Protograph-Based Interleavers For Punctured Turbo Codes

Kwame Ackah Bohulu

July 8, 2020

1 Introduction

1. Over the years, the number of users for services like live streaming and interactive gaming have increased and there is a need for future generations of mobile networks(5G and beyond) to be able to provide high capacity and data rates for these applications.

2. This can be acheived by providing bit error rates which are 10^-5 lower than the frame error rates and the current communication systems (including 4G LTE) are unable to meet this demand[4]

3. 4G LTE is unable to meet this demand due to using the HARQ retransmission method and the fact that the rate matching technique causes undesired interaction between the interleaver and the puncturing mechanism[5] .

4. The turbo code is a popular error-correcting code which is used in the LTE standard as well as WiMAX and DVB-RCS/RCS2. The error performane of the turbo code is closely related to the interleaver used and a lot of interleavers have been proposed over the years

5. The QPP interleavers is used in the LTE[4] standard, whiles the ARP and DRP interleavers have both been used in WiMAX[7] and DVB-RCS/RCS2[8],[13] standards

6. Methods that have been used for the construction of the ARP interleaver include using the correlation girth[15] as well as the minimum distance of the turbo code[12].

7. Paper [16] proposes a method where parity puncturing constraints are included in the design of the interleaver to improove performance.

8. In this research joint optimization of puncturing pattern and interleavers in investigated. The aim is to guarantee low error floor and good convergence thresholds

9. To this end, a layered construction of interleaver is introduced which bears similarities to protographs used in LDPC code construction[17]. The focus of this work is the ARP interleaver model[12]

10. in the end, there is a reduction in the search space for good interleaver-parameters as well as an increase in error correcting performance

2 System Description

1. Tail-biting trellis termination is best suited method for turbo codes becasue it avoids the loss of spectral efficiency of the code and also guarantees equal protection for the information bit sequence

2. To fully utilize these benefits, this research paper uses circular recursive systematic convolutional (CRSC) codes as component codes.

3. Input information sequence $\mathbf{d}$ and corresponding interleaved sequence $\mathbf{d}'$ are encoded by CRSC1 and CRSC2 respectively. Both have length K

4. The outputs of the turbo encoder (ie $\mathbf{d}$,$\mathbf{r}_1$,$\mathbf{r}_2$) are puntured using puncturing mask of period M before transmission.

Interleaver Model

1. The turbo code interleaver is responsible for determining the value of d_{min} as well as reducing the correlation between exchanged extrinsic information during the decoding process

2. The interleaving rule used in this research paper states that elements from $\mathbf{d} = (d_0, d_1, ..., d_{K-1})$ are mapped to the interleaved vector $\mathbf{d}' = (d_{\Pi(0)}, d_{\Pi(i)}, ..., d_{\Pi(K-1)})$

3. A symbol read out of address $\Pi(i)$ of $\mathbf{d}$ is written to address i of $\mathbf{d}'$

4. In this research the ARP interleaver[12] model is used. According to [20], the ARP interleaver can produce d_{min} values which are at least equal to that of QPP and DRP interleaver

ARP Interleaver

1. The basis of this interleaver is the Regular (linear) interleaver

$$\Pi(i) = (P \cdot i) \bmod K$$

where P is the period of the interleaver

2. Due to the regular structure, rectangular RTZ inputs cannot be effectively broken[12] and a vector of shifts $\mathbf{S}$ is used to add some disorder to the regular interleaver

3. The corresponding ARP interleaver becomes

$$\Pi(i) = (P \cdot i + S(i \bmod Q)) \bmod K$$

where Q is the size of the shift vector and is a divisor of K[15]

Puncturing Pattern

1. The puncturing pattern can be used to increase the coding rate of the turbo code and periodic puncturing method with period M is considered in this research paper

2. Since a rate 1/2 CRSC code is used for both component codes, a puncturing mask made up of 2 vectors of length M is used.

3. The patterns correspond to the positions in $\mathbf{d}$,$\mathbf{r}_1$,$\mathbf{r}_2$ to be punctured

4. To avoid edge effect due to puncturing, M is assumed to be a multiple of K

3 Interleaver Design Criteria

1. In the design of turbo code interleavers, the Hamming distance spectrum as well as the correlation between the channel information and the a priori information at the decoder output.

2. The first terms of the distance spectrum should be maximized and have low multiplicity and the correlation between the channel information and the a priori information must be minimized

3. The minimum span and correlation girth are the parameters which we can use to check if the above conditions are met.

A. Minimum Span

1. The span of symbols at position i and j according to [12],[21] is defined as
$$S_p(i,j) = f(i,j) + f(\Pi(i), \Pi(j))$$
where
$$f(u,v) = \min[|u-v|, K - |u-v|]$$

2. minimum span $S_{p\ min} = min_{i \neq j}[S_p(i,j)]$ and according to [23] using tail-biting trellis termination, the upper bound for $S_{p\ min}$, $S_{ub} = \lfloor \sqrt{2K} \rfloor$

3. As shown in [21], the higher the value of $S_{p\ min}$ the larger the value of d_{min} for random interleavers and therefore, $S_{p\ min}$ must be maximized.

