SOFT-OUTPUT DETECTOR FOR CHANNELS WITH INTERSYMBOL INTERFERENCE AND MARKOV NOISE MEMORY

Aleksandar Kavčić
Division of Engineering and Applied Sciences
Harvard University, Cambridge, MA 02138

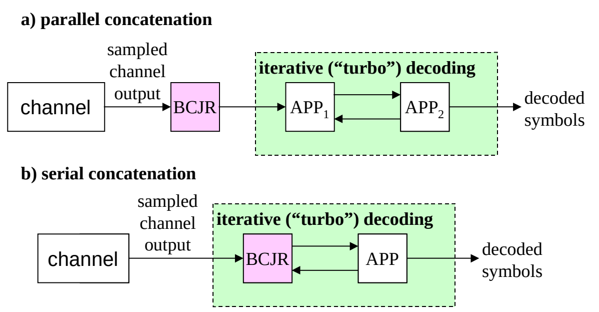

Fig. 1. Turbo-decoding strategies applied to channels with intersymbol interference and correlated noise. The blocks labeled BCJR are the soft channel *detectors*, while the blocks labeled APP are the error correcting *decoders*.

Abstract The paper presents the forward-backward soft-output detector for channels with intersymbol interference and correlated (and/or signal-dependent) channel noise. We utilize the Markov noise assumption to keep the detector complexity in check. Particular attention is given to channels with Gauss-Markov noise statistics. We demonstrate that the detector can be built with the same hardware used for a maximum likelihood sequence detector (Viterbi detector), i.e., using a bank of finite impulse response (FIR) filters. Possible applications of the detector are in iterative decoding applications (e.g., turbo coding).

1. Introduction

The near-capacity performance of iterative decoding strategies (turbo decoding [1]) in additive white Gaussian noise channels has recently sparked interest in iterative decoding applied to channels with intersymbol interference and correlated signal-dependent noise. Parallelly concatenated schemes [2] are typically more complex to implement than serially concatenated decoders [3]–[5] that treat the partial response channel as a convolutional decoder. Both these strategies are depicted in Figure 1. In Figure 1, the block labeled BCJR is a forward-backward soft-output MAP detector [6]. The BCJR detector (or possibly another trellis-based soft-output detector [7]), being the point of contact between the channel and the decoder, needs to be tuned to the signal and noise statistics of the channel. In many instances, particularly in magnetic recording channels, the channel is nonlinear and the noise is correlated and/or signal-dependent [8]. This paper develops a forward-backward soft-output MAP detector (BCJR detector) for such channels.

In Section 2, we present the modified BCJR detector for channels with intersymbol interference and signal-dependent Markov noise. The detector is defined on a state trellis similar to the Viterbi detector for such channels [9]. We emphasize the Markov noise assumption in the solution of this problem, since it is precisely this assumption that lets us formulate both the Viterbi detector and the BCJR detector with finite complexities. The derivation of relationships that define the detector are presented in the Appendix A.

In Section 3, we briefly expose the structure of the detector when the signal-dependent and correlated Markov noise has Gaussian statistics, i.e., when the noise is Gauss-Markov. We show that the detector uses the same branch metrics as the Viterbi detector for Markov noise memory channels [9], and can therefore be implemented with the same hardware, i.e., with banks of FIR filters.

Notation Column vectors are denoted by underlined characters, and the superscript $^{\mathrm{T}}$ is used to denote vector transposition. If z_k is a discrete-time indexed sequence where k denotes the time, then the column vector of sequence samples at time k_1 through $k_2 \geq k_1$ is denoted by $\underline{z}_{k_2}^{k_1} = [z_{k_1}, z_{k_1+1}, \cdots, z_{k_2}]^{\mathrm{T}}$. The probability of an event A is denoted by $\mathrm{P}(A)$. The probability density function (pdf) of a random vector $\underline{z}$ is denoted by $f(\underline{z})$. The notations $\mathrm{P}(A|B)$ and $f(\underline{z}|B)$ denote the conditional probability and the conditional pdf , conditioned on the event B, respectively.

