Concatenated Signal Codes with Applications to Compute and Forward

Nihat Engin Tunali and Krishna R. Narayanan
Department of Electrical and Computer Engineering
Texas A&M University
{engintunali@neo.tamu.edu, krn@ece.tamu.edu}

Abstract—We present a new coding scheme based on concatenating a newly introduced class of lattice codes called signal codes with interleaved Low Density Parity Check (LDPC) codes. These codes are shown to possess a special algebraic structure which makes them suitable for recovering linear combinations (over a finite field) of the transmitted signals in a multiple access channel. This facilitates their use as a coding scheme for the recently proposed compute and forward paradigm. The decoding algorithm is based on an appropriate combination of the stack decoder with a message passing algorithm. Simulation results show that our proposed scheme can approach the uniform input AWGN capacity within 1.5 db, which is a 2 db improvement compared to using only signal codes when decoding using a stack algorithm with the same stack size. Simulation results for our proposed scheme applied to compute and forward are also presented.

Index Terms—Compute and Forward, Lattice codes, Signal codes, LDPC codes, Bidirectional Relaying.

I. INTRODUCTION

Consider the following multiple access channel (MAC) which may be a part of a wireless network. Let the two transmitters Tx1 and Tx2 have data \( u_1 \) and \( u_2 \) which are encoded into \( x_1 \) and \( x_2 \) and transmitted over the MAC channel. Let \( c_1 = c(u_1) \) be the codeword corresponding to \( u_1 \). Say the receiver (a relay node perhaps) receives \( y = x_1 + x_2 + w \), where \( w \) is a noise vector. Compute and Forward is a new paradigm in wireless networks where we attempt to recover a function \( f(u_1, u_2) \) at the relay from \( y \) [1], [2], [3].

A class of codes that are suited to implement compute and forward are lattice codes where \( x_1 = G_{u_1} \) and \( x_2 = G_{u_2} \), where \( u_1 \) and \( u_2 \) are integer vectors and \( G \) is the lattice generator matrix. This is because, \( c(x_1 + x_2) = x_1 + x_2 \) which is also another codeword from the same lattice. Therefore, we can choose the function \( f(u_1, u_2) = u_1 + u_2 \). However, the construction of good lattice codes with an efficient shaping and decoding algorithm is still not well understood.

Inspired by this property of lattice codes, we define the pseudo-lattice property as follows. Let \( u_1 \) and \( u_2 \) be two information vectors chosen from an \( m \) dimensional finite field \( \mathbb{F}_m \). Furthermore, assume that \( \mathbb{F} \) is a Galois field \( \text{GF}(M) \) where \( M \) is prime and addition is defined as integer addition modulo \( M \). Afterwards, \( u_1 \) and \( u_2 \) are encoded to codewords \( x_1 \) and \( x_2 \) respectively where \( x_1, x_2 \in \mathbb{R}^n \). We say that a code satisfies a pseudo-lattice property if

\[
c(u_1 + u_2) = c((u_1 + u_2) \mod M) = (c(u_1) + c(u_2)) \mod M
\]

where \( \oplus \) represents modulo-\( M \) addition, i.e., addition in \( \text{GF}(M) \). In this case, we can choose \( f(u_1, u_2) = u_1 + u_2 \) and recover this from \( y \mod M \). This algebraic structure is a special case of that considered in [3].

Before we proceed, we quickly review some recent developments in the construction of practically decodable lattice codes. In [4], a new class of lattice codes called Low Density Lattice Codes (LDLC) have been introduced and shown to have an error rate that decreases with length and have reasonable decoding and shaping complexities. However, LDLC suffers greatly from performance degradation at low rates and the message passing algorithm that is used to decode them is still computationally demanding for high dimensions. Another recently introduced class of lattice codes are signal codes [5] which seem to be an interesting alternative to LDLC. However, their performance does not improve with length since their structure is identical to that of convolutional codes. In this work, we have introduced a new class of codes that are based on the concatenation of signal codes and LDPC codes which also have low decoding complexity. This new class of codes do not suffer from the drawbacks of LDLC or signal codes and they satisfy the pseudo lattice property which make them a perfect candidate for a compute and forward coding scheme in the bidirectional relay network.

