2014-09

Assessment of non transmittable codewords enhancement to Viterbi Algorithm Decoding

Mrutu, Salehe

LJS Publishing

Assessment of Non Transmittable Codewords Enhancement to Viterbi Algorithm Decoding

Salehe I. Mrutu[1], Anael Sam[2] and Nerey H. Mvungi[3]

[1, 2] *School of Computational and Communication Science and Engineering (CoCSE), Nelson Mandela African Institution of Science and Technology (NM-AIST), Arusha, Tanzania*

[3] *College of Information and Communication Technologies, University of Dar Es Salaam (UDSM), Dar Es Salaam, Tanzania*

Abstract—**Researchers have shown that practical mobile communication channels introduce errors that are concentrated in a certain locality rather than random errors. These are burst errors caused by deep fading of the wireless channel or a lightning strike. The existing Viterbi Algorithm (VA) capable of correcting random errors is inefficient in correcting burst errors and therefore resulting in unacceptable amount of residual errors. This paper presents an assessment of Non-Transmittable Codewords (NTCs) enhancement technique to VA in decoding the received signals subjected to burst errors that may occur in poor channels. A hard decision, 1/2 rate and constraint length K is equal to 3 Viterbi Algorithm decoding technique, Binary Phase-Shift Keying (BPSK) and Additional White Gaussian Noise (AWGN) are components used in MATLAB software based simulation when assessing the proposed technique. Applying 6NTCs to VA decoder enables the decoder to reduce 83.7 percent of its residual errors. However, the technique reduces the encoder's data transmission rate from 1/2 to 1/6.**

***Keywords**-Locked Convolutional encoder; Bust errors; Residual errors; Non-Transmittable Codewords (NTCs); Viterbi Algorithm Decoding; Data Interleaver*

I. Introduction

A pair of binary convolutional encoder and Viterbi decoder is one of the mostly used components in digital communication to achieve low error rate data transmission. Convolution codes are popular Forward Error Correction (FEC) codes in use today. This type of code was first introduced by Elias in 1955 [1], [2] . VA introduced in 1967 [3], is known to be the maximum likelihood decoding algorithm of Convolutional codewords transmitted over unreliable channel [4]. VA is efficient in decoding random errors that occurred in a channel. However, the occurrence of burst errors in a received data block results in uncorrected or residual errors. Practical mobile communication channels are sometimes affected by errors which are concentrated in a certain locality rather than random errors [5]. These burst errors occur due to deep fading of the wireless channel or a strike of lightning in case of poor weather conditions or intensive interference with other radio communication systems in the environments [6].

VA decoder can increase its error correction capability by increasing its constraints length (its memory size) [7]. However, increasing the memory size beyond 10 leads the decoder into prohibitive delay(not preferred by most real time applications) due to exponential growth of its decoding computation complexity [8], [2]. For decades, VA had been dealing with burst errors using an interleaving utility support. Without an interleaver, burst errors drive a viterbi decoder's decision unit into a confusion state which leads the decoder into failure and thus resulting in residual errors [9].

The basic idea behind the application of interleaved codes is to shuffle the received data. This action leads to randomization of the received burst errors that are closely located and then apply the VA decoder. Thus, the main function done by interleaver at transmitter is to change the input symbol sequence. At the receiver, de-interleaver changes the received sequence to get back the original sequence as the one at transmitter. There are two main categories of Interleaver utilities in communication systems that are block and convolutional interleavers.

Block interleaver just writes the received data row by row in a matrix and read them out for transmission column by column. Fig. 1 demonstrates how the block interleaver and de-interleaver work to jumble the received data and disperse burst errors. If a sufficient interleaver depth (number of rows in the interleaver/de-interleaver) is applied, then Interleaver successfully removes the effects of burst errors and turns them into controllable pattern of random errors by VA decoder.

