Article

Low–Complexity, Fast–Convergence Decoding in AWGN Channels: A Joint LLR Correction and Decoding Approach

Anoush Mirbadin * and Abolfazl Zaraki

Posted Date: 15 May 2025

doi: 10.20944/preprints202505.1199.v1

Keywords: Low Comlplexity; Fast Convergence Decoding; LLR Updating; KL Divergence; Joint; LLR Correction; Decoding

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2025 doi:10.20944/preprints202505.1199.v1

Article

Low–Complexity, Fast–Convergence Decoding in AWGN Channels: A Joint LLR Correction and Decoding Approach

Anoush Mirbadin [1,†] and **Abolfazl Zaraki** [2,†]

[1] Department of Engineering, SELTA Digital Platforms Company, 29010 Cadeo, Italy
[2] School of Physics, Engineering and Computer Science (SPECS), and Robotics Research Group of the University of Hertfordshire, Hatfield, AL10 9AB, United Kingdom
[*] Correspondence: anoush.mirbadin@dplatforms.it
[†] These authors contributed equally to this work.

Abstract: Thanks to the product rule of Gaussian distributions, the memory of channel coding schemes, such as low–density parity–check (LDPC) codes used in this paper, is reflected in the mean of a single Gaussian distribution, obtained through the product of re–scaled Gaussian observations in additive white Gaussian noise (AWGN) channels. Consequently, employing a novel bit log–liklihood ratio (LLR) updating algorithm, in conjunction with an appropriate scheduling procedure, increases the convergence speed of the decoder considerably. Bit LLR values close to zero are accumulated with those obtained in the current iteration of the receiver. Simulation results demonstrate a substantial improvement (close to 50%) in the convergence speed of the proposed algorithm compared to traditional ones. This approach can also be applied to conventional sequence and symbol detection strategies in the presence of memory. Although this approach only affects convergence in AWGN channels, it could play a vital role in scenarios involving nonlinear parameters, such as phase noise and multipath channels.

Keywords: low comlplexity; fast convergence decoding; LLR updating; KL divergence; joint; LLR correction; decoding

1. Introduction

The Additive White Gaussian Noise (AWGN) channel serves as a foundational model in the analysis and design of digital communication systems. Despite the existence of more sophisticated models that capture fading, interference, and non-linearities, the AWGN channel remains a critical benchmark for evaluating the performance of coding and decoding techniques.

Among the most widely adopted techniques for decoding over AWGN channels are iterative decoding schemes, such as the Viterbi algorithm for convolutional codes and Belief Propagation (BP) for Low-Density Parity-Check (LDPC) codes. These methods iteratively refine symbol estimates using local message passing or state evaluation to achieve near optimal decoding. In particular, LDPC and Turbo codes (both decoded iteratively) have demonstrated remarkable performance, approaching the Shannon limit in AWGN environments [1]. However, while these algorithms are effective, their convergence speed remains a practical limitation, especially in low-latency or power-constrained applications.

To address this, recent research has explored enhancing iterative decoding through machine learning (ML) and deep learning (DL) frameworks. For instance, DeepTurbo integrates recurrent neural networks into the Turbo decoding process, improving convergence in the low-SNR regime [2]. Similarly, recent studies have explored the use of deep learning to model and enhance decoding dynamics through end-to-end optimization [3]. These techniques reflect a growing trend toward data-driven improvements in classical decoding frameworks. Additionally, hybrid architectures such as the Attention Turbo-Autoencoder [4] and deep unfolding networks [5] combine the structural advantages of traditional iterative decoding with the adaptability of neural networks. These methods often yield faster or more stable convergence under AWGN conditions by learning optimal update schedules, message weights, or stopping criteria.