2. BCJR Detector for Markov noise channels

Essentially, the channel model is the same as used in [9]. The basic channel model is depicted in Figure 2. The input symbols a_k, $0 \leq k \leq K$, drive a finite-state machine. The state of the finite-state machine at time

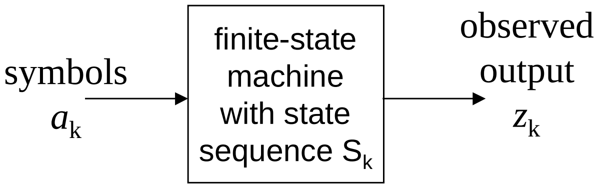

Fig. 2. Finite-state machine channel model. The observed sequence at the receiver end is z_k.

k, denoted by S_k, can take one of M values in the set $\{0, 1, \ldots, M-1\}$. We assume that at time instances 0 and K, we have $S_0 = S_K = 0$. The finite state machine produces a random sequence z_k. The following Markov assumptions apply to the state sequence S_k and the observable sequence z_k.

Channel Assumptions

1° $\mathrm{P}\left(S_k{=}m|S_{k-1}{=}m', \underline{z}_K^{-\infty}\right) = \mathrm{P}(S_k{=}m|S_{k-1}{=}m')$

2° $f\left(S_k{=}m, z_k|S_{k-1} = m', \underline{z}_{k-1}^{-\infty}\right) = f\left(S_k{=}m, z_k|S_{k-1}{=}m', \underline{z}_{k-1}^{k-L}\right)$

3° $f\left(\underline{z}_K^{k+1}|S_k = m, \underline{z}_k^{-\infty}\right) = f\left(\underline{z}_K^{k+1}|S_k = m, \underline{z}_k^{k-L+1}\right)$

Assumption 1° is the standard Markov state sequence assumption stating that the transition probability is independent of the observable process. Assumption 2° is the length L Markov noise memory assumption. Assumption 3° can actually be derived from 1° and 2°, but we list it for use later on. Contrasted to the assumptions in [6] where the observed process has a memory dependent only on the finite-state machine state transition, we additionally model the memory of the observed process to be dependent on its own past up to a finite memory length L.

We assume that the initial L observations $z_{-L+1}, \ldots,$ z_0 are *a priori* known, typically zero. Upon observing the whole sequence of samples $z_1, \ldots, z_K$, the detector's task is to determine the *a posteriori* probabilities $\mathrm{P}\left(S_k = m|\underline{z}_K^{-L+1}\right)$ and $\mathrm{P}\left(S_{k-1} = m', S_k = m|\underline{z}_K^{-L+1}\right)$. Following the notation of Bahl et.al. [6], but expanding it to allow for the Markov memory of length L, we define the following sequences of variables.

$$\alpha_k(m) = f\left(S_k = m, \underline{z}_k^{-L+1}\right) \tag{1}$$

$$\beta_k(m) = f\left(\underline{z}_K^{k+1}|S_k = m, \underline{z}_k^{k-L+1}\right) \tag{2}$$

$$\gamma_k(m', m) = f\left(S_k = m, z_k|S_{k-1} = m', \underline{z}_{k-1}^{k-L}\right) \tag{3}$$

$$\lambda_k(m) = f\left(S_k = m, \underline{z}_K^{-L+1}\right) \tag{4}$$

$$\delta_k(m', m) = f\left(S_{k-1} = m', S_k = m, \underline{z}_K^{-L+1}\right) \tag{5}$$

With these definitions, the *a posteriori* probabilities may be expressed as

$$\begin{aligned} \mathrm{P}\left(S_k = m|\underline{z}_K^{-L+1}\right) &= \frac{f\left(S_k = m, \underline{z}_K^{-L+1}\right)}{f\left(\underline{z}_K^{-L+1}\right)} \\ &= \frac{\lambda_k(m)}{\lambda_K(0)} \end{aligned} \tag{6}$$

and

$$\begin{aligned} &\mathrm{P}\left(S_{k-1} = m', S_k = m|\underline{z}_K^{-L+1}\right) \\ &\quad= \frac{f\left(S_{k-1} = m', S_k = m, \underline{z}_K^{-L+1}\right)}{f\left(\underline{z}_K^{-L+1}\right)} \\ &\quad= \frac{\delta_k(m', m)}{\lambda_K(0)} \end{aligned} \tag{7}$$

The modified BCJR algorithm is executed by the following 4 steps.

