Piggybacking an Additional Lonely Bit on Linearly Coded Payload Data

Erik G. Larsson and Reza Moosavi

Original Publication:

Erik G. Larsson and Reza Moosavi, Piggybacking an Additional Lonely Bit on Linearly Coded Payload Data, 2012, IEEE Wireless Communications Letters, (1), 4, 292-295.
http://dx.doi.org/10.1109/WCL.2012.041612.120093

Piggybacking an Additional Lonely Bit on Linearly Coded Payload Data

Erik G. Larsson and Reza Moosavi

***Abstract*—We provide a coding scheme, by which an additional lonely bit can be piggybacked on a payload data packet encoded with a linear channel code, at no essential extra cost in power or bandwidth. The underlying principle is to use the additional bit to select which of two linear codes that should be used for encoding the payload packet, this way effectively creating a nonlinear code. We give a fast algorithm for detecting the additional bit, without decoding the data packet. Applications include control signaling, for example, transmission of ACK/NACK bits.**

I. Introduction

In many applications, coded blocks of payload data are multiplexed with an occasional additional "lonely" bit. An example of this situation is the transmission of HARQ ACK/NACK bits in wireless systems such as LTE [1]. Typically, this extra bit, called the "additional lonely bit" (ALB) herein, is not jointly encoded with the payload data because it is desirable to detect the ALB separately, without decoding the whole payload block. This in turn is useful if one wants to know the value of the ALB without waiting for the channel decoder to finish. In some cases, the contents of the payload data may not even be of interest to the receiver.

The simplest way of conveying the ALB is to encode it with a repetition code and transmit it on resources that are orthogonal to those used by the payload data. If N is the number of real-valued (one-dimensional) channel uses nominally allocated for the transmission of the payload data and K is the number of times the ALB is repeated, then repetition coding of the ALB requires the payload data to be squeezed into $N-K$ channel uses as compared to the situation where an ALB is not sent. As shown in the appendix, maintaining the same nominal spectral efficiency β (bits/real dimension) for the payload then requires an increase in transmit power of about $6.0 \cdot \frac{\beta}{1-2^{-2\beta}} \cdot \frac{K}{N}$ dB. This cost can be substantial. If, for example, $N=200$, $\beta=0.5$ and $K=5$, then the power increase is in the order of 0.15 dB. By contrast, if the ALB were jointly coded with the payload data, the extra power cost would be negligible. But this requires a code that separates the original payload packet from the ALB, so that the ALB can be detected without decoding the payload data.

A. Contribution and related work

We provide a coding scheme (Sec. II) and an associated fast decoder (Sec. III), by which an ALB can be piggybacked on a payload data packet encoded with a linear channel code, at no essential cost in power or bandwidth, and detected with very high reliability at the receiver *without decoding the data packet*. The underlying idea is to use the ALB to select which of two given linear codes that should be used for encoding the payload packet, this way effectively creating a nonlinear code. Fundamentally, the distinction of our scheme is that basing the nonlinear construction on two linear codes enables us to use an efficient detector. In practice, a simple (but not the only) way to choose the two linear codes that works in many cases is to let the second code be a permuted version of the first. The proposed scheme is primarily useful for relatively small N, say in the order of one hundred. While the complexity of the proposed detector for the ALB is not high, the detection does not come entirely for free. If the ALB is a very important bit, such as an ACK/NACK bit in an HARQ scheme, then this complexity should be justifiable.

Overall this work is motivated by the cost of physical-layer control signaling in mobile broadband systems. This cost can be substantial [2]–[4], but appears to often be overlooked in the literature. We are not aware of any work that has studied the problem of conveying an ALB together with linearly coded payload data specifically. However, the proposed coding scheme is related to the secondary synchronization signaling (SSS) procedure used in 4G/LTE [1, p. 301]. SSS signals are constructed from m-sequences which are permuted in two different ways, depending on in which subframe they are sent. The SSS sequences carry payload data consisting of the cell identity group number and this information is encoded by choosing a specific m-sequence. They also carry an ALB (in our terminology) via the two possible ways the permutation is performed; this bit of information tells the receiver about one of two possible subframe locations.

