

Research Article

Nonlinear Demodulation and Channel Coding in EBPSK Scheme

Xianqing Chen and Lenan Wu

School of Information Science and Engineering, University of Southeast, 2 Sipailou, Nanjing 210096, China

Correspondence should be addressed to Xianqing Chen, xqchen213@126.com

Received 18 September 2012; Accepted 30 October 2012

Academic Editors: P. Colantonio, J. Dauwels, A. Ruano, and S. Sinha

Copyright © 2012 X. Chen and L. Wu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The extended binary phase shift keying (EBPSK) is an efficient modulation technique, and a special impacting filter (SIF) is used in its demodulator to improve the bit error rate (BER) performance. However, the conventional threshold decision cannot achieve the optimum performance, and the SIF brings more difficulty in obtaining the posterior probability for LDPC decoding. In this paper, we concentrate not only on reducing the BER of demodulation, but also on providing accurate posterior probability estimates (PPEs). A new approach for the nonlinear demodulation based on the support vector machine (SVM) classifier is introduced. The SVM method which selects only a few sampling points from the filter output was used for getting PPEs. The simulation results show that the accurate posterior probability can be obtained with this method and the BER performance can be improved significantly by applying LDPC codes. Moreover, we analyzed the effect of getting the posterior probability with different methods and different sampling rates. We show that there are more advantages of the SVM method under bad condition and it is less sensitive to the sampling rate than other methods. Thus, SVM is an effective method for EBPSK demodulation and getting posterior probability for LDPC decoding.

1. Introduction

Nowadays, wireless communication is playing a very important role in our daily life. The growing demands on wireless multimedia services and products lead to increasing needs for radio spectrum and data rates. Thereby, the research on modulations with high bandwidth efficiency is on focus [1]. In order to satisfy the higher and higher demand for communication systems, an extended binary phase shift keying (EBPSK) system with very high spectra efficiency is introduced in [2]. A special impacting filter (SIF) which can produce high impact at the phase jumping point, narrow in bandwidth, and great improvement in SNR, was applied at the demodulator [3]. Therefore, a simple amplitude detector followed would perform the demodulation of EBPSK signals [4]. However, the conventional threshold decision may not be best to achieve the optimum performance, and the SIF used in EBPSK demodulator brings more difficulty in obtaining posterior probability for low-density parity check (LDPC) codes decoding. A simple and general bit metric generation method is proposed by Hyun and Yoon [5] for the soft information to initial channel decoding. We modify the scheme to suit our system and the method is referred

to as MHY in this paper. Meanwhile, nonlinear detectors are specifically designed to get the optimum performance of a blind multiuser detector [6, 7] and nonlinear channel equalization [8–10] and providing accurate posterior probability estimates (PPEs) for LDPC decoding [11, 12]. All results have shown that a nonlinear demodulator performs similar to an optimum receiver. One of the goals of this paper is the analysis of nonlinear demodulation with the channel decoder. We make use of the fact that the demodulator performance should not only be measured by low BER, but also in its ability to provide accurate PPEs that can be exploited by a soft-input channel decoder to achieve capacity. In this paper, we will introduce a nonlinear demodulation technique called the support vector machine (SVM) classifier [13]. The design approach is completely novel, where we select only a few samples of the SIF output for SVM training and testing at intermediate frequency (IF) without downconversion. We propose to measure the performance of this demodulator after an LDPC channel decoder, and the ability of SVM to provide accurate posterior probability predictions boosts the demodulator performance compared to the MHY method.

The rest of the paper is organized as follows. Section 2 is devoted to introducing SVM. We present the receiver scheme in Section 3 and briefly describe the EBPSK modulation and LDPC decoding. In Section 4, we include illustrative experiments to compare the performance of the proposed demodulators. We conclude in Section 5 with some final comments.

2. Support Vector Machine

The SVM is a classifier introduced by Cortes and Vapnik [14], which can realize the same performance as the so-called artificial neural networks (ANNs) for classification. Generally, ANN has the problem of a local minimum. On the other hand, the SVM is mathematically transparent and can provide global and unique solutions.

2.1. Binary Classification of SVM. For the binary classification problem, the training set consists of vectors from the pattern space $\mathbf{x}_i \in \mathbf{R}^n$, $i = 1, 2, \dots, L$ and to each vector a classification $y_i \in \{1, -1\}$. During the initial training stage, a decision function is constructed via

$$f(x) = \sum_{i=1}^L \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b, \quad (1)$$

where α_i is a Lagrangian constant, $K(\mathbf{x}, \mathbf{x}_i) = \Psi(\mathbf{x}_i)^T \cdot \Psi(\mathbf{x})$ is a kernel function, $\Psi(\mathbf{x})$ maps the training data vector \mathbf{x}_i into the high-dimensional feature space, and b is a bias term.

