

# Improving DVB-S2 Performance Through Constellation Shaping and Iterative Demapping

Xingyu Xiang and Matthew C. Valenti

Lane Department of Computer Science and Electrical Engineering  
West Virginia University, Morgantown, WV, U.S.A.

**Abstract**—Because it is widely supported by commercial-off-the-shelf (COTS) technology, the DVB-S2 waveform standard has become an attractive solution for military communication links. The waveform uses a combination of amplitude-phase-shift keying (APSK) modulation and low-density parity-check (LDPC) codes. Typical DVB-S2 implementations select signals from the APSK signal set with uniform probability. However, information-theoretic results suggest that performance may be improved by selecting lower-energy signals more frequently than higher-energy signals. In this paper, we propose and analyze a DVB-S2-compatible system that shapes the APSK constellation by selecting signals with a nonuniform probability. The receiver iterates between the APSK demapper, the shaping decoder, and the LDPC decoder. Using 32-APSK and a rate of 3 data bits per symbol, the system described in this paper achieves a gain of over 1 dB relative to a standard DVB-S2 system (i.e. one that does not use shaping or iterative demodulation) in additive white Gaussian noise (AWGN) at a bit-error rate of  $10^{-5}$ . About 0.7 dB of the gain can be attributed to shaping, and rest of the gain can be attributed to iterative demodulation and decoding. Lesser gains can be achieved over Rayleigh fading channels.

## I. INTRODUCTION

Digital Video Broadcasting Satellite - Second Generation (DVB-S2) is the successor to the ubiquitous DVB-S satellite digital video broadcasting standard [1]. DVB-S2 supports the broadcast of standard-definition and high-definition television (HDTV), interactive services, and data content distribution. Because of the wide availability of commercial-of-the-shelf (COTS) technology, DVB-S2 has become an attractive solution for military communication links [2]–[5]. Compared with the first generation, DVB-S2 uses much stronger coding combined with high-order modulation. In particular, DVB-S2 uses low-density parity-check (LDPC) codes and amplitude-phase-shift keying (APSK) modulation with constellations containing up to 32 symbols. Together, these features achieve a 30% gain in transmission rate when using the same satellite transponder bandwidth and emitted signal power as DVB-S.

With APSK modulation, the signals are located on several concentric circles, each with a different amplitude. In typical implementations, the APSK symbols are selected with uniform probability. However, as we have shown in [6], the performance of APSK may be improved by selecting lower-energy signals more frequently than higher-energy signals. The strategy in [6] is an instance of the general concept of

The authors were sponsored by the National Science Foundation under Award No. CNS-0750821 and by the United States Army Research Laboratory under Contract W911NF-10-0109.

constellation shaping [7]–[10]. Our strategy for APSK is based on that proposed by Le Goff et al. in [10] for bit-interleaved turbo-coded pulse-amplitude modulation (PAM). The strategy in [10] partitions the basic constellation into two or more sub-constellations of increasing average energy and uses a *shaping code* to select signals from the lower energy sub-constellations more often than the signals from higher energy sub-constellations.

Our previous work in [6] on shaping for APSK was limited to only turbo codes. However, DVB-S2 uses LDPC codes. Because the nature of turbo and LDPC decoders are quite different, the adaptation of shaping to LDPC-coded systems is not trivial. The DVB-S2 system can be considered as an instance of bit-interleaved coded modulation (BICM) [11]. Like other BICM systems, DVB-S2 can be iteratively decoded using the strategy known as BICM with iterative demapping and decoding (BICM-ID) [12]. BICM-ID has been previously considered for DVB-S2 in [13], but that paper does not consider shaping.

In this paper we apply a combination of BICM-ID and constellation shaping to the DVB-S2 system. On the one hand, the paper can be considered to be an extension of [6] to accommodate LDPC codes, while on the other hand, it can be considered an extension of [13] to accommodate shaping. At a rate of 3 data bits per APSK symbol and using 32-APSK, the system in this paper achieves a gain of over 1 dB at a bit-error rate of  $10^{-5}$  relative to a typical DVB-S2 implementation in additive white Gaussian noise (AWGN). About 0.7 dB of the gain can be attributed to shaping, and the other 0.3 dB can be attributed to iterative demapping and decoding. Lesser gains can be achieved over Rayleigh fading channels. The gains are achieved without an expansion in bandwidth or a change to the standardized LDPC codes or APSK constellations. All that is needed on the transmitter side is the inclusion of a shaping encoder between the LDPC encoder and the APSK modulator. The receiver architecture, which is disclosed in this paper, requires iteration between the APSK demapper, the shaping decoder, and the LDPC decoder.