The similarity between SSS in LTE and our proposed scheme is somewhat superficial, because the set of SSS sequences do not span a linear code (since m-sequences do not include the all-zero codeword). While the SSS procedure uses an ALB to switch between two *nonlinear* codes, we use it to switch between two *linear* codes that could be any standard channel codes. Our fast detector (see Sec. III) is designed to detect whether an observed signal was encoded with a given *linear* code or not. Since the SSS signals are not based on a linear code our detector cannot be used to detect them, and in fact as SSS signals are selected from a small set, a much simpler detector would suffice.

II. Proposed Coding Scheme for Conveying an ALB

Here we describe the coding scheme used to convey the ALB. The associated fast detector is given in Sec. III. For

The authors are with the Dept. of Electrical Engineering (ISY), Linköping University, Linköping, Sweden. Email:{egl,reza}@isy.liu.se. This work was supported in part by Ericsson, VINNOVA and ELLIIT. E. Larsson is a Royal Swedish Academy of Sciences (KVA) Research Fellow supported by a grant from the Knut and Alice Wallenberg Foundation.

simplicity of the discussion we assume BPSK modulation per real dimension so that the number of channel uses N equals the number of coded bits. The extension to general modulation offers no fundamental difficulties but requires more notation.

The underlying principle of the scheme is to augment the original payload data codebook with another "sufficiently different" codebook (to be made precise in the next paragraph) and use the ALB to select which code that should be used. More precisely, let $\mathcal{M}$ be the payload message and let b be the ALB. Let $\mathcal{C}_0$ and $\mathcal{C}_1$ be two different channel codes. The transmitter now operates by encoding $\mathcal{M}$ with $\mathcal{C}_0$ if $b = 0$, and conversely, encoding $\mathcal{M}$ with $\mathcal{C}_1$ if $b = 1$. If the original codebook has $2^{N\beta}$ codewords, the new codebook created this way has $2 \times 2^{N\beta} = 2^{N\beta+1}$ codewords. Provided that $\mathcal{C}_0$ and $\mathcal{C}_1$ are appropriately chosen, the structure of the codebook $\mathcal{C}_0 \cup \mathcal{C}_1$ facilitates detection of the ALB *without decoding the payload bits*, and does not complicate the decoding of the payload data.

Ideally, $\mathcal{C}_0$ and $\mathcal{C}_1$ would just be two randomly chosen good codebooks of length N and with $2^{N\beta}$ codewords. In practice, $\mathcal{C}_0$ and $\mathcal{C}_1$ could be two random instances of an LDPC code with a given degree distribution. Alternatively, as a simple and well-performing alternative, they can be obtained by permuting (interleaving) the bits of a standard channel code (convolutional, turbo, LDPC) in two different ways. For example, one can take $\mathcal{C}_0$ to be a randomly generated LDPC code and $\mathcal{C}_1$ to be the same code but with the bits rearranged so that if $\{\boldsymbol{c}_1, ..., \boldsymbol{c}_N\}$ are the codewords of $\mathcal{C}_0$ then the codewords of $\mathcal{C}_1$ are given by $\{\boldsymbol{c}_{i_1}, ..., \boldsymbol{c}_{i_N}\}$, where $\{i_1, ..., i_N\}$ is a randomly chosen permutation. For implementation reasons, a structured permutation $\{i_1, ..., i_N\}$ may be desirable; for example, for non-cyclic codes one could perform "half-swapping" by setting $\{i_1, ..., i_{N/2}\} = \{N/2 + 1, ..., N\}$ and $\{i_{N/2+1}, ..., i_N\} = \{1, ..., N/2\}$. (Let N be even here for simplicity.) Judging from our numerical experiments, the specific way the permutation is performed does not appear to be important.

In general, there is a tradeoff in that the longer codes are used (larger N), the more reliable the detection of the ALB will be. This is so because the minimum distance between the codebooks $\mathcal{C}_0$ and $\mathcal{C}_1$ increases with N, for fixed β. On the other hand, the larger N, the relatively smaller is the advantage of transmitting the ALB jointly with the payload. We stress that the construction proposed here does not require $\mathcal{C}_0$ or $\mathcal{C}_1$ to be linear, but the detector (Sec. III) does.

