
© 2000 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
On the Interpretation of the APP Algorithm as an LLR Filter

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

Ulrich Sogger
Institute for Communications Technology
Darmstadt University of Technology
Merckstr. 25, D-64283 Darmstadt, Germany
e-mail: ulisogger@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 e path through the trellis associated with the info word u(e) and the code word x(e). u, x ∈ {+1, −1}.

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

\[ L^-(U) \triangleq \ln \frac{P(U = +1)}{P(U = -1)} \]
\[ L^-(X) \triangleq \ln \frac{P(X = +1|y)}{P(X = -1|y)} \] (1)

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

\[ L^+(U) \triangleq \ln \frac{P(U = +1|y)}{P(U = -1|y)} \]
\[ L^+(X) \triangleq \ln \frac{P(X = +1|y)}{P(X = -1|y)} \] (2)

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

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):

\[ \text{ILA} \triangleq \frac{E_y \left| L^+(U) \right|_{L^-(U)}}{E_y \left| L^-(X) \right|_{L^-(X)}} \], \quad \text{CLA} \triangleq \frac{E_y \left| L^+(X) \right|_{L^-(X)}}{E_y \left| L^-(X) \right|_{L^-(U)}} \] (3)

where \( E_y \) denotes the expected value with respect to \( y \). 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-repeat, 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 \( E_y \left| L^-(X) \right| \) 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