ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 2, August 2012

Error Dependent Soft-Decision Power Saving Viterbi Decoder for Mobile Devices

Offor K. J., Mbachu C. B., Okonkwo I. I.

***Abstract*—This paper investigates the Viterbi Algorithm for decoding convolutionally coded messages for forward error correction. A new power saving strategy that may enable mobile receivers to decode convolutionally coded transmissions with lower energy utilization using soft decision decoding techniques is developed. The idea behind the work is the selective use of the Viterbi decoder only when error is present in the transmitted message. When there is no error, a simple decoder is used. The analysis of the proposed method was done using the profiler tool in MATLAB®. Results obtained by MATLAB® simulation demonstrate that, there are no increases in bit-error probability as a result of the proposed method. The MATLAB processing time (and hence energy) shows that, in comparison to that obtained with a standard Viterbi decoder, the proposed method uses less energy at EbNo above 3dB.**

Index Terms— **Convolution Codes, Error Dependent, Viterbi Algorithm, Power Saving, Soft Decision.**

I. INTRODUCTION

The Viterbi algorithm which was proposed by Andrew J. Viterbi [13] in 1967 is a maximum likelihood decoder used in decoding convolutionally coded messages. Because of the complexity of the algorithm, it is energy intensive irrespective of whether the received signal contains errors or not. As a result of globalisation, there has been proliferation of mobile receivers/devices especially handsets, wireless sensor networks (WSN) and Mobile Adhoc Networks (MANET). These devices consume a lot of energy especially in decoding the encoded signals. Energy is a scarce and economic commodity the world over. There is therefore the need to design a decoder that conserves energy in mobile devices. Many researchers have been researching and reporting on ways of minimising energy used in the decoding [5, 2, 3, 10, 11, 12, 14, 16, 17, 18, 19]. Olaf J. Joeressen and H. Meyr [17] based their design on the two steps soft out Viterbi algorithm proposed by [19]. In their design, instead of constructing competing paths for all states and performing the update operation along the paths for all the states, they output only the information along the single decoded state transition sequence (the final survivor). Oh and Hwang [12] proposed a trace back scheme to cut down the switching activities incurred while tracing back. Their scheme is designed for a decoder, where the trace back starts before the end of the code word. The key idea is to reuse the information from the previous trace to shorten the trace back. [19] Used clock gating – In clock gating, the clock of each register is enabled only when the register updates it survivor path information. Their simulations showed a 30% reduction in dynamic power dissipation which gives a good indication of power reduction on implementation. [5] Investigated the use of an algorithm proposed by Barry Cheetham for selective use of Viterbi decoder in decoding codes when there is error and the use of a simple decoder when there is no error. He employed hard decision decoding and concluded that the new algorithm gives better performance above 5dB when compared to the standard Viterbi decoder. The purpose of this paper is to propose a new and adaptive error dependent power saving Viterbi decoder based on the work done by [5] using soft-decision decoding. In a conventional Viterbi decoder, the complex and energy consuming Viterbi decoder is always used to decode a message irrespective of whether the message contains error or not. In this approach, an error dependent version of the algorithm is proposed. It consists of two parts, one simple decoder and the second a normal Viterbi decoder. The simple decoder is used when there is no error in the transmission and the Viterbi decoder is used only when the transmission contains errors.

II. DESIGN METHODS FOR THE VITERBI ALGORITHM

This research will involve the study of energy saving in communications system through observation and evaluation of performance of the new algorithm in comparison with conventional systems, the simulation/empirical approach is employed [4, 6, and 9]. This is achieved by modelling a communications system in Matlab and modifying the decoding section to minimize energy consumption. The Viterbi Algorithm was developed by Andrew J. Viterbi and first published in the IEEE transactions journal on Information theory in 1967 [13]. It is a maximum likelihood decoding algorithm for convolutional codes. This algorithm provides a method of finding the branch in the trellis diagram that has the highest probability of matching the actual transmitted sequence of bits. Since being discovered, it has become one of the most popular algorithms in use for convolutional decoding.

