

PERFORMANCE ANALYSIS OF SOFT-DECISION AND HARD-DECISION DECODING FOR ERROR DEPENDENT POWER SAVING VITERBI DECODER FOR MOBILE DEVICES

Offor Kennedy John

Department of Electrical/Electronic Engineering, Anambra State University, Uli

ABSTRACT

This paper investigates the Viterbi Algorithm for decoding convolutionally coded messages. A new energy saving strategy that may enable receivers to decode convolutionally coded transmissions with lower energy utilization using soft decision decoding techniques and hard decision decoding techniques is analyzed. 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®. When compared to the same implementation using hard decision decoding, the soft decision decoding gives a better performance of about 2dB.gives.

KEYWORDS

Convolution codes, error dependent, Viterbi algorithm, power saving

1. INTRODUCTION

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. These electronic devices are usually carried around and use rechargeable battery as a source of energy. Sometimes the energy stored in the battery between recharges is barely enough to last for the whole day before next recharge is made. In digital communications, the information or message to be transmitted from the sender to the receiver undergoes many processes before it is received error free or near error-free. The Figure 1 below shows a block diagram of a digital communications system.

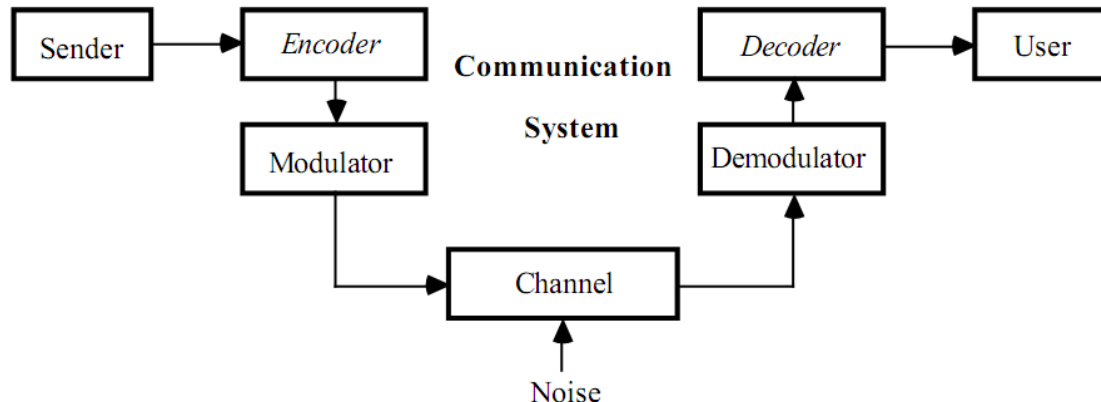


Figure 1. The Digital Communications System [1]

In the communication system shown above, the message that is sent from the sender to the receiver undergoes various processes as shown in the named blocks.

The **channel** is the medium through which the message is sent from the **sender** to the **receiver**. The channel could be either wired (twisted pair cable, coaxial cable, fibre optic cable), wireless (radio waves), storage media (like magnetic tape, optical disc), or any other noisy medium. This part of the communication system introduces errors.

The **modulator** transforms the output of the encoder, which is digital, into a format suitable for the channel, which is usually analogue (e.g., a telephone channel). The **demodulator** attempts to recover the correct channel symbol in the presence of noise. When the wrong symbol is selected, the decoder tries to correct any errors that result.

Because of the error introduced by the channel, an encoder and decoder is often used to detect and correct such errors. The **encoder** adds redundant bits to the sender's bit stream to create a codeword. The **decoder** uses the redundant bits to detect and/or correct as many bit errors as the particular error-control code will allow. The most popular algorithm used in encoding and decoding messages to be transmitted are respectively the **convolutional coding** and the **Viterbi Algorithm** which was proposed by Andrew J. Viterbi in 1967 [13]. It is a maximum likelihood (ML) decoding algorithm in the sense that it finds the closest coded sequence U to the received sequence Z by processing the sequences on an information bit-by-bit basis. The Viterbi Algorithm consumes considerable energy because of its complexity irrespective of whether the received signal contains errors or not. Various researchers have proposed many ways of minimizing the energy used in the decoding [5, 17, 19, 32, 33, 34, 35,]. This research compares the performance of soft decision decoding with hard decision decoding in saving energy especially in mobile devices.

2. METHODOLOGY

This research employs the quantitative and applied research approach. Applied research aims at finding a solution for an immediate problem facing a society or an industrial/business organization while quantitative research is based on the measurement of quantity or amount [22]. A quantitative research approach is employed in this research to ensure that reliable results are obtained and that all aspects of the problem to be solved are addressed. A quantitative research approach is sub-

divided into inferential, experimental and simulation approaches. The purpose of inferential approach to research is to form a data base from which to infer characteristics or relationships of population. Experimental approach is characterized by much greater control over the research environment and in this case some variables are manipulated to observe their effect on other variables. Simulation approach involves the construction of an artificial environment within which relevant information and data can be generated.

