Application of Protograph-based LDPC Codes for UWB Short Range Communication

Grosa, Patrick; Fonseca dos Santos, Andre; Lentmaier, Michael; Rave, Wolfgang; Fettweis, Gerhard

Published in:
2010 IEEE International Conference on Ultra-Wideband

DOI:
10.1109/ICUWB.2010.5616834

2010

Application of Protograph-based LDPC Codes for UWB Short Range Communication

Patrick Grosa, Andre Fonseca dos Santos, Michael Lentmaier, Wolfgang Rave and Gerhard Fettweis
Vodafone Chair Mobile Communications
Dresden University of Technology (TU Dresden), 01062 Dresden, Germany
Emails: {grosa, andre.santos, michael.lentmaier, rave, fettweis}@ifn.et.tu-dresden.de

***Abstract*—We studied the behavior of an iterative receiver using protograph based LDPC codes (PG-LDPCC) in an UWB short range communication system. An EXIT-Chart-Analysis is applied by means of density evolution of protographs. We show that AWGN-optimized protograph ensembles perform also well on frequency-flat UWB channels, outperform rate-compatible punctured convolutional codes (RCPC), but degrade with increasing intersymbol interference (ISI). Slight modifications of the underlying protograph can improve this performance, but lead to a trade-off between the minimum required SNR and the minimum achievable BER. It is shown that both thresholds can be predicted by the applied tools.**

***Index Terms*—Turbo Equalizer, Density Evolution, Protographs, EXIT-Charts, UWB**

I. Introduction

The demand for wireless high data rate services has been rapidly growing for years. Ultra-wideband (UWB) communication has been considered to satisfy this need in the field of short-range communication. However, achieving data rates of about 1 Gbit/s typically requires a large bandwidth. Such ultra-wide band systems experience a long channel impulse response with a high number of resolvable paths. One way to mitigate the effects of ISI is the usage of turbo equalization, i.e. a continuous improvement of the equalization due to an iterative exchange of information between equalizer and channel decoder. However, such systems require the adjustment of the behavior of both components in order to optimize the overall performance. Several different approaches have been discussed in this area, e.g. in [1] or [2].

Throughout this work, we study the interaction of so-called protograph based low-density parity-check codes (PG-LDPCC) and a soft equalizer (consisting of a soft mapper and demapper and equalizer) as described in [3]. In our case, the analysis of LDPC codes is done in a very accurate way by density evolution, that neither reduces the inner decoder to a single iteration as in [2] nor has to rely on the assumption of Gaussian distributed messages within the channel decoder. However, the original version of density evolution does not consider the structure of the code. It is possible to overcome this issue by using protographs. Hence, we can view PG-LPDCCs as structured LDPC codes. Another benefit of protographs is the possibility to construct quasi-cyclic codes with lower hardware complexity .

In the following section the considered communication system is described. Subsequently, a brief introduction into protgraphs and their analysis is given. The results of this analysis are used to characterize the overall system. The predicted performance is compared with simulations and discussed.

II. System Description

We consider a single carrier transmission system with one antenna at transmitter and receiver side. A data word $\mathbf{d} \in \{0,1\}^{K \times 1}$ is encoded and the resulting vector of coded bits $\mathbf{v} \in \{0,1\}^{N \times 1}$ is randomly interleaved, yielding the vector $\mathbf{x}$. Subsequently, the coded interleaved bit vector is linearly mapped onto the elements of the complex symbol vector $\mathbf{s_d} = \left[s_{d_1} \; \dots \; s_{d_p} \; \dots \; s_{d_P}\right]^T$ of length $N/Q = P$, where 2^Q is the size of the modulation alphabet.

