Spatial Coupling in Turbo Equalization

Mashauri, Mgeni Makambi; Lentmaier, Michael

Published in:
2020 IEEE Global Communications Conference, GLOBECOM 2020 - Proceedings

DOI:
10.1109/GLOBECOM42002.2020.9348218

2020

Spatial Coupling In Turbo Equalization

Mgeni Makambi Mashauri and Michael Lentmaier
Department of Electrical and Information Technology, Lund University, Sweden
Email: mgeni_makambi.mashauri@eit.lth.se, michael.lentmaier@eit.lth.se

***Abstract*—In this paper we consider spatial coupling in turbo equalization and demonstrate that the code design trade-off between the performance in waterfall and error floor regions can be avoided. We introduce three coupling schemes and compare their performances, where the first method introduces coupling between the encoder and the channel, while the second uses a spatially coupled (SC) code. In the third scheme we use both a coupled code and couple between the code and the channel. We show by computer simulations that, with spatial coupling, we can have good performance in both the error floor and the waterfall region with reasonable decoding latency by using a window decoder. We show this for both the maximum *a posteriori* (MAP) and linear minimum mean square (MMSE) equalizers.**

I. Introduction

Turbo equalization has been shown to be effective in mitigating the effect of intersymbol interference (ISI) by having the equalizer and decoder exchange soft information iteratively rather than each component working separately [1], [2]. This iterative exchange is an instance of belief propagation (BP) and can be analyzed by factor graphs [3]. An optimal receiver, however, is a MAP detector of the transmitted symbols taking into account the joint effect of the code constraints and the ISI channel.

The choice of codes for a turbo equalization usually involves a trade-off between the performance in the waterfall versus the performance in the error floor region. Choosing a weak code results in good waterfall performance but bad error floor, while choosing a strong code results in a bad waterfall performance but good error floor. This trade-off however is a result of the BP decoding process and not the codes themselves, that is, if we use a joint MAP detector with a sufficiently large codeword length the strong code would result in a better waterfall performance approaching the MAP threshold of the combined factor graph.

Spatially coupled low-density parity-check (SC-LDPC) codes have been shown to exhibit threshold saturation, whereby the BP threshold of the coupled ensemble approaches the MAP threshold of the uncoupled ensemble, [4], [5], [6]. In [7] it was proved that threshold saturation also occurs in spatially coupled turbo-like codes. Furthermore, [8] outlined a new trade-off between error floor and waterfall performance. In particular, it was shown that when spatially coupled, serially concatenated codes (SCCs) can have both better waterfall and error floor performance than parallel concatenated codes (PCCs). This is despite the fact that SCCs have a poorer waterfall performance than PCCs when not coupled.

This work was supported in part by the Swedish Research Council (VR) under grant #2017-04370. The simulations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at center for scientific and technical computing at Lund University (LUNARC).

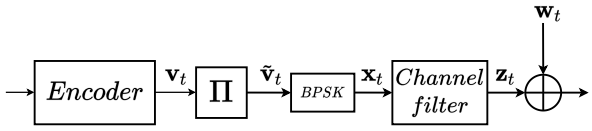

Fig. 1. Block diagram showing the transmitter and the ISI channel.

SC-LDPC codes were also investigated in coded modulation [9]. It was observed that with spatial coupling the performance of the codes was less sensitive to the chosen type of mapping, thus demonstrating some universality behavior with spatial coupling. In [10], spatial coupling between the code and detector for faster-than-Nyquist signaling and coded modulation was investigated. The output block of an encoder is split such that the input to the detector is a combination of various sub-blocks at different times. The work investigated the best way to split the output from the encoder to optimize convergence with the fewest number of iterations. Binary erasure channels with memory were studied in [11], [12]. It was shown that with SC-LDPC codes, threshold saturation also occurs in this channel. In [12] it was also shown empirically that SC-LDPC codes exhibit threshold saturation in an ISI channel with AWGN.