BCJR soft-output detection algorithm for Markov noise with memory length L

1° Initialization
$\alpha_0(0) = 1$, and $\alpha_0(m) = 0$ for $m \neq 0$
$\beta_K(0) = 1$, and $\beta_K(m) = 0$ for $m \neq 0$
$z_k = 0$ for $-L+1 \leq k \leq 0$

2° For $k = 1, 2, \ldots, K$
(A) $\gamma_k(m', m) = \mathrm{P}\left(S_k = m|S_{k-1} = m'\right) \cdot f\left(z_k|S_{k-1}{=}m', S_k{=}m, \underline{z}_{k-1}^{k-L}\right)$
(B) $\alpha_k(m) = \sum_{m'=0}^{M-1} \gamma_k(m', m)\, \alpha_{k-1}(m')$

3° For $k = K-1, K-2, \ldots, 0$
(C) $\beta_k(m) = \sum_{m''=0}^{M-1} \gamma_{k+1}(m, m'')\, \beta_{k+1}(m'')$

4° For $k = 0, 1, \ldots, K$
(D) $\lambda_k(m) = \alpha_k(m)\, \beta_k(m)$
(E) $\delta_k(m', m) = \alpha_{k-1}(m')\, \gamma_k(m', m)\, \beta_k(m)$

The proofs of (A)-(E) are given in Appendix A.

Application of the algorithm is very similar to the originally proposed algorithm [6]. We form a trellis of the finite-state machine, with the beginning state $S_0 = 0$ and the ending state $S_K = 0$. State pairs $S_{k-1} = m'$ and $S_k = m$ that do not have a branch assigned to them, have $\mathrm{P}(S_k = m|S_{k-1} = m') = 0$. Hence, in (A), for these pairs, we set $\gamma_k(m', m) = 0$. As we receive an observable sample z_k, for every branch in the trellis, we compute the value of the conditional pdf $f\left(z_k|S_{k-1} = m', S_k = m, \underline{z}_{k-1}^{k-L}\right)$, which is actually

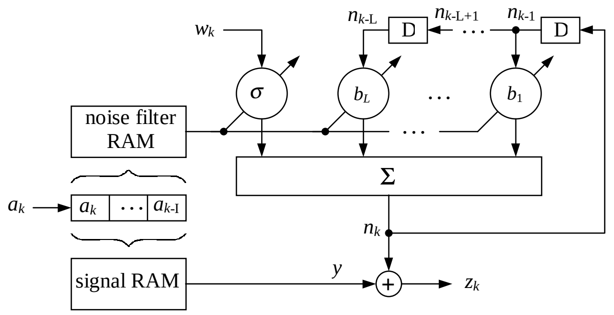

Fig. 3. Channel with intersymbol interference (ISI) of length I and signal-dependent Gauss-Markov noise with memory length L created by a signal dependent L-tap autoregressive filter.

