Finite-Length Performance Analysis of LDPC Coded Continuous Phase Modulation

Md. Noor-A-Rahim, Zilong Liu, Yong Liang Guan, Lajos Hanzo

***Abstract*—Serial concatenation of LDPC codes and continuous phase modulation (CPM) has recently gained significant attention due to its capacity-approaching performance, efficient detection as well as owing to its constant-envelope nature. Most of the previous contributions on LDPC coded CPM were devoted to the design of LDPC codes and their asymptotic performance analysis. However, there is a paucity of work on the finite-length performance estimation of LDPC coded CPM, primarily because existing performance estimation techniques cannot be readily applied to the LDPC coded CPM. To fill this gap, we conceive an analytical bit error probability estimation technique for finite-length LDPC coded CPM in the waterfall region. Numerical results are provided both for regular and irregular LDPC codes having different codeword lengths, demonstrating that the estimated performances are closely matched by the simulated ones.**

***Keywords*—Low-density parity check (LDPC) code, Continuous phase modulation (CPM), Density evolution, Waterfall region.**

I. Introduction

Continuous phase modulation (CPM) is an attractive non-linear modulation scheme whose signals exhibit constant envelope (i.e., high power efficiency) and tight spectrum confinement [1], [2]. Hence CPM has been used in many areas such as military and satellite communications [3], the 2nd-generation global system of mobile communications (GSM), and in millimeter communications [4]. In most applications CPM is serially concatenated with different channel codes (e.g., convolutional codes and BCH codes) [6]. At the receiver, typically iterative turbo decoding is used for exchanging soft extrinsic information between the CPM (inner) decoder and the channel (outer) decoder.

Low-density parity check (LDPC) codes are also eminently suitable for concatenation with CPM as a benefit of their near-capacity performance and efficient message-passing decoding [7–9], given their superiority over other coding schemes. Apart from simulations, the performance of LDPC coded CPM has also been evaluated for sufficiently long codewords using EXIT charts [10] and density evolution [11]. However, their finite-length performance has only been studied by Monte-Carlo simulations. To elaborate a little further, the prediction of finite-length behavior of LDPC codes was investigated in [12–14], but most of them considered phase shift keying (PSK) modulation. To the best of our knowledge, however, the finite-length performance estimation of LDPC coded CPM has not been considered before.

To fill this gap, we conceive a finite-length performance estimation technique for LDPC-coded binary CPM schemes in the waterfall region[1] by exploiting the inherent properties of CPM. In the impressive treatises of [13], [14] the pseudo-random occurrence of errors was exploited for the associated performance estimation. However, their techniques are not applicable to our LDPC coded CPM scheme, which has distinctive turbo equalization properties owing to the intentional intersymbol interference imposed by partial-response CPM. Moreover, the performance prediction schemes of [13], [14] are based on density evolution analysis, but density evolution of LDPC coded CPM is not available in the open literature. We will present simulation results for both regular and irregular LDPC codes having different codeword lengths and then compare them to the results obtained by the proposed estimation algorithm. It is shown that the estimation algorithm conceived closely matches the finite-length BER of LDPC coded CPM schemes.

II. Preliminaries

A. Low-Density Parity Check (LDPC) Codes

LDPC codes are often represented by their Tanner graph, which consists of N variable nodes (N is also known as the codeword length) and M check nodes. An LDPC code is said to be (d_v, d_c)-regular, when each variable node has d_v edges to connect with check nodes and each check node has d_c edges to connect with the variable nodes. We refer to d_v as the variable node degree and d_c as the check node degree. By contrast, an LDPC code with non-uniform variable node degrees and/or non-uniform check node degrees is known as an irregular LDPC code. Formally, an LDPC code can be characterized by the edge perspective degree distribution polynomials $\lambda(x) = \sum_{i=2}^{d_{v_{\max}}} \lambda_i x^{i-1}$ and $\rho(x) = \sum_{i=2}^{d_{c_{\max}}} \rho_i x^{i-1}$, where λ_i is the fraction of edges that are connected to degree i variable nodes, ρ_i is the fraction of edges that are connected

Md. Noor-A-Rahim is with School of Computer Science & IT, University College Cork, Ireland (E-mail: m.rahim@cs.ucc.ie), Zilong Liu is with School of Computer Science and Electronic Engineering, University of Essex, UK (E-mail: zilong.liu@essex.ac.uk), Yong Liang Guan is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore (E-mail: eylguan@ntu.edu.sg), and Lajos Hanzo is with the School of Electronics and Computer Science, University of Southampton, UK (E-mail: lh@ecs.soton.ac.uk).

