Joint Equalization and LDPC Decoding

Werner Henkel, *senior member, IEEE*, Nazia S. Islam, and M. Ahmed Leghari
Jacobs University Bremen
Bremen, Germany
Emails: werner.henkel@ieee.org, {w.henkel, n.islam, m.leghari}@jacobs-university.de

***Abstract*—Encoding and channel convolution and the corresponding equalization and decoding are a serial concatenation. One might consider the solution to be iterative equalization and decoding in a Turbo fashion. We are showing that it is not necessary to run LDPC decoding and equalization as separate entities in a Turbo manner, but integrate decision-feedback equalization (DFE) directly into the LDPC decoder, performing DFE-like operations as part of the LDPC message passing.**

I. Introduction

A very common equalizer structure is the decision feedback equalizer shown in Fig. 1, [1], [2].

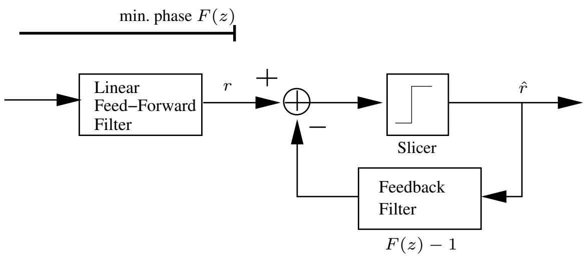

Fig. 1. Decision Feedback Equalizer

After the feed-forward filter, the overall transfer function is minimum phase, hence the energy is concentrated at the beginning of the impulse response. The slicer cannot easily be replaced by a decoder due to the decoding delay. One typical solution is Tomlinson-Harashima precoding [3], [4], where the feedback filter is moved to the transmitter, thereby avoiding any error propagation. The structure is shown in Fig. 2, where the feed-forward filter is seen as part of the minimum phase overall channel $F(z)$.

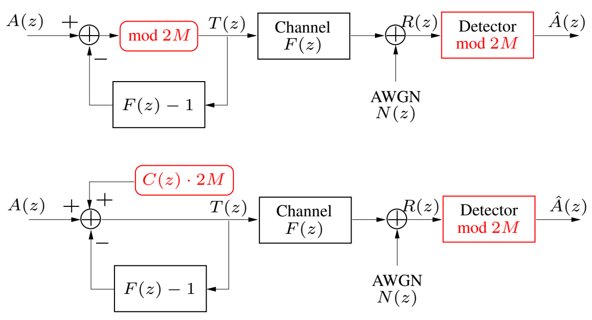

Fig. 2. Tomlinson-Harashima precoding

Tomlinson-Harashima precoding requires a duplex channel, since the precoder coefficients need to be known at the transmitter. At best, some compromise coefficient might be chosen together with leaving the remaining equalization to the feed-forward filter.

Another typical solution is suitable for convolutional codes, where one computes the feedback operations for all paths in a Viterbi algorithm, thereby, of course, increasing the complexity substantially.

With the invention of Turbo coding by Berrou et al. [5], many applications apart from parallel and serial concatenation of codes were investigated, the serial concatenation of coding and an ISI (inter-symbol interference) channel and the corresponding iterative treatment of equalization and decoding being one of them [6]–[10]. Two figures from Tuchler and Singer [6], [7] are summarized here as Fig. 3 to show the typical Turbo decoding of the serial structure, without defining all the annotations that they are using.

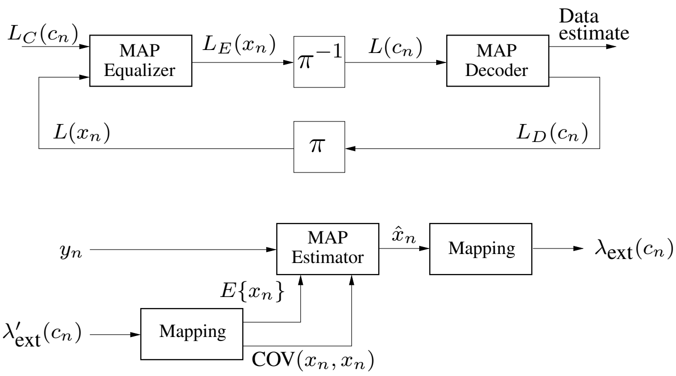

Fig. 3. Turbo equalization and its central operation to realize a MAP equalizer, from [7], Fig. 3 and [6], Fig. 7