Define a coefficient vector \mathbf{w} , such that

$$\mathbf{w} = \sum_{i=1}^L \alpha_i y_i \Psi(\mathbf{x}_i), \quad (2)$$

then the training is completed by solving the following optimization problem:

$$\min_{\mathbf{w} \in \mathbf{H}, b \in \mathbf{R}, \xi \in \mathbf{R}^L} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^L \xi_i y_i ((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1 - \xi_i, \quad (3)$$

$$\xi_i \geq 0, \quad i = 1, 2, \dots, L,$$

where C is the tradeoff parameter between the training error and the margin of the decision function, and ξ_i is a slack variable to compensate for any nonlinearly separable training points.

In this paper, the SVM demodulator uses two types of kernel functions to compare the performance with each other. The first is the simplest linear kernel, shown as

$$K(x_i, x_j) = x_i^T x_j. \quad (4)$$

The second is a more popular radial basis function (RBF) kernel, shown as

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right), \quad \gamma > 0, \quad (5)$$

where γ controls the width of the function.

2.2. Complexity Analysis. The complexity of training an SVM for binary classification is $O(n^2)$, using the sequential minimal optimization [15], and Platt's method adds a computational complexity of $O(n^2)$. However, the SVM demodulator should be analyzed for the testing stage only because the training time is very small compared with the actual testing time. The main focus thus becomes analyzing the complexity required for the computing decision function in (1), which is using the simplest kernel. This issue will be discussed in detail later. A great amount of complexity can be reduced further in (1) if the expression is simplified as follows:

$$\begin{aligned} f(x) &= \sum_{i=1}^L \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b = \sum_{i=1}^L \alpha_i y_i (x_i^T x) + b \\ &= \left[\sum_{j=1}^N y_j \alpha_j \left(\sum_{i=1}^n x_{j,i} x_i \right) \right] + b = \sum_{i=1}^n x_i \left(\sum_{j=1}^N y_j \alpha_j x_{j,i} \right) + b \\ &= \sum_{i=1}^n A_i x_i + b, \end{aligned} \quad (6)$$

where N is the number of support vectors, and the constants $A_i = \sum_{j=1}^N y_j \alpha_j x_{j,i}$ and b can be precomputed before the testing stage to save the computation time. Therefore, the complexity of the SVM demodulator is $O(n)$.

2.3. Probabilistic Outputs of SVM. Instead of predicting the label, many applications require a posterior class probability $P(y = 1 | x)$. The transformation of SVM output into posterior probabilities has been proposed by Platt in [16]. Platt's method squashes the SVM soft output through a trained sigmoid function to predict posterior probabilities:

$$p(y = 1 | \mathbf{x}) \approx P_{A,B}(f) = \frac{1}{1 + \exp(Af + B)}, \quad (7)$$

where $f = f(\mathbf{x})$, let each f_i be an estimate of $f(\mathbf{x}_i)$. The best parameter setting $z^* = (A^*, B^*)$ is determined by solving the following regularized maximum likelihood problem:

$$\min_{z=(A,B)} F(z) = - \sum_{i=1}^l (t_i \log(p_i) + (1 - t_i) \log(1 - p_i)), \quad (8)$$

where $p_i = P_{A,B}(f_i)$, $t_i = (y_i + 1)/2$.

Furthermore, log and exp could easily cause an overflow, if $Af_i + B$ is large, $\exp(Af_i + B) \rightarrow \infty$ and $1 - p_i = 1 - 1/(1 + \exp(Af_i + B))$ is a "catastrophic cancellation" when p_i is close to one. The problem can usually be resolved by reformulation [17]:

$$-(t_i \log p_i + (1 - t_i) \log(1 - p_i)) \quad (9)$$

$$= (t_i - 1)(Af_i + B) + \log(1 + \exp(Af_i + B)) \quad (10)$$

$$= t_i(Af_i + B) + \log(1 + \exp(-Af_i - B)). \quad (11)$$

If $Af_i + B \geq 0$ then use (11), else use (10). Then (7) can be rewritten as follows:

$$p(y = 1 | \mathbf{x}) \approx \begin{cases} \frac{1}{1 + \exp(Af + B)}, & Af + B < 0, \\ \frac{\exp(-Af - B)}{1 + \exp(-Af - B)}, & Af + B \geq 0. \end{cases} \quad (12)$$

From (12), we can see that SVM does not provide PPE and its output needs to be transformed, before it can be interpreted as posterior probabilities; therefore, the posterior probability is an approximate one.