The remainder of this paper is organized as follows. A general model for bit-interleaved LDPC-coded APSK with constellation shaping is provided in Section II. Section III discusses the shaping code. The BICM-ID receiver implementation is discussed in Section IV. Numerical results and conclusions are given in Sections V and VI, respectively.

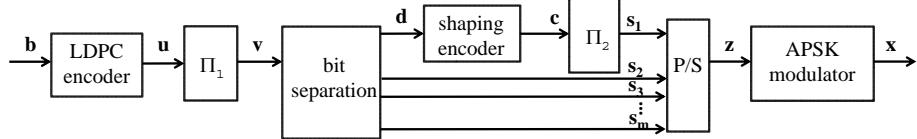


Fig. 1. Transmitter structure.

## II. SYSTEM MODEL

### A. Transmitter

A block diagram of the transmitter is shown in Fig. 1. The input to the system is a length- $k_c$  vector  $\mathbf{b}$  of information bits, which is fed into an LDPC encoder. The LDPC encoder produces a length- $n_c$  codeword  $\mathbf{u}$ , which is permuted by interleaver  $\Pi_1$  to produce the vector  $\mathbf{v}$ . A *bit separator* separates  $\mathbf{v}$  into  $m$  streams, where  $m = \log_2 M$  and  $M$  is the number of symbols in the APSK constellation. The first stream, denoted  $\mathbf{d}$ , is passed through a shaping encoder to produce the shaping codeword  $\mathbf{c}$ , which is then permuted by interleaver  $\Pi_2$  to produce the vector  $\mathbf{s}_1$ . The purpose of the shaping encoder, as described below, is to produce a nonuniform output with a higher likelihood of 0's than 1's. The remaining streams at the output of the bit separator are denoted  $\{\mathbf{s}_2, \dots, \mathbf{s}_m\}$  and do not pass through the shaping encoder. The  $m$  streams  $\mathbf{s}_i$  are all of equal length and pass through a parallel-to-serial converter (P/S) and into an APSK modulator. Since the  $\mathbf{s}_i$  are of the same length and the shaping encoder has a rate less than unity, it follows that the first stream  $\mathbf{d}$  output by the bit separator is actually shorter than the other stream output by the bit separator.

The P/S converter assures that each APSK symbol is selected using one bit from each of the  $\mathbf{s}_i$ . The bit from  $\mathbf{s}_1$  is called a *shaping* bit and is used to select from among two subconstellations with nonequal probability. When the shaping bit is equal to 0, the lower-energy subconstellation is selected, while when it is 1, the higher-energy subconstellation is selected. The remaining  $m - 1$  bits, which are not shaped, are used to select from among the  $M/2$  symbols within the subconstellation. An example symbol labeling map is shown in Fig. 2, which is the labeled 32-APSK constellation from the DVB-S2 standard. To be consistent with the DVB-S2 standard, the shaping bit is the second bit from the left. With this bit used as the shaping bit, the high-energy subconstellation is the outer ring of 16 symbols, while the lower-energy subconstellation is the union of the two inner rings of symbols.

Note that in Fig. 1, there is just a single shaped bit, which corresponds to partitioning the constellation into two subconstellations. More generally,  $g$  bits could be shaped, in which case the constellation is partitioned into  $2^g$  subconstellations [6]. To accommodate such a possibility, we use the shorthand  $\mathbf{s}_{1:g}$  to represent the first  $g$  streams, which have passed through the shaping encoder, and  $\mathbf{s}_{g+1:m}$  to represent the other  $m - g$  streams, which do not pass through the shaping encoder.

The overall rate of the system  $R$  is the number of information bits per modulated symbol, and is related to the rates of

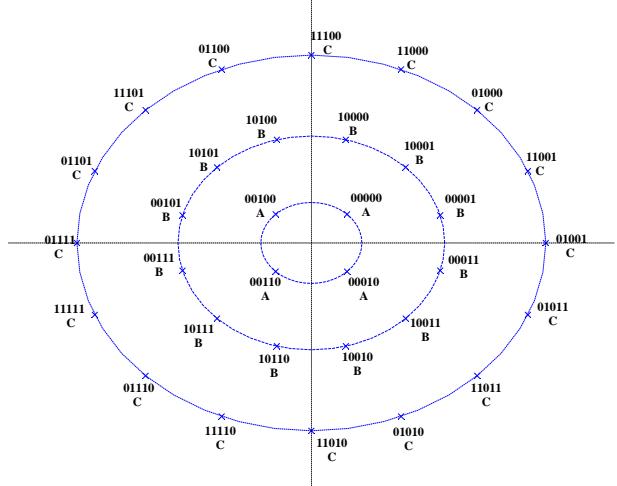


Fig. 2. 32-APSK constellation. The shaping bit is the second from left.

the LDPC code and shaping codes rate by:

$$R = R_c [m + g(R_s - 1)] \quad (1)$$

where  $R_c$  is the rate of LDPC code and  $R_s = k_s/n_s$  is the rate of shaping code. When shaping is used,  $g > 0$  and  $R_s < 1$ , which implies that the rate of the LDPC code  $R_c$  used by a system with shaping must be higher than the  $R_c$  used without shaping if the overall system rate  $R$  is to remain fixed. In other words, the loss of the system rate  $R$  caused by the shaping code must be compensated by an appropriate increase in  $R_c$ .

