SOFT LDPC DECODING IN NONLINEAR CHANNELS WITH GAUSSIAN PROCESSES FOR CLASSIFICATION

Pablo Martínez-Olmos, Juan José Murillo-Fuentes and Fernando Pérez-Cruz

Dept. Teoría de la Señal y Comunicaciones,Universidad de Sevilla
Escuela Superior de Ingenieros, Paseo de los Descubrimientos s/n, 41092, Sevilla
E-mail: {olmos,murillo}@us.es

Electrical Engineering Department, Princeton University
Princeton (NJ)
E-mail: fp@princeton.edu

ABSTRACT

In this paper, we propose a new approach for nonlinear equalization based on Gaussian processes for classification (GPC). We also measure the performance of the equalizer after a low-density parity-check channel decoder has detected the received sequence. Typically, most channel equalizers concentrate on reducing the bit error rate, instead of providing accurate posterior probability estimates. GPC is a Bayesian nonlinear classification tool that provides accurate posterior probability estimates with short training sequences. We show that the accuracy of these estimates is essential for optimal performance of the channel decoder and that the error rate outputted by the equalizer might be irrelevant to understand the performance of the overall communication receiver. We compare the proposed equalizers with state-of-the-art solutions.

1. INTRODUCTION

In wireless communications systems, efficient use of the available spectrum is one of most critical design issues. Therefore, modern communication systems must evolve to work as close as possible to channel capacity to achieve the demanded binary rates. We need to design digital communication systems that implement novel approaches for both channel equalization and coding and, moreover, we should be able to link them together to optimally detect the transmitted information.

Communication channels introduce linear and nonlinear distortions and, in most cases of interest, they cannot be considered memoryless. Inter-symbol interference (ISI), mainly a consequence of multipath in wireless channels [1], accounts for the linear distortion. The presence of amplifiers and converters explain the nonlinear nature of communications channels [2]. Communication channels also contaminate the received sequence with random fluctuations, which are typically regarded as additive white Gaussian noise (AWGN) fluctuations [1].

In the design of digital communication receivers the equalizer precedes the channel decoder. In this paper, we focus on one-shot receivers, where the equalizer deals with the dispersive nature of the channel and delivers a memoryless sequence to the channel decoder. The channel decoder corrects the errors in the received sequence using the controlled redundancy introduced at the transmitter. Alternatively, there is a class of iterative receivers that iterate between the equalization and decoding tasks. In most studies, see [3]-[9] and the references therein, the dispersive nature of the channel and the equalizer are analyzed independently from the channel decoder. Moreover their performance gains are typically measured at very low bit error rate (BER), as if there were no channel decoder. One of the goals of this paper is the analysis of state-of-the-art nonlinear equalizers together with the channel decoder. We show that this analysis is not only beneficial to understand the properties of the equalizers, but it is also meaningless to compare the equalizers performance at very low BER.

We employ low-density parity-check (LDPC) codes [10] to add redundancy to the transmitted binary sequence. LDPC codes have recently attracted a great research interest, because of their excellent error-correcting performance and linear complexity. The Digital Video Broadcasting standard uses LDPC codes for protecting the transmitted sequence and they are being considered in various applications such as 10Gb Ethernet and high-throughput wireless local area networks [11].

For linear channels, the equalizers based on the Viterbi algorithm [12] minimize the probability of returning the incorrect sequence to the channel decoder, and they are known as maximum likelihood sequence equalizers (MLSEs). The subsequent channel decoder must treat the output of the MLSE as a binary symmetric channel, because it has no information about which bits could be in fault. Instead, we could use the BCJR algorithm [13] to design our equalizer. The BCJR algorithm returns the posterior probability (given the received sequence) for each bit, but it does not minimize the probability of returning an incorrect sequence as the Viterbi algorithm does. Nevertheless the BCJR algorithm provides a probabilistic output for each bit that can be exploited by the LPDC decoder to significantly reduce its error rate, because it has individual information about which bits might be in error. Thereby, the subsequent channel decoding stage substantially affects the way we measure the performance of our equalizer.