The work of Yong Liang Guan was supported by Temasek Laboratories, NTU, through the Signal Research Programme 3 under Grant DSOCL17187.

L. Hanzo would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council projects EP/N004558/1, EP/P034284/1, EP/P034284/1, EP/P003990/1 (COALESCE), of the Royal Society's Global Challenges Research Fund Grant as well as of the European Research Council's Advanced Fellow Grant QuantCom.

[1] In general, the LDPC code's bit error ratio (BER) curve can be divided into two regions, namely the waterfall and the error-floor region. In the waterfall region, the BER decreases rapidly, as the Signal-to-Noise Ratio (SNR) is increased. By contrast, in the error floor region imposed by short cycles [11] in the code's structure, the BER only decays gracefully.

to degree i check nodes and the maximum degrees of variable and check nodes are denoted by $d_{v_{\max}}$ and $d_{c_{\max}}$, respectively. The code rate with these degree distributions is given by $R = 1 - \frac{\sum_j \rho_j / j}{\sum_j \lambda_j / j}$.

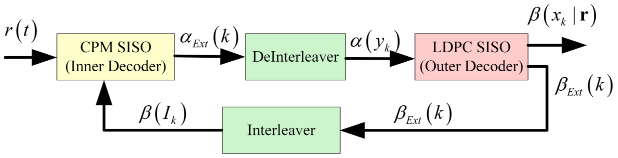

Fig. 1. Structure of LDPC-coded CPM's iterative receiver

B. LDPC-coded CPM Structure

Let $\mathbf{x} = [x_0, x_1, \cdots, x_{K-1}]$ be the message vector to be coded, where K denotes the message length. After LDPC encoding, we obtain $\mathbf{y} = [y_0, y_1, \cdots, y_{N-1}]$ (where $N = K/R$) which will be sent to a random interleaver (with size N). By applying the mapping operation of $1 \rightarrow +1, 0 \rightarrow -1$ to $\mathbf{y}$, the input vector at the CPM modulator is $\mathbf{I} = [I_0, I_1, \cdots, I_{N-1}]$ which leads to the CPM signal $s(t)$ using the following modulation method. Specifically, an equivalent lowpass binary CPM waveform $s(t)$ is expressed as $s(t) = \exp[i\phi(t; \mathbf{I})]$, where

$$\phi(t; \mathbf{I}) = 2\pi h \sum_{k=-\infty}^{n} I_k \psi(t - kT), \quad nT \leq t \leq (n+1)T, \tag{1}$$

is the time-varying phase depending on the information sequence $\mathbf{I} = \{I_k\}_{k=-\infty}^{n}$ with $I_k \in \{-1, 1\}$, h is the modulation index, and T is the symbol duration. The phase-shaping waveform $\psi(t)$ is defined as the integral of $g(t)$, a frequency pulse of duration LT, i.e., $g(t) = 0$ for $t < 0$ and $t \geq LT$, where L is a positive integer. When $L = 1$, we have a *full-response* CPM, and for $L > 1$, a *partial-response* CPM. After passing through an AWGN channel, the received signal is $r(t) = s(t) + n(t)$, where $n(t)$ is a circularly-symmetric white Gaussian process with $n(t) \sim \mathcal{N}(0, \sigma_n^2)$, $\sigma_n^2 = N_0/2$. The CPM soft-input soft-output (SISO) decoder has two inputs, namely the received signal $r(t)$ and the *a priori* information of the CPM symbols, i.e., $\beta(I_k) = \log P(I_k = +1)/P(I_k = -1)$ for $k = 0, 1, \cdots, N-1$. In fact, $\beta(I_k)$ is the interleaved *extrinsic* information $\beta_{Ext}(k)$ gleaned from the LDPC decoder. In this paper, the CPM decoder relies on the BCJR algorithm [15] for maximum *a posteriori* (MAP) detection, whereas the LDPC decoder employs the message-passing based sum product algorithm (SPA). As seen from Fig. 1, the LDPC SISO decoder only has one *a priori* input, namely the deinterleaved *extrinsic* information from $\alpha_{Ext}(k)$, shown as $\alpha(y_k) = \log P(y_k = 1)/P(y_k = 0)$ for $k = 0, 1, \cdots, N-1$. Let us define $\mathbf{r} = [r_0, r_1, \cdots, r_{N-1}]$, where $r_k = \{r(t) : kT \leq t < (k+1)T\}$ and N is the total number of CPM symbols. In addition to $\beta_{Ext}(k)$ arriving from the LDPC SISO decoder, we also obtain the *a posteriori* probability (APP) of x_k as $\beta(x_k|\mathbf{r}) = \log P(x_k = 1|\mathbf{r})/P(x_k = 0|\mathbf{r})$. After a number of iterations the receiver will then carry out detection based on the sign of $\beta(x_k|\mathbf{r})$.

