Performance of Hybrid Automatic Repeat Request Scheme with Turbo Codes

Norfishah Ab. Wahab[1], Member IEEE , Ir. Muhammad Ibrahim[2], Member IEEE, Suzi Seroja Sarnin[3], Member IEEE, Naimah Mat Isa[4]

Abstract— The introduction of turbo-codes in digital communication systems represents a step towards more reliable data transmissions. The reliability can be enhanced by combining the codes with automatic repeat request (ARQ), known as hybrid-ARQ (HARQ). This paper provides analysis and computer simulations in MATLAB for evaluating the performance of HARQ schemes using turbo-codes as the FEC. The performance of the scheme is analyzed in term of bit error rate (BER) and bit energy to noise ratio spectral density (E_b/N_0). Various parameters that influence the performance were investigated. Puncturing effects, number of iterations, decoding algorithm, size of packets and constrain length have been analyzed and evaluated to determine the effects and performances.

Index Terms— Bit Error Rate, Hybrid ARQ, iterative decoding, Log-Map algorithm, SOVA, Turbo codes,

I. INTRODUCTION

The introduction of the turbo codes [1], [2] has spurred a lot of research on the field of iterative decoding. Turbo code is a very effective error correction technique, reaching a performance considerably close to the capacity limit. Even though turbo codes have a very good bit error rate performance, they do not guarantee error-free data transmission. When error free data transmission is a system requirement, automatic repeat request (ARQ) scheme are employed.

In conventional ARQ schemes, frame errors are examined at the receiving end by an error detection code, usually cyclic redundancy checks (CRC). The CRC will compute the syndrome of the frame. If the syndrome is ‘0’ (means the frame passes the CRC), the receiving end sends an acknowledgement (ACK) of successful transmission to the receiver. If the receiver detects that the frame was erroneously decoded, it sends a negative acknowledgement (NAK), request for retransmission of that frame until a successful decoding is achieved. However, the system must tolerate certain delay for the retransmission to occur. As a result, ARQ scheme is more suitable for data, facsimile or still-image transmissions rather than voice or real-time communication transmissions in which retransmission is impossible. The basic ARQ schemes are the stop-and-wait, go-back-N, and selective repeat protocols. To improve the reliability of communication systems, these basic ARQ schemes are applied in conjunction with forward error correcting (FEC) methods, yielding the so-called hybrid ARQ/FEC schemes.

ARQ mechanism in conjunction with turbo codes have been considerably investigated in the recent years. One of the earliest works on this approach was done by Narayanan and Stuber [3]. They proposed a hybrid ARQ/FEC scheme where the information message is turbo coded prior to transmission. In the first transmission, the receiver operates as a regular turbo decoder, where no a priori information about the information message is known (equal a priori probabilities of ’0’s and ’1’s). If an error is detected at the receiver, then a request for retransmission is sent to the transmitter. The transmitter sends the coded bits again, but from the second transmission on the receiver operates as a ’modified’ turbo decoder. The soft decision of the last transmission is used as the a priori information for starting the decoding of the current transmission. With this simple and ingenious modification, it was shown in [3] that a considerable performance improvement can be achieved.

In iterative decoding, several decoding algorithms have been used, including the symbol and sequence detection techniques. The optimal MAP symbol estimation and its simplification called the Log-MAP algorithm have been introduced by Benedetto et al. in 1997. Further exploration of decoding algorithm uses the modified soft-output Viterbi algorithm (SOVA) which works for sequence detection techniques in SISO decoding algorithm [20]. Log-MAP algorithm is a transformation of MAP which has equivalent performance in practical implementation. Its performance is better compared to the Max-Log-MAP algorithms, which is suboptimal of MAP algorithms. Even though, the Max-Log-MAP algorithm was designed for easy implementation, but the performance suffers especially at the low signal-to-noise region [9]. Since Log-MAP performs better than Max-Log-MAP and more suited for parallel concatenates, only Log-MAP and SOVA are used as the decoding algorithms to equate the performance.