For nonlinear channels the computational complexity of the BCJR and the Viterbi algorithms grows exponentially with the number of transmitted bits and they require perfect knowledge of the channel state information. Neural networks and, recently, machine-learning approaches have been

This work was partially funded by Spanish government (Ministerio de Ciencia e Innovacin TEC2006-13514-C02-01/TCM y TEC2006-13514-C02-02/TCM) and Consolider-Ingenio 2010 CSD2008-00010), and the European Union (FEDER).

proposed to approximate these equalizers at a lower computational complexity and they can be readily adapted for nonlinear channels. An illustrative and non-exhaustive list of examples for nonlinear equalizers are: radial basis functions (RBFs) [4]; recurrent RBFs [5]; wavelet neural networks [6]; kernel adaline [2]; support vector machines [7]; self-constructing recurrent fuzzy neural network [8]; and, Gaussian processes for regression [9]. But, as mentioned earlier, these approaches only compare performance at low BER without considering the channel decoder.

The aforementioned equalizers are designed to minimize their BER by undoing the effect of the channel: multipath and nonlinearities. But their soft-outputs cannot be directly interpreted as posterior probability estimates, which significantly limit the performance of soft-inputs channel decoders, such as LDPC codes. In this paper, we proposed a channel equalizer based on Gaussian processes for classification (GPC). GPC are Bayesian machine-learning tools that assign accurate posterior probability estimates to its binary decisions, as the BCJR algorithm does for linear channels. GPC can equalize linear and nonlinear channels using a training sequence to adjust its parameters and it does not need to know a priori the channel estate information.

In a previous paper [9], we have shown that equalizers based on GPC are competitive with state-of-the-art solutions, when we compare their performance at low bit error rate. In this paper, we focus on their performance after the sequence has been corrected by an LDPC code. The ability of GPC to provide accurate posterior probability predictions boosts the performance of these equalizers compare to the state-of-the-art solutions, based on support vector machines (SVMs). SVM does not provide posterior probability estimates and their soft-outputs can be transformed into posterior probabilities using Platt's method [14].

The rest of the paper is organized as follows. Section 2 is devoted to presenting Gaussian processes for classification. We describe the proposed receiver scheme in Section 3 together with the channel model and the transmitter. In Section 4, we include illustrative computer experiments to show the performances of the proposed equalizers. We conclude in Section 5 with some final comments.

2. GAUSSIAN PROCESSES FOR CLASSIFICATION

Gaussian processes for classification is a Bayesian supervised machine learning tool for predicting the posterior probability of the output (b_*) given an input ($\mathbf{x}_*$) and a training set $\mathscr{D} = \{\mathbf{x}_i, b_i\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^d$ and $b_i \in \{0,1\}$), i.e.

$$p(b_*|\mathbf{x}_*, \mathscr{D}). \tag{1}$$

GPC assumes that a real-valued function, known as *latent function*, underlies the classification problem and that this function follows a Gaussian process. We assume this latent function has been drawn from a zero-mean Gaussian process prior with its covariance function denoted by $k(\mathbf{x}, \mathbf{x}')$. The covariance function describes the relations between each pair of inputs and characterizes the functions that can be described by the Gaussian process.

For any finite set of samples, a Gaussian process becomes a multidimensional Gaussian defined by its mean (zero in our case) and covariance matrix. Our Gaussian process prior becomes:

$$p(\mathbf{f}|\mathbf{X}) = \mathscr{N}(0, \mathbf{K}), \tag{2}$$

where $\mathbf{f} = [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)]^\top$, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ and $(\mathbf{K})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j), \forall\, \mathbf{x}_i, \mathbf{x}_j \in \mathscr{D}$.

Once the labels are revealed, $\mathbf{b} = [b_1, b_2, \ldots, b_n]^\top$, together with the location of the test point, $\mathbf{x}_*$, we can compute (1) using the standard tools of Bayesian statistics: Bayes rule, marginalization and conditioning.

