Binary LDPC Woven Convolutional Codes Decoding

ZYABLOV V zyablov@iitp.ru
Inst. for Information Transmission Problems
Russian Academy of Science
Russia
KONDRASHOV K k_kondrashov@iitp.ru
Inst. for Information Transmission Problems
Russian Academy of Science
Russia

Decoding performance of two low-density parity-check (LDPC) woven convolutional codes being decoded with majority voting hard decision algorithm and erasure inserting algorithm is studied. One of subject codes is Braided Block Code (BBC) with two Hamming constituent codes while the other one is certain LDPC woven convolutional code with four parity-check constituents.

1 Introduction

As the younger brother of LDPC block codes LDPC convolutional codes [1] do not attract a lot of attention – most works on LDPC are devoted to block codes. However dynamics changes and recently a new class of LDPC convolutional codes was introduced [2]. In this paper we study properties and compare decoding performance of BBC with two Hamming constituent codes and certain LDPC woven convolutional code with increased to four number of constituent codes. For convenience we describe both codes by means of a two dimensional infinite shifting array. Thus any codeword is an array with code symbols placed into the cells. Checks are performed row-wise and column-/diagonal-wise by the constituent codes.

2 Braided Block Code

Braided Block Codes, first introduced in [2], are constructed by the interconnection of two block codes, a horizontal component code and a vertical component code. Code array consists of three ribbons (Fig.1). Central ribbon represent information sequence $\boldsymbol{u} = [\boldsymbol{u}_0 \boldsymbol{u}_1 \ldots \boldsymbol{u}_t \ldots]$. Leftmost ribbon consists of vertical component check symbols which in combination with corresponding information blocks form horizontal component informational parts. The same applies to the

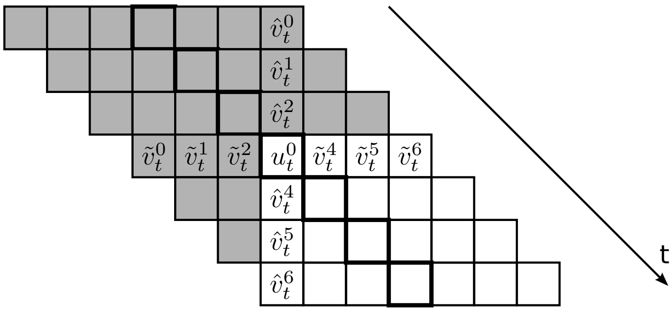


Figure 1: BBC with Hamming (7,4)-codes array representation

rightmost ribbon except that vertical components are changed to horizontal and vice versa. Consider BBC with Hamming (7,4)-codes as constituents. At any time t information block $\boldsymbol{u}_t = \left[u_t^0\right]$ is put into central ribbon. Then horizontal component encodes $\left[\tilde{v}_t^0\tilde{v}_t^1\tilde{v}_t^2u_t^0\right]$ giving $\left[\tilde{v}_t^0\tilde{v}_t^1\tilde{v}_t^2u_t^0\tilde{v}_t^4\tilde{v}_t^5\tilde{v}_t^6\right]$. Vertical component encodes $\left[\hat{v}_t^0\hat{v}_t^1\hat{v}_t^2u_t^0\right]$ giving $\left[\hat{v}_t^0\hat{v}_t^1\hat{v}_t^2u_t^0\hat{v}_t^4\hat{v}_t^5\hat{v}_t^6\right]$. Output code block at time t is $\boldsymbol{v}_t = \left[u_t^0\tilde{v}_t^4\tilde{v}_t^5\tilde{v}_t^6\hat{v}_t^4\hat{v}_t^5\hat{v}_t^6\right]$. Resulting rate $R = \frac{1}{7}$. In general the rate of a BBC is given by $R = R_1 + R_2 - 1$, where R_1 and R_2 are the rates of the horizontal and vertical constituent codes.