The main aim of this work is to speed up the convergence of the traditional iterative decoding algorithms such as LDPC or product codes. To this end, re–scaled Gaussian observations are multiplied together in order to satisfy the Bayes' theorem and reflect the channel coding memory on the observation side of the factor graph. This is followed by a bit log–liklihood ratio (LLR) correction procedure which selects and updates the provided bit LLRs to the decoder. The algorithm will accumulate the value of a bit LLR only if it was in the ambiguity region, i.e., the region close to zero. The length of the region is adjusted during the simulations. The algorithm applies the sum–product algorithm (SPA) on factor graphs. At every iteration, the receiver makes hard decisions on the received symbols using extrinsic information provided by the decoder. At the first iteration, receiver applies Kullback–Leibler (KL) divergence to the observations prior to re–scaling and multiplying procedures. Te reach the maximum performance of the algorithm, iterating strategy uses a unique scheduling procedure. This guarantees that the first iteration of the receiver equals the traditional decoding algorithm. The simulation results demonstrate significant gains of the proposed algorithm.

The paper is organized as follows: Section 2 introduces the system model. Section 3 discusses the corresponding factor graph and its SPA equations. Receiver implementations are detailed in Section 4, with performance analysis provided in Section 5. Finally, conclusions are drawn in Section 6.

2. System Model

In a modern wireless communication system, the data bits are encoded using an LDPC encoder. This is followed by a Gray encoder and an M–ary digital modulator. The sequence of the coded modulation symbols $\mathbf{c} = (\mathbf{c_0}, \mathbf{c_1}, ..., \mathbf{c_{K-1}})$ is then transmitted over an AWGN channel. The baseband equivalent of the received samples can be described by

$$r_k = c_k + n_k, \ (k = 0, 1, ..., K-1) \tag{1}$$

where $\{n_k\}$ is a sequence of independent and identically distributed (i.i.d.), complex circularly symmetric Gaussian noise components, i.e., $n_k \sim \mathcal{CN}(n_k; 0, 2\sigma^2)$.

3. Factor Graph

According to the Bayes' theorem, the receiver can be implemented by factorizing the posterior distribution of data bits $\boldsymbol{b}$ given the observation samples $\boldsymbol{r} = \{r_k\}$ and applying the SPA to the corresponding factor graph. It is assumed that the encoding function η_c maps the information sequence $\boldsymbol{b}$ to the codeword $\boldsymbol{c}$. The factorization yields

$$p(\boldsymbol{b}|\boldsymbol{r}) \propto P(\boldsymbol{b})p(\boldsymbol{r}|\boldsymbol{b}) \propto T[\boldsymbol{c} = \eta_c(\boldsymbol{b})] \prod_{k=0}^{K-1} p(r_k|c_k) \propto T[\boldsymbol{c} = \eta_c(\boldsymbol{b})] \prod_{k=0}^{K-1} p(r_k|c_k) \tag{2}$$

where $T[.]$ is the code indicator function which equals to 1 if $\boldsymbol{c}$ is the codeword corresponding to $\boldsymbol{b}$ and zero otherwise. Equation (2) ends with the product of Gaussian observations. To achieve this, re–scaling of distributions is necessary. The re–scaled observation at time epoch k is shown below.

$$p(r_k|c_k) \propto \exp\left\{-\frac{|r_k - c_k|^2}{2\sigma^2}\right\} \propto \exp\left\{-\frac{|1 - \frac{c_k}{r_k}|^2}{\frac{2\sigma^2}{|r_k|^2}}\right\} \tag{3}$$

According to the Gaussian product rule, the product of re–scaled distributions provides a single Gaussian distribution which contains the memory of the channel coder in its mean. It should be mentioned that we employed the product of two re–scaled Gaussian distributions at every time epoch prior to bit LLR computations in our last work on multipath mitigation [6]. The corresponding factor graph is depicted in Fig. 1.

