PRE-PROCESSING OF CONVOLUTIONAL CODES FOR REDUCING DECODING POWER CONSUMPTION

Wei Shao

APT Group, School of Computer Science
The University of Manchester
Oxford Road, M13 9PL

Linda Brackenbury

APT Group, School of Computer Science
The University of Manchester
Oxford Road, M13 9PL

ABSTRACT

This paper presents a new pre-processing approach for convolutional codes that can provide the adaptive capability to a standard Viterbi decoder. By identifying and generating the decoded data from the zero Hamming distance path, all computations in the Viterbi decoder can be avoided. This makes it possible to stop the Viterbi decoder in real time as long as no error occurs in the received code words; the Viterbi decoder is restarted to correct errors otherwise. With this approach, significant power, i.e. 97% of a standard Viterbi decoder power dissipation, can be saved when the Eb/No is as low as 13dB.

Index Terms— convolutional coding, adaptive decoding algorithm, pre-decoding

1. INTRODUCTION

Convolutional decoding algorithms and their hardware implementation, such as the Viterbi algorithm, have shown good noise tolerance for error-correcting codes in many applications. However, these implementations require an exponential increase in VLSI area and power consumption to achieve increased decoding accuracy.

As an example, Fig 1 illustrates the amount of (normalised to 1 at Eb/No=2dB) energy consumed by a normal Viterbi decoder in correcting each error; this increases exponentially with the reduction of the noise level. This energy efficiency issue is caused by the error-independency associated with most of the current decoding methods for convolutional codes. With the increasing requirement of decoding accuracy for digital communication implementations, more and more powerful coding schemes are proposed using concatenated decoders with multiple decoding iterations, i.e. Turbo codes [1]. Thus, it becomes common to use a complex decoder to process a code sequence with a low error probability. The energy efficiency issue, therefore, becomes a major concern for these coding schemes in applications requires low power.

This paper proposes a new approach to identify the possible error-free part of the received convolutional code words sequence so that decoding operations can be avoided; this enables significant power saving at low noise level with no degradation in error-correction capabilities. In this paper, the algorithm of identifying the no-error sequence of a convolutional code is described in Section 2. In order to improve the possible power saving, a simple syndrome decoding scheme is proposed and discussed in Section 3. Section 4 describes the test results of FPGA implementations. Finally, section 5 summarizes our efforts and offers directions for future work in this area.

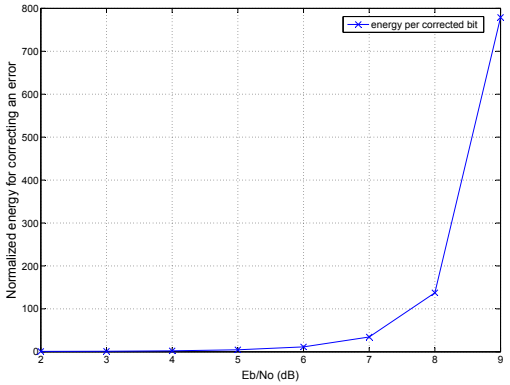

Fig. 1. Normalised energy in correcting an error of a standard Viterbi decoder (rate=1/2 and k=7) at different noise levels.

2. PRE-DETECTION OF THE ZERO-HAMMING DISTANCE CODE WORDS SEQUENCE FOR THE CONVOLUTIONAL CODE

2.1. Zero-Hamming distance code words path

Consider the coded digital communication system model shown in Fig 2. Let the input and the output of the (n, m, k) convolutional encoder of rate n/m and constraint length k be represented by the n-component vector:

$$X_t = (x_t^{(0)}, x_t^{(1)}, x_t^{(2)}, \cdots, x_t^{(n-1)}) \quad (1)$$

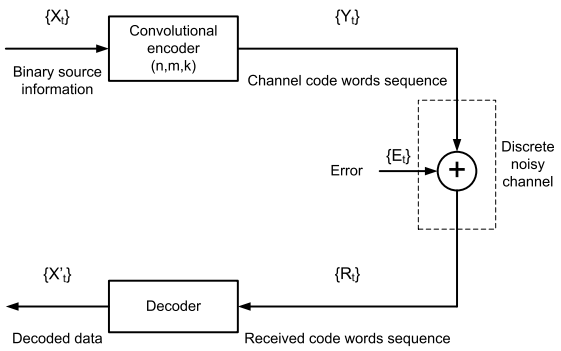

Fig. 2. Convolutional coded digital communication system model.