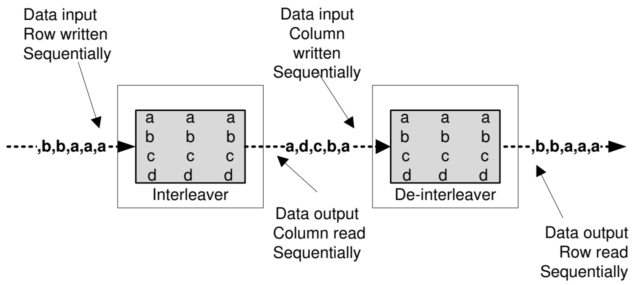

Fig. 1. Block Interleaver-De-interleaver

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

As it is clear from fig. 1, the columns are read sequentially from the interleaver. The receiver can only interpret a complete row when all the involved columns in the interleaver depth have arrived and not before that. In addition, receiver requires a considerable amount of memory in order to store the received symbols until all the involved rows in the interleaver depth have arrived. These facts raise two basic drawbacks to the technique, one is latency and another is the storage (large amount of memory). The mentioned drawbacks are of great concern and challenge to delay sensitive real time applications [10], [11].

The recently introduced convolutional interleaver [12] is reported to have reduced large part of the mentioned drawbacks. However, it is important to note that the application of interleaver is necessary only when the involved code fails to deliver the required quality of error correction.

This paper assesses a technique of using Non Transmittable Codewords (NTCs) [9] to support VA decoder in decoding burst errors. The rest of this paper is organized in the following manner: Section II briefly discusses the encoding and decoding process using 1/2 rate and constraint length K=3 binary convolutional encoder and the viterbi decoder. Section III discuses the assessed technique that enhance VA decoder in reducing number of residual errors when it receives burst errors for decoding. Section IV describes the model used in building assessment simulation in MATLAB software. Section V discusses the results obtained from the simulation and section VI is a conclusion to these efforts.

II. Encoding and Decoding Processes

Binary Convolutional Coding is a technique that adds binary redundancy bits to original bit or bits sequence to increase the reliability of data communication. In this part of the paper, is the discussion of a simple binary convolutional coding scheme at the transmitter and its associated VA (maximum likelihood) decoding scheme at the receiver.

A. Encoding

There are various binary convolutional coding schemes having a designing data rate of 1/2. Fig. 2 shows the architecture of binary convolutional encoder with designing data rate 1/2, constraint length K=3 and generator polynomial is $[7, 8]_8$ which is equivalent to $[111, 101]_2$.

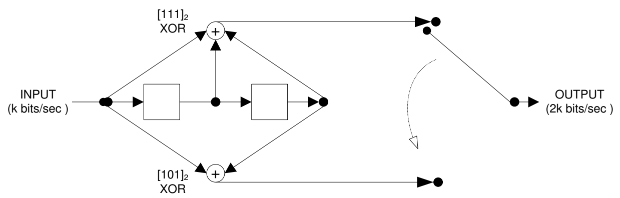

Fig. 2. Binary Convolutional code with Rate 1/2, K=3, Generator Polynomial $[111, 101]_2$

Designing data rate, constraint length and generator matrix specifies the convolutional encoder. In this regards the consideration is on the simplest encoder that has the following features:

- Rate: Ratio of the number of input bits to the number of output bits. In this encoder, rate is 1/2 that means there are two output bits for each input bit.
- Constraint length: The number of delay elements in the convolutional coding. In this encoder, K=3 where there are two delay elements (in memory) plus a single input bit.
- Generator polynomials: These refer to the wiring of the input sequence with the delay elements to form the output. In this example, generator polynomials are $[111, 101]_2$. The output from the $[111]_2$ arm uses the exclusive OR (XOR) of the current input, previous input and the previous to the previous input. The output from the $[101]_2$ uses the XOR of the current input and the previous to the previous input only.

Table 1 demonstrates the encoding process by showing the relation between the input and output bits and their corresponding encoder state transitions. The binary convolution encoding process usually starts when the encoder is at all zero state (i.e. 00). Suppose we want to encode the following stream of bits {1-1-0-1…}, it is obvious that when we pick the first input bit (i.e. 1) and the initial encoder state is all zeros (i.e. 00) then table 1 leads us to line SN. 2 where the input data is one (i.e. 1) and current state is all zeros (i.e. 00). This line indicates that the next state to be used is one-zero (i.e. 10) and the decoders output is one-one (i.e. 11). Following the same scenario one can see that encoding {1-1-0-1…} results into {11-01-01-00…}as output codewords and the encoder goes through the following state transitions (00-10-11-10-10…).

