A binary block concatenated code based on two convolutional codes

Igor Zhilin Victor Zyablov Dmitry Zigangirov

Institute for Information Transmission Problems
Russian Academy of Sciences

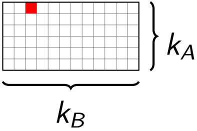
XV International Workshop on
Algebraic and Combinatorial Coding Theory
June 18–24, 2016, Albena, Bulgaria

Outline

- Construction Description and Encoding
- Derivation of Code Distance
  - $\rightarrow$ upper bounding
  - $\rightarrow$ lower bounding
- Conclusion

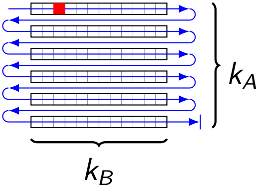
- We consider a *block* code that uses *terminated* convolutional codes as component codes.

Let us start with information matrix:

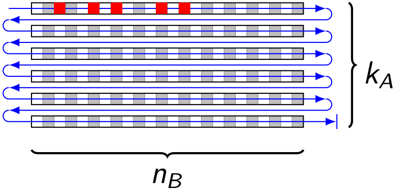
$\mathbf{I} =$ 

Construction Description // Encoding Outer

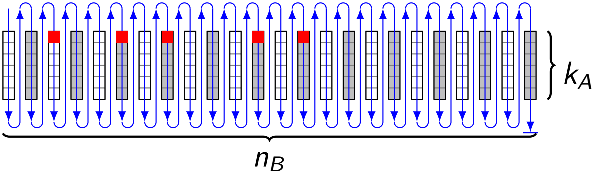
- At first it is read in row-wise order and encoded by the outer convolutional coder, $\mathbf{I} =$



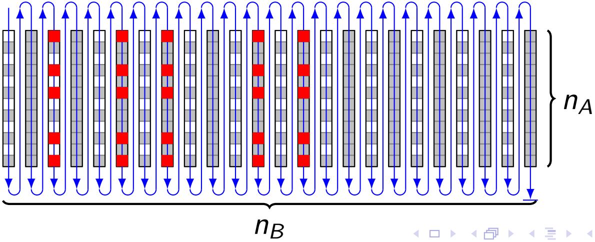
- The resulting matrix is written in row-wise order too, $\mathbf{I}_A = Enc_B(\mathbf{I}) =$



- Then $\mathbf{I}_A$ is read in column-wise order by inner convolutional code encoder, $\mathbf{I}_A =$



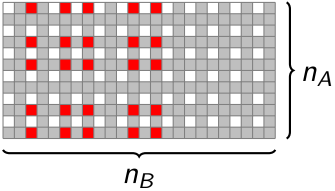
- And written in the same column-wise order to a matrix that is a codeword, $\mathbf{C} = Enc_A(\mathbf{I}_A) = Enc_A(Enc_B(\mathbf{I})) =$



Construction Description // Codeword

- The result is a codeword:

$$\mathbf{C} = Enc_A(Enc_B(\mathbf{I})) =$$



  Shaded cells correspond to parity-check symbols.
  Red cells schematically depict minimal-weight codeword.

- Note: a single encoder is used for encoding all rows. Then a single encoder is used for encoding all columns.

Notations

$R_A = b_A/c_A$ — rate of inner code,
d_A — free distance of inner code (in binary symbols),
f_A — maximum length of word (packet) of inner code that has weight d_A, measured in c_A-tuples,

R_B, b_B, c_B, d_B, f_B — the same for outer code.

Let us consider code construction where $n_A \geq f_A c_A$, $n_B \geq f_B c_B$. That means that the longest word of minimal weight of outer/inner code fits in a single row/column (probably with wrapping).

Code Distance

Theorem

There exist such sizes n'_A and n'_B that binary block concatenated code based on two convolutional codes with $n_A \geq n'_A$ and $n_B \geq n'_B$ has minimum Hamming distance $d = d_A d_B$, where d_A and d_B are free distances of inner and outer codes respectively.

