

Traceback Operation in Viterbi Decoding for Rate-k/n Convolutional Codes

Field of the Invention

5 The present invention relates generally to improvements in traceback operation in the Viterbi decoder that is used in decoding rate-k/n convolutional codes or trellis codes. The present invention is applicable to the implementation of Viterbi decoder in decoding the Trellis-coded modulation (TCM). A typical application of this invention is decoding of the encoded TCM symbol data in the baseband signal of the PHY layer of a network.

10

Background

15 In most wireless communication systems, particularly those with significant intersymbol interference, convolutional encoding and Viterbi decoding are used to perform error detection and correction. The convolution encoder achieves error free transmission by adding enough redundancy to the source symbols. The choice of the convolutional code is application dependent and varies with the frequency characteristics of the transmission medium as well as the desired transmission rate. The Viterbi algorithm is a method commonly used for decoding bit streams encoded by convolution encoders. Figure 1 shows the block diagram of a Viterbi decoder. Viterbi decoding is mainly composed of an Add-Compare-Select (ACS) circuit (110) and a traceback circuit (130). The details of a particular decoder algorithm depend on the encoder.

25 Figure 2 shows an example of a convolutional code encoder having two time delay units (210 and 220). $x(n)$ is the input and $G_0(n)$ and $G_1(n)$ are the encoded outputs. The rate of the convolution encoder is defined as the number of input bits to output bits. This system has 1 input and 2 outputs thereby resulting in a coding rate of $1/2$. Its number of delay units determines the number of states of a convolution coder. For a

system with k delay units, there are 2^k states. There are 2 delay units in this system and therefore, there are $2^2 = 4$ states.

The system block diagram can be expressed with the following equations:

5 $G0(n) = x(n) + x(n-1) + x(n-2)$

$G1(n) = x(n) + x(n-2)$

If there are m inputs, then there will be 2^m state transitions. The total number of state transitions at a point in time is the product of the number of state transitions and the number of states, 2^{m+k} . Figure 3 shows a trellis diagram of the systems in Figure 2 which include time in state transitions. Each node in the trellis diagram (310) denotes a state at a point in time. The branches connecting the nodes denote the state transitions (320). The state transitions are independent of the system equations. This a consequence of including the input, $x(n)$, in the state definition. In certain systems, definition of the states may be more complicated and may require special considerations.

Viterbi decoding is composed of an Add-Compare-Select (ACS) operation and a traceback operation. In ACS, two things are done in every symbol interval (Refer to Figure 2, 1 symbol = 1 input bit = 2 encoded bits): the accumulated metric or path metric, which is the addition between the corresponding branch metric and the path metric of the originating state, is calculated for each state and the optimal incoming path or maximum likelihood path associated with each state is determined. The optimal incoming path, which is also called survivor path for each state will be stored in the storage unit. For the example from Figure 2 with 1 input, only 1 bit is needed to indicate the chosen path. A value of 0 indicates that the top incoming path of the given state is chosen as the best path whereas a value of 1 indicates the bottom path is chosen.

Traceback begins after completing the ACS operation of the last symbol in the frame, which is equal to the traceback length. The length of the traceback is typically chosen as 5 times of the constraint length. Traceback uses the survivor path information

from the storage unit to derive an optimal path through the trellis. The starting state for the traceback operation is either known or estimated to be correct. During traceback operation, the survivor path value is sent to the decoder output and the originating state of the survivor path is determined. The number of repetition for this process is equal to the traceback length. A more exact method is to wait until an entire frame of data is received before beginning traceback. In this case, tail bits or tail symbols are added to force the trellis to the zero state, providing a known point to begin traceback. The order with which the decoded output is generated is reversed as the path taken by traceback operation is from the end to the beginning of trellis, which is in the opposite direction to the path taken by ACS operation. In this example, coincidentally, the survivor path value can be directly sent to the decoder output without any other added operation. Lastly, the ordering of the decoded output need to be reversed to exactly reconstruct the sample input, $x(n)$.

As described above, the traceback operation for a rate $\frac{1}{2}$ convolution encoder only needs 1 bit to indicate the survivor path for an individual state and that particular bit is actually the desired output of the decoder. However, this method will not be sufficient if it applies to rate- k/n convolutional codes or trellis codes. Figure 4 shows an example of a rate- $2/3$ TCM encoder with 3 time delay units (410, 420 and 430) and Figure 5 shows the trellis diagram (510) of the system. In order to get the desired output of the decoder with the methods described in the prior art, it will need at least 2 bits to indicate the survivor path and some look up tables to determine the desired outputs from the survivor path information. This will increase the complexity of hardware implementation and time taken to process to get the desired output of the decoder.