1. The Branch Metric Unit (BMU) which calculates the BMs;
2. The Path Metric Unit (PMU) includes a number of Add Compare Select Units (ACSU) which add the BMs to the corresponding PMs, compares the new PMs, and select the PMs indicating the most likely path; at the same time, the

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 2, August 2012

PMU passes the associated survivor path decisions, called local winners, to the Survivor Memory Unit (SMU);

3. The Survivor Metric Unit (SMU) which stores the survivor path decisions; then the accumulated history in the SMU is searched to track down the most likely path so that the decoded sequence can be decided.

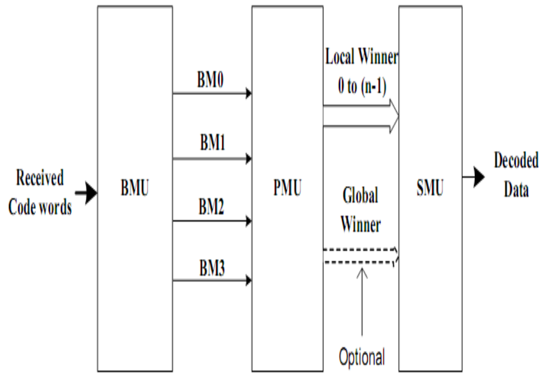

Fig. 1. Classical Three Functional Block of a Rate 1/2 Viterbi Decoder Design.

III. BMU DESIGN

The operation of BMU is crucial as it is the first stage of the Viterbi algorithm and the consequent decoding process depends on all the information it provides. In a hard-decision Viterbi decoder, the BMU design is straightforward since the BMs are the Hamming distances between the received code words and expected branches. For a soft-decision Viterbi decoder, the received code words are quantised into different levels according to the signal strength then the BMU maps the levels of code words into BMs according to their likelihood. In hard-decision the Hamming weight of the code word is used as the branch metric, which is simply the number of positions in which the received code word differs from the ideal code word. The case of soft-decision can be derived from the generalized unquantised (analogue) channel. For an unquantised channel, assume binary antipodal signalling is used with a convolutional code of rate m/n. If a code word S, which consists of n symbols, x0 x1 · · · xn−1, is transmitted through the channel, the decoder receives R which is a sequence of n sampled actual voltages, r0 r1 · · · rn−1, from the filter. The conditional probability of sending S and receiving R is [7]

$$P(S|R) = P(R| \text{ ---------------- } 1$$

If the transmitted code words have an equal probability, an optimum decoder identifies the S which maximizes P (R|S) so that the maximum P (S|R) can be achieved. Since a code word has n symbols, for the Gaussian noise with zero mean and variance σ2 = No/2 where No is the noise power spectral density, P(R|S) becomes the product of n Gaussian density functions of each symbol. As given in [7]

$$P(R|S) = \prod_{i=0}^{n-1} P(r_i|s_i) = \left[\frac{1}{(\pi \times N_0)^{1/2}}\right]^n \times exp\left[-\sum_{i=0}^{n-1}\frac{(r_i - s_i)^2}{N_0}\right] \quad 2$$

For a specific noise level, the P (R|S) is maximized when

$$d^2 = \sum_{i=0}^{n-1}(r_i - s_i)^2 \text{---} 3$$

is minimized, where d^2 is the squared Euclidian distance between the hypothesized sequence and the received signal. For an unquantised channel, d^2 can be used as the measurement of the unlikelihood of the code word branch, e.g. the branch metric, since a minimum value of d^2 indicates the most likely branch and its accumulated value indicates the most likely path. This squared Euclidian distance is defined in [7] as the generalised concept of the distance between the received and ideal code words.

