Communication system with SISO channel decoding using bit stuffing

Nataša Živić
Institute for Data Communications Systems
University of Siegen
Siegen,Germany
Natasa.Zivic@uni-siegen.de

***Abstract* - This paper introduces a communication system with typically wireless transmission, with SISO convolutional decoding using predefined reliability values at input of the SISO decoder. "Dummy" bits are stuffed into the information which is to be transmitted over a noisy channel and SISO decoded. Stuffed bits have to be known at the receiver, so that their reliability values can be defined at the input of the SISO decoder, before the information is to be decoded. The results of simulations have been presented and discussed, showing how different number of stuffed bits has to be chosen depending on the length of information.**

Keywords-SISO channel decodig; bit stuffing; L-values; feedback; pucturing; segmentation

I INTRODUTION

This paper defines the L-values (reliability values or soft values) of SISO (Soft Input Soft Output) convolutional channel decoding [1] [2] [3] [4] for the improved correction of information which has to be transmitted over a noisy channel. Therefore, stuffed bits with predefined L-values are generated and added to information bits. Noisy channels are very often wireless channels in today's data communications.

It is known that L-values are used in turbo decoding [5] [6] which is nowadays the only decoding approaching the Shannon limit very closely. Turbo codes are standard for mobile communications and use the concept of SISO decoding, with preferred algorithms of Maximum A Posteriori (MAP) [7] and Soft Output Viterbi Algorithm (SOVA) [8]. MAP algorithm is used in this paper.

In this paper L-values are feedback information from the correct decoded bits to the input of the SISO decoder for the improvement of the next decoding step [9], but using stuffed bits. Stuffed bits are used as correct decoded bits and their L-values are set at input of the SISO convolutional decoder. In this way, an outer code, whose errorless output would be used for the feedback, has been avoided. [9] uses cryptographic mechanisms as an outer code, because they can guarantee with the high probability the errorless output of the inner code. The reason for avoiding the outer code in this paper is that there are not many coding schemes in the praxis, which would guarantee with the high probability the errorless output. On the one hand, CRC mechanisms are used as an outer code which gives the information if inner decoding is correct, but depending on the CRC length is the probability of the false information (collisions) higher. On the other hand, if only convolutional coding and no concatenated codes are used, the method presented in this paper enables the increase of the coding gain.

Including stuffed bits in a communication scheme decreases the code rate. Therefore, puncturing is used in this paper to enable a transfer of information without degradation of the coding rate, i.e. the number of punctured bits is the same as the number of stuffed bits. Puncturing takes place before transmission over the channel and depuncturing after channel transmission. The price of using puncturing is a decrease of the coding gain. Nevertheless, the resulting coding gain using puncturing shows that the method presented in this paper can be used for improvement of SISO decoding results without code rate degradation.

The organization of the paper is following: the feedback algorithm in combination with Soft Input Decryption, which is basic for this paper, is explained in Section 2. Algorithm of a feedback using bit stuffing is given in Section 3. Results of computer simulations of the algorithm presented in Section 3 are given in Section 4. As the coding gain depends on the ratio of the number of stuffed and information bits, the influence of the length ratio to the coding gain is examined in Section 5. Section 6 introduces puncturing into the system. The conclusion of the paper and suggestions for the future work are given in Section 7.

II FEEDBACK ALGORITHM COMBINED WITH SOFT INPUT DECRYPTION

The algorithm of feedback was presented in combination with Soft Input Decryption [10] for improvement of channel decoding of cryptographic information. The algorithm is presented in [9] and shown in Fig. 2. Feedback algorithm functions as follows:

The source encoder outputs a data block or data stream *v* which has to be authentic (realized by use of cryptographic check values). *v* is split in two parts, called message *ma* and message *mb*. Messages *ma* and *mb* are extended by a cryptographic check value generated using a cryptographic check function (see Fig. 1):

 ISBN: 978-1-61208-171-7

$$a = a_1 a_2 ... a_{m_1+n_1} = ma_1 ma_2 ... ma_{m_1} na_1 na_2 ... na_{n_1} \quad (1)$$

$$b = b_1 b_2 ... b_{m_2+n_2} = mb_1 mb_2 ... mb_{m_2} nb_1 nb_2 ... nb_{n_2} \quad (2)$$

m_1 and m_2 are lengths of messages *a* and *b* respectively and n_1 and n_2 are lengths of cryptographic check functions *na* and *nb* respectively. For each message with the cryptographic check value Soft Input Decryption algorithm is applied.