Because 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, and also investigate whether soft decision decoding will give a better performance than hard decision decoding in terms of dB gains as stated by [15], the simulation/empirical approach is employed [22, 23, and 24]. Some of the main objectives of this approach are to learn from collective experience of the field and to identify, explore, confirm and advance theoretical concepts. An emphasis will be laid on utilizing the appropriate test cases, data collection and analysis techniques. To evaluate and analyse the energy saving capability of the new algorithm using bit-error performance and execution time as the basis.

3. DESIGN METHODS FOR THE VITERBI ALGORITHM

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.

3.1: Encoding mechanism

Data is coded by using a convolutional encoder, which consists of a series of shift registers and an associated combinatorial logic. The combinatorial logic is usually a series of exclusive-or gates. The conventional encoder rate $\frac{1}{2}$ K=7, (171,133) is used for the purpose of this project. The octal numbers 171 and 133 when represented in binary form (1111001, 1011011) correspond to the connection of the shift registers to the lower and upper exclusive-or gates respectively. Figure. 2 represent this convolutional encoder that was used for the project.

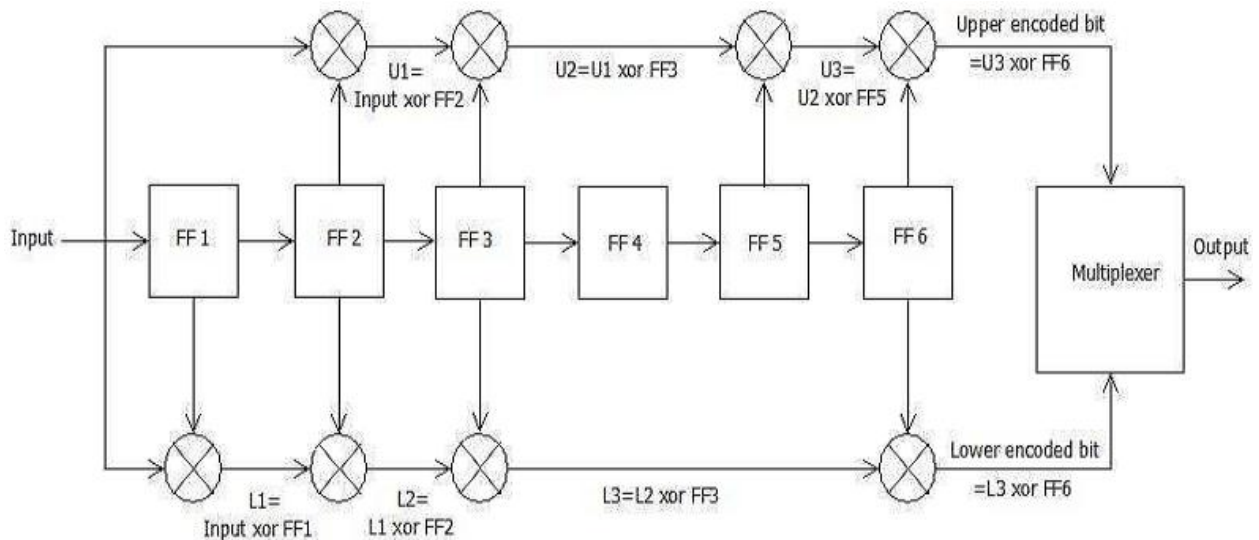


Figure 2. Rate = $\frac{1}{2}$ K = 7, (171,133) Convolutional Encoder

3.2: Decoding mechanism

The classical Viterbi decoder design is a straightforward implementation of the basic processes of the Viterbi algorithm. The design consists of three functional units, as shown in Figure 3.

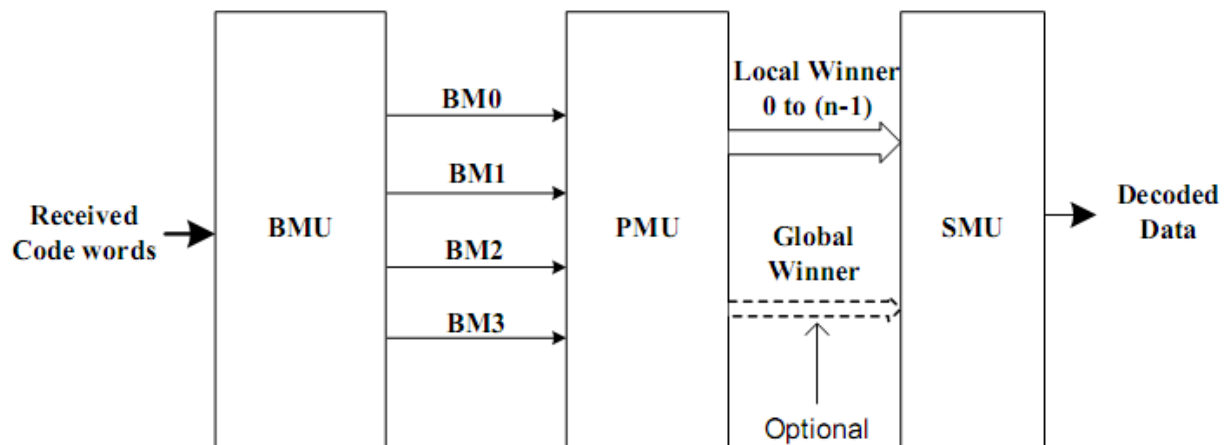


Figure 3. Classical three functional block of a rate 1/2 Viterbi decoder design.