Correlation Girth

1. In terms of the turbo decoding process, the decoder output at position $\Pi(i)$ depends on the received symbol at the same position. This received symbol also depends on the a number of past symbols close to $\Pi(i)$

2. Furthermore the same decoder output at position $\Pi(i)$ also depends on the a priori information provided by the second from position i and the correlation properties in turn depend on information from bits within the vicinity of i in the second decoder.

3. The interleaver should be designed to reduce the level of correlation between the a priori information and the data sequence of each constituent code [25]. A correlation graph may be used to achive this during interleaver design.

4. In the end, the interleaver should maximize the correlation girth, where the correlation girth is the minimum correlation cycle.

5. the graph shows the correlation cycles in the turbo decoding process. I t should be noted that this graph is a regular grapgh of degree $r = 4$ ie each vertex has 4 neighbours.

6. An upper bound on the girth value based on the Moore bound[26] can be calculated as
$$g \leq 2 \log_{r-1}(x(r,g)) + O(1)$$
where $O(1)$ is the error term of the approximation and $x(r,g)$ lowest number of vertices in a r-regular graph.

7. The above equation is an implication of Moore bound which is that the girth g can be at most proportional to the logarithm of x(r,g)

4 Interleaving With Puncturing Constraints

1. According to [27],[28] puncturing of well-chosen systematic bits increases d_{min} and reduce convergence threshold of high-rate turbo codes.

2. Because of this, the design of puncturing masks including data puncturing is considered.

3. Details abot puncturing mask selection and puncturing constraints on interleaver design are discussed

A. Puncturing Mask Selection

1. The puncturing mask configuration is defined in terms of desired rate R and puncturing period M

2.
$$R = \frac{M}{M(1 - D_p) + 2U_p}$$

   where D_p is the ratio of punctured systemic bits to total number of data bits and U_p is the number of unpuctured parity bits for each CRSC encoder.

3. For a given R and M, $D_p = \frac{m}{M}$, $m = 0, ..., M$. Practically, D_p is chosen to ensure that $R_c < 1$ (R_c is the constituent encoder rate)

4. In this research paper, U_p is an integer value and is the same for both component encoders.

5. The proposed puncturing mask design is described as follows

1) Find the best puncturing pattern for each D_p value

(a) The FAST algorithm[29] is used for this stage. Traditionally it is used to find the distance spectrum for unpuntured convolutional codes.

(b) For M periodic puncturing of a CRSC code the FAST algorithm is run M times, each time beginning at different points in the puncturing mask.

(c) In the end, the all the distance spectrum are joined together to get the distance spectrum of the punctured CRSC code

(d) the best puncturing mask is the one which gives the largest minimum distance value at a low multiplicity.

2) Carry out a mutual information exchange analysis to select a restricted set of puncturing masks

(a) it was shown in [28] that the distribution of the extrinsic information for the punctured data bit positions is different from that of the unpunctured data bit positions.

(b) For this reason, in the use of the EXIT chart analysis done in [30] the iterative evolution of the a priori information during turbo decoding (which is obtained via Monte Carlo simulation) is used instead of the Gaussian approximation of the a priori information.

(c) Also, the uniform interleaver is used to average out the effect of the interleaver on the extrinsic information exchange. This approach is similar to that used in [32] to identify best turbo code precoding structure.

(d) From [33] and with respect to each constituent encoder, the average mutual information between the a posteriori LLR L and and data frame X is given by

$$I(L; X) = 1 - \frac{1}{K} \sum_{i=1}^{K} \log_2 \left(1 + e^{-x_i L_i} \right)$$

(e) In the modified EXIT chart, the best pucturing mask is the one that provides the closest crossing point (IA, IE) to $(1, 1)$

(f) The error correction performance of the remaining puncturing masks are evaluated using the uniform interleaver and the one that provides the best tradeoff between performance in the waterfall and error floor region is selected.

B. Data Puncture-Constrained Interleavers

1. For punctured Turbo codes, puncturing contstraints must be taken into account when designing the interleaver.

2. Failure to do so will lead to the use of catastrophic or semi-catastrophic puncture masks which can produce d_{min} as low as 1

3. the puncturing constraint must ensure that the same puncturing pattern is preserved for both component codes.

C. Protograph-based Interleavers

1. Useful constraints for interleaver design are presented here.

2. The analysis in this section is done based on the observation that the reliability of an information bits extrinsic information is depent on parameters such as symbol position in puncturing period, wether or not corresponding parity is punctured and number of punctured parities.