III. Fast Algorithm for Detection of the ALB

The detection of b amounts to determining which code, $\mathcal{C}_0$ or $\mathcal{C}_1$, was used to encode $\mathcal{M}$. More specifically, we are interested in the posterior probability that *given the received data*, denoted $\mathcal{R}$ here, the code $\mathcal{C}_b$ was used for the encoding. We conclude that code $\mathcal{C}_b$ was used for encoding the data if all syndrome checks associated with $\mathcal{C}_b$ are satisfied. Hence, given $\mathcal{R}$, and given a hypothetical code $\mathcal{C}_b$ (corresponding to a hypothetical value of b) we want to evaluate the probability that all syndrome checks of $\mathcal{C}_b$ are satisfied. This probability is given by

$$\begin{aligned} &P(b|\mathcal{R}) \triangleq P(\text{all syndrome checks for } \mathcal{C}_b \text{ satisfied}|\mathcal{R}) \\ &= P\left(\bigcup_k \bigoplus_l c_{p_{bkl}} = 0 \middle| \mathcal{R}\right) \approx \prod_k P\left(\bigoplus_l c_{p_{bkl}} = 0 \middle| \mathcal{R}\right) \end{aligned} \quad (1)$$

where c_i is the ith coded bit, and p_{bkl} is the index of the lth nonzero element of the kth row of the parity check matrix associated with the code $\mathcal{C}_b$.[1] In (1), we assumed in the last step that the syndrome checks are independent in the sense that the two events $\bigoplus_l c_{p_{bkl}} = 0$ and $\bigoplus_l c_{p_{bk'l}} = 0$ are independent for $k \neq k'$, given $\mathcal{R}$. This independence assumption should be justifiable for large N, but it is not crucial for the detector to work in practice.

Let

$$\ell_i = \log\left(\frac{P(c_i = 0|\mathcal{R})}{P(c_i = 1|\mathcal{R})}\right)$$

for $i = 1, ..., N$ be the log-likelihood ratios (LLRs) of the bits c_i obtained at the output of the channel demodulator. Then the LLR associated with the probability that the kth syndrome check of the code $\mathcal{C}_b$ is satisfied, is

$$\gamma_k^b \triangleq \log\left(\frac{P\left(\bigoplus_l c_{p_{bkl}} = 0 \middle| \mathcal{R}\right)}{1 - P\left(\bigoplus_l c_{p_{bkl}} = 0 \middle| \mathcal{R}\right)}\right) = \boxplus_k \ell_{p_{bkl}} \quad (2)$$

where $\boxplus$ is the standard "Boxplus" operator [5].[2] Using (2) in (1) and taking the logarithm yields

$$\log(P(b|\mathcal{R})) = \sum_k \log\left(\frac{e^{\gamma_k^b}}{1 + e^{\gamma_k^b}}\right) = -\sum_k \log(1 + e^{-\gamma_k^b}) \quad (3)$$

To determine which code, $\mathcal{C}_0$ or $\mathcal{C}_1$ that was used by the transmitter (and hence the value of b), we form the test

$$\begin{aligned} \log\left(\frac{P(b=0|\mathcal{R})}{P(b=1|\mathcal{R})}\right) =& \sum_k \log(1 + e^{-\gamma_k^1}) \\ &- \sum_k \log(1 + e^{-\gamma_k^0}) \underset{b=1}{\overset{b=0}{\gtrless}} 0. \end{aligned} \quad (4)$$

The left hand side of (4) may be used as a soft decision on b. Also, the threshold value "0" on the right hand side of (4) could be taken to be different from zero, should a biased test (i.e., a test that is more likely to produce a "1 → 0 error" than a "0 → 1 error", or vice versa) be desired for some reason.

To our knowledge the concept of an "all syndromes satisfied" posterior probability as defined by the syndrome posterior probability (1), and the associated independence approximation leading to (4), is an original contribution. A preliminary version of the soft all-syndromes-satisfied posterior

[1] $\oplus$ denotes addition over GF(2), i.e. XOR.

[2] More precisely,

$$\boxplus_{i=1}^{n} \ell_i \triangleq \log\left(\frac{1 + \prod_{i=1}^{n} \tanh(\ell_i/2)}{1 - \prod_{i=1}^{n} \tanh(\ell_i/2)}\right).$$

probability concept appeared in [6], in the context of blindly identifying channel codes for adaptive modulation and coding. The test in [6] was based on a heuristic argument and led to a different test that underperforms the test properly derived in Eq. (4) here. There is also some relation to methods for estimation of parameters of convolutional codes [7], but it does not appear that the parameter estimates derived therein could be usefully exploited for the detection task at hand.