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 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.

3.2.1: 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 signaling is used with a convolutional code of rate m/n . If a code word S , which consists of n symbols, $x_0 x_1 \dots x_{n-1}$, is transmitted through the channel, the decoder receives R which is a sequence of n sampled actual voltages, $r_0 r_1 \dots r_{n-1}$, from the filter. The conditional probability of sending S and receiving R is [7]

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

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 $\sigma^2 = N_0/2$ where N_0 is the noise power spectral density, $P(R|S)$ becomes the product of n Gaussian density functions of each symbol. As given in [7]

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

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

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

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.

For the received signal from the additive white Gaussian noise (AWGN) channel, the signal level of each symbol is independent. Thus, a code word which consists of n symbols forms an n -dimensional space. For instance, Figure. 4 shows the distances of the code word with 2 symbols, X and Y . There are four ideal code words, $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$, which are located at the four corners in this 2-dimensional space. The received signal (x, y) are unquantised and represent the two received code word symbols having the value range from 0 to 1. Due to the noise the received signals do not correspond to any of the ideal points. Thus, the distance labelled with d_{00} , d_{01} , d_{10} , and d_{11} are the Euclidian distances; d , between (x, y) and the four ideal points, where, for example, $d_{00}^2 = x^2 + y^2$. In the 3-dimensional space formed by 3-symbol code words, as shown in Figure. 5 there are 8 Euclidian distances, d_{000} , d_{001} , d_{010} , d_{011} , d_{100} , d_{101} , d_{110} , and d_{111} , for the received signals (x, y, z) and the distance d_{000} becomes $d_{000}^2 = x^2 + y^2 + z^2$ [16].

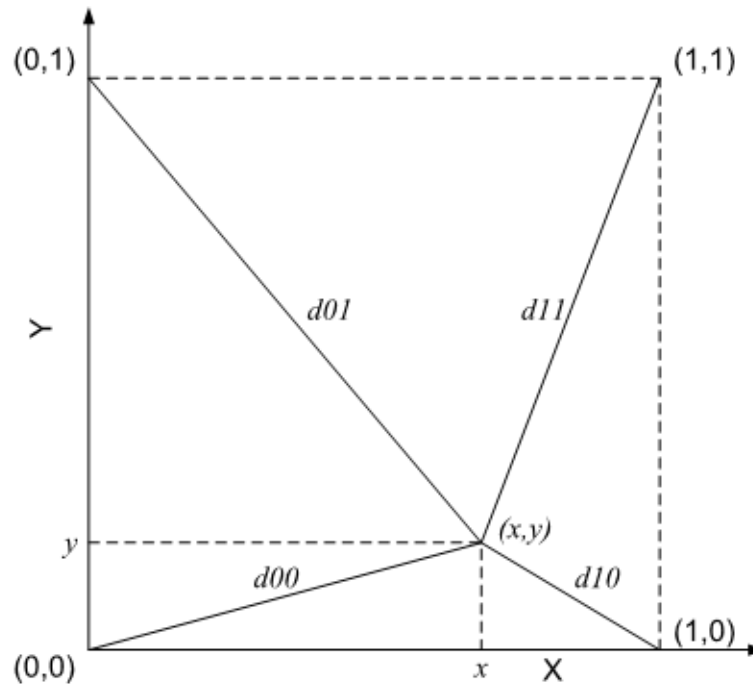


Figure 4. The Euclidian distance 2-symbol codeword represented in 2-Dimension.

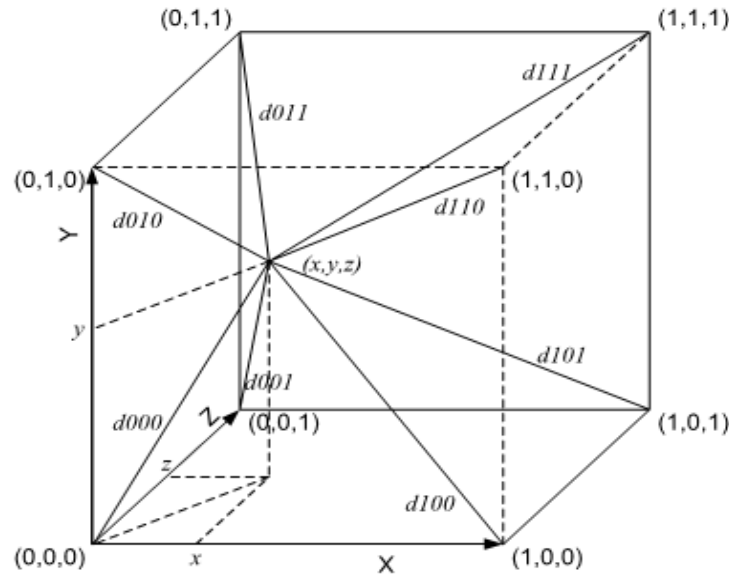


Figure 5. The Euclidian distance 3-symbol codeword represented in 3-Dimension.