In this paper, we first demonstrate the challenge involved in the choice of codes by examining simulation results with a simple convolutional code versus a 5G LDPC code as component codes and explain the design trade-off using extrinsic information transfer (EXIT) charts. We then consider the application of spatial coupling in three different ways in turbo equalization to show that this trade-off can be avoided. In the first scheme, we couple between the encoder and channel leaving the code uncoupled. Then we derive an SC-LDPC code from the 5G LDPC code and use it without any coupling at the channel and lastly we couple both at the code and channel level. We show that for both MAP and MMSE equalizers, the spatially coupled code results in a larger gain than coupling at the channel input, while coupling both components is only slightly better than the coupled code alone. We hence managed to show that with spatial coupling we can use a strong code and obtain best performance in both the waterfall and error floor regions.

II. Turbo equalization

A model for the transmitter is shown in Fig. 1 where a block of K information bits, $\mathbf{u}_t$, is encoded by a code of rate

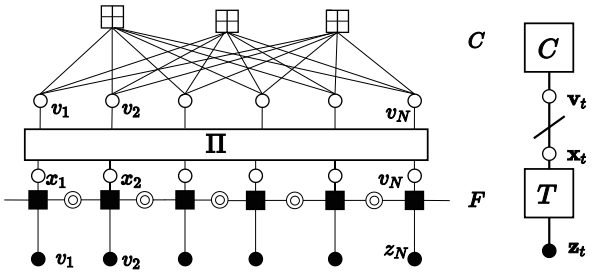

Fig. 2. (a) Combined factor graph of an LDPC code and channel without coupling (b) compact factor graph representation.

$R_c = K/N = 1/2$ to produce N code bits $\mathbf{v}_t$. These are then permuted to a new sequence $\tilde{\mathbf{v}}_t$ which is mapped to symbols $\mathbf{x}_t$ using binary phase shift keying (BPSK) modulation. The input symbol sequence $\mathbf{x}_t$ passes through the ISI channel filter with discrete impulse response $\mathbf{h}$ to obtain the output $\mathbf{z}_t$. The received symbol sequence is the sum of $\mathbf{z}_t$ and AWGN $\mathbf{w}_t$. The channel has M_h+1 taps $h[0], \ldots, h[M_h]$, where M_h is the channel memory. Throughout this paper we use the following channel from [13]:

$$\mathbf{h} = \begin{bmatrix} 0.277 & 0.46 & 0.688 & 0.46 & 0.277 \end{bmatrix} . \qquad (1)$$

The system can be represented by a factor graph, which shows the relationship between variables in the system [14]. As an example, Fig. 2 shows a factor graph of a regular $(3, 6)$ LDPC code and a channel. Black circles represent variables which have noisy observations at the receiver while white circles represent variables which are not observed at the receiver. State variables are represented by double circles and square nodes represent constraints in the code or channel. Following the notation in [8] we can represent the factor graph in a compact form by introducing node types and represent variables of equivalent distributions by a single node. For example, neglecting the edge effects we can represent a block of input symbols to the channel by a single node $\mathbf{x}_t$ as shown in Fig. $2(b)$ since the messages along connected edges have the same distribution. In the Figure, code constraints are denoted by $\mathcal{C}$ while the channel is denoted by $\mathcal{H}$.

To get good results in terms of bit error rate (BER) with relatively low complexity, the equalizer and decoder exchange information in a number of iterations. This iterative equalization and decoding is often called turbo equalization [2]. The MAP equalizer is an optimal equalizer and is implemented by a trellis following the work in [2]. Its complexity however grows as 2^{qM_h}, making it impractical when the memory of the channel or the modulation order (q) is large. A linear MMSE equalizer, on the other hand, though less accurate does not suffer from this exponential growth in complexity. The linear equalizer is implemented using a window approach [2].

