Reduced-Complexity Sequence Detection for E\textsuperscript{3}PR4 Magnetic Recording Channel

Borivoje Nikolić\textsuperscript{1}, Michael Leung\textsuperscript{2}, Leo Fu\textsuperscript{2}, Vojin G. Oklobdžija\textsuperscript{3,4}, Richard Yamasaki\textsuperscript{2}

\textsuperscript{1}Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720-1770, USA
\textsuperscript{2}Silicon Systems Inc, Texas Instruments Storage Products Group, San Jose/Tustin, CA, USA
\textsuperscript{3}Department of Electrical and Computer Engineering, University of California, Davis, CA, 95616, USA
\textsuperscript{4}Integration, 1285 Grizzly Peak Blvd, Berkeley, CA, 94708, USA

Abstract

Hardware complexity required for the implementation of E\textsuperscript{3}PR4 magnetic read channel results in slower operation and increased area and power consumption. The Viterbi sequence detector for E\textsuperscript{3}PR4 channel requires 16 states that could be realized as an array of 16 single step or two-step add-compare-select units. The method presented here reduces the complexity of the two-step implementation of the detector, by eliminating less probable branches from the trellis, using the sign of the input sample as a control value. Unlike taken branches are dynamically eliminated from the detector, decreasing its complexity by roughly 50% with negligible penalty in signal to noise ratio. The resulting detector consists of four-way, three three-way, three two-way ACS units and one adder that are dynamically assigned to 14 states of the detector.

1. Introduction

The level of intersymbol interference between neighboring recorded bits in magnetic recording channels increases with increased recording density. For such channels, maximum likelihood (ML) sequence detection is proven to be the optimum decoding method [1]. The asymptotically optimal method of implementing the ML detection technique is to use the Viterbi algorithm [2,3]. A practical technique to employ ML detection is to equalize the channel to partial response targets (4,5) of the form:

\[ h(D) = (1 - D)(1 + D)^N, \quad N \geq 1, \]

where \( D \) denotes a unity sample delay.

When \( N = 1 \), channel is known as partial response class-4 (PR4), \( N = 2 \) corresponds to extended PR4 (EPR4), and \( N = 3 \) is usually denoted as E\textsuperscript{3}PR4. As the recording density increases, the number of terms in the suitable partial response target will grow. Thus, hardware complexity will also increase, doubling the complexity of the sequence detector for every additional allowed bit of interference.

Viterbi sequence detector for E\textsuperscript{3}PR4 channel requires 16-state implementation. It can be realized as an array of 16 single-step add-compare-select (ACS) units. Applying one-step look-ahead to ACS unit roughly doubles the throughput of the system [6], resulting in a two-step implementation. In the case of a single-step implementation, ACS units are two-way and in case of two-step, they are four-way. The performance of the Viterbi detector is key to the performance of the entire magnetic read channel. However, its complexity makes it difficult to achieve desired speed of the channel, while in the same time increasing the power and area of the implementation. Thus, reducing the complexity of the detector allows for trading the area and power for the speed.

It was demonstrated [7] that the sequence can be estimated using reduced-state estimation with small loss in error rate. Near-optimum performance was achieved in EPR4-equalized magnetic read channels [8], with significantly reduced computational requirements. A method proposed in [9], was based on elimination of paths with higher error distances, which resulted in possibility to share detector states.

Modulation coding can affect the trellis and complexity of the detector, by defining constraints that restrict the occurrence of certain patterns in the code. One of the commonly used codes with that property was rate 2/3 (1,7) RLL code [10]. Trellis codes increase the minimum error distance in the code, by restricting the occurrence of three or more transitions in the medium [13].

Recently, several trellis codes have been proposed for partial response signaling that eliminate the most common error events by using coding constraints [11, 12, 13]. An example is the code described in [12], which, when applied to E\textsuperscript{3}PR4 trellis, requires only a 14-state 2-step (radix-4)
stationary Viterbi detector instead of a 16-state detector. These 14 states require eight 4-way and six 3-way ACS units for the implementation.

The method described in this paper significantly reduces the complexity of the detector by eliminating the less probable paths in the trellis. The resulting detector consists of four four-way, three three-way, two two-way ACS units and one adder that are dynamically assigned to 14 states of the detector.

This method is general, and can be applied to any type of coding as well as to a single-step detector. However, the resulting reduction in complexity depends on the used code. In this paper, the method, which we developed, is applied to trellis coded channel, resulting in roughly 50% reduced complexity.

2. Trellis Reduction

The complexity of the Viterbi detector can be reduced by dynamically eliminating improbable branches from the trellis. This elimination is dependent on the values of channel samples, by dividing the range of data into “ambiguity zones” [8]. The EPR4 signal levels and example input signal sample are shown in Figure 1.