In digital communication systems, it is not possible to process the actual analogue voltages r_i ; instead, the sampled voltages are quantised into m -bit numbers. In hard-decision, a signal is quantised into a one-bit binary number. In receiving the code word 00, for instance, the $d_{11}^2 = (1 - 0)^2 + (1 - 0)^2 = 1 + 1 = 2$ and is consistent with the Hamming distance described above. Other than single bit, three-bit quantization is the most commonly used scheme in communication system designs. Figure 6 illustrates the three-bit quantization for the 2-symbol code words. As shown in Figure 6., the dimensional space is partitioned into $2^3 \times 2^3 = 64$ regions with the ideal code words 00, 01, 10, and 11, located at positions (0, 0), (0, 7), (7, 0), and (7, 7), respectively.

The actual voltage within each region is approximated to the point, marked in black dots, with the smallest X and Y values. For example, the received signals within the shaded region in Figure 6 are approximated to the point (x, y) . Then the squared Euclidian distances for (x, y) can be used as the approximated distance of the received signal. Although a squared Euclidian distance is a simple addition of numbers, it still involves squares of the quantised signal values and causes engineering difficulties. Fortunately, the squared Euclidian distances of (x, y) can be linearly transformed into the Manhattan distance [7]

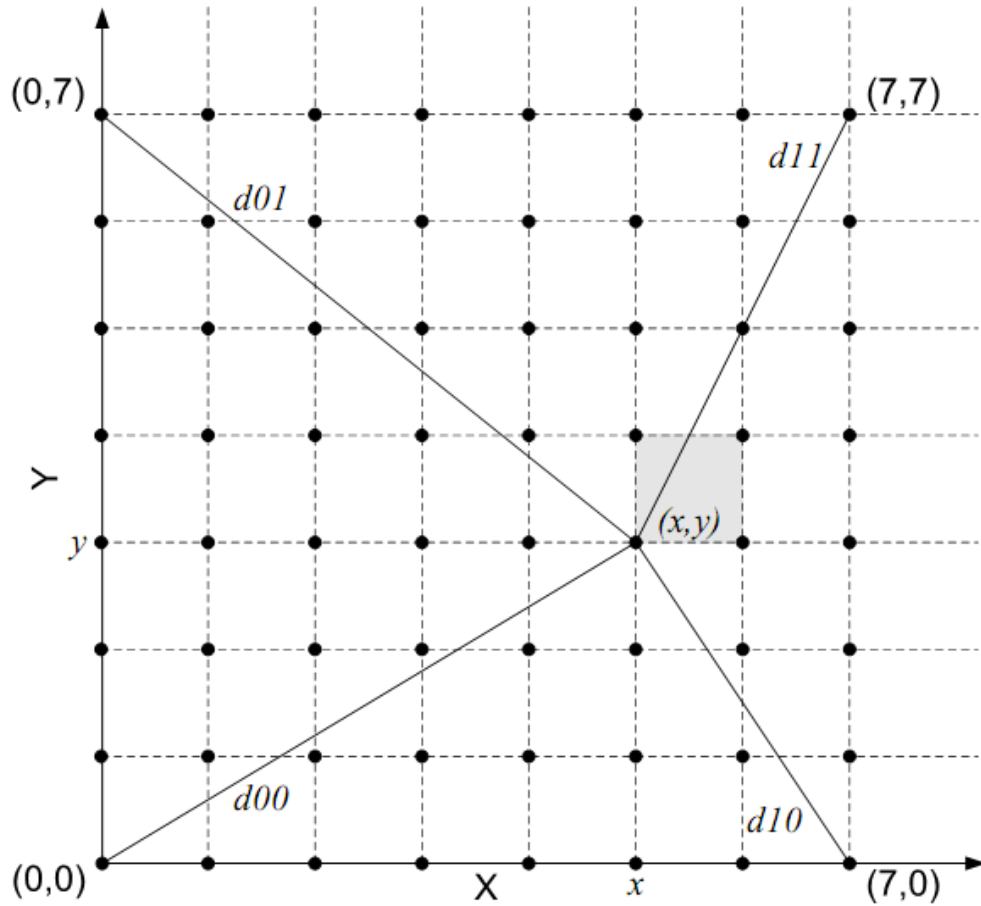


Figure 6. Three-bit quantization in a 2-dimensional space.