IV. SOFT DECISION/HARD DECISION

In hard-decision decoding, the decoder operates on data that take on a fixed set of possible values (typically 0 or 1 in a binary code) [20]. Hard decision decoding takes a stream of bits say from the 'threshold detector' stage of a receiver, where each bit is considered definitely one or zero. For binary signaling, received pulses are sampled and the resulting voltages are compared with a single threshold. If a voltage is greater than the threshold it is considered to be definitely a 'one' say regardless of how close it is to the threshold. If it's less, it's definitely zero. For a Binary Symmetric Channel (BSC), the maximum decoding is equivalent to choosing a codeword $U^{(m')}$ that is closest in Hamming distance to the received decoder sequence Z. Thus the Hamming distance is an appropriate metric to describe the distance or closeness of fit between $U^{(m')}$ and Z. From all the transmitted sequences $U^{(m')}$, the decoder chooses the $U^{(m')}$ for which the distance to Z is minimum. Suppose that $U^{(m')}$ and Z are each L-bit-long sequences and that the Hamming distance between them is d_m. The conditional probability of receiving a channel symbol correctly or incorrectly is expressed as:

P(0|1)=P(1|0)=p and P(1|1)=P(0|0)=1-p ----------- 4

Taking equation (1) into consideration, the likelihood function can be expressed as:

$$P(Z|U^{(m)}) = p^{d_m}(1-p)^{L-d_m} \text{ --------------------- } 5$$

And he log-likelihood function is:

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 2, August 2012

$$logP\left(Z|U^{(m)}\right) = Llog\left(1-p\right) - d_m \, log\left(\frac{1-p}{p}\right) \text{------ } 6$$

Where p is the probability of 1 or 0.
It should be noticed that the last term of the above equation is constant for each possible transmitted sequence. Assuming that p<0.5, we can re-write Equation (3.10) as

$$logP\left(Z|U^{(m)}\right) = -d_m A + LB) \quad \text{------- } 7$$

Where A and B are both positive constants.
Therefore, finding the closest coded sequence U $^{(m')}$ to the received sequence Z is equal to maximizing the likelihood or log-likelihood metric. Consequently, log-likelihood metric can be conveniently replaced by the Hamming distance if the channel is BSC. And a maximum likelihood decoder will choose a path $U^{(m')}$ for which the Hamming to Z is minimum in the tree of trellis diagram. In soft decision decoding, the inputs to a soft-decision decoder may take on a whole range of values in-between. This extra information indicates the reliability of each input data point, and is used to form better estimates of the original data. Thus, soft decision decoding requires a stream of 'soft bits' where we get not only the 1 or 0 decision but also an indication of how certain we are that the decision is correct. It furnishes the decoder with more information than is provided in hard-decision decoding. For a Gaussian channel, the soft decision bits the demodulator sends to the decoder can be viewed as a family of conditional probabilities of the different symbols. It can be verified that maximizing $P(Z|U^{(m)})$ is equivalent to maximizing the inner product between the codeword sequence U $^{(m')}$ (consisting of binary symbols represented as bipolar values) and the received sequence Z [8] .Thus, the decoder chooses the codeword U $^{(m')}$ if it maximizes

$$P\left(Z|U^{(m)}\right) = \sum_{i=1}^{\infty}\sum_{j=1}^{n} Z_{ji}\, U_{ji}^{m} \quad \text{------} \qquad 8$$

This is equivalent to choosing the codeword U $^{(m)}$ that is closest in Euclidean distance to Z. Even though the hard- and soft-decision channels require different metrics, the concept of choosing the codeword that is closest to the received sequence is the same in both cases.

V. RESULTS AND CONCLUSION

Timing measurements are used to compare the likely energy consumption of the two decoders. To a first degree of approximation, it is expected that energy consumption will be proportional to the execution time. Using a data length of 10000 bits, the decoders were run using the MATLAB® Profiler tool. Data for the profiler was collected for single packets, each containing 10,000 bits, when bit-errors result from constant AWGN channel noise. Simulations were run for Eb/No varying from 1 to 13 dB. For each run the value of Eb/No remained constant. The results as shown in Table 1 and Fig. 2 to 7 below indicates the corresponding time required by each decoder and hence the power requirement.

