

PG-LDPCC Codes in Turbo Equalizer Systems: Trade-off between Design Parameters of the Protograph and the Permutation Size

Patrick Grosa and Gerhard Fettweis

Vodafone Chair Mobile Communications Systems

Dresden University of Technology (TU Dresden), 01062 Dresden, Germany

Emails: {grosa, fettweis}@ifn.et.tu-dresden.de

Abstract—Turbo equalization is a technique to enable wireless transmissions by means of an iterative exchange of information of several components at the receiver incorporating a soft equalizer. The performance achievable can be improved by adjusting the components regarding each other. The focus of this paper lies on the study of the possible adjustments to terminated protograph-based Low-Density Parity Check convolutional codes. Some of the parameters (syndrome former memory m_s , termination length L) are related to the underlying terminated convolutional protograph, while other comes in to play during the actual code construction process like the permutation size P . Compared to the previous work where the design parameter of the protograph itself are studied only, this work will also focus on the final constructed codes and their performance. In order to evaluate the code's performance the length of the code is an important property. Along with the permutation size P and the protograph parameters (m_s , L), it is possible to make a fair comparison between the codes and to decide in favor of the code used to achieve the desired requirements of code length, performance, and code rate.

Index Terms—LDPC Convolutional Codes, Protographs, Parameter Study, Permutation Size, Syndrome Former Memory, Termination Length, Turbo Equalizer, Density Evolution, EXIT-Charts

I. INTRODUCTION

A promising technique to meet the increasing data rate demands of short range communications is the ultra wideband technique. One of the main properties of such systems is the need to utilize a large part of the spectrum available. With the knowledge of the sampling theorem in mind a high temporal resolution is needed to capture the energy of the transmitted signal. Additionally, this high sampling rate helps to distinguish the number of delayed versions of the original signal from multiple paths. In order to overcome the issue of multipath propagation or even exploiting the energy that is received at the receiver from multiple paths the turbo equalization is a suitable approach. The adjustment of the components regarding each other can boost the overall performance of the system, e.g. as it is shown for convolutional codes in [1]. The behavior of improving the overall performance is later also shown for low-density parity-check codes, e.g. in [2], where the first iteration of the channel decoder is considered only. A different approach of using the outage probability is shown in [3]. While the latter studies are based on unstructured LDPC code ensembles the shift to a structured variant is desired

due to implementation benefits. A very common variant is based on the so-called protograph [4] and it is shown that protograph-based low-density parity-check (LDPC) codes that are optimized for the AWGN channel also perform good for channels with a small number of relevant taps and therefore in the presence of low intersymbol-interference (ISI) [5]. Nevertheless, this is not the case when ISI is increasing as shown in [6], but it is also shown that small modifications of the underlying protographs can improve the performance under these circumstances. In recent years an extension of low-density parity-check by a convolutional component, LDPC convolutional (LDPCC) codes [7], has shown that they can reach the channel capacity when the termination length tends to infinity [8]. Since these codes offer more degrees of freedom by introducing several parameters, their influence on the code's behavior in a turbo equalizer needs to be studied. While a previous study in [9] is concentrated on the convolutional protograph parameters (termination length L , syndrome former memory m_s) only, another parameter influences the derived LDPC convolutional code. This parameter is the permutation size P for the construction of the LDPC code from the protograph. Along with the original protograph itself and the termination length L , the permutation size P defines the code length. The trade-off between these parameters is the focus of this work. The remainder of this paper is structured as follows. In the subsequent section II the communication system considered will be explained in detail. Since this work is dealing with LDPCC codes derived from protographs, the basics of these kind of templates for block, convolutional and terminated convolutional (i.e. block codes of particular structure) codes are explained in section III. Following the prediction of the codes' performance based on the protographs via EXIT-Charts in section IV, the performances are evaluated by means of Monte-Carlo-System incorporating the permutation size P . Finally, the work is concluded in section VI.