We will assume that $m_2 \geq m_1$ and $n_1 \geq n_2$. We will further assume that. block *a* and block *b* form the joint message *u* (assuming that $(m_2 + n_2) \bmod (m_1 + n_1) = 0$, for simplicity):

$$u = \begin{cases} a_1 b_1 a_2 b_2 ... a_{m_1+n_1} b_{m_2+n_2}, \; if \; m_1 + n_1 = m_2 + n_2 \\ a_1 b_1 ... b_{\frac{m_2+n_2}{m_1+n_1}} a_2 ... a_{m_1+n_1} b_{m_2+n_2-\frac{m_2+n_2}{m_1+n_1}+1} ... b_{m_2+n_2} \\ \qquad if \; m_1 + n_1 < m_2 + n_2 \end{cases} \quad (3)$$

u is encoded, modulated and transferred over an AWGN channel.

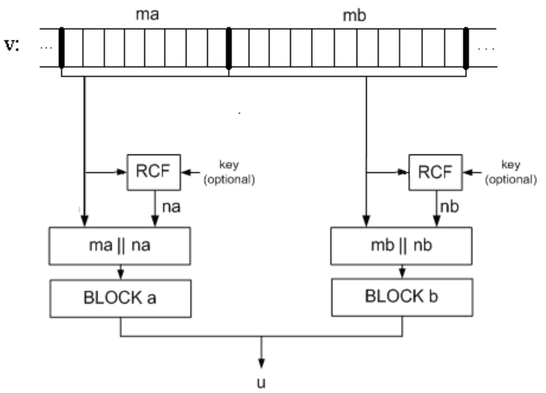

Figure 1. Formatting the message *u*

The feedback method has 2 steps, as presented in Fig. 2:

In the step 1 of the feedback method the output c' of the line decoder is decoded into u', the output of the channel decoder u' is segmented into block a' and block b', and then block a' is corrected by Soft Input Decryption, using the L-values of a'. If Soft Input Decryption is successful, each bit of block a' is known, the L-values of a' bits are set to ± ∞ (+ ∞ accords decoded "0" and - ∞ decoded "1", because BPSK modulation with bit mapping "0"-> "1" and "1"-> "-1" is used) and "fed back" to the next decoding step. The L-values of block b' are set to 0.

In the second step of feedback method c' is decoded again, but now with different L-values. The resulted BER is lower than after the first decoding step: bits of block a are already corrected and the bits of block b' have lower BER compared to the case that the bits of block a are unknown.

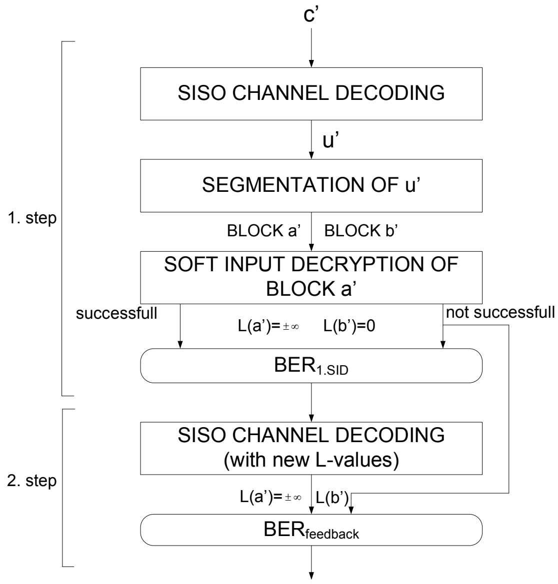

Figure 2. Feedback algorithm in combination with Soft Input Decryption [9]

III BIT STUFFING

In this paper the data block *b* is considered and. the block *a*, which enables the usage of feedback, is missing: stuffed bits will be used as bits of block *a* and will be exploited for the feedback to block *b* - the information which has to be corrected.

A random generator produces stuffed bits at the transmitter, which are sent over a separate channel to the receiver, so that the receiver knows them before SISO decoding starts. As stuffed bits are known at the receiver side, their L-values can be set to ± ∞ depending on bit values 0 or 1 (if BPSK modulation is used, for example), and used like in a feedback method. As no previously correctly decoded information is "fed back" in a form of set L-values, we are talking about SISO channel decoding using stuffed bits and no more about a feedback method.

Stuffed bits form a block *a* in Fig. 3, and information bits form a block *b*.