We first apply Bayes rule to obtain the posterior density for the latent function:

$$p(\mathbf{f}, f(\mathbf{x}_*)|\mathscr{D}, \mathbf{x}_*) = \frac{p(\mathbf{b}|\mathbf{f}, \mathbf{X}) p(\mathbf{f}, f(\mathbf{x}_*)|\mathbf{X}, \mathbf{x}_*)}{p(\mathbf{b}|\mathbf{X})}, \tag{3}$$

where $\mathscr{D} = (\mathbf{X}, \mathbf{b})$, $p(\mathbf{f}, f(\mathbf{x}_*)|\mathbf{X}, \mathbf{x}_*)$ is the Gaussian process prior in (2) extended with the test input, $p(\mathbf{b}|\mathbf{f}, \mathbf{X})$ is the likelihood for the latent function at the training set, and $p(\mathbf{b}|\mathbf{X})$ is the evidence of the model, also known as the partition function, which guarantees that the posterior is a proper probability density function. A factorized model is used for the likelihood function:

$$p(\mathbf{b}|\mathbf{f}, \mathbf{X}) = \prod_{i=1}^{n} p(b_i|f(\mathbf{x}_i), \mathbf{x}_i), \tag{4}$$

because the training samples have been obtained independently and identically distributed (iid). The likelihood for the latent function at $\mathbf{x}_i$ is obtained using a *response function* $\Phi(\cdot)$:

$$p(b_i = 1|f(\mathbf{x}_i), \mathbf{x}_i) = \Phi(f(\mathbf{x}_i)). \tag{5}$$

The response function "squashes" the real-valued latent function to an $(0,1)$-interval that represents the posterior probability for b_i [15]. Standard choices for the response function are the logistic and the probit.

We can obtain the posterior density for the test point by conditioning on the training set and $\mathbf{x}_*$ and by marginalizing the latent function:

$$p(b_*|\mathbf{x}_*, \mathscr{D}) = \int p(b_i|f(\mathbf{x}_*), \mathbf{x}_*) p(f(\mathbf{x}_*)|\mathscr{D}, \mathbf{x}_*) df(\mathbf{x}_*), \tag{6}$$

where

$$p(f(\mathbf{x}_*)|\mathscr{D}, \mathbf{x}_*) = \int p(\mathbf{f}, f(\mathbf{x}_*)|\mathscr{D}, \mathbf{x}_*) d\mathbf{f}. \tag{7}$$

is the integral of the posterior in (3). We divide the marginalization in two separate equations to show the marginalization of the latent function at the training set in (7) and the marginalization of the latent function at the test point in (6).

The integrals in (6) and (7) are analytically intractable, because the likelihood and the prior are not conjugated. Therefore, we have to resort to numerical methods or approximations to solve them. The posterior distribution in (3) is typically single-mode and the standard methods approximate it with a Gaussian [15]. Using a Gaussian approximation for (3) allows exact marginalization in (7) and we can use numerical integration for solving (6), as it involves marginalizing a single real-valued quantity. The two standard approximations for the posterior of the latent function are the Laplace method or expectation propagation (EP) [15]. In [16], EP is shown to be a more accurate approximation for the posterior probabilities $p(b_*|\mathbf{x}_*, \mathscr{D})$ and we use it throughout our implementation. GPC with EP approximation requires $n^3/(6+n^2)$ operations once and n^2 per test case [15].

In this section, we have assumed that the kernel function is known. If the kernel function is unknown, it has to be inferred from the training samples. Typically a parametric model is assumed over the kernel function and a maximum likelihood procedure is used to infer the parameters of the kernel, also known as *hyperparameters* to distinguish them from the latent function. A versatile covariance function that we have previously proposed to solve channel equalization problems is described by:

$$k(\mathbf{x}_i,\mathbf{x}_j) = \alpha_1 \exp\left(-\gamma \sum_{\ell=1}^{d} (\mathbf{x}_{i\ell} - \mathbf{x}_{j\ell})^2\right) + \alpha_2 \mathbf{x}_i^T \mathbf{x}_j^T + \alpha_3 \delta_{ij}, \tag{8}$$

where $\theta = [\alpha_1, \alpha_2, \alpha_3, \gamma]$ are the hyperparameters. The suitability of this covariance function for the channel equalization problem has been discussed in detail in [9]. Further details about the inference of the hyperparameters can be found in [15] for general covariance functions.