The choice of the code usually involves a trade-off between the performance in the waterfall region and the error floor region. Using a weak code gives good waterfall performance but results in poor error floor while if we choose a code which is strong in an AWGN channel, it results in poor

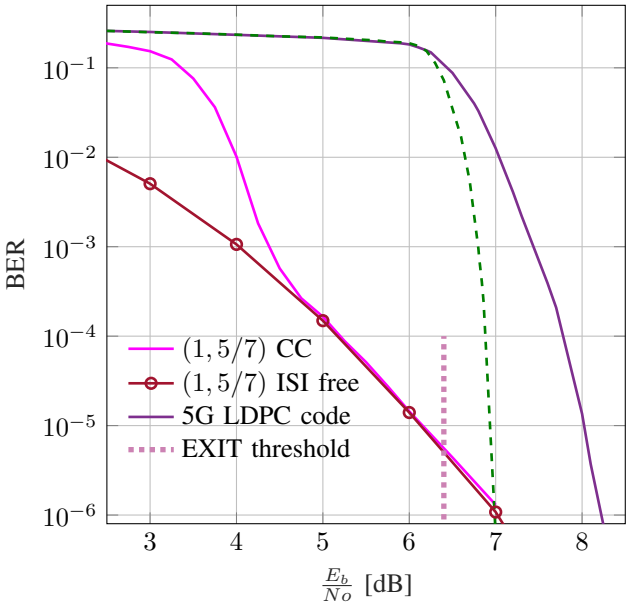

Fig. 3. Comparison of turbo equalization using a MAP equalizer with LDPC code and a convolutional code demonstrating the trade-off of choosing a weak versus a strong code. For both codes $N = 5120$. The dashed line shows the 5G code with a permutation of length 51200.

waterfall performance but good error floor. Two codes are used to illustrate this, a $(1, 5/7)$ systematic convolutional encoder (representinga weak code) and a 5G LDPC code (representing a strong code). Both codes have block length $N = 5120$ and the 5G code is obtained from base graph BG2 by lifting the graph by 256 as detailed in [15]. The number of iterations between the code and channel is 8 in all scenarios, while for the LDPC code we use 30 iterations within the code with a parallel schedule. It can be seen in Fig. 3 that when $\mathcal{C}$ is a convolutional code (CC) the waterfall performance is good but the error floor is bad, limited by the performance of the code in an ISI free case. While when $\mathcal{C}$ is the LDPC code it shows very poor waterfall performance. For example the BER is still above 10^{-1} at an SNR of 6 dB while for the convolutional code it is close to 10^{-5}.

This trade-off can be explained by observing the EXIT charts [16] for both codes and the channel in Fig. 4. The EXIT curve of the channel shifts up with increasing SNR, while those of the codes do not vary with SNR as the codes have no direct observations. The LDPC code has a nearly flat inverse EXIT function, which makes it intersect the channel at points of low mutual information (hence higher BER) for all SNR values below 6.4 dB, while the convolutional code being a weaker code has a shape which makes it intersect the channel at points of higher mutual information thus resulting in good waterfall performance. The waterfall performance of the 5G code can be improved slightly by increasing the permutation length as shown by the dashed line in Fig. 3 but it can not exceed the EXIT threshold shown by the vertical dotted line.

As a solution to this problem, a method using irregular convolutional codes, optimized together with precoders for a

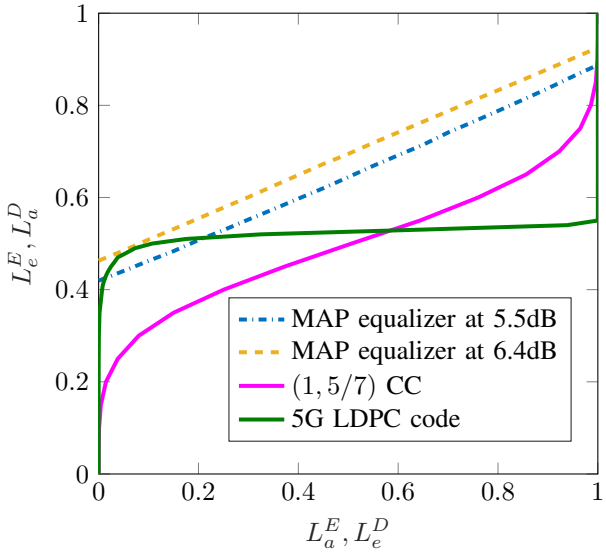

Fig. 4. EXIT chart predicting the performance of 5G LDPC code and $(1, 5/7)$ CC in turbo equalization.