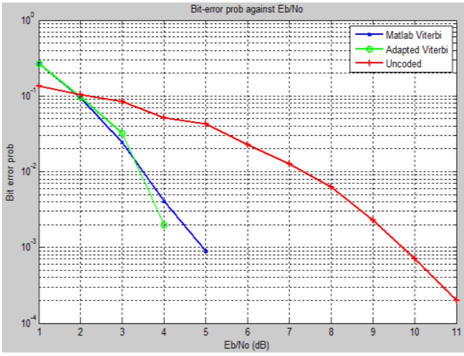

Fig. 2. Plot Of Total Execution Time For Normal Viterbi

Fig. 2 above shows the plot of bit error probability for the proposed error dependent viterbi decoder, conventional viterbi decoder and uncoded message. It could be seen from the plot that the proposed method did not introduce extra errors in the decoded message. Thus, it could be infered that there is no extra error introduced to the decoded message as a result of using the proposed method.

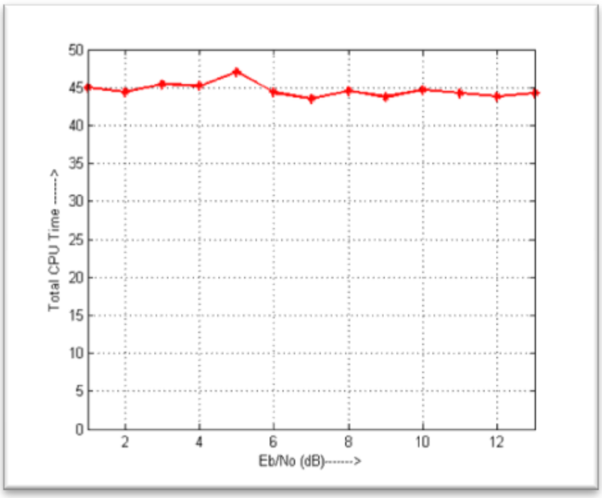

Fig. 3. Plot Of Total Execution Time For Conventional Viterbi Decoder

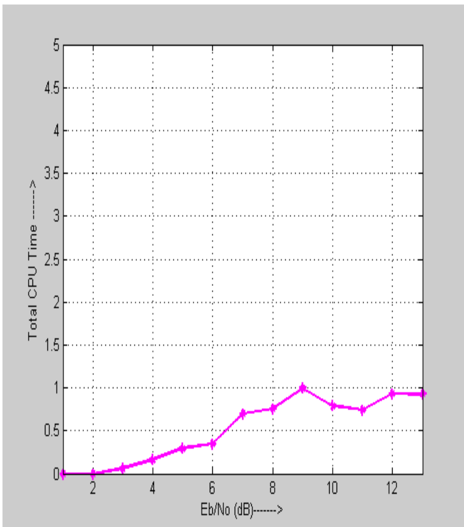

Fig. 4. Plot Of Total Execution Time For Simple Decoder

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 2, August 2012

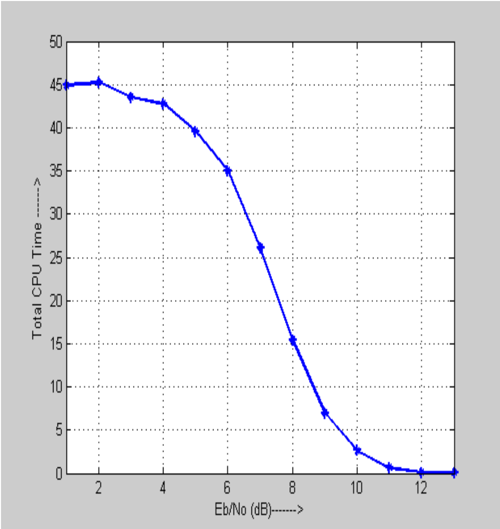

Fig. 5. Plot Of Total Execution Time For Modified Viterbi

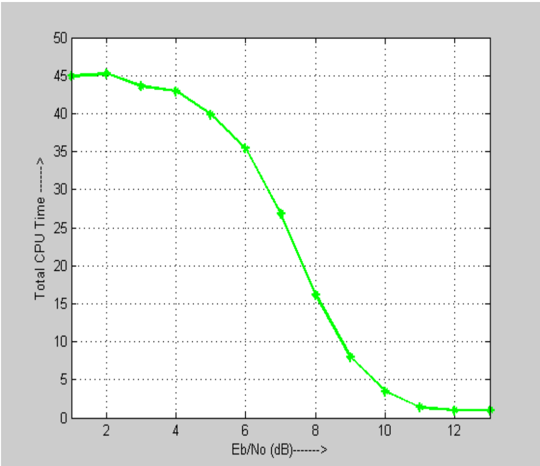

Fig. 6. Plot Of Total Execution Time For Proposed Switching Decoder

Table 1: Timing Measurement of the Proposed Decoder and Conventional Viterbi Decoder

Eb/ No	Time (cpu seconds)			
		Proposed Decoder		
	Viterbi Decoder	Simple Decoder	Modified Viterbi	Total
1	44.9832	0	44.9227	44.9227
2	44.4317	0	45.2464	45.2464
3	45.4324	0.0595	43.5473	43.6068
4	45.2278	0.1628	42.7586	42.9214
5	47.0208	0.2967	39.5631	39.8598
6	44.3703	0.3478	35.0368	35.3846
7	43.5426	0.7012	26.1414	26.8426
8	44.5563	0.7552	15.4371	16.1922
9	43.7472	0.9951	7.0020	7.9971
10	44.7237	0.7896	2.6514	3.4410