Before we proceed to the next section, it should be pointed out that there are two types of iterations in the iterative LDPC-coded CPM receiver: *internal iterations* refers to the SPA iterations in the LDPC decoder, whereas *turbo iterations* refers to the iterations between the CPM detector and LDPC decoder (as shown in Fig. 1). Albeit a higher computational complexity is incurred compared to the non-iterative receiver, turbo decoding leads to significantly improved error rate performance and hence it has been widely used in many communication scenarios where a low BER is desired (e.g., satellite communication channels with low SNR).

III. Finite-Length Performance Estimation

It is widely recognized that the per-codeword BER fluctuates from codeword to codeword when the codeword length is limited [13]. By exploiting this observation and taking into account the properties of CPM, we now proceed to propose our estimation technique to predict the finite-length performance of LDPC coded binary CPM.

Since we consider unity signal power, the SNR E_s/N_0 solely depends on the standard deviation σ_n of the noise. For a given SNR E_s/N_0 and codeword length of N, the procedure to predict the BER of LDPC coded CPM is summarized in Algorithm 1, where the following notations are used for a sample value of noise standard deviation (s.d.) q:

- $z_q^{(\Gamma)}$ is the mean of LLR passed from the CPM to the LDPC decoder at *turbo iteration* Γ, i.e., $z_q^{(\Gamma)} = \mathbb{E}[\alpha(y_k)]$, where $\alpha(y_k)$ is shown in Fig. 1.
- $u_q^{(\Gamma)}$ is the mean of LLR passed from the LDPC decoder to the CPM at *turbo iteration* Γ, i.e., $u_q^{(\Gamma)} = \mathbb{E}[\beta(I_k)]$ [see Fig. 1 for $\beta(I_k)$].
- $V_q^{(\Gamma,\ell)}$ is the mean of LLR passed from the variable node to the check node in the LDPC decoder at *turbo iteration* Γ and *internal iteration* ℓ.
- $C_q^{(\Gamma,\ell)}$ is the mean of LLR passed from a check node to a variable node of the LDPC decoder during a *turbo iteration* Γ and *internal iteration* ℓ.
- $P_{b|q}$ is the asymptotic bit error probability returned by the LDPC coded iterative CPM detector.
- δ_q is the bit error probability of CPM when the noise s.d. is q.

Explanation of Algorithm 1: In Step 1, we first take samples of the noise standard deviation (s.d.) from a feasible range, so that the resultant SNR limits vary from very high to very low values[2]. For each sample, we find the corresponding analytical error probability of the CPM-LDPC decoder (Steps 2-6) for a predefined maximum number of turbo iterations. At each turbo iteration, we utilize the Gaussian approximated density

[2] Quantitatively, the samples of the noise standard deviation are obtained from the range 0 to 5 relying on a sampling interval of 0.01.

evolution[3] technique [16] for evaluating the performance of the LDPC decoding process, while assuming that the code is cycle free. Although the Gaussian approximated density evolution is a well established technique, there is no study in the open literature on how to carry out density evolution for LDPC coded CPM schemes, to the best of our knowledge. In Steps 3-6, we exploit the properties of the CPM-LDPC turbo-receiver in the density evolution update equations. To initialize the density evolution in Step 3, we set the initial mean LLR (equivalent to the channel output mean LLR used in the traditional density evolution) equal to the mean LLR value, which is a function of the sampled noise standard deviation and the previous mean LLR obtained from the density evolution (i.e., the mean LLR obtained from Step 5) using

$$z_q^{(\Gamma)} = 2Q^{-1}\left[P_{\text{CPM}}\left(u_q^{(\Gamma-1)}, q\right)\right].$$