In [6], the authors describe the transition from LLRs (log-likelihood ratios) from a decoder to analog information using $\tanh(\lambda'_{\text{ext}}(c_n)/2)$, which we will also later introduce, commonly named as "soft bit". After the MMSE estimator, a mapping determines the LLR again from the analog estimate $\hat{x}_n$. The MMSE estimate is the common

$$\hat{\mathbf{x}} = \mathbf{H}^T(\sigma^2\mathbf{I}_L + \mathbf{H}\mathbf{H}^T)^{-1}\mathbf{y} \tag{1}$$

with the Toeplitz convolutional channel matrix $\mathbf{H}$, not to be mistaken with the parity-check matrix that will otherwise be used in this paper.

In a Turbo-like scheme, instead of an MMSE variant, one could think of realizing the channel equalization by a soft-output Viterbi algorithm, BCJR, or windowed BCJR algorithm, if the number of states is or can be made sufficiently small. This allows straight forward exchange of LLRs between the equalizer and the decoder. If the decoder is a Turbo or LDPC decoder, this would mean some iterations there in between the equalizer operations.

The next section discusses, how a separate realization with "outer" Turbo-like iterations can be avoided and how a DFE can directly be integrated into the Tanner graph for decoding an LDPC code.

II. Integrated Equalization and Message Passing Decoding

In a recent paper at ISTC 2018 [11], we showed how Markov properties of a source can directly be integrated into the LDPC decoding without the need of an iterative, Turbo-like procedure exchanging information between a BCJR algorithm handling the Markov source and message passing inside the Tanner graph of the LDPC code. Although the iterative procedure is slightly superior to the integrated one, the complexity is lower for the integrated solution, since no separate BCJR algorithm is needed. An ISI channel, however, creates the dependencies after encoding. Hence, the sequence between a component creating memory and the LDPC encoding is the opposite for both applications. One has a Markov source followed by an LDPC encoder, the other consists of an LDPC encoder followed by an ISI channel. The Markov source properties could be integrated by additional LLR forwarding between variable nodes. An equalizer is, however, an operation related to analog signals. This means an extension of the Tanner graph with a transition to analog signals and also the reverse to process LLRs.

For illustration purposes, we assume a very short minimum phase impulse response after the feed-forward filter to be described in z domain by $F(z) = 1 + f_1 z^{-1}$ and for the actual simulations, for now, we have only taken $F(z) = 1 + 0.5z^{-1}$. The generalization to higher orders is obvious and later shown in the formulation of the modified variable node equation.

DFE is a deterministic operation, wherein a hard decided value after the slicer is subtracted from the next sample(s), thereby eliminating the tail $F(z) - 1$ of the impulse response in a denoised fashion. In an LDPC Tanner graph with message-passing decoding, the variable nodes represent the decisions, although their quality will improve with time and the values are not hard decided but represented by log-likelihood ratios, which we call ξ_i at the i^{th} variable node. The LLR ξ_i represents a so-called soft bit, which we write as

$$\begin{aligned} \lambda(\xi) = E_X\{x\} &= (+1) \cdot P(X = +1) + \\ &\quad +(-1) \cdot P(X = -1) \\ &= \frac{e^{\xi}}{1 + e^{\xi}} - \frac{1}{1 + e^{\xi}} \\ &= \frac{e^{\xi} - 1}{1 + e^{\xi}} = \frac{e^{\xi/2} - e^{-\xi/2}}{e^{\xi/2} + e^{-\xi/2}} \\ &= \tanh(\xi/2) \ . \end{aligned} \quad (2)$$