### B. Receiver

The receiver structure is shown in Fig. 3. While Section IV describes the receiver in detail, a general description is provided here to provide a better understanding of the entire system. Throughout this discussion, it is assumed that all information exchanged within the decoder is in the form of log-likelihood ratios (LLRs). The notation  $L_a$  denotes an LLR input to a module (which may possibly be fed back from another module), while  $L_e$  denotes the LLR output by a module. The received symbols  $\mathbf{y}$ , which are corrupted by the channel, are passed into the APSK demodulator. The APSK demodulator produces log-likelihood ratios (LLRs), which are divided into  $m$  streams  $L_e(\mathbf{s}_i)$  by reversing the P/S operation of the transmitter. The bit-separation operation is expressed as S/P in the diagram, and  $L_e(\mathbf{s}_{2:m})$  is used to represent the LLR of streams  $\{\mathbf{s}_2, \dots, \mathbf{s}_m\}$ .

The first stream  $L_e(\mathbf{s}_1)$  is de-permuted to produce  $L_a(\mathbf{c})$ , which is fed into the shaping decoder. Initially, the *a priori* input to the shaping decoder  $L_a(\mathbf{d})$  is set to all zeros. A parallel to serial converter combines the output of the shaping

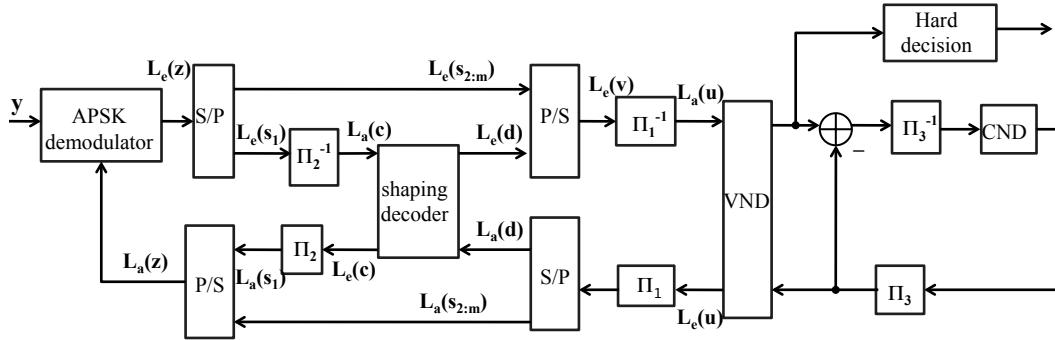


Fig. 3. Receiver structure.

decoder  $L_e(\mathbf{d})$  with the LLRs of the other streams  $L_e(\mathbf{s}_{2:m})$  to produce a vector  $L_e(\mathbf{v})$  of combined LLRs. The vector  $L_e(\mathbf{v})$  is de-permuted by  $\Pi_1^{-1}$ , and the resulting vector  $L_a(\mathbf{u})$  is introduced as the input to the LDPC decoder.

Rather than illustrating the decoder structure using a Tanner or factor graph, Fig. 3 interprets the LDPC code as the serial concatenation of an inner repetition code and outer single parity check code, i.e. as an extended irregular repeat-accumulate (eIRA) code [14]. Such a representation helps to illuminate the interaction between the LDPC and shaping decoders. The LDPC decoder is comprised of two processing blocks: A variable node decoder (VND) and a check node decoder (CND) [15]. The VND produces *a posteriori* messages  $L_e(Q_i)$  of bits  $u_i$ , which, after interleaving and subtraction by the CND output, will form the input to the CND. The interleaver, denoted by  $\Pi_3$  in the diagram, corresponds to the edges in the code's Tanner graph that connect the variable and check nodes. After interleaving, the output of the CND is used as the *a priori* input to the VND.

The output  $L_e(\mathbf{u})$  of the LDPC decoder is fed back to the shaping decoder and APSK demodulator, also called the *demapper*, to be used as extrinsic information during the next iteration. A second S/P converter sends the LLRs of the shaped bits into the shaping decoder as input  $L_a(\mathbf{d})$ , where after the initial iteration it is used as the *a priori* input. Another P/S converter, which is identical to the P/S in Fig. 1, combines the interleaved output of the shaping decoder with the LLRs of the unshaped bits to create the vector  $L_a(\mathbf{z})$ . The output of that P/S converter is used as the *a priori* input to the demapper after the initial iteration (prior to the first iteration,  $L_a(\mathbf{z})$  is initialized according to the average bit probabilities for each bit position, as described in Section IV).

