Varsha P. Patil, Prof. D. G. Chougule / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 2,Mar-Apr 2012, pp.1303-1307

ARM based Implementation of Viterbi Algorithm for Wireless Data Communication.

Varsha P. Patil [1], Prof. D. G. Chougule [2]
1 Tatyasaheb Kore Institute of Engineering & Technology, Warananagar, India
2 Tatyasaheb Kore Institute of Engineering & Technology, Warananagar, India

ABSTRACT:
Convolution encoding with Viterbi decoding is a powerful method for forward error correction. It has been widely deployed in many wireless communication systems to improve the limited capacity of the communication channels. Viterbi algorithm, is the most extensively employed decoding algorithm for convolutional codes. In this paper, we present ARM based implementation of Viterbi decoder that uses survivor path with parameters for wireless communication in an attempt to reduce the power and cost and at the same time increase the speed. The decoder was simulated using Keil software.

Keywords-CommunicationViterbi, Encoder, Decoding, Algorithm

I.INTRODUCTION

In wireless communication AWGN (Additive White Gaussian Noise) properties of most of the communication media introduce noise in real data during transmission. The approach to error correction coding taken by modern digital communications system starts with the ground breaking work of Shannon, Hamming and Golay [1-3]. Channel Coding is a technique to introduce redundant code in real code to remove interference and error during transmission. Coded data in sender side thus increased by volume and error effect becomes less compare with uncoded data. Receiver end receives this data and decodes the data using some techniques. Viterbi decoding is one of the popular techniques to decode data effectively. Viterbi algorithm (VA) is an optimum decoding algorithm for the convolutional code. Convolutional encoding and Viterbi Decoding are widely used for reliable data transmission.

The task facing the designer of a digital communication system is that of providing a cost-effective facility for transmitting information from one end of the system at a rate and a level of reliability and quality that are acceptable to the user at the other end. Viterbi decoding was developed by Andrew J. Viterbi in 1967 [1] . Since then, other researchers have expanded on his work by finding good Convolutional codes, exploring the performance limits of the technique, and varying decoder design parameters to optimize the implementation of the technique in hardware and software. The Viterbi decoding algorithm is also used in decoding trellis-coded modulation technique which is used as an effective coding technique for band limited channels as in telephone line modems to squeeze high ratios of bits-per-second in 3 kHz-bandwidth analog telephone lines [1,2].

For years, convolutional coding with Viterbi decoding has been the predominant Forward error correction (FEC) technique used in space communications, particularly in geostationary satellite communication networks, such as VSAT (very small aperture terminal)[1].

II. CONVOLUTIONAL CODES

Convolutional codes are commonly specified by three parameters; (n, k, m), where: n is the number of output bits, k is the number of input bits, and m is the number of shift register stages of the coder. The constraint length L of the code represents the number of bits in the encoder memory that affect the generation of the n output bits and is defined as L = m k. The code rate r of the code is a measure of the code efficiency and is defined by r = k/n [4]

1.1. Code Parameters and the Structure of the Convolutional Code

Figure1 shows the convolution encoder structure CC (2, 1, 3) used in this paper and is built from its parameters. It consists of 3 (m=3) shift register stages and two modulo-2 adders (n= 2) giving the outputs of the encoder. The rate of the code is r = 1/2. The outputs of the adders are sampled sequentially yielding the code symbols. The outputs V1 and V2 of the adders are governed by the following generator polynomials:

V1=U0 XOR U1 XOR U2 (1)

V2=U0 XOR U2 (2)

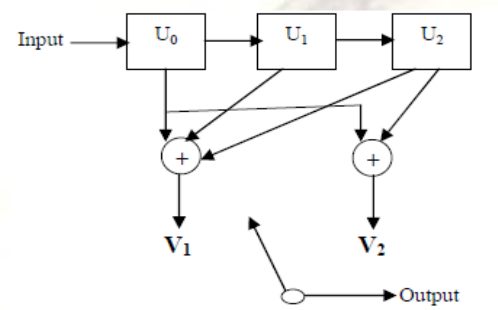

Figure 1- Convolutional code CC(2 , 1, 3)

1.2. Coding a Message Sequence

In this section a 4 bit sequence of 1001 is coded to show how the encoder works. The bits are passed through the encoder sequentially as shown in table 1 and the outputs of the encoder are calculated using equations (1) and (2) for each time step.