3. Communication System

3.1. EBPSK Modulation. EBPSK is a modulation method with high frequency spectra efficiency, which is defined as follows:

$$f_0(t) = A \sin 2\pi f_c t, \quad 0 \leq t < T,$$

$$f_1(t) = \begin{cases} B \sin(2\pi f_c t + \theta), & 0 \leq t < \tau, 0 \leq \theta \leq \pi, \\ A \sin(2\pi f_c t), & \tau \leq t < T, \end{cases} \quad (13)$$

where f_0 and f_1 are modulation waveforms corresponding to bit "0" and bit "1," respectively, $T = N/f_c$ is the bit duration, $\tau = K/f_c$ is the phase modulation duration, and θ is the modulating angle. Obviously, if $\tau = T$ and $\theta = \pi$, (13) degenerates to the classical binary phase shift keying (BPSK) modulation.

3.2. LDPC Decoding. LDPC codes can be decoded by an iterative message-passing (MP) algorithm which passes messages between the variable nodes and check nodes iteratively. If the messages passed along the edges are probabilities, then the algorithm is also called belief propagation (BP) decoding, which is the optimal if there are no cycles or cycles are ignored. Moreover, with BP decoding, complicated calculations are distributed among simple node processors, and after several iterations, the solution of the global problem is available. The steps of BP decoding are as follows.

(1) Initialization: $p_n^0(x) = q_{nm}^0 = p(x_n = x | y_n)$, where $p(x_n = x | y_n)$ is the soft information of channel outputs.

(2) Horizontal step: the MAP output from c_m to v_n :

$$\begin{aligned} r_{mn}^k(0) &= p(v_n = 0 | c_m = 0, y_{i \in B(m) \setminus n}), \\ r_{mn}^k(0) &= \frac{1}{2} + \frac{1}{2} \prod_{i \in B(m) \setminus n} (1 - 2q_{im}^k(1)), \\ r_{mn}^k(1) &= 1 - r_{mn}^k(0). \end{aligned} \quad (14)$$

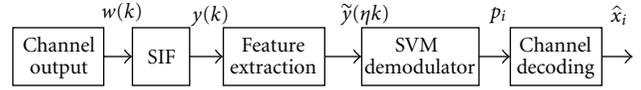


FIGURE 1: The block diagram of EBPSK receiver.

(3) Vertical step: updating the message from v_n to c_m :

$$\begin{aligned} q_{nm}^{k+1}(0) &= \theta p_n^0(0) \prod_{j \in A(n) \setminus m} r_{jn}(0), \\ q_{nm}^{k+1}(1) &= \theta p_n^0(1) \prod_{j \in A(n) \setminus m} r_{jn}(1), \quad \theta \text{ is chosen to ensure} \\ q_{nm}^{k+1}(0) + q_{nm}^{k+1}(1) &= 1, \text{ Compute } p_n^k(x), \\ p_n^{k+1}(0) &= \theta p_n^0(0) \prod_{j \in A(n)} r_{jn}(0), \\ p_n^{k+1}(1) &= \theta p_n^0(1) \prod_{j \in A(n)} r_{jn}(1). \end{aligned} \quad (15)$$

(4) Tentative output:

$$v_n^{k+1} = \begin{cases} 1, & p_n^{k+1}(1) \geq 0.5, \\ 0, & p_n^{k+1}(1) < 0.5, \end{cases} \quad (16)$$

if all parity check equations are satisfied or the maximum iteration number is reached, stop iteration, else return to Step (2).

In this paper, we focus on the initialization step for the posterior probabilities obtained by the nonlinear demodulator.

3.3. System Model. Figure 1 shows the receiver of EBPSK system. Suppose the system is synchronized, the signal of the channel output can be expressed as $w(k) = z(k) + n(k)$, where $n(k)$ is Gaussian white noise with zero mean. Input $w(k)$ into a SIF, and then the output signal can be expressed as $y(k) = w(k) * h(k)$, where $h(k)$ is the impulse response of SIF. In order to reduce the demodulation complexity, we select a few sample points as the features for SVM training and testing. Then, using the decision function (1), we can get the binary output as follows:

$$\text{sign}(f(\tilde{\mathbf{y}}(\eta k))) = \text{sign}\left(\sum_{i=1}^l \alpha_i c_i K(\tilde{\mathbf{y}}_i(\eta k), \tilde{\mathbf{y}}(\eta k)) + b\right). \quad (17)$$

Then, we can get the posterior probability $p(\tilde{x}_i = 1 | \tilde{\mathbf{y}}_i(\eta k))$ and $p(\tilde{x}_i = 0 | \tilde{\mathbf{y}}_i(\eta k)) = 1 - p(\tilde{x}_i = 1 | \tilde{\mathbf{y}}_i(\eta k))$ through (12); finally, we use $p(x_i = x | y_i)$ to initiate the LDPC decoder.