II. SYSTEM DESCRIPTION

A single carrier transmission with a single antenna on the transmitter and the receiver side is considered in this work. A binary source emits a data vector $\mathbf{d} \in \{0,1\}^{K \times 1}$. The redundancy necessary for the decoding process is added by a channel encoder that produces the codeword vector $\mathbf{v} \in$

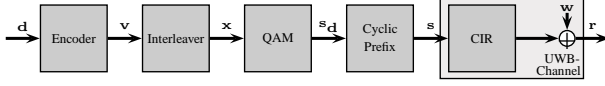


Fig. 1. Transmitter chain

$\{0, 1\}^{N \times 1}$ from data vector \mathbf{d} . The statistical independence of consecutive bits, which is especially necessary for standard convolutional codes, is ensured by means of a subsequent interleaver. The interleaved codeword \mathbf{x} is then mapped to the symbol vector $\mathbf{s}_d = [s_{d_1} \dots s_{d_p} \dots s_{d_P}]^T$ of length $N/Q = S$ in a linear manner, where 2^Q is the size of the modulation alphabet. Although higher modulation schemes are defined within this framework it is sufficient to set $Q = 2$ in order to study the general behavior of the system. After the mapping, a cyclic prefix is added to the symbol vector and the resulting transmission is sent over the channel. Since the equalization is done in the frequency domain, the cyclic prefix is needed to ensure a cyclic convolution with the channel. The transmitter and the channel model is illustrated in Fig. 1.

A block fading multipath channel is assumed with channel impulse response (CIR) given by $\mathbf{h} = [h(0) h(1) \dots h(l) \dots h(T-1)]$. In this context block fading means that the CIR does not change during the transmission of one symbol block. Additionally, white Gaussian noise samples $\mathbf{w} \in \mathbb{C}^N$ with variance σ_w^2 are added to the symbol vector. The received samples $r(i)$ at time i are given as

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

After the distorted symbol vector \mathbf{r} is received on the receiver side (Fig. 2), the signal is equalized based on the channel estimates. There, the frequency domain equalizer proposed in [10] is employed. Then, the soft demapper produces log-likelihood ratios (LLRs) for each bit. The channel estimate can be provided by a channel estimator based on pilot symbols or by the statistical knowledge of the CIR, that is the power delay profile (PDP). The latter one is used throughout this paper. The extrinsic LLRs $L_e^E(\mathbf{x})$ are deinterleaved to recover the original bit order and passed to the channel decoder as a-priori knowledge, denoted by $L_a^D(\mathbf{v})$. The extrinsic output of the decoding process, that are also LLRs and denoted by $L_e^D(\mathbf{v})$, are interleaved in the same way as on the transmitter side as a means to act as a-priori LLRs for the equalizer, indicated as $L_a^E(\mathbf{x})$, and mapped to symbols according to their reliability. The process carries on in an iterative manner which then in turn improves the detection. After a fixed number of iterations are performed the output of the decoder can be used to obtain the estimates of the information word $\hat{\mathbf{d}}$.

III. PROTOGRAPH ENSEMBLES

A. Protograph LDPC codes

Protographs (PG) are bipartite graphs $G(\mathcal{V}_P, \mathcal{C}_P, \mathcal{E}_P)$ that consist of two distinct sets of vertices ($|\mathcal{V}_P| = N_P, |\mathcal{C}_P| =$

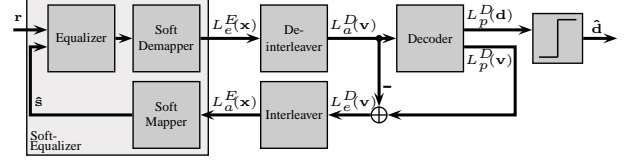
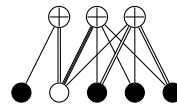