This paper analyzes the performance of decoding algorithm for Turbo codes with HARQ type 1 in AWGN fading channel using BPSK as the modulation scheme. The

1. Norfishah Ab Wahab is a lecturer at Electrical Engineering Faculty, UiTM, Malaysia. Tel: +603-55436092, Fax: +603-55435077, email: fishahahu@yahoo.com
2. Ir. Muhammad Ibrahim is a lecturer at Electrical Engineering Faculty, UiTM, Malaysia. Tel: +603-55435015, Fax: +603-55435077, email: muhdyus@yahoo.com
3. Suzi Seroja Sarnin is a lecturer at Electrical Engineering Faculty, UiTM, Malaysia. Tel: +603-55436011, Fax: +603-55435077, email: seroja121@yahoo.com
4. Naimah Mat Isa is a lecturer at Electrical Engineering Faculty, UNISEL, Malaysia. Tel: +603-55436011, Fax: +603-55435077, email: Naimah@yahoo.com

study is mainly focused on the performance of Turbo-codes as the FEC hybrid with ARQ type 1 without considering the time taken in each of the coded bits retransmission and other fading effects. This paper also investigates more parameters which affect the performance of the coding.

The rest of this paper is organized as follows. The system model is discussed in Section II. A brief introduction on the turbo-codes and the ARQ protocols is given. Retransmission scheme are presented in this section too. In Section III, results based on computer simulations using MATLAB are presented; also the performance of using Log-MAP algorithm and SOVA are analyzed as well as other comparatively studied on various factors that influenced the coding performance. Finally, conclusions are drawn in Section V.

II. SYSTEM MODEL

A. Turbo codes

Like all error correcting codes, Turbo codes work by imposing a structure on the transmitted bit sequence. If the received bit sequence does not match this known structure, the receiver knows an error has occurred. If the number of error is low enough and the structure is good enough, the receiver can determine which bits were received in error and reconstruct the correct sequence.

The difference between a Turbo code and conventional codes is the use of a recursive systematic encoder. Conventional, convolutional coders normally use a non-recursive structure. By feeding one of the outputs back to the input, a recursive encoding structure is obtained.

For moderate to large code length, turbo code with pseudo random interleaver shows best performance close to the Shannon limit, as long as the desired BER is not substantially lower than 10^{-5} [9]. Maximum likelihood decoding of turbo codes is not feasible due to interleaving. Therefore, Berrou et al. [1], [2] proposed an iterative decoding scheme for Turbo codes with relative low complexity, which seems to be “near maximum likelihood”. It requires soft-in/soft-out constituent decoders.

Iteratively, each constituent code is decoded individually using channel information of the systematic code word part and of the parity code word part belonging to this constituent code. Channel information of punctured parity symbols is inserted in such a way that the transition probabilities are equal for both binary symbols. Additional to the channel information, each constituent decoder takes into account extrinsic information of the data symbols generated by the other decoder.

Feeding forward extrinsic information is the key point in turbo decoding. Extrinsic information represents reliability information of a data symbol exclusively due to the channel information as well as the a priori information of all other code symbols and the constraints of the constituent code. In the subsequent decoding of the other constituent code extrinsic information is used as a priori information on the data symbols. The iteration stops if a given number of decoding steps is reached or a specified criterion indicates sufficient reliability of the decoding decision [9]. Thereby, decoding step means the individual decoding procedure for one constituent code.

B. Turbo Encoder

Two Recursive Systematic Convolutional (RSC) encoders are arranged in parallel, to construct a turbo encoder. The encoders are combined with a random interleaver, together with a puncturing block, as seen in Fig. I.