During each receiver iteration, four information exchanges are performed: (1) Between the demapper and LDPC decoder, (2) Between the demapper and shaping decoder, (3) Between the LDPC decoder and shaping decoder, and (4) Between the VND and CND blocks inside the LDPC decoder. Notice that only one iteration of LDPC decoding is executed per iteration of demapping and shaping decoding. In fact, the shaping decoder is the only additional processor required for this system compared to the BICM-ID system in [13]. Although the inclusion of the shaping decoder increases the complexity of the receiver, the additional complexity is moderate provided that the reasonably small length of the shaping code.

### III. SHAPING FOR 32-APSK

Prior to encoding, the shaping encoder's input  $\mathbf{d}$  must be segmented into  $L$  short blocks, each of length  $k_s$ . The encoder operates on each of these blocks by mapping the  $k_s$  bits onto a length- $n_s$  shaping codeword based on a codeword table. The  $L$  code blocks that are generated by the encoder are then reassembled into a length- $Ln_s$  shaping codeword  $\mathbf{c}$ . Let  $p_0$  denote the probability that a particular bit in  $\mathbf{c}$  (or  $\mathbf{s}_1$ ) equals zero, and  $p_1$  be the probability that it equals one.  $p_0$  should be always greater than  $p_1$  according to the construction methodology of the shaping codeword table.

The shaping codeword table  $\mathcal{C}$  is a size  $2^{k_s} \times n_s$  matrix. Each row of  $\mathcal{C}$  is a length- $n_s$  codeword. Each unique length- $k_s$  binary input sequence is used to select a particular row from the matrix. The codeword table  $\mathcal{C}$  is initialized to contain the all-zeros codeword of length  $n_s$ , and codewords with higher weight are recursively added to  $\mathcal{C}$ . Suppose that  $\mathcal{C}$  contains all codewords of weight  $w-1$  or lower but the number of rows in  $\mathcal{C}$  is still less than  $2^{k_s}$ . Then weight- $w$  codewords are drawn and added to  $\mathcal{C}$  until the number of distinct codeword is  $2^{k_s}$ . To assure each bit position in the codeword has approximately the same value of  $p_0$ , each new weight- $w$  codeword is selected with the goal of balancing the column weights of the codeword table  $\mathcal{C}$ .

As an example, consider the  $(n_s, k_s) = (5, 3)$  code. There are  $2^3 = 8$  codewords in  $\mathcal{C}$ . The number of binary codeword of weight two or less is:

$$\binom{5}{0} + \binom{5}{1} = 6.$$

In addition to these six codewords,  $\mathcal{C}$  will contain two more codeword of weight 2. There are  $\binom{5}{2} = 10$  possible weight-2 codewords to chose from when creating the code table. For instance, if the last two codewords in  $\mathcal{C}$  are  $\{(00011), (00101)\}$ , then the column weights would be  $[1 \ 1 \ 2 \ 2 \ 3]$ . Alternatively, if they are  $\{(01100), (00011)\}$ , then the column weights would be  $[1 \ 2 \ 2 \ 2 \ 2]$ . In the second design, the probability  $p_0$  of each bit position, which is equal to the column weight divided by 8 (the number of rows), is nearly constant. Thus, the second design is preferred over the first. This design policy is corroborated by the simulation results, which demonstrate improved performance when the column weights are carefully balanced.

The 32-APSK constellation in Fig. 2 follows the DVB-S2 standard and therefore consists of three concentric rings, with 4 points in the inner ring, 12 points in the middle ring, and 16 points in the outer ring. When there are two equal sized sub-constellations ( $g = 1$ ), the shaping bit is the second bit of the word based on the labeling rule in Fig. 2. The first partition, which is selected with probability  $p_0$ , contains the signals in the first two rings (i.e. signals labeled  $A$  and  $B$  in the diagram). The second partition, which is selected with probability  $p_1$ , contains the signals in the outer ring (i.e. those labeled  $C$ ). Due to the larger number of 0's in shaping code, signals in the inner two rings are more likely to be selected than signals in the outer ring.