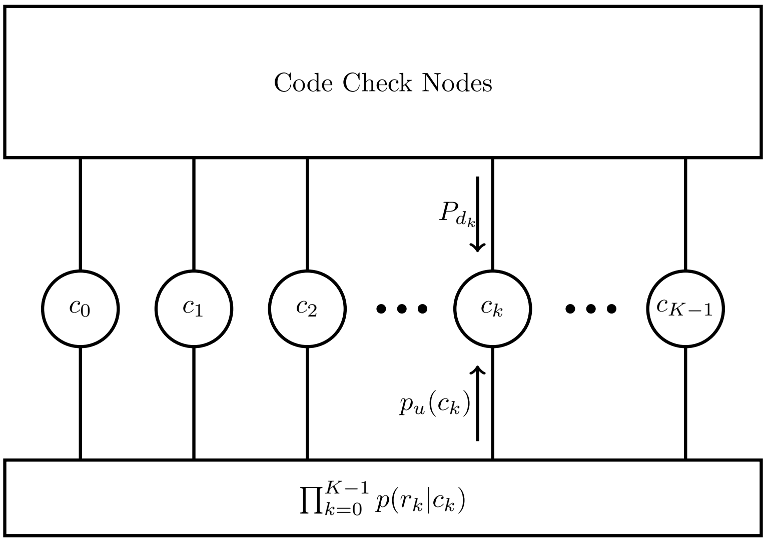

Figure 1. Factor graph representation of equation (3).

4. Implementation Strategies

At every iteration, the receiver performs hard decisions on the received bits, $\mathbf{b}^{'}$. This is accomplished by using the extrinsic information provided by the LDPC decoder. Consequently, a set of M–ary modulation symbols $\mathbf{c}^{'} = (\mathbf{c}_{\mathbf{0}}^{'}, \mathbf{c}_{\mathbf{1}}^{'}, ..., \mathbf{c}_{\mathbf{k-1}}^{'}, \mathbf{c}_{\mathbf{k+1}}^{'}, ..., \mathbf{c}_{\mathbf{K-1}}^{'})$ is also detected. At the time epoch k, we perform message passing on the corresponding factor graph to compute the belief about the symbol c_k provided by the rest of the factor graph. The message $p_u(c_k)$ is a Gaussian distribution whose mean and variance are computed by

$$\sigma_k^2 = \frac{\sigma^2}{\sum_{i=0}^{i=K-1} |r_i|^2} \tag{4}$$

$$\mu_k = \sigma_k^2 \left(\sum_{k^{'} \neq k} \frac{\frac{c_{k^{'}}^{'}}{r_{k^{'}}}}{\frac{\sigma^2}{|r_{k^{'}}|^2}} + \frac{\frac{c_k}{r_k}}{\frac{\sigma^2}{|r_k|^2}} \right) \tag{5}$$

This is followed by computations of bit LLR for every bit associated with the constellation symbol c_k in the Gray ordering. When the bit LLR value falls in the ambiguity region, whose length is adjusted during simulations, its value will be accumulated by the obtained LLR at the current iteration. At the first iteration of the receiver, KL divergence is applied to each observation prior to the Gaussian multiplication. In this way, only at the first iteration, the receiver works like the traditional decoders. The performance of the receiver is maximized by employing a proper scheduling procedure as elaborated on extensively in the next section.

5. Simulation Results

To assess the performance of the proposed algorithm in AWGN environments, computer simulations were conducted. Specifically, bit error rate (BER) curves were plotted as a function of the signal-to-noise ratio (SNR). We used a $(3,6)$-regular LDPC code with a block length of 4000 bits. Binary and quadrature phase shift keying (BPSK and QPSK) modulation schemes were used as the modulation schemes. Since a bit error rate (BER) of 10^{-4} is considered sufficient for performance evaluation in practical applications, the curves were plotted down to this error level. To enhance clarity

doi:10.20944/preprints202505.1199.v1

and better visualize the differences, the BER curves were also plotted as a function of $\log_{10}(\text{BER})$. Figures 2 and 3 show the simulation results for BPSK and QPSK modulation formats, respectively. The red curves are associated with the original LDPC codes, named by their maximum number of decoder iterations, e.g., 200–LDPC. The blue curves correspond to the proposed joint bit LLR correction and decoding algorithms. Their names reflect the employed scheduling strategy that assumes a single receiver iteration. In this iteration, the LDPC decoder first performs 2 internal iterations, followed by the bit LLR correction step. Afterward, the LDPC decoder continues until the maximum number of decoding iterations is reached, i.e., it performs a total of the maximum number of LDPC iterations. The length of the decision region for LLR correction algorithm is set to 1 for BPSK modulation format and 0.25 for QPSK one.