Block *a* with a length *la* and block *b* with a length *lb* form the message *v* by interleaving (Fig. 3):

$$v = \begin{cases} a_1 b_1 a_2 b_2 ... a_{la} b_{lb}, \; if \; la = lb \\ a_1 b_1 ... b_{\frac{lb}{la}} a_2 ... a_{la} b_{lb-\frac{lb}{la}+1} b_{lb}, \; if \; la < lb \end{cases} \quad (4)$$

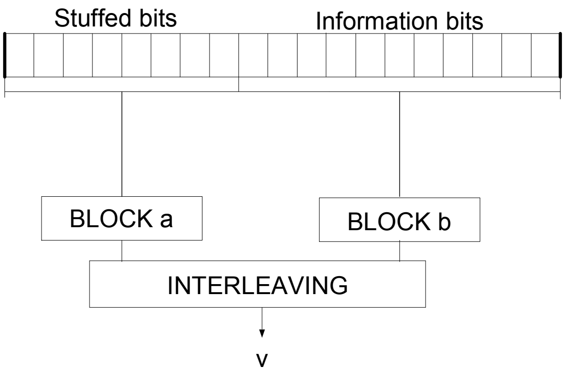

Figure 3. Forming of a message *v*

We will consider the case of *la* = *lb* for the further simplicity, without the loss of generality. *v* is encoded with a convolutional encoder, modulated and transferred over an AWGN channel.

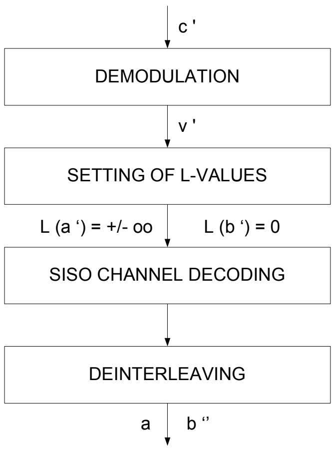

Figure 4. SISO decoding using stuffed bits

By using the feedback method explained in [9], the output *c'* of the demodulator is decoded to *v'* (Fig. 4). Afterwards *v'* is deinterleaved into block *a'* and block *b'*. Each bit of block *a'* (stuffed bits) is known to the receiver and, therefore, the L-values of *a'* bits are set to ± ∞ (+ ∞ means decoded "0" and - ∞ decoded "1", if BPSK modulation with bit mapping "0"-> "1" and "1"-> "-1" is used) as input to the SISO decoder. The L-values of block *b'* are set to 0, as there is no knowledge about their values.

c' is SISO channel decoded, using predefined L-values. The decoding results will be shown in Section 4: the bits of block *b"* will have lower BER compared to the case when stuffed bits are not used.

IV SIMULATION RESULTS

Simulations of SISO channel decoding using bit stuffing are performed using the following parameters:

- length of *a* block: 160 bits; length of *b* block: 480 bits
- convolutional 1/2 encoder (5,7)
- BPSK modulation
- AWGN channel
- SISO channel decoder using MAP algorithm [7]
- random generated stuffed bits.

For each point of the resulting curves (Fig. 5) 50.000 simulations in C/C++ have been realized.

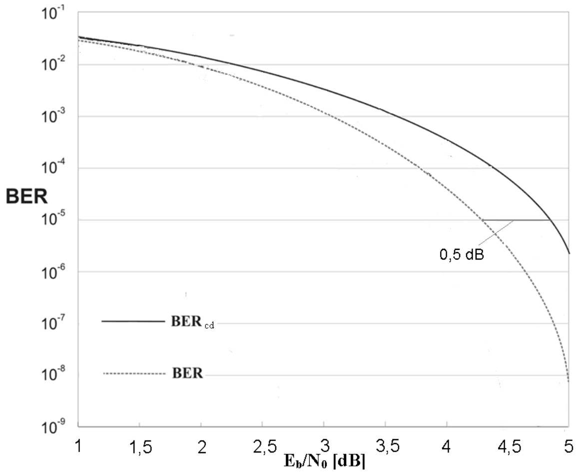

Figure 5. Coding gain of the method using stuffed bits

The results of simulations are presented in Fig. 5. They show the coding gain of the method using stuffed bits in comparison to the SISO convolutional decoding without stuffed bits. The coding gain is significant for the whole range of E_b/N0. For higher values of E_b/N_0, coding gain achieves 0.5 dB (for BER of 10^{-5}).

Obviously, the usage of stuffed bits increases the probability of finding the right decoding solution of a Trellis, depending on a number of stuffed bits. The cost of SISO channel decoding improvement is the decrease of the code rate: in this case, the overall code rate is:

$$\frac{480}{160+480}\frac{1}{2}=\frac{3}{8} \tag{5}$$

instead of a ½ code rate in case that no stuffed bits are used.

