Low-Power Approach for Decoding Convolutional Codes with Adaptive Viterbi Algorithm Approximations

R. Henning and C. Chakrabarti
Department of Electrical Engineering
Arizona State University
Tempe, Arizona 85287-5706
480-965-9516
chaitali@asu.edu

ABSTRACT
Significant power reduction can be achieved by exploiting real-time variation in system characteristics while decoding convolutional codes. The approach proposed herein adaptively approximates Viterbi decoding by varying truncation length and pruning threshold of the T-algorithm while employing trace-back memory management. Adaptation is performed according to variations in signal-to-noise ratio, code rate, and maximum acceptable bit error rate. Potential energy reduction of 70 to 97.5% compared to Viterbi decoding is demonstrated. Superiority of adaptive T-algorithm decoding compared to fixed T-algorithm decoding is studied. General conclusions about when applications can particularly benefit from this approach are given.

Categories and Subject Descriptors
C.3 [**Special-Purpose and Application-Based Systems**]: *Signal processing systems.*

General Terms
Algorithms, Performance, Experimentation.

Keywords
Low Power, Viterbi Algorithm, Adaptive T-algorithm Decoding, Convolutional Codes.

1. INTRODUCTION
A Viterbi decoder [1, 2] is an important target for power reduction in many low-power communications devices. It can account for more than one-third of power consumption during baseband processing in current generation cellular telephones [3]. As integrated circuits continue to become smaller and faster, the appeal of higher complexity Viterbi decoders for higher memory order convolutional codes increases. Higher memory order codes can achieve superior coding performance without requiring precious additional channel bandwidth. However, to counteract the exponential dependence of Viterbi decoder complexity on memory order in low-power designs, good power reduction methods that exploit variations in the communications system are needed.

As is the case in many designs today, significant untapped power reduction potential lies in dynamically varying a Viterbi decoder implementation according to real-time changes in system characteristics. Examples of other applications in which system characteristics have been successfully exploited to reduce energy consumption include Reed-Solomon channel coding [4] and an encryption processor [5]. The goal of the approach proposed in this paper is to reduce energy consumption while decoding high memory order punctured convolutional codes in a system where channel bandwidth availability, channel signal-to-noise ratio (SNR), and/or maximum acceptable bit error rate (BER) vary real-time.

There are four main contributions of this paper: 1) A new system-dependent, low-power approach for decoding convolutional codes called adaptive T-algorithm decoding is proposed. 2) Variation in the potential of this approach as system characteristics (code rate, E_b/N_0, maximum acceptable BER) vary is studied. 3) The superior energy reduction potential of adaptive T-algorithm decoding versus Viterbi and fixed T-algorithm decoding [6] is demonstrated. For example, potential energy reduction of 70% to 97.5% compared to hardware Viterbi decoding is demonstrated. 4) Guidance is provided for application of the new approach.

2. BACKGROUND
For the purposes of this work, a Viterbi decoder implementation that employs trace-back memory management can be described in terms of four basic characteristics—convolutional code memory order *m* supported, trace-back memory length, truncation length, and punctured code rates *R* supported. A T-algorithm decoder can be described with one additional characteristic, pruning threshold.

Viterbi decoding with trace-back memory management is performed by 1) using the bits received to generate paths that represent likely transitions made by the convolutional encoder state machine over time and 2) periodically tracing one of these paths back to determine one or more bits that were likely encoded. The possible paths that can be generated by a convolutional encoder are illustrated with a trellis diagram. The convolutional code memory order determines the height or number of states per stage of this trellis, which represents the number of paths stored at any time by the Viterbi decoder. Trace-back memory length represents the width of the trellis or the length of the stored paths in terms of bits.

ISLPED '02, August 12-14, 2002, Monterey, California, USA.

Truncation length is the number of bits a path is followed back before a decision is made about what bit was encoded. If truncation length is shorter, more bits can be decoded per trace-back operation given the same trace-back memory length but likelihood of a decoding mistake also increases. One of the ways the proposed adaptive T-algorithm decoder reduces decoder energy consumption is by exploiting this tradeoff.

The other way the adaptive T-algorithm decoder reduces energy consumption is by adapting the pruning threshold of the T-algorithm decoder. The T-algorithm decoder is a variation of the Viterbi decoder that applies a threshold to the accumulated path metrics the Viterbi decoder uses to select only one path per state of a trellis stage for storage. A lower pruning threshold causes fewer paths to be found and stored but also increases likelihood of a decoding mistake.