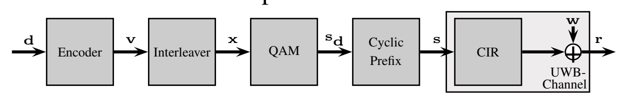


Fig. 1. Transmission chain

A cyclic prefix is added to the symbol vector in order to enable equalization in the frequency domain. The transmitter is illustrated in Figure 1 including the channel. We assume a multipath channel, with channel impulse response (CIR) given by $\mathbf{h} = [h(0) \; \dots \; h(l) \; \dots \; h(L-1)]$. We assume block fading, i.e. the CIR does not change during the transmission of one block. In addition, white Gaussian noise samples $\mathbf{w} \in \mathbb{C}^N$ with variance σ_w^2 are added to the symbol vector and the received samples $r(i)$ at time i are observed as:

$$r(i) = \sum_{l=0}^{L-1} h(l)s(i-l) + w(i) \qquad . \qquad (1)$$

The distorted signal is received and equalized at the receiver side (Figure 2). Then, the symbol vector is demapped and the extrinsic log-likelihood-ratios (LLRs) $L_e^E(\mathbf{x})$ are computed for the coded interleaved bits. The deinterleaved vector of LLRs, denoted by $L_a^D(\mathbf{v})$, is then forwarded to the channel decoder to perform a first decoding attempt. The extrinsic LLRs of the decoder, i.e. the difference between a-posteriori LLRs, denoted by $L_p^D(\mathbf{v})$ and a-priori LLRs, is designated by $L_e^D(\mathbf{v})$ and is fed back to the soft equalizer. In the next iteration, these LLRs are used to improve the equalization of the signal and the result is forwarded to the decoder, again. This process is repeated for a fixed number of iterations.

III. Protograph Ensembles

A. Protograph LDPC Codes

In Gallager's original work on Low-Density Parity-Check codes (LDPCCs) [4], he did not only introduce regular LDPCCs (same number of ones in each row and each column,

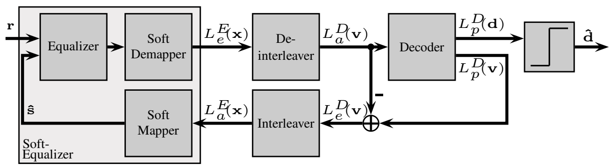
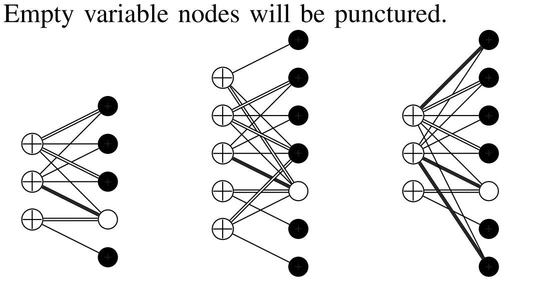


Fig. 2. Receiver chain

respectively) and the belief propagation decoding scheme, but he also proposed an algorithm to construct structured LDPCCs composed of individual permutation matrices. In [5], it was shown that these codes become Quasi-Cyclic-Codes (QC-Codes) if permutation matrices are replaced by circulants. Since irregular LDPCCs have a very good performance, the interest in irregular and structured LDPCCs has led to an LDPC ensemble description called *protographs* [6]. A protograph is a bipartite graph $G(\mathcal{V}, \mathcal{C}, \mathcal{E})$ that consists of a set of variable nodes $v_i \in \mathcal{V}$ of degrees $d_{v_i},\ i = 1,..,N_P$, a set of check nodes $c_r \in \mathcal{C}$ of degrees $d_{c_r},\ r = 1,..,M_P$, and a set of edges $e_t \in \mathcal{E},\ t = 1..t_{max}$ that connect the nodes. We further distinguish between the set of edges $e_j^{v_i} \in \mathcal{E}(v_i),\ j = 1,..,d_{v_i}$ and $e_k^{c_r} \in \mathcal{E}(c_r),\ k = 1,..,d_{c_r}$ emanating from variable node v_i and check node c_r, respectively. The variable node v_i is connected by its j-th edge to the check node c_r with its k-th edge if $e_j^{v_i} = e_k^{c_r} = e_t$. As an example, the so-called Accumulate-Repeat-Jagged-Accumulate (ARJA) protographs (proposed in [7]) are illustrated in Figure 3, where nodes with a plus represent check nodes and all other nodes are variable nodes. Empty variable nodes will be punctured.