The outline of the paper is as follows. We first give some background on signal codes and how to implement shaping. Then we propose the encoding and decoding scheme of our new class of codes which is based on concatenating signal codes with LDPC codes. Afterwards, we extend our coding scheme such that it can be implemented to compute and forward. Finally, we present simulation results for the point to point case and an application to compute and forward.

II. BACKGROUND ON SIGNAL CODES

We will begin with a brief review of the encoding, shaping and decoding procedures for signal codes.

A. Encoding and Shaping

Let \( a = \{a_n\} \) denote a sequence of \( M^2 \)-QAM symbols where \( a_n = a_nI + j a_nQ \) with \( a_nI, a_nQ \in \{0, 1, \ldots, M - 1\} \).

A (unshaped) codeword of the signal code is obtained by
convolving $a$ with a monic causal filter with the transfer function $F(z) = 1 + \sum_{i=1}^{L} f_{i} z^{-i}$. That is,

$$x_{n} = a_{n} + \sum_{i=1}^{L} f_{i} a_{n-i}$$  \hspace{1cm} (1)$$

for $n = 0, 1, \ldots, N + L - 1$ and $a_{n}$ is assumed to be zero outside the range $n = 0, 1, \ldots, N - 1$. It can also be seen that $w = G_{a}$ where $a$ is an $N$ dimensional column vector, $w$ is an $N + L$ dimensional column vector and $G$ is the $(N + L) \times N$ generator matrix as mentioned in [5]. Since the columns of $G$ are linearly independent and $w = G_{a}$ is a complex integer linear combination of the columns of $G$, the set of all $w$ namely $\mathcal{X}$ forms a lattice code over $\mathbb{C}^{N+L}$. More specifically, it is a subset of an infinite lattice over $\mathbb{C}^{N+L}$.

It has been shown in [5] that the minimum distance between any two codewords in $\mathcal{X}$ is higher than that of uncoded $M^{2}$-QAM. However, this comes at the cost of a higher average power of $\mathcal{X}$. In order to reduce the average transmit power and maintain a power constraint on the codeword $w$, hypercube shaping based on Tomlinson-Harashima precoding can be implemented as follows [6]. Recall that $a_{n}$ belongs to an $M^{2}$-QAM constellation. The shaping operation maps each $a_{n}$ to $b_{n}$ via

$$b_{n} = a_{n} - M(k_{n})$$  \hspace{1cm} (2)$$

where $k_{n}$ is a complex integer. $k_{n}$ is determined according to

$$k_{n} = \left\lfloor \frac{1}{M} \left( a_{n} + \sum_{i=1}^{L} f_{i} b_{n-i} \right) \right\rfloor$$  \hspace{1cm} (3)$$

and $\lfloor x \rfloor$ denotes the complex integer closest to $x$. After $k_{n}$ and $b_{n}$ have been computed, $x_{n}$ can be computed as

$$x_{n} = b_{n} + \sum_{i=1}^{L} f_{i} b_{n-i}$$  \hspace{1cm} (4)$$

which is nothing but the equivalent of $w = G_{b}$. This shaping method ensures that each element $x_{n}$ of the codeword $w$ lies within the interval $[-M/2, M/2]$, i.e a hypercube, and is uniformly distributed which results in an average power of $\frac{1}{M} M^{2}$. Notice that $a_{n}$ can be determined uniquely from $b_{n}$ by a modulo $M$ operation.

B. Decoding

Assume that each $x_{n}$ is transmitted over an AWGN channel such that $y_{n} = x_{n} + w_{n}$ where $w_{n}$ are zero mean, i.i.d Gaussian random variables. In order to decode to the corresponding $a$, the maximum likelihood decoder should maximize

$$L(y|a) = -\sum_{n} \left| y_{n} - \sum_{i=0}^{L} f_{i} b_{n-i}^{a} \right|^{2}$$  \hspace{1cm} (5)$$

where $b_{n}^{a}$ is the sequence of shaped symbols that corresponds to $a$. In the decoding algorithm proposed in [5], the $b_{n}$’s are treated as free variables and the new likelihood decoder maximizes

$$L(y|b) = -\sum_{n} \left| y_{n} - \sum_{i=0}^{L} f_{i} b_{n-i} \right|^{2}$$  \hspace{1cm} (6)$$

The decoder is the equivalent of decoding to the nearest lattice point without taking the shaping region into account. For lattice codes of large dimensions, disregarding the shaping algorithm will not affect the performance of the decoder.

