An improved Log-MAP algorithm based on polynomial regression function for LTE Turbo decoding

Duy Huy Nguyen, Hang Nguyen

HAL Id: hal-01283990
https://hal.science/hal-01283990

Submitted on 7 Mar 2016

An improved Log-MAP algorithm based on polynomial regression function for Turbo decoding in LTE system

Duy-Huy Nguyen and Hang Nguyen
Department of Wireless Network and Multimedia Services
Institut Mines-Telecom, Telecom SudParis
Samovar Laboratory, UMR 5157, CNRS, Evry, France
{duy_huy.nguyen, hang.nguyen}@telecom-sudparis.eu

Abstract—**This paper proposes an improved Logarithmic Maximum A Posteriori (Log-MAP) algorithm for Turbo decoding in the Third Generation Partnership Project Long Term Evolution (3GPP LTE). In the proposed algorithm, we exploit the understanding of polynomial regression function to approximately compute the logarithm term (also called correction function) in the Jacobian logarithmic function. The goal is to replace the correction function on an other function with the approximated performance and the reduced computational complexity. Simulation results show that the performance of the proposed algorithm is closest to the Log-MAP algorithm for Turbo decoding under Additive White Gaussian Noise (AWGN) channel and can offer about maximum 0.4dB performance gain than the Max-Log-MAP algorithm and higher than other Log-MAP-based algorithms. The proposed algorithm has much simpler computational complexity in comparison with the Log-MAP algorithm and slightly increased compared to the Max-Log-MAP algorithm.**

Index Terms—**LTE, Turbo codes, Log-MAP, polynomial regression, iterative decoding, channel coding**

I. Introduction

In present market of mobile communication in the world, 3GPP LTE is developing strongly and is deployed by the most communications operators [1]. It is a mobile network which has high data rate, low delay, and fully packet-based. In LTE network, Turbo code is widely utilized for channel coding. The 3GPP Turbo code is a systematic Parallel Concatenated Convolution Code (PCCC) with two 8-state constituent encoders and one turbo code internal interleaver [2]. It was first introduced by Berrou et al [3], and until now it is still the most powerful error correcting codes that obtains the performance closest to the Shannon capacity. It is also applied in many modern wireless communication standards, such as HSDPA [4] and LTE [2]. The Turbo Encoder and Decoder structure of LTE system is simply presented in Figure 1 [4], where x_k and L_k denote the systematic bits and the Log-looklihood ratio (LLR), respectively.

In the Turbo decoder, the symbol-by-symbol Log-MAP algorithm is an optimal algorithm for iterative decoding in white Gaussian noise [5]. However, this algorithm is executed in logarithmic domain or reading data from a big table will spend much time and logarithmic operations are not easy to implement in hardware. In order to reduce the high

complexity of the optimal algorithm, its sub-optimal variants were proposed, such as the table-lookup Log-MAP, Max-Log-MAP [6] and SOVA [7], which are utilized in practice to meet the tradeoff between performance and complexity. In these algorithms, the Max-Log-MAP algorithm has the least computational complexity, but it has the worst performance in comparison with the Log-MAP algorithm. It has a performance degradation about 0.4dB [6] so will reduce about 10% capacity of the system. Therefore, in order to improve the performance of the Max-Log-MAP algorithm while keeping the acceptable complexity, many proposals have been devoted in the literatures [8], [9], [10], [11], [12].

In this paper, we propose a novel algorithm based on the exploitation of the polynomial regression function to approximate the correction function in the Log-MAP algorithm. In terms of performance, the proposed algorithm has the closest performance to the Log-MAP algorithm. In addition, the proposed algorithm can be easily implemented in hardware involving shift registers, multiplications, comparators and addition operations. Therefore, the proposed algorithm can meet the tradeoff between performance and complexity.

This paper is organized as follows: The optimal algorithm and the sub-optimal algorithms for decoding in Turbo codes are reviewed in section II. The proposed algorithm is introduced in section III. Section IV represents simulation results and performance evaluation of the proposed algorithm and the other sub-optimal algorithms. The conclusion is represented in section V.

II. Decoding algorithms in Turbo codes

In this section, we briefly review some typical decoding algorithms in Turbo codes, we mainly focus on family of the MAP algorithms, consist of the optimal algorithm and some typical sub-optimal algorithms. Detailed description are represented in [6],[12].

A. The Log-MAP algorithm

The Log-MAP algorithm is a MAP algorithm which is implemented in logarithmic domain to reduce the computational complexity. It is an optimal algorithm for iterative decoding in
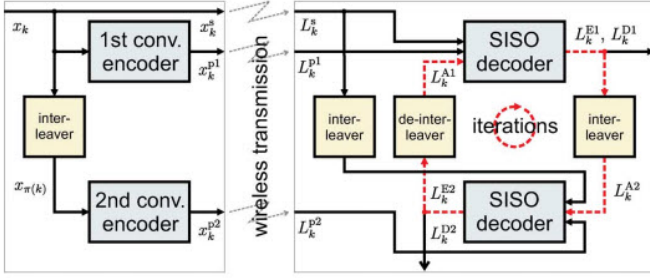
Fig. 1. The structure of Turbo Encoder and Decoder of LTE system

Turbo decoder. The objective of the Log-MAP algorithm is to calculate the log-likelihood ratio [12]. The a priori information for the information bit u_k is computed as follows:

$$L(u_k) = ln\Big(\sum_{(s_{k-1},s_k),u_k=+1} e^{\alpha_k^*(s_{k-1})+\beta_{k+1}^*(s_k)+\gamma_k^*(s_{k-1},s_k)}\Big)$$
$$-ln\Big(\sum_{(s_{k-1},s_k),u_k=-1} e^{\alpha_k^*(s_{k-1})+\beta_{k+1}^*(s_k)+\gamma_k^*(s_{k-1},s_k)}\Big) \tag{1}$$

In which:

- u_k is the information bits
- s_k and s_{k-1} denote the state at *k*th and *k-1*th time instant, respectively.

In order to calculate the equation (1), we need to compute forward and backward recursive metrics, denoted as $\alpha_k(s_k)$ and $\beta_k(s_k)$. They are calculated as follows:

$$\alpha_k^*(S_k) = \ln(\alpha_k(S_k)) = \max_{s_{k-1}\in\sigma_{k-1}}^* (\gamma(S_{k-1},S_k)+\alpha_{k-1}^*(S_{k-1})) \tag{2}$$

$$\beta_k^*(S_k) = \ln(\beta_k(S_k)) = \max_{s_{k+1}\in\sigma_{k+1}}^* (\gamma(S_k,S_{k+1})+\beta_{k+1}^*(S_{k+1})) \tag{3}$$

In which:

- σ_{k-1} and σ_k are collection of all states at the moment $k-1$ and k, respectively
- γ is the branch metrics

Function $\overset{*}{\max}(.)$ in (2) and (3) is reckoned by using the Jacobian algorithm as follows:

$$\overset{*}{\max}(\delta_1,\delta_2) = \ln(e^{\delta_1}+e^{\delta_2}) = max(\delta_1,\delta_2)+ln(1+e^{-|\delta_1-\delta_2|}) \tag{4}$$

Where $ln(1+e^{-|\delta_1-\delta_2|})$ is a correction function which corrects the error caused by the max approximation and makes the optimization for the Log-MAP algorithm. By replacing (2) and (3) into (4), we attain:

$$L(u_k) = \max[\beta_k^*(S_k) + \gamma(S_{k-1},S_k) + \alpha_{k-1}^*(S_{k-1})]$$
$$- \max[\beta_k^*(S_k) + \gamma(S_k,S_{k-1}) + \alpha_{k-1}^*(S_{k-1})] \tag{5}$$

B. The Log-MAP-based algorithms

The sub-optimal algorithms are based on the Log-MAP algorithm by replacing the correction function by an approximated function. Some typical sub-optimal algorithms consist of Max-Log-MAP [6], Constant Log-MAP [10], Linear Log-MAP [11] and Non-linear Log-MAP [12]. These

algorithms try to reach the performance close to the Log-MAP algorithm with an acceptable complexity, obviously, they have the computational complexity lower than the Log-MAP algorithm. The approximated correction functions of these sub-optimal algorithms are expressed in following subsections:

1) The Max-Log-MAP algorithm: The Max-Log-MAP algorithm has the least complexity because it omits the correction function. Hence, it is the simplest algorithm to implement but has the worst performance. The performance for the Max-Log-MAP algorithm gives up to a 10% performance drop [6] when compared to the Log-MAP algorithm. With the Max-Log-MAP algorithm, the correction function $f(x) = ln(1 + e^{-x})$ with $x = |\delta_1 - \delta_2|$ is worked out as follows:

$$ln(1 + e^{-|\delta_1-\delta_2|}) \approx 0 \tag{6}$$

2) The Constant Log-MAP algorithm: This algorithm is proposed by [9], [10], the correction function is approximated with the following principle:

$$ln(1 + e^{-|\delta_1-\delta_2|}) \approx \begin{cases} \frac{3}{8}, if |\delta_1 - \delta_2| < 2 \\ 0, otherwise \end{cases} \tag{7}$$

The Constant Log-MAP algorithm has a simple execution in hardware but with swap in performance.

3) The Linear Log-MAP algorithm: In [11], the authors use the MacLaurin series to calculate the linear approximation for the correction function. It is observed that the correction function is effective when $f(x)$ is around zero. Thus, the MacLaurin series can be applied to approximate the correction function about zero. The authors propose the approximation for the correction term is given as:

$$ln(1 + e^{-|\delta_1-\delta_2|}) \approx \max(0, \ln 2 - \frac{1}{2}|\delta_1 - \delta_2|) \tag{8}$$

4) The Non-linear Log-MAP algorithm: This algorithm is proposed by the authors in [8]. It is the non-linear approximation for the correction term. It is inspired by observing the curve of the exact correction terms of Constant Log-MAP and Linear Log-MAP algorithms. The correction function is approximated as follows:

$$ln(1 + e^{-|\delta_1-\delta_2|}) \approx \frac{\ln 2}{2^{|\delta_1-\delta_2|}} \tag{9}$$

III. THE PROPOSED LOG-MAP ALGORITHM BASED ON POLYNOMIAL REGRESSION FUNCTION

Although the Log-MAP algorithm has the best performance, but its correction function carried out in logarithmic domain would bring some undesirable issues. According to [12], saving the results of the $ln(1 + e^{-|\delta_1-\delta_2|})$ in a lookup table would involve a quantization error caused by truncation of the input of the lookup. Another problem with the Log-MAP algorithm is that many lookup tables for a wide range of operating signal-to-noise ratios (SNRs), this will increases the hardware cost. Moreover, reading data from logarithmic tables is a time consuming process. So the correction function in the Log-MAP algorithm need be replaced by an other
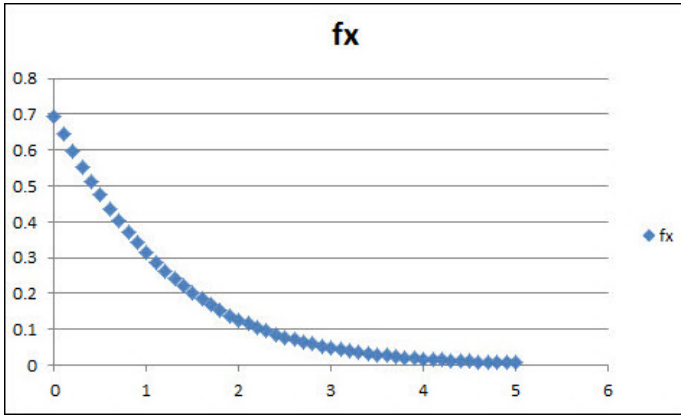
Fig. 2. The distribution of the correction function

Fig. 3. The proposed approximated function for the correction function in the Log-MAP algorithm

function which has approximated performance but has simpler complexity. Therefore, we put forward a novel function to replace it. In order to find the new correction function, we exploit the understanding of polynomial regression function. The correction function is deployed into a form of polynomial function as follows:

$$f(x) = ln(1 + e^{-x}) \approx a_0 + a_1x + a_2x^2 + ... + a_nx^n \quad (10)$$

In order to determine factors of $a_0, a_1, a_2, ..., a_n$ and the suitable value of n, we have to determine a dataset consists of points presenting the relationship between $f(x)$ and x. According to [6], the values of x should only selecting between 0 and 5 to obtain ideal approximation. Thus, with $f(x) = ln(1 + e^{-x})$, for $x = 0$ to 5 with step 0.1, we obtain a dataset of 51 values of $f(x)$, respectively. This work is performed in Microsoft Excel, and then, we use the the Scatter method to represent the distribution of points of $f(x)$ [13]. The plot performs the distribution of $f(x)$ is represented in Figure 2. In order to determine the approximated polynomial regression function, we use the Trendline method [14]. This is a method widely utilized in statistical probability field. It permits building a regressive function exactly from a set of known points.

Microsoft Excel supports the minimum value of the degree of polynomial regression function (n) is 2 and the maximum value is 6. The accuracy of the approximation is verified through the goodness of fit test against the exact Log-MAP curve with parameter of R-squared [15]. After changing the value of n from 2 to 6, we obtain parameters of of R-squared are 0.9828, 0.9996 and 1s, respectively. However, we see that if choose $n = 2$ then we do not obtain the performance very close to the Log-MAP algorithm, otherwise, if we choose n = 4, 5 or 6 then will increase the computational complexity. Hence, we propose the value of $n = 3$. For this reason, we attain the approximated function for logarithmic term in the Log-MAP algorithm presented in Figure 3 and is given as:

$$f(x) = -0.0098x^3 + 0.1164x^2 - 0.474x + 0.6855 \quad (11)$$

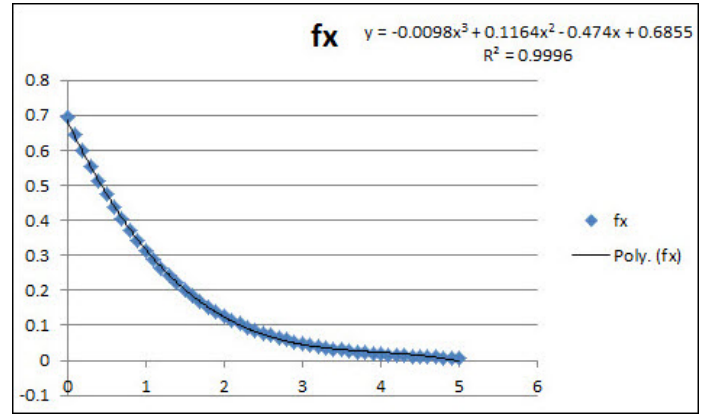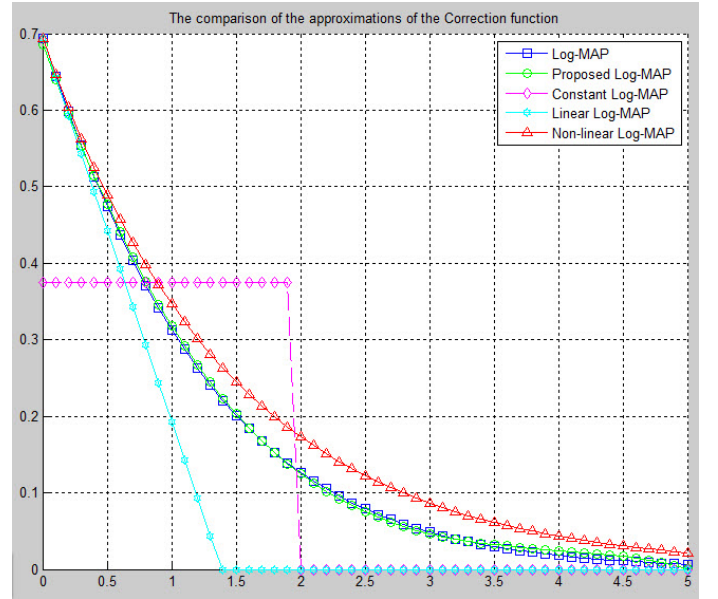As mentioned above, according to [6], in order to attain the

Fig. 4. The comparison of the approximations of the correction function

ideal approximation, just selecting the value of argument x between 0 and 5. Therefore, we can rewrite the proposed approximated correction function as follows:

$$ln(1 + e^{-x}) \approx$$

$$\begin{cases} -0.0098x^3 + 0.1164x^2 - 0.474x + 0.6855, if x \leq 5 \\ 0, otherwise \end{cases}$$

(12)

Through equation (12), we can calculate the logarithmic term in (4), and so, we attain the proposed Log-MAP algorithm.

IV. SIMULATION RESULTS AND PERFORMANCE EVALUATION

Figure 5 shows the simulated performance under AWGN channel for the proposed algorithm and the other Log-MAP-based algorithms, including Log-MAP, Max-Log-MAP, Constant Log-MAP, Linear Log-MAP and Non-linear Log-MAP
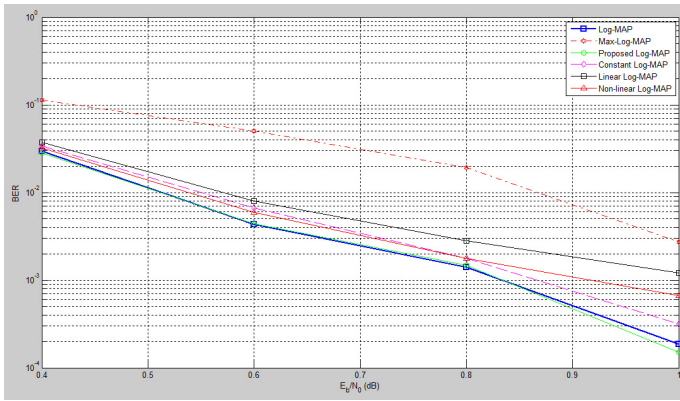
3

Fig. 5.   BER performances of the proposed Log-MAP algorithm and the other Log-MAP algorithms with block size $N = 1024$
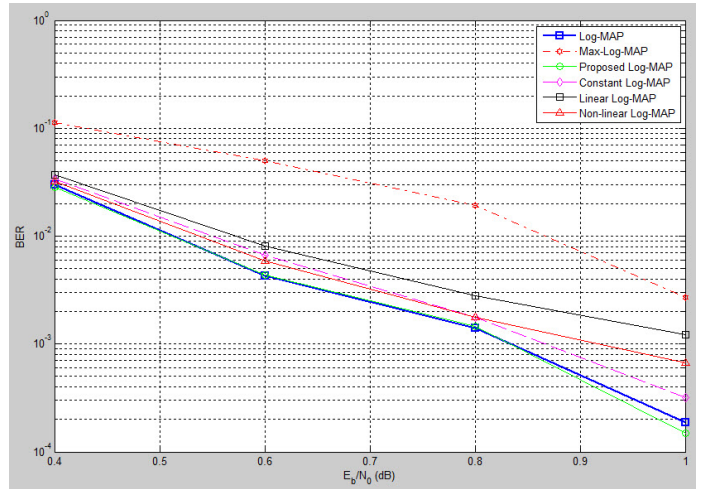

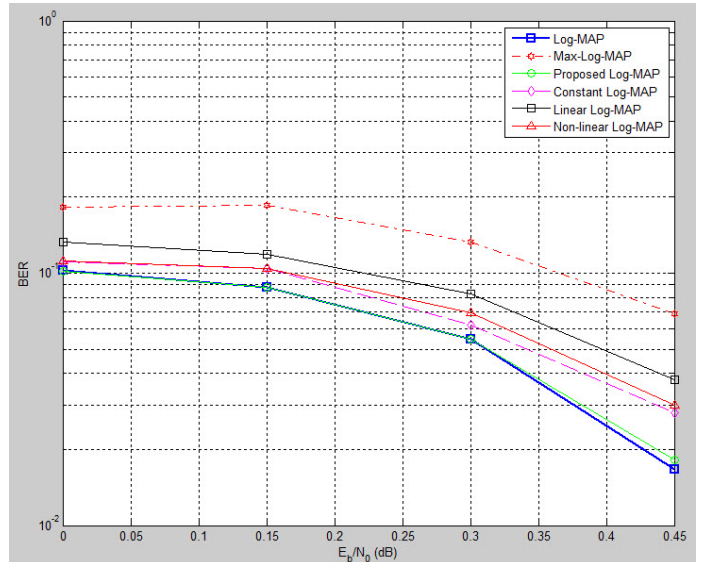Fig. 6.   BER performances of the proposed Log-MAP algorithm the other Log-MAP algorithms with block size $N = 512$

Fig. 7.   BER performances of the proposed Log-MAP algorithm and the other Log-MAP algorithms at low SNRs with block size $N = 1024$

algorithms. The Bit Error Rate (BER) performance is simulated in a rate-1/3, 8-states Turbo coded system with the transfer function for the constituent code $G = [1, \frac{1+D+D^3}{1+D^2+D^3}]$ [2]. The block size is $N = 1024$, using random interleaver and the maximum number of iterations for decoding was set to 5. As shown in Figure 5, the proposed algorithm offers the BER performance closest to the Log-MAP algorithm and it outperforms the other sub-optimal algorithms.

Figure 6 shows the performance for the proposed algorithm, the Log-MAP algorithm and the other sub-optimal algorithms with the similar simulated parameters to figure 5, while its block size is $N = 512$. It is clear that although block size N changed, the proposed algorithm still obtains the BER performance most close to the Log-MAP algorithm.

The simulated parameters utilized in Figure 7 is similar to Figure 5 but at low SNRs. The simulation result shows that the proposed algorithm has the BER performance nearly as identical as the Log-MAP algorithm, this is because the Log-MAP algorithm is sensitive to the SNR [16]. According to [15], at high SNRs, the performance of the Turbo code approaches the Max-Log-MAP algorithm and does not depend heavily on the correction function. Otherwise, at low SNRs, the decoder examines the a priori or extrinsic information from the previous decoder more. The distribution of argument x in the correction function $f(x)$ increased to areas close to zero where it is most effective. This is because the a priori is Gaussian distributed with increased number of iterations for decoding. The sensitivity of the Log-MAP algorithm to SNR is also more reported in encoders with more memory components [16].

The simulation results demonstrate that the proposed algorithm can be applied in real system in practice. It permits obtaining the nearly best performance with the computational complexity decreased in comparison with the Log-MAP algorithm because it eliminates the logarithmic and exponential operations from the correction function.

V. CONCLUSION

In this paper, we represent a novel approximated function for the correction function in the Log-MAP algorithm. This approximated function is deployed based on the understanding of the polynomial regression function. The proposed Log-MAP algorithm is a sub-optimal algorithm which achieves the performance closest to the Log-MAP algorithm with the much simpler computational complexity than the Log-MAP algorithm. Thus, it can be easily implemented in hardware involving shift registers, multiplications, comparators and addition operations. The simulation results show that the proposed algorithm outperforms the other Log-MAP-based algorithms, particularly are superior to the Max-Log-MAP algorithm with slightly increased complexity.

REFERENCES

[1] Long Yu, Xu Wang, and Jian Liu. An improved rate matching algorithm for 3gpp lte turbo code. In *Communications and Mobile Computing*

(CMC), 2011 Third International Conference on, pages 345–348. IEEE, 2011.

[2] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding. TS 36.212, 3rd Generation Partnership Project (3GPP), September 2011.

[3] Claude Berrou and Alain Glavieux. Near optimum error correcting coding and decoding: Turbo-codes. *Communications, IEEE Transactions on*, 44(10):1261–1271, 1996.

[4] Christoph Studer, Christian Benkeser, Sandro Belfanti, and Quiting Huang. Design and implementation of a parallel turbo-decoder asic for 3gpp-lte. *Solid-State Circuits, IEEE Journal of*, 46(1):8–17, 2011.

[5] Ji-Hoon Kim and In-Cheol Park. Bit-level extrinsic information exchange method for double-binary turbo codes. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 56(1):81–85, 2009.

[6] Patrick Robertson, Emmanuelle Villebrun, and Peter Hoeher. A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain. In *Communications, 1995. ICC'95 Seattle,'Gateway to Globalization', 1995 IEEE International Conference on*, volume 2, pages 1009–1013. IEEE, 1995.

[7] Joachim Hagenauer and Peter Hoeher. A viterbi algorithm with soft-decision outputs and its applications. In *Global Telecommunications Conference and Exhibition'Communications Technology for the 1990s and Beyond'(GLOBECOM), 1989. IEEE*, pages 1680–1686. IEEE, 1989.

[8] Hao Wang, Hongwen Yang, and Dacheng Yang. Improved log-map decoding algorithm for turbo-like codes. *Communications Letters, IEEE*, 10(3):186–188, 2006.

[9] Warren J Gross and P Glenn Gulak. Simplified map algorithm suitable for implementation of turbo decoders. *Electronics Letters*, 34(16):1577–1578, 1998.

[10] S Papaharalabos, P Sweeney, and BG Evans. Constant log-map decoding algorithm for duo-binary turbo codes. *Electronics Letters*, 42(12):709–710, 2006.

[11] Jung-Fu Cheng and Tony Ottosson. Linearly approximated log-map algorithms for turbo decoding. In *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, volume 3, pages 2252–2256. IEEE, 2000.

[12] Shahram Talakoub, Leila Sabeti, Behnam Shahrrava, and Majid Ahmadi. An improved max-log-map algorithm for turbo decoding and turbo equalization. *Instrumentation and Measurement, IEEE Transactions on*, 56(3):1058–1063, 2007.

[13] Richard M Heiberger and Erich Neuwirth. *R through Excel: A spreadsheet interface for statistics, data analysis, and graphics*. Wiley Online Library, 2009.

[14] Bruce R Hargreaves and Thomas P McWilliams. Polynomial trendline function flaws in microsoft excel. *Computational Statistics & Data Analysis*, 54(4):1190–1196, 2010.

[15] Li Li Lim and David Wee Gin Lim. Hybrid log-map algorithm for turbo decoding over awgn channel. In *ICWMC 2011, The Seventh International Conference on Wireless and Mobile Communications*, pages 211–214, 2011.

[16] M-A Khalighi. Effect of mismatched snr on the performance of log-map turbo detector. *Vehicular Technology, IEEE Transactions on*, 52(5):1386–1397, 2003.