Finally, punctured code rates must be treated separately by the adaptive T-algorithm decoder because codes of different rates can react differently to the same pruning threshold and truncation length. Multi-rate coding with punctured codes is important because higher rates conserve channel bandwidth when channel quality is higher. They are easily decoded with the same trellis structure by simply ignoring some of the additional bits that would normally arrive for the lower rate codes but have been punctured.

3. ADAPTIVE T-ALGORITHM DECODING

The proposed approach, referred to as adaptive T-algorithm decoding, reduces energy consumption by adapting the pruning threshold and truncation length of a T-algorithm decoder employing trace-back memory management to real-time system changes. These changes include variation in code rate, E_b/N_0, and maximum acceptable BER. This approach greatly impacts decoder energy consumption because pruning threshold controls the average number of paths stored per trellis stage in a T-algorithm decoder, while truncation length controls the frequency with which trace-back through the trellis is performed. The number of operations performed by the decoder, especially memory accesses, is highly dependent on the average number of paths stored and the trace-back frequency.

The energy optimal pruning threshold and truncation length pair to employ for any combination of code rate, E_b/N_0, and maximum acceptable BER can be determined experimentally as the decoder is designed. An appropriate subset is stored in a look-up table for adaptation use at runtime. The Viterbi decoder and fixed T-algorithm decoder can be thought of as special cases of the adaptive T-algorithm decoder that are not as energy efficient because they are not as versatile. Next, it is shown that adaptive T-algorithm decoding can be performed with significantly less energy consumption than these other decoders in systems where code rate, E_b/N_0, and maximum acceptable BER vary.

4. EXPERIMENTAL RESULTS

In this section, the energy reduction potential of the adaptive T-algorithm decoder is assessed through experimental results. Two examples involving punctured codes are studied. The first example involves a system with fixed maximum acceptable BER and variable E_b/N_0. This example applies to an implementation that requires consistent minimum error correction performance over a variable quality channel. The second example is for a system with fixed E_b/N_0 and variable maximum acceptable BER. This example is applicable when minimum error correction performance needs vary over time because of variation in the type of information being received. The particular convolutional codes studied in these examples are quite practical for contemporary applications. The BER ranges considered were chosen to reduce simulation time.

4.1 Energy Estimates

In order to provide generally applicable results in the examples of this section, normalized decoder energy consumption estimates are given that do not assume a particular hardware implementation technology. To obtain these high-level estimates, a decoder is broken down into simple operations (8-bit adds, 8-bit compares, memory reads, memory writes, and associated control) for which reasonable relative energy estimates are assumed [7]. The estimate for the entire decoder is then calculated from this breakdown; the relative energy estimate for each operation is simply multiplied by the number of times that operation is executed on average per decoder trellis stage.

The reason energy consumption can be significantly reduced for the T-algorithm by reducing pruning threshold and truncation length is mainly because these parameters greatly affect the number of times path memory is accessed. Actually, the average number of times nearly all calculations are performed per trellis stage is proportional to the average number of surviving paths per stage, which is controlled by pruning threshold. However, the average number of trace-back memory reads performed per stage is proportional to the number of trellis stages that are accessed during trace-back, *trace-back memory length*, divided by the number of stages that are processed between trace-back operations, *trace-back memory length – truncation length*.

The overhead of an adaptive versus fixed T-algorithm implementation comes from monitoring the code rate, E_b/N_0, and BER expectation and then choosing the appropriate threshold and truncation length to use from a practically sized table. Because none of these overhead operations are expected to be complex or performed at a high rate relative to decoder operations in practice, their energy consumption is assumed to be negligible.