Upper Bound

- To prove $d \leq d_A d_B$ we can just provide an example of codeword of weight $w = d_A d_B$
- Since we've chosen $n_B \geq f_B c_B$, we can place a sequence of weight $w' = d_B$ in any rows of $\mathbf{I}_A$. These rows would be independent since such sequence has length $f_B c_B$ that is less than row width n_B.
- We should place that sequences in rows of $\mathbf{I}_A$ in a such way that nonzero symbols would form information sequences of smallest weight d_A in columns of $\mathbf{C}$.
- This encoding procedure yields a codeword that has d_A rows of weight d_B, or, alternatively, d_B columns of weight d_A, thus its weight $w = d_A d_B$.

Lower Bound

- Let us prove it from the encoding standpoint.
- Encoding of the outer code is just a plain encoding of the convolutional code with arbitrary input. Its output sequence has at least d_B nonzero symbols. Since we've chosen $n_B \geq f_B c_B$, all these bits would be in different columns of $\mathbf{I}_A$ yielding at least d_B nonzero columns.
- Now we should consider two options:
  1. In case the columns would be encoded by the inner code independently from column to column, the result is straightforward: it yields a codeword similar to the one considered for upper bound (probably with wrapped rows or columns). Encoding of each column by the inner encoder gives a word of weight at least d_A, so in this case $d \geq d_A d_B$.
  2. Counting for dependencies in columns-to-column encoding requires use of active distances of inner code.

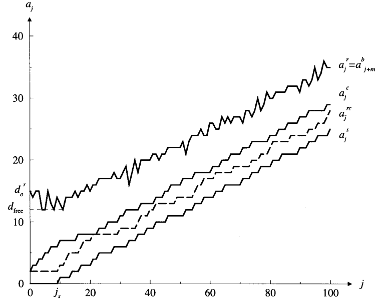
Active Distances

- A concept of active distance was introduced in 1999 by Host et. al.[1].
- Active distances lower bound weight of a code sequence generated by a coder that does not pass through two consequent zero states.
- Authors[1] proved that convolutional codes with active distances that grow with sequence length and are lower-bounded by a linearly increasing function exist . . .
- and also showed a couple of examples of known codes where increasing active distances are seen.

[1] S. Host, R. Johannesson, K. Sh. Zigangirov, V. V. Zyablov, "Active Distances for Convolutional Codes," *IEEE Transactions on Information Theory, Vol. 45, No. 2, March 1999*.

Active Distances

- Example of active column distance curve from [1]:



- Let us write a bound on the active column distance a_j^r in simplified form:

$$a_j^r \geq uj + v \qquad (1)$$

where $u > 0$ is a constant that depends on code properties, j is a sequence length in c_A-tuples.

Lower Bound (continues)

- Since we need two consequent columns to have weight of at least $2d_A$, three columns to have weight $3d_A$ and so on, we need to choose such n_A that active column distance

$$a_j^r \geq sd_A, s \in \overline{1, n_B}, \tag{2}$$

where $j = sn_A/c_A$.

- and (after a couple of transformations)

$$n_A \geq d_A c_A / u = const \tag{3}$$

- This ends the proof of $d \geq d_A d_B$ and thus $d = d_A d_B$.

Conclusion

- We proved that code distance of binary block concatenated code based on two convolutional codes equals $d = d_A d_B$ — the product of free distances of component codes for large enough n_A and n_B.
- This construction differs from other constructions of concatenated codes based on convolutional codes:
  - It is not a convolutional code like the one proposed in [2]
  - It doesn't use separate codes for each row and each column like in, i.e., [3]

[2] M. Bossert, C. Medina, V. Sidorenko, "Encoding and distance estimation of product convolutional codes," *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005., Adelaide, SA, 2005, pp. 1063-1067*

[3] O. Gazi, A. O. Yilmaz, "Turbo Product Codes Based on Convolutional Codes," *ETRI Journal, Volume 28, Number 4, August 2006*

Thank you for your attention.