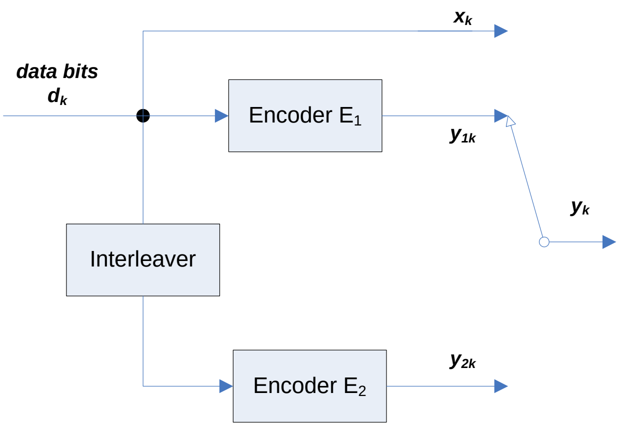

Fig.I: A turbo encoder

Data bits information message is denoted as d_k. The encoder E_1 operates on the input bits in their original order, while the encoder E_2 operates on the input bits as permuted by the interleaver. These sequences are modulated and sent through the channel.

Puncturing technique is used to improve the rate of the code. The puncturing selection process is performed by periodically eliminating one or more of the outputs generated by the constituent RSC encoders. The parity bits generated by these two encoders can be alternately eliminated so that the redundant bit of the first encoder is first transmitted, eliminating that of the second decoder, and in the following time instant the redundant bit of the second encoder is transmitted, eliminating that of the first.

The excellent BER performance of these codes is enhanced when the length of the interleaver is significantly large. The interleaving block, and its corresponding de-interleaver in the decoder, does not much increase the complexity of a turbo scheme, but it does introduce a significant delay in the system, which in some cases can be a strong drawback, depending on the application. The RSC-generated convolutional codes are comparatively simple, but offer excellent performance when iteratively decoded using soft-input–soft-output algorithms.

C. Turbo Decoder

The Turbo decoder is built in the similar way as the turbo encoder. Two elementary decoders are interconnected to each other, but in serial way, not parallel. An iterative decoding is proposed in [1], [2] which is basically a modification of the Bahl decoding algorithm [9]. The modification is necessary due to the recursive nature of the encoders. This algorithm differs from the Viterbi algorithm [10], where this algorithm produces soft outputs while the former algorithm produces hard outputs. Thus instead of outputting only 0 or 1, the output range is continuous and is a measure of log-likelihood ratio of every bit estimate.

Block diagram in Fig. II shows the iterative feedback scheme in a turbo decoder.!

!

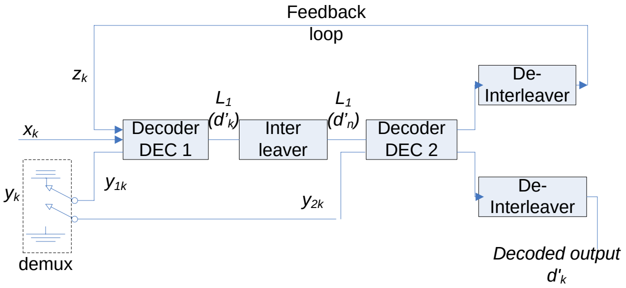

Fig. II: Turbo decoder

The inputs to decoder 1 are the kth systematic bit x_k, the redundant encoding information y_{1k}, and a third 'a priori' input z_k, which accepts feedback information from decoder 2. Decoder 2 accepts information from decoder 1, which it cannot derive from its own redundant input y_{2k}. Then, decoder 2 makes final decision based on this cumulative information for any bit.!!

D. Decoding Algorithms

The decoding algorithms are divided into two different families. The family is categorized based on the trellis-based detection algorithm. A Viterbi algorithm (VA) is using the sequence detection to accept and produce the a priori probabilities (APP) data in Soft-Input/Soft-Output. The conventional VA was introduced for convolutional codes [3]. As for Turbo codes which involves the parallel concatenates convolutional code, some modification and improvement have been made and referred as Soft Output Viterbi Algorithm (SOVA).