4. Simulation Results and Discussions

In this section, we illustrate the performance of the proposed SVM demodulation and its soft output for LDPC decoding.

TABLE 1: Comparison of SVM models.

	Selected kernel	
	RBF	Linear
C	4	2
γ	8	—
SVs	271	210

Unless specified otherwise, all simulations assume that the system had 3000 random symbols for training and the reported BER is computed using 10^5 symbols and we average the results over 1000 independent trials with random training and test data. We choose $K = 2$, $N = 20$, $A = B = 1$, $\theta = \pi$ as the parameters of EBPSK modulation. LDPC codes are also applied to measure the BER performance of the communication system and the accurate posterior probability obtained by the SVM method. During simulations, we use a 1/2 rate regular LDPC code with 1000 bits per codeword and 3 ones per column. The whole system was simulated under MATLAB.

4.1. Kernel Selection and Demodulation. In this subsection, the performance of the SVM demodulator, using the kernel functions (4) and (5), introduced in Section 2, is compared. For the RBF kernel, a 10-fold cross-validation sweep from the training samples was used to find the optimum parameters of C and γ . A similar search was conducted for the linear kernel, but it only has the C parameter to adjust. Table 1 summarizes the optimum SVM model obtained after the parameter search.

The linear kernel has less support vectors than the RBF one; therefore, it has a less computational complexity and thus would perform faster. In order to compare the BER performance fairly, both kernels used by the SVM receiver were classifying exactly the same received signals.

Figure 2 shows the BER performance of the SVM demodulator when employing different kernels; also, the performance of conventional threshold decision is analyzed. Evidently, the linear kernel, though much simpler, has slightly better performance than the RBF kernel. Moreover, the SNR gain between the SVM method and the threshold decision is around 1.8 dB; therefore, a linear SVM is chosen for the task. Training on a “worse-case” scenario works well (SNR = -7 dB in this case), proving that the SVM receiver needs not frequently retraining in different SNRs.

4.2. Kernel Optimization. To optimize the linear kernel, the only controlling parameter is C , which restrains the maximum size of the Lagrangian dual variable. The SVM detector is tested on the 20 sets of 20000 noisy sequences at SNR = 2 dB for various C values. The results are shown in Figure 3. While the error performance for various C is very similar, it is still ideal to choose a model with the least number of support vector (SV) in order to reduce the complexity. In this case, when C is beyond 6, the model gives the same number of SV because variable α_i is no longer

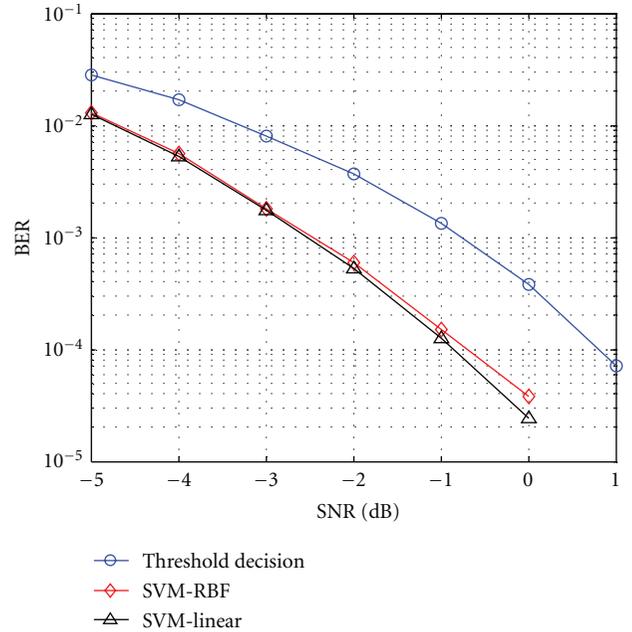
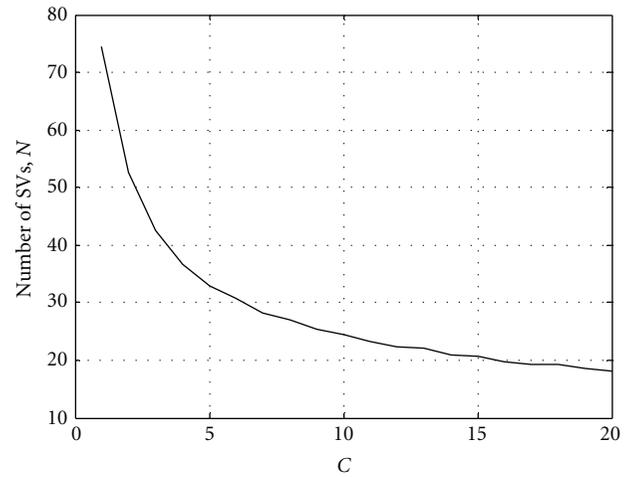


FIGURE 2: Demodulation with SVM-RBF, SVM-linear, and threshold decision.

