

“ Time-Varying Periodic Convolutional Codes With Low-Density
Parity-Check Matrix” by Alberto Jimenez Felstrom and Kamil
Sh. Zigangiro

Kwame Ackah Bohulu

July 12, 2019

Abstract- We present a class of convolutional codes defined by a low-density parity-check (LDPC) matrix and an iterative algorithm of the decoding of these codes. The performance of this decoding is close to the performance of turbo decoding. Our simulation shows that for the rate $R = 1/2$ binary codes, the performance is substantially better than for ordinary convolutional codes with the same decoding complexity per information bit. As an example, we constructed convolutional codes with memory $M = 1025, 2049$ and 4097 showing that we are about 1 dB from the capacity limit at a bit-error rate (BER) of 10^{-5} and a decoding complexity of the same magnitude as a Viterbi decoder for codes having memory $M = 1$

0.1 Introduction

In the 1960's Gallager first introduced LDPC block codes and also studied an iterative decoding algorithm which showed promise in achieving low error probabilities at high data rates with very reasonable costs.

Iterative decoding has since gained a lot of attention[2] -[4] especially with the recent simulation results [5],[6] which revealed that LDPC codes have performance close to the Shannon limit, which is also similar to turbo codes.

This paper focuses on the convolutional version of LDPC codes. It is revealed that they perform better than block codes of the same length and have a much regular structure which makes encoder and decoder implementation easier. Decoding can also be sped up by using a pipeline structure involving several processors.

Due to statistical dependence, a mathematical performance analysis of iterative decoding is generally difficult. A successful attempt of LDPC iterative decoding was made by V. Zyablov and M. Pinsker [7]. They were able to prove that the number of corrected errors increases linearly with block length. In practise however BER is of more interest and as far as the authors know, most of the BER analysis that exists is based on simulation results.

0.2 LDPC Block Codes and Iterative Decoding

An (N, J, K) LDPC matrix \mathbf{H} of a block code over $\text{GF}(2)$ is a matrix with N columns, each column having weight J and each row having weight K . In general \mathbf{H} can be represented as shown in 0-1

$$\mathbf{H} = \begin{bmatrix} \mathbf{M}\mathbf{P}_1 \\ \mathbf{M}\mathbf{P}_2 \\ \vdots \\ \mathbf{M}\mathbf{P}_J \end{bmatrix} \quad (0-1)$$

where

\mathbf{M} is an $\frac{N}{K} \times N$ matrix made up of K concatenated $\frac{N}{K}$ unity matrices \mathbf{I}

\mathbf{P} is any $N \times N$ permutation matrix, so \mathbf{H} has $\frac{NJ}{K}$

The rate R of the code is $R \geq 1 - \frac{J}{K}$ since not all the rows are linearly independent. It is worth mentioning that in principle, it is possible to construct LDPC codes with different column weight J and or row weights K

Using a slightly different construction, Gallager[1] proved the existence of LDPC codes whose minimum distance d_{min} increases linearly with block length N when $J > 2$. Furthermore, he

proved that for $J > 2$, the ratio of the minimum distance to the code length of a typical code limits to a nonzero constant α , which is dependent on the selection J and K

We continue by considering the case where an LDPC codeword over GF(2) is $\mathbf{v} = [v_1 v_2 \dots v_N]$ is transmitted over an AWGN channel with one sided power spectral density N_0 . Code symbols are mapped into the signal points

$$w_n = (1 - 2v_n) \quad (0-2)$$

The sequence $\mathbf{w} = [w_1 w_2 \dots w_N]$ is in turn modulated using Nyquist pulses of unit energy. Using a matched filter at the receiver, the sampled filter output r_n has the conditional probability density function .

$$p(r_n | w_n) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(r_n - w_n)^2}{N_0}} \quad (0-3)$$

Making the assumption that the code symbols v_n are random and equiprobable, the log-likelihood metrics for the received values can be written as

$$\ell(n) \stackrel{\text{def}}{=} \log \frac{\Pr(v_n = 0 | r_n)}{\Pr(v_n = 1 | r_n)} = 4r_n \frac{RE_b}{N_0} \quad (0-4)$$