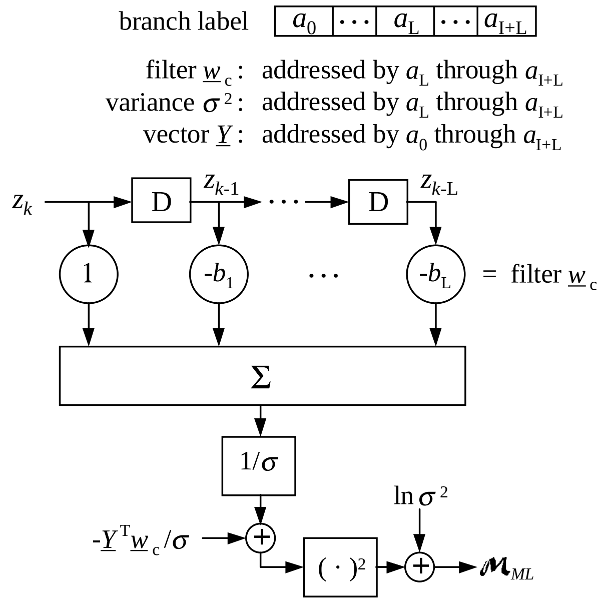

Fig. 4. Block diagram of an FIR filter computing the ML branch metric for a channel with ISI of length I and Gauss-Markov noise with memory length L. Since there are 2^{I+L+1} branches in each stage of the trellis, the branches may be labeled by $I+L+1$ bits. Of these bits, only $I+1$ are used to address the FIR filters. A bank of 2^{I+1} different FIR filters is needed for optimal detection, while 2^L branches share the same filter.

a function of $L+1$ consecutive samples $\underline{z}_k^{k-L}$. Substituted into (A), we get $\gamma_k(m', m)$. The algorithm proceeds with steps (B)-(E). Finally, Equations (6) and (7) deliver the *a posteriori* probabilities.

3. Channels with ISI and correlated Gauss-Markov noise

A Markov noise channel of particular interest in practice is the channel with intersymbol interference and correlated signal-dependent Gauss-Markov noise [8]. The sampled channel output is given by

$$z_k = y\left(\underline{a}_k^{k-I}\right) + n_k. \tag{8}$$

Here n_k is the additive Markov signal-dependent correlated noise term which we describe below. The noiseless signal $y\left(\underline{a}_k^{k-I}\right)$ is a function of $I+1$ binary input symbols a_{k-I} through a_k denoted by the vector $\underline{a}_k^{k-I} = [a_{k-I}, \ldots, a_k]^{\mathrm{T}}$. We call I the intersymbol interference length. To allow for general *nonlinear* channels, we do *not* model the noiseless response $y\left(\underline{a}_k^{k-I}\right)$ as a convolution between an impulse response and the input symbols.

The noise n_k in Equation (8) is obtained via a signal-dependent autoregressive (AR) L-tap filter, i.e.,

$$n_k = \sigma\left(\underline{a}_k^{k-I}\right) w_k + \underline{b}\left(\underline{a}_k^{k-I}\right)^{\mathrm{T}} \underline{n}_{k-1}^{k-L}. \tag{9}$$

Here w_k is a zero-mean unit-variance white Gaussian process, while $\sigma\left(\underline{a}_k^{k-I}\right)$ is the standard deviation term dependent on $I+1$ binary input symbols $\underline{a}_k^{k-I}$. The symbol $\underline{b}\left(\underline{a}_k^{k-I}\right) = \left[b_L\left(\underline{a}_k^{k-I}\right), \ldots, b_1\left(\underline{a}_k^{k-I}\right)\right]^{\mathrm{T}}$ is a vector of L autoregressive coefficients whose values depend on the $I+1$ binary input symbols $\underline{a}_k^{k-I}$. The vector $\underline{n}_{k-1}^{k-L}$ collects L previous samples of the noise process n_k, i.e., $\underline{n}_{k-1}^{k-L} = [n_{k-L}, \ldots, n_{k-1}]^{\mathrm{T}}$. We call L the Markov memory length of noise. Figure 3 depicts the channel modeled by Equations (8) and (9). We emphasize the finite Markov memory length (length L) assumption on the noise. It is precisely this assumption that lets us formulate the optimal detector on a trellis with a finite number of states.

The trellis for this channel has 2^{I+L} states. A state is defined by $I+L$ consecutive input symbols $S_k = \underline{a}_k^{k-L-I+1}$. The maximum likelihood sequence detector Viterbi detector is implemented with a bank of 2^{I+1} finite impulse response (FIR) filters. The maximum likelihood (ML) branch metric $\mathcal{M}_{ML}\left(\underline{z}_k^{k-L}, S_{k-1} = \underline{a}_{k-1}^{k-L-I}, S_k = \underline{a}_k^{k-L-I+1}\right)$ for this detector are the outputs of the FIR filters shown in Figure 4, see [9] for details. The filter in Figure 4 is the exact inverse of the autoregressive filter in Figure 3 that creates the channel noise. The ML branch metrics, i.e., the outputs of the bank of FIR filters, are also needed in the BCJR detector. It can easily be verified (see [9]) that the conditional pdf needed in step (A) of the algorithm in Section 2 is given by

