www.ijramr.com

ISSN : 2350-0743

International Journal of Recent Advances in Multidisciplinary Research
Vol. 03, Issue 07, pp.1646-1648, July, 2016

REVIEW ARTICLE

AN EFFICIENT DESIGN OF DYNAMIC VITERBI DECODER ARCHITECTURE

*[1]Parsekar Gaurav Rajanikant, [1]Veena, H.S. and [2]Girish G Satardekar

Bangalore Institute of Technology/ECE Dept., Bangalore, Karnataka, India
Goa College of Engineering/E&TC Dept., Farmagudi, Ponda, Goa, India

ARTICLE INFO

Article History:
Received 16th April, 2016
Received in revised form
21st May, 2016
Accepted 29th June, 2016
Published online 31st July, 2016

Keywords:
Emerging applications,
Communication
Protocols.

ABSTRACT

The emerging applications of wireless networks enforce new challenges in design of algorithms and communication protocols. In such scenario of challenges, coding for error control has be- come extremely important to provide robust communication and maintain quality of service. One method to improve Bit Error Rate (BER) while maintaining high data reliability, is to use an error correction technique like the Viterbi algorithm. The Viterbi algorithm provides an efficient method for Forward Error Correction (FEC) that improves channel reliability. As constraint length associated with input bits increases it needs to implement it with lesser computations and lesser hardware to decode the original data. Therefore Dynamic Viterbi Algorithm is used for decoding which reduces error probability, computation and employ lesser hard- ware with increased speed. The purpose of this paper is to understand Viterbi Algorithm, Adaptive Viterbi Algorithm and to find the alternative to shortcomings in the design and implement the idea on a hardware.

INTRODUCTION

In most communication system nowadays, a convolutional code is a kind of error correcting code that generates parity symbols via the sliding application of a Boolean function to a data stream [1]. The sliding application represents the 'convolution' of the encoder over the data, which gives rise to the term 'convolutional coding.' The sliding nature of the convolutional codes facilitates trellis decoding using a time-invariant trellis. Time invariant trellis decoding allows convolutional codes to be maximum likelihood soft-decision decoded with reasonable complexity. The ability to perform economical maximum likelihood soft decision decoding is one of the major benefits of convolutional codes. This is in contrast to classic block codes, which are generally represented by a time-variant trellis and therefore are typically hard-decision decoded. Viterbi decoding algorithm is being widely used in many wireless and mobile communication systems for optimal decoding of convolutional codes.

Convolutional Encoder

Convolutional encoding is widely used in communication systems as forward error correction scheme. Convolutional codes are also used as a building block in more powerful modern codes, such as turbo codes, which are used in wide-area cellular wireless network standards such as 3G, LTE, and 4G.

Corresponding author: Parsekar Gaurav Rajanikant,
Bangalore Institute of Technology/ECE Dept., Bangalore, Karnataka, India

This encoding scheme is often used in space communications and wireless communications. A convolutional encoder is a finite state machine, where the output is a function of the current state and the current input. Opposed to block codes convolutional codes don’t divide data into blocks.

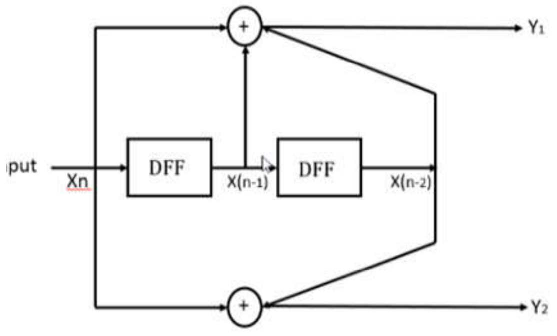

Fig. 1. Convolutional Encoder (rate=1/2)

The convolutional encoder in Figure 1 produces two output bits for single bit of input signal, therefore it is called a rate 1/2 encoder. Convolutional encoder are commonly specified by three parameters; (n,k,m).

n = number of output bits k
= number of input bits
m = number of memory registers (flip flops)

The quantity *k/n* called the code rate, is a measure of the efficiency of the code. Commonly *k* and *n* parameters range from 1 to 8, *m* from 2 to 10 and the code rate from 1/8 to 7/8 except for deep space applications. Where code rates as low as 1/100 or even longer have been employed. The constraint length, K, is the number of input frames that are held in the K-bit shift register. Convolutional encoder can be described in terms of state table, state diagram and trellis diagram.