Since decoding is done using an iterative algorithm, the log-likelihood metric below is of interest

$$\begin{aligned} \ell^{(1)}(n) &\stackrel{\text{def}}{=} \log \frac{\Pr(v_n = 0 | r_n, \{\mathcal{C}_j(n)\}_{j=1}^J)}{\Pr(v_n = 1 | r_n, \{\mathcal{C}_j(n)\}_{j=1}^J)} \\ &= \ell(n) + \sum_{j=1}^J \log \frac{1 + \prod_{n' \in \mathcal{C}_j(n)} \Lambda(n')}{1 - \prod_{n' \in \mathcal{C}_j(n)} \Lambda(n')} \end{aligned} \quad (0-5)$$

where

$$\Lambda(n) \stackrel{\text{def}}{=} \frac{e^{\ell(n)} - 1}{e^{\ell(n)} + 1} = \tanh(\ell(n)/2)$$

and

$$\mathcal{C}_j(n) \stackrel{\text{def}}{=} \{i : \bar{h}_{ji} = 1\}, j = 1, 2, \dots, J$$

also \bar{h}_{ji} refers to the element at row j , column i of \mathbf{H} where $h_{jn} = 1$ and

$$\Pr(v_n = \pm 1 | r_n, \{\mathcal{C}_j(n)\}_{j=1}^J)$$

is the probability that $v_n \pm 1$ based on r_n and the received symbols corresponding to the transmitted code symbol $v_k, k \in \bigcup_{j=1}^J \mathcal{C}_j(n)$

0-5 is made up of the intrinsic information defined by 0-4 and the extrinsic information which is knowledge obtained partly from the knowledge of the code structure.

The calculation of 0-5 is simplified using the inverse of the hyperbolic tangent

$$\tanh^{-1} x = \frac{1}{2} \log \frac{1+x}{1-x}, \quad |x| < 1$$

and approximating the product $\prod_{k \in C_j} \Lambda(k)$ to the term with the lowest reliability ie

$$\prod_{\ell \in C_j(n)} \Lambda(n') \approx \text{sgn} \left(\prod_{n' \in C_j(n)} \Lambda(n') \right) \min_{n' \in C_j(n)} |\Lambda(n')|$$

Inserting the above approximations into 0-5 we get the equation below

$$\ell^{(1)}(n) \approx \ell(n) + \sum_{j=1}^J \text{sgn} \left(\prod_{n' \in C_j(n)} \ell(n') \right) \min_{n' \in C_j(n)} |\ell(n')| \quad (0-6)$$

0.2.1 Decoding Procedure

It is possible to use 0-6 to find a log-likelihood metric to the symbol v_n depending on r_n and the symbols received corresponding to those which are included in $C_j(n), j = 1, \dots, J$. However to improve the procedure, J different statistics $\ell_j^{(1)}(n), j = 1, \dots, J$ are calculated instead of one. This is similar to the procedure used by Gallager in [1]

1. $\ell^{(1)}(n)$ is calculated using the equation below

$$\ell_j^{(1)}(n) = \ell(n) + \sum_{\substack{j'=1 \\ j \neq j'}}^J \text{sgn} \left(\prod_{n' \in C_{j'}(n)} \ell_{j'}(n') \right) \min_{n' \in C_{j'}(n)} |\ell_{j'}(n')| \quad (0-7)$$

2. These metrics are further used to calculate a new metric upon the next iteration using the equation below

$$\ell_j^{(i)}(n) = \ell(n) + \sum_{\substack{j'=1 \\ j \neq j'}}^J \text{sgn} \left(\prod_{n' \in C_{j'}(n)} \ell_{j'}^{(i-1)}(n') \right) \min_{n' \in C_{j'}(n)} |\ell_{j'}^{(i-1)}(n')| \quad (0-8)$$