Time instant	Input	U0	U1	U2	V0	V1
t=0		0	0	0	0	0
t=1	**1**	**1**	0	0	1	1
t=2	**0**	**0**	**1**	0	1	0
t=3	**0**	**0**	**0**	**1**	1	1
t=4	**1**	**1**	**0**	**0**	1	1
Reset process	0	0	**1**	**0**	1	0
	0	0	0	**1**	1	1
	0	0	0	0	0	0

Table 1-Coding process of CC (2, 1, 3)

The data in bold are the input bits which are shifted in the memory register for every time clock. The code output sequence is: 11 10 11 11 10 11 00 where the last 6 bits (= nm) are the reset bits and they do not contain any information.

1.3 Trellis Diagram

Trellis diagrams represent linear time sequencing events. They can also be used to encode messages. The trellis diagram of CC (2, 1, 3) encoder shown in figure 2 is drawn from table 2. It is built by using the Horizontal axis as a discrete time and all possible 2(L-1) states lined up on the vertical axis. One moves through the trellis every time step. New bits arrive every time step. Each state is connected to the next state by the allowable codeword's for that state. There are only two possible choices at each state. These are determined by the arrival of input bits 0 or 1 which are shown between brackets in the diagram. It can be noticed that the arrows go upwards for a 0 input bit and they go downwards for a 1 input bit. The trellis diagram is unique to each code.

Input bits	Input states		Output bits		Output states	
I	**S1**	**S2**	**V1**	**V2**	**S1**	**S2**
0	0	0	0	0	0	0
1	0	0	1	1	1	0
0	0	1	1	1	0	0
1	0	1	0	0	1	0
0	1	0	1	0	0	1
1	1	0	0	1	1	1
0	1	1	0	1	0	1
1	1	1	1	0	1	1

Table2- Input and output states related to the input bits.

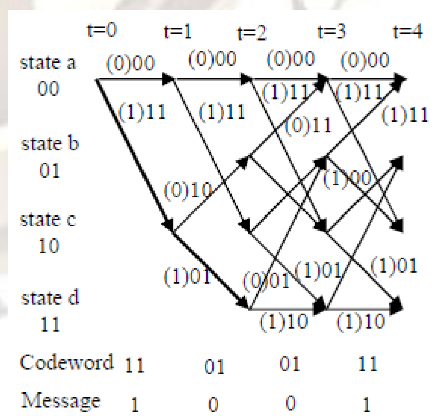

Figure 2 – Trellis diagram for the code CC (2, 1, 3)

For instance to encode 1100 message, one starts from the state 00 and goes downwards for each input bit 1 and goes upwards for each input bit 0. After 4 time steps which correspond to the number of input bits, one gets the encoded word 11 01 01 11 the trellis diagram of figure.2 will be used later to decode code words.

III. DECODING USING LIKELIHOOD AND VITERBI ALGORITHM

There are several different approaches to decoding of convolutional codes. These are grouped in two basic categories: sequential decoding (Fano algorithm), Maximum likelihood decoding (Viterbi algorithm). Both these methods represent two different approaches to the same basic idea behind decoding [3].