and the m-component vector:

$$Y_t = (y_t^{(0)}, y_t^{(1)}, y_t^{(2)}, \cdots, y_t^{(m-1)}) \tag{2}$$

for $t \geq 0$, respectively. A binary information sequence $\{X_t\}$ is encoded into a code word sequence $\{Y_t\}$ and sent to a discrete noisy channel at time t. Assuming the hard-decision is used, the channel noise causes binary errors $\{E_t\}$ to be added to the transmitted data, which gives the received code words sequence

$$R_t = (r_t^{(0)}, r_t^{(1)}, r_t^{(2)}, \cdots, r_t^{(m-1)}) = Y_t + E_t. \tag{3}$$

To recover the encoded information, $\{R_t\}$ is compared with all possible legal code words sequences, $\{\hat{Y}_t\}_0$, $\{\hat{Y}_t\}_1$, $\{\hat{Y}_t\}_2$, $\cdots$, $\{\hat{Y}_t\}_i$, by using the Hamming distances

$$d(\{R_t\}/\{\hat{Y}_t\}_i) = \sum_{j=0}^{t} d(R_j/\hat{Y}_j) \tag{4}$$

as their measurements of likelihood. Then, $\{\hat{Y}_t\}_i$ that is closest in Hamming distance to $\{R_t\}$ is chosen as the most-likely sequence and used to generate the decoded data. This decoding method is well known as most-likelihood (ML) decoding, e.g. Viterbi algorithm and sequential decoding algorithms which have been widely used for decoding convolutional codes. In order to achieve the best accuracy, the most-likely path needs to be longer than a certain length, e.g. 5 times the constraint length for the rate 1/2 codes.

When the received sequence is error-free, $\{R_t\}$ and $\{Y_t\}$ are identical and have a minimum Hamming distance of zero. A legal path $\{\hat{Y}_t\}_i$ with a zero Hamming distance, of length $L \geq 5 * k$, will be selected by the Viterbi decoder with the same trace back length L. Therefore, if we can identify a $\{\hat{Y}_t\}_i$ with a zero Hamming distance prior to the decoding process, the decoded data can be directly provided by this sequence with no loss in decoding accuracy; this avoids complex decoding operations and the large amount of power dissipation in the Viterbi decoder.

The inverse of a convolutional code, introduced in [2], has been implemented as the pre-decoding approach in [3], and can also be used to identify the code words path of the zero-Hamming distance.

2.2. Architecture of Zero-Hamming distance code words path detection

The structure of this detection approach for a R=1/2, k=7 convolutional code of a generator in octal (171,133) is shown in Fig 3. In this approach, a received code word is first pre-

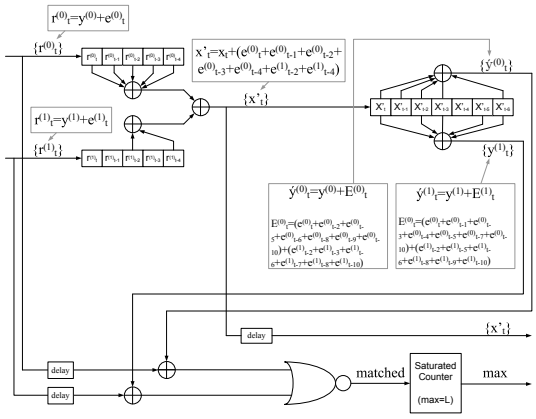

Fig. 3. Architecture of zero-Hamming distance path detection for convolutional code of rate=1/2 and k=7.

decoded using the inverse circuits from [3] which gives the pre-decoded data

$$X'_t = X_t + E'_t \tag{5}$$

where $\{E'_t\}$ represents the convolutionally encoded error sequence given by

$$E'_t = \sum_{i=0}^{4} e_{t-i}^{(0)} + e_{t-2}^{(1)} + e_{t-4}^{(1)} \tag{6}$$

and $e_t^{(0)}$ and $e_t^{(1)}$ are hard-decision channel errors at time t associated with $y_t^{(0)}$ and $y_t^{(1)}$, respectively. Then, the pre-decoded data is re-encoded by the convolutional encoder with the same generator (171,133) as the encoder used for encoding the source information. The re-encoded legal code words $R'_t = (\acute{y}_t^{(0)}, \acute{y}_t^{(1)})$ at time t are:

$$\acute{y}_t^{(0)} = y_t^{(0)} + E_t^{(0)} \tag{7}$$

$$\acute{y}_t^{(1)} = y_t^{(1)} + E_t^{(1)} \tag{8}$$

where $E_t^{(0)}$ and $E_t^{(1)}$ are defined by (9) and (10):