3. Using 0-8 $I - 1$ iterations are carried out and after that

$$\begin{aligned} \ell^{(I)}(n) = & \ell(n) + \sum_{j=1}^J \text{sgn} \left(\prod_{n' \in C_j(n)} \ell_j^{(I-1)}(n') \right) \\ & \times \min_{n' \in C_j(n)} |\ell_j^{(I-1)}(n')| \end{aligned} \quad (0-9)$$

4. Finally $\ell^{(I)}(n), n = 1, \dots, N$ is used to make a hard decision where $\hat{v}_n = 1$, if $\ell^{(I)}(n) < 0$ and $\hat{v}_n = 0$, if $\ell^{(I)}(n) > 0$

A summary to clarify the decoding step is shown below.

1. Initialization

Let $i = 0$, with all (j, n) pairs, $1 \leq j \leq NJ/K$ and $n = 0, \dots, N - 1$ satisfying $h_{jn} = 1$ define

$$\ell_j^{(i)}(n) = \ell(n) \quad (0-10)$$

2. Vertical Step

With all (j, n) pairs, $1 \leq j \leq NJ/K$ and $n = 0, \dots, N-1$ satisfying $h_{jn} = 1$, compute

$$Z_j^{(i)}(n) = \text{sgn} \left(\prod_{n' \in \mathcal{C}_j(n)} \ell_j^{(i)}(n') \right) \min_{n' \in \mathcal{C}_j(n)} \left| \ell_j^{(i)}(n') \right| \quad (0-11)$$

3. Horizontal Step

With all (j, n) pairs, $1 \leq j \leq NJ/K$ and $n = 0, \dots, N-1$ satisfying $h_{jn} = 1$, compute

$$\ell_j^{(i+1)}(n) = \ell(n) + \sum_{\substack{j'=1 \\ j \neq j'}}^{NJ/K} Z_{j'}^{(i)}(n) \quad (0-12)$$

if $i < I$, then let $i = i + 1$ and proceed to step 2

4. Decision Step

Make hard decision

$$\hat{v}_n = \frac{1}{2} \left(1 - 2 \text{sgn} \left\{ \ell(n) + \sum_{j=1}^{NJ/K} Z_j^{(I)}(n) \right\} \right) \quad (0-13)$$

0.3 Study of Asymptotic Behaviour of LDPC Codes

Show the relationship between the BER and the selection of (N, J, K) as N approaches infinity. To do this easily, the AWGN model is changed to a quantized AWGN model (量子化).

First, define quantized log-likelihood metric as

$$\hat{\ell}(n) = \begin{cases} t, & \left(t - \frac{1}{2}\right) \Delta \leq \ell(n) < \left(t + \frac{1}{2}\right) \Delta, \quad t = -q + 1, \dots, q - 1 \\ -q, & \ell(n) < \left(\frac{1}{2} - q\right) \Delta \\ +q, & \ell(n) \geq \left(\frac{1}{2} + q\right) \Delta \end{cases} \quad (0-14)$$

where the number of quantization steps is $2q+1$ and the size of the steps is δ . It is worth noting that for a given q the quantization step is chosen to maximize the cutoff rate of the channel.

The probability function of $\hat{\ell}(n)$ is given by

$$\begin{aligned} & \Pr(\hat{\ell}(n) = t | v_n) \\ &= \begin{cases} \int_{(t-1/2)\Delta}^{(t+1/2)\Delta} \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y-w_n)^2}{N_0}} dy, & t = -q + 1, \dots, q - 1 \\ \int_{-\infty}^{(1/2-q)\Delta} \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y-w_n)^2}{N_0}} dy, & t = -q \\ \int_{(q-1/2)\Delta}^{\infty} \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y-w_n)^2}{N_0}} dy, & t = q \end{cases} \end{aligned} \quad (0-15)$$