The symbol detection procedure has been used to decode the Turbo codes in Maximum A Posteriori (MAP) family. This family of decoding algorithms includes MAP which is the optimal method that produces the APP information, its additive form, Log-MAP, and its suboptimal form, Max-Log-MAP. According to the previous performance analysis [14], the Log-MAP algorithm is a transformation of MAP which has equivalent performance in practical implementation. The performance is better compared to the Max-Log-MAP algorithms. Even though the Max-Log-MAP algorithm was designed for easy implementation, the performance suffers especially at the low signal-to-noise region.

Since Log-MAP performs better than Max-Log-MAP and more suited to parallel concatenates, only Log-MAP and SOVA are used as the decoding algorithms to evaluate the performance.

E. ARQ Protocols

As mentioned in the previous section, the most popular ARQ protocols are stop-and wait, go-back-N and selective repeat. In this paper, we present the performance of hybrid ARQ which implies stop and wait protocols.

i) Stop-and-wait

In Stop and Wait (SW) ARQ protocol, the transmitter continues transmit the packets only when the previously transmitted packet has been successfully acknowledged. Hence, after transmit a packet, the transmitter waits for its acknowledgement. Once its acknowledgement has been received, the next packet is transmitted. However, if an acknowledgement does not arrive until a timeout timer expires, the packet is retransmitted. Therefore, in SW there is never more than a single packet that is unacknowledged at any given instant of time. Since the transmitter does not use the available channel during time intervals it waits for an ACK, the maximum data transfer rate that can be supported is limited.

ii) Go-Back-N

When Go-Back-N (GBN) is employed, packets are transmitted continuously. However, at the receiver, the packets are only accepted in the order in which they were transmitted. Packets received out of sequence are discarded and not acknowledged. Since the receiver accepts packets only in-sequence, after a timeout, the transmitter retransmits the packet that timed out and all packets with sequence numbers that follow the one that was retransmitted. Hence, each time a timeout occurs all packets that are yet to be acknowledged are retransmitted.

iii) Selective Repeat

Unlike SW, when using Selective Repeat (SR), packets, if available, are transmitted continuously at the transmitter. As before, the receiver acknowledges each successfully received packet by transmitting an ACK bearing the sequence number of the packet being acknowledged. If an acknowledgement is not received for a packet before the expiration of the timeout, the packet is retransmitted. Once a packet has been retransmitted the transmitter resumes transmission of packets from where it left off, i.e., if '*a*' is the packet with the largest sequence number that has been transmitted, packet with sequence number '*a*+1' is transmitted next (assuming that no other timers have expired in the meantime). Observe that when SR is employed packets can be accepted out of sequence. Hence, packets received out of sequence have to be buffered and sequenced before they can be delivered.

F. HARQ type I using Turbo codes

In this paper, the performance of hybrid ARQ based on Turbo-codes has been evaluated. The objective for the implementation of the system is to detect the error burst, discard the affected packets or frames, and request a retransmission.

A hybrid ARQ system consists of an FEC subsystem contained in an ARQ system. In this paper, the FEC is Turbo codes while the ARQ is SW protocol. Fig. III shows the state diagram of the system. Source data will be broken up into packets. These packets have to be encoded at the CRC and FEC encoder, consequently. Purpose of CRC is to detect error at the receiver end while FEC is responsible to correct error before passes to the CRC to re-check the received packets.

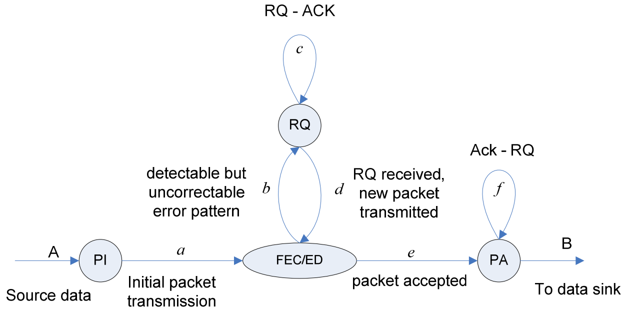

Fig. III: State diagram for a Type I hybrid ARQ.