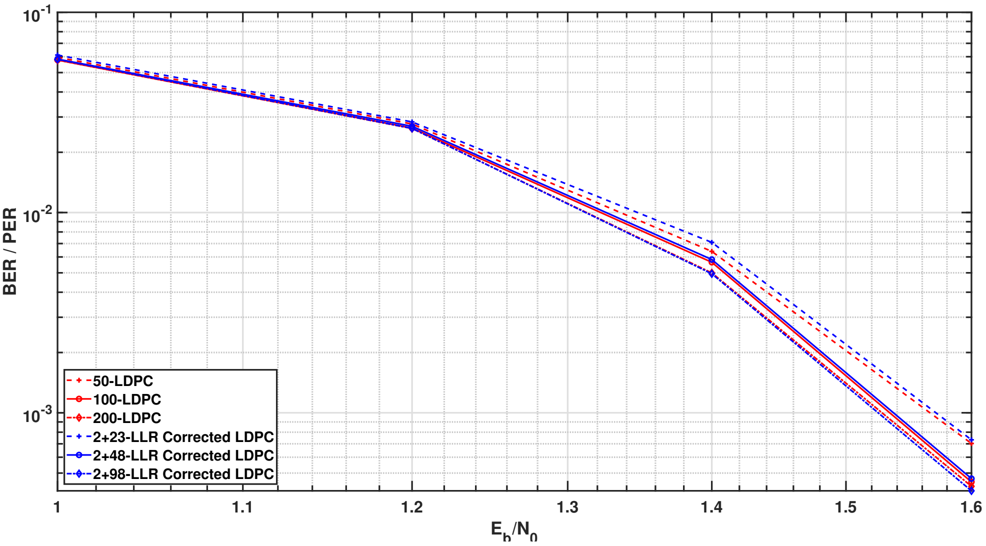

Figure 2. LDPC decoding for BPSK modulation format. Red curves are the original LDPC schemes while the Blue ones represents the proposed bit LLR correction and decoding algorithm.

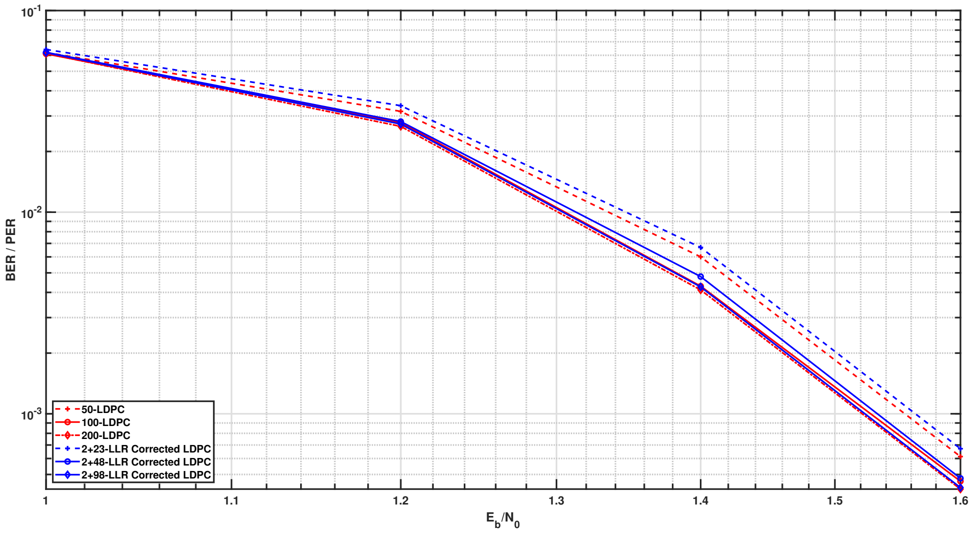

Figure 3. LDPC decoding for QPSK modulation format. Red curves are the original LDPC schemes while the Blue ones represents the proposed bit LLR correction and decoding algorithm.