Double-decoder benchmark scheme for detecting b:

An alternative scheme for detecting the ALB is to run the decoders of both $\mathcal{C}_0$ and $\mathcal{C}_1$. If the decoder of $\mathcal{C}_0$ converges to a valid codeword but that of $\mathcal{C}_1$ does not then take the ALB to be 0, and vice versa. If none, or both of the decoders converge, use the above fast ALB detection algorithm. This *double decoder* scheme requires decoding of the payload data to determine b (avoiding this was the main objective of proposing the above fast detection algorithm) and its computational complexity is twice that of the original system without an ALB. However, this scheme has better error performance than the fast algorithm above (see Sec. VI) and hence it serves as a useful benchmark.

IV. Decoding the Payload Data

If b were known, we would decode $\mathcal{M}$ as in the conventional system without the ALB. By contrast, if b is completely unknown, we could run both the decoder for $\mathcal{C}_0$ and that for $\mathcal{C}_1$, and pick the output of the decoder that returns a valid codeword. This would work because it is very unlikely that both the $\mathcal{C}_0$ and $\mathcal{C}_1$ decoders will return a valid codeword. A standard CRC check could always be used to guarantee that an invalid codeword is practically never returned.

In our case, a good estimate of b, say $\hat{b}$, is available by the algorithm provided above. To decode the payload data, we first try decoding $\mathcal{M}$ with the decoder for $\mathcal{C}_{\hat{b}}$. If this decoder returns a valid codeword, we take its output as the final result, otherwise, we try with the other decoder.

V. Complexity Analysis

a) Encoder: Two encoders must be implemented in the transmitter. However, the computational complexity of performing the encoding is unchanged as compared to transmission without an ALB. If $\mathcal{C}_1$ is chosen as a permuted version of $\mathcal{C}_0$, then no extra encoder but one extra interleaver must be implemented.

b) ALB Detection with the Fast Algorithm: The total number of terms in (4) is equal to the sum of the number of parity checks for each codeword. For an $M \times N$ LDPC code, this is equal to the number of parity check nodes $M = N(1-r)$, where r is the code rate. Computing these M terms requires $\sum_{m=1}^{M}(d_m - 1)$ boxplus operations, where d_m is the degree of the mth check node.

For an (n, k) convolutional code, the syndrome checks can be obtained from the syndrome former of the code. A syndrome former is an $n \times (n-k)$ matrix consisting of polynomials with possibly different degrees [8, p. 91]. The maximum polynomial degree, denoted by m_s, is called the memory of the syndrome former and determines how any length-n bit string obtained at the output of the encoder shift registers at time instance i, is related to the previous m_s such strings. Each column of the syndrome former imposes one syndrome check for each string. This means that for a block consisting of N coded bits, there are in total $N(n-k)/n$ syndrome checks that need to be satisfied. Note that these syndrome checks are not independent in general, due to the memory of the code. However, by skipping every other m_s length n-string, we can obtain approximately $\frac{N(n-k)}{nm_s}$ independent syndrome checks. Computing each syndrome check requires at most $nm_s - 1$ boxplus operations.

The terms of the form $\log(1 + e^{-x})$ in (4) are numerically very well-behaved and can be easily implemented via a table-lookup for $x \geq 0$. If $x < 0$, we can write $\log(1 + e^{-x}) = -x + \log(1 + e^{x})$ and then use the same table-lookup for the last term, where now $x \leq 0$. Very low precision fixed point arithmetic can be used to represent x. On a final note, the decoding delay for the ALB with the proposed scheme will be N, whereas with repetition coding it is only K.

c) Payload Decoding: The computational complexity of payload decoding is essentially equal to the decoding complexity of the original system without an ALB. This is so because we normally only need to run one of the two decoders corresponding to $\mathcal{C}_0$ or $\mathcal{C}_1$, depending on the value of b delivered by the ALB detector. Note that for LDPC codes, the quantities $\gamma_k^{\hat{b}}$ computed by the algorithm in Sec. III will be used as intermediate quantities in the decoder. Hence, half of the boxplus computation results obtained in the ALB detection algorithm can be reused in the decoder.