Fig. 2. Receiver chain

M_P) and a set \mathcal{E}_P of edges that connect the two sets. Although PGs are closely related to Tanner-Graphs (TG) [11] and even used as templates for them, PGs do not have any restrictions on the number of parallel edges between the same nodes. Since PGs should simplify the analysis they are much smaller than TGs, but it is possible to derive a TG from a PG by means of a copy-and-permute procedure with permutation matrices of size P . The parameter P plays an important role, since it directly influences the length of the code N by

$$N = P \cdot N_P \quad . \quad (2)$$

The common approach to determine the permutation matrices can be performed by means of the PEG [12] and/or the ACE [13] algorithm. An often used example for protographs due to their easy adaptation to different rates is the so-called Accumulate-Repeat-Jagged-Accumulate (ARJA) protograph (proposed in [14]). A rate- $1/2$ variant is illustrated in Fig. 3a. Circles with a plus represents check nodes and all other nodes are variable nodes. Empty variable nodes represent punctured nodes, i.e. they will not be transmitted over the channel. The number of unpunctured nodes is denoted by N_P^u . Following this analogy, the rate of the unpunctured protograph is defined as $\mathcal{R}_P^u = \left(1 - \frac{M_P}{N_P}\right)$ and the rate with puncturing can be calculated by $\mathcal{R}_P = \mathcal{R}_P^u \left(\frac{N_P}{N_P^u}\right)$.



(a) protograph

$$\mathbf{B} = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 \\ 0 & 3 & 1 & 1 & 1 \\ 0 & 1 & 2 & 1 & 2 \end{pmatrix}$$

(b) base matrix

Fig. 3. Equivalent representations of $R = 1/2$ -ARJA protograph

An equivalent description of a protograph can be done in matrix form, where every row represent a check node and every column a variable node. This matrix, called base matrix, \mathbf{B} is an extended bi-adjacency matrix, where elements are integer values and can be greater than one in order to reflect parallel edges. The equivalent base matrix of the previous example is illustrated in Fig. 3b.

B. Protograph based LDPC convolutional codes

The ability to extend the codes from block codes to a convolutional variant [7] is also available for protograph-based LDPC codes. Although the explanation can also be done by means of the graph it is easier to understand the procedure with the help of the base matrix \mathbf{B} with dimensions

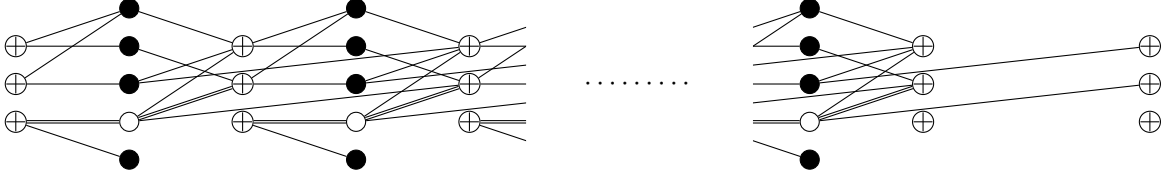


Fig. 4. Unwrapped convolutional protograph with $m_s = 2$ and without termination ($L \rightarrow \infty$)

$M_P \times N_P$. This matrix is spread over $m_s + 1$ matrices having the same dimensions, where m_s is called the syndrome former memory. Each matrix \mathbf{B}_i is called a *partial base matrix* and the superposition of all matrices spread yields to the base matrix, again. A decomposition of the example base matrix in partial base matrices for $m_s = 1$ can be found in 3.

$$\mathbf{B} = \sum_{i=0}^{m_s=1} \mathbf{B}_i = \underbrace{\begin{pmatrix} 1 & 2 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 2 \end{pmatrix}}_{\mathbf{B}_0} + \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}}_{\mathbf{B}_1} \quad (3)$$

This partial base matrices can be arranged in an infinite band matrix in the following manner:

$$\mathbf{B}_{[-\infty, \infty]} = \begin{bmatrix} \ddots & & & & \\ & \mathbf{B}_{m_s} & \cdots & \mathbf{B}_0 & \\ & & \ddots & & \\ & & & \mathbf{B}_{m_s} & \cdots & \mathbf{B}_0 \\ & & & & \ddots & \\ & & & & & \ddots \end{bmatrix} \quad (4)$$