$$f\left(z_k | S_{k-1} = \underline{a}_{k-1}^{k-L-I}, S_k = \underline{a}_k^{k-L-I+1}, \underline{z}_{k-1}^{k-L}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\mathcal{M}_{ML}\left(\underline{z}_k^{k-L}, S_{k-1}=\underline{a}_{k-1}^{k-L-I}, S_k=\underline{a}_k^{k-L-I+1}\right)}. \quad (10)$$

With (10) substituted in line (A), the rest of the algorithm follows as described in Section 2.

Suboptimal versions of the BCJR algorithm can be obtained by applying maximum entropy banded covariance matrix extensions [10] in similar fashion as described for suboptimal Viterbi detectors [9]. Essentially, these suboptimal detectors are based on low-order autoregressive channel model approximations [8].

4. Summary

We presented the forward-backward soft-output detector (BCJR detector) for channels with intersymbol interference and correlated (and/or signal-dependent) channel noise. The detector is the result of the merging of two algorithms, the BCJR detector for channels with white noise [6] and the Viterbi detector for channels with correlated signal-dependent noise [9]. The detector is developed under the assumption that the noise has a finite-length Markov memory. The finiteness of the Markov noise memory-length is a necessary condition for formulating the detector on a trellis with a finite number of states. In channels with Gaussian (generally, correlated and signal-dependent) noise statistics, the detector is implemented with a bank of FIR filters, each filter being the inverse of the autoregressive filter that creates the Gauss-Markov noise.

A Proofs

Proof of (A)

$$\begin{aligned}
\gamma_k(m', m) &= f\left(S_k = m, z_k | S_{k-1} = m', \underline{z}_{k-1}^{k-L}\right) \\
&= f\left(z_k | S_{k-1} = m', S_k = m, \underline{z}_{k-1}^{k-L}\right) \cdot \\
&\quad \mathrm{P}\left(S_k = m | S_{k-1} = m', \underline{z}_{k-1}^{k-L}\right) \\
&\overset{1^\circ}{=} \mathrm{P}\left(S_k = m | S_{k-1} = m'\right) \cdot \\
&\quad f\left(z_k | S_{k-1} = m', S_k = m, \underline{z}_{k-1}^{k-L}\right)
\end{aligned}$$

Here, $\overset{1^\circ}{=}$ denotes that Assumption 1° in Section 2 was used to prove the equality. Similar notation is used below.

Proof of (B)

$$\begin{aligned}
\alpha_k(m) &= f\left(S_k = m, \underline{z}_k^{-L+1}\right) \\
&= \sum_{m'=0}^{M-1} f\left(S_k = m, z_k | S_{k-1} = m', \underline{z}_{k-1}^{-L+1}\right) \cdot \\
&\quad f\left(S_{k-1} = m', \underline{z}_{k-1}^{-L+1}\right) \\
&\overset{2^\circ}{=} \sum_{m'=0}^{M-1} f\left(S_k = m, z_k | S_{k-1} = m', \underline{z}_{k-1}^{k-L}\right) \cdot \\
&\quad f\left(S_{k-1} = m', \underline{z}_{k-1}^{-L+1}\right) \\
&= \sum_{m'=0}^{M-1} \gamma_k(m', m)\, \alpha_{k-1}(m')
\end{aligned}$$

Proof of (C)

$$\begin{aligned}
\beta_k(m) &= f\left(\underline{z}_K^{k+1} | S_k = m, \underline{z}_k^{k-L+1}\right) \\
&= \sum_{m''=0}^{M-1} f\left(S_{k+1}=m'', \underline{z}_K^{k+1} | S_k=m, \underline{z}_k^{k-L+1}\right) \\
&= \sum_{m''=0}^{M-1} f\left(S_{k+1}=m'', z_{k+1} | S_k=m, \underline{z}_k^{k-L+1}\right) \cdot \\
&\quad f\left(\underline{z}_K^{k+2} | S_{k+1} = m'', \underline{z}_{k+1}^{k-L+2}\right) \\
&= \sum_{m''=0}^{M-1} \gamma_{k+1}(m, m'')\, \beta_{k+1}(m'')
\end{aligned}$$