Assume that 4 bits were sent via rate 1/2 code. One receives 8 bits. These 8 bits may or may not have errors. One knows from encoding process that these bits map uniquely. So a 4 bit sequence will have a unique 8 bit output. But due to errors one can receive any and all possible combinations of the 8 bits, the permutation of 4 input bits results in 16 possible input sequences. Each of these has a unique mapping to an 8 bit output sequence by the code as shown in table.3

These form the set of permissible sequences and the encoder's task is to determine which one was sent. Let us say one received 11111100. It is not one of the 16 possible sequences shown in table2. How does one decode it? One can do two things

1. Compare this received sequence to all Permissible sequences and pick the one with the Smallest Hamming distance

2. Perform a correlation and pick a sequence with the best correlation.

Message number	Message	Codeword
1	0000	00 00 00 00 00 00 00
2	0001	00 00 00 11 10 11 00
3	0010	00 00 11 10 11 00 00
4	0011	11 01 01 11 00 00 00
5	0100	00 11 10 11 00 00 00
6	0101	11 01 00 01 11 00 00
7	0110	00 11 01 01 10 00 00
8	0111	11 01 10 01 01 00 00
9	1000	00 00 00 11 10 11 00
10	1001	11 10 11 11 10 11 00
11	1010	00 11 10 00 10 11 00
12	1011	11 01 01 00 10 11 00
13	1100	11 01 01 11 00 00 00
14	1101	11 10 00 01 01 11 00
15	1110	11 01 10 01 11 00 00
16	1111	11 01 10 10 01 11 00

Table-3 Code words of different 4-bit messages

.

2.1 Maximum Likelihood and Viterbi decoding

The Viterbi algorithm [5] examines an entire received sequence of a given length. The decoder computes a metric for each path and makes a decision based on this metric. All paths are followed until two paths converge on one node. Then the path with the higher metric is kept and the one with the lower metric is discarded. The selected paths are called the survivors. For an N bit sequence, the total number of possible received sequences is2^N; only $2^{K(L-1)}$ of these are valid. The Viterbi algorithm applies the maximum likelihood principles to limit the comparison to $2^{K(L-1)}$surviving paths instead of checking all paths. The most common metric used is the hamming distance metric. This is just the dot product between the received code word and the allowable codeword. These metrics are cumulative so that the path with the largest total metric is the final winner.The metric branch $mj^{(\alpha)}$ at $j^{(th)}$instant is given by:

$$mj^{(a)} = Ln \prod_{i=1}^{n} p\left(\frac{r_{ij}}{c_{ji}^{(\alpha)}}\right) = \sum_{i=1}^{n} Ln\left(p\left(\frac{r_{ij}}{c_{ji}^{(\alpha)}}\right)\right) \quad (3)$$

Where r_{ij} is the $i^{(th)}$bit received at $j^{(th)}$ instant. And c_{ji} is the $i^{(th)}$bit transmitted at instant. The metric path $m^{(\alpha)}$ of the path $j^{(th)}$ at the $j^{(th)}$ instant is the sum of the metric branches of the path α from the first instant to the $j^{(th)}$ instant and is given by:

$$M^{(\alpha)} = \sum_j^J mj^{(a)} \quad (4)$$

The winner is the path with the highest metric path. Let's decode the received sequence 11 01 01 11 to show how the Viterbi algorithm works. The technique uses the trellis diagram of the CC (2, 1, 3) given in figure 2. It is supposed that the transmission channel is free of noise, which means that the transmitted sequence is the same as the received sequence.

2.2 The procedure used in The Viterbi algorithm is as follows

At the instant t=1:

The received symbols are 11.They are compared to the possible transmitted symbols 11 and 00 of the branches-
a→ a & a→ c respectively. The metrics of the branches are respectively 0 and 2.

At the instant t=2:

The received symbols are 01.They are also compared to the possible transmitted symbols 00, 11, 01 of the branches a→a, a→c, a→d respectively. The metrics of the branches in this case are 1, 1, 2

At the instant t= 3:

The following branches meet at a node:

-At the node (a =00, t =3): the paths a→a→a→a and a→ c→b →a and their metric paths are 2, 3 respectively. Therefore the path a→a→a→a is discarded.