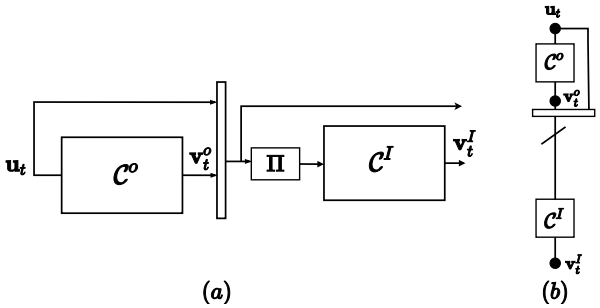

Fig. 5. (a) Block diagram of a SCC encoder (b) Compact graph representation.

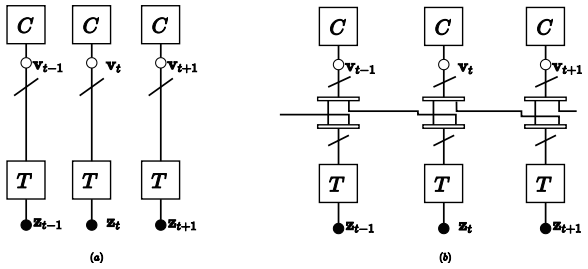

Fig. 6. Compact graph representation of (a) channel and code without coupling (b) Coupling between the encoder and channel. (In both figures the symbol $\mathbf{x}_t$ is not shown.)

desired waterfall performance is proposed in [3]. A drawback of such a solution is that it depends much on the channel and the equalizer type thus making it unsuitable in changing channel conditions. It can also result in bad error floors due to the weak component codes used in the optimization. Furthermore, since there is a limit to the choice of precoders determined by the memory of the channel [3], for some channels like the one we chose for this case the use of precoders does not show significant improvement in the performance. It is also not possible to use a precoder with linear equalizers without increasing the decoding complexity, which can be done with MAP equalizers [3].

But this trade-off is not inherent in the system itself but rather in the decoding process. If we were able to use a joint MAP decoder for both the code and the channel, the LDPC code would outperform the convolutional code in both waterfall and error floor regions. Spatially coupled codes have been shown to exhibit threshold saturation, where the BP threshold of the coupled codes approaches the MAP threshold of the underlying uncoupled codes.

III. Spatial Coupling in Turbo Equalization

With spatial coupling, memory is introduced in the factor graph of the turbo equalization system such that blocks at different time instants are interconnected. Three options are discussed.

A. Coupling between encoder output and channel

In [8] it was shown that the waterfall performance of serially concatenated codes (SCC) is improved significantly when they are spatially coupled. Comparing the factor graph of a SCC in Fig. 5 and that of the channel and code in Fig. 2, we observe that the two systems are equivalent. The channel acts as an inner code, $\mathcal{C}^I$, with rate 1 and non-binary outputs $\mathbf{z}_t$ which have noisy observations, while the symbols from the outer component code (equivalent to $\mathcal{C}^o$) are not transmitted which can be viewed as punctured symbols in the corresponding SCC code. Thus we can also couple the output of the code and the channel in a fashion similar to the one applied in [8] as follows.

Consider a normal system with the code and channel without any coupling. In this setting each block $\mathbf{v}_t$ of coded bits is permuted and put into the channel as shown in Fig. 6(a). In Fig. 6(b) each block of permuted bits $\tilde{\mathbf{v}}_t$ is split into two sub-blocks of equal lengths. One sub-block is put to the channel at time t while the other sub-block is connected another sub-block produced at time $t+1$. In this way we are effectively introducing blockwise memory between the code and the channel. The memory in this case is $m = 1$, since we need output from one previous block in the past in order to find the current input to the channel.