3. COMMUNICATION SYSTEM

In this paper we propose the use of GPC for channel equalization followed by an LDPC channel decoder. In Fig. 1 we depict a discrete-time digital-communication system with a nonlinear dispersive communication channel. We transmit independent and equiprobable binary symbols $s[j] \in \{\pm 1\}$, which are encoded into a binary sequence $b[j]$ using an LDPC code. The time-invariant impulse response of the channel is given by:

$$\mathrm{h}(z) = \sum_{\ell=0}^{n_L-1} h[\ell] z^{-\ell}, \tag{9}$$

where n_L denotes the channel length. The nonlinearities in the channel, mainly due to amplifiers and mixers, are modeled by $g(\cdot)$, as proposed in [2]. Hence, the output of the communication channel is given by:

$$x[j] = g(r[j]) + w[j] = g\left(\sum_{\ell=0}^{n_L-1} b[j-\ell]h[\ell]\right) + w[j], \tag{10}$$

where $w[j]$ represents independent samples of AWGN. The receiver equalizes the nonlinear channel and decodes the received sequence.

First, the equalizer collects a set of m consecutive received symbols:

$$\mathbf{x}_j = [x[j+\tau], x[j+\tau-1], \ldots, x[j+\tau-m+1]]^\top, \tag{11}$$

to predict each encoded bit,

$$b_j = b[j-\tau], \tag{12}$$

where m and τ represents, respectively, the order and delay of the equalizer.

Second, the channel decoder removes the errors using the added redundancy by the LDPC encoder in the transmitted sequence. The decoder performance improves if the equalizer provides a probability estimate for each detected bit, instead of a hard (point-wise) decision. Optimally, we desire the posterior probability for each bit given the complete received sequence, i.e. $p(b_j|x[1],\ldots,x[N])\ \forall j = \{1,\ldots,N\}$. The BCJR algorithm computes this posterior probability, when the channel is linear and perfectly known at the receiver. For nonlinear channels, the computational complexity of this posterior probability grows exponentially with the number of transmitted bits and, in most cases of interest, it cannot be computed analytically.

In this paper, we estimate this posterior probability using $\mathbf{x}_j$:

$$p(b_j|x[1],\ldots,x[N]) \approx p(b_j|\mathbf{x}_j,\mathscr{D}), \quad \forall j = 1,\ldots,N, \tag{13}$$

and a GPC, detailed in the previous section. The GPC is trained with a set $\mathscr{D}$, independent from the test samples. We experimentally show its performance is close to the BCJR algorithm for linear communication channels and it outperforms other nonlinear equalizers for linear and nonlinear communication channels. We compare its performance with an SVM equalizer, using Platt's method [14] to obtain posterior probability estimates. SVMs are state-of-the-art machine learning tool for classification [17]. SVM with Platt's method has complexity $\mathscr{O}(n^2)$ [17]. For the GPC equalizer, we use the kernel proposed in Section 2 and a Gaussian kernel for the SVM [18]. For the SVM we use a simpler kernel, because (9) has far too many parameters that cannot be set by cross-validation. This topic was discussed in detail in [19]. We also use the BCJR algorithm as the optimal baseline performance for the linear channel. For the nonlinear channel, the BCJR complexity makes it impractical. We refer to GPC-EQ, SVM-EQ and BCJR-EQ when analyzing the equalizers with no decoding stage and use GPC-LDPC, SVM-LDPC and BCJR-LDPC to denote the joint equalizer and channel decoder using the GPC, SVM-Platt and BCJR approaches respectively.

4. EXPERIMENTAL RESULTS

In this section, we illustrate the performance of the proposed joint equalizer and channel decoder. We use a rate $1/2$ regular LPDC code with 1000 bits per codeword and 3 ones per column and we have selected a single channel model for our three experimental settings:

$$\mathrm{h}(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}. \tag{14}$$

This channel was proposed in [2] for modeling radiocommunication channels. In all three experiments, we use 200 training samples and a four-tap equalizer ($m = 4$). The reported bit error rate (BER) and frame error rate (FER) are computed using 10^5 test codewords and we average the results over 50 trials, where random training and test data are generated independently.