The problem to be solved by this invention is to develop a new method for traceback operation having less hardware and time complexities than the methods described in the prior art to decode a rate- k/n convolutional codes or trellis codes.

Summary of the Invention

In accordance with a first aspect of the invention, there is provided a method for performing a traceback operation in Viterbi decoding for decoding a stream of code signals, said method comprising at least the step of sequentially performing traceback operations on a current frame starting from a state through which a path of
5 a previous frame passes.

The new method of performing traceback operation for rate-k/n convolutional codes or trellis codes, for example in Figure 5 (510) is described as follows. The order of the survivor path for each state in a symbol interval that is stored to the storage unit need
10 to be arranged in such a way that it will simplify the process of determining the originating state. All the states of the encoder will be equally divided into two conditions: even or odd. The condition is the key element in determining the originating state of the survivor path and obtaining the desired outputs of the decoder.

Two stage of operation is needed to obtain the desired final outputs. First stage is the error-free starting state calculation stage. Two storage elements are needed for storing the survivor path and its originating state. Then, the survivor path of the originating state will be used to determine the next symbol's originating state that will overwrite the storage element that stored the previous survivor path. Other means of
20 performing said first stage of operation are a counter means, a memory pointer means, a condition indicator means, a dependent constant means and a shift & mask means. Second stage is the result output calculation stage. Said second stage uses the same means that performs said first stage operation and other additional means. Said additional means are a third storage element and a dedicated logic means.

25

In accordance with a second aspect of the invention, there is provided a method for performing traceback operation in Viterbi decoding for decoding a stream of code symbols encoded using a high rate trellis code wherein the constraint length of an encoder is K and the traceback length is m, each of K
30 and m being a positive integer, said method comprising the steps of: reading the

survivor path information from a storage unit; determining a starting state which is a state location through which a most likely path passes of the previous frame; obtaining the result outputs by sequentially performing traceback operations on the current frame from said starting state determined by said step of determining a starting state; obtaining said result outputs which includes tail bits or tail symbols by sequentially performing traceback operations on said current frame from said starting state defined as the state zero; and reversing the order of said obtained result outputs to get the final desired outputs.

A description will now be made on the operation of the traceback operation invented for viterbi decoding. Said step of determining a starting state location begins with initializing said first and second storage element to zero, said counter value to at least five times of said constraint length but less than said traceback length and said storage unit pointer to a constant. Said condition indicator updates value to zero if the

value of said first storage element is even and one if the value of said first storage element is odd. Reading a word of survivor path information from said storage unit overwrites said first storage element. Said condition indicator and said second storage element determine said number of state location in said first storage element to be right shifted and masked. Said condition indicator updates value but this time is based on said second storage element. The operation repeats until said counter value decreases to zero. The role of said first storage element and said second storage element interchanges whenever the operation repeats. The final value of condition indicator and said storage element determines the starting state for said step of obtaining said m decoded outputs, where m is the traceback length.

The operations of obtaining said m decoded outputs are now explained. Said initializing step includes initialize said third storage element to zero, said counter value to said traceback length and said storage unit pointer to a constant. Said values of said first or second storage element and condition indicator of said previous frame determines said starting state of said traceback operation. Said third storage element updates its value in a manner that the least significant bit is equal to said bit location 1 of said first storage element from said previous frame and the most significant bit is equal to said condition indicator from said previous frame. Said condition indicator updates value to zero if the value of said first storage element is even and one if the value of said first storage element is odd. Reading a word of survivor path information from said storage unit overwrites said first storage element. Said condition indicator and said second storage element determine said number of state location in said first storage element to be right shifted and masked.

The operation continues with interchanging the role of said first storage element with said second storage element. Said third storage element updates its value in a manner that the least significant bit is equal to said bit location 1 of said second storage element from said previous frame and the most significant bit is equal to said condition indicator from said previous frame. Said condition indicator updates value to zero if the

value of said second storage element is even and one if the value of said second storage element is odd. Reading a word of survivor path information from said storage unit overwrites said second storage element. Said condition indicator and said first storage element determine said number of state location in said second storage element to be
5 right shifted and masked. Also, said value of condition indicator determines the type of exclusive-or operation to perform in order to get said result outputs. The operation repeats until said counter value decreases to zero. The role of said first storage element and said second storage element interchanges whenever the operation repeats. The final desired outputs that are equal to the source data could be obtained by reversing the order
10 of said calculated result.