In general, for a coupled system with memory m, a block of N code bits $\tilde{\mathbf{v}}_t$ produced by the encoder at time t after interleaving by the permutation Π^1, is split into $m+1$ sub-blocks $\tilde{\mathbf{v}}_{t,0}, \ldots, \tilde{\mathbf{v}}_{t,m}$ each having $\frac{N}{m+1}$ bits. The input to the channel at time t is a sequence of symbols from the set $\{\tilde{\mathbf{v}}_{t-m,m}, \ldots, \tilde{\mathbf{v}}_{t,0}\}$ after being permuted by a second permutation Π^2. This is repeated for $t = 0 \ldots L-1$, where L is the length of the chain. The code bits $\tilde{\mathbf{v}}_t$ are set to zero for $t < 0$ and $t > L - m - 1$. This introduces known bits at the beginning and at the end of the chain incurring a rate loss which becomes negligible as L becomes large, as it can be

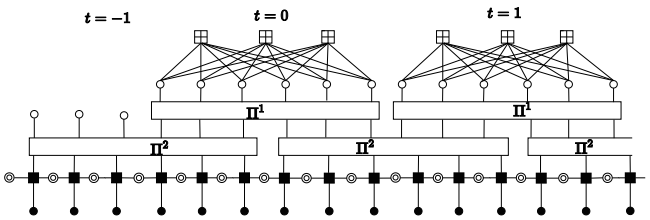

Fig. 7. Factor graph representation of coupling between code and channel for $m = 1$.

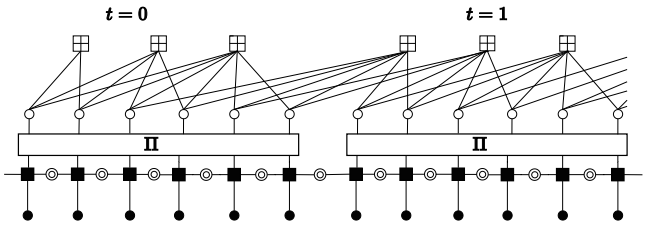

Fig. 8. Factor graph representation of turbo equalization with a coupled LDPC code.

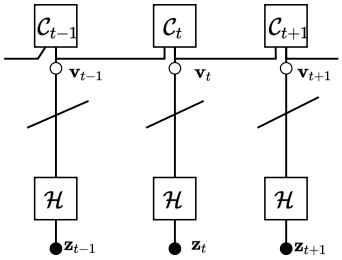

Fig. 9. Compact graph representation of turbo equalization with a spatially coupled code.

seen in the rate of the coupled system given by

$$\begin{aligned} R &= \frac{K(L-m)}{NL} \\ &= R_c\left(1-\frac{m}{L}\right) \ , \end{aligned} \tag{2}$$

which approaches the rate of the code R_c as L grows. These known bits, however, play an important role in improving performance of the belief propagation decoder as discussed in Subsection III-D. Fig. 7 shows the factor graph of such a system at the start of the chain.

B. Using a spatially coupled code

We can also use a spatially coupled code as the component code. We use a spatially coupled LDPC code constructed as described in [17]. Coupling is done on the protograph, followed by lifting the protograph by some chosen lifting factor and some permutations. With coupling memory $m = 1$, a variable node v_i with degree d_{v_i} splits its d_{v_i} edges into check nodes at time t and $t+1$. This splits the base matrix $\mathbf{B}$ into two sub-matrices $\mathbf{B}_0$ and $\mathbf{B}_1$ such that $\mathbf{B}_0 + \mathbf{B}_1 = \mathbf{B}$. The chain is terminated after L sections. The base matrix of the terminated convolutional protograph is given by

$$\mathbf{B}_{[0,L-1]} = \begin{bmatrix} \mathbf{B}_0 & & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & & \\ & \mathbf{B}_1 & \ddots & & \\ & & \ddots & \mathbf{B}_0 & \\ & & & \mathbf{B}_1 & \end{bmatrix} . \tag{3}$$

For simplicity we illustrate this in Fig. 8 with a regular (3,6) code with base matrices

$$\mathbf{B}_0 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} . \tag{4}$$

In this scheme every block of M bits at time t is permuted, mapped to symbols and sent over the channel. Fig. 9 shows the corresponding compact graph representation. This scheme is used in order to exploit the advantages of window decoding as elaborated in Section III-D, especially reduced latency, as it makes it possible to decode a block without waiting for the whole chain to be received.