This allows to represent the LLRs as analog values to be subtracted after weighting with f_1 (f_i in general for higher order ISI channel models) from the intrinsic input to the next variable node. The intrinsic information depends on the analog value in the form

$$L_{\text{intr}i} = \ln \frac{p(r_i|c_i = +1)}{p(r_i|c_i = -1)} = \ln \frac{e^{+2r_i/2\sigma_n^2}}{e^{-2r_i/2\sigma_n^2}} = \frac{2}{\sigma_n^2} \cdot r_i . \quad (3)$$

The analog decoder input r_i is the one to be used in the equalizer operations according to

$$r_i - f_1 \cdot \lambda(\xi_{i-1}) = r_i - f_1 \cdot \tanh(\xi_{i-1}/2) \ , \quad i = 1, 2, \ldots \quad (4)$$

where i is the variable node counter.

Equation (4) is illustrated in Fig. 4. Comparing with the standard DFE structure in Fig. 1 will make the operation (4) obvious. In the DFE, the effect of the tail of the impulse response is subtracted from subsequent symbols. Now, instead of a hard decision value after a slicer, we take the variable-node estimate instead, which combines *all* impinging LLRs. However, not the LLRs themselves are used, but, since the equalizer is processing analog values, the corresponding soft-value is computed following Eq. (2). Clearly, Fig. 4 shows a DFE extension integrated into the Tanner graph. Some care has to be taken regarding the scheduling of the operations. Especially, one should not perform any up-front equalizing step just based on the intrinsic (not code protected) input, i.e., not running the equalizer on the channel output directly, without having passed the Tanner graph share of the LDPC code. Note, the direct equalization would process very noisy values, not yet protected by any extrinsic information. This will lead to inferior performance.

At the beginning, the intrinsic information will be used to initialize the variable node contents to feed the DFE graph segment for the following variable-to-check node equation and to feed the first run of the check-node equation. The first actual variable-node operation, i.e., from variable to check node will combine the direct intrinsic information, the impulse-response-tail cancellation through tanh and f_i and the