Since deriving an analytical expression of the error probability $P_{\text{CPM}}(\cdot)$ is an open challenge, $P_{\text{CPM}}(\cdot)$ can be evaluated numerically (please see Fig. 4). For a predefined maximum number of LDPC decoding iterations and with the initial mean LLR, we iteratively update the density evolution equations associated with the variable nodes and the check nodes in Step 4. In Step 5, we calculate the LLR returned by the LDPC decoder (which is used in Step 3) by considering all the edges connected to the variable nodes. Once the maximum number of turbo iterations is reached, we find the conditional error probability (i.e., for a given sample of noise standard deviation assigned in Step 2) in Step 6 by using the tail distribution function of the normal distribution. To obtain the decoded error probability at the target SNR, we marginalize all the conditional error probabilities (obtained in Step 6) with appropriate distribution of the observed bit error probability. As mentioned earlier, the bit error probability observed varies from one codeword to another for a finite codeword length. The distribution of the bit error rate experienced in an AWGN channel may be readily approximated by $\mathcal{N}(p_c, p_c(1-p_c)/N)$ [13], where p_c is the probability of receiving a bit in error and N is the codeword length. Thus, to marginalize the conditional error probabilities, first we find p_c for a given SNR E_s/N_0 (i.e., for a given σ_n) in Step 7 and then perform marginalization in Step 8 to obtain the BER at a given SNR.

Note that in this algorithm, an all-zero codeword is assumed in the density evolution, while the performance of CPM is evaluated using random input sequences.

Algorithm 1 Finite-length performance estimation of LDPC coded CPM

```
1: Take samples of noise standard deviation $\sigma_n$ and repeat Steps 2-6 for each sample.
2: For a sample $q$, repeat Steps 3-6 for a predefined maximum number of turbo iteration $\Gamma_m$:
3: Find $z_q^t$ from $z_q^{(\Gamma)} = 2Q^{-1}\left(P_{\text{CPM}}\left(u_q^{(\Gamma-1)}, q\right)\right)$, where $Q^{-1}(\cdot)$ is the inverse Q-function with
       $Q(g) = \frac{1}{2\pi}\int_g^{\infty} \exp(-w^2/2)dw$
   and $P_{\text{CPM}}(\cdot)$ is a function that characterizes the SISO CPM demodulator and returns the error probability based on the observation gleaned from the channel with noise s.d. $q$ and mean of LLR obtained from the LDPC decoder. We initialize $u_q^{(\Gamma)}$ by $u_q^{(0)} = 0$.
4: For $\ell = \{1, 2, \ldots, \ell_{\text{m}}\}$, where $\ell_{\text{m}}$ is the maximum number of internal iterations, compute $V_q^{(\Gamma,\ell)}$ and $C_q^{(\Gamma,\ell)}$ by the following recursive equation:
       $V_q^{(\Gamma,\ell)} = \sum_{i=2}^{d_{v_{\text{max}}}} \lambda_i \big(z_q^{(\Gamma)} + (i-1)C_q^{(\Gamma,\ell-1)}\big),$
       $C_q^{(\Gamma,\ell)} = \sum_{j=2}^{d_{c_{\text{max}}}} \rho_j \phi^{-1}\left(1 - \left(1 - \sum_{i=2}^{d_{v_{\text{max}}}} \lambda_i \phi\Big(z_q^{(\Gamma)} + (i-1)C_q^{(\Gamma,\ell-1)}\Big)\right)^{j-1}\right),$
   where $y_q^{(\Gamma,0)} = 0$ and
       $\phi(g) = \begin{cases} 1 - \frac{1}{\sqrt{4\pi g}}\int_{-\infty}^{\infty}\big(\tanh\frac{f}{2}\big)e^{-\frac{(f-g)^2}{4g}}df, & \text{if } g > 0 \\ 1, & \text{if } g = 0 \end{cases}$
5: Find $u_q^{(\Gamma)}$ by
       $u_q^{(\Gamma)} = \phi^{-1}\left(\sum_{i=2}^{d_{v_{\text{max}}}} v_i \phi\Big(z_q^{(\Gamma)} + (i-1)C_q^{(\Gamma,\ell_m)}\Big)\right).$
6: Compute the asymptotic bit error probability $P_{b|q}$ by
       $P_{b|q} = \sum_{i=2}^{d_{v_{\text{max}}}} v_i Q\left(\sqrt{\frac{z_q^{(\Gamma_m)} + (i-1)C_q^{(\Gamma_m,\ell_m)}}{2}}\right),$
   where $v_i = \frac{\lambda_i/i}{\sum_j \lambda_j/j}$.
7: With realized noise variance $\sigma_n^2$, find $p_c = P_{\text{CPM}}(\sigma_n, 0)$.
8: Compute the bit error probability $P_b = \sum_q P_r(\delta_q)P_{b|q}$, where $\delta_q = P_{\text{CPM}}(q, 0)$. $P_r(\delta_q)$ is obtained from the quantized approximation of the Gaussian distribution $\mathcal{N}(p_c, p_c(1-p_c)/N)$.
```

