Aalborg Universitet

On the interpretation of the APP algorithm as an LLR filter

Land, Ingmar Rüdiger; Hoeher, Peter; Sorger, Ulrich

Published in:
Proc. Int. Symp. Inform. Theory (ISIT)

Publication date:
2000

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (APA):
Land, I., Hoeher, P., & Sorger, U. (2000). On the interpretation of the APP algorithm as an LLR filter. In Proc. Int. Symp. Inform. Theory (ISIT) (pp. 415)

ISIT2000, Sorrento, Italy, June 25–30, 2000

On the Interpretation of the APP Algorithm as an LLR Filter

Ingmar Land and Peter Hoeher
Information and Coding Theory Lab
University of Kiel
Kaiserstr. 2, D-24143 Kiel, Germany
e-mail: `{il,ph}@techfak.uni-kiel.de`

Ulrich Sorger
Institute for Communications Technology
Darmstadt University of Technology
Merckstr. 25, D-64283 Darmstadt, Germany
e-mail: `uli@nesi.tu-darmstadt.de`

Abstract — **A channel decoder employing the a posteriori probability (APP) algorithm can be formulated so that its inputs and its outputs are log-likelihood-ratios (LLR): channel LLRs of the code bits are accepted, and a posteriori LLRs of the info bits and/or the code bits are delivered. Since decoding improves the reliability, the APP algorithm can be interpreted as a non-linear filter for LLRs. The "LLR amplification" depends on the distance properties of the channel code; for high signal-to-noise ratios it is dominated by the minimum distance.**

SUMMARY

The APP algorithm [1] accepts a priori probabilities and channel probabilities as inputs and delivers a posteriori probabilities as outputs. With additional computation of soft outputs for the code bits [2][3] and with usage of LLRs instead of probabilities [4], it can be extended to the logarithmic APP (LogAPP).

Consider a binary linear convolutional encoder of rate $R = k/n$. Let $\boldsymbol{e}$ denote the path through the trellis associated with the info word $\boldsymbol{u}(\boldsymbol{e})$ and the code word $\boldsymbol{x}(\boldsymbol{e})$, $u, x \in \{+1, -1\}$. The code bits are transmitted over a memoryless channel; the received value of a single bit is denoted by y, and the received word is denoted by $\boldsymbol{y}$.

The LogAPP algorithm takes the a priori LLRs of the info bits U and the channel LLRs of the code bits X,

$$L^-(U) \triangleq \ln \frac{P(U=+1)}{P(U=-1)},\ L^-(X) \triangleq \ln \frac{P(X=+1|y)}{P(X=-1|y)}, \quad (1)$$

and computes the a posteriori LLRs of the info bits and of the code bits

$$L^+(U) \triangleq \ln \frac{P(U=+1|\boldsymbol{y})}{P(U=-1|\boldsymbol{y})},\ L^+(X) \triangleq \ln \frac{P(X=+1|\boldsymbol{y})}{P(X=-1|\boldsymbol{y})}. \quad (2)$$

These inputs and outputs of the LogAPP algorithm are depicted in Fig. 1. In the following, the info bits are assumed to be equally distributed, i.e. $L^-(U) = 0$.

Fig. 1: The input and the output LLRs of the LogAPP algorithm.

The purpose of decoding is to improve the reliability of the bits. This motivates to interpret decoding as non-linear filtering, as mentioned in [2]. In this paper, the LogAPP is treated as a *non-linear LLR filter*. This point-of-view suggests to define an *info bit LLR amplification* (ILA) and a *code bit LLR amplification* (CLA):

$$\text{ILA} \triangleq \left. \frac{\mathrm{E}_{\boldsymbol{y}}\ L^+(U)}{\mathrm{E}_{\boldsymbol{y}}\ L^-(X)} \right|_{L^-(U)},\ \text{CLA} \triangleq \left. \frac{\mathrm{E}_{\boldsymbol{y}}\ L^+(X)}{\mathrm{E}_{\boldsymbol{y}}\ L^-(X)} \right|_{L^-(U)}, \quad (3)$$

where $\mathrm{E}_{\boldsymbol{y}}$ denotes the expected value with respect to $\boldsymbol{y}$. The ILA can be regarded as the transfer function of a *soft-decoder*; since there are less output values than input values, the soft-decoder is similar to a decimator. The CLA can be regarded as the transfer function of a *soft-repeater*, i.e. a device which performs decoding and re-encoding using soft values.

For rate 1/2 convolutional codes with memories 2 to 8, binary transmission over an AWGN channel was simulated. In Fig. 2, the ILA and the CLA are depicted as a function of the mean channel LLR $\mathrm{E}_{\boldsymbol{y}}\ L^-(X)$ of the code bits. The following characteristics can be justified analytically:

1. For low input LLRs, the ILA approaches 0 and the CLA approaches 1.
2. For high input LLRs, both the ILA and the CLA approach a constant value which can be identified with the free distance of the code.

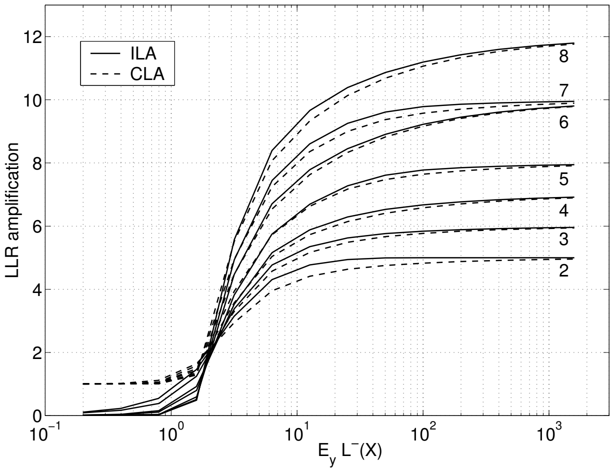

Fig. 2: The LLR amplifications of the convolutional codes with memories 2 to 8.

REFERENCES

[1] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, pp. 284–287, March 1974.

[2] J. Lodge, R. Young, P. Hoeher, and J. Hagenauer, "Separable MAP "filters" for the decoding of product and concatenated codes," in *IEEE Int. Conf. Commun.*, May 1993, pp. 1740 – 1745.

[3] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Commun. Letters*, vol. 1, no. 1, pp. 22 – 24, January 1997.

[4] Patrick Robertson, Peter Hoeher, and Emmanuelle Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *Europ. Trans. Telecommun.*, vol. 8, no. 2, pp. 119 – 125, March - April 1997.