$$d00_{Mah} = x + y \quad \text{-----} 4$$

$$d01_{Mah} = x + [(2^m - 1) - y] \quad \text{-----} 5$$

$$d10_{Mah} = [(2^m - 1) - x] + y \quad \text{-----} 6$$

$$d11_{Mah} = [(2^m - 1) - x] + [(2^m - 1) - y] \quad \text{-----} 7$$

by subtracting $(x^2 + y^2)$, dividing by $2^m - 1$ and then adding $x + y$, where m is the number of bits in the quantization. Since the Viterbi algorithm is a linear process, using the Manhattan distance yields no accuracy degradation compare to the squared Euclidian distance, but simplifies the implementations. This can also be generalized to the squared Euclidian distance of any n-symbol

code word, so that the Manhattan distance is used as the branch metric for a received code word. Since the Manhattan distance is the addition of the distance in n independent directions, it can be further normalized in each direction. For the 2-symbol example shown in Figure 6, all the Manhattan distances of the point (x, y) , listed in equations 4 – 7, can be simplified in the form $d_{Mah} = d_x + d_y$, where the d_x and d_y are distances on the X and Y axis, respectively. Assuming $x \leq (2^{m-1} - 1)$ d_x on the axis X can be subtracted by x , then the normalized distance on axis X between the symbol X and ideal symbol 0 is always zero; whereas the normalized distance to ideal symbol 1 is $(2^m - 1) - 2x$.

Similarly distances on Y axis can be normalized to be either 0 or $(2^m - 1) - 2y$ when $y \leq (2^{m-1} - 1)$. Based on this, the branch weight scheme for the 2-symbol, 3-bit quantization can be simplified as shown in Table 3.1. Therefore, a standard BMU design assigns the weights to each symbol based on its quantized level and the weight scheme and adds the weights of each symbol together to make the branch metric.

Table 1. Branch weight scheme for 2-symbol, 3-bit quantization.

| Quantised Level | Weight referenced to 0 | Weight referenced to 1 |
|-----------------|------------------------|------------------------|
| 0 (strongest 0) | 0 | 7 |
| 1 | 0 | 5 |
| 2 | 0 | 3 |
| 3 | 0 | 1 |
| 4 | 1 | 0 |
| 5 | 3 | 0 |
| 6 | 5 | 0 |
| 7 (strongest 1) | 7 | 0 |

4. 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.

If the decoder operates on the hard decisions made by the demodulator, the decoding is called hard-decision decoding. The Binary Symmetric Channel (BSC, where symmetric means the conditional probability $P(j|i)$ is symmetric) is an example of a hard-decision channel, which means that, even though continuous-valued signals may be received by the demodulator, the demodulator can still output firm decisions which consist of the one of two binary values. For a 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:

$$P(0|1)=P(1|0)=p \text{ and } P(1|1)=P(0|0)=1-p \quad 8$$

Taking equation (3.8) into consideration, the likelihood function can be expressed as:

$$P(Z|U^{(m)}) = p^{d_m}(1-p)^{L-d_m} \quad 9$$

And the log-likelihood function is:

$$\log P(Z|U^{(m)}) = -d_m \log\left(\frac{1-p}{p}\right) + L \log(1-p) \quad 10$$

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

$$\log P(Z|U^{(m)}) = -d_m A + LB \quad 11$$

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 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

$$P(Z|U^{(m)}) = \sum_{i=1}^{\infty} \sum_{j=1}^n Z_{ji} U_{ji}^m \quad 12$$

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.

When the demodulator sends eight-level quantized soft decision to the decoder, it sends a 3-bit word. In fact, sending 3-bit decision instead of a hard decision is equivalent to sending the decoder a measure of confidence. If the demodulator sends 111 to the decoder, it declares the code symbol to be a one with very high confidence. If the demodulator sends 100 to the decoder, it declares the code symbol to be a one with very low confidence. In contrast, if the demodulator sends 000 to the decoder, it declares the code symbol to be a zero with very high confidence. And if the demodulator sends 011 to the decoder, it declares the code symbol to be a zero with very low confidence. Sending the decoder soft decisions instead of hard decision can provide the decoder with more information, which the decoder uses for recovering the message sequence. The eight-level soft-decision is often shown as $-7, -5, -3, -1, 1, 3, 5, 7$ [8]. Such a designation lends itself to a simple interpretation of the soft-decision: the sign of the metric represent a decision and the magnitude of the metric represent the confidence level of the decision. For a Gaussian channel, eight-level quantization will improve the decoding performance by approximately 2dB in required signal-to-noise ratio E_b/N_0 compared to two level quantization.

4. RESULTS AND CONCLUSION

As described earlier, 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 E_b/N_0 varying from 1 to 13 dB. For each run the value of E_b/N_0 remained constant. The results as shown in Table 2 and Figures 7 to 11 below indicates the corresponding time required by each decoder and hence the power requirement.