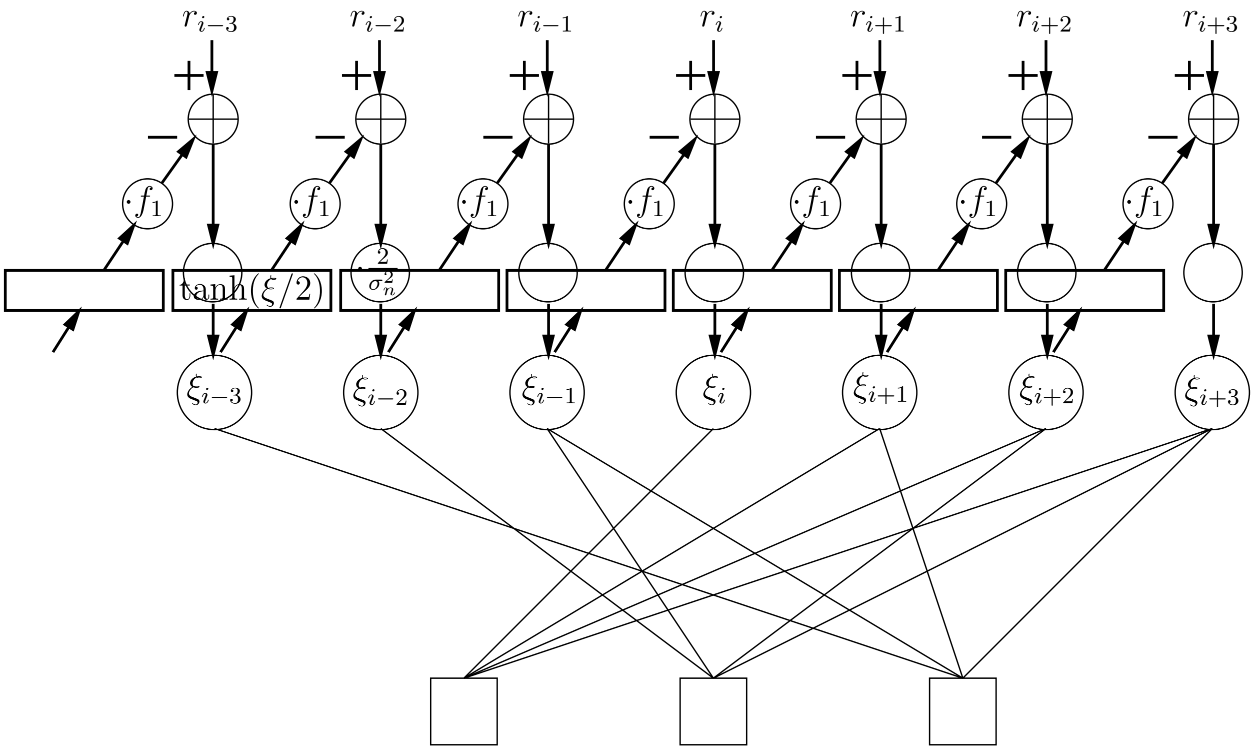

Fig. 4. LDPC integrated equalization

other incoming LLRs from check nodes. The variable and check node equations are:

$$v_m^{(l)}(i) = \frac{2}{\sigma_n^2}\left[r(i) - \sum_{m=1}^{M} f_m \tanh(\xi_{i-m})\right] + \\ + \sum_{k=1,k\neq m}^{d_v(i)} u_k^{(l-1)}(i), \forall m = 1..d_v(i) \ , \quad (5)$$

$$\tanh\frac{u_k^{(l)}(j)}{2} = \prod_{m=1,m\neq k}^{d_c(j)} \tanh\frac{v_m^{(l)}(j)}{2}, \forall k = 1..d_c(j) \ . \quad (6)$$

M denotes the memory of the channel (assumed to be minimum phase), i.e., the length of the feedback filter of a standard DFE. The check-node equation (6) is the standard one.

$v_m^{(l)}(i)$ is the variable-to-check LLR at the mth edge of the ith variable node. $u_k^{(l)}(j)$ is the check-to-variable LLR at the kth edge of the jth check node. $d_v(i)$ and $d_c(j)$ denote the variable and check node degree at variable and check nodes, i and j, respectively. The term in rectangular brackets represents the analog DFE-like operation with the usual factor given by Eq. (3).

Note that ξ_{i-m} denotes the sum of all incoming LLRs, which in usual LDPC decoding would only be computed as final result after all iterations are completed. With our integrated DFE/LDPC solution, this overall sum will be needed at every iteration.

In Fig. 4, we show the case with $M = 1$.

III. Simulation result

In the following, we show first results based on an LDPC code of length 2048 of rate 1/2 with the following variable and check node degree polynomials:

$$\lambda(x) = 0.28286x + 0.39943x^2 + 0.31771x^7, \quad (7)$$

$$\rho(x) = 0.6x^5 + 0.4x^6. \quad (8)$$

For determining the actual systematic triangular **H** matrix, we used the PEG algorithm in a "zigzag" fashion according to [12]. We show **H** as a dot pattern in Fig. 5.