Because the DVB-S2 standard only defines LDPC codes for a fixed set of eleven different rates from  $R_c = 1/4$  to  $R_c = 9/10$ , then according to equation (1), the possible choice of shaping code rates  $R_s$  is limited if the overall rate  $R$  is to remain constant for the shaped and unshaped systems. We found that using a rate  $R_s = 1/2$  shaping code provides good shaping gain, while allowing shaped and unshaped systems to be compared at a constant  $R$ . If the  $R$  of the shaped system is not required to match that of an unshaped system, then the optimization procedures discussed in [6] could be used to pick the optimal rate of the shaping code, which may be a value other than  $R_s = 1/2$ . When the rate of the shaping code is limited to  $R_s = 1/2$ , there are still several choices of the shaping code, i.e.  $\{n_s, k_s\}$ :  $\{2, 1\}$ ,  $\{4, 2\}$ ,  $\{6, 3\}$ ,  $\{8, 4\}$ , etc. In general, as  $k_s$  increases, so does the complexity of the shaping decoder (which is exponential in  $k_s$ ). The simulation results analyzes the different rate-1/2 shaping codes to determine the preferred choices.

#### IV. BICM-ID RECEIVER STRUCTURE

This section provides a more detailed description of the receiver than the one given in Section II-B. The section consists of three subsections, describing the demodulator, shaping decoder, and LDPC decoder, respectively. Because information is exchanged among all these components, the notion of an “iteration” may be ambiguous. In the following discussion, one full iteration is specifically meant to include an iteration of the shaping decoder, an iteration of the LDPC decoder (one iteration of each of the variable-node decoder and check-node decoder), and, for the case of BICM-ID, one iteration of symbol demapping. For the BICM system, the symbol demapper is only executed once, and information is not fed back to the demapper. Note that in [10], it is shown that feeding back information to the demapper provides little gain when gray-mapped PAM or QAM is used. However, since the present system uses APSK, a gray mapping is not possible, and BICM-ID is beneficial, as demonstrated for unshaped systems in [13] and for shaped systems in Section V.

##### A. The Demodulator

The demodulator is implemented on a symbol by symbol basis. For ease of exposition, we drop the dependence on the symbol interval in this subsection, so that symbols may

be expressed without subscripts. During a particular symbol interval, the demodulator computes the LLRs  $L_e(\mathbf{z})$  of the  $m$  code bits associated with the transmitted symbol  $x \in \mathcal{X}$ , where  $\mathcal{X}$  is the signal constellation. The inputs to the demodulator are the received discrete-time signal  $y$ , which is produced by a matched-filter front end, as well as the set of  $m$  *a priori* LLRs  $L_a(\mathbf{z})$ , which is extrinsic information generated by the shaping and LDPC decoders during the previous iteration. Prior to the first iteration,  $L_a(\mathbf{z})$  is initialized to  $L_a(\mathbf{s}_1) = \log(\frac{1-p_0}{p_0})$  for the shaped bit and  $L_a(\mathbf{s}_{2:m}) = 0$  for the unshaped bits. Since the first stream  $\mathbf{s}_1$  corresponds to the shaping code, it follows that  $p_0 > 1/2$  and  $L_a(\mathbf{s}_1)$  will initially be negative.

Let the function  $z_k(x)$  return the  $k^{th}$  bit labeling symbol  $x$ . Using the MAP demodulator described in [10] and [16], the *a posteriori* probability that  $z_k(x) = q, q \in \{0, 1\}$ , is

$$P(z_k(x) = q|y) = \sum_{x' \in \mathcal{X}_k^q} p(y|x') \prod_{j=0}^{m-1} \frac{e^{z_j(x')L_a(z_j)}}{1 + e^{L_a(z_j)}} \quad (2)$$

where  $\mathcal{X}_k^q$  is the subset of  $\mathcal{X}$  containing those signals whose  $k^{th}$  bit positions are labeled with  $q$ .

For the given channel, the conditional probability of  $y$  given  $x$  is

$$p(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-ax)^2}{2\sigma^2}} \quad (3)$$

where  $\sigma^2 = N_0/2$  is the noise variance,  $N_0$  is the one-sided power spectral density of the white Gaussian noise, and  $a$  is the fading coefficient (equal to 1 in an AWGN channel).

The demodulator output may be expressed as an LLR by combining (2) with (3) and performing some simplifications,

$$\begin{aligned} L_e(z_k) &= \ln \frac{\sum_{x' \in \mathcal{X}_k^1} p(y|x') \prod_{j=0}^{m-1} \exp(z_j(x')L_a(z_j))}{\sum_{x' \in \mathcal{X}_k^0} p(y|x') \prod_{j=0}^{m-1} \exp(z_j(x')L_a(z_j))} \\ &= \ln \frac{\sum_{x' \in \mathcal{X}_k^1} \exp\left(-\frac{(y-ax')^2}{N_0} + \sum_{j=0}^{m-1} z_j(x')L_a(z_j)\right)}{\sum_{x' \in \mathcal{X}_k^0} \exp\left(-\frac{(y-ax')^2}{N_0} + \sum_{j=0}^{m-1} z_j(x')L_a(z_j)\right)}. \end{aligned} \quad (4)$$

The above computation is facilitated by the use of the max-star operator in place of  $\ln(\sum_i e^{x_i})$  [16].