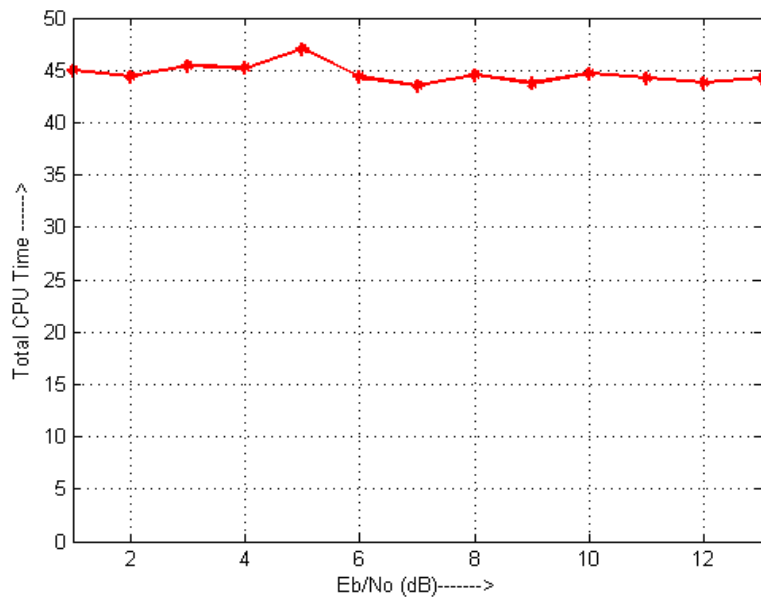


Figure 7. Plot of Total execution time for Normal Viterbi

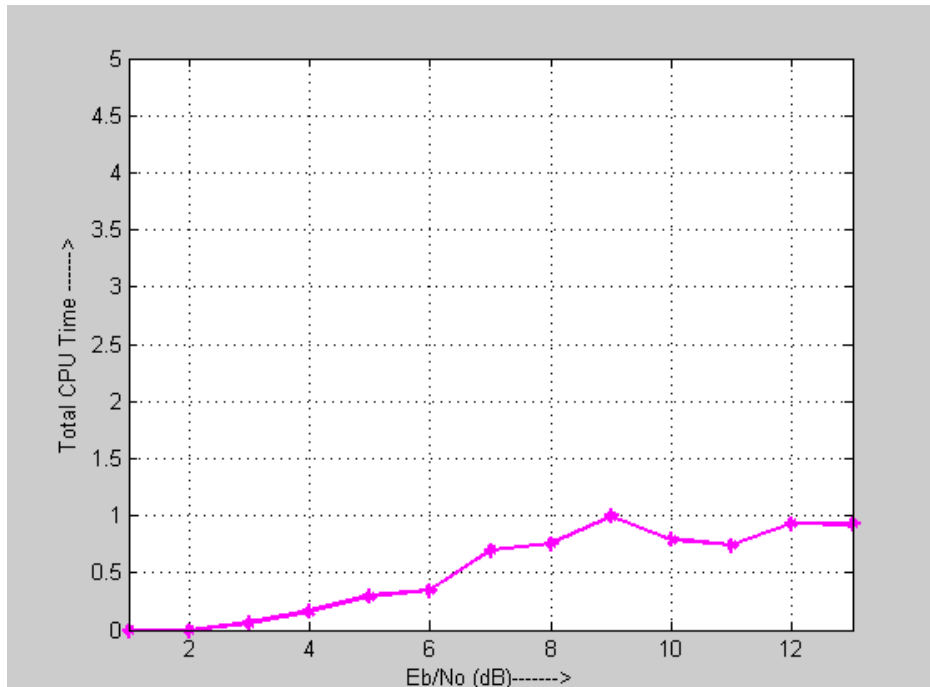


Figure 8. Plot of Total execution time for Simple Decoder

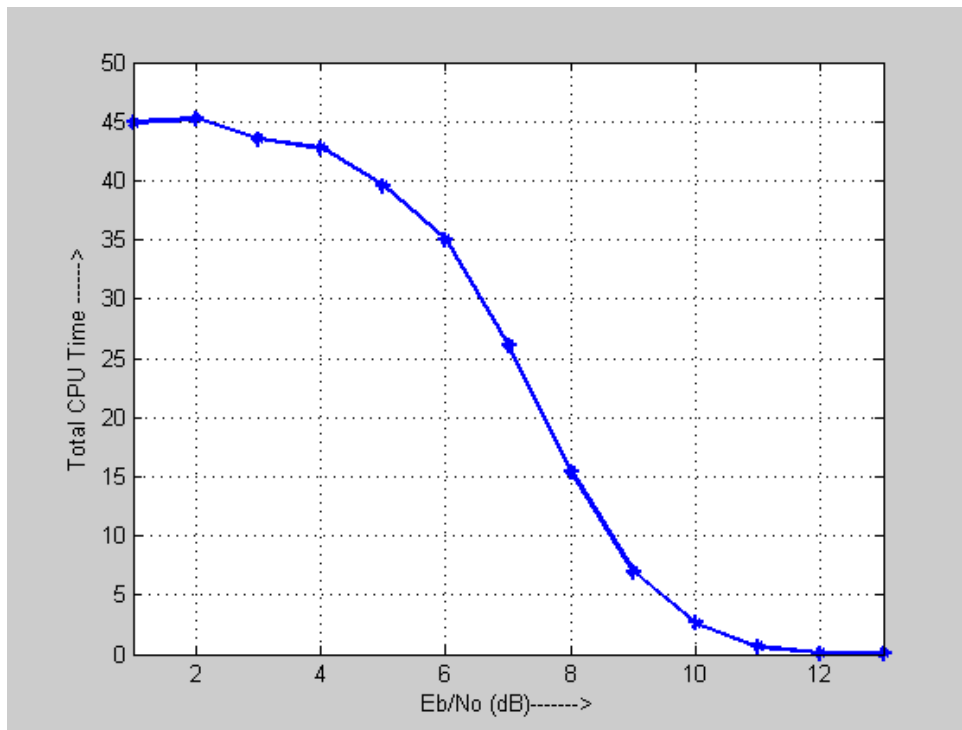


Figure 9. Plot of Total execution time for Modified Viterbi

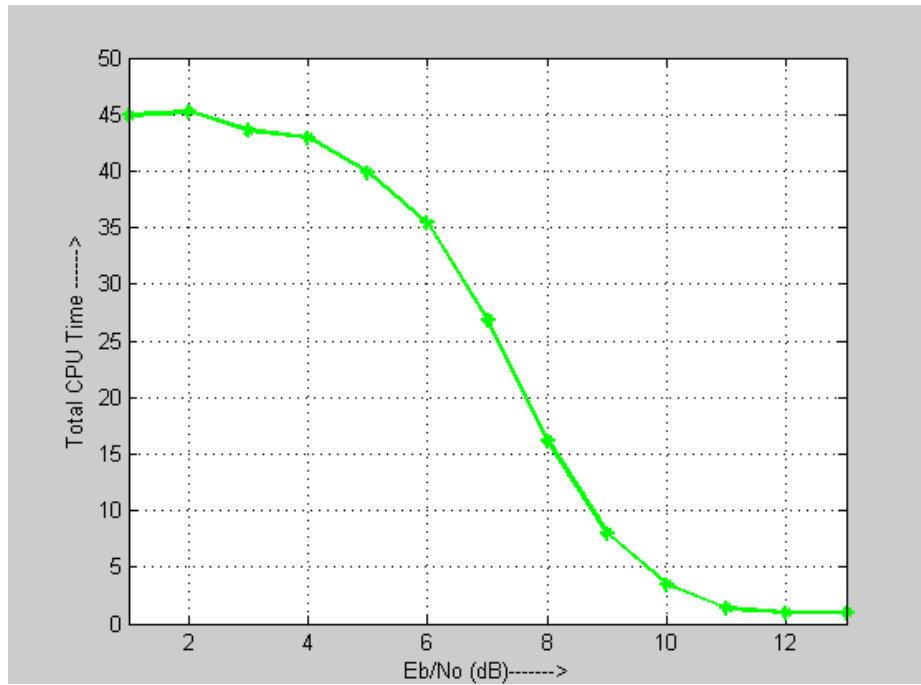


Figure 10. Plot of Total execution time for Proposed Switching Decoder

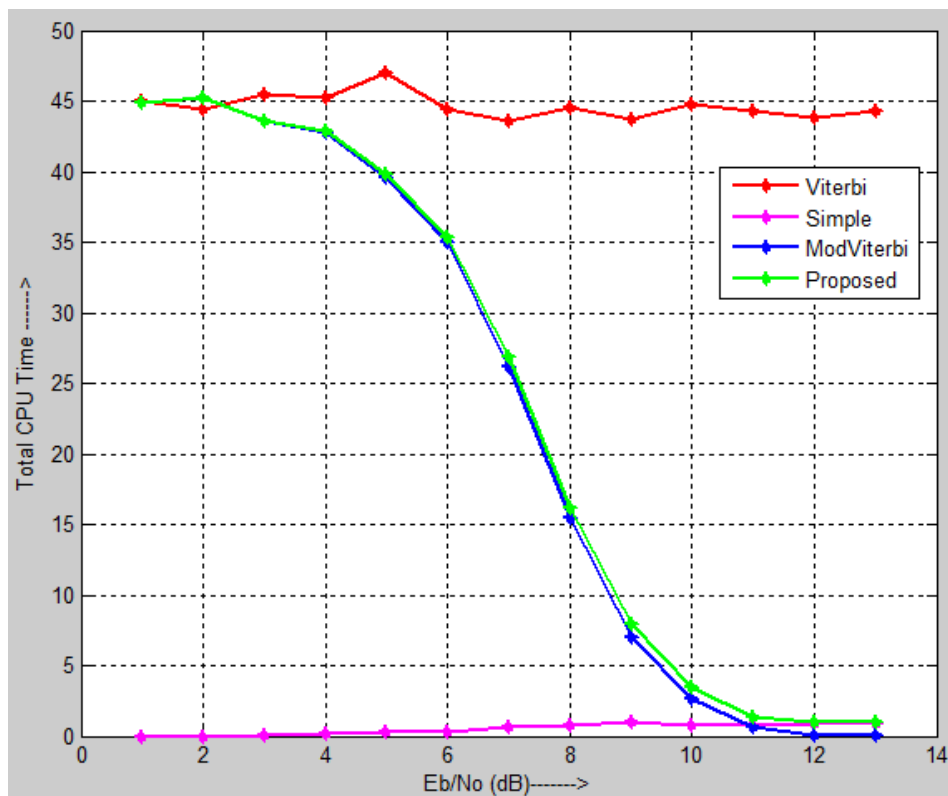


Figure 11. Plot of Total execution time for all the decoders

Table 4.3: Timing Measurement of the Proposed Decoder and Conventional Viterbi Decoder

| EbNo | Time (cpu seconds) | | | |
|------|--------------------|------------------|---------------------|---------|
| | My Viterbi Decoder | Proposed Decoder | | |
| | | Simple Decoder | Modified My 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 |

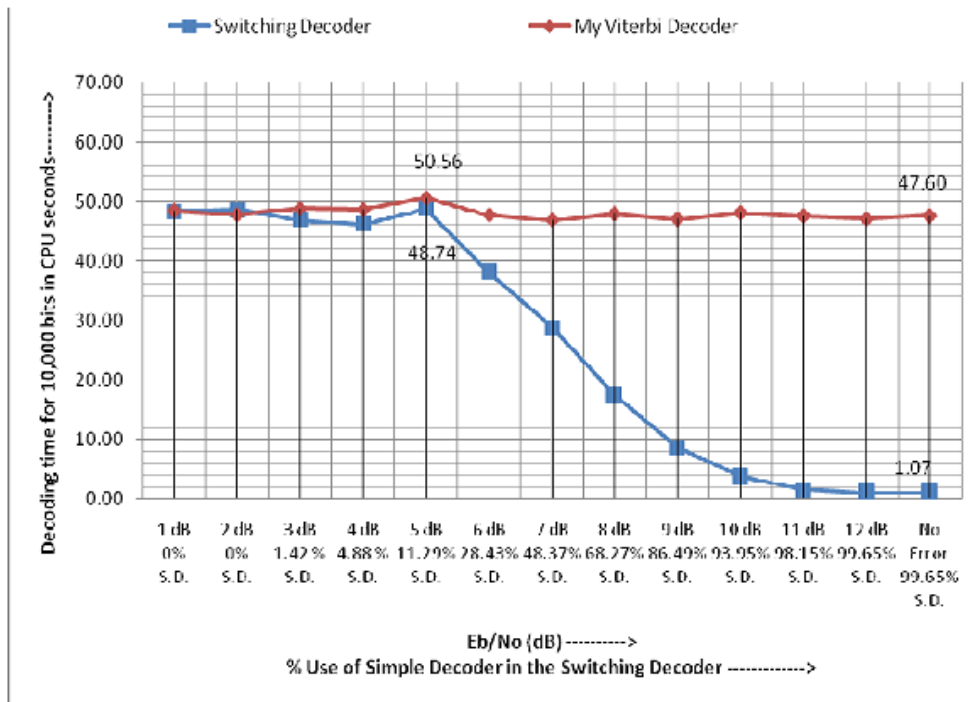


Figure 12 Timing measurements for a hard-decision decoding [5]