(a) ARJA (R=1/2) (b) ARJA (R=1/3) (c) ARJA (R=2/3)

Fig. 3. ARJA protographs for different rates

A corresponding representation of a protograph is an $M_P \times N_P$ bi-adjacency matrix $\mathbf{B}$, which is called the *base matrix* of a protograph. Each row and each column corresponds to a check node and a variable node, i.e. the entry in column i and row r is equal to the number of edges between variable node v_i and check node c_r.

The construction of derived LDPCCs is done by replacing each 0 in $\mathbf{B}$ by an all-zero-matrix and each $a > 0$ of $\mathbf{B}$ by the addition of a permutation matrices, whereas the position of the one in each row must be different for every permutation matrix of size T. This procedure is equivalent to the copy-and-permute method proposed in [6]. Both procedures preserve the structure of the protograph, but multiple edges are spread between the copies. Both algorithms are used to *lift* a given protograph. The set of matrices $\mathbf{H}$ that can be derived from a given protograph by all possible combinations of size T permutation matrices defines an ensemble of protograph based LDPCC (PG-LDPCC) codes of length $N = TN_P$. The main point is that both procedures conserve the structure of the protograph. However, these procedures do not specify the permutation matrices, which can than be arbitrarily chosen. For that purpose two similar approaches, the Progressive-Edge-Growth (PEG) [8] algorithm and the ACE algorithm [9], were introduced and adapted to the protograph case. Moreover, both algorithms can be combined by using the result of the first algorithm as a protograph for the second algorithm.

B. Density Evolution for Protograph Ensembles

Introduced in [10] the density evolution is used for the construction of irregular LDPC codes. Since all protograph based construction methods of LDPC codes conserve the structure of the protograph, it is reasonable to apply the density evolution within the protograph. The basic idea of density evolution is to track the probability density function (pdf) of the LLRs during the decoding process. Therefore, equivalent pdf calculations for the update functions must be found. Since specifying a transformation of the check node update in closed form is not feasible, it is necessary to use the quantized version of the density evolution, which is explained in detail, e.g. for protograph based LDPC codes in [6].

IV. EXIT-Chart Analysis

For the study of an iterative (also called Turbo) system, the amount of information that is exchanged between the different components plays an important role. An appropriate way to analyze the system and to visualize the amount of information is the so-called *Ex*trinsic *I*nformation *T*ransfer (EXIT) Chart. Introduced in [11] for parallel concatenated codes it became also popular for LDPC codes and other iterative systems as it is shown, e.g. in [12].

Starting point is the extrinsic information transfer function, i.e. the function that maps the a-priori mutual information (MI) I_a to the extrinsic MI I_e of a component. An EXIT-Chart illustrates the transfer function of two components in a single graph, whereas the axes of the second component are swapped by using the inverse mapping from I_e to I_a. The straight forward way to estimate the EXIT function of a component is to generate random variables with a specific pdf, yielding to different values of a-priori MI, at the input of a component and to measure the extrinsic MI at its output. This approach is used to determine the transfer function of the equalizer (based on [12], detailed explanations for the used equalizer can be found in [3]). A common way to analyze LDPC decoder is to assume Gaussian distributed messages that circulate inside the decoder. Then, the transfer function can be calculated by means of the J-function. This approach is explained in more detail in e.g. [13].

A. EXIT Function by means of Density Evolution

However, we use a more precise way to obtain the EXIT function by means of density evolution, where we use the general relationship of the MI between the equally likely bit $V \in \{\pm 1\}$ and the respective LLRs $L(V)$ for symmetric and consistent L-values, that is given by

$$I(L(V); V) = 1 - \mathrm{E}\left\{\log_2\left(1 + \exp^{-L(V)}\right)\right\} . \qquad (2)$$

Due to the fact that the density evolution tracks the pdfs of the LLRs during the decoding process, we even know the pdf of the a-priori LLRs and the pdf of the extrinsic LLRs after decoding. Hence, we can use this knowledge to compute the a-priori MI and the extrinsic MI by Eq. (2).