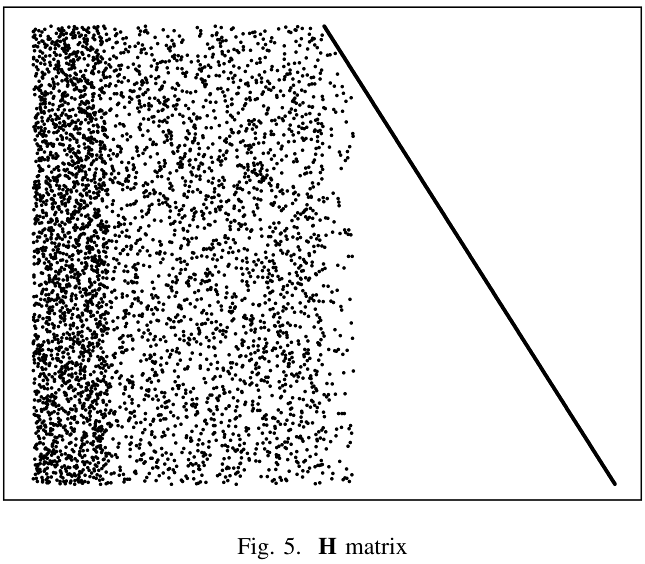

Fig. 5. **H** matrix

As a termination criterion, we used 100 errored words. BER results are presented in Fig. 6. We show results for the ISI-free case, a sequential (non-iterative) equalization and the proposed integrated equalizer/LDPC realization. For the final version, we will also add a Turbo-MMSE/LDPC alternative from [6], [8]. However, we do not expect big performance differences between the Turbo scheme and our proposal. The current results are using a very simple ISI channel with two taps of 1 and 0.5. This was chosen for a first check of the validity of the approach. We will also then add simulation result for another ISI channel.

Note that the E_b/N_0 scaling in Fig. 6 uses a two-sided noise power spectral density. The Shannon limit for rate 1/2 is then at 3 dB.

Figure 6 clearly shows that the integrated approach is working as expected. It shows the intended gain due to the iterative inclusion of the equalization. The number of iterations was limited to 20. One recognizes that the gain compared with a sequential procedure is slightly growing with the number of iterations. Sequential means to first equalize, followed by LDPC decoding without any interplay. As expected in Section II, this is, of course, worse. We also recognize a flooring at a BER of around 10^{-6}. This flooring is probably due to shorter cycles that are resulting from the addition of extensions bridging neighboring variable nodes. This indicates that the next aspect to be looked into is a modification of the design of the H-matrix, e.g., modifying the PEG zigzag design to take into account the links caused by the equalizing paths. Going further, one could, of course, also think of modifying density evolution to account for the modification of the intrinsic information entering the variable nodes ξ_i in Fig. 4. One of the convergence formulas describing the mutual information from variable to check nodes used in linear programs for determining the optimum degree distributions is typically written as

$$x_{vc}^{(l)} = \sum_{i=2}^{d_{\mathrm{vmax}_j}} \lambda_i J\left(\frac{2}{\sigma^2} + (i-1)\, J^{-1}\left(x_{cv}^{(l-1)}\right)\right), \quad (9)$$

which is, of course, the counterpart of (5) when omitting the sum representing the equalization. This would mean, when actually modifying density evolution formulas, i.e., the convergence formula, one would have to include exactly this component as an add-on to the intrinsic $2/\sigma^2$ term. This requires some approximation if one likes to stick to the assumption of Gaussian LLRs, i.e., working with consistent densities. The actual density evolution (that almost nobody does for obvious complexity reasons) would mean to take the processing steps into account with its transform properties on the densities and convolving densities according to the summation or multiplying the corresponding characteristic functions. Practically, however, one cannot perform density evolution for varying channels. One should imagine that this would mean a linear program for every channel change, optimizing the degree distributions, followed by PEG or similar algorithms to construct the parity-check matrix. Then, the resulting H matrix has to be communicated to the transmitter. This is a possibility for an assumed fixed channel. A modification of PEG under a certain memory assumption and hence, corresponding links between variable nodes will be investigated as a next step, having realized the flooring seemingly due to the introduced dependencies.

