SOFT-OUTPUT DETECTOR FOR PARTIAL-RESPONSE CHANNELS USING VECTOR QUANTIZATION

Brian M. Kurkoski[1], Paul H. Siegel[2], Jack K. Wolf[2]
(1) University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo
(2) University of California San Diego, 9500 Gilman Dr., La Jolla CA, 92037

Introduction and system

Turbo equalization has been shown to have excellent performance for magnetic recording channels, and permits operation at low signal-to-noise ratios. Soft-output algorithms, such as the BCJR or SOVA algorithms, are used as the detector in proposals for turbo equalization. However, these algorithms are difficult to implement effectively at the bit rates and power consumption levels required for hard drives. We consider an implementation of the BCJR algorithm where the state metrics are vector quantized, and the operations are performed using table lookups. Table lookup operations are efficiently implemented in both hardware and software. The proposed algorithm uses off-line computation to generate vector quantizers and lookup tables, and the only run-time operations are table lookups. Using vector quantization, the size of the lookup tables can be of reasonable size. Further, the compressed representation reduces the required state metric storage.

We assume the following partial-response system. At time t, the channel inputs and outputs are denoted by x_t and c_t, respectively. The inputs are i.i.d. distributed, equiprobable zeros and ones. The impulse response of the channel is $h(D)$. A trellis with M states describes the partial-response channel. The receiver observes $y_t = c_t + n_t$, where n_t is AWGN with known variance σ^2. The channel SNR is E_s/N_0, where E_s is the average power in c_t and $N_0 = 2\sigma^2$. The BCJR algorithm is the optimal detector for intersymbol interference channels with white noise, and it computes the soft output $U_t = \log P[x_t = 1|y_1^N] - \log P[x_t = 0|y_1^N]$. For each state m and each time t, the BCJR algorithm recursively computes forward state metrics $A_t(m)$ and backward state metrics $B_t(m)$. The state metrics at time t can be represented as a vector $\mathbf{A}_t = (A_t(0), \ldots, A_t(M-1))$. In this case, the usual one-step forward recursion can be represented as a function $f : \mathbf{A}_{t+1} = f(\mathbf{A}_t, y_t)$.

Vector quantization of state metrics

In conventional implementations, each state metric is individually quantized, this is rectangular quantization of vector the $\mathbf{A}_t$. However, for vector sources with correlated components, such as the BCJR state metrics, true vector quantization requires fewer quantization points than a rectangular quantizer, for some fixed distortion. Further, the overflow technique used to avoid rescaling uses an additional quantization bit per state [1].

Consider the combination of the even-mark modulation (EMM) constraint, and the PR1 channel ($h(D) = 1 + D$). For Viterbi detection on the EMM-PR1 trellis, which has $M = 3$ states, tight bounds on the reachable space, or the recurrent region, of the state metrics have been found [2]. Samples of the BCJR algorithm's state metrics almost always fall inside the Viterbi algorithm's recurrent region. The state metrics are normalized such that $A_t(2) = 0$. Fig. 1-(a) compares the conventional rectangular quantization scheme to the recurrent region; it can easily be seen that most quantization points fall outside of the recurrent region, and thus are never used. The vector quantizer shown in Fig. 1-(b), designed using a generalized Lloyd algorithm, is more efficient than the conventional quantizer.

If $\mathbf{A}_t$ and y_t have been quantized to α_t and η_t, respectively, then the forward recursion can be implemented by a look-up table $f_Q, \alpha_{t+1} = f_Q(\alpha_t, \eta_t)$; similarly backward quantized state metrics are β_t. Finally, the BCJR algorithm outputs the log likelihood ratio U_t, computed by lookup table g: $U_t = g(\alpha_t, \eta_t, \beta_{t+1})$.

This research was supported by a JSPS Postdoctoral Fellowship, and by NSF Grant CCR-0219582, part of the Information Technology Research Program.

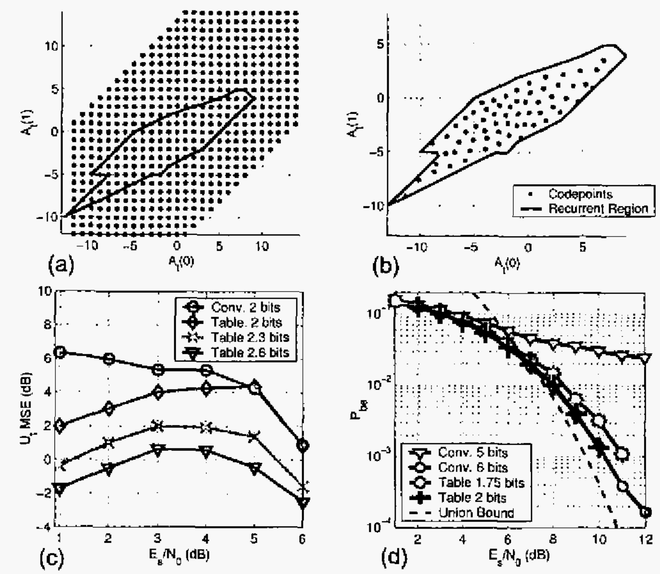

Fig. 1. (a) Conventional quantization for EMM-PR1. (b) Vector quantization for EMM-PR1. (c) Mean-squared quantization error (d) BER performance, on PR2 channel

Method for higher-order channel

The above method is direct, and has a complexity of three table-lookup operations per bit. As the number of trellis states increases, the lookup table size also increases. We note that the table lookup algorithm can operate on subtrellises of the original partial-response trellis, using smaller tables. If the trellis is partitioned into two sub-trelllies, the algorithm can be implemented with twelve table operations per bit.

Simulation results

For the EMM-PR1 channel, Fig. 1-(c) compares the conventional BCJR implementation with 2 bits per state, to the table lookup implementation, with 2, 2.3 and 2.6 bits per state. The figure of merit is the mean squared error with the soft output of the floating-point BCJR algorithm. At low SNR, the lookup algorithm has significant advantage, for fixed quantization per state.

For the four-state PR2 channel ($h(D) = (1 + D)^2$), Fig 1-(d) compares the BER of the lookup table and conventional algorithms. The BER of the lookup table quantized to 2 bits/state is roughly the same as that of the conventional algorithm with 6 bits/state. With y_t quantized to 5 bits, the table size for the forward recursion is $2^{13} \times 8$ bits.

[1] A. P. Hekstra, "An alternative to metric rescaling in Viterbi decoders," *IEEE Trans. on Comm.*, vol. 37, pp. 1220–1222, Nov. 1989.

[2] R. Calderbank, P. Fishburn, and P. Siegel, "State-space characterization of Viterbi detector path metric differences," in *26th Asilomar Conf. on Signals, Systems and Computers*, pp. 940–944, Oct 1992.