3 LDPC Woven Convolutional Code

While BBCs may use strong constitutional codes they are limited to only two components – vertical and horizontal. Meanwhile in [3] was suggested certain LDPC woven convolutional code allowing to increase number of constituent codes up to four at the cost of their simplification. New LPDC woven convolutional code construction uses specifically arranged simple codes with one parity check (Fig. 2). Encoding is done slightly different compared to BBC. At the time instance t first step is to encode vertical and diagonal components. This could be done concurrently. Then informational block $\boldsymbol{u}_t$ and check symbols obtained at previous step is encoded by horizontal constituent code. The rate of resulting code is given by $1 - J/K$, where J is the number of constituent codes and K is their length.

4 Encoding

Along with the array representation, abovementioned codes, as a particular case of LDPC convolutional codes, can be described by syndrome formers. These

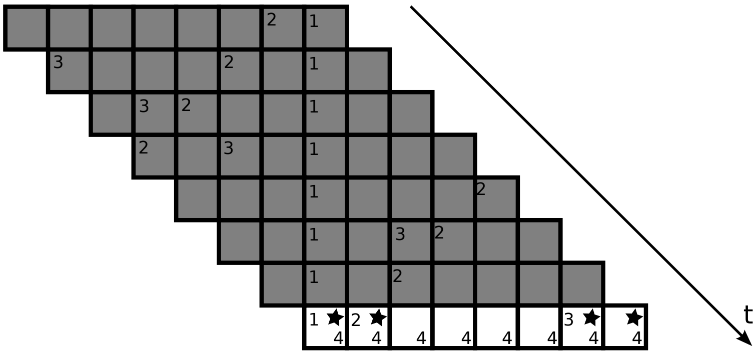


Figure 2: LDPC woven convolutional construction with 4 constituent codes. Indices in cells denote belongings to constituent codes. Gray shaded cells stand for encoded symbols. Information symbols are put in empty cells. Check symbols are positioned within cells marked with stars.

are semi-infinite check matrices

$$\boldsymbol{H}^{\mathrm{T}} = \begin{pmatrix} \boldsymbol{H}_0^{\mathrm{T}}(0) & \dots & \boldsymbol{H}_{m_s}^{\mathrm{T}}(m_s) & & \\ & \ddots & & \ddots & \\ & & \boldsymbol{H}_0^{\mathrm{T}}(t) & \dots & \boldsymbol{H}_{m_s}^{\mathrm{T}}(t+m_s) \\ & & \ddots & & \ddots \end{pmatrix}, \tag{1}$$

where submatrix $\boldsymbol{H}_i^{\mathrm{T}}(t)$ is of dimension $c \times (c-b)$, b is information block length, c is encoded block length and m_s is code memory. At any time instance t code sequence satisfies:

$$\boldsymbol{v}_t \boldsymbol{H}_0^{\mathrm{T}}(t) + \boldsymbol{v}_{t-1} \boldsymbol{H}_1^{\mathrm{T}}(t) + \dots + \boldsymbol{v}_{t-m_s} \boldsymbol{H}_{m_s}^{\mathrm{T}}(t) = 0, t \in \mathbb{Z} \tag{2}$$

and

$$\boldsymbol{v}_{[0,t-1]} \boldsymbol{H}_{[0,t+m_s-1]}^{\mathrm{T}} = \left[\mathbf{0}_{[0,t-1]} | \boldsymbol{s}_t \right], \tag{3}$$

where $\boldsymbol{s}_t = \left[\boldsymbol{s}_t^0 \boldsymbol{s}_t^1 \dots \boldsymbol{s}_t^{m_s-1} \right]$ is partial syndromes vector with recurrent update rule

$$\boldsymbol{s}_t^i = \begin{cases} \boldsymbol{s}_{t-1}^{i+1} + \boldsymbol{v}_t \boldsymbol{H}_{i+1}^{\mathrm{T}}(t+i+1), & i = 0, \dots, m_s - 2 \\ \boldsymbol{v}_t \boldsymbol{H}_{i+1}^{\mathrm{T}}(t+i+1), & i = m_s - 1. \end{cases} \tag{4}$$

