most significant bits $(Y0_n, Y1_n, Y2_n, \cdots, Yj_n)$ can be eliminated and made apparent in the indexing of the table entries. The smallest Euclidean distance for every subset can then be determined from the subset path values obtained from Table B, completing the *subset decoding* portion of the Viterbi decoder.

Table 1 Fractional amount of computation and memory consumption for data rates used.

Data rate (bps)	*Fraction of computation*	*Memory consumption* (words)
19,200	1/32	6000
16,800	1/16	3000
14,400	1/8	1500
12,000	1/4	750
9600	1/2	375
7200	1	187

Implementation results

The algorithm was applied to subset decoding of trellis-encoded data during the Viterbi-decoding phase of V.32, V.32bis, and IBM V.32ter high-speed, full-duplex, voice-band modem implementations. The trellis encoder used for the above schemes was an eight-state, rate 2/3 encoder, and the implementation was carried out on an IBM MSP2780 digital signal processor. **Table 1** shows the reduced amount of computation required for all of the data rates of the above modems compared to that required using the direct computation method described in the section on subset decoding. As can be seen, reductions by as much as a factor of 32 were achieved. Since this is a table-based method, it involves the trading of computational resources for memory resources—preferable because of the relatively high cost of computational resources compared to that of memory resources. As can be seen from the table, the memory consumption of the algorithm is relatively small and increases with data rate.

Summary

A unified table-based subset decoder for the Viterbi algorithm has been designed which can be applied to trellis-coded high-speed full-duplex voice-band modems. The number of computations necessary for decoding remains constant for all data rates of the CCITT V.32, V.32bis, and IBM V.32ter high-speed modem implementations. In addition, no error is introduced as the result of the subset decoding. Computational reductions by as much as a factor of 32 have been observed in comparison to existing methods. The memory consumption of the algorithm used is small and increases proportionally with data rate. Because of the relatively high cost of computational resources compared to that of memory resources, its use should be advantageous for many applications.

Note

The work described in this paper was carried out while the author was a member of the Digital Communications Development Department at the IBM Microelectronics Division facility in Research Triangle Park, North Carolina.

References

1. A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. Info. Theor.* **IT-13,** 260–269 (1967).
2. G. Ungerboeck, "Trellis-Coded Modulation with Redundant Signal Sets. Part I: Introduction," *IEEE Commun. Magazine* **25,** 5–11 (1987).
3. J. A. Heller and I. M. Jacobs, "Viterbi Decoding for Satellite and Space Communication," *IEEE Trans. Commun. Technol.* **COM-19,** 835–848 (1971).
4. G. Ungerboeck, "Channel Coding with Multi-Level/Phase Signals," *IEEE Trans. Info. Theor.* **IT-28,** 55–67 (1982).
5. R. A. Nobakht, "An Efficient Trellis Subset Decoder Design for V.32 Class Modems," *Technical Report TR 29.1909*, IBM Microelectronics Division, Research Triangle Park, NC, July 1994.
6. *Data Communication over the Telephone Network*, International Telegraph and Telephone Consultative Committee (CCITT), International Telecommunications Union, Geneva, Switzerland, 1989.

Received September 7, 1994; accepted for publication June 14, 1995

Ramin A. Nobakht *Rockwell International Corporation, 4311 Jamboree Road, Newport Beach, California 92658 (ramin.nobakht@nb.rockwell.com).* Dr. Nobakht received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from North Carolina State University, Raleigh, in 1984, 1987, and 1991, respectively. From 1991 to 1994, he was a member of the Digital Communications Development Department at the IBM Microelectronics Division facility at Research Triangle Park, North Carolina. His work pertained to the development of cellular radio and high-speed voice-band modems, with emphasis on adaptive equalization, bandwidth-efficient modulation, echo cancellation, carrier and timing recovery, and digital signal processing. Since 1995 he has been a member of the Advanced Product Development Department at the Rockwell Digital Communications Division facility at Newport Beach, California. Dr. Nobakht's current interests include digital communications, adaptive digital signal processing, digital speech and image processing, optimization theory, neural networks, and computer-aided design and simulation of advanced digital communications systems.