It can be observed that there is significant similarity in the structure of convolutional codes and signal codes. Just as convolutional codes, signal codes can be decoded using the Viterbi Algorithm, which would have states depending on $\{b_{n-1}, \ldots, b_{n-L}\}$. In other words, the number of trellis branches would be equal to the constellation size of $b_{n}$ to the power $L$ which is at least $M^{L}$. However, the shaping method that we use increases the constellation size of $b_{n}$ to a value much larger than $M$. Therefore, the Viterbi Algorithm would have an extremely high computational complexity.

Therefore, in [5] suboptimal decoders can be implemented quite efficiently resulting in a negligible performance degradation. One of these suboptimal decoders is the stack decoder which stores the candidate $b_{n}$’s in a stack and updates the stack after each step by sorting the metrics of the candidates and only allowing $L'$ of them to remain in the stack where $L'$ is the maximal stack length [7]. The Fano metric to be used in the stack decoder with Tomlinson-Harashima shaping has been derived in [5] and is given by

$$L(y|b) = -\sum_{n} \left[ y_{n} - \sum_{i=0}^{L} f_{i} b_{n-i} \right]^{2} - B$$  \hspace{1cm} (7)$$

and

$$B \approx \sigma^{2} \cdot \log \frac{4}{\sigma^{2}}$$  \hspace{1cm} (8)$$

where $B$ is the bias term [5].

Since the structure of signal codes is similar to that of convolutional codes, it can be seen that the symbol error rate for signal codes does not improve as their dimension (length) is increased which is an undesirable property. In order to overcome this disadvantage, signal codes can be concatenated with LDPC codes. However, a straightforward concatenation would make it difficult to implement a good decoding scheme. This is due to the fact that when the stack decoder fails, it results in bursty errors which cripples the error correcting capability of the LDPC code. This has motivated us to develop a more sophisticated concatenation scheme and an appropriate decoding scheme which are discussed below.

III. PROPOSED SCHEME

A. Encoding and Shaping

Our proposed encoder structure is shown in Fig. 1. The information symbols are inserted into a $K \times N_{2}$ matrix $U$ which will be referred as the information symbol matrix. The $(i,j)$th entry $u_{i,j}$ is chosen from an $M^{2}$-QAM constellation such that $\Re(u_{i,j}), \Im(u_{i,j}) \in \text{GF}(M) = \{0, 1, \ldots, M - 1\}$ where $M$ is prime. Then, an LDPC code of rate $r = K/N_{1}$ over $\text{GF}(M)$ with a corresponding $N_{1} \times K$ generator matrix $G_{L}$ is constructed. The real and imaginary components of the $j$th column of the $U$ matrix, namely, $u_{j}$ are separately encoded.
into LDPC codewords using $G_L$ and the corresponding complex LDPC codewords $C_j$ can be inserted as column vectors to an $N_1 \times N_2$ matrix $C$. $C$ will be referred as the LDPC codeword matrix. Taking into account that addition in $GF(M)$ is defined as addition modulo $M$, we will represent the entire encoding operation using matrix notation as follows:

$$C = G_L U \mod M$$  \hspace{1cm} (9)

It should be understood that the above notation means that each column $C_j$ is obtained by multiplying each column $u_j$ by $G_L$. Then using a signal code with transfer function $F(z) = 1 + \sum_{i=1}^{L} f_i z^{-i}$ and implementing hypercube shaping as mentioned in Section II-A, each row of $C$ denoted as $c_T^T$ is encoded and shaped into a row vector $z_T^T$ where each $x_{i,j}$ belongs to the interval $[-\frac{M}{2}, \frac{M}{2}]$. In order to terminate the shaping operation, the last $L - 1$ symbols in each $z_T^T$ can be chosen directly from the $M^2$-QAM constellation and protected with a lower rate LDPC code in order to ensure reliability. Note that for long block lengths the rate loss would be negligible. Finally, all $z_T^T$’s are inserted as row vectors into an $(N_1 + L) \times N_2$ matrix $X$ which will be referred as the signal codeword matrix. Then, $X$ is transmitted over an AWGN channel where a noise matrix $N$ is added to $X$, each element $n_{i,j}$ being i.i.d Gaussian random variables, which results in $X = X + N$. $X$ will be referred as the signal codeword matrix. The corresponding encoding operation for encoding $C$ to $X$ is as follows.