From (3) it follows that newly encoded block may be obtained as solution of

$$\boldsymbol{v}_{t+1} \boldsymbol{H}_0^{\mathrm{T}}(t+1) = -\boldsymbol{s}_t^0. \tag{5}$$

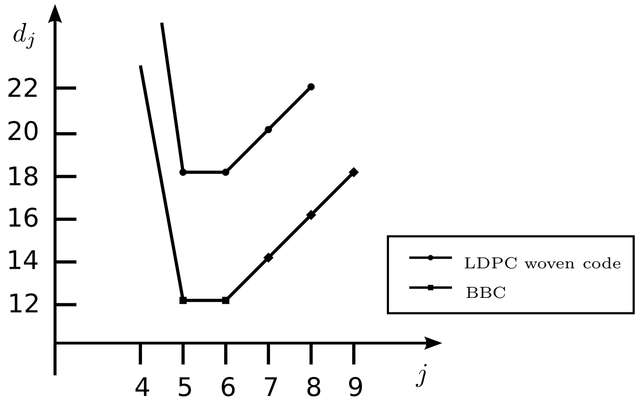


Figure 3: Free distances

5 Distance Properties

One of the most important code properties having direct impact on code correcting capability is the minimum Hamming distance between any two different codewords

$$d_{free} = \min_{\boldsymbol{v} \neq \boldsymbol{v}'} \left\{ d_{\mathrm{H}}(\boldsymbol{v}, \boldsymbol{v}') \right\}. \tag{6}$$

In the case of convolutional codes it is called *free distance.* From convolutional codes linearity it follows immediately that d_{free} is the minimum Hamming weight over all non-zero codewords. Convolutional code free distance may be computed by means of code active distances d_j:

$$d_j = \min \left\{ \omega_{\mathrm{H}}(\boldsymbol{v}_{[1,j]}) \right\} : \ \boldsymbol{v}_{[1,j]} \boldsymbol{H}^{\mathrm{T}}_{[1,j+m_s-1]} = \boldsymbol{0} \tag{7}$$

i.e minimum weights among codewords merging with zero codeword after j information blocks. Their relation is $d_{free} = \min_j \{d_j\}$. To find active distances we solve for different j system of linear equations

$$x \boldsymbol{H}^{\mathrm{T}}_{[1,j+m_s-1]} = \boldsymbol{0}. \tag{8}$$

For fixed j this system has either single zero solution or a set of solutions. In the last case the set of solutions forms fundamental solutions system (FSS) so that all possible solutions are found as its linear combination. From FSS we find codewords with minimum weight. BBC and LDPC woven convolutional code active distances are shown at Fig. 3.

6 Decoding Algorithms

In this paper we consider two hard decision decoding algorithms. These are majority voting algorithm $\mathcal{A}_1$ and its erasures inserting modification $\mathcal{A}_2$. Let us describe these algorithms in common for both codes and then outline differences.

Denote k-th constituent code decoder as $\mathcal{D}^{(k)}$. Let iterative outer decoder input at iteration i be $\boldsymbol{r}^{(i)}$, where $\boldsymbol{r}^{(1)}$ is received erroneous word. Algorithm $\mathcal{A}_1$ consists of two major steps. At first step constituent codewords are decoded. At second step the decision is made on symbols of received word.

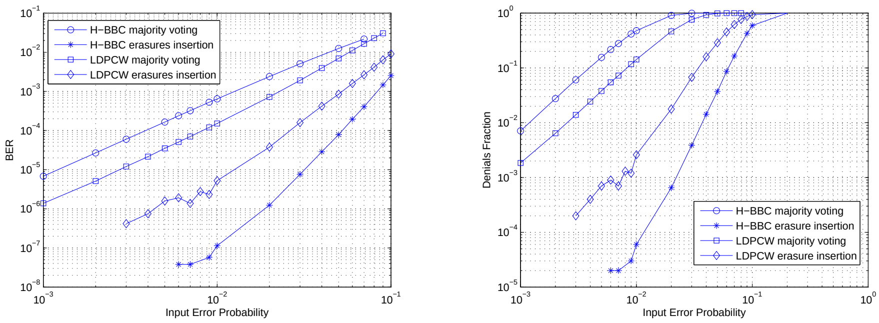
A l g o r i t h m $\mathcal{A}_1$:

1. For each constituent code k its decoder $\mathcal{D}^{(k)}$ decodes all corresponding codewords within $\boldsymbol{r}^{(i)}$ giving solutions for all symbols of $\boldsymbol{r}^{(i+1)}$
2. For each symbol of $\boldsymbol{r}^{(i+1)}$ the decision is made. If more than half of constituent codes have reached the agreement on concerned symbol then it is updated with agreed decoders decision. Otherwise its value from $\boldsymbol{r}^{(i)}$ is kept.
3. $\boldsymbol{r}^{(i+1)}$ syndrome is calculated. If it is not zero and decoder did not run out of iterations start new iteration from step one. Zero syndrome completes decoding with either decoding success or decoding failure (different codeword).

Algorithm $\mathcal{A}_2$ is modification of $\mathcal{A}_2$. It modifies step 2 so that when there are no agreement among $\mathcal{D}^{(k)}$-s on certain symbol it inserts erasure at that symbol position. It also requires constituent codes erasure decoders $\mathcal{E}^{(k)}$.

A l g o r i t h m $\mathcal{A}_2$:

1. For each constituent code k its decoder $\mathcal{D}^{(k)}$ decodes all corresponding codewords within $\boldsymbol{r}^{(i)}$ giving solutions for all symbols of $\boldsymbol{r}^{(i+1)}$
2. For each symbol of $\boldsymbol{r}^{(i+1)}$ the decision is made. If more than half of constituent codes have reached the agreement on concerned symbol then it is updated with agreed decoders decision. Otherwise it is erased.
3. If there are erasures in resulting word $\boldsymbol{r}^{(i+1)}$, repeat step one with erasure decoders $\mathcal{E}^{(k)}$ until it eliminate all erasures. At this step we can not introduce new erasures. If some repeat operation do not lessen number of iterations, change remaining erasures either with all zeroes or all eigenvalues, depending on which case results in lesser syndrome weight.
4. At this step there are no erasures. $\boldsymbol{r}^{(i+1)}$ syndrome is calculated. If it is not zero and decoder did not run out of iterations start new iteration from step one. Zero syndrome completes decoding with either decoding success or decoding failure (different codeword).



(a) Bit error rate

(b) Denials fraction

Figure 4: Decoding H-BBC and LDPC woven convolutional code

Both algorithms need to be slightly adjusted for LDPC wooven convolutional code with parity-check constituent-codes. Since parity-check code can only detect one error but not correct it we do not need step one. Decision on symbol at step two is then made according to next rule. If more than half of constituent codes, this symbol belongs to, detected an error, invert the symbol.

Simulation results for BBC with Hamming (15,11)-constituent codes (H-BBC) and LDPC woven convolutional code (LDPCW) with parity-check codes of length 8 are shown at Fig. 4.

7 Conclusions

Two additional constituent codes in spite of their simplicity made LDPC wooven convolutional code beat BBC when decoding with majority voting algorithm. However more interesting result is that erasure insertion may significantly improve decoding performance.

References

[1] A. J. Felström and K. Sh. Zigangirov, Periodic time-varying convolutional codes wit low-density parity-check matrices, *IEEE Trans Inf. Theory*, vol. 45, no. 45, 2181–2190, 1999

[2] A. J. Felström, M. Lentmaier, D. V. Truhachev and K. Sh. Zigangirov, Braided block codes, *IEEE Trans Inf. Theory submission*, 2006

[3] V. V. Zyablov and K. A. Kondrashov, Two LDPCCconstructions *Information Technologies and Systems Workshop, Becasovo, Russia, 2009* 156–159.