Brief Description of the Drawings

- 15 Figure 1: Illustrates the block diagram of a Viterbi decoder.
- Figure 2: Illustrates an example of a convolutional code encoder having two time delay units.
- Figure 3: Illustrates the trellis diagram of the system shown in Figure 2.
- 20 Figure 4: Illustrates an example of a rate-2/3 TCM encoder with 3 time delay units.
- Figure 5: Illustrates the trellis diagram of the system shown in Figure 4.
- 25 Figure 6(a): Illustrates the modified trellis diagram after bit-reverse method and separates into odd and even condition.
- Figure 6(b): Illustrates the position of the state location, which stores the survivor path information.
- 30

Figure 7: Illustrates the operation flow of the error-free starting state calculation stage.

5 Figure 8: Illustrates the operation flow of the result output calculation stage.

Figure 9: Illustrates the operation flow of the result output calculation stage with consideration of tail bits or tail symbols.

10 Detailed Description

An embodiment of this invention is shown in Figure 7. Before the real traceback operation begins, the order of the survivor path for each state stored on the storage unit need to be arranged in such a way shown in Figure 6(a) and 6(b). Each state location
15 will have 2 bits to indicate the survivor path. The arrangement is made successful by bit-reverse every state of the trellis diagram in Figure 5 and categorize it to two groups, which are odd and even. The definition of odd or even in this context is based on the 2 most significant bits of every current state (610-680). The order to read the survivor path is from the tail to the tip of the arrow (610-680). According to Figure 6(a), the left side
20 of the diagram represents the even condition while the right side of the diagram represents the odd condition. The 2 bits will represent the survivor path and stored accordingly to the predefined state location (690).

The operation of the embodiments shown in Figure 7 is now explained. Said step
25 of determining a starting state location begins with initializing said first and second storage element to zero, said counter value to at least five times of said constraint length but less than said traceback length and said storage unit pointer to a constant (710). Said condition indicator updates value to zero if the value of said first storage element is even and one if the value of said first storage element is odd (720). Reading a word of
30 survivor path information from said storage unit overwrites said first storage element

(730). Said condition indicator and said second storage element determine said number of state location in said first storage element to be right shifted and masked (740 and 750). Said condition indicator updates value but this time is based on said second storage element. The operation repeats until said counter value decreases to zero (760). The role of said first storage element and said second storage element interchanges whenever the operation repeats (770). The final value of condition indicator and said storage element determines the starting state for said step of obtaining said m decoded outputs, where m is the traceback length.

Figure 8 shows the embodiment of the second stage of the traceback operation, which is obtaining said result outputs. Said initializing step includes initialize said third storage element to zero, said counter value to said traceback length and said storage unit pointer to a constant (810). Said values of said first or second storage element and condition indicator of said previous frame determines said starting state of said traceback operation (810). Said third storage element updates its value in a manner that the least significant bit is equal to said bit location 1 of said first storage element from said previous frame and the most significant bit is equal to said condition indicator from said previous frame (811). Said condition indicator updates value to zero if the value of said first storage element is even and one if the value of said first storage element is odd (812). Reading a word of survivor path information from said storage unit overwrites said first storage element (813). Said condition indicator and said second storage element determine said number of state location in said first storage element to be right shifted and masked (814 and 815).

The operation continues with interchanging the role of said first storage element with said second storage element. Said third storage element updates its value in a manner that the least significant bit is equal to said bit location 1 of said second storage element from said previous frame and the most significant bit is equal to said condition indicator from said previous frame (820). Said condition indicator updates value to zero if the value of said second storage element is even and one if the value of said second

storage element is odd (821). Reading a word of survivor path information from said storage unit overwrites said second storage element (822). Said condition indicator and said first storage element determine said number of state location in said second storage element to be right shifted and masked (823 and 824). Also, said value of condition
 5 indicator determines the type of exclusive-or operation to perform in order to get said result outputs (825). The operation repeats until said counter value decreases to zero (830). The role of said first storage element and said second storage element interchanges whenever the operation repeats (840). The final desired outputs that are equal to the source data could be obtained by reversing the order of said calculated
 10 result.

