List Viterbi Decoding with Continuous Error Detection for Magnetic Recording

Dragan Petrović, Borivoje Nikolić, Kannan Ramchandran
{dragan,kannanr,bora}@eecs.Berkeley.edu
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

***Abstract*-- The List Viterbi Algorithm (LVA) with an arithmetic coding based continuous error detection (CED) scheme is applied to high-order partial-response magnetic recording channels. Commonly used magnetic recording systems employ distance-enhancing codes along with parity-check post-processors to correct most dominant error events. The system presented here inserts a CED encoder between the outer Reed-Solomon (RS) code and the channel, and utilizes LVA along with CED's error detection capabilities to improve the performance of the Viterbi decoder. Simulations show that this system results in a 2 dB improvement over MTR encoded EEPRML decoding at a BER of 2×10^{-6} in additive white Gaussian noise and localizes error occurrences to the end of the sector.**

I. Introduction

Contemporary magnetic recording systems use high-order partial response channels with maximum-likelihood (ML) sequence detection to achieve high recording densities. To provide greater noise immunity at such high recording densities, distance-enhancing codes such as maximum transition run (MTR) constraints [1-6] are used to eliminate most dominant error events. In addition to dominant error eliminating codes, parity-check bits with post-processing [7], [8] and high-rate inner forward error correcting (FEC) codes [9] have been proposed to correct some of the remaining dominant error events.

The use of continuous error detection (CED) has been proposed as a replacement for cyclic redundancy check (CRC) codes in automatic repeat request (ARQ) based transmission, list Viterbi algorithm (LVA) decoding of rate compatible punctured convolutional (RCPC) codes, and joint equalization and coding for inter-symbol interference (ISI) channels [10], [11]. In [10] it was shown that LVA with CED outperforms standard, CRC-based, LVA. All of these schemes make use of CED's ability to detect errors "on the fly," i.e. without having to wait for an entire block of data to be received before possible errors in transmission can be detected. In this paper, that property is used to improve the performance of the ML decoder in magnetic recording systems.

The next section gives an introduction to arithmetic coding based CED. Section 3 explains the basic idea of LVA. Section 4 describes how LVA with CED was incorporated into the magnetic recording channel. Section 5 