TABLE I. Input, Output and State transition relations

*S/N	Input Data	Current State	Next State	Output Data
1	0	00	00	00
2	1	00	10	11
3	0	10	01	10
4	1	10	11	01
5	0	01	00	11
6	1	01	10	00
7	0	11	01	01
8	1	11	11	10
*SN : means a row serial number				

B. Decoding

VA decoding is a dynamic programming algorithm for finding the correct path from a number of given paths. The decoding process starts from all zeros (i.e. 00) state and opens out to other states as the time goes on. Therefore, State and trellis diagrams describe the internal operations of the Viterbi decoder. Fig. 3 is a state diagram of the decoding rate 1/2 and constraint length K=3 Viterbi decoder showing the allowed state transitions during the decoding process.

To make the explanation easy, a hard-decision symbol inputs is applied and thus Hamming Distance (HD) metric weighs the path branches. HD is a bitwise comparison

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

between a pair of received codeword from a channel and allowed codewords from the decoder at that particular time interval.

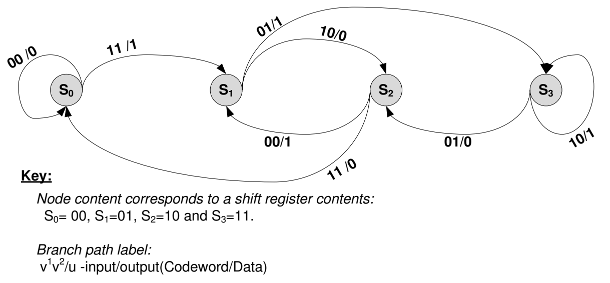

Fig. 3 Allowed state transition diagram of 1/2 decoding rate and Constraint length K=3 Viterbi Algorithm decoder

There are two ways of calculating HD [8] to find codeword's bits similarities and differences. In this paper codewords' bits similarities method is used. Therefore, similar bits are granted a value (i.e.1 HD) and non-similar bits have a zero value (i.e. 0 HD). In this case, results in each pair of comparison can be zero, one or two HDs. Fig.4 shows the calculation of HDs of each branch in each time interval and results are put in round brackets (i.e. (x)). After obtaining HD the algorithm continues as follows:

- Using a relation in (1) to recursively calculate Cumulative Branch Metrics (CBM) in each time interval *t* by adding the obtained $HD_{(t)}$ to the $CBM_{(t-1)}$ in each path and put results in a square bracket (i.e. [x]). Note that, for the time interval *t=1* there is no $CBM_{(t-1)}$, thus its value is zero.

$$CBM_{(t)} = CBM_{(t-1)} + HD_{(t)} \qquad (1)$$

- At each node, find the path having the highest CBM up to time *t* by comparing CBMs of all paths converging to that node. In this step, decisions are used to recursively update the survivor path of that node. Equation (2) shows how the survivor path is obtained.

$$PM_{(t)} = \max(CBM^{1}_{(t)}, CBM^{2}_{(t)}) \qquad (2)$$

- Eventually when the decoder terminates at time interval *t*, survivor paths leading to each node are compared to obtain a Winning Survivor Path (WSP). Equation (3) shows this relation. If more than one node has the same highest WPM then one of them is randomly selected and data are extracted from it.

$$WPM_{(t)} = \max(PM^{1}_{(t)}, PM^{2}_{(t)}, PM^{3}_{(t)}, PM^{4}_{(t)}) \qquad (3)$$

Figure 4 is a trellis diagram of 1/2 decoding rate and constraint length K=3 Viterbi decoder demonstrating the discussed steps. The same codewords obtained in subsection A of this part {i.e. 11-01-01-00…} are assumed to have been received with a bit error in the second bit of its third codeword {i.e. 11-01-00-00…}.