If CRC detects any error in the received packets, the receiver will request for retransmission of the same packets. The retransmissions in this scheme are limited to three times to reduce delay.

III. SIMULATION RESULTS AND DISCUSSIONS

In this section, we investigate the performance of HARQ scheme using Turbo codes in term of bit error rate (BER) and bit energy to noise ratio spectral density (E_b/N_0) through MATLAB simulations. The algorithm for this proposed scheme has been designed based on the state diagram in Fig 3. Modulation scheme applied in this simulation is BPSK. AWGN model acts as noise model in this simulation. The punctured (r=1/2) and unpunctured (r=1/3) codes are considered to show the effects of puncturing process on BER performance.

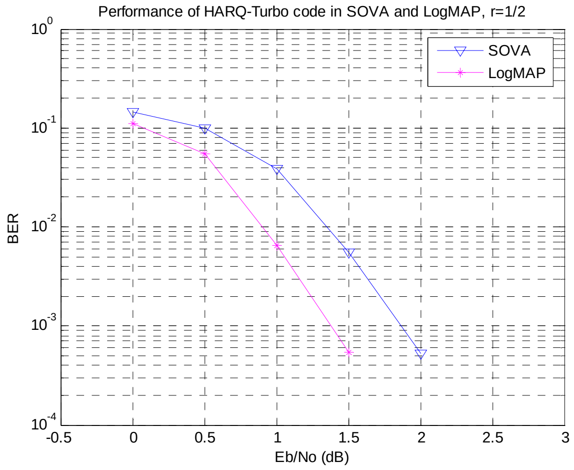

Fig. IV: Performance of Harq for SOVA and LogMAP with code rate, r=1/2

Fig. IV and Fig. V depict the performances comparison between Log MAP and SOVA decoding algorithm in the HARQ scheme using turbo codes. It shows that for both punctured and unpunctured code rate, the Log MAP outperform SOVA approximately 0.5dB at BER of $10^{-3.5}$. Since Log-MAP performs better than SOVA and more suited to parallel concatenates, only Log-MAP is used as the decoding algorithms to equate the study of influenced parameters performance. The simulation results proved the reported previous works [4], [5], [12], [17] and [19].

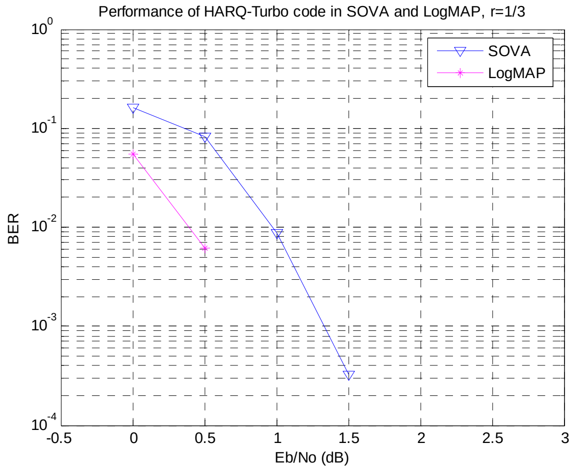

Fig V: Performance of Harq FOR SOVA and LogMAP with code rate, r=1/3

Fig. VI and VII depict the performances comparison for both R=1/2(punctured) and 1/3(unpunctured) with different number of iteration up to 5. The performance of Turbo code is also included in both simulations which were obtained from previous research, done by [5]. The results show that the highest value of iteration (I=5) have the lowest BER, approximately 10^{-4} at Eb/No=1.5dB in HARQ scheme. However, when compared to the Turbo scheme from the previous work by [5], at Eb/No = 1.5dB, the BER=$10^{-2.7}$[5]. From Fig. VI and Fig. VII, it is proven that the proposed scheme shows better performance than a single turbo code.