V SIMULATION OF THE RATIO OF THE NUMBER OF STUFFED AND INFORMATION

Obviously, the coding gain of SISO channel decoding using stuffed bits depends on the ratio of the number of stuffed and information bits, i.e. lengths of blocks a and b (Fig. 3). Therefore, simulations with different information lengths, whereby the number of stuffed bits remains the same, are performed using the following parameters:

- convolutional 1/2 encoder (5,7)
- BPSK modulation
- AWGN channel and
- SISO channel decoder using MAP algorithm.

The used lengths of blocks *a* and *b* and their length ratio are given in Table 1:

TABLE I. LENGTH RATIOS AND BLOCK LENGTHS

BER	Length ratio	Length of block *a*	Length of block *b*
$BER_{1\text{-}1}$	1 : 1	160	160
$BER_{1\text{-}2}$	1 : 2	160	320
$BER_{1\text{-}3}$	1 : 3	160	480

For each point of curves 50 000 simulations have been performed. The results of simulation with length ratios as in Table 1 are presented in Fig. 6 in comparison to 1/2 convolutional coding using the same SISO channel decoder (BER_{cd} is BER of channel decoding). Stuffed bits have different influence on decoding results, depending on the length of block *b*: the best decoding results are in 1-1 case of the length ratio, and the worst in 1-3 case of the length ratio.

The best decoding results are obtained in case 1-1, because the number of stuffed bits is the same as the number of information bits, so that the stuffed bits “help” finding correct paths in the Trellis diagram by every second known path. Vice versa, the worst decoding results, which are very close to the results of standard 1/2 convolutional decoding (BER_{cd}), are achieved, as expected, in 1-3 case. The reason for such decoding results is that every forth path of the Trellis diagram is known, so that more decoding solutions are available than in other cases in Table 1. For that reason it happens oft that wrong decoding solutions are chosen by decoding algorithm.

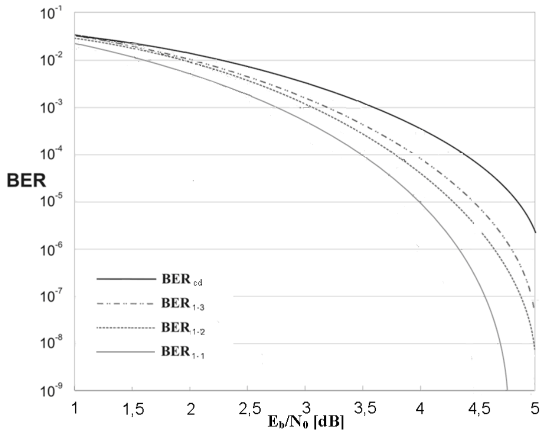

Figure 6. Bit Error Rate for different length ratios

VI SIMUALTION OF STUFFED BITS WITH PUNCTURING

Puncturing /depuncturing is introduced (Fig. 9) to override a problem of a code rate reduction caused by the usage of stuffed bits. If puncturing rate is chosen in such a way, that the number of punctured bits equals the number of stuffed bits, a code rate remains the same as of a used convolutional encoder.

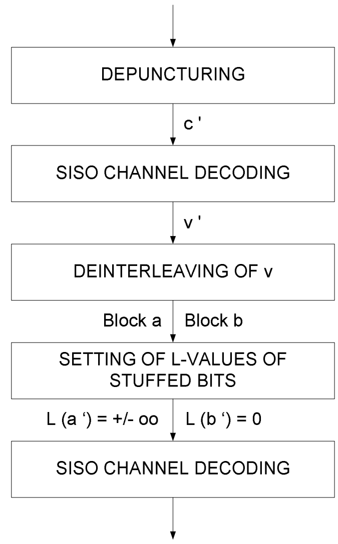

Figure 7. Stuffed bits with puncturing

The simulation of stuffed bits with puncturing are performed using the same parameters as in Section 4. As block *a* is three times shorter than block *b*, every fourth bit of encoded information *c* has to be punctured. In this way, the overall code rate remains ½.

$$R = \frac{480}{160 + 480 - 160} \cdot \frac{1}{2} = \frac{1}{2} \qquad (6)$$

The results of simulations are presented in Fig. 8.

After puncturing, the coding gain of convolutional decoding using stuffed bits is 0.25 dB for BER of 10^{-4} and 0.28 dB for BER of 10^{-3}. Although coding gain using stuffed bits with puncturing is lower than coding gain in Fig. 3 without puncturing, a fair comparison of decoding results is realized.