C. Coupling at both the code and the channel level

Looking at the overall factor graph, we can see that it is possible to use a spatially coupled code and at the same time coupling the encoder output blocks, thus having a graph as shown in Fig. 10, whereas Fig. 11 shows the compact graph. The encoding of the code is not affected by the memory introduced at the input of the channel which is done as in Section III-A.

D. Window decoding of a coupled system

In order to get good results with coupling and minimize the latency it is essential to use window decoding. Since blocks which are m or more apart are not affecting each other directly and the effect further decays with increasing distance we can decode a block at time t by considering blocks within a window W, with $W \geq m+1$ [18]. With the scheme introducing memory between the code and the channel, as depicted in Fig. 12, the fist input block to the channel at time t contains known bits which corresponds to log-likelihood ratios (LLRs) with large magnitudes. These known bits result in improved estimates which are passed on to the code as good extrinsic information which can now correct more errors. As the exchange is repeated the code can correct more and more errors and spread the good beliefs in the window through the interconnections between blocks. For the SC-LDPC codes, the almost known bits are provided by the low-degree check nodes at the beginning of the chain in $\mathbf{B}_0$, which provide more reliable messages.

As the window moves to the next block, the elements from the first block are now mostly known and the process repeats itself. With window decoding the effect of known bits can spread into the graph with less complexity compared to a scheme which would involve the whole chain, as such a system

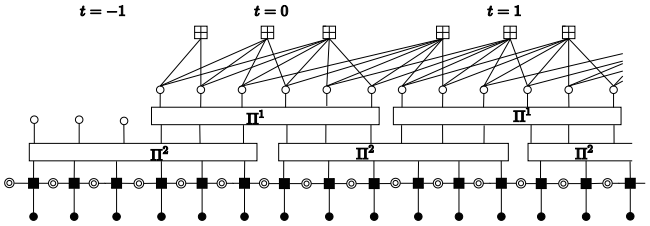

Fig. 10. Factor graph representation of turbo equalization with coupling at both the encoder and the channel level.

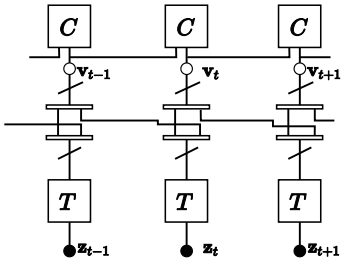

Fig. 11. Compact graph representation of coupling both the code and channel.

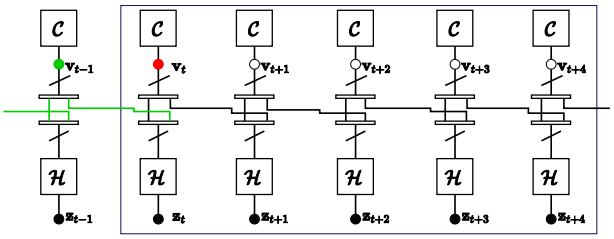

Fig. 12. Window decoder with $W = 5$, decoding block t with coupling between the code and channel only. The green dots represents decoded blocks of bits and the red ones the block being decoded.

will require a lot more iterations to spread the effect in the graph. Furthermore, the latency is reduced as we can decode a block without waiting for the whole chain to be received. Once we have waited for the first W blocks to decode the first block, subsequent blocks can be decoded after reception of only one more block. One drawback of the window decoding scheme is error propagation, since errors in one block can, in rare cases, affect all subsequent blocks. Solutions to this drawback are suggested in [19], [20] but in this paper we use window decoding without any modifications.