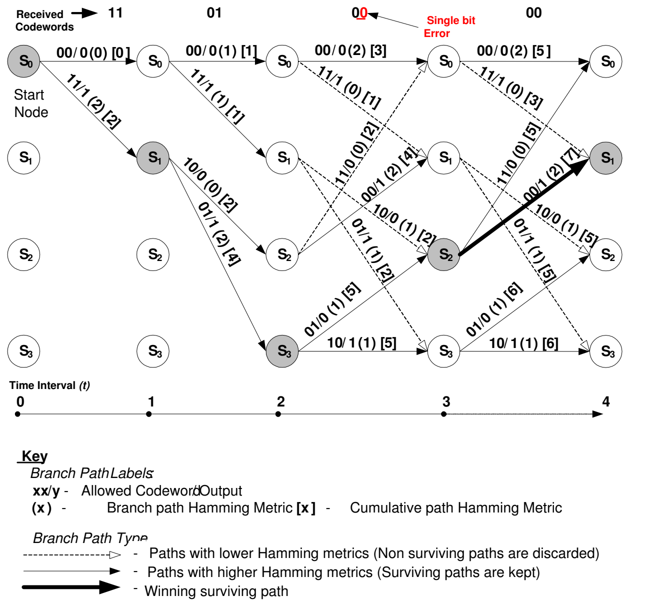

Fig. 4 Trellis diagram of 1/2 decoding rate and Constraint length K=3 Viterbi Algorithm decoder

III. NON TRANSMITTABLE CODEWORDS ENHANCEMENT

Non transmittable Codewords (NTCs) technique can be applied at the data receiving machine where data encoded by a locked convolutional encoder arrives to be decoded by a VA decoder [9]. There are two different ways of locking a 1/2 rate and constraint length K=3 convolutional encoder. The methods include either adding two zero bits (i.e. 00) to the encoder after every data bit to be encoded (for the lower end locked encoder) or adding two one bits (i.e. 11) after each data input bit (for the higher end locked encoder) [9]. All examples and simulation in this paper applies a lower end locked 1/2 rate and constraint length K=3 convolutional encoder. Fig 5 shows the locking process. Lock bits reset a decoder to all zeros state after every data input bit. Suppose we have {1, 1…} as binary data stream ready for the encoding process. Fig. 5 shows the encoder locking process where letter "D" stands for a data bit or bits, and "L" stands for the integrated lock bits.

$$\overbrace{\{1-1...\}}^{D} \xrightarrow{locking} \{\overbrace{1}^{D}-\overbrace{0}^{L}-\overbrace{0}^{L}-\overbrace{1}^{D}-\overbrace{0}^{L}-\overbrace{0}^{L}...\}$$

Fig.5. 1/2 rate and K=3 Convolutional encoder locking Process

After the encoding process, all codewords corresponding to both data and lock bits are transmitted over a noisy channel to the receiving machine. This fact lowers the data transmission rate of the encoder from 1/2 to 1/6. NTCs are known all zero codewords for the lower locked convolutional encoders that are added to the received codewords to channel to enhance the VA decoder in correcting the received errors

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

$$\{\overbrace{11}^{D}-\overbrace{10}^{L}-\overbrace{11}^{L}-\overbrace{11}^{D}-\overbrace{10}^{L}-\overbrace{11}^{L}...\}\xrightarrow{locked+2NTCs}\{\overbrace{00}^{N}-\overbrace{00}^{N}-\overbrace{11}^{D}-\overbrace{10}^{L}-\overbrace{11}^{L}-\overbrace{00}^{N}-\overbrace{00}^{N}-\overbrace{11}^{D}-\overbrace{10}^{L}-\overbrace{11}^{L}..\}$$

Fig.6. 1/2 rate and K=3 Convolutional encoder locking and 2NTCs addition process

successfully. NTCs can be added as a one, two, three codewords and so on; to each codeword corresponding to data bit. Suppose an example in fig. 5 {i.e. 1-0-0-1-0-0…} were encoded using the procedure discussed in table 1, the following codeword stream {11-10-11-11-10-11…} could be obtained for transmission. Fig. 6 demonstrates how 2NTCs are integrated to the received codewords corresponding to both data and lock bits before the received codewords are submitted for decoding. After the decoding process, all bits corresponding to the received lock codewords and the added NTCs are removed and the remaining data are submitted for use. In fig. 6 a letter "D" indicates a codeword corresponding to data bit, letter "L" is a codeword corresponding to lock bit and "N" is the added NTC.