Since $\hat{\ell}(n)$ is quantized, the statistics

$$\hat{Z}_j^{(i)}(n) = \text{sgn} \left(\prod_{n' \in \mathcal{C}_j(n)} \hat{\ell}_j^{(i)}(n') \right) \min_{n' \in \mathcal{C}_j(n)} |\hat{\ell}_j^{(i)}(n')| \quad (0-16)$$

and

$$\hat{\ell}_j^{(i+1)}(n) = \hat{\ell}(n) + \sum_{j'=1, j' \neq j}^{NJ/K} \hat{Z}_{j'}^{(i)}(n), \quad i = 1, \dots, I-1 \quad (0-17)$$

will have only integer values. Assuming that the code is chosen to ensure independence between the random variables $\hat{\ell}_j^{(i+1)}(n)$ and $\hat{Z}_j^{(i)}(n) = \forall j = 1, \dots, J, \quad 1 \leq i \leq I$. Then the probability function of $\hat{\ell}_j^{(i+1)}(n)$ is given by a convolution between the probability functions of $\hat{Z}_j^{(i)}(n)$ and $\hat{\ell}(n)$ carried out J times. This can in turn be expressed through the probability function of $\hat{\ell}_j^{(i)}(n)$

Using this model, the probability function of the statistics

$$\begin{aligned} \hat{\ell}^{(I)}(n) = \hat{\ell}(n) + \sum_{j=1}^{\infty} \text{sgn} \left(\prod_{n' \in \mathcal{C}_j(n)} \hat{\ell}_j^{(I-1)}(n') \right) \\ \times \min_{n' \in \mathcal{C}_j(n)} |\hat{\ell}_j^{(I-1)}(n')| \end{aligned} \quad (0-18)$$

have been calculated and are used to obtain the BER with regards to wrongly decoding v_n .

$$\begin{aligned} \text{BER} = \Pr \left(\hat{\ell}^{(I)}(n) < 0 | v_n = 0 \right) \\ + \frac{1}{2} \Pr \left(\hat{\ell}^{(I)}(n) = 0 | v(n) = 0 \right) \end{aligned} \quad (0-19)$$

The results for rate $R = 2$ and $q = 4$ is shown for different iterations. In conclusion, J must be as small as possible but larger than 2

0.4 Time Varying Periodical Binary Convolutional Codes

The transparent form of the parity-check matrix of a time-varying, periodical, rate $R = b/c$ binary convolutional code with memory M . is shown below. It should be noted that it is a semi-infinite matrix.

$$\mathbf{H}^T = \begin{bmatrix} \mathbf{H}_0^{(0)} & \mathbf{H}_1^{(1)} & \dots & \mathbf{H}_{t-1}^{(M-1)} & \mathbf{H}_t^{(M)} \\ & \mathbf{H}_1^{(0)} & \dots & \mathbf{H}_{t-1}^{(M-2)} & \mathbf{H}_t^{(M-1)} & \mathbf{H}_{t+1}^{(M)} \\ & \ddots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (0-20)$$

where $\mathbf{H}_t^{(m)}, m = 0, 1, \dots, M, t = 0, 1, \dots$ are $c \times (c - b)$ binary submatrices. $\mathbf{H}_t^{(0)}$ should be full rank and $\mathbf{H}_t^{(m)} = \mathbf{H}_{t+T}^{(m)}, T$ is the period of the code. For $R = 1/2$, the memory of the parity-check matrix is the same as that of the code, though this is not always the case for other codes with different rates.

Similarly, an (M, J, K) low-density parity-check matrix \mathbf{H}^T (in symbolwise form) of a time-varying periodical rate $R = b/c$ convolutional code, has J ones in each row, K ones in each column starting from the $M(c - b)$ th column, and zeros elsewhere. Only rate $R = 1/2$ codes having a

different number of ones in the rows of \mathbf{H}^T corresponding to the information symbols and to the parity-check symbols are considered. Specifically three ones for those corresponding information symbols and two for the parity-check symbols. Each column, starting from the $M(c-b)$ th column has five ones. We call the corresponding matrix \mathbf{H}^T an $(M, 3(2), 5)$ parity-check matrix. We also limit ourselves to the case when the period $T = M - 1$.