IV. Performance Analysis

The performance of the different forms of coupling is analyzed through computer simulations. For all types of coupling we consider $L = 100$ and the decoder uses $W = 5$. The capacity limit shown is the constrained capacity, where the input to the channel is restricted to be identically and uniformly distributed (i.u.d) and is computed numerically as described in [21]. Using the MAP equalizer we can see in Fig. 13 that coupling at the channel alone, results in a gain of about 2 dB for a BER below 10^{-5}. It is interesting to note that this occurs below the EXIT threshold of the code of

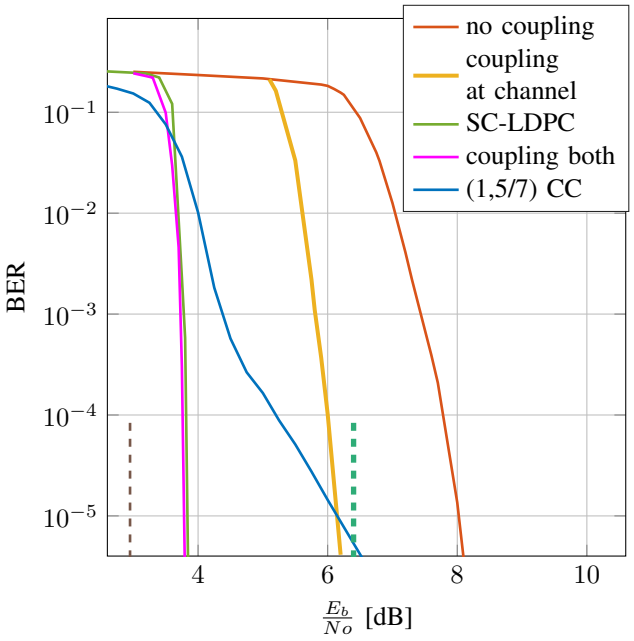

Fig. 13. The effect of different types of coupling using a MAP equalizer. The vertical dashed lines at 2.9 dB and 6.4 dB mark the i.u.d capacity limit and the uncoupled EXIT threshold, respectively.

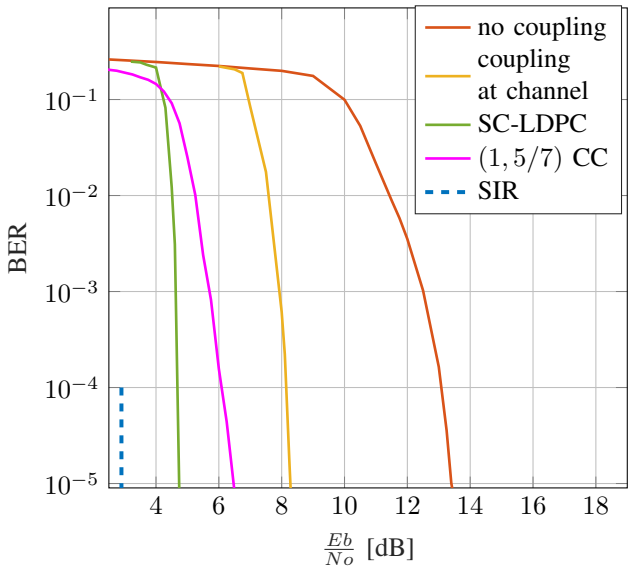

Fig. 14. Simulation results using MMSE equalizer showing the effect of coupling at the channel and using a coupled LDPC code.

6.4 dB, which can not be exceeded by the code alone even if we use a very long interleaver. When a spatially coupled code is used we observe a larger gain of around 4 dB with 1 dB gap to the i.u.d capacity limit. The gain is 2 dB more than the case with coupling at the channel alone, but it comes at the cost of changing the code and thus the encoder and decoder. Using both a spatially coupled code and coupling at the channel results in very small gain (about 0.05 dB) when compared to using a spatially coupled code alone.

When a linear MMSE equalizer is used (see Fig. 14), we observe a similar trend as in the MAP case. Coupling at the channel shows a gain of about 5 dB while the coupled code shows a gain of around 8.5 dB. The gains in each case are higher than their MAP counterparts. As a result of this the coupled code with linear equalizer is only 1 dB away from that of the MAP case, as opposed to 5 dB difference when no coupling is applied.

These increased performance comes at the cost of increased complexity at the decoder as each block (except at the boundaries) is visited 5 times (the window size) making the effective number of iterations five times more than the uncoupled case.