IV. MODEL DESCRIPTION AND SIMULATION

This work, evaluates error correction capability of VA decoder and the Enhanced VA (EVA) decoder supported by NTCs. Both VA and EVA decoders, decode codewords from the same 1/2 rate and constraint length K=3 binary convolutional encoder. However, the encoder is locked using the discussed technique for the EVA decoder to enable it to utilize the technique described in section III of this paper. The number of residual errors from both the decoders forms a performance comparison factor between the two decoders. Therefore, a decoder with less residual errors is identified.

The Performance Measure of error correcting capability of the implemented codes is also given by Bit Error Rate (BER), which is obtained by the number of erroneous bits divided by the total number of transmitted bits. BER is affected by several factors including quantization technique used, noise in the channel, energy per symbol to noise ratio (*Es/No)*, code rate, and transmitter power level [13]. In their work [14], Akyildiz and colleagues showed that BER is directly proportional to the code rate and inversely proportional to energy per symbol noise ratio and transmitter power level. The use of a proper decoder that corrects errors controls the increase in BER in transmitted data. The difference in BER that can be achieved by using error correction codes to that of uncoded transmission is known as coding gain.

A MATLAB software simulation that follows the procedures described in a block diagram described in fig. 7 performs the following:

- Generation of random binary data (i.e. 0 and 1) ;
- Addition and removal of encoder lock bits and NTCs for the case of EVA;
- Encode binary data using rate 1/2, generator polynomial $[7,5]_8$ Convolutional code;
- Passing codewords through a noisy channel;
- Modulate and demodulate the codeword signals using hard decision technique for decoding process;
- Pass the received coded signals to Viterbi decoder and Enhanced Viterbi decoder;
- Counting the number of residual errors from the output of Viterbi decoder and Enhanced Viterbi decoder; and
- Repeating the same for multiple Signal-to-Nose Ratio (SNR) values

All the comparisons assume that both the algorithms have the same execution time. Table 2 shows the list of all parameters chosen for simulation.

TABLE II. SIMULATION PARAMETERS

Parameter	**Value**
Data length	10^6
Constraint Length (K)	3
Generator polynomial	$(7,5)_8$
Rate (r)	1/2
Encoder lock bits	2 zero bits (i.e. 00)
NTCs	1,2,3,4,5,6,7,8,9,10,11,12
Modulation/Demodulation	BPSK
Noise model	AWGN
Quantization	Hard Decision
Path evaluation	Hamming Distance Metric

A. *Implementation of Codes*

Figure 7 illustrates the procedure of encoding and decoding in a communication system, where randomly binary generated data from binary data source are sent directly to the binary convolutional encoder and data that will be decoded by EVA are sent through the lock bit addition node before they are submitted to the encoder. Codewords from the encoder are submitted to the discrete channel.