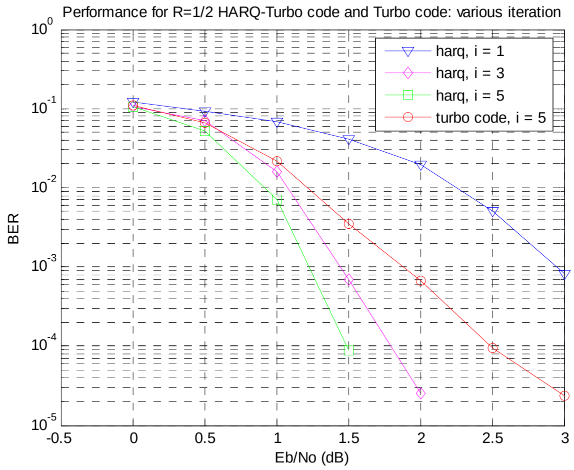

Fig.VI: Performance of Harq vs Turbo code with various iterations, r=1/2

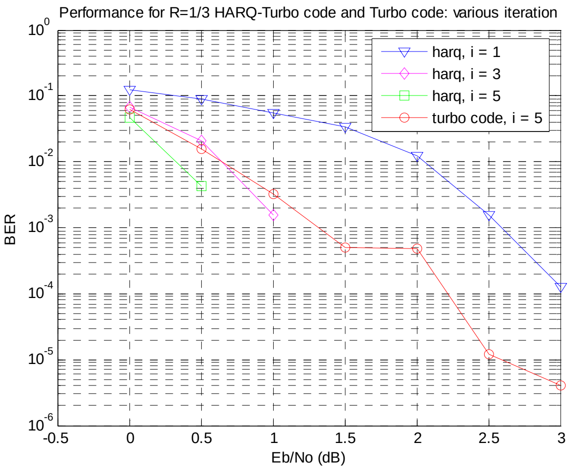

Fig. VII: Performance of Harq vs Turbo code with various iterations=1/3

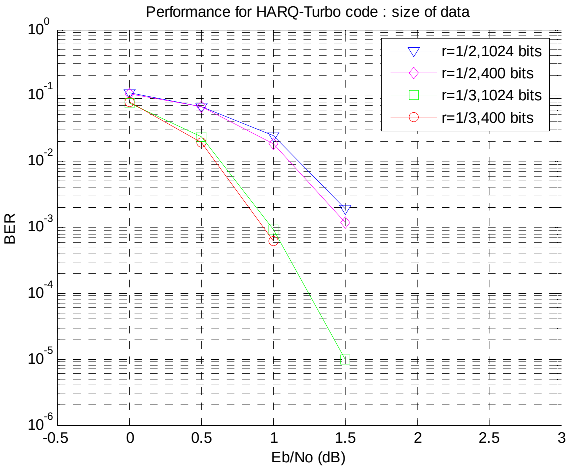

Fig. VIII: Performance of Harq with different size of data

We observed that different sizes of data did not give much impact on BER performance at Eb/No=1dB as shown in Fig. VIII. However, it can also be observed that, the performance of the unpunctured code rate improved with lower BER and Eb/No when the frame size is reduced.

Apparently, the performance of the proposed scheme is much better for unpunctured code rate(r=1/3) as compared to the punctured code rate(r=1/2) with the difference of Eb/No approximately 0.5 dB at BER = 10^{-3}.

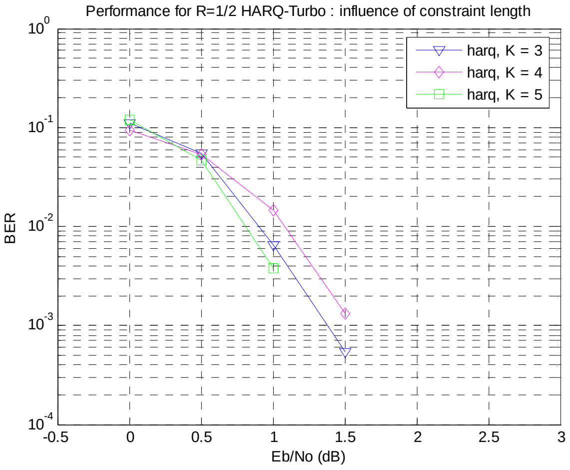

Fig. IX: Performance of Harq with different constraint length for r=1/2

Performance of R=1/3 HARQ – Turbo : influence of constraint length.