##### B. The Shaping Decoder

The first LLR stream  $L_e(\mathbf{s}_1)$  is de-interleaved and fed in to the shaping decoder as  $L_a(\mathbf{c})$ . The shaping decoder outputs the extrinsic LLRs  $L_e(\mathbf{d})$  and  $L_e(\mathbf{c})$  based on the input from the demodulator and the extrinsic information fed back from the

LDPC decoder. The implementation of the shaping decoder is similar to that of the demodulator, but the summations are now over subsets of the shaping code rather than subsets of the signal constellation. The exponent of the transition probabilities are found by taking the inner product of the  $n_s$  bit LLRs with the candidate codeword. Taking into account these differences, the output of the MAP decoder for the shaping code is

$$L_e(d_j) = \ln \frac{\sum_{\mathbf{d}' \in \mathcal{D}_j^1} \exp \left( \sum_{n=1}^{n_s} c_n(\mathbf{d}') L_a(c_n) + \sum_{\ell=1, \ell \neq j}^{k_s} d'_\ell L_a(d_\ell) \right)}{\sum_{\mathbf{d}' \in \mathcal{D}_j^0} \exp \left( \sum_{n=1}^{n_s} c_n(\mathbf{d}') L_a(c_n) + \sum_{\ell=1, \ell \neq j}^{k_s} d'_\ell L_a(d_\ell) \right)} \quad (5)$$

where  $\mathcal{D}_j^t$  denotes the set of messages  $\mathbf{d}$  whose  $j^{th}$  bit position is labeled with  $t$ ,  $t \in \{0, 1\}$ , and  $c_n(\mathbf{d})$  is the  $n^{th}$  bit in the codeword associated with message  $\mathbf{d}$ .

The extrinsic information  $L_e(c)$  produced by the shaping decoder can be implemented in a similar manner,

$$L_e(c_j) = \ln \frac{\sum_{\mathbf{c}' \in \mathcal{C}_j^1} \exp \left( \sum_{n=1}^{k_s} d_n(\mathbf{c}') L_a(d_n) + \sum_{\ell=1, \ell \neq j}^{n_s} c'_\ell L_a(c_\ell) \right)}{\sum_{\mathbf{c}' \in \mathcal{C}_j^0} \exp \left( \sum_{n=1}^{k_s} d_n(\mathbf{c}') L_a(d_n) + \sum_{\ell=1, \ell \neq j}^{n_s} c'_\ell L_a(c_\ell) \right)} \quad (6)$$

where  $\mathcal{C}_j^t$  denotes the shaping codewords  $\mathbf{c}$  whose  $j^{th}$  bit position is labeled with  $t$ ,  $t \in \{0, 1\}$ , and  $d_n(\mathbf{c})$  is the  $n^{th}$  bit in the message associated with codeword  $\mathbf{d}$ . Note that, because there are more zeros than ones in the shaping codewords,  $|\mathcal{C}_j^0| > |\mathcal{C}_j^1|$ , and therefore there are more terms in the denominator of (6) than in the numerator. This is in contrast with (5), which has the same number of terms in the numerator and denominator since the message bits are equally likely to be 0 or 1.

### C. The LDPC decoder

The decoder can be divided into two parts, a variable node decoder (VND) and a check node decoder (CND). An edge interleaver connects the VND with the CND. As shown in Fig. 3, iterative decoding is performed by passing interleaved messages between the VND and the CND until the correct codeword is found or the maximum number of iterations is reached. Each check node in the CND represents one parity-check equation; all parity check equations should be satisfied when the codeword is correctly decoded.

A variable node that connects to  $d_u$  check nodes processes information coming in from all connected check nodes and from the decoder input ( $d_u + 1$  input messages, in total). The VND simply performs the summation

$$L_e(Q_i) = L_a(u_i) + \sum L_a(r_{ji}) \quad (7)$$

where  $L_a(r_{ji})$  is the *a priori* LLR coming in from the  $j^{th}$  check node, the summation is over the incoming edges, and  $L_a(u_i)$  is the LLR of code bit  $u_i$  at the input of the LDPC decoder.  $L_e(Q_i)$  is a LLR that represents the *a posteriori* probability of the  $i^{th}$  bit  $u_i$  of the LDPC codeword. During each iteration, a hard decision is made on  $L_e(Q)$  to produce an estimate of the codeword and the corresponding message. Decoding can halt once a correct codeword has been found