11	44.2401	0.7403	0.6473	1.3876
12	43.7937	0.9328	0.0865	1.0193
13	44.2680	0.9272	0.0716	0.9988

As can be seen from the Fig.s above the number of bits decoded by the modified My Viterbi decoder decreases as the EbNo increases. This decrease in the number of bits gives rise to a considerable decrease in the time required to decode the bits. The cumulative sum of the time required by the simple decoder and modified My Viterbi Decoder gives the total time required by the proposed decoder. Thus the time taken by the Proposed Decoder is dependent on Eb/No. At higher Eb/No values, where a large portion of the decoding is being done by the simple decoding part, less time is required to complete the decoding. As the Eb/N0 value decreases, a greater portion of decoding is done by the Adapted Viterbi decoding part. Therefore the time required to complete the decoding increases.

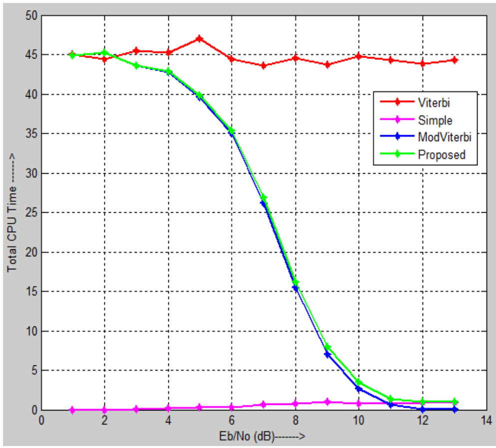

Fig. 7. Plot Of Total Execution Time For All The Decoders

From the Fig. 7 above, which shows a family of curves for the various decoders together, it is observed that when Eb/No equals 3 dB, the time requirement of the Switching Decoder is almost equal to that of the standard Viterbi decoder. Below 3 dB the time requirements for the Switching Decoder and standard Viterbi decoder remain more or less constant and equal. Thus, it could be reliably implied that for any Eb/No above 3dB, the proposed decoder will give a better energy saving without increasing the error probability of the decoder.