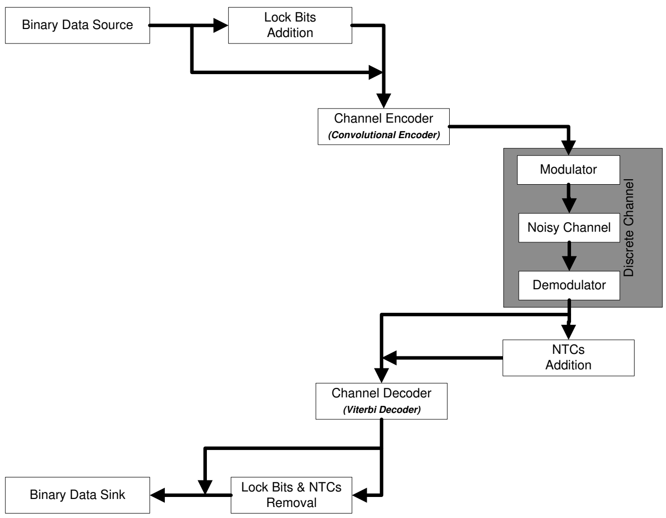

Fig. 7 Binary communication system block diagram used in simulation

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

The discrete channel modulates and demodulates the sent signal using Binary Phase-Shift Keying (BPSK) where zero bit (i.e. 0) is mapped to (-1) and one bit (i.e. 1) is mapped to (+1) and back, after the modulation signals are then released to the Additive White Gaussian Noise (AWGN) channel. Adding AWGN to signals in the transmission channel involves generating Gaussian random numbers, scaling the numbers according to the desired energy per symbol to noise density ratio (E_s/N_0), and adding the scaled Gaussian random numbers to the channel symbol values.

Received codewords from the discrete channel are ether sent to VA decoder directly or routed through NTCs node for adding NTCs. After decoding both lock bits and bits corresponding to the added NTCs are removed from data stream. Eventually, data obtained from the two streams (VA and EVA) are compared with the original generated data to determine the number of residual errors in each SNR value.

B. Performance Measure

The theoretical uncoded BER using a relation in (4) is used to compare the code gain [7]. Where, E_b/N_o is expressed as a ratio of the involved factors; and "*erfc*" is a complementary error function in MATLAB software. For uncoded channel, $E_s/N_0 = E_b/N_0$, since there is one channel symbol per bit.

$$BER = 0.5 * erfc(\sqrt{E_b / N_0}) \qquad (4)$$

However, the coded channel uses a relation (5) in the simulation.

$$E_s / N_0 = E_b / N_0 - 10\log_{10}(2) \qquad (5)$$

V. Results and Discussion

This section presents performance comparison between the VA and EVA decoders basing on their error correction capability in terms of BER and the residual errors. Fig. 8 compares the BER of uncoded channel, VA and 6 NTCs-EVA. Table 3 presents the counted residual errors from both VA and 6NTCs-EVA and the improvement obtained in each SNR value.

A. Code gain

It is clear from fig. 8 that 6NTC-EVA decoder has the lowest BER curve with the highest constant code gain of 2 dB almost over all SNR values. While the VA decoder has highest BER (above theory uncoded) below 4dB and it is persistently higher than that of 6NTC-EVA. The minimal VA code gain is minus two (-2) dB. It is important to note that, around and below 4 dB, VA has a negative code gains because VA faces difficult in decoding burst errors in this area. However, 6NTCs-EVA performs better than both VA and the theory-uncoded curves with a difference of more than 2dB. It can also be observed that, The VA and 6NTCs-EVA curves are far apart in lower SNR values (let say below 4 dB) and tend to come closer and closer as SNR values increase. This is because there are more errors generated in low SNR values which results in burst errors and create a great challenge to VA decoder. As the SNR value increases, few and random errors are generated and therefore VA decoder gains error correction power.

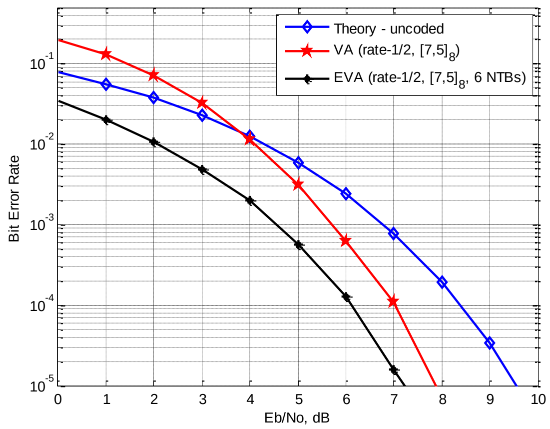

Fig.. 8. BER performance for theory-uncoded; VA and EVA in BPSK and AWGN

B. Residual Error

Table 2. Compares residual error obtained from the simulation between VA and 6NTCs-EVA. The results show that, 83.7 percent of total residual errors occurred in VA were corrected by applying 6NTC to the EVA. Averagely, 84.3 percent of residual errors that occurred in and below 4 dB in VA were successfully corrected by 6NTC-EVA decoder