The input to the  $j^{th}$  check node is the extrinsic information, which may be found by subtracting the message coming in from check node  $j$  from the LLR  $L_e(Q_i)$ ,

$$\begin{aligned} L_e(q_{ij}) &= L_e(Q_i) - L_a(r_{ji}) \\ &= L_a(u_i) + \sum_{j' \neq j} L_a(r_{j'i}) \end{aligned} \quad (8)$$

where  $L_e(q_{ij})$  is the extrinsic information that is passed from the  $i^{th}$  variable node to the  $j^{th}$  check node and the summation is over all edges except for the one connecting the variable node to the  $j^{th}$  check node, i.e. all incoming edges  $j' \neq j$ . The subtraction operation in the first line of (8) is shown in Fig. 3. The CND essentially decodes the single parity-check code associated with each row of the parity-check equation. While there are several ways to formulate the CND decoder, we use a formulation described in [17] which is not be discussed here.

## V. NUMERICAL RESULTS

In this section, we compare the bit-error performance of bit-interleaved LDPC coded APSK both with and without shaping. The APSK modulation and LDPC coding are as specified in the DVB-S2 standard [1], with 32-APSK modulation used along with the length  $n_c = 64,800$  LDPC code. The amplitudes of the 32-APSK constellation are selected such that the middle ring's radius is 2.64 times the inner ring's radius, and the outer ring's radius is 4.64 times the inner ring's radius. The system uses only one shaping bit ( $g = 1$ ) which we previously showed to strike a balance between performance and system complexity [6]. The overall system rate is 3 bits/symbol ( $R = 3$ ), in which case it is easy to identify shaping codes whose rates  $R_s$  are compatible with the limited LDPC rates  $R_c$  defined in the standard. To ensure  $R = 3$  both with and without shaping,  $R_c = 3/5$  is used for the uniform (unshaped) system, and  $R_s = 1/2$  and  $R_c = 2/3$  are used for the shaped system.

The bit-error rates for both AWGN and ergodic (fully-interleaved) Rayleigh fading channels are shown in Figs. 4 and 5, respectively. By comparing the two curves on the right of Fig. 4, it is clear that BICM-ID provides a gain of about 0.3 dB over BICM in an AWGN channel without shaping. Comparing the curve for unshaped BICM-ID (second from

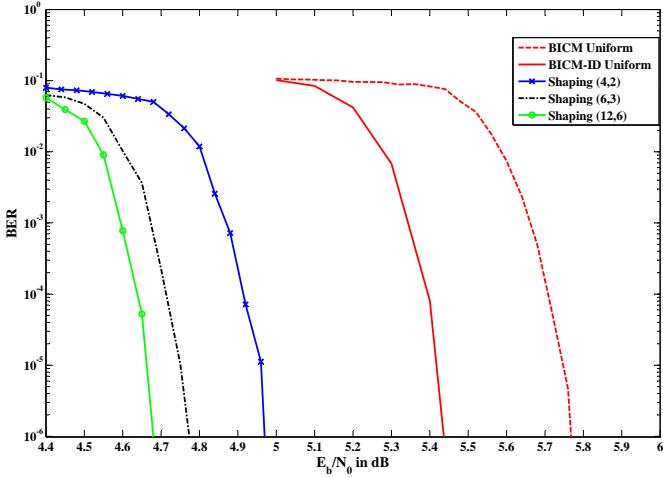


Fig. 4. Bit-error rate of LDPC-coded 32-APSK in AWGN at rate  $R = 3$  bits/symbol both with and without shaping. Curves are shown for the unshaped (uniform) system using BICM and BICM-ID. Curves are shown for three shaping codes, and the shaped system uses BICM-ID.

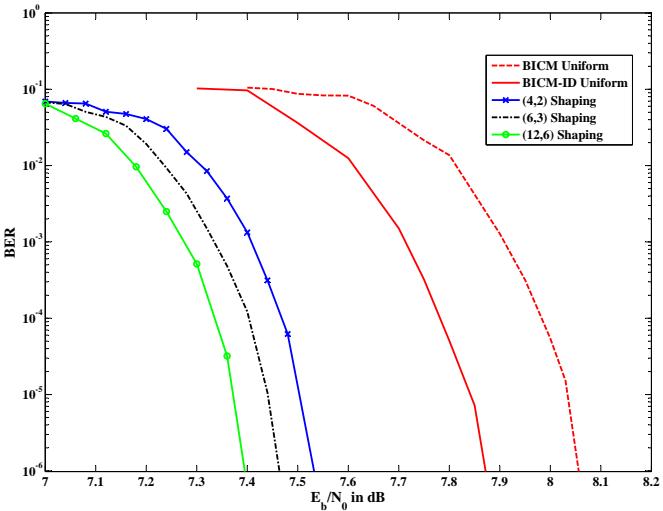


Fig. 5. Bit-error rate for the same system described in Fig. 4 in fully interleaved Rayleigh fading.