FIGURE 3: Number of support vectors from the SVM model for different C parameters, $n = 5$.

constrained by C . The correct rate remains around 99.47%, as shown in Figure 4.

The training size for the SVM detector is another parameter that the designer needs to control. In general, for any machine learning algorithms, the training size should be as large as possible to improve the prediction of the unknown testing data. The tradeoff in this application is the increased time required to produce and collect the training data. Figures 5 and 6, respectively, show the SVM demodulator’s error performance and the number of SVs required on the same system as stated above with different training sizes. When the C parameter is fixed at 2, and with a training size of about 200, the performance of the SVM detector would reach

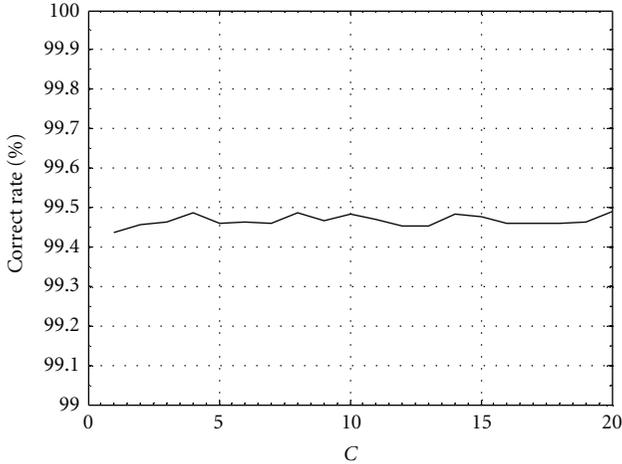


FIGURE 4: Correct rate of the SVM model with linear kernel for different C parameters, SNR = -4 dB, n = 5.

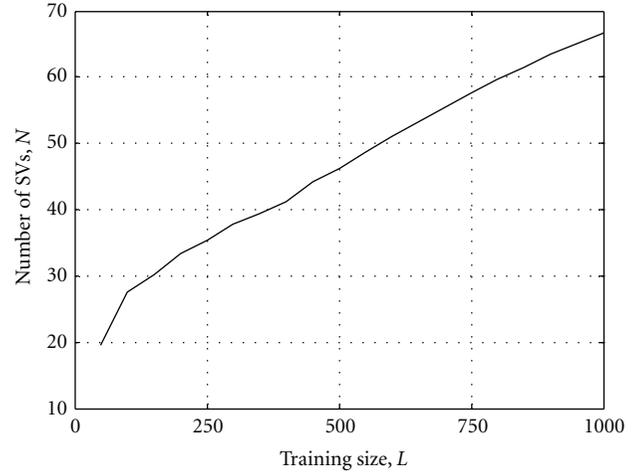


FIGURE 6: Number of support vectors from the SVM model for different training sizes, n = 5.

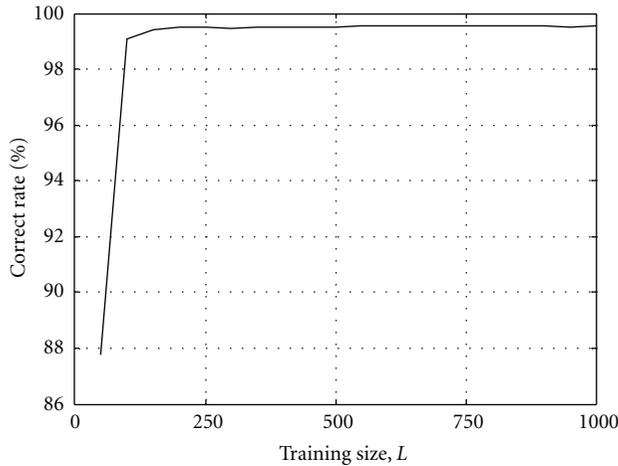


FIGURE 5: Correct rate of the SVM model with linear kernel for different training sizes, n = 5.