4.2 Viterbi vs. Fixed T-algorithm Decoding

To provide a comparison baseline for assessment of the adaptive T-algorithm in the examples that follow, the alternative Viterbi and fixed T-algorithm decoders need to be discussed. First, consider a Viterbi decoder implementation employing trace-back memory management that decodes $R = \{1/2, 2/3, 3/4\}$, $m = 8$ punctured convolutional codes over an AWGN channel using 3-bit soft decisions. This implementation uses a trace-back memory length of 94 trellis stages to minimize latency and memory size while supporting a truncation length of 93 stages. Truncation length of about 93 is needed to decode the rate-3/4 code with negligible quality loss. Assume that with this decoder, acceptable application quality is achieved for E_b/N_0 as low as 2 dB while decoding the rate-1/2 code. The BER achieved in this case, approximately 0.0037, is the maximum acceptable BER

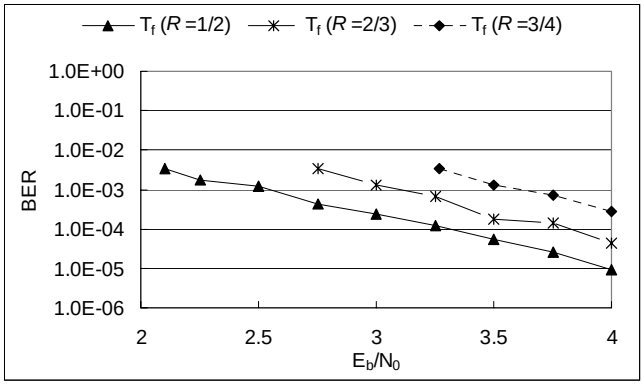

Figure 1. BER versus E_b/N_0 curves for the fixed T-algorithm (T_f) decoder as code rate varies.

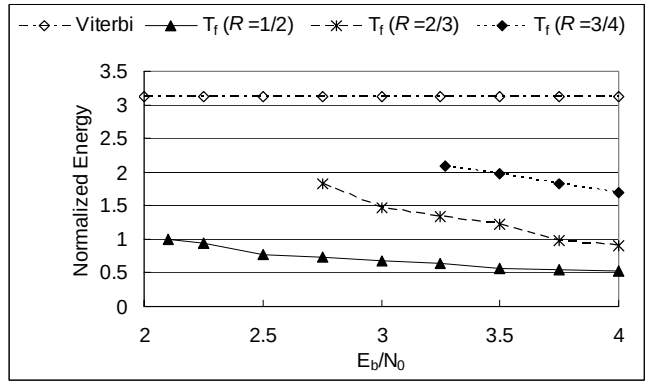

Figure 2. Normalized energy estimates for the Viterbi and fixed T-algorithm (T_f) decoders as code rate and E_b/N_0 vary.

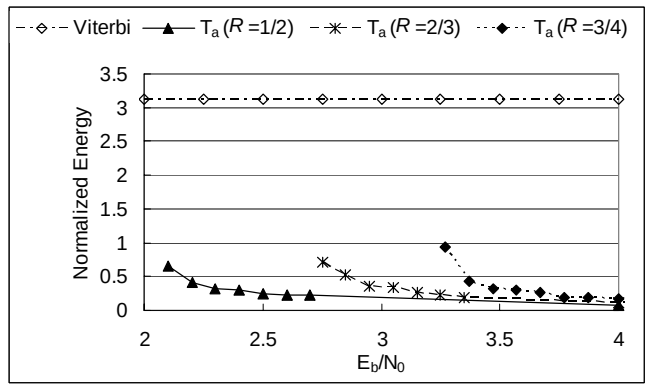

Figure 3. Normalized energy estimates for the Viterbi and adaptive T-algorithm (T_a) decoders as code rate and E_b/N_0 vary. Maximum acceptable BER is fixed at 0.0037.

To save significant energy, the T-algorithm with a fixed pruning threshold and fixed truncation length can be substituted for the Viterbi algorithm in this decoder while maintaining the same trace-back memory structure. This fixed T-algorithm decoder implementation is referred to as the T_f decoder. By employing a fixed pruning threshold of 17, the 0.0037 BER is not exceeded for E_b/N_0 greater than 2.1 dB while decoding the rate-1/2 code. Thus, with this pruning threshold, a slight coding loss of about 0.1 dB is incurred when decoding the rate-1/2 code. The loss for rate-2/3 and rate-3/4 codes is negligible. Figure 1 shows decoder performance for all three codes. Only points having BER below the maximum acceptable 0.0037 are shown, since this constraint is assumed to govern operation of the multi-rate system.