right) with the curves for shaping (the three curves on the left), it can be seen that shaping provides gains between 0.45 dB to 0.7 dB, depending on the shaping code used. Combining both gains (shaping and BICM-ID gains), a gain of over 1 dB is observed compared to the uniform BICM system. Over the Rayleigh fading channel, BICM-ID provides a gain of 0.17 dB over the BICM system, and the additional shaping gain ranges from 0.4 dB to 0.5 dB. Both figures show that longer shaping codewords provide higher shaping gains, at the cost of a more complex decoder. When rate-1/2 shaping is required, the (6, 3) shaping code gives a good tradeoff between performance and complexity. Since such a code only contains  $2^3 = 8$  codewords, its decoding complexity is quite low.

## VI. CONCLUSION

Thanks to its high coding gain and availability of COTS equipment, the DVB-S2 standard is a viable option for military

and other communication links. By incorporating constellation shaping along with iterative demodulation and decoding, an additional decibel of coding gain can be readily attained at a rate of 3 bits per symbol over an AWGN channel. Gains can also be achieved over Rayleigh fading channels and at other transmission rates. Due to the inclusion of a nonlinear shaping code and its corresponding decoder, the proposed system has a higher per-iteration complexity than a standard DVB-S2 system. However, by carefully choosing parameters, a shaping code that provides a good tradeoff between performance and complexity can be identified.

## REFERENCES

- [1] European Telecommunications Standards Institute, "Digital video broadcasting (DVB) second generation: Framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications," *ETSI EN 302 307 version 1.2.1*, Aug. 2009.
- [2] B. Bennett, D. Hannan, G. Fitzgerald, G. Kinal, J. Marshall, and R. Gibbons, "Performance of GBS over WGS1, 2, and 3 using DVB-S and DVB-S2," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, (Boston, MA), Nov. 2009.
- [3] F. De Rango, A.-F. Santamaria, M. Tropea, F. Veltri, L. Belcastro, and S. Marano, "An enhanced two-stages packet scheduler for DVB-S2 satellite system based on adaptive strategies," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, (Boston, MA), Nov. 2009.
- [4] C. Timmerman, M. Benson, J. Delisle, J. Delva, R. Elliott, J. Hillger, A. Miller, T. Brick, J. Long, and N. Humphrey, "Ground based high data rate DVB-S2 demodulator for high data rate AISR transport," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, (San Jose, CA), Nov. 2010.
- [5] R. Shoup, N. List, A. Fletcher, and T. Royster, "Using DVB-S2 over asymmetric heterogeneous optical to radio frequency satellite links," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, (San Jose, CA), Nov. 2010.
- [6] M. Valenti and X. Xiang, "Constellation shaping for bit-interleaved coded APSK," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, (Kyoto, Japan), June 2011.
- [7] G. D. Forney Jr. and L. F. Wei, "Multidimensional constellations – part 1: Introduction, figures of merit, and generalized cross constellations," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 877–892, Aug. 1989.
- [8] G. D. Forney Jr., "Trellis shaping," *IEEE Trans. Inform. Theory*, vol. 38, pp. 281–300, Mar. 1992.
- [9] A. K. Khandani and W. Tong, "Application of shaping technique with turbo coset codes," *IEEE Trans. Veh. Tech.*, vol. 56, pp. 3770–3779, Nov. 2007.
- [10] S. Y. Le Goff, B. K. Khoo, and C. C. Tsimenidis, "Constellation shaping for bandwidth-efficient turbo-coded modulation with iterative receiver," *IEEE Trans. Wireless Comm.*, vol. 6, pp. 2223–2233, Jun. 2007.
- [11] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, pp. 927–946, May 1998.
- [12] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding using soft feedback," *Electronics Letters*, vol. 34, pp. 942–943, May 1998.
- [13] Q. Xie, K. Peng, J. Song, and Z. Yang, "Bit-interleaved LDPC-coded modulation with iterative demapping and decoding," in *Vehicular Technology Conference 2009*, (Barcelona, Spain), April 2009.
- [14] M. Yang, W. E. Ryan, and Y. Li, "Design of efficiently encodable moderate-length high-rate irregular LDPC codes," *IEEE Trans. Commun.*, vol. 52, pp. 564–571, Apr. 2004.
- [15] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, pp. 670–678, Apr. 2004.
- [16] M. C. Valenti and S. Cheng, "Iterative demodulation and decoding of turbo coded  $M$ -ary noncoherent orthogonal modulation," *IEEE J. Select. Areas Commun.*, vol. 23, Sept. 2005.
- [17] D. J. C. MacKay, "Good error correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, Mar. 1999.