TABLE III. VA VERSUS 6NTCS-EVA RESIDUAL ERRORS

Eb/No, dB	VA Residual Errors	6NTCs-EVA Residual Errors	Data Error Recovery Improvement (Bits)	Data Error Recovery Improvement (Percentage)
1	198187	34407	163780	82.6
2	129604	20417	109187	84.2
3	72308	10650	61658	85.3
4	32492	4824	27668	85.2
5	11581	1985	9596	82.9
6	3094	571	2523	81.5
7	614	127	487	79.3
8	110	16	94	85.5
9	7	2	5	71.4
10	0	0	0	0.0
Total	**447997**	**72999**	**374998**	**83.7**

C. Impact of Various NTCs Values on EVA

NTCs can be added as one, two, three codewords and so on. Fig. 9 shows that the increase in number of NTCs to EVA has an increasing impact on the decreasing rate of the number of residual errors. It further shows that, there is no significant reduction of residual errors with the further increase in the number of NTCs after 6 NTCs. These results concur with the explanation given by researchers in their work [9]

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

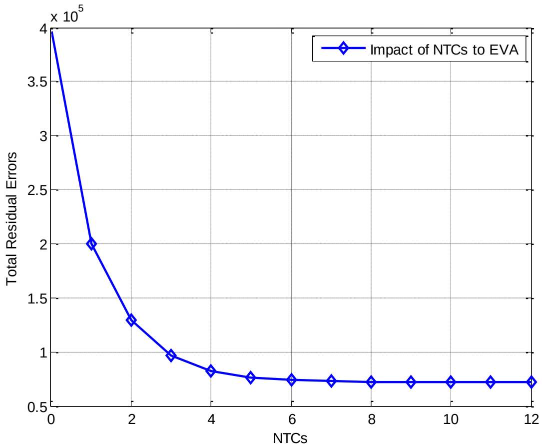

Fig. 9 Impact of NTCs to EVA

VI. CONCLUSIONS & RECOMMENDATIONS

This paper presented and assessed the NTCs-enhancement technique to Viterbi Algorithm at the receiving machine. The EVA is used to recover distorted codewords from a noisy channel. The decoder at the receiving machine recovers erroneous received codewords from a 1/2 rate, constraint length K is equal to 3 convolutional encoder at the sender point. MATLAB software was designed where a hard decision modulation and demodulation schemes using Binary Phase-Shift Keying (BPSK), Addition White Gaussian Noise (AWGN) were implemented. The simulation results showed 83.7 percent overall improvements in reducing residual errors when 6 NTCs were applied to VA decoder. This is a very significant improvement to VA decoders. The enhanced VA can be used in industries that demand for error free transmission such as telemedicine. However, the technique lowered the encoder's data transmission rate from 1/2 to 1/6. Further research of the proposed technique is highly recommended to show the impact of the technique in different platforms and applications using Viterbi Algorithm.

REFERENCES

[1] R. H. Morelos-Zaragoza, *The Art of Error Correcting Coding*, Second ed. The Atrium, Southern Gate, Chichester,West Sussex PO19 8SQ, England.: John Wiley & Sons Ltd., 2006.

[2] A. Viterbi, "Convolutional Codes and Their Performance in Communication Systems," *Communication Technology, IEEE Transactions on,* vol. 19, pp. 751-772, 1971.

[3] A. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *Information Theory, IEEE Transactions on,* vol. 13, pp. 260-269, 1967.

[4] D. Forney Jr, "Convolutional Codes Ii. Maximum-Likelihood Decoding," *Information and control,* vol. 25, pp. 222-266, 1974.

[5] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. Hoboken, New Jersey: John Wiley & Sons, Inc. , 2005.

[6] Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, and M. Guizani, "Home M2m Networks: Architectures, Standards, and QOS Improvement," *Communications Magazine, IEEE,* vol. 49, pp. 44-52, 2011.