If b is incorrectly detected, both decoders will have to be run. In the typical operating regime, the error rate of the ALB, $P_e(\text{ALB})$, would be 10^{-2} or smaller. Hence, the chance that both decoders have to be run is negligible. The complexity increase is $P_e(\text{ALB})$ times the complexity of decoding $\mathcal{M}$ with the wrong code. For a code with a fixed-complexity decoder, e.g. a convolutional code, the overall complexity increase is then about $P_e(\text{ALB})$, i.e., typically less than 1%. For an LDPC code, the total complexity increase depends on the number of allowed iterations in the decoder, and since the decoder must iterate more times when trying with the wrong code, the complexity increase will be somewhat higher.

VI. Numerical Example

We provide two examples using BPSK transmission with spectral efficiencies $\beta = 1/2$ and $\beta = 1/6$, over an AWGN channel. The specific choice of code is unimportant so to illustrate the principles of operation we chose, somewhat arbitrarily, a randomly generated regular rate-1/2 LDPC code with degree distribution (3,6) to achieve $\beta = 1/2$ bpcu, and a regular rate-1/6 LDPC code with degree distribution (5,6) to achieve $\beta = 1/6$. LDPC codes were used for the ease of simulation and since they offer a natural possibility of error checking. The codes were optimized by performing some cycle removal, and the resulting parity check matrices had no cycles of length 4. The payload blocks had $N = 200$ coded bits resulting in 100 and 33 information bits for the two cases

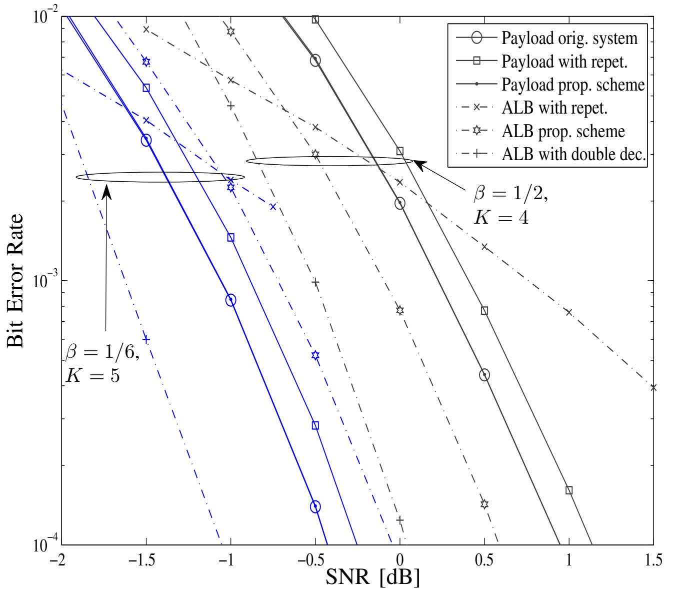

Fig. 1. BER performance at nominal spectral efficiencies $\beta = 1/2$ and $\beta = 1/6$. In the comparisons with repetition coding, $K = 4$ respectively $K = 5$ repetitions for the ALB were used. The curves for repetition coding are truncated to improve readability of the plot.

respectively. The LDPC decoder was forced to terminate after at most 10 iterations. The half-swap permutation discussed in Sec. II was used to obtain $\mathcal{C}_1$ from $\mathcal{C}_0$.

Fig. 1 shows the simulation results for spectral efficiency $\beta = 1/2$ bpcu with the number of repetitions $K = 4$, and for $\beta = 1/6$ bpcu with $K = 5$, respectively. The following graphs are shown for each case: (i) the payload data BER of the original system, without transmitting an ALB, (ii) the payload data BER with repetition coding (using $N - K$ out of N channel uses; hence K of the LLRs are set to zero before decoding), (iii) the payload BER of a system using the proposed scheme, (iv) the error probability the of ALB with the proposed scheme, (v) the error probability of the ALB using repetition coding, and (vi) the error probability of the ALB when using the double decoder benchmark scheme.