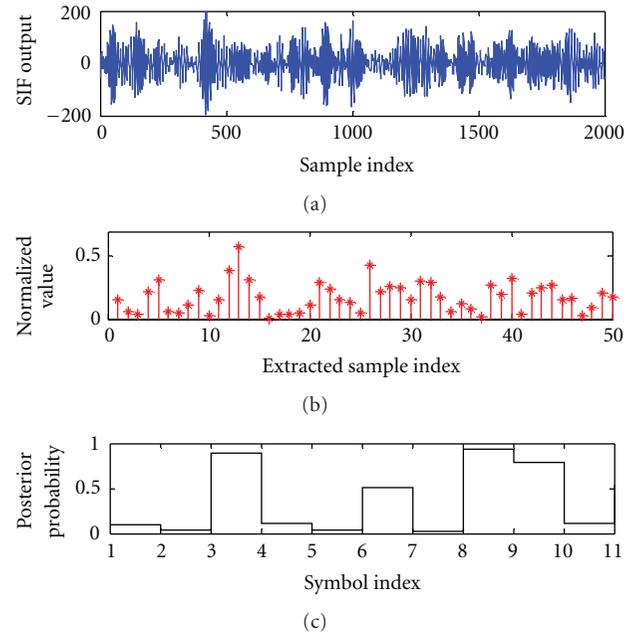


FIGURE 7: The waveform of SIF output and the posterior probability output obtained by SVM at SNR = -9 dB.

to its limit where the increase of SVs would not improve its accuracy.

4.3. Posterior Probability Estimates. In order to reduce the complexity of the SVM analyzed in Section 2, we select only a few samples from the filter output as the features for training and testing (i.e., $n = 5$ in this case). We depict the probabilities obtained by the SVM output of SNR = -9 dB in Figure 7. The signal in Figure 7 is submerged in noise, so the optimal performance cannot be achieved by using a conventional threshold decision. Yet, the probability which the demodulator output by SVM technique is accurate while a source symbol sequence [0, 0, 0, 0, 1, 1, 0, 1, 0, 1] is transmitted, and the noise from the part which did not carry any information of the waveform of symbol “1” is almost removed.

To understand the difference in PPEs, we have plotted the curves for the SVM and the MHY in Figures 8(a) and 8(b), respectively, with SNR = -5 dB. We depict the

estimated probabilities $P(y = 1 | x)$ versus the ones when a source symbol sequences with all ones are transmitted. We can appreciate that the SVM PPEs are closer to “1” and less spread, most of the values of demodulation output are between 0.9 and 1. Thereby, SVM estimates are closer to the true posterior probability, which explains its improved performance with respect to the MHY, when we measure the BER after the LDPC decoder.

In a previous subsection, we have shown that the demodulator is based on an SIF and SVM classifier, when we compare performances at a low BER. In this section, we focus on the performance after the sequence has been corrected by an LDPC decoder. The ability of SVM to provide accurate

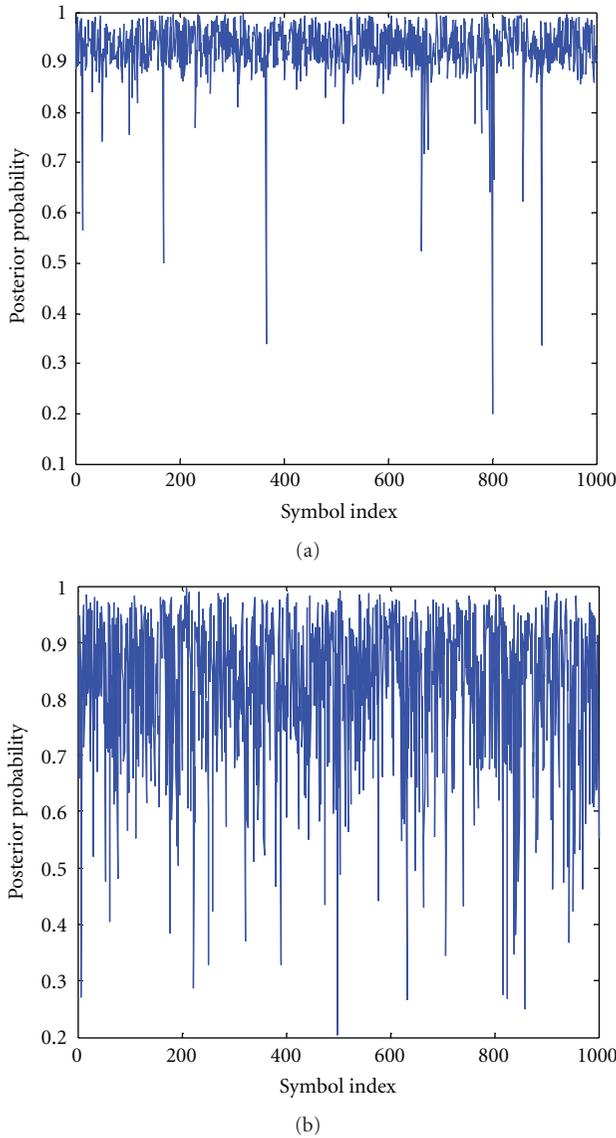


FIGURE 8: The posterior probability $P(y = 1 | x)$ obtained by SVM and MHY method, in (a) and (b), respectively, where source symbols with all ones are transmitted.

posterior probability predictions boosts the demodulator performance compared to the MHY method.