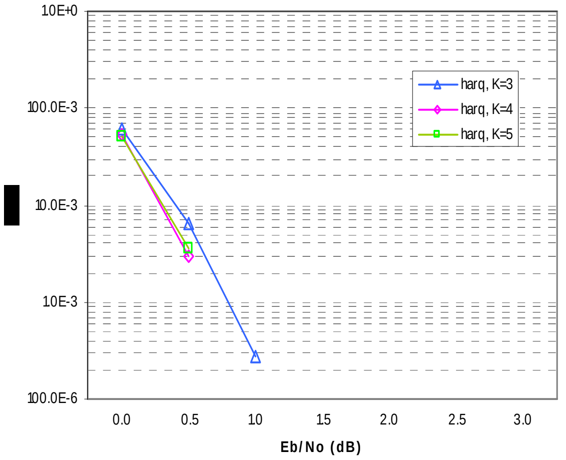

Fig. X: Performance of Harq with different constraint length for r=1/3

Fig. IX and Fig. X depict the performance comparisons of different constraint length, K = 3, 4 and 5. With all the simulations performed earlier, the parameters are chosen to achieve optimum efficiency. By choosing 400 information bits and AWGN channel, will simulate the wireless environment. The frame size is chosen closed to the CDMA applications (200 bits). The simulation is carried out using punctured and unpunctured codes to show the effects of puncturing process on BER performance with number of iterations=4

We observed that, K=5 depicts the best performance when compared to K = 3 and 4 in this proposed scheme. The performance of the punctured and unpunctured shows that the BER for unpunctured for all the Ks outperform the punctured code.

IV. CONCLUSIONS

This paper presented the results obtained from simulating the performance of HARQ type 1 using Turbo codes. By varying the parameters, a detailed summary was undertaken with the following important outcome:

- The combination of HARQ type 1 using Turbo (FEC) codes shows a better performance as compared to the FEC alone.
- Log-MAP decoding algorithm showed to have a better performance than the SOVA algorithms in terms of reducing bit error rate.
- The HARQ type 1 using Turbo codes performance has significant influenced by puncturing the parity bits to the encoder. Unpunctured codes with the lower code rate (rate $= \frac{1}{3}$) lead to a better performance than punctured codes.
- The increase of iteration numbers will increase the performance of the HARQ type 1 using Turbo codes. Hence, as the number of iteration increased, the performance of this scheme improved with lower BER and E_b/N_o.
- With the same number of iteration and code length, the performance of HARQ type 1 using Turbo (FEC) outperform the performance of Turbo codes only.
- The HARQ type 1 using Turbo code shows

ISBN: 978-988-18210-4-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

insignificant performance on different sizes of frame. However, the unpunctured code shows significant improvement in BER when compared to the punctured codes.

- Increase of the constraint length K, which composed from the generator matrices, contributes to better performance in both punctured and punctured codes.

V. FUTURE WORKS

The HARQ type 1 using Turbo codes proved to have a good performance for the AWGN fading channel as compared to the turbo codes alone. Thus, future work will include:

1. The effect of noise and multipath fading channels such as Rayleigh Fading, Rician Fading and Nakagami Fading channels can be determined using the MATLAB simulations.
2. To look into the performance of Hybrid Turbo codes with ARQ type 1 in OFDMA environment for the WiMax application.
3. Channel equalization techniques can also aid to mitigate inter-symbol interference in order to improve the performance of the system.
4. The various designed of interleaver can also be applied for next development to analyze the HARQ type 1 using Turbo codes performance.
5. To look into the performance of Hybrid Turbo codes with ARQ type 2 and 3 in wireless environment.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon Limit Error-Correcting Coding And Decoding: Turbo-Codes,” in ICC 1993, (Geneva, Switzerland), pp. 1064–1070, May 1993

[2] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: Turbo-codes.” IEEE Trans. Commun, vol. 44, no. 10, pp. 1261-1271, Oct. 1996.

[3] K. R. Narayanan and G. L. Stuber, “A novel ARQ technique using the turbo coding principle,” IEEE Commun. Letters, vol. 1, no. 2, pp. 49-51, Mar. 1997

[4] S.Wicker, “Error Control Systems for Digital Communication and Storage,” Englewood Cliffs: Prentice Hall, 1995

[5] Noni Nadiah, “Performance analysis on decoding algorithms using turbo codes,” M.Sc Thesis, University Technology MARA, Shah Alam, Nov 2007.

[6] U. Elisabeth, “Hybrid ARQ Using Serially Concatenated Block Codes for Real-Time Communication - An Iterative Decoding Approach,” Licentiate Thesis, Chalmers University of Technology, Sweden, 2001

[7] K. Oteng-Amoako, J. Yuan, S. Nooshabadi, “Selective Hybrid-ARQ turbo schemes with various Combining methods in Fading Channels,” in Proceeding of Workshop on Modelling and Optimisation on Mobile Ad Hoc Networking and Computing (Wiopt03), France, 2003

[8] F. Chiti, R. Fantacci, F. Versaci, “Turbo soft combining hybrid ARQ techniques: theory and application to 3G wireless networks,” in Annales des Télécommunications, 2005, v:60, n:1-2, pp:169-183

[9] L. Bahl, J. Jelinek, J. Raviv, and F. Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate,” IEEE Trans. on Information Theory, vol. IT-20, pp. 284–287, Feb 1974.

[10] G. D. Forney, “The Viterbi Algorithm,” Proc. Of the IEEE, vol. 61, pp. 268-278, Mar 1973.

[11] Tsg-Ran Wg1#17, “Performance Comparison of Hybrid-ARQ Schemes,” Motorola, Stockholm, Sweden, Tech. Rep. TSGR1#17(00)1396-Adhoc#24-HSDPA, 2000.

[12] A. Banerjee, D. J. Costello Jr., T. E. Fuja, “Performance of hybrid ARQ schemes using turbo trellis coded modulation for wireless channels,” University of Notre Dame, 2002.

[13] T. M. Kim, H. Shin, and J. H. Lee, “A novel error detection scheme for Turbo coded hybrid ARQ,” in Proc. of the IEEE VTC 2002-Fall, Vancouver, Canada, Sept. 24-28, pp. 2168-2171, 2002.

[14] T. Rodrigues, R. D. Souza, M. E. Pellenz, “Hybrid ARQ Scheme Based on Recursive Convolutional Codes and Turbo Decoding”, IEEE Transactions on Communications, 2008.

[15] M.A. Kousa and A.H. Mugaibel, “Puncturing effects on turbo codes,” IEE Pme. Commun. Vol. 149, No 3, June 2002

[16] M. Liinaharja, “Studies On the Performance of Some ARQ Schemes,” Helsinki University of Technology Communications Laboratory Tech. Rep. T54, 2006

[17] S. Falahati, P. Frenger, P. Orten, T. Ottosson and A. Svensson, “Convolutional Coding and ARQ Schemes for Wireless Communications,” in Proc. IEEE Nordic Radio Symposium, Saltsjöbaden, Sweden, October 1998, pp203-210.

[18] P. Robertson, E. Villebrum and P. Hoher, “A comparison of optimal and suboptimal MAP decoding algorithms operating in the Log-domain”, Int. Conf. on Communications, pp 1009-1013, 1995

[19] S. Lin and D. J. Costello, “Error Control Coding: Fundamentals and Applications,” 2nd ed. Prentice Hall, 2004.

[20] M. Kouraichi, O. B. Belghith, A. Kachouri and L. Kamoun, “Evaluation of SOVA algorithm in Turbo code”, IEEE Transactions on Information Theory, pp. 659-663, March 2004.