4.1 Experiment 1: BPSK over linear multipath channel

In this first experiment we deal with the linear channel in (14) and we compare the GPC and the SVM methods to the BCJR algorithm with perfect channel estate information at the receiver. The performance of the BCJR-LDPC is the optimal solution to this problem.

In Fig. 2 we plot (dash-dotted) the BER outputted by the equalizers. The GPC-EQ ($\triangledown$) and SVM-EQ ($\circ$) BER plots in Fig. 2 are almost identical and they perform slightly worse than the BCJR-EQ ($\diamond$). These results are similar to the ones reported in [9]. The BER at the outputs of the channel decoders is plotted in solid line. We can observe that, although

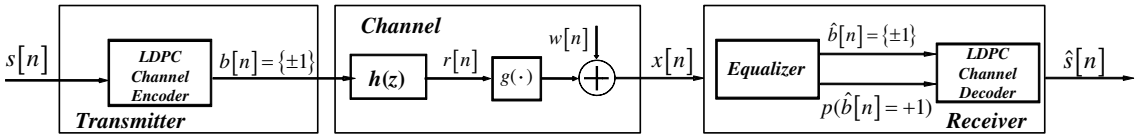


Figure 1: Discrete-time channel model, together with the proposed transmitter and receiver.

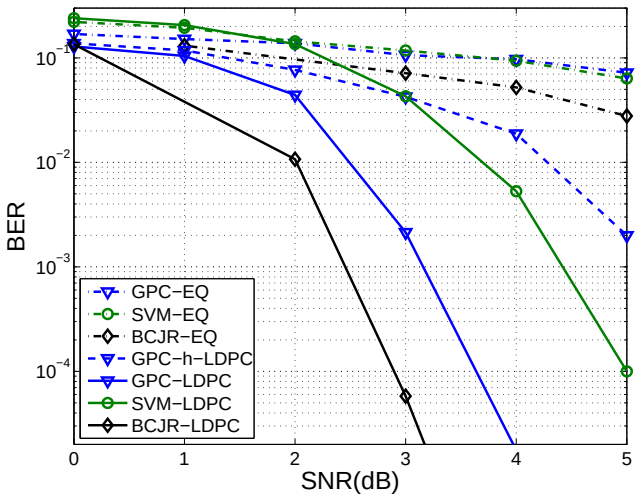
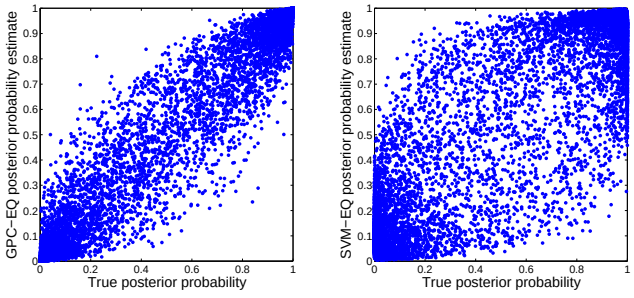


Figure 2: BER performance at the output of the equalizers (dash-dotted lines) and the channel decoder with soft inputs (solid lines) and hard inputs (dashed line) with the channel model in (14) for the BCJR (⋄), the GPC (▽) and the SVM (○).

the decisions provided by the GPC-EQ and SVM-EQ are similar to each other, their estimate of the posterior probability are quite different. Therefore, the GPC-LDPC significantly reduces the BER at lower SNR, because GPC-EQ posterior probability estimates are more accurate and the LDPC decoder can rely on these trustworthy predictions. Moreover, the GPC-LDPC is only 1dB from the optimal performance achieved by the BCJR-LDPC receiver. The SVM-LDPC performance is substantially worse and its solution is almost 2dB away from the optimal performance.

In Fig. 2 we have also included the BER performance of the GPC-h-LDPC, whose inputs are the hard decisions given by the GPC-EQ. Its BER is depicted as a dashed line in the plot. For this receiver, the LDPC does not have information about which bits might be in error and it has to treat each bit with equal suspicion. The BER performance of SVM-h-LDPC is similar to GPC-h-LDPC and it is not shown for the sake of clarity. Even though the posterior probability estimates do not match the BCJR outputs, our soft-output equalizers provide much lower BER than a hard decision equalizer.