A general method for constructing this parity check matrix is outlined below.

1. Begin by constructing a $(T, T/2, T/2)$ matrix
2. Split the matrix along the diagonal and interchange the positions of the left and right diagonal
3. Add zeros beneath each row of the matrix
4. Finally, if the row index is odd, add a 1, 0 to the beginning and end of the row respectively. If not add a 1, 1 instead.

Next is the encoding procedure for this convolutional code. Let

$$\mathbf{u}_{[0,n]} = [u_0, u_1, \dots, u_n]$$

and

$$\mathbf{v}_{[0,n]} = [v_{0,1}, v_{0,2}, v_{1,1}, v_{1,2}, \dots, v_{n,1}, v_{n,2}]$$

be segments of an information sequence and the corresponding encoded respectively. $\mathbf{v}_{[0,n]}$ satisfies the equation

$$\mathbf{v}_{[0,n]} \mathbf{H}_{[0,n]}^T = \mathbf{0}, \quad n = 0, 1, \dots \quad (0-21)$$

and $\mathbf{H}_{[0,n]}^T$ is a segment of the low-density parity-check matrix of the time varying periodical convolutional code shown below

$$\mathbf{H}_{[0,n]}^T = \begin{bmatrix} h_1^{(0)}(0) & h_1^{(1)}(1) & \dots & h_1^{(M)}(M) & 0 & \dots & 0 \\ h_2^{(0)}(0) & h_2^{(1)}(1) & \dots & h_2^{(M)}(M) & 0 & & \\ 0 & h_1^{(0)}(1) & \dots & \vdots & h_1^{(M)}(M+1) & & \\ \vdots & h_2^{(0)}(1) & \dots & & h_2^{(M)}(M+1) & & \\ & 0 & \ddots & & \vdots & \vdots & \\ & & \ddots & & & 0 & \\ & & & & & h_1^{(M)}(n) & \\ & & & & & h_2^{(M)}(n) & \\ & & & & & \vdots & \\ & & & & & 0 & h_2^{(0)}(n) \end{bmatrix}$$

Since $h_1^{(0)}(n) = h_2^{(0)}(n) = 1$. We find a systematic encoder using the equation below

$$\begin{aligned} v_{n,1} &= u_n \\ v_{n,2} &= \sum_{i=0}^M h_1^{(i)}(n) u_{n-i} + \sum_{i=1}^M h_2^{(i)}(n) v_{n-i,2} \end{aligned} \quad (0-22)$$

Generally, the encoder can be implemented using c length $M + 1$ shift registers in parallel and time varying connections from the register stages to the modulo 2 adders.

0.5 Iterative Decoding of Low-Density Convolutional Codes

It is assumed that a rate $1/2$ $(M, 3(2), 5)$ low-density time-varying periodical convolutional code is used for transmission over the AWGN channel. The code symbols are modulated and mapped to the signal points w_n as

$$\begin{aligned} w_{2n} &= 1 - 2v_{n,1}, & n = 0, 1, \dots \\ w_{2n+1} &= 1 - 2v_{n,2}, & n = 0, 1, \dots \end{aligned} \tag{0-23}$$

The procedure is similar to that of the previous section. The major difference is the pipeline implementation of the decoder, which is made up of I separate processors and $J + 1$ first-in-first-out FIFO shift registers. (FIFO). The first register contains the log-likelihood metrics $\ell^{(1)}(n)$ received from the channel, while the other J registers contain intermediate statistics computed by the different processors. Initially, the encoder contains dummy zeroes. Therefore, the FIFO 's are initially filled with 1, which is the log-likelihood ratio for the absolute reliable received statistics, which correspond to the code symbol zero.

Each time a new channel statistic is received, it is shifted in all the FIFO 's. Each statistic $\ell^{(1)}(n)$, flows through the FIFO, as in a pipeline, and is processed sequentially by the I processors. The I processors are $2(M + 1)$ registers apart from each other so they are processing different and consecutive segments of the FIFO stages.

The decoding procedure is outlined below.