V. Analysis and Simulation Results

For the study and simulation we use two different channel types. The first one is the NLOS desktop channel model from [14], which is a relatively frequency-flat channel. The second one has an exponential power delay profile with 10 taps and a decay rate of 10/tap ($\text{pdp}(l) = \text{e}^{-\frac{l}{10}}$; $l = 0, .., 9$).

As a starting point for our studies, we used the earlier mentioned protographs known to have a good performance for an AWGN channel. A useful property of this protograph is the possibility to derive different code rates with only slight changes. These protographs are illustrated in Figure 3. At first we lift the protographs by the PEG algorithm with $T = 7(14)$ and afterwards by the ACE algorithm with $T = 128$, where we define the permutation matrices of the second step to be circulants, to achieve two different block lengths of $N = 4480(8960)$. For comparison, we use rate-compatible punctured convolutional (RCPC) codes as introduced in [15]. All codes in this family are obtained from a mother code of rate 1/4 using the generator polynomials $G_{oct} = \{23, 35, 27, 33\}$. We use a block length of $6000(12000)$ bits, for which it is known that the performance of a convolutional code does not depend on the block length.

A. Low ISI Channels

The first performance estimate is obtained by the EXIT-Chart analysis. We determine the EXIT function of the NLOS desktop channel model for different signal-to-noise ratios (SNR) as depicted in Figure 4. As one can see, the transfer functions of the equalizer and the ARJA-based code fit well together, i.e. match for such boundary conditions of low ISI. Hence, both curves intersect only at very small SNR values. A different behavior can be found for the convolutional codes, which still intersect for high SNR values.

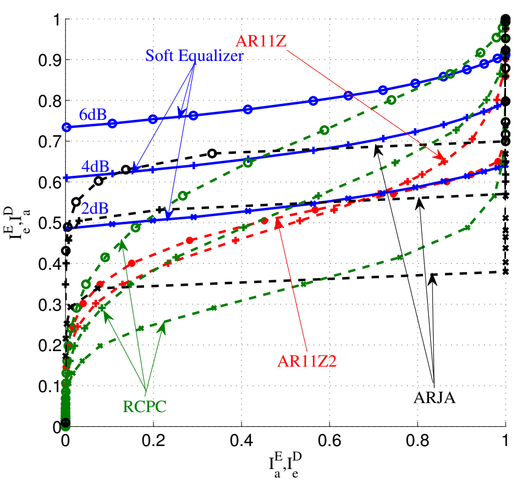


Fig. 4. UWB desktop channel - EXIT-Chart

In the bit-error-rate (BER) simulation we use QPSK modulation. The result is illustrated in Figure 5. As already expected from the EXIT chart analysis the LDPC codes outperform the convolutional codes for all rates. At a BER of 10^{-4} the gain amounts to approximately $2-3$ dB. A small additional gain is possible with longer LDPC code words.

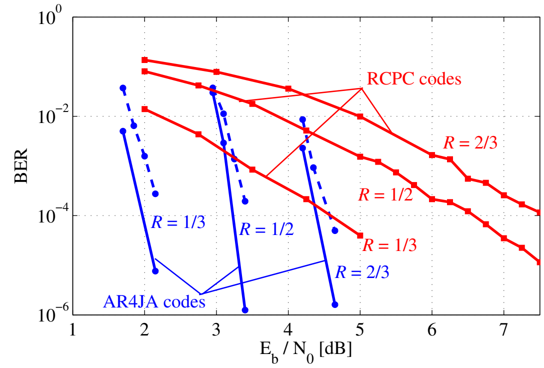


Fig. 5. UWB desktop channel - Bit-Error-Rate Simulation

B. Strong ISI Channels

For the strong ISI channel we again determine the EXIT function for different SNRs. The corresponding EXIT-Chart is illustrated in Figure 6, where we only consider the rate 1/2 code.