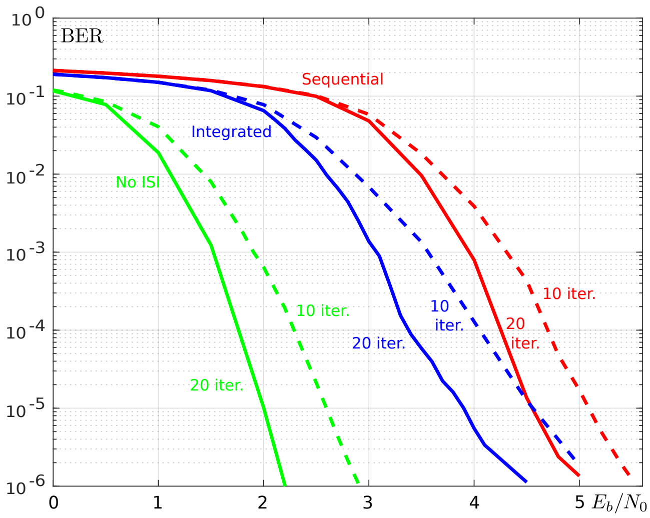

Fig. 6. Simulation results with 10 and 20 iterations

IV. Conclusions

Initiated by our earlier works on the inclusion of Markov source properties into the LDPC decoding, this paper has extended the joint treatment to LDPC codes followed by ISI channels. We proposed to integrate equalization steps into the actual LDPC message passing. This requires to move to analog domain from a current variable-node estimate and after the DFE-like operation move back to log-likelihood ratios (LLRs) for a modified variable node equation. In there, for computing an LLR on an outgoing edge to a check node, all other incoming LLRs from check-nodes are now combined with intrinsic and ISI canceling components.

First simulation results show that indeed the integration of equalization into LDPC decoding is functioning as expected and delivers a coding gain with respect to non-iterative equalization and decoding. It does, of course, not reach the performance of the non-ISI pure AWGN case, just as in [6]. We realized a flooring due to introduced cycles from the inclusion of equalizing links into the Tanner graph. Later works will address this issue by adjusting the algorithms for the parity-check matrix design.

ACKNOWLEDGMENT

The underlying question for this work goes back to a discussion with Liuling Ling (Huawei) at ISTC 2018, which we gratefully acknowledge.

REFERENCES

[1] J. Proakis, *Digital Communications*. Electrical engineering series, McGraw-Hill, 2001.
[2] J. R. Barry, D. G. Messerschmitt, and E. A. Lee, *Digital Communication: Third Edition*. Norwell, MA, USA: Kluwer Academic Publishers, 2003.
[3] M. Tomlinson, "New automatic equaliser employing modulo arithmetic," *Electronics Letters*, vol. 7, pp. 138–139, March 1971.
[4] H. Harashima and H. Miyakawa, "Matched-transmission technique for channels with intersymbol interference," *IEEE Transactions on Communications*, vol. 20, pp. 774–780, August 1972.
[5] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes," in *Proceedings of ICC '93 - IEEE International Conference on Communications*, vol. 2, pp. 1064–1070 vol.2, May 1993.
[6] M. Tuchler and A. C. Singer, "Turbo equalization: An overview," *IEEE Transactions on Information Theory*, vol. 57, pp. 920–952, Feb 2011.
[7] M. Tuchler, R. Koetter, and A. C. Singer, "Turbo equalization: principles and new results," *IEEE Transactions on Communications*, vol. 50, pp. 754–767, May 2002.
[8] R. Koetter, A. C. Singer, and M. Tuchler, "Turbo equalization," *IEEE Signal Processing Magazine*, vol. 21, pp. 67–80, Jan 2004.
[9] V. D. Trajkovic, M. Fu, and P. J. Schreier, "Near-capacity turbo equalization using optimized turbo codes," in *2007 Australasian Telecommunication Networks and Applications Conference*, pp. 480–484, Dec 2007.
[10] V. D. Trajkovic, M. Fu, and P. J. Schreier, "Turbo equalization with irregular turbo codes," in *2007 4th International Symposium on Wireless Communication Systems*, pp. 153–157, Oct 2007.
[11] N. S. Islam and W. Henkel, "Information forwarding in LDPC decoding for Markov sources," in *2018 IEEE 10th International Symposium on Turbo Codes & Iterative Information Processing (ISTC)*, pp. 1–5, Dec 2018.
[12] Xiao-Yu Hu, E. Eleftheriou, and D. . Arnold, "Progressive edge-growth tanner graphs," in *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270)*, vol. 2, pp. 995–1001 vol.2, Nov 2001.