The resulting graph is called a *convolutional protograph*. As long as an infinite structure is assumed, the number of edges emanating the variable and check nodes remains the same and in the same way the rate. However, this work focuses on the terminated version due to their better threshold behavior as it is shown in [8]. Therefore, suppose the convolutional code is started at time $t = 0$ and terminated after L time instants. The equivalent graph of this matrix is called the *convolutional protograph of termination length L* .

$$\mathbf{B}_{[0, L-1]} = \begin{bmatrix} \mathbf{B}_0 & & \\ \vdots & \ddots & \\ \mathbf{B}_{m_s} & \cdots & \mathbf{B}_0 \\ & \ddots & \vdots \\ & & \mathbf{B}_{m_s} \end{bmatrix}_{(L+m_s)M_P \times LN_P} \quad (5)$$

A limitation to a minimum number of edges emanating from a check node is necessary in order to avoid unreasonable protograph configurations. In particular, this limitation is important to be implemented for both ends of the protograph. In case of termination of the infinite convolutional code, the L -terminated code has the code rate \mathcal{R}_{CC} , that also depends on the syndrome former memory m_s .

$$\mathcal{R}_{CC} = 1 - \left(\frac{L + m_s}{L} \right) \frac{c}{b} = 1 - \left(\frac{L + m_s}{L} \right) (1 - \mathcal{R}_P) \quad (6)$$

Since it is important for the further evaluation of the codes proposed it is necessary to mention that the length of derived codes can be calculated by

$$N = N_P \cdot L \cdot P \quad (7)$$

IV. EXIT-CHART ANALYSIS

In the following section the basic results of [9] are recapitulated. There the permutation size P is fixed. A first estimate of the performance and the behavior of codes derived from convolutional protographs is carried out by means of Extrinsic-Information-Transfer (EXIT) charts [15]. Hereby, two different approaches to determine the transfer characteristics are used. The first one uses the general procedure of measuring the a-priori and extrinsic information during Monte-Carlo simulations. The second approach obtains these measures by the density evolution approach [16] modified for protographs, i.e. a certain probability density function (pdf) of the L-value for each particular node type is assumed and the evolution of this pdf during a decoding process is tracked. Thereby, the averaged pdf before and after the decoding process are used to determine the a-priori and extrinsic information values.

In Fig. 5 the EXIT-Chart for the studied cases is illustrated. Since the transfer-function of the soft equalizer is directly influenced by the channel it is necessary to mention that the channel assumed has an exponentially decaying power delay profile (PDP) given by

$$h(i) = \exp\left(-\frac{i+1}{10}\right), \quad i = 0, \dots, 9 \quad (8)$$

Since the energy of the signal of the last taps are not negligible, the signal experiences a significant intersymbol interference. The question that arises is that, how a channel code must be designed to work well in this environment. For that reason, the transfer functions are shown for several convolutional versions of the protograph, where the design parameters L and m_s are varied. Additionally, the transfer function for the block code of the ARJA protograph illustrated. In general, it is shown that the convolutional variants beat their block code variant and has a lower threshold. Moreover, it can be seen that an increase of the memory increases the extrinsic information for low a-priori information values. The opposite behavior can be observed for the termination length, i.e. increasing L results in a decrease in the extrinsic information produced. The first parameter influences the number of low degree check nodes at both sides of the convolutional protograph. On the other hand, the low degree check nodes in the convolutional protograph are pushed away with the increase of L . However, increasing