From Figure 8, we can understand that the improved performance of the SVM with respect to the MHY is based on its ability to provide accurate PPEs. In Figure 9, we can appreciate that the SVM-LDPC significantly reduces the BER at lower SNR, because the PPEs are more accurate and the LDPC decoder can rely on these trustworthy predictions. Also, Figure 9 shows that the performance of SVM-RBF-LDPC is a little more superior to SVM-linear-LDPC, it is not the same as the results in Section 4.1 which are analyzed without channel coding. Moreover, the SVM-linear-LDPC decoding outperforms the MHY-LDPC decoding by 4.5 dB and by 18 dB without channel coding when $BER = 10^{-4}$ and sampling rate $f_s = 4f_c$. In Figure 10, we compare the

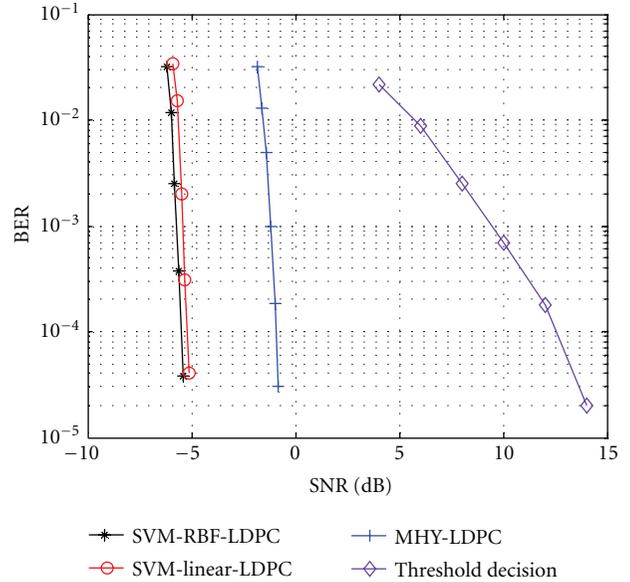


FIGURE 9: Performance at the output of the LDPC decoder with the soft-input and threshold decision.

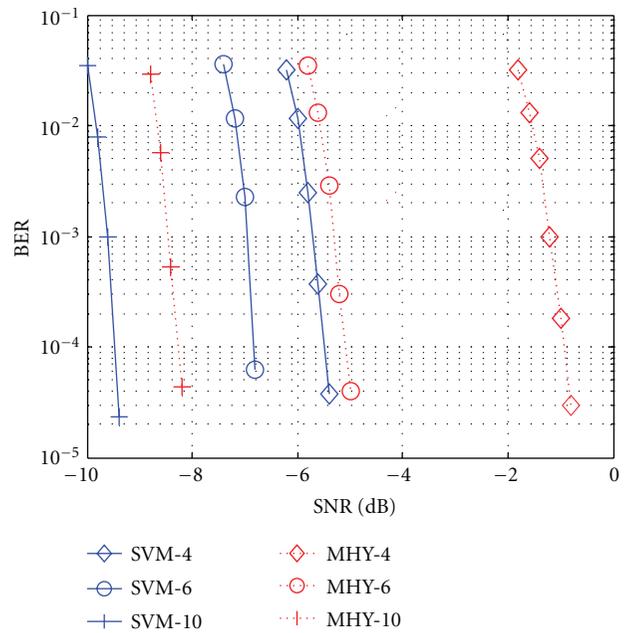


FIGURE 10: BER performance comparisons of the SVM with MHY method at the output of the LDPC decoder with different sampling rates. Using SVM-4, SVM-6, and SVM-10 for the SVM method (solid lines) and MHY-4, MHY-6, and MHY-10 for the MHY method (dashed lines) with $f_s = 4f_c$, $f_s = 6f_c$, and $f_s = 10f_c$, respectively.

BER performance of the SVM-LDPC with MHY-LDPC by a different sampling rate. Compared to the MHY-LDPC, the SVM-LDPC can upgrade more than 4.6 dB, 1.7 dB, and 1.2 dB for $f_s = 4f_c$, $f_s = 6f_c$, and $f_s = 10f_c$, respectively. This means that the performance of SVM-LDPC improved significantly while the sampling rate is low, and it is not

sensitive to the sampling rate for SVM-LDPC. Also, Figure 10 illustrates that it is more superior for the SVM demodulator than MHY in a bad condition.

We have shown that SVM-LDPC is far superior to the MHY method. This result shows that using a method that can predict accurately the PPEs allows the LDPC decoding algorithm to perform to its fullest.