As can be seen from the figures above the number of bits decoded by the modified My Viterbi decoder decreases as the E_b/N_0 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 E_b/N_0 . At higher E_b/N_0 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 E_b/N_0 value decreases, a greater portion of decoding is done by the Adapted Viterbi decoding part. Therefore the time required to complete the decoding increases. It is observed that when E_b/N_0 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. When compared to the result of the same algorithm using hard decision decoding (as seen in figure 12, it was observed that the power saving started at about 5dB as against 2.3dB (for the soft decision implementation (as seen in figure 11). This finding goes to demonstrate the assertion that soft decision decoding gives a better performance of about 2dB to 3dB over the hard decision decoding [15].

REFERENCES

- [1] Jacobsmeyer, M.J., (1996). Introduction to Error Control Coding. Pericle Communciations 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.
- [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, PPA 696 Research Methods, California State University. [Online]. Available 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]. Manor College. Available 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 traceback 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]. Available 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.

Author

Offor Kennedy John received his B.Eng. in 2002 from Nnamdi Azikiwe University, Awka Anambra State Nigeria. He is a registered engineer with Council for the Regulation of Engineering in Nigeria (COREN) and a member of The Nigerian Society of Engineers (NSE). He is currently a senior Engineer and an MSc. Student in Anambra State University, Nigeria. His research interest in the field of Software engineering, Error Control Coding, FPGA and Power systems/efficiency.