References

[1] Jacobs Meyer, M.J., (1996). Introduction to Error Control Coding. Pericle Communications Company. [Online]. Available at: <http://www.pericle.com/papers/Error_Control_Tutorial.pdf> [Accessed 10 June 2011].

[2] Jin, J., Chi-Ying Tsui., (2006). A low power Viterbi decoder implementation using scarce state transition and path pruning scheme for high throughput wireless applications. Proceedings of the 2006 international symposium on Low power electronics and design, Germany, 4-6 Oct. 2006, pp. 406 – 411.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 2, August 2012
[3] Shaker, S.W., (2009). Design and Implementation of Low-Power Viterbi Decoder for Software-Defined WiMAX Receiver. 17th Telecommunications forum TELFOR, Belgrade, 24-26 Nov 2009, pp. 468-471.

[4] Kothari C. R. (2004). Research Methodology: Methods and Techniques, New Age International (P) Ltd., New Delhi.

[5] Anjali, K. S., (2010). Energy saving Viterbi Decoder for Forward Error Correction in Mobile Networks. M.Sc Thesis, University of Manchester.

[6] Michelle, A. (2002). Steps in Empirical Research, [Online].at :< http://www.csulb.edu/~msaintg/ppa696/696steps.htm> [Accessed 11 July 2011].

[7] Clark, G. C. Jr. and Cain. J. B., (1981). Error-Correction Coding for Digital Communications. New York: Plenum Press.

[8] Sklar, B., (2001). Digital Communications – Fundamentals and Applications. 2nd ed. New Jersey: Prentice Hall.

[9] Basilead Library. (2006), what is Empirical Research. Tutorials and Research Guides. [Online]. at:<http://library.manor.edu/tutorial/empiricalresearch.htm> [Accessed 11 July 2011].

[10] Seki K.; Kubota S.; Mizoguchi M. and Kato S., (1994) "Very low power consumption Viterbi decoder LSIC employing the SST (scarce state transition) scheme for multimedia mobile communications," Electronics-Letters, IEE, Vol.30, no.8, April p.637-639.

[11] Lang L.; Tsui C.Y. and Cheng R.S., (1997) "Low power soft output Viterbi decoder scheme for turbo code decoding," Conference-Paper, ISCAS '97(Cat. No97CH35987). IEEE, New York, NY, USA, 4 vol. Lxvi+2832 pp. 1369-1372.

[12] Oh D. and Hwang S., (1996) "Design of a Viterbi decoder with low power using minimum-transition trace back scheme," Electronic-Letters, IEE, Vol.32, No.22, pp. 2198-2199.

[13] Viterbi, A. J., (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Information Theory., vol. 13 no.2, pp.260-269.

[14] Garrett D. and Stan M., (1998) "Low power architecture of the soft-output Viterbi algorithm," Electronic-Letters, Proceeding 98 for ISLEPD '98, p 262-267.

[15] Fleming, C., (2002). Tutorial on Convolutional Coding with Viterbi Decoding. Spectrum Applications.[Online].at:<http://home.netcom.com/~chip.f/viterbi/algrthms2.html> [Accessed 8 June 2011].

[16] Shao, W., (2007). Low Power Viterbi Decoder Designs. PhD Thesis, University of Manchester.

[17] O. J. Joeressen and H. Meyr, (1995) "A 40MBits/s Soft Output Viterbi Algorithm" IEE Journal of Solid State Circuits, vol. 30 No. 7.

[18] G. Ouyang (2010) "A High Throughput Low Power Soft-Output Viterbi Decoder" MSc Thesis, School of Information Science and Technology Fudan University, Shanghai, China.

[19] C. Berrou, P. Adde, E. Angui and S. Faudeil (1993) "A Low Complexity Soft-Output Viterbi Decoder Architecture," IEEE Int. Conf. Communications, vol. 2, pp. 737–740.