 ISBN: 978-1-61208-171-7

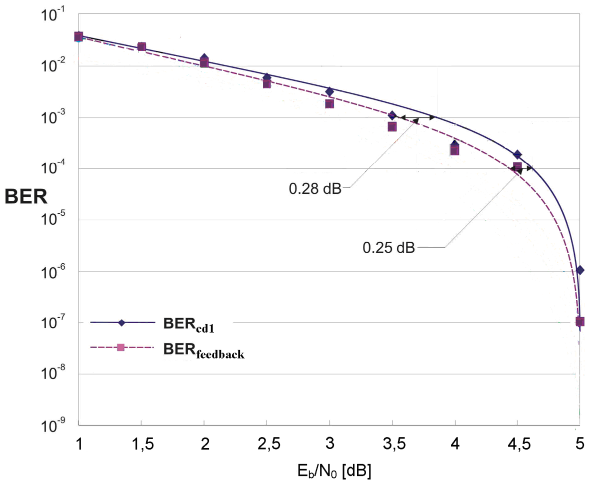

Figure 8. BER after the first and second step using stuffed bits with puncturing

By using algorithm shown in Fig. 7, the coding gain of convolutional MAP decoding increases introducing no additional costs: date rate on the transmission channel remains the same and there are no additional receiver elements. The processing time of the receiver negligible only increases because of the additional de-interleaving.

VII CONCLUSION AND FUTURE WORK

The presented paper introduces the usage of stuffed bits in combination with the feedback algorithm. Stuffed bits are exploited here for the improvement of SISO channel decoding using MAP. The stuffed bits interleaved with information bits are known to the receiver, and therefore their *L*-values are set to $\pm \propto$ before SISO decoding starts. Better decoding results are accomplished, as paths through the Trellis diagram are partially determined. The simulation results show the coding gain of up to 2 dB depending on the number of used stuffed bits.

The usage of stuffed bits decreases the overall code rate. If more stuffed bits are added, the resulting coding gain is bigger and the code rate is lower. Therefore, puncturing / depuncturing of bits is involved: if the number of punctured bits equals the number of stuffed bits, the code rate remains the same as in the used convolutional encoder. In case that the number of information bits is three times bigger than the number of stuffed bits, simulations show coding gain of up to 0.28 dB. The presented algorithm can be used for single antenna systems, as well as for MIMO systems.

The future work has to examine the influence of values of stuffed bits to decoding results. For the future work, new simulation with specific values of stuffed bits have to be performed. Depending on information bits and channel characteristics, the specific values of stuffed bits could improve the coding gain.

Other aspect of future work should include the influnce of channel coding, by new simualtion using different channel encoders, and modulation. The influence of stuffed bits to the execution time of the presented method should be also examinated.

REFERENCES

[1] H. D. Kammeyer: Nachrichtenuebertragung, *B.G. Teubner*, Reihe Informationstechnik, 3., Stuttgart 2004, ISBN 3-519-26142-1

[2] S. Lin, D. J. Costello: Error Control Coding, Pearson Prentice Hall, USA, 2004

[3] M. Bossert: Kanalcodierung, B. G. Treubner, Stuttgart, 1998

[4] G. Kabatiansky, E. Krouk, S. Semenov: Error Correcting Coding and Security for Data Networks, Analysis of the Superchannel Concept, John Wily and Sons, Ltd 2005

[5] C. Berrou, A. Glavieux, P. Thitimajshima: Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes, Proc. IEEE International Conference on Communication, Geneva, Switzerland, vol. 2/3, pp. 1064-1070, 1993

[6] S. A. Barbulescu: What a wonderful turbo world, ISBN 0-9580520-0-X, Adelaide (2002)

[7] J. Hagenauer, P. Hoeher: A Viterbi algorithm with soft-decision outputs and its applications, Proc. IEEE GLOBECOM `89, Dallas, Texas, USA, vol. 3, pp. 1680-1686, November 1989

[8] L. Bahl, J. Cocke, J. Jelinek, J. Raviv: Optimal decoding of linear codes for minimizing symbol error rate, IEEE Transactions on Information actical Design Rules, IEEE Transactions on Information Theory, IT-45, pp. 1361-1391, July 1999

[9] C. Ruland, N. Zivic: Feedback in Joint Channel Coding and Cryptography, 7th Source and Channel Code Conference, VDE/IEEE, Ulm, January 2008

[10] C. Ruland, N. Zivic: Soft Input Decryption, 4th Turbo-code Conference, 6th Source and Channel Code Conference, VDE/IEEE, Munich, April 2006