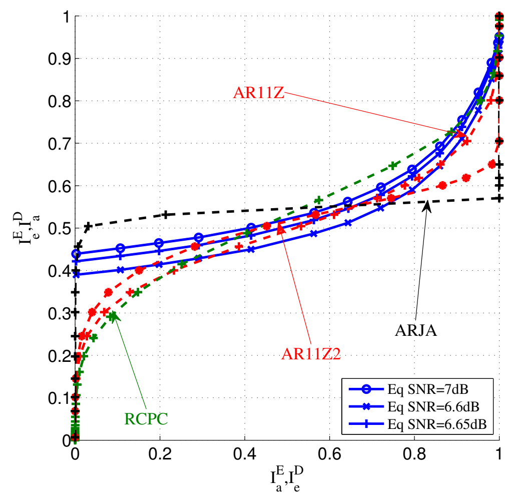
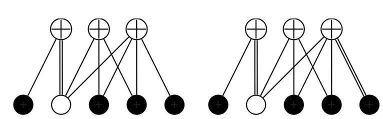


Fig. 6. 10 tap channel - EXIT-Chart

Because the matching of EXIT functions is an indication for the performance of a system, we also expect a smaller gain of the codes for the channel that is impaired by significant ISI. One can see that there is still an intersection for relatively high values of I_a, but in case of the ARJA code also in the intermediate region. Therefore, we slightly changed the protograph (cmp. Figure 7a-7b) by reducing the number of parallel edges to influence the shape of the transfer functions and to obtain a better curve fit, which can be observed in the figure.



(a) AR11Z (b) AR11Z2

Fig. 7. Proposed protographs

In this context, two regions are of special interest for the performance of such an iterative system. The first one is the threshold region at $I_a = 0.5$ and the other one is the error floor region at $I_a \approx 1$. The first region determines the starting point of a strong decay (waterfall) in the BER curve, i.e. the amount of exchanged information is sufficient to eliminate a large fraction of error events due to the good curve fit (no intersection). The match of the EXIT curves in the second region strongly influence the achievable BER, because the values for I_a must be very close to one to obtain very good BERs.

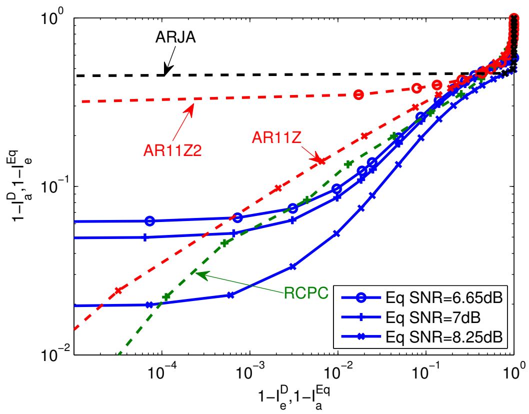


Fig. 8. 10 tap channel - gap-to-one EXIT Chart (loglog-scale)

In Figure 8, one can see the different behavior in more detail. While the transfer functions of the RCPC and the AR11Z code intersect the equalizer EXIT functions, the curves for the ARJA and the AR11Z2 code do not intersect them in that region. Hence, we expect a small decaying BER curve or even an error floor for the first ones. The required SNR and the BER can be estimated by means of the EXIT-Chart analysis.

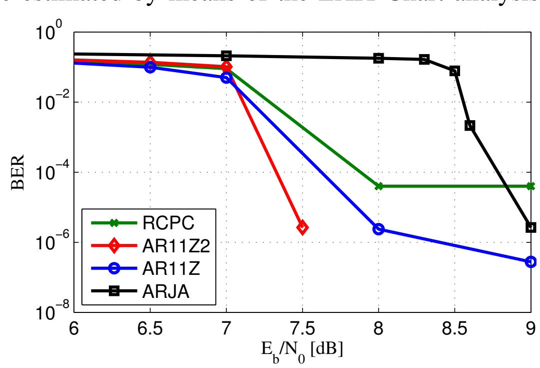


Fig. 9. 10 tap channel - Bit-Error-Rate Simulation

Corresponding simulation results of the BER are illustrated in Figure 9. One can see that, a gain of about 1.5dB can be obtained but not for both protograph codes. The slope of the AR11Z based code starts to decrease at lower E_b/N_0 than the AR11Z2 based code, but is getting flatter at 8dB.