$$E_t^{(0)} = E'_t + E'_{t-1} + E'_{t-2} + E'_{t-3} + E'_{t-6} \tag{9}$$

$$E_t^{(1)} = E'_t + E'_{t-2} + E'_{t-3} + E'_{t-5} + E'_{t-6} \tag{10}$$

Finally, it compares each bit of the received code word with the corresponding re-encoded legal code word and counts the number of continuous matching bits. When the count reaches the maximum number L, this indicates that the received code words path matches a legal code words path of length L. For a Viterbi decoder with the same trace back length L, it will select this re-encoded code words path as the most-likely path for generating the decoded data. The pre-decoded data X'_{t-L} corresponding to the first stage of the matched code words path will be the same as that from Viterbi decoding. Therefore, it can be used as the decoded output from the Viterbi decoder. Generally, as long as the length of the matched paths is the same as the trace back length of the Viterbi decoder, there is no accuracy degradation by replacing the Viterbi decoder output with the valid pre-decoded output.

2.3. Adaptive Viterbi decoder and simulation results

Fig 4 illustrates the structure of the adaptive Viterbi decoder using the pre-detection of the zero-Hamming distance code words path. In this design, when the Zero-Hamming distance

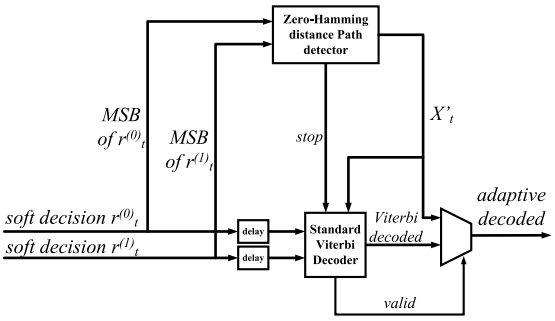

Fig. 4. Architecture of the adaptive Viterbi decoder using the Zero-Hamming distance code words path detection.

detector finds a zero-Hamming distance path of a length equal to the standard Viterbi decoder trace back length, it stops the Viterbi decoder from processing the delayed received code words and the multiplexer selects the pre-decoded data x'_t as the adaptive decoder output; otherwise, the output from the standard Viterbi decoder is selected as the adaptive decoder output. In case of being valid, x'_t is also used to initialise the decoder state.

The percentages of the Viterbi decoding operations of the adaptive decoder are obtained by Matlab simulations using data streams of 10^6 bits with AWGN. The results are shown in Fig 5. It indicates that the percentage of the Viterbi decoding operations start to reduce at 4dB and reaches the minimum level at around 13dB where there is almost no Viterbi decoding operation as very few errors occurs at this Eb/No level. This, therefore, suggests that a significant power saving can be achieved at low noise levels by the adaptive Viterbi

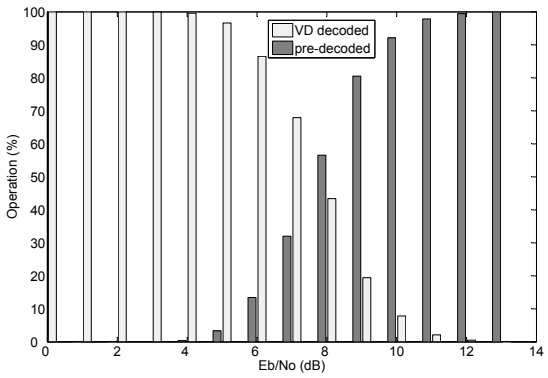

Fig. 5. Percentages of the Viterbi decoding operations.

decoder with the zero-Hamming distance code words path detection approach.

3. FPGA IMPLEMENTATION AND TEST RESULTS

The proposed adaptive Viterbi decoder has been implemented on a Virtex4 XC4VSX35 FPGA and tested within circuit simulations. The design includes a Viterbi decoder IP core from Xilinx and the post place and route adaptive decoder runs at a maximum frequency of 165MHz which is the same frequency as the standard IP core. The BER performance of the adaptive Viterbi decoder is shown in Fig 6. The results indicate that

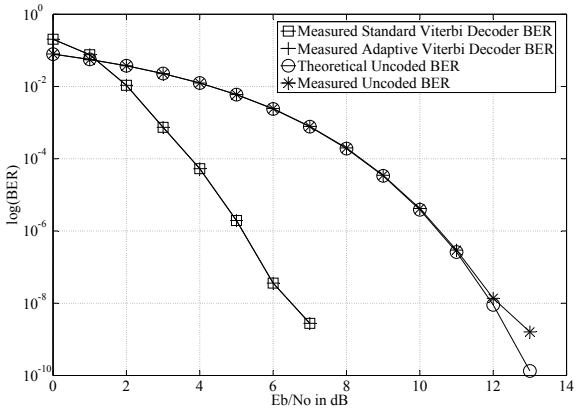

Fig. 6. Measured BER of the FPGA implementation of the adaptive Viterbi decoder.

the BER of the proposed adaptive decoder exactly matches the standard Viterbi decoder BER. This suggests the adaptive decoder provides the same decoding accuracy as a standard Viterbi decoder.