Figure 9 shows the embodiment of the second stage of the traceback operation with tail bits or tail symbols, which is obtaining said result outputs. Said initializing step includes initialize said first storage element, second storage element and third storage
 15 element to zero, said condition indicator to zero, said counter value to said traceback length and said storage unit pointer to a constant (910). Said third storage element updates its value in a manner that the least significant bit is equal to said bit location 1 of said first storage element from said previous frame and the most significant bit is equal to said condition indicator from said previous frame (911). Said condition indicator
 20 updates value to zero if the value of said first storage element is even and one if the value of said first storage element is odd (912). Reading a word of survivor path information from said storage unit overwrites said first storage element (913). Said condition indicator and said second storage element determine said number of state location in said first storage element to be right shifted and masked (914 and 915).

25

The operation continues with interchanging the role of said first storage element with said second storage element. Said third storage element updates its value in a manner that the least significant bit is equal to said bit location 1 of said second storage element from said previous frame and the most significant bit is equal to said condition
 30 indicator from said previous frame (920). Said condition indicator updates value to zero

if the value of said second storage element is even and one if the value of said second storage element is odd (921). Reading a word of survivor path information from said storage unit overwrites said second storage element (922). Said condition indicator and said first storage element determine said number of state location in said second storage
5 element to be right shifted and masked (923 and 924). Also, said value of condition indicator determines the type of exclusive-or operation to perform in order to get said result outputs (925). The operation repeats until said counter value decreases to zero (930). The role of said first storage element and said second storage element interchanges whenever the operation repeats (940). The final desired outputs that are
10 equal to the source data could be obtained by reversing the order of said calculated result.

This invention provides an efficient method to perform traceback operation in Viterbi decoding for rate-k/n convolutional codes. The two stages of traceback
15 operation further improve the accuracy of Viterbi decoding wherein the error-free starting state is determined before performing the traceback operation to obtain the desired outputs. This method reduces the hardware and time complexities to perform traceback operation by avoiding the usage of look up tables in order to determine the desired outputs from the survivor path information. The ability to separate the survivor
20 path information into two basic conditions with proper arrangement is the key element in this invention.

WHAT IS CLAIMED IS:

1. A method for performing traceback operation in Viterbi decoding for decoding a stream of code symbols encoded using a high rate trellis code wherein the constraint
5 length of an encoder is K and the traceback length is m , each of K and m being a positive integer, said method comprising the steps of:
 - reading the survivor path information from a storage unit;
 - determining a starting state which is a state location through which a most likely path passes of the previous frame;
 - 10 obtaining the result outputs by sequentially performing traceback operations on the current frame from said starting state determined by said step of determining a starting state;
 - obtaining said result outputs which includes tail bits or tail symbols by sequentially performing traceback operations on said current frame from said starting
15 state defined as the state zero; and
 - reversing the order of said obtained result outputs to get the final desired outputs.
2. The method of claim 1, wherein:
 - said high rate trellis code is a rate- k/n convolutional code wherein the number of
20 input of an encoder is k and the number of output of an encoder is n .
3. The method of either one of claim 1 or claim 2, wherein said step of reading the survivor path information further comprises the steps of:
 - determining the order of said survivor path for each state stored and the state
25 location defined on said storage unit;
 - assigning the $(n-1)$ most significant bits of every current state as said survivor path value; and
 - dividing the trellis diagram into two conditions: odd or even which is determined by said survivor path value and indicated by condition indicator.

4. The method of any one of the preceding claims, wherein said step of determining a starting state location further comprises the steps of:

initializing first and second storage element to zero, a counter value to at least five times of said constraint length but less than said traceback length and a storage unit
5 pointer to a constant;

determining the condition of said first storage element and update said condition indicator;

reading a word from said storage unit to said first storage element and increasing the storage unit pointer;

10 determining said number of state location in said first storage element to be right shifted;

shifting and masking said first storage element and decreasing said counter value by one; and

determining said counter value;

15 and wherein said step of determining the condition will be repeated by interchanging the role of said first storage element with said second storage element and vice versa if said counter value is not equal to a constant value otherwise the operation will end.

20 5. The method of any one of the preceding claims, wherein said step of obtaining said result outputs further comprises the steps of:

initializing third storage element to zero, said counter value to said traceback length and said storage unit pointer to a constant;

25 determining said starting state from said final value of said previous frame of said first or second storage element and condition indicator;

updating said third storage element wherein the least significant bit is equal to said bit location 1 of said first storage element and the most significant bit is equal to said condition indicator;

30 determining the condition of said first storage element and update said condition indicator;

reading a word from said storage unit to said first storage element and increasing the storage unit pointer;

determining said number of state location in said first storage element to be right shifted;

5 shifting and masking said first storage element;

updating said third storage element wherein the least significant bit is equal to said bit location 1 of said second storage element and the most significant bit is equal to said condition indicator;

10 determining the condition of said second storage element and update said condition indicator;

reading a word from said storage unit to said second storage element and increasing the storage unit pointer;

determining said number of state location in said second storage element to be right shifted;

15 shifting and masking said second storage element;

determining said value of condition indicator to perform exclusive-or operation to obtain said result outputs and decreasing said counter value by one; and

determining said counter value;

20 and wherein said step of updating said third storage element wherein the least significant bit is equal to said bit location 1 of said second storage element and the most significant bit is equal to said condition indicator will be repeated by interchanging the role of said first storage element with said second storage element and vice versa if said counter value is not equal to a constant value otherwise the operation will end.