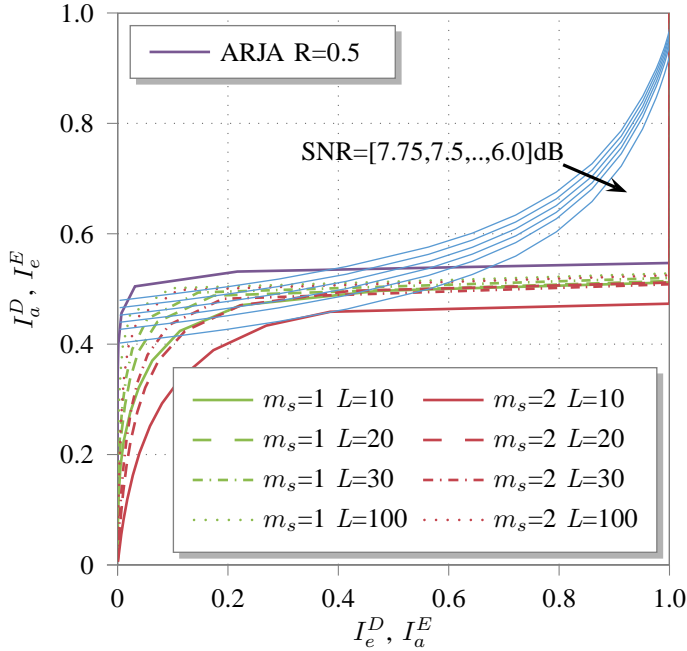


Fig. 5. EXIT-Chart: Behavior estimate for codes based on convolutional protographs for several L s and $m_s = 1, 2$

m_s leads to a rate loss as indicated in (6). This rate loss can be decreased by choosing a larger L , but it leads to a longer code when we assume P to remain constant.

V. SIMULATION AND DISCUSSION

Based on the analysis of the convolutional protographs in the previous section an estimate of the codes' performance can be done. However, one of the main drawbacks of the density evolution is that the density evolution assumes an infinite block length. In lights of that the second requirement to meet the performance predicted by the EXIT-Chart analysis is $P \rightarrow \infty$. Since this is not a reasonable option for real world communications systems, the question that arises is what is a permutation size P sufficient to reach the predicted performance. Consequently, another question needs to be answered. Since the permutation size P and the termination length L directly influences the length of the terminated convolutional code, which one of these parameter should be preferred to be adjusted, if a certain block length constraint should be met. The answers to these questions can only be determined by Monte-Carlo simulations. In general, the number of outer iterations, i.e. between soft equalizer and decoder is limited to a reasonable number of six iterations, since no further improvements can be seen in the author's simulations. In case of decoding an LDPC code there are 100 inner loop iterations, i.e. between check and variable node decoder, for the first five iterations of the outer loop and a number of 200 iterations in the last iteration.

In order to benchmark the codes proposed, the performance of two widely known competitors are also used for the comparison. The first one is the rate $1/2$ -variant of the rate-

compatible-punctured-convolutional codes introduced in [17]. The second code is based on the Accumulate-Repeat-Jagged-Accumulate protograph [18], which is known to have a good performance in non-fading channels and often used as starting point for studies on protograph-based LDPC (convolutional) codes. The termination length and permutation size varies between $L = 10, 20, 30$ and $P = 100, 200, 300$, respectively in the simulations. The choice of these parameters makes it possible to compare codes of the same block length, e.g. codes with $L = 10, P = 300$ (code 1) and $L = 30, P = 100$ (code 2) have the same block length. Another pair with the same block length has $L = 30, P = 200$ (code 3) and $L = 20, P = 300$ (code 4). There are several other combinations possible for comparing codes of the same block length, which are not mentioned explicitly.

In Fig. 6 the results of the Monte-Carlo simulations for PG-LDPC codes with $m_s = 1$ are shown. At the first glance it can be seen that the block code and the standard convolutional code are beaten by all convolutional variants. While a higher E_b/N_0 is needed with the increase of L for the code to work, the behavior in the low BER regime is exactly the opposite. It is also visible that the performance improves with the size of the permutation matrix P in the low BER regime, while the point of decaying remains the same for each ensemble with the same L . When the performance of the previously mentioned examples of the same block length is compared it can be seen that the performance of code 1 is about $0.3dB$ better for a $BER = 10^{-6}$ than the performance of code 2. However it must be kept in mind that $L = 10$ for code 1, which means it has a lower code rate. A different behavior can be observed for the second pair, where code 3 is slightly better than code 4, although the latter one has also a better rate, i.e. is not possible to make a general statement on this issue.