$$X = (C + MK) G_{SC}^T \hspace{1cm} (10)$$

Where $G_{SC}$ is the $(N_2 + L) \times N_2$ generator matrix of the Signal Code, constructed identically as in section II-A and $K$ is a complex integer matrix which is a result of the shaping operation . The complete encoding operation can be summarized by combining equations (10) and (11) as:

$$X = \left((G_L U \mod M) + MK\right) G_{SC}^T \hspace{1cm} (11)$$

### B. Pseudo-Lattice Property

This new class of codes satisfies the pseudo-lattice property which can be explained as follows. Assume that we have two information symbol matrices $U_1$ and $U_2$ and they are encoded into LDPC codeword matrices $C_1$ and $C_2$ and then into signal code matrices $X_1$ and $X_2$, respectively. The sum of $X_1$ and $X_2$ can be expressed as

$$X_1 + X_2 = \left(((C_1 + C_2) + (K_1 + K_2))\right) G_{SC}^T = \left(((C_1 + C_2) \mod M) + (K_1 + K_2)\right) G_{SC}^T \hspace{1cm} (12)$$

where

$$K = \frac{(C_1 + C_2 - ((C_1 + C_2) \mod M))}{M} + K_1 + K_2 \hspace{1cm} (13)$$

and

$$\left((C_1 + C_2) \mod M = \frac{\left(G_L \left(\left(U_1 + U_2\right) \mod M\right)\mod M\right)}{M}\right) \hspace{1cm} (14)$$

Note that $C_1 + C_2$ is not a valid LDPC codeword matrix. However if $(C_1 + C_2) \mod M$, which is a valid LDPC codeword matrix, can be obtained from $X_1 + X_2$ the corresponding information symbol matrix would be $\left(U_1 + U_2\right) \mod M$. Hence, $U_1$ or $U_2$ can be recovered from $(C_1 + C_2) \mod M$ with the knowledge of $U_2$ or $U_1$.

### C. Decoding

The proposed scheme can be decoded efficiently as follows. Once all rows are received, $N_1 + L$ stack decoders, which were mentioned in section II-B, are run in parallel. Once the stack decoder reaches a depth of $\tau \approx 25$ a hard decision is made on the $1st$ symbol for each row, thus making a hard decision on the $1st$ column. Taking into account that the real and imaginary parts of every column are LDPC codewords, an LDPC decoder using a message passing algorithm is used to decode the $1st$ column. Assuming that the LDPC code is operating above an SNR threshold such that we can decode with arbitrarily small error probability, the $1st$ column can be estimated correctly with high probability. Therefore, it can now be assumed that the first $1st$ column is known. Then, $N_1 + L$ stack decoders are extended in parallel under the assumption that the first column is known and once hard decisions are made on each symbol in the $2nd$ column, an LDPC code is used to decode the $2nd$ column. This process is repeated iteratively until all $N_2$ columns are decoded. The schematic for such a decoder is shown in Fig. 2.

The encoder and decoder structure proposed here are motivated by results on the optimality of this structure for achieving the uniform input capacity for inter-symbol interference channels in [8],[9] and [10]. The exact schedule of incrementing the stack decoder followed by decoding one column of the LDPC decoder is key in ensuring that good performance can be obtained. The main advantage of this scheme is as follows: when the $jth$ column is being decoded, it can be assumed that the previous $j - 1$ columns have been successfully decoded thereby ensuring that when decoding $c_{i,j}$ , the entire sequence $c_{i,1}, c_{i,2} \cdots c_{i,j-1}$ is known. Therefore the stack decoder, which is a suboptimal decoder, is forced towards extending the correct path and performing much closer to a maximum-likelihood decoder. That is, the error correction capability of
the LDPC code is used in enhancing the performance of the stack decoder.

In the scheme that we have proposed, it is crucial that the LDPC code is able to correct all errors with a high probability in each column of C where the stack decoder fails. The reason for this is that unless we have a good estimate of every column, it is not feasible to assume it has been decoded correctly and once the stack decoder moves to the following column in the next iteration, it is impossible to correct the errors in the previous column and the decoding algorithm fails. For LDPC codes of long length, there will be a sharp SNR threshold which above the threshold they have almost no error capability and below the threshold they have almost no error capability. In order to determine these SNR thresholds, we proceed as follows.