V. Conclusions

We illustrated three ways of coupling in turbo equalization and analyzed their performance. We showed that with spatial coupling the trade-off between the performance in the waterfall versus the error floor region in the choice of codes can be avoided. We also showed that with spatial coupling we can get good performance with both MAP or linear MMSE equalizers. Furthermore spatial coupling is superior to other approaches in the design of iterative receivers, which require a code adaptation to the particular channel, making them impractical in changing channel conditions. By using window decoding, the improved performance can be obtained with relatively low latency.

References

[1] C. Douillard, M. Jzquel, C. Berrou, D. Electronique, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *European Transactions on Telecommunications*, vol. 6, no. 5, pp. 507–511, 1995.

[2] M. Tüchler and A. C. Singer, "Turbo equalization: An overview," *IEEE Trans. Information Theory*, vol. 57, no. 2, pp. 920–952, 2011.

[3] M. Tüchler, "Turbo equalization," Ph.D. dissertation, Technical University Munich, 2005.

[4] M. Lentmaier, A. Sridharan, D. J. Costello, and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 5274–5289, 2010.

[5] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the bec," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 803–834, 2011.

[6] A. Yedla, Y. Jian, P. S. Nguyen, and H. D. Pfister, "A simple proof of maxwell saturation for coupled scalar recursions," *IEEE Transactions on Information Theory*, vol. 60, no. 11, pp. 6943–6965, 2014.

[7] S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled turbo-like codes," *IEEE Transactions on Information Theory*, vol. 63, no. 10, pp. 6199–6215, Oct 2017.

[8] ——, "Spatially coupled turbo-like codes: A new trade-off between waterfall and error floor," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3114–3123, May 2019.

[9] L. Schmalen and S. ten Brink, "Combining spatially coupled LDPC codes with modulation and detection," in *SCC 2013; 9th International ITG Conference on Systems, Communication and Coding*, 2013, pp. 1–6.

[10] T. Benaddi, C. Poulliat, and R. Tajan, "A general framework and optimization for spatially-coupled serially concatenated systems," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.

[11] S. Kudekar and K. Kasai, "Threshold saturation on channels with memory via spatial coupling," in *2011 IEEE International Symposium on Information Theory Proceedings*, 2011, pp. 2562–2566.

[12] P. S. Nguyen, A. Yedla, H. D. Pfister, and K. R. Narayanan, "Threshold saturation of spatially-coupled codes on intersymbol-interference channels," in *2012 IEEE International Conference on Communications (ICC)*, June 2012, pp. 2181–2186.

[13] J. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 1987.

[14] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[15] J. H. Bae, A. Abotabl, H.-P. Lin, K.-B. Song, and J. Lee, "An overview of channel coding for 5g nr cellular communications," *APSIPA Transactions on Signal and Information Processing*, vol. 8, p. e17, 2019.

[16] J. Hagenauer, "The EXIT chart - introduction to extrinsic information transfer in iterative processing," in *2004 12th European Signal Processing Conference*, 2004, pp. 1541–1548.

[17] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, "Spatially coupled LDPC codes constructed from protographs," *IEEE Transactions on Information Theory*, vol. 61, no. 9, pp. 4866–4889, Sep. 2015.

[18] M. Zhu, D. G. M. Mitchell, M. Lentmaier, D. J. Costello, and B. Bai, "Braided convolutional codes with sliding window decoding," *IEEE Transactions on Communications*, vol. 65, no. 9, pp. 3645–3658, 2017.

[19] M. Zhu, D. G. M. Mitchell, M. Lentmaier, D. J. Costello, and B. Bai, "Combating error propagation in window decoding of braided convolutional codes," in *2018 IEEE International Symposium on Information Theory (ISIT)*, 2018, pp. 1380–1384.

[20] M. Zhu, D. Mitchell, M. Lentmaier, and D. Costello, Jr., "Decoder error propagation mitigation for spatially coupled LDPC codes," in *International Symposium on Information Theory and Its Applications, ISITA 2020*, 2020.

[21] D. M. Arnold, H. A. Loeliger, P. O. Vontobel, A. Kavcic, and W. Zeng, "Simulation-based computation of information rates for channels with memory," *IEEE Transactions on Information Theory*, vol. 52, no. 8, pp. 3498–3508, Aug 2006.