In general, the same behavior can be observed in Fig. 7 for a syndrome former memory of $m_s = 2$. This shows that the behavior seems to be independent of the syndrome former memory. However, it can be shown that the intersection point, that separates the high and the low BER regime behavior, is shifted to a lower BER rate. It depends on the application if this regime is still of interest, otherwise the behavior above the intersection is the more dominant one and should be considered.

In order to compare both figures it is shown that the increase of the syndrome former memory leads to performance gain for low SNR values. However, this leads to a reduced code rate (6), which can be increased by L but leads to a longer code (for the same P) (7). In order to maintain the same block length the permutation size P may can be reduced without a loss of performance depending on which side of the intersection point the communication system should work.

VI. CONCLUSIONS

It is shown that a general code search via EXIT-charts leads to reasonable performance gains in fading channels. However, for a final decision on the deployed code the trade-off between performance (dominated by m_s and P), code rate (mostly

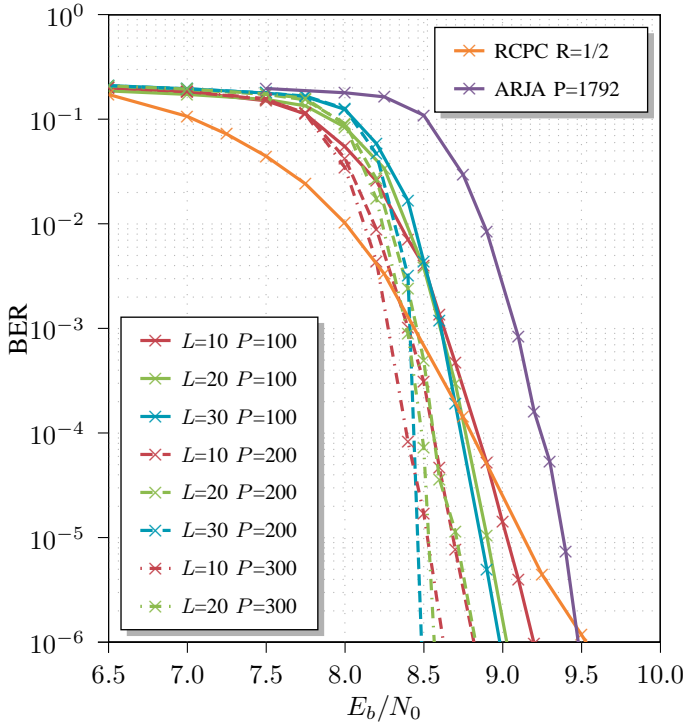


Fig. 6. Bit error rate simulation of ARJA-based terminated convolutional codes with $m_s = 1$ and their competitors