-At the node (b =01, t =3): the paths a→a→c→b and a→c→d→b and their metric paths are 1, 4 respectively. Therefore the path a→a→c→b is discarded

-At the node (c =10, t =3): the paths a→a→a→c and a→c→b→c and their metric paths are 2, 3 respectively. Therefore the path a→a→a→c is discarded

-At the node (d =00, t =3): the a→c→d→d and a→a→c→d and their metric paths are 4, 2 respectively. Therefore the path a→a→c→d is discarded. Therefore at the instant t = 3 there are 4 Survivors: a→c→b→a, a→c→d→b, a→c→b→c, a→c→d→d

2.2.1 A correct decoding of a corrupted sequence with two errors

The precedent technique demonstrates how the Viterbi algorithm decodes a codeword into its message. In this section, the notion of channel error [6] is considered through an example. Suppose that a code word 00 00 00 00 (00 00 00) of 0000 message is transmitted and at the receiver the sequence 01 00 01 00 (00 00 00 00) is received where the digit '1' of the second and sixth symbols are the errors introduced by the transmitting channel. Using the trellis diagram of figure 2 and the steps of decoding described above, at the instant t =4, the four survivors shown in table 4 are obtained. The path a→a → a→a→a having the highest metric (= 8) is the winner. The decoder chooses this path which gives 0000 as a decoded message which is the correct result.

	Path	Metric Path	Possible Decoded Message
1	**a→a→a→a→a**	**2+2+2+2=8**	**0000**
2	a→a→a→a→d	2+2+2+0=6	0001
3	a→a→a→c→b	2+2+0+1=5	0010
4	a→a→a→c→d	2+2+0+1=5	0011

Table 4- Survivor paths of the trellis diagram for the Corrupted sequence 01000100(000000)

IV. IMPLEMENTATION OF VITERBI ALGORITHM USING ARM

There are different approaches to implementation of convolutional encoder & viterbi decoder. Here we implement viterbi algorithm using ARM processor i.e.lpc2138. To reduce complexity & increase speed this method is used.For the simulation purpose we use keil software.

Steps:

1) First we develop PC interfacing code with ARM CONTROLLER in embedded C
2) Development Viterbe encoder code in embedded C
3) ARM controller interfaced with RF transmitter in embedded C for wireless transmissions we have use either GSM or zigbee.

4) ARM CONTROLLER interfaced with PC and data bits transmitted serially through UART
5) Same set up present at receiver side also virebi algorithm is implemented at receiver side to perform decoding of data stream.

V.CONCLUSION:

In this work, error detection and correction techniques have been used which are essential for reliable communication over a noisy channel. The effect of errors occurring during transmission is reduced by adding redundancy to the data prior to transmission. A convolutional code with Viterbi decoding is used in wireless communication & satellite communication. The various examples are cellular phone i.e. GSM, IS-54 digital cellular phone standards and IS-95 CDMA standard, modems and video and audio broadcasting. Motivation for low power has been derived from needs to increase the speed, to extend the battery life and to reduce the cost.

VI. REFERENCES

[1] **Chip Fleming**, "A tutorial on convolutional coding with Viterbi algorithm", 2003.

[2] **ManjeetSingh, Ian J. Wassel** "Comparison between Soft and Hard decision decoding using quaternary convolutional encoders and the decomposed CPM model", Laboratory for communications Engineering, Department of Engineering, University of Cambridge.

[3] **Charon Langton**, "Coding and decoding with conventional codes", www.complextoreal.com,July 1999

[4] **A. J. Viterbi**, "Convolutional codes and their performance in communication systems,"IEEE ***Trans. Commun.,*** vol. COM-19, pp. 751 -772, Oct., 1971.

H. Yang and X. Lang, "Design and implementation of high speed and area cefficient Viterbi decoder," IEEE, Proceedings of the 8th International Conference on Solid State and Integrating Circuit Technology, ICSICT'06, pp. 2108–2110, 2006

[5] **[Vit67] A. J. Viterbi**, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Trans. on Inf. Theory, vol. IT-13, pp.260-269, 1967