[7] C. Fleming. (2006, August, 13). *A Tutorial on Convolutional Coding with Viterbi Decoding. Spectrum Applications*. Available: http://home.netcom.com/~chip.f/viterbi/tutorial.html

[8] S. I. Mrutu, A. Sam, and N. H. Mvungi, "Forward Error Correction Convolutional Codes for RTAs' Networks: An Overview," *International Journal of Computer Network and Information Security(IJCNIS),* vol. 6, pp. 19-27, 2014.

[9] S. I. Mrutu, A. Sam, and N. H. Mvungi, "Trellis Analysis of Transmission Burst Errors in Viterbi Decoding " *International Journal of Computer Science and Information Security (IJCSIS)* , vol. 12-8, pp.46-53, 2014.

[10] P. Gaj, J. Jasperneite, and M. Felser, "Computer Communication within Industrial Distributed Environment—a Survey," *Industrial Informatics, IEEE Transactions on,* vol. 9, pp. 182-189, 2013.

[11] J. Silvestre-Blanes, L. Almeida, R. Marau, and P. Pedreiras, "Online Qos Management for Multimedia Real-Time Transmission in Industrial Networks," *Industrial Electronics, IEEE Transactions on,* vol. 58, pp. 1061-1071, 2011.

[12] Z. Xu, S. Guan, and F. Yao, "A Novel Low-Time-Delay Convolutional Interleaver and Its Performance," in *Information and Communications Technologies (IETICT 2013), IET International Conference on*, 2013, pp. 208-212.

[13] G. Balakrishnan, M. Yang, Y. Jiang, and Y. Kim, "Performance Analysis of Error Control Codes for Wireless Sensor Networks," in *Information Technology, 2007. ITNG'07. Fourth International Conference on*, 2007, pp. 876-879.

[14] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer networks,* vol. 38, pp. 393-422, 2002.

AUTHORS PROFILE

Salehe I. Mrutu received his B.Sc. and M.Sc. degrees in Computer Science from The International University of Africa in 2003 and the University of Gezira in 2006 respectively. He is currently a Ph.D. scholar at the school of Computational and Communication Science and Engineering of the Nelson Mandela African Institution of Science and Technology in Arusha, Tanzania. He is also serving as assistant lecturer at the University of Dodoma under the school of informatics since 2007. His research interests include Forward Error Correction codes, quality-of-service provisioning and resource management for multimedia communications networks.

Anael Sam received his B.Sc., M.Sc. and Ph.D. in Electronics Engineering (Institute of Electronics and Photonics, Slovak University of Technology, Slovak Republic). He works as senior lecturer at the Nelson Mandela Institution of Science and Technology, Arusha, Tanzania. Dr Sam's specialization and research interests are in radio, multimedia and mobile communication systems; electronics and telecommunication engineering, software quality assurance engineering and mobile networks optimization. He is also a member of IEEE and ISQTB international.

Nerey H. Mvungi received the B.Sc. degree in electrical engineering from the University of Dar Es Salaam, Tanzania, in 1978; the M.Sc. degree in electronics control from Salford University, U.K. in 1980; and the Ph.D. degree from Leeds University Leeds, U.K. in 1989. He worked for a year with the Phillips Center for Technology, Eindhoven, Eindhoven, and the Netherlands February 1992 to Feb 1993. He was attached to Onersol Solar Energy Research Centre in Niamey June-July 1991 as ILO Consultant on Solar Energy Systems. Since his undergraduate graduation in 1978, he has worked as an academician and is now a full professor. He has mostly worked in the University of Dar es Salaam but for the period of September 2008 to June 2012 when he was at the University Dodoma in Tanzania to starting a new IT College as its founding Principal. Prof. Mvungi's research interests are in control and instrumentation, computer communication and applied electronics, lightning protection, rural access, power-quality aspects, and remote monitoring and control of energy consumption and digital broadcasting. He received a 2010 IBM Faculty Award.

http://sites.google.com/site/ijcsis/
ISSN 1947-5500