IV. Validation of Estimated Results

Let us now compare the results obtained from the proposed estimation technique to the simulated results. We set the codeword lengths to $N = \{500, 2000, 10000\}$. All the codes were constructed using the progressive edge growth (PEG) algorithm [17] and the girths were $4, 6,$ and 8 for the codeword lengths of $500, 2000,$ and 10000, respectively. The corresponding simulated and estimated results are compared in Fig. 2 for regular LDPC codes. We show the results for the LDPC codes ($(2,4)$ and $(3,6)$) having a maximum of $\ell_m = 20$ internal iterations and $\Gamma_m = 10$ turbo iterations. The binary CPM with $h = 1/2$ and $L = 1$ (also called CFPSK) is considered. Note that all codes are randomly constructed and the

[3] Since we aim for predicting the performance of finite-length LDPC coded CPM schemes in the waterfall region (the region which is independent of cycles in LDPC code), we consider the density evolution technique in the proposed algorithm. Briefly, density evolution constitutes a popular analytical tool of analysing the performance of iterative decoding of a particular code ensemble assuming that the code ensemble is cycle-free. Density evolution tracks the evolution of probability density functions (pdfs) of the messages (that are exchanged between the variable and the check nodes) throughout the decoding process and characterizes the decoding behaviour for a given code ensemble. The rationale behind choosing the Gaussian approximated density evolution technique is that it allows the tracking of the mean of LLR in the evolution process instead of tracking the actual pdfs, and hence reduces the computational complexity.

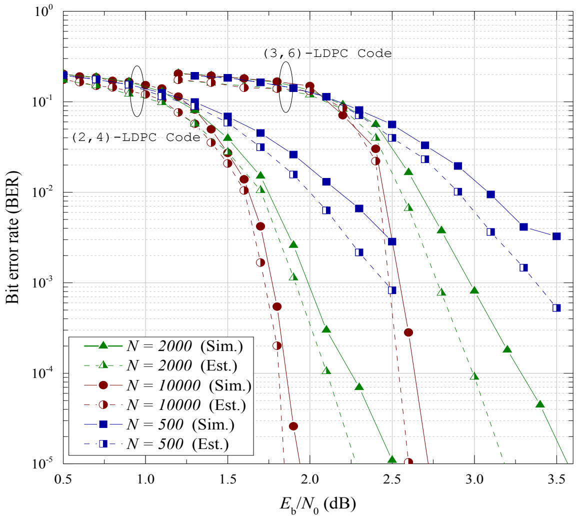

Fig. 2. Simulated and estimated performance comparison for $(2, 4)$ and $(3, 6)$-regular LDPC coded CPFSK system.