The input sample shown is most likely to have a true, noiseless value of +2, +1, 0, +1 and is much less likely to have a +3, −2 or −3 sample. With the application of a trellis code in EPR4 channel, the minimum squared distance for an error event is 10. This corresponds to a single-bit channel-input error event, Ex = [+], or equivalently, Ey = [1 2 0 −2 −1] at the output of the channel [11]. For sample-by-sample threshold detection, the equivalent sample error event for a noiseless sample of value s being sliced into the region \{s + 2, s + 3\} or \{s − 2, s − 3\} is Ey = [4]. Therefore, for an observed sample of value s, pruning off all trellis branches except those that correspond to \{sk+1, sk, sk, sk−1\}, where sk and sk are the ceiling and floor of s, will have little effect on the overall system BER. The effective error distance of pruning off the true branch is 10*\log_{10}(1.6) = 2.04dB larger than that of the dominant error event of type Ex = [+]. Moreover, the error distance of common trellis-coded EPR4 error events, e.g. Ex = [+], [+ −], [+ 0 0 +], will further be reduced by correlation between noise samples. The probability of misdetecting the true signal sequence S_k as another signal sequence S'_k = S_k + Ey_k (assuming S'_k is also a valid sequence) is:

\[ P(Ey_k) = Q(d_k/2\sigma_k^2) \]  

where \(d_k^2 = \sum Ey_k^2\) and \(\sigma_k^2\) is the variance of the noise projected into the error event subspace along the direction of the error event Ey_k:

\[ \sigma_k^2 = (Ey_k^\prime REy_k) / (Ey_k^T Ey_k) \]  

\(Ey_k\) is the vector representation of Ey_k and \(R\) is the autocorrelation matrix of the noise at the output of the equalizer. For error event Ey = [+, + −, + 0 0 +], the corresponding Ey are \{1 2 0 −2 −1\}, \{1 1 −2 −2 1\} and \{1 2 0 −1 0 −2 −1\}. To compare the probability \(P(Ey_k)\) for pruning the true branch (Ey = [4]) to various dominant error events in trellis coded EPR4 systems, we denote \(d_k/\sigma_k\) as the error event SNR for Ey_k.

Figure 2 shows the relative error event SNR for Lorentzian pulse with white Gaussian noise. Four error event SNR curves are charted, including the trellis pruning error event, Ey = [4], and the three most dominant error events in trellis coded EPR4 system. As shown in the Figure 2, the SNR for the pruning error event is at least 3dB larger than the other dominant error events. It is around 4dB better than Ex = [+], rather than only 2.04dB, due to the effect of noise sample correlation. Therefore, the effect of branch pruning on system BER should be insignificant and will be further demonstrated by the simulation results for the practical system.

3. Detector Implementation

The separation of all the possible input cases based on Figure 1, would result in very complicated control and selection logic. In this paper, a simple methodology of trellis reduction is traced. In order to achieve it, the threshold must be set at zero.

The method uses the signs of a short sequence of data samples:

1. If the input signal level is greater than 0, the allowed input levels are +3, +2, +1, 0, −1. If the input signal level is less than 0, the allowed input levels are +1, 0, −1, −2, −3, as shown in Figure 3.

2. A sequence of four incoming samples is analyzed and the resulting two-step trellis is formed. The selection is based on a previous sample, two current samples and the next sample.

\[ +3 \]
\[ +2 \]
\[ +1 \]
\[ -1 \]
\[ -2 \]
\[ -3 \]

Figure 1: EPR4 signal levels.
Figure 2: Relative Error Event SNR for pruning error and three dominant regular error events. SNR is normalized such that the relative SNR for Ex = [+] is zero at user density of 3.5.

Figure 3: Allowed input signal levels.

With the application of a time-invariant trellis code [1], states 0101 and 1010 are permanently eliminated and resulting two-step EPR4 trellis as shown in Fig. 4 has only 14 states. Based on the above criteria, the reduced trellises in Figure 5 are formed. Only four out of sixteen reduced trellises are shown in Figure 5.

For the implementation of this type of sequence detector, a smaller number of ACS units is needed as compared to full trellis. A minimal implementation of 14-state EPR4 trellis contains:

- 4 4-way ACS units
- 1 3-way ACS unit
- 4 2-way ACS units
- 2 adders (1 way ACS)

The 14 states will be dynamically assigned to 11 ACS units. The dynamic assignment reduces the size of ACS array, but requires additional multiplexers in the critical path and control logic outside of the critical path.