Proof of (D)

$$\begin{aligned}
\lambda_k(m) &= f\left(S_k = m, \underline{z}_K^{-L+1}\right) \\
&= f\left(S_k = m, \underline{z}_K^{-L+1}\right) \cdot \\
&\quad f\left(\underline{z}_K^{k+1} | S_k = m, \underline{z}_K^{-L+1}\right) \\
&\overset{3^\circ}{=} f\left(S_k = m, \underline{z}_K^{-L+1}\right) \cdot \\
&\quad f\left(\underline{z}_K^{k+1} | S_k = m, \underline{z}_K^{k-L+1}\right) \\
&= \alpha_k(m)\, \beta_k(m)
\end{aligned}$$

Proof of (E)

$$\begin{aligned}
\delta_k(m', m) &= f\left(S_{k-1}=m', S_k=m, \underline{z}_K^{-L+1}\right) \\
&= f\left(S_{k-1}=m', \underline{z}_{k-1}^{-L+1}\right) \cdot \\
&\quad f\left(S_k=m, z_k | S_{k-1}=m', \underline{z}_{k-1}^{-L+1}\right) \cdot \\
&\quad f\left(\underline{z}_K^{k+1} | S_{k-1}=m', S_k=m, \underline{z}_k^{-L+1}\right) \\
&\overset{2^\circ,3^\circ}{=} f\left(S_{k-1}=m', \underline{z}_{k-1}^{-L+1}\right) \cdot \\
&\quad f\left(S_k=m, z_k | S_{k-1}=m', \underline{z}_{k-1}^{k-L}\right) \cdot \\
&\quad f\left(\underline{z}_K^{k+1} | S_{k-1}=m', S_k=m, \underline{z}_k^{k-L+1}\right) \\
&= \alpha_{k-1}(m')\, \gamma_k(m', m)\, \beta_k(m)
\end{aligned}$$

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. on Communications*, (Geneva, Switzerland), pp. 1064–1070, May 1993.

[2] W. Ryan, "Performance of high-rate turbo codes on PR4-equalized magnetic recording channels," in *Proc. IEEE Int. Conf. on Communications*, (Atlanta, GA), pp. 947–951, June 1998.

[3] M. C. Reed and C. B. Schlegel, "An iterative receiver for partial response channels," in *Proc. IEEE Int. Symp. Inform. Theory*, (Cambrdge, MA), p. 63, Aug. 1998.

[4] T. Souvignier, A. Friedmann, M. Öberg, P. Siegel, R. E. Swanson, and J. K. Wolf, "Turbo codes for PR4: Parallel versus seral concatenation," in *Proc. IEEE Int. Conf. on Communications*, (Vancouver, Canada), pp. 1638–1642, June 1999.

[5] L. L. McPheters, S. W. McLaughlin, and K. R. Narayanan, "Precoded PRML, serial concatenation, and iterative (turbo) decoding for digital magnetic recording," *IEEE Trans. Magn*, vol. 35, Sep. 1999.

[6] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Sept. 1974.

[7] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, March 1996.

[8] A. Kavčić and A. Patapoutian, "A signal-dependent autoregressive channel model," *IEEE Trans. Magn.*, vol. 35, September 1999.

[9] A. Kavčić and J. M. F. Moura, "The Viterbi algorithm and Markov noise memory," *IEEE Trans. Inform. Theory*, submitted Nov. 1997. 28 pages, accepted for publication in Jan. 2000 issue.

[10] A. Kavčić and J. M. F. Moura, "Correlation structures for optimizing information criteria," in *Proc. IEEE Information Theory Workshop on Detection, Estimation, Classification and Imaging*, (Santa Fe, NM), p. 25, Feb. 1999.