results are averaged over 5000 transmission frames. Observe the close match between the estimated and simulated results in the waterfall region. However, the estimation results start to deviate from the simulation results when the performance curves approache the error floor region. However, estimation of the error-floor region (caused by having short cycles) is beyond the scope of this paper, which we might consider in our future research. We also notice that the gap between the estimated and simulated results reduces as the codeword length increases. For example, we observe a gap of 0.2 dB at a BER of 10^{-4} for the $(2, 4)$-LDPC codes associated with $N = 2000$, while the gap is reduced to less than 0.1 dB for $N = 10000$. This is due to the fact that upon increasing the codeword length, the number of short cycles is reduced. Note that for an LDPC-coded PSK scheme, the $(3, 6)$-LDPC code performs better than other rate-$1/2$ regular LDPC codes. By contrast, $(2, 4)$-LDPC codes outperform all other rate-$1/2$ regular LDPC codes concatenate with CPM schemes. In Fig. 3, we compare the simulated and estimated performance of the irregular LDPC codes presented in [18], while considering the GSM GMSK scheme of $(L = 3, BT = 0.3, h = 1/2)$. We show our performance comparison for the rate $R = 0.75$ (degree distribution pairs $\lambda(x) = 0.496x + 0.083x^{11} + 0.198x^{12} + 0.223x^{49}$ and $\rho(x) = 0.4x^{13} + 0.6x^{14}$) and rate $R = 0.333$ (degree distribution pairs $\lambda(x) = 0.597x + 0.142x^8 + 0.085x^9 + 0.176x^{49}$ and $\rho(x) = 0.35x^3 + 0.65x^4$). For two different codeword lengths, the simulated and estimated performances are compared. Similar to the regular LDPC codes, a close match is observed between the simulated and estimated performances. Our experiments show that the gap between the estimated and simulated performance of LDPC coded CPM schemes is slightly wider than for the LDPC-coded BPSK scheme [13]. This may be because the error probability of CPM is influenced by many factors (e.g., modulation index, frequency pulse shape and length, and the non-linearity of CPM) which makes the performance prediction less tractable. Additionally, for low variable node degrees, the Gaussian approximation slightly loses its accuracy, and hence the prediction performance is less accurate for such cases. Finally, Fig. 4 presents our numerical simulations of P_{CPM} (used in Step 3 of Algorithm 1), which is a function of the mean LLR and of the noise s.d. $q \in \{0.75, 1.00, 1.25, 1.50, 1.75, 2.00\}$ for both CPFSK and GMSK. Note that deriving a closed-form expression for P_{CPM} is challenging, since it depends on numerous parameters, such as the minimum Euclidean distance of CPM signals, the number of turbo iterations, the interleaver size and the outer code used (LDPC code in this paper). It is found that for a fixed value of q, the P_{CPM} in the function of the "mean of LLR" for GMSK exhibits similar behavior to that of CPFSK. Interestingly, when the "mean of LLR" is higher than 3, the P_{CPM} of GMSK is slightly better than that of CPFSK.

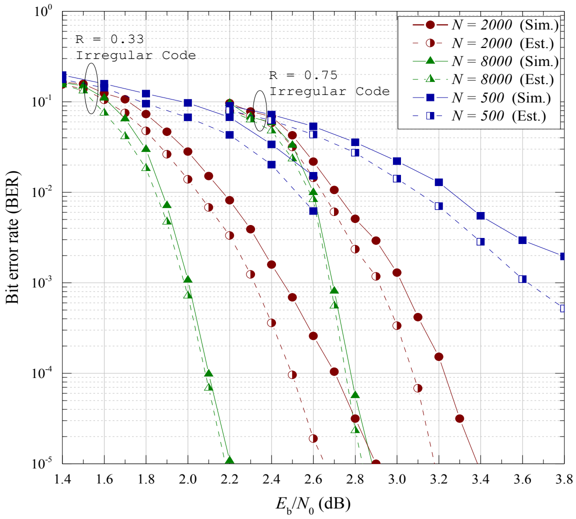

Fig. 3. Simulated and estimated performance comparison for irregular LDPC ([18]) coded GMSK system.

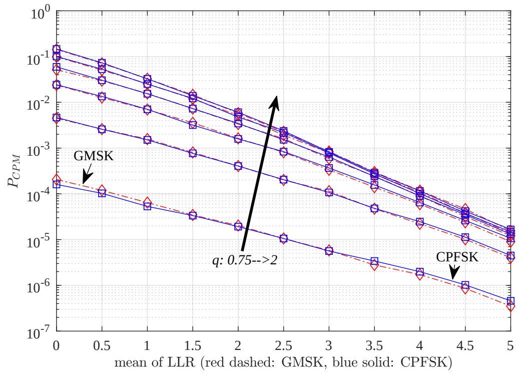

Fig. 4. Numerical evaluation of P_{CPM} as a function of mean of LLR and noise s.d. q for both CPFSK and GMSK.