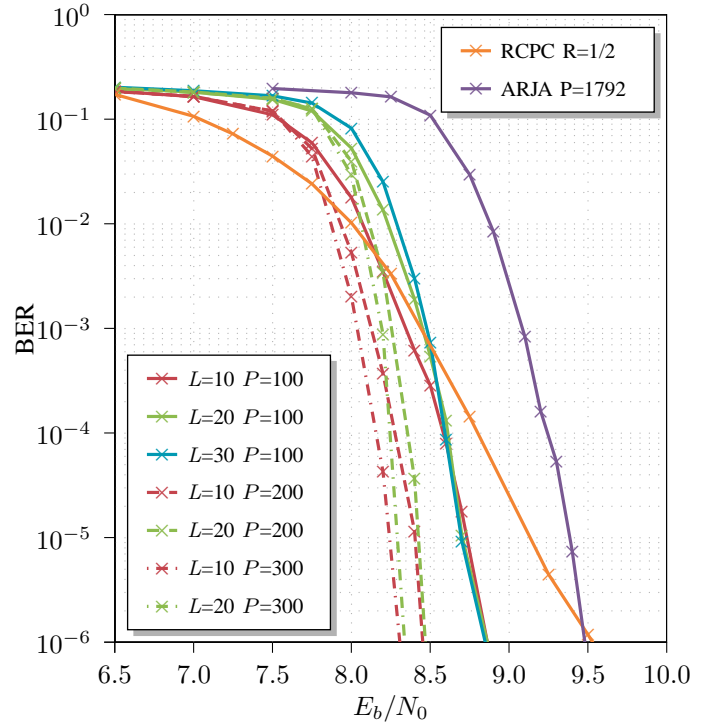


Fig. 7. Bit error rate simulation of ARJA-based terminated convolutional codes with $m_s = 2$ and their competitors

dominated by m_s and L) and code length (dominated by P and L) needs to be observed and the parameter must be adjusted in order to meet the requirements of the system. In some cases even two parameters can be changed in a positive way without impairing the third parameter.

ACKNOWLEDGEMENT

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) within the UKoLoS framework under grant SPP 1202/3.

REFERENCES

- [1] M. Tüchler and R. Otnes, "Exit Chart Analysis applied to adaptive Turbo Equalization," in *Nordic Signal Processing Symposium*, 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*, 2005.
- [3] R. Wohlgenannt, K. Kansanen, D. Tujkovic, T. Matsumoto, A. Austriamicrosystems, and A. Premstaetten, "Outage-based LDPC code design for SC/MMSE turbo-equalization," in *2005 IEEE 61st Vehicular Technology Conference, 2005. VTC 2005-Spring*, vol. 1, no. c, 2005.
- [4] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *IPN Progress Report 42-154, JPL*, pp. 1–7, 2005.
- [5] W. Rave, A. Fonseca dos Santos, P. Grosa, M. Lentmaier, and G. Fettweis, "Iterative Equalization and Decoding for Short Range Communication at 60 GHz," *Frequenz*, vol. 63, no. 9-10, pp. 192–195.
- [6] P. Grosa, A. F. dos Santos, M. Lentmaier, W. Rave, and G. Fettweis, "Application of protograph-based LDPC codes for UWB short range communication," in *2010 IEEE International Conference on Ultra-Wideband*. IEEE, Sep. 2010, pp. 1–4.
- [7] A. Jimenez and K. Zigangirov, "Periodic time-varying convolutional codes with low-density parity-check matrices," in *Proceedings. 1998 IEEE International Symposium on Information Theory (Cat. No.98CH36252)*. IEEE, p. 305.
- [8] M. Lentmaier and G. P. Fettweis, "On the thresholds of generalized LDPC convolutional codes based on protographs," in *2010 IEEE International Symposium on Information Theory*. IEEE, Jun. 2010, pp. 709–713.
- [9] P. Grosa and G. P. Fettweis, "PG-LDPC Codes in Turbo Equalizer Systems - A Study on the Influence of Design Parameters of Terminated Convolutional Protographs," in *9th International ITG Conference on Systems, Communications and Coding*, 2013.
- [10] K. Kansanen, "Wireless broadband single-carrier systems with MMSE turbo equalization receivers," Ph.D. thesis, University of Oulu, 2005.
- [11] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, Sep. 1981.
- [12] E. Eleftheriou and D. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.
- [13] T. Tian, C. Jones, J. Villasenor, and R. Wesel, "Construction of irregular LDPC codes with low error floors," in *IEEE International Conference on Communications, 2003. ICC'03*, vol. 5, 2003, pp. 2–6.
- [14] D. Divsalar, S. Dolinar, and C. Jones, "Low-rate LDPC codes with simple protograph structure," in *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, 2005, pp. 1622–1626.
- [15] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, 2001.
- [16] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.
- [17] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 389–400, 1988.
- [18] D. Divsalar, C. Jones, S. Dolinar, and J. Thorpe, "Protograph based LDPC codes with minimum distance linearly growing with block size," in *IEEE Global Telecommunications Conference, 2005. GLOBECOM'05*, vol. 3, 2005, pp. 1152–1156.