After the stack decoder attempts to decode to each element of the column \( \mathbf{c}_j^T \), these estimates \( \Re(\hat{c}_{i,j}) \) and \( \Im(\hat{c}_{i,j}) \) are fed into two LDPC decoders. Therefore, these LDPC decoders see an equivalent discrete memoryless channel for every \( c_{i,j} \). The equivalent channel can be determined empirically through Monte-Carlo simulations. This equivalent channel is used to calculate the probability \( p(\Re(\hat{c}_{i,j})|\Re(c_{i,j})) \) and \( p(\Im(\hat{c}_{i,j})|\Im(c_{i,j})) \) for all M possible values of \( \Re(c_{i,j}) \) and \( \Im(c_{i,j}) \). Then an M dimensional probability density function is constructed from these values which is passed to the LDPC decoders in order to initialize the message passing algorithm. These M dimensional probability density functions are functions of SNR and below a certain SNR, they will not be sufficient for the LDPC codeword to converge to the correct codeword with high probability.

Due to the fact that performing density evolution for non-binary LDPC codes over discrete memoryless channels is computationally quite challenging, we have used LDPC codes of codeword length \( 10^5 \) in order to determine thresholds. If the message passing decoder succeeds, it is assumed that the SNR is above the threshold level and if it does not, it is assumed that the SNR is below the threshold level. By gradually adjusting SNR thresholds were determined up to 3 significant digits. Note that this method for determining thresholds can be used only for LDPC codes with very long lengths.

IV. EXTENDING THE SCHEME TO COMPUTE AND FORWARD

Motivated by our scheme’s performance for the point-to-point case and the pseudo-lattice property it satisfies, we extend our scheme such that it can be implemented for the compute and forward problem in the bidirectional relay network as follows.

During the multiple access phase, the two nodes encode their information data matrices \( \mathbf{U}_1 \) and \( \mathbf{U}_2 \) to signal codes matrices \( \mathbf{X}_1 \) and \( \mathbf{X}_2 \) respectively via the same LDPC code generator matrix \( \mathbf{G}_L \) and Signal Code generator matrix \( \mathbf{G}_{SC} \) as mentioned before and transmit them to the relay.

Once the relay receives \( \mathbf{X}_1 + \mathbf{X}_2 + \mathbf{N} \), it will attempt to decode to \( (\mathbf{C}_1 + \mathbf{C}_2) \mod M \) as seen in (12). The stack decoder in section II-B should be adapted to the distribution of \( \mathbf{X}_{1,j} + \mathbf{X}_{2,j} \), which is triangularly distributed within the interval \([-M, M] \). Unfortunately, we do not provide more details regarding to the implementation of the modified stack decoder due to space limitations.

If the relay succeeds in decoding, it can encode \( (\mathbf{C}_1 + \mathbf{C}_2) \mod M \) to a signal codeword matrix \( \mathbf{X}_{BC} \) as \( \mathbf{X}_{BC} = \left( [\mathbf{C}_1 + \mathbf{C}_2] \mod M \right] + MK)\mathbf{G}_{SC}^T \) and broadcast this to both nodes.

Upon receiving \( \mathbf{X}_{BC} \), the nodes would attempt to decode to \( (\mathbf{C}_1 + \mathbf{C}_2) \mod M \). If they succeed, they can subtract their own codeword matrix from \( \mathbf{C}_1 + \mathbf{C}_2 \mod M \) and eventually recover the other node’s codeword matrix and hence information symbol matrix.