We selected K so that the ALB error probability crossing point between the proposed scheme and repetition coding occurred at slightly below $P_e \approx 10^{-2}$. At SNR above the crossing point, the proposed scheme offers better protection of the ALB than what repetition coding does and vice versa. Observe that for SNR above the crossing point, no substantial payload BER penalty of transmitting an ALB could be measured for the proposed scheme. The power penalty Δ when using repetition coding is about 0.18 and 0.19 dB, respectively, for $\beta = 1/2$ and $1/6$. (The order-of-magnitude estimate in the Appendix gives 0.12 dB, for both cases.)

C++ code for reproducing the numerical results is available from the authors.

VII. Conclusions

The proposed scheme facilitates detection of the ALB without decoding the payload. We believe that the additional complexity of this can be justified if the ALB is important (e.g., an ACK/NACK bit in an HARQ scheme). For some channel codes, parts of the operations of the ALB detection algorithm can be reused in the decoding of the payload data.

The scheme presented here can be extended in several ways. For example, if the amount of error protection on the ALB offered by the proposed scheme is insufficient, the ALB can be transmitted several times, piggybacked onto several different payload packets. Alternatively, the other way around, more than one ALB, say L bits, could be transmitted on top of one single payload codeword by choosing among 2^L codes $\mathcal{C}_1, ..., \mathcal{C}_{2^L}$ at the transmitter. This is likely to be useful only for small values of L, as the associated detection complexity scales exponentially with L. Also, naturally, when L is increased, the payload BER would eventually be compromised.

Appendix

The AWGN channel with noise variance $N_0/2$ and transmit power P per real dimension supports a spectral efficiency of

$$\beta \triangleq \frac{1}{2}\log_2\left(1 + \frac{P}{N_0/2}\right) \tag{5}$$

bits/real channel use (bpcu). If N channel uses are nominally allocated for the transmission and the ALB is to be repeated K times, then only $N - K$ channel uses remain for the transmission of the payload data. Hence the spectral efficiency of the payload data transmission must be increased by a factor $N/(N-K)$, compared to the case when no ALB is present. This requires an increase in transmit power from P to P', where P' satisfies

$$\beta = \frac{1}{2}\left(1 - \frac{K}{N}\right)\log_2\left(1 + \frac{P'}{N_0/2}\right). \tag{6}$$

Let $\Delta \triangleq P'/P$ be the increase in transmit power needed to maintain the same spectral efficiency as in the original system, when the ALB consumes payload resources. Solving (5)–(6) for Δ in dB, we obtain to the first order in $1/N$:

$$10\log_{10}(\Delta) = 10\log_{10}\left(\frac{2^{\frac{2\beta}{1-K/N}} - 1}{2^{2\beta} - 1}\right) \approx 6.0 \cdot \frac{\beta}{1 - 2^{-2\beta}} \cdot \frac{K}{N}$$

References

[1] E. Dahlman, S. Parkvall and J. Sköld, *4G LTE/LTE-Advanced for Mobile Broadband*, 1st edition Academic Press, 2011.

[2] J. Gross, H. F. Geerdes, H. Karl and A. Wolisz, "Performance analysis of dynamic OFDMA systems with inband signaling," *IEEE J. Select. Areas Commun.*, vol. 24, pp. 427-436, Mar. 2006.

[3] M. Sternad, T. Svensson and M. Döttling, "Resource allocation and control signaling in the WINNER flexible MAC concept," *in Proc. of IEEE VTC*, pp. 1-5, Sep. 2008.

[4] R. Moosavi, J. Eriksson, E. G. Larsson, N. Wiberg, P. Frenger and F. Gunnarsson, "Comparison of strategies for signaling of scheduling assignments in wireless OFDMA," *IEEE Trans. Veh. Technol.*, vol. 59, pp. 4527-4542, Nov. 2010.

[5] J. Hagenauer, E. Offer and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Info. Theory*, vol. 42, pp. 429-445, Mar. 1996.

[6] R. Moosavi and E. G. Larsson, "A fast scheme for blind identification of channel codes," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, Dec. 2011.

[7] J. Dingel and J. Hagenauer, "Parameter estimation of a convolutional encoder from noisy observations," *IEEE International Symposium on Information Theory (ISIT)*, Jun. 2007.

[8] R. Johannesson and K. Sh. Zigangirov, *Fundamentals of Convolutional Coding*, IEEE Press, 1999.