For this small coding loss, it is estimated that the T_f decoder can consume 33% to 83% less energy than the multi-rate Viterbi decoder when E_b/N_0 is between 2.1 dB and 4.0 dB, as shown by the normalized energy estimates in Figure 2. Additional energy reduction is possible for higher E_b/N_0. However, energy consumption slowly converges to the same point for all code rates as the average number of paths kept by the T-algorithm per decoding stage converges to one. Thus, the upper limit on energy reduction for the T_f decoder relative to the Viterbi decoder is estimated to be about 87.5%.

4.3 Fixed BER, Variable E_b/N_0 Example

The energy consumption of the T_f decoder can be reduced by adapting pruning threshold and truncation length according to variations in E_b/N_0, as well as code rate, without allowing BER to exceed 0.0037. This implementation is referred to as the T_a decoder. Trace-back memory size remains the same for the T_a decoder as the Viterbi and the T_f decoders. On the other hand, it does not necessarily employ a pruning threshold of 17 or a truncation length of 93 stages.

Figure 3 shows normalized energy estimates for the T_a decoder as punctured code rate and E_b/N_0 vary. The points in this plot correspond to optimal pruning threshold and truncation length settings. These pairs were determined experimentally by finding the pairs that result in the lowest energy estimate for each E_b/N_0 value shown without causing the BER of 0.0037 to be exceeded. It is important to note that these pairs are energy estimate dependent. Thus, different pairs could result for different energy estimates. In practice, optimal values are predetermined and a reasonably sized subset is stored in a look-up table. Adaptive decoding is performed by applying the lowest energy look-up table pair pointed to by estimates of the instantaneous value of *R*, E_b/N_0, and maximum acceptable BER at the decoder.

As code rate and E_b/N_0 vary over the ranges shown in Figure 3, energy reduction estimated at 70% to 97.5% can be achieved with the T_a decoder compared to the Viterbi decoder. Recall, the T_f decoder achieves energy reduction of 33% to 83% energy reduction for the same code rates and E_b/N_0 values.

Similar to the T_f decoder, energy consumption of this implementation converges to a common point for all code rates as E_b/N_0 increases. For the T_a decoder, this limit is about 98.5% less than the energy estimate for the Viterbi decoder. The reason this limit is smaller than that of the T_f decoder is that truncation length is not fixed for the T_a decoder. The T_a decoder limit is reached when both truncation length and the average number of paths kept by the T-algorithm per decoding stage are one.

Another interesting point to consider is that the system in which the T_a decoder is employed might conserve bandwidth whenever possible by always choosing the highest code rate that will not cause the maximum acceptable BER to be exceeded for a given E_b/N_0. Though this policy does not significantly affect the energy consumption of the Viterbi decoder, it can cause the T_a decoder, as well as the T_f decoder, to expend more energy. In these decoders, energy consumption of a lower rate code is generally lower than a higher rate code because fewer paths on average are

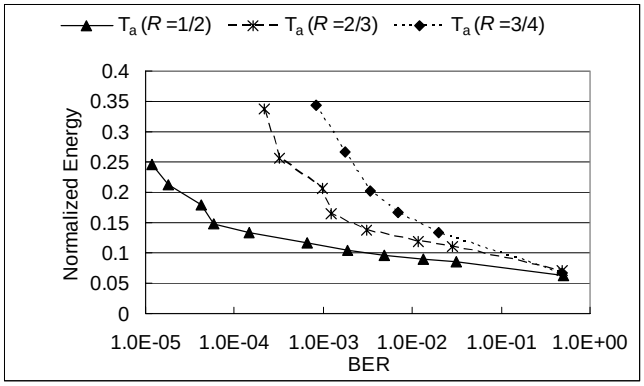

Figure 4. Normalized energy estimates for the adaptive T-algorithm (T_a) decoder as code rate and maximum acceptable BER vary. E_b/N_0 is fixed at 3.75 dB. The normalized energy estimate for the Viterbi decoder ≈ 3.12.

kept by the T-algorithm for the same E_b/N_0. Therefore, in some cases, significant energy might be conserved if a lower code rate (higher bandwidth) than necessary is used when surplus bandwidth is available. For example, if E_b/N_0 = 3.27 dB, rather than conserve bandwidth by employing a rate-3/4 code, about 35% less energy is required by the T_a decoder if the rate-2/3 code is used and about 85% less is consumed for the rate-1/2 code.