V. SIMULATION RESULTS

We shall now demonstrate the performance of our scheme for the point to point case using simulations. Simulations were carried out for an \( M^2 \)-QAM constellation which each information symbol matrix entry \( c_{i,j} \) was chosen from \( M = 3 \). The red line represents AWGN capacity with uniform input constraint due to the fact that hypercube shaping results in \( x_{i,j} \) to be uniformly distributed between \([-\frac{M}{2}, \frac{M}{2}]\). SNR is defined as \( SNR = \frac{E[(x_{i,j})^2]}{\sigma^2} \) where \( E[(x_{i,j})^2] = \frac{M^2}{6} \) which is a result of uniform input distribution and noise variance is \( 2\sigma^2 \), which is a result of complex AWGN. Therefore \( SNR = \frac{M^2}{12\sigma^2} \). A filter pattern of \( F(z) = \left( 1 + 0.98z^{-1} + 0.9z^{-2}\right)^3 \), which was shown to be a good choice of a monic causal filter in [5], was used for encoding the LDPC codeword matrix \( C \) to the signal codeword matrix \( \mathbf{X} \) and a block length of 2000 was used for each row of \( \mathbf{X} \). The dashed line represents the mutual information \( I(\hat{c}_{i,j};c_{i,j}) \), which is referred as the equivalent channel capacity, for every SNR where \( \hat{c}_{i,j} \) is the estimate of \( c_{i,j} \) using a stack decoder of stack length 1000 that reaches a depth of \( \tau \approx 25 \). Finally, LDPC codes of dimension \( n = 100000 \) over \( GF(M) \) were constructed and a regular degree distribution of \( (3, k) \) was chosen. \( k \in \{3, 6, 9, 12, 30\} \) was used to simulate LDPC codes with rates \( R \in \{\frac{1}{3}\log_2 M, \frac{2}{3}\log_2 M, \frac{1}{2}\log_2 M, \frac{3}{4}\log_2 M, \frac{5}{6}\log_2 M, \frac{7}{8}\log_2 M\} \) respectively and the SNR thresholds are determined as mentioned in section III-C. At this threshold or above, the LDPC
code will be able to decode each column $c_j^T$ correctly with a very high probability, thus it can be concluded that transmission is reliable at this threshold. Note that the effective rate of the scheme will be $2R$ because two LDPC codes are used for the real and imaginary part of each column. The rate $2R$ and the corresponding SNR threshold are marked by ‘×’ in the plots.

As seen from the results, our scheme can perform as close as 1.5 db away from the AWGN capacity with uniform input constraint in the higher SNR regime for the point-to-point case. However, at low SNRs the stack decoder gives a bad estimate of the symbol $c_{i,j}$, therefore the equivalent channel capacity reduces significantly resulting in performance degradation. The thresholds of the LDPC codes stand at a distance of 0.5 to 0.8 db away from the SNR of the equivalent channel capacity. Therefore, our scheme performs the best where the distance between equivalent channel capacity and uniform AWGN capacity is minimal. Comparing our results to plain signal codes, our scheme attains arbitrarily low symbol error rate as close as 1.5 db away from uniform AWGN capacity with a stack length of $10^3$ where as the plain Signal Code scheme attains frame error rate of $10^{-3}$ at a distance of 3.5 db away from uniform AWGN capacity with the same stack length [5]. Furthermore, plain signal codes require a stack length of $10^6$ in order to match our scheme’s performance which is far more computationally demanding.

We have also obtained simulation results for our scheme applied to compute and forward. The specifications of our scheme applied to compute and forward are identical with the point to point case. At low rates our proposed scheme requires an additional 1.6 db in order to reliably transmit data over the MAC channel whereas at higher rates the gap drops down to 1.3 db. We believe there are two main reasons for this gap: the decision region in decoding to the modulo sum results in higher error rates and the performance of the stack decoder may deteriorate. These aspects are currently under investigation.

![Graph](image-url)  
Fig. 3. Performance of the proposed scheme over 9-QAM for the point-to-point case

![Graph](image-url)  
Fig. 4. Performance of the proposed scheme applied to compute and forward

VI. CONCLUSION AND FURTHER IMPROVEMENTS

A new coding scheme based on concatenating signal codes with LDPC codes was introduced. Simulation results have shown a 2 db improvement between the proposed scheme and plain signal codes of the same stack length. Good asymptotic performance can be achieved due to the use of LDPC codes in the proposed scheme. By optimizing degree distributions for the LDPC codes, there is a potential for further improvement in the performance which is left for future studies. Furthermore, the pseudo-lattice property that the proposed scheme satisfies makes it an excellent application to the compute and forward problem in the bidirectional relay network where simulation results show a 1.3-1.6 db gap between the point-to-point scheme and the MAC channel. We hope to bridge this gap in our future work using more suitable shaping techniques and decoding algorithms.

REFERENCES