The power of the standard and adaptive designs were estimated by power simulations using the xpower tool from Xilinx. The results are illustrated in Fig 7. The adaptive decoder

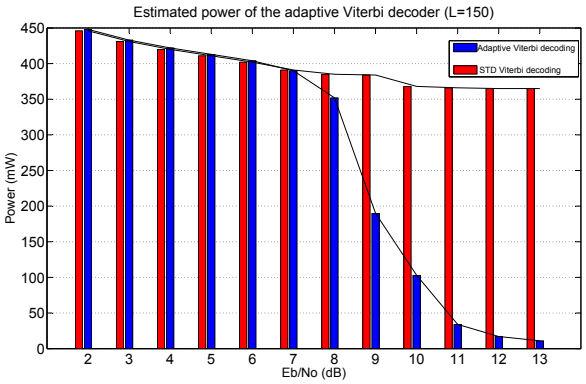

Fig. 7. Estimated dynamic power dissipation of the adaptive Viterbi decoder designs. Here the length of the zero-Hamming distance path is 150 instead of 35 in order to match the delay of the standard Viterbi decoder IP core. This results in the power saving starting at a lower noise level compared to using a length of 35.

starts to save power at the Eb/No of 7dB and reaches the maximum saving of 97% at 13dB. For Eb/No, therefore, larger than 7dB, the new adaptive decoder consumes less power than the original adaptive decoder design.

The energy efficiency of the adaptive designs can be analysed using energy per-correction measurements. The results

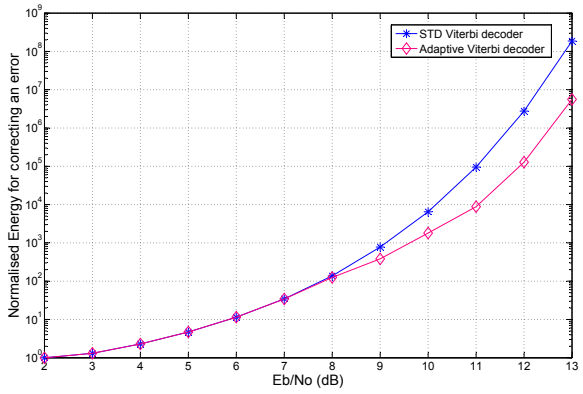

Fig. 8. Energy per-correction of the adaptive Viterbi decoder designs.

are shown in Fig 8 and indicate that the adaptive design consumes the lower energy in correcting an error when the Eb/No is larger than 7dB and, thus, it is the more power efficient than the standard Viterbi decoder.

3.1. Conclusion

The proposed zero-Hamming distance path detection approach is a simple way to estimate the appearance of an error in the received code words path. Therefore, it can provide adaptive capability to a Viterbi decoder with little hardware overhead. Test results indicate that with L equal or larger than 5 times the constraint length, the decoding accuracy of the adaptive decoder is identical to the standard Viterbi decoder with the same trace back length. Potential reduction of the power consumption in the adaptive decoder is from 1.4% to 97% as Eb/No increases from 7dB to 13dB.

The accuracy of the pre-decoded data depends on the length of the zero-Hamming distance path. Therefore, the zero-Hamming distance path detection approach can also be applied to other convolutional decoding schemes by predetermining the length of zero-Hamming distance paths corresponding to the target decoding accuracy.

4. REFERENCES

[1] Claude Berrou, Alain Glavieux, and Punya Thitimajshima, "Near Shannon Limit Error-correcting Coding and Decoding: Turbo-codes," in *Proc. IEEE International Communication Conference*, France, Oct. 1993, pp. 1064–1070.

[2] J. L. Massey and M. K. Sain, "Inverses of linear sequential circuits," *IEEE Trans. Comput.*, vol. C-17, pp. 330–337, Apr. 1968.

[3] Shuji Kubota, Shuzo Kato, and Tsunehachi Ishitani, "Novel Viterbi decoder vlsi implementation and its performance," *IEEE Trans. Commun.*, vol. 41, pp. 1170–1178, Aug. 1993.