4.4 Fixed E_b/N_0, Variable BER Example

The energy consumption of the T_f decoder can also be reduced by adapting pruning threshold and truncation length according to variations in maximum acceptable BER. Figure 4 shows normalized energy estimates for this T_a decoder. (E_b/N_0 can vary as it did in the previous example but is assumed fixed at 3.75 dB here for simplicity.) Each experimentally determined parameter pair shown minimizes energy consumption for E_b/N_0 = 3.75 dB while achieving a certain BER. The point in this figure that lies closest to the maximum acceptable BER at a given time without exceeding it corresponds to the minimum-energy pair. Interestingly, for all the points in this plot, a truncation length of 74 can be used with negligible effect on energy consumption or maximum acceptable BER that can be supported.

For maximum acceptable BER ranging from about 10^{-5} to 0.03 and depending on code rate variation, energy reduction estimated at 89% to 97% can be achieved with the T_a decoder versus the Viterbi decoder. The T_f decoder always achieves the same BER and energy consumption for a given code rate and E_b/N_0, regardless of variation in maximum acceptable BER. For E_b/N_0 = 3.75 dB, estimates of these values can be found from Figures 1 and 2. Normalized energy consumption for the T_f decoder is estimated to be about 0.54 at a BER of $2.6\text{x}10^{-5}$ for the rate-1/2 code, 0.97 at a BER of $1.4\text{x}10^{-4}$ for the rate-2/3 code, and 1.83 at a BER of $7.3\text{x}10^{-4}$ for the rate-3/4 code. As shown in Figure 4, the same BERs can be achieved with the T_a decoder with energy savings compared to the T_f decoder of about 63% for the rate-1/2 code, 65% for the rate-2/3 code, and 81% for the rate-3/4 code. As maximum acceptable BER increases, these energy savings increase because the T_f decoder cannot exploit this change. For a maximum acceptable BER of 0.03, the T_a decoder outperforms the T_f decoder by about 84% for the rate-1/2 code, 89% for the rate-2/3 code, and 93% for the rate-3/4 code.

Like the previous example, in some cases, significant energy might be conserved if a lower code rate (higher bandwidth) than necessary is used when surplus bandwidth is available. For example, if maximum acceptable BER ≈ 0.001, rather than conserve bandwidth by employing a rate-3/4 code, about 40% as much energy is required by the T_a decoder if the rate-2/3 code is used and about 66% less is consumed for the rate-1/2 code.

5. CONCLUSIONS

The adaptive T-algorithm decoder described in this paper adaptively approximates the Viterbi algorithm according to variations in convolutional code rate *R*, E_b/N_0, and maximum acceptable BER. Significant energy reduction is achieved by adapting truncation length and pruning threshold of the T-algorithm while employing trace-back memory management. While meeting maximum acceptable BER constraints, the adaptive T-algorithm decoder can provide considerable energy reduction compared to Viterbi and fixed T-algorithm decoders.

6. ACKNOWLEDGMENTS

This work was funded by the National Science Foundation's S/IUCRC Center for Low Power Electronics.

7. REFERENCES

[1] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", IEEE Trans. Information Theory, vol. IT-13, pp. 260-269, 1967.

[2] G. D. Forney, Jr., "The Viterbi Algorithm", Proc. IEEE, vol. 61, no. 3, pp. 218-278, Mar. 1973.

[3] I. Kang and A. Willson Jr., "Low-Power Viterbi Decoder for CDMA Mobile Terminals", IEEE J. of Solid-State Circ., vol. 33, no. 3, pp. 473-482, Mar. 1998.

[4] M. Goel and N. Shanbhag, "Low-power Channel Coding via Dynamic Reconfiguration," in Proc. of ICASSP, pp. 1893-1896, Mar. 1999.

[5] J. Goodman, A. Chandrakasan, and A. Dancy, "Design and Implementation of a Scalable Encryption Processor with Embedded Variable DC/DC Converter", ACM Design Automation Conference (DAC), pp. 855-860, June 1999.

[6] S. J. Simmons, "Breadth-First Trellis Decoding with Adaptive Effort", IEEE Trans. on Comm., vol. 38, no. 1, pp. 3-12, Jan. 1990.

[7] F. Catthoor et al., "Custom Memory Management Methodology – Exploration of Memory Organization for Embedded Multimedia System Design," Norwell, MA: Kluwer, 1998.