5. Conclusions

In this paper, we introduce a nonlinear demodulator which is a novel solution for the EBPSK scheme. We have shown that the performance can be significantly improved by using a linear kernel for demodulation, which has a less computational complexity thus saves the computation time.

SVM is a nonlinear probabilistic classifier that produces accurate PPEs. The performance comparisons of different probabilistic demodulators at the output of an LDPC channel decoder are made, which has shown that the SVM outperforms the MHY with probabilistic output.

The SVM probability output method does not need to estimate the channel noise power σ , and uses only a few samples as the features of SVM for training and testing, which reduces the complexity significantly.

A simulator of the system was designed and the BER performance was significantly improved for the SVM-LDPC comparing with the MHY-LDPC approach. Moreover, the SVM method is more robust to sampling rate than MHY method.

Yet, the performance of the system can be improved significantly at the cost of complexity, and the probability is still approximate. More investigations are undertaken to reduce the computational complexity of this approach and test its performance under more severe channel conditions, such as the fading channel.

Acknowledgment

The authors would like to thank the support of the National Natural Science Foundation of China (NSFC) under Grant 61271204.

References

- [1] H. R. Walker, "VPSK and VMSK modulation transmit digital audio and video at 15 Bits/Sec/Hz," *IEEE Transactions on Broadcasting*, vol. 43, no. 1, pp. 96–103, 1997.
- [2] M. Feng and L. Wu, "Special non-linear filter and extension to Shannon's channel capacity," *Digital Signal Processing*, vol. 19, no. 5, pp. 861–873, 2009.
- [3] L. Wu and M. Feng, "On BER performance of EBPSK-MODEM in AWGN channel," *Sensors*, vol. 10, no. 4, pp. 3824–3834, 2010.
- [4] M. Feng, L. Wu, J. Ding, and C. Qi, "BER analysis and verification of EBPSK system in AWGN channel," *IEICE Transactions on Communications*, vol. 94, no. 3, pp. 806–809, 2011.
- [5] K. Hyun and D. Yoon, "Bit metric generation for Gray coded QAM signals," *IEE Proceedings Communications*, vol. 152, no. 6, pp. 1134–1138, 2005.
- [6] J. W. H. Kao, S. M. Berber, and V. Kecman, "Blind multiuser detector for chaos-based CDMA using support vector machine," *IEEE Transactions on Neural Networks*, vol. 21, no. 8, pp. 1221–1231, 2010.
- [7] K. C. Ho, X. Lu, and V. Mehta, "Adaptive blind narrow-band interference cancellation for multi-user detection," *IEEE Transactions on Wireless Communications*, vol. 6, no. 3, pp. 1024–1033, 2007.
- [8] F. Perez-Cruz, J. 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, 2008.
- [9] H. Zhao and J. Zhang, "Functional link neural network cascaded with Chebyshev orthogonal polynomial for nonlinear channel equalization," *Signal Processing*, vol. 88, no. 8, pp. 1946–1957, 2008.
- [10] J. C. Patra, P. K. Meher, and G. Chakraborty, "Nonlinear channel equalization for wireless communication systems using Legendre neural networks," *Signal Processing*, vol. 89, no. 11, pp. 2251–2262, 2009.
- [11] P. M. Olmos, J. J. Murillo-Fuentes, and F. Pérez-Cruz, "Joint nonlinear channel equalization and soft LDPC decoding with Gaussian processes," *IEEE Transactions on Signal Processing*, vol. 58, pp. 1183–1192, 2010.
- [12] N. Singla and J. A. O'Sullivan, "Joint equalization and decoding for nonlinear two-dimensional intersymbol interference channels," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '05)*, pp. 1353–1357, Adelaide, Australia, September 2005.
- [13] W. J. Park and R. M. Kil, "Pattern classification with class probability output network," *IEEE Transactions on Neural Networks*, vol. 20, no. 10, pp. 1659–1673, 2009.
- [14] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [15] J. Platt, "Sequential minimal optimization: a fast algorithm for training support vector machines," in *Advances in Kernel Method: Support Vector Learning*, B. Scholkopf, Ed., pp. 185–208, The MIT Press, Cambridge, Mass, USA, 1998.
- [16] J. Platt, "Probabilities for SV machines," in *Advances in Large Margin Classifiers*, A. J. Smola and P. L. Bartlett, Eds., pp. 61–73, The MIT Press, Cambridge, Mass, USA, 2000.
- [17] H. T. Lin, C. J. Lin, and R. C. Weng, "A note on Platt's probabilistic outputs for support vector machines," *Machine Learning*, vol. 68, no. 3, pp. 267–276, 2007.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