To understand the difference in posterior probability estimates, we have plotted calibration curves for the GPC-EQ and SVM-EQ, respectively, in Fig. 3(a) and Fig. 3(b) for SNR= 2dB. We can appreciate that the GPC-EQ posterior probability estimates are closer to the main diagonal and they are less spread. Thereby GPC-EQ estimates are closer to the true posterior probability, which explains its improved performance with respect to the SVM-EQ, when we measure the BER after the LDPC decoder.



(a) (b)

Figure 3: GPC-EQ and SVM-EQ calibration curves, respectively in (a) and (b) for SNR= 2dB.

We have shown that GPC-LDPC is far superior to the other schemes and its performance is close to optimal. This result shows that using a method that can predict accurately the posterior probability estimates allows the LDPC decoding algorithm to perform to its fullest. From this first experiment, it is clear that we need to compare the equalizers performance after the channel decoder, otherwise the BER measured after the equalizers does not tell the whole story. Also, we do not need to compare the equalizers at low BER, because the channel decoder reduces the BER significantly.

4.2 Experiment 2: Nonlinear multipath channel

In the next experiment, we face nonlinear multipath channels. We assume the nonlinearities in the channel are unknown at the receiver and transmitter, and we need a nonlinear equalizer, which is able to compensate the nonlinearities in the channel. For this experiment we employ the channel model proposed in [2, 8]:

$$|g(r)| \quad = |r| + 0.2|r|^2 - 0.1|r|^3. \tag{15}$$

This model represents an amplifier working in saturation with 0dB of back-off, which distorts the amplitude of our modulated BPSK signal.

In Fig. 4 we compare the BER performance of the GPC-LDPC (▽) with the SVM-LDPC (○). We also plot for completeness the BER after the equalizer with dash-dotted lines for the compared receivers. For this channel model the BCJR complexity is exponential in the length of the received sequence and we did not include it.

The equalizers perform equally well, while the BER by the GPC-LDC is significantly lower than that of SVM-LDPC. Again, the ability of the GPC to return accurate posterior probability estimates notably improves the performance of the channel decoder that can trust this posterior probability estimates and reduce the BER at lower SNR. In this

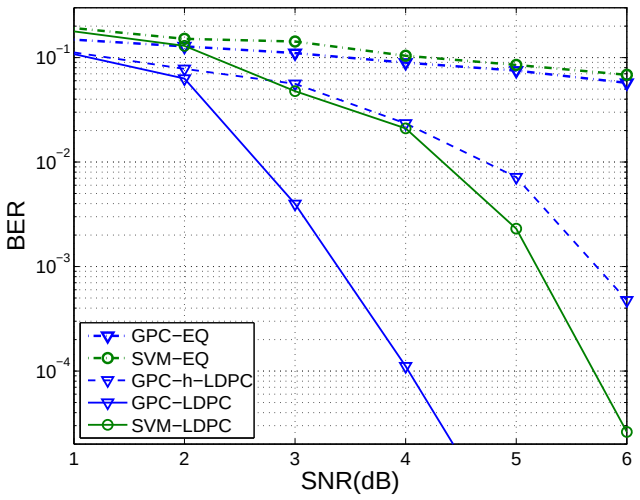


Figure 4: BER performance at the output of the equalizers (dash-dotted lines) and the channel decoder with soft inputs (solid lines) and hard inputs (dashed line) with the channel model in (14) and nonlinearity in (15) for the BCJR ($\diamond$), the GPC ($\triangledown$) and the SVM ($\circ$).

example, the coding gain is about 2dB between the GPC-LDPC and the SVM-LPDC. Also the BER of the LDPC with hard outputs (GPC-h-LPDC) is higher than the soft-outputs receivers. This result is relevant because, even though our posterior probability estimates are not accurate, we are better off with them than without.