25 6. The method of any one of the preceding claims, wherein said step of obtaining said result outputs which includes tail bits or tail symbols further comprises the steps of:

initializing first and second storage element to zero, a third storage element to zero, said condition indicator to zero, said counter value to said traceback length and said storage unit pointer to a constant;

updating said third storage element wherein the least significant bit is equal to said bit location 1 of said first storage element and the most significant bit is equal to said condition indicator;

5 determining the condition of said first storage element and update said condition indicator;

reading a word from said storage unit to said first storage element and increasing the storage unit pointer;

determining said number of state location in said first storage element to be right shifted;

10 shifting and masking said first storage element;

updating said third storage element wherein the least significant bit is equal to said bit location 1 of said second storage element and the most significant bit is equal to said condition indicator;

15 determining the condition of said second storage element and update said condition indicator;

reading a word from said storage unit to said second storage element and increasing the storage unit pointer;

determining said number of state location in said second storage element to be right shifted;

20 shifting and masking said second storage element;

determining said value of condition indicator to perform exclusive-or operation to obtain said result outputs and decreasing said counter value by one; and

determining said counter value;

25 and wherein said step of updating said third storage element wherein the least significant bit is equal to said bit location 1 of said second storage element and the most significant bit is equal to said condition indicator will be repeated by interchanging the role of said first storage element with said second storage element and vice versa if said counter value is not equal to a constant value otherwise the operation will end.

7. The method of claim 3, wherein said step of determining said order further includes bit-reversing every current and next state in a symbol interval.

8. The method of any one of claims 4, 5 or 6, wherein said step of determining the condition of said first storage element further comprises the steps of:

updating the value of said condition indicator to zero if the value of said first storage element is even; and

updating the value of said condition indicator to one if the value of said first storage element is odd.

10

9. The method of any one of claims 4, 5 or 6, wherein said step of determining the condition of said second storage element further comprises the steps of:

updating the value of said condition indicator to zero if the value of said second storage element is even; and

15 updating the value of said condition indicator to one if the value of said second storage element is odd.

10. The method of any one of claims 4, 5 or 6, wherein said step of determining said number of state location in said first storage element to be right shifted further comprises the steps of:

20

right shifting said number of state location which is equal to the value in said second storage element if the value of said condition indicator is even; and

right shifting said number of state location which is equal to the summation of said value in said second storage element and a dependent constant, if the value of said condition indicator is odd.

25

11. The method of any one of claims 4, 5 or 6, wherein said step of determining said number of state location in said second storage element to be right shifted further comprises the steps of:

right shifting said number of state location which is equal to the value in said first storage element if the value of said condition indicator is even; and

right shifting said number of state location which is equal to the summation of said value in said first storage element and a dependent constant, if the value of said condition indicator is odd.

12. The method of either one of claims 5 or 6, wherein said step of determining said value of condition indicator further comprises the steps of:

performing exclusive-or operation between said third storage element and said first storage element to get said result outputs if the value of said condition indicator is even; and

performing exclusive-or operation between said third storage element and said first storage element and a dedicated logic to get said result outputs if the value of said condition indicator is odd.

13. The method of either one of claims 5 or 6, wherein said step of determining said value of condition indicator further comprises the steps of:

performing exclusive-or operation between said third storage element and said second storage element to get said result outputs if the value of said condition indicator is even; and

performing exclusive-or operation between said third storage element and said second storage element and a dedicated logic to get said result outputs if the value of said condition indicator is odd.

14. The method of either one of claims 10 11, wherein said dependent constant is the value of said total number of states of said encoder divided by two.

15. The method of either one of claims 12 or 13, wherein said dedicated logic has (n-1) bits which the value depends on the nature of said encoder.

16. A method for performing a traceback operation in Viterbi decoding for decoding a stream of code signals, said method comprising at least the step of sequentially performing traceback operations on a current frame starting from a state through which a path of a previous frame passes.

5