The RCPC codes also show nearly an error floor, due to the intersection of the EXIT function. By appropriate design of the protographs this intersection should be avoidable. It is likely that we can construct codes with better performance, but the proposed ones illustrate the major problems of the code construction for iterative systems.

VI. Summary and Conclusions

We studied the performance of existing AWGN-optimized PG-LDPCCs in terms of their applicability for turbo equalization in UWB-channels. Due to the fact that the performance of the equalizer is strongly influenced by the considered channel, we have investigated a frequency-flat channel at first and extended the study to a channel that leads to strong ISI. In order to analyze and predict the performance we used EXIT charts as analysis and visualization tool of the iterative information exchange and the density evolution as analysis tool for the PG-LDPCCs. By these means, we generalized the analysis so that we did not have to assume Gaussian distributed messages in the decoding process. We have shown that the performance of such optimized codes degrade with increasing ISI. In order to avoid degradation, we proposed two exemplary protographs that have a better curve fit and a better performance. The analysis and simulation results indicate that an interesting problem for future investigations is the trade-off between the minimum achievable signal-to-noise-ratio and minimum achievable bit-error-rate.

References

[1] M. Tüchler, R. Koetter, and A. Singer, "Turbo equalization: principles and new results," *Communications, IEEE Transactions on*, vol. 50, no. 5, pp. 754 –767, may 2002.

[2] M. Franceschini, G. Ferrari, and R. Raheli, "EXIT Chart-based Design of LDPC Codes for Inter-Symbol Interference Channels," in *Proc. IST Mobile Summit Dresden*, 2005.

[3] A. F. dos Santos, "Efficient receiver methods for coded systems under channel uncertainty," Ph.D. dissertation, Dresden University of Technology, 2010.

[4] R. Gallager, *Low-Density Parity-Check Codes*. MIT Press, 1963.

[5] R. Tanner, D. Sridhara, A. Sridharan, T. Fuja, and D. Costello Jr., "LDPC block and convolutional codes based on circulant matrices," *Information Theory, IEEE Transactions on*, vol. 50, no. 12, pp. 2966 – 2984, dec. 2004.

[6] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," JPL, IPN Progress Report 42-154, Aug 2003.

[7] D. Divsalar, C. Jones, S. Dolinar, and J. Thorpe, "Protograph based LDPC codes with minimum distance linearly growing with block size," vol. 3, nov.-2 dec. 2005, p. 5 pp.

[8] X.-Y. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *Information Theory, IEEE Transactions on*, vol. 51, no. 1, pp. 386 –398, jan. 2005.

[9] T. Tian, C. Jones, J. Villasenor, and R. Wesel, "Construction of irregular LDPC codes with low error floors," vol. 5, may 2003, pp. 3125 – 3129 vol.5.

[10] S.-Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, Massachusetts Institute Of Technology, September 2000.

[11] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *Communications, IEEE Transactions on*, vol. 49, no. 10, pp. 1727 –1737, oct 2001.

[12] J. Hagenauer, "The EXIT Chart - Introduction to Extrinsic Information Transfer," in *in Iterative Processing, In Proc. 12th Europ. Signal Proc. Conf (EUSIPCO)*, 2004, pp. 1541–1548.

[13] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *Communications, IEEE Transactions on*, vol. 52, no. 4, pp. 670 – 678, april 2004.

[14] IEEE 802.15 Task Group 3c, "Channel modeling subcommitee final report," IEEE, Tech. Rep., March 2007. [Online]. Available: http://www.ieee802.org/15/pub/TG3c.html

[15] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *Communications, IEEE Transactions on*, vol. 36, no. 4, pp. 389 –400, apr 1988.