5. CONCLUSIONS

The probabilistic nonlinear channel equalization is an open problem, since the standard solutions such as the nonlinear BCJR exhibit exponential complexity with the length of the received sequence. Moreover, they need an estimation of the nonlinear channel and they only approximate the optimal solution [20]. In this paper, we propose GPC to solve this long-standing problem. GPC is a Bayesian nonlinear probabilistic classifier that produces accurate posterior probability estimates. We compare the performance of the different probabilistic equalizers at the output of an LDPC channel decoder. We have shown the GPC outperforms the SVM with probabilistic output, which is a state-of-the-art nonlinear classifier.

Finally, as a by-product, we have shown that we need to measure the performance of the equalizers after the channel decoder. The equalizers' performance is typically measured at low BER without considering the channel decoder. But if we do not incorporate the channel decoder the BER outputted by the equalizer might not give an accurate picture of its performance. Furthermore, the equalizer performance at low BER might not be illustrative, as the channel decoder will significantly reduce it even for low signal to noise ratio values.

REFERENCES

[1] J. G. Proakis, *Digital Communications*, 4th ed. New York, NY: McGraw-Hill, 2000.

[2] B. Mitchinson and R. F. Harrison, "Digital communications channel equalization using the kernel adaline," *IEEE Transactions on Communications*, vol. 50, no. 4, pp. 571–576, 2002.

[3] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Adaptative equalization of finite non-linear channels using multilayer perceptrons," *Signal Processing*, vol. 10, pp. 107–119, 1990.

[4] ——, "Reconstruction of binary signals using an adaptive radial-basis function equalizer," *Signal Processing*, vol. 22, pp. 77–93, 1991.

[5] J. Cid-Sueiro, A. Artés-Rodríguez, and A. Figueiras-Vidal, "Recurrent radial basis function networks for optimal symbol-by-symbol equalization," *Signal Processing*, vol. 40, 1994.

[6] P. Chang and B. Wang, "Adaptive decision feedback equalization for digital satellite channels using multilayer neural networks," *IEEE Journal on Selected Areas on Communications*, vol. 13, 1995.

[7] F. Pérez-Cruz, A. Navia-Vázquez, P. L. Alarcón-Diana, and A. Artés-Rodríguez, "SVC-based equalizer for burst TDMA transmissions," *Signal Processing*, vol. 81, no. 8, pp. 1681–1693, Aug. 2001.

[8] R. Lin, W. Weng, and C. Hsueh, "Design of an SCRFNN-based nonlinear channel equaliser," *IEEE Proceedings on Communications*, vol. 1552, pp. 771–779, Dec. 2005.

[9] F. Pérez-Cruz, J. Murillo-Fuentes, and S. Caro, "Nonlinear channel equalization with gaussian processes for regression," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 5283–5286, Oct. 2008.

[10] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[11] H. Zhong and T. Zhang, "Block-LDPC: A practical LDPC coding system design approach," *IEEE Transactions on Circuits and Systems*, vol. 52, no. 4, 2005.

[12] G. Forney, "The Viterbi algorithm," *IEEE Proceedings*, vol. 61, no. 2, pp. 268–278, Mar. 1973.

[13] F. J. L. R. Bahl, J. Cocke and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[14] J. C. Platt, "Probabilities for SV machines," in *Advances in Large Margin Classifiers*, A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, (MA): M.I.T. Press, 2000, pp. 61–73.

[15] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press, 2006.

[16] M. Kuss and C. Rasmussen, "Assessing approximate inference for binary gaussian process classification," *Machine learning research*, vol. 6, pp. 1679–1704, Oct. 2005.

[17] B. Schölkopf and A. Smola, *Learning with kernels*. M.I.T. Press, 2001.

[18] F. Pérez-Cruz and O. Bousquet, "Kernel methods and their potential use in signal processing," *Signal Processing Magazine*, vol. 21, no. 3, pp. 57–65, 2004.

[19] F. Pérez-Cruz and J. Murillo-Fuentes, "Digital communication receivers using gaussian processes for machine learning," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, 2008.

[20] M. Mesiya, P. McLane, and L. Campbell, "Maximum likelihood sequence estimation of binary sequences transmitted over bandlimited nonlinear channels," *IEEE Transactions on Communications*, vol. 25, pp. 633–643, April 1977.