3. During the turbo decoding process, the extrinsic information of one decoder becomes the a priori information of the next decoder and it is expected that extrinsic information gotten from unpuntured parity positions will be more reliable than those gotten from puntured parity positions. This is confirmed in Fig.5

4. With the above consideration, a possible strategy for the interleaver construction could be connecting the positions with highly reliable extrinsic information to the positions with unreliable extrinsic information, which are more prone to errors in order to spread the error correcting capability of the turbo code

5. A specific strategy would be to sort the data positions in a puncturing period in order of increasing reliability and via the interleaver connecting the least error-prone data position of one component code to the most error-prone data position of the other component code.

6. The connection graph that results is named a protograph in relation to protograph LDPC coded[17]. The protograph is defined to via the following steps.

   1. Sorting of unpunctured data positions by error-prone level in a puncturing period M:

   (a) Each unpunctured data position in the puncturing mask is punctured and the distance spectrum of the associated code is evaluated. After that they are arranged according to the d_{min} and multiplicity

   (b) the lower the d_{min} and higher the multiplicity value, the more error prone that data position is. It it worth noting that this extra puncturing is only done to rank the data positions. An example is shown in Fig. 6

   2. Cross connection of unpunctured data positions:

   (a) This step involves using an interleaver to connect the most error prone data positions of one component encoder to the least error-prone data positions of another component code.

   (b) An example is shown in Fig.7

   (c) For $M = 8$, the protograph is represented as in Fig. 7 by 8 different sub-interleavers $(\Pi_0, ..., \Pi_7)$, where for example Π_1 ensures that symbols at position 1 within a puncturing period in d are interleaved to position 3 within a puncturing period in d.

5   Layered Construction of ARP Interleavers

1. In [12] and [15] methods for the selection of the ARP interleaver parameters have been investigated. However the codes are still not suitable for high reliability applications.

2. in this research paper an alternative construction is approached which is based on a layered approach.

3. This method makes it possible to design codes with high d_{min} whiles introducing punturing constraints into the interleaver design. It also makes it possible to verify other design criteria such as minimum span and correlation girth.

4. To make parameter selection easier, the interleaver addresses $\Pi(i)$ are divided into different groups that are incrementaly defined.

5. The number of groups is Q since from (2) we can see that

$$\Pi(i + Q) \bmod Q = \Pi(i) \bmod Q$$

. Also each group corresponds to a given modulo Q value

6. The sequences in $\mathbf{d}$ and $\mathbf{d}'$ are then divided into these Q different layers each containing K/Q bits

7. The layer index l for bit $\Pi(i)$ in $\mathbf{d}$ and the layer index l' for bit i in $\mathbf{d}'$ are given by

$$l = \Pi(i) \bmod Q$$

and

$$l' = i \bmod Q$$

8. With this configuration, the interleaver is defined by a group of Q regular permutations each linking a layer l in $\mathbf{d}$ to its corresponding layer l' in $\mathbf{d}'$ as is shown in Fig 8.

9. For a given interleaver address i at layer l' the proposed interleaver choses the shift value $S(l') = T_{l'} + A_{l'}Q$ where $T_{l'} = 0, 1, ...Q-1$ is the inter-layer shift and $A_{l'} = 0, 1, ..., (K/Q) - 1$ is the intra-layer shift

10. $A_{l'}$ determines which position within layer l to connect to address i

11. $T_{l'}$ determines the value of the layer position l in $\mathbf{d}$ that will be connected to layer position l of $\mathbf{d}'$

12. To simplify the introduction of parity check constraints into the interleaver design Q is set as a multiple of M and for this study $Q = M$

13. $T_{l'}$ values are chosen in order to apply parity check constraints.

14. Note that for a given layer l' in $\mathbf{d}'$ the corresponding layer position l in $\mathbf{d}$ is obtained by $l = (Pl' + T_{l'}) \bmod Q$. Thus, a periodic connection pattern,with period Q, is established by the inter-layer shifts $T_{l'}$ between $\mathbf{d}'$ and $\mathbf{d}$, for a given P.

15. The Q layers of the interleaver structure can be defined incrementally by choosing their corresponding value $S_{l'}$.

A. Overall Interleaver Construction Method

For a set of design parameters ie minimum span, girth, interleaver length K and desired rate R, the ARP interleaver design process is as follows.

1. Select possible values of P which give minimum span and girth values greater that or equal to the desired values

2. For each possible value of P select the best Q value using the algorithm in Appendix B

3. Select best ARP interleaver by selecting the one with the best turbo code distance spectrum

It is worth noting that finding suitable values for minimum span and girth in order to find a good ARP interleaver is not a trivial problem. However, a possible selection strategy involves the following steps: first, evaluate the convergence of the algorithm in Appendix B for values of minimum span and girth set to their corresponding upper bounds. Then, if the algorithm does not converge, progressively reduce minimum span and girth until the algorithm converges to a group of ARP candidate interleavers

A. Summary of Puncture-Constrained interleaver design method

1. Select Puncturing mask with reference to section 4-A

2. Define puncturing constraints with respect to section 4-B and 4-C

3. Generate candidate interleavers with respect to section 5

6 Application Examples

7 Simulation Results