This approach reduces the hardware complexity of the Viterbi detector by approximately 50%. For example, the total number of adders in ACSs is reduced to 29, as compared to 64 in the full trellis implementation.

Figure 4: Two-step EPR4 trellis, with two states eliminated by coding constraints.

For different sequences of inputs, different structures of detector are needed as seen in Figure 5. The requirements for the number and size of ACS for each state vary with different sample sequences. The assignment plan for the states that share ACS units if applied directly, would lead to a complicated control resulting in significant overhead in multiplexer logic. The multiplexers of different sizes, from 2-way to 5-way would be needed.

A simple solution is to add more resources besides initial 4 4-way, 1 3-way 4 2-way ACS units and two adders. The straightforward assignment plan can be derived if the symmetrical states (0 and 15, 1 and 14, 2 and 13, 3 and 12, 4 and 11, 6 and 9, 7 and 8) share the same resources. In that case some of the ACS units need to be extended to accept more inputs. The minimum solution requires:

- 4 4-way ACS units
- 3 3-way ACS units
- 3 2-way ACS units
- 1 adder.

In the proposed implementation, the branch metrics (BM) and ACS units would be assigned together. This implementation eliminates the branch metrics multiplexers in front of ACS units.

Sharing of ACS units can be done by adding the multiplexers both at the inputs and at the outputs of the ACSs. Since the assignment is simple, only two control signals are needed at the input and at the output, that will control 11 2-way multiplexers at the input and 11 2-way demultiplexers at the output.
Moreover, the demultiplexers after ACSs can be eliminated if the implementation is based on existing ACS resources, Figure 6. In this case, the detection is based on ACSs, which hold corresponding states for a given input sequence. ACS units are named as 0/15, 15/0, 1/14, 2/13, 13/2, 3/12, 4/11, 11/4, 6/9, 9/6, 7/8. This results in slightly more complicated control of 11 2-way multiplexers at the input [14].

The problem that arises by adding more branches in the trellis is that some of the states that generate them do not exist. Such states should not be considered in the calculation of new states. That can be done by adding a single (valid) tag bit to every state. Invalid states exist in cases where two states share only one ACS resource (states 1 and 14, 3 and 12, 7 and 8).

Reduced complexity system was simulated to verify the proper operation and to evaluate the possible loss in BER comparing to full implementation. The resulting comparison of trellis coded channels, at user density of 3.0, implementing 8/9 code with full trellis from [13] and code from [12] with reduced trellis is shown in Figure 7. A baseline performance is shown for the RLL-coded, rate 16/17 EPR4 detector. Simulations confirm that indeed there is a small loss, below 0.3 dB at lower signal-to-noise ratios, and virtually no loss at higher SNR with BER lower than $10^{-6}$. SNR is defined as the zero to peak amplitude of the input signal over the noise power within 2 times the Nyquist bandwidth. BER curves are generated with one hundred errors for each data point in the graph (i.e. $10^6$ bits are exercised for the BER at $10^{-6}$).

4. Complexity Estimation

The implementation of single step E²PR4 Viterbi detector, requires 7 branch metrics unit (BMU), 16 ACS unit and 16 survival registers. Each ACS unit consists of two adders and one subtractor. In case of two-step implementation, branch metrics are added together, to form 39 different two-step branch metrics. Four-way ACS unit consist of 4 adders and 6 parallel subtractors.

Table 1 shows that reduced complexity detector implements Viterbi algorithm at less than 30% of hardware increase from single-step, while achieving two-step throughput. Full two-step implementation is approximately 2.7 times bigger than full single-step implementation.
5. Conclusion

The proposed method reduces the complexity of a two-step Viterbi detector implementation by half. This was achieved by eliminating less likely taken branches from the trellis, in conjunction with applied trellis coding. The elimination uses a simple method based on the sign of the input sample. The smaller area and power of the detector could be traded for speed enhancement. The system level simulations have proven that there is no significant loss in bit-error rate as compared to full trellis implementation.

References


Table 1: Complexity comparison between different implementations of E²PR4 detectors.

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Full 2-step</th>
<th>Full 1-step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit</td>
<td>Count</td>
<td>%</td>
</tr>
<tr>
<td>ACS Adders</td>
<td>64</td>
<td>100</td>
</tr>
<tr>
<td>ACS Subtract</td>
<td>96</td>
<td>100</td>
</tr>
<tr>
<td>1-step BMUs</td>
<td>14</td>
<td>100</td>
</tr>
<tr>
<td>BM adds</td>
<td>39</td>
<td>100</td>
</tr>
<tr>
<td>Survival Regs.</td>
<td>16</td>
<td>100</td>
</tr>
</tbody>
</table>

Reduced, 2-st.