V. Summary and Conclusions

An LDPC-coded and turbo-detected CPM scheme was investigated. Explicitly, we have proposed an estimation algorithm for predicting the finite-length performance of LDPC coded binary CPM schemes in the waterfall region. The results obtained by our estimation technique have been confirmed by the simulated results for different codes and different codeword lengths. Our numerical results have demonstrated that the proposed estimation technique accurately predicts the finite-length performance of LDPC coded CPM schemes in the waterfall region. We have also observed that the gap between the estimation and simulation results reduces upon increasing the codeword length.

References

[1] J. B. Anderson, T. Aulin, and C. E. Sundberg, *Digital Phase Modulation*, New York: Plenum, 1986.

[2] Z. Liu, Y. L. Guan, and C.-C. Chui, "CPM training waveform with autocorrelation sidelobes close to zero," *IEEE Trans. Veh. Technol.* , vol. 67, no. 11, pp. 11269-11273, Nov. 2018.

[3] A. Emmanuele, A. Zanier, F. Boccolini, et al., "Spread-spectrum continuous-phase-modulated signals for satellite navigation," *IEEE Trans. Aeros. Electro. Systems*, vol. 48, no. 4, pp. 3234-3249, Oct. 2012.

[4] R.C. Daniels and R.W. Heath, Jr., "60 GHz wireless communications: emerging requirements and design recommendations," *IEEE Veh. Technol. Mag.*, vol. 2, no. 3, pp. 41-50, Sept. 2007.

[5] C. Brutel, J. Boutrost and F. Belvbze, "Serial encoding and iterative detection of continuous phase modulations", *1999 IEEE Global Telecommun. Conf.*, pp. 2375-2379, Dec. 1999.

[6] A. G. i Amat, C. A. Nour, and C. Douillard, "Serially concatenated continuous phase modulation for satellite communications," *IEEE Trans. Wireless Commun.*, vol. 8, no. 6, pp. 3260-3269, Jun. 2009.

[7] K. R. Narayanan, I. Altunbas, and R. S. Narayanaswami, "Design of serial concatenated MSK schemes based on density evolution," *IEEE Trans. Commun.*, vol. 51, no. 8, pp. 1283-1295, Aug 2003.

[8] M. Xiao and T. Aulin, "Irregular repeat continuous phase modulation," *IEEE Commun. Letters*, vol. 9, no. 8, pp. 723-725, Aug 2005.

[9] T. Benaddi, C. Poulliat, M. L. Boucheret, B. Gadat, and G. Lesthievent, "Design of unstructured and protograph-based LDPC coded continuous phase modulation," in *2014 IEEE International Symposium on Information Theory*, June 2014, pp. 1982-1986.

[10] S. ten Brink, "Convergence of iterative decoding," *Electronics Letters*, vol. 35, no. 10, pp. 806-808, May 1999.

[11] G. Yue, B. Lu, and X. Wang, "Analysis and design of finite-length LDPC codes," *IEEE Trans. Veh. Technol.* , vol. 56, no. 3, pp. 1321-1332, May 2007.

[12] Z. Mei, K. Cai, and G. Song, "Performance analysis of finite-length LDPC codes over asymmetric memoryless channels," *IEEE Trans. Veh. Technol.* , vol. 68, no. 11, pp. 11338-11342, Nov. 2019.

[13] R. Yazdani and M. Ardakani, "Waterfall performance analysis of finite-length LDPC codes on symmetric channels," *IEEE Trans. Comm.*, vol. 57, no. 11, pp. 3183-3187, nov. 2009.

[14] M. Noor-A-Rahim, K. D. Nguyen, and G. Lechner, "Finite length analysis of LDPC codes," in *Proc. IEEE Wireless Communications and Networking Conference*, Istanbul, Turkey, Apr. 2014.

[15] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, pp. 284-287, March 1974.

[16] S. Y. Chung, T. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 657 -670, Feb. 2001.

[17] T. Richardson and R. Urbanke,"Modern Coding Theory," *ambridge University Press*, New York, NY, USA, 2008.

[18] T. Benaddi, "Sparse graph-based coding schemes for continuous phase modulations," *Ph.D. dissertation*, University of Toulouse, 2015.