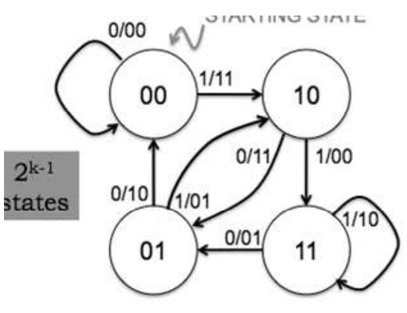

Fig. 2. State Diagram

Viterbi Decoder

This section describes various parts of Viterbi de- coder and algorithm [2]. The Viterbi decoder receives successive code symbols, in which the boundaries of the symbols and the frames have been identified. The Viterbi decoder is basically divided into two types

A. Hard decision Viterbi decoding

In received signal, the voltages have been digitized to form a received bit sequence. If we decode this received bit sequence, the decoding process is termed hard decision decoding (aka "hard decoding")[3]. In the hard-decision decoding, the path through the Trellis is determined using the Hamming distance measure. Thus, the most optimal path through the trellis is the path with the minimum Hamming distance. In this type of decoding it shows only two possibilities, one is the presence of signal and the other is absence of signal i.e. either '1' or '0'. The Hamming distance can be defined as a number of bits that are different between the observed symbol at the decoder and the sent symbol from the encoder. Furthermore, the hard decision decoding applies one bit quantization on the received bits.

B. Soft decision Viterbi decoding

Soft-decision decoding is applied for the maximum likelihood decoding, when the data is transmitted over the Gaussian channel. On the contrary to the hard decision decoding, the soft decision decoding uses multi- bit quantization for the received bits, and Euclidean distance as a distance measure instead of the hamming distance [3].

C.Viterbi Decoder Architecture

A hardware Viterbi decoder for basic (not punctured) code usually consists of the following major blocks:

Branch metric unit (BMU)

Add Compare Select Unit (ACSU) Survivor memory unit (SMU). The block diagram of the hardware Viterbi decoder is shown in Fig. 3.

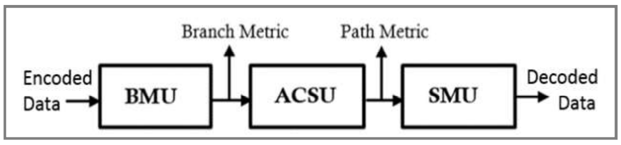

Fig. 3. Viterbi decoder architecture

In the above fig the encoded data input is given to the branch metric unit which gives the branch metrics, and these branch metrics are again given to the add compare and select unit which again produces the path metrics from which using trace back method the de- coded stream is obtained[6].

Viterbi Algorithm

The Viterbi decoding algorithm, proposed in 1967 by Viterbi, is a decoding process for convolutional codes in memory-less noise. The algorithm can be applied to a host of problems encountered in the design of communication systems. The Viterbi Algorithm (VA) finds the most- likelihood path transition sequence in a state diagram, given a sequence of symbols [5].

A Viterbi algorithm consists of the following three major parts:

A. Branch metric calculation

Calculation of a distance between the input pair of bits and the four possible "ideal" pairs ("00", "01", "10", "11") encoder. Here the received data symbols are com- pared to the ideal outputs of the encoder from the transmitter and branch metric is calculated.

B. Path metric calculation

The PMU calculates new path metric values and decision values. Because each state can be achieved from two states from the earlier stage, there are two possible path metrics coming to the current state. The core elements of a PMU are ACS (Add-Compare-Select) units.

C. Trace back

This step is necessary for hardware implementations that don't store full information about the survivor paths, but store only one bit decision every time when one survivor path is selected from the two.

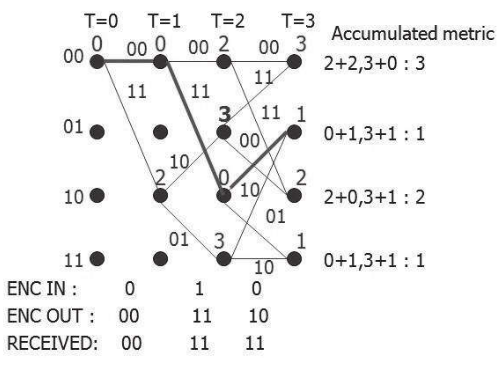

Fig. 4. Trellis Diagram

Adaptive Viterbi Decoder (Avd)

The aim of the adaptive Viterbi algorithm is to reduce the average computation and path storage required by the Viterbi algorithm. Instead of computing and retaining all 2*K-1* possible paths, only those paths which satisfy certain cost conditions are retained for each received symbol at each state node. Path retention is based on the following criteria.

- A threshold T indicates that a path is retained if its path cost is less than dm + T, where dm is the minimum cost among all surviving paths in the previous trellis stage.
- The total number of survivor paths per trellis stage is limited to a fixed number, Kmax, which is pre-set prior to the start of communication.

Path metrics are marked in bold on the nodes; dot lines indicate the least error path, upper branch indicates input bit ‘0’, lower branch indicates input bit ‘1’. Output bits corresponding to the given input bit and state is shown on the branches.

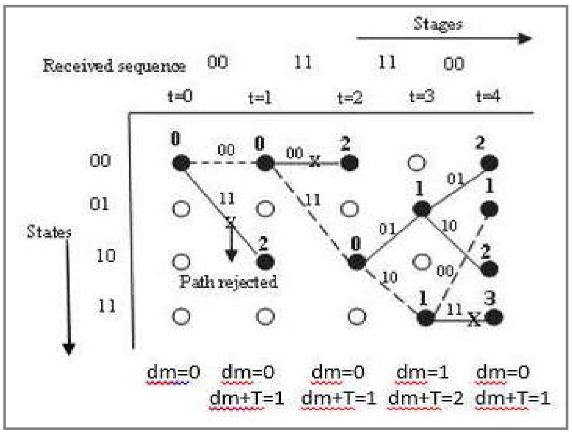

Fig. 5. Trellis diagram for a hard-decision Adaptive Viterbi Decoder with T = 1 and K = 3

Dynamic Viterbi Decoder

As seen in above Viterbi algorithm and Adaptive Viterbi algorithm, there is lot of computation involved and chances of error probability. Viterbi algorithm calculates maximum likelihood path through the trellis but it involves lot of computation since it calculates all the branch distances within a trellis and compares at the end the least metric path. As a modification to Viterbi algorithm is done in adaptive Viterbi algorithm where average computation and path storage required by the AVD are reduced .Also in- stead of computing and retaining all 2^{k-1} possible paths, only those paths which satisfy certain cost conditions are retained for each received symbol at each state node. But in the process of reducing the computation by half, there is increase in probability of error while decoding. The solution to the shortcomings of both the above methods is proposed in this paper.

In Dynamic Viterbi Decoder we choose the threshold according to the nature of the transmitting channel. In- stead of choosing threshold T=1 as seen in above ex- ample of Adaptive Viterbi Decoder, we set threshold by understanding the channel. If the channel is very noisy threshold value increases and vice versa. For this design initially a training sequence is sent from encoder and transmitted through the channel. The decoder output is compared to encoder output, this determines the number of errors occurred while trans- mission. From this we set the threshold as this value is related to number of errors, (number of errors = threshold value). Considering example shown in fig 5 of Adaptive Viterbi Decoder, if both the initial bits (00) were in error practically but according to AVD example if we considered (11) to have received then we would have to choose other path which is actually rejected in AVD. This design overcomes this problem by choosing threshold in such a way that it will allow paths not to reject directly but retain until the cost conditions are satisfied. in this design there is a complexity of computation but error probability is checked.

Conclusion

This paper starts with a study of understanding of Viterbi decoder and optimisation techniques of the same. This leads to scheme called Adaptive Viterbi Decoder algorithm which reduces computation by almost 50% and higher speed but it gives rise in error probability. The Dynamic Viterbi Decoder proposed in this paper overcomes the computational and hardware complexity of Viterbi decoder, also it reduces the error probability of Adaptive Viterbi Decoder.

REFERENCES

Arun, C. and P. Rajamani, “Design and VLSI imple mentation of a low probability of error Viterbi decoder,” First international conference on Emerging trends in Engineering and technology, pp. 418-423, 2008.

Gemmeke, T., Gierenz, V. S. and Noll, T. G. 2006. “RTL implementation of Viterbi decoder,” Dept. Of Computer Engineering, IEEE Transactions on circuits and systems, June.

Kalavathidevi, T. and Venkatesh, C. 2011. “Area Efficient Low Power VLSI Architecture for A Viterbi Decoder Using Gate Diffusion Input(GDI) Logic Style”, *European Journal of Scientific Research*, Vol. 49, no.4, pp. 521-532, March.

Mahender Veshala, Tualsagari Padmaja, Karthik Ghanta, “FPGA Based Design and Implementation of Modified Viterbi Decoder for a Wi-Fi Receiver,” IEEE Conference on Information and Communication Technologies (ICT 2013), 2013.

Swaminathan, S., Tessier, R. Goeckel, D. and Burleson, W. 2002. “A dynamically reconfigurable adaptive Viterbi decoder,” Monterey, California, USA, February 24-26, 2002.

****