A preliminary analysis of the results indicates that increasing the maximum number of iterations leads to improved convergence speed with the proposed algorithm, yielding approximately a 50% gain.

Furthermore, a comparison of convergence regions and receiver iteration counts reveals a trade-off between the ambiguity region length and the constellation size. Specifically, the decision region for higher-order constellations can be determined by dividing the binary region by the constellation size (M).

6. Conclusion and Future Work

The proposed method achieves faster communication performance compared to existing approaches. This represents a significant advancement for satellite communications and low–power communication systems. To address this, the memory of channel coding schemes has first been reflected in the observation part of the receiver. This has been obtained by re–scaling the independent observations and computing their product in line with the SPA algorithm on factor graphs. Then, a novel bit LLR correction has been proposed to increase the convergence speed of the decoder. For future work in this context, the receiver might also be implemented by applying KL divergence to the symbols whose LLRs fall in the ambiguity region. In such a scenario, the exact computation of the KL divergence is unachievable (even by using cell function in the MATLAB simulations) due to the large number of the calculations. Therefore, the KL divergence can only be applied to the extreme situations, e.g., all bits being zeros or ones.

Author Contributions: Conceptualization, A.M. and A.Z.; methodology, A.M. and A.Z.; software, A.M. and A.Z.; validation, A.M. and A.Z.; formal analysis, A.M. and A.Z.; investigation, A.M. and A.Z.; resources, A.M. and A.Z.; data curation, A.M. and A.Z.; writing—original draft preparation, A.M. and A.Z.; writing—review and editing, A.M. and A.Z.; visualization, A.M. and A.Z.; supervision, A.M. and A.Z.; project administration, A.M. and A.Z.; funding acquisition, A.M. and A.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Richardson, T.J.; Urbanke, R.L. Modern Coding Theory; Cambridge University Press: Cambridge, UK, 2008. https://doi.org/10.1017/CBO9780511791338
2. Jiang, Y.; Kim, H.; Asnani, H.; Kannan, S.; Oh, S.; Viswanath, P. DEEPTURBO: Deep Turbo Decoder. In Proceedings of the 2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Cannes, France, 2–5 July 2019; pp. 1–5. https://doi.org/10.1109/SPAWC.2019.8815400
3. Nachmani, E.; Marciano, E.; Lugosch, L.; Gross, W.J.; Burshtein, D.; Be'ery, Y. Deep Learning Methods for Improved Decoding of Linear Codes. IEEE J. Sel. Top. Signal Process. 2018, 12, 119–131. https://doi.org/10.1109/JSTSP.2017.2788405
4. Lu, M.; Zhou, B.; Bu, Z. Attention-Empowered Residual Autoencoder for End-to-End Communication Systems. IEEE Commun. Lett. 2023, 27, 1140–1144. https://doi.org/10.1109/LCOMM.2023.3242281
5. Gruber, T.; Cammerer, S.; Hoydis, J.; ten Brink, S. On Deep Learning-Based Channel Decoding. In Proceedings of the 2017 51st Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 22–24 March 2017; pp. 1–6. https://doi.org/10.1109/CISS.2017.7926071
6. Mirbadin, A.; Zaraki, A. Partial Path Overlapping Mitigation: An Initial Stage for Joint Detection and Decoding in Multipath Channels Using the Sum–Product Algorithm. Appl. Sci. 2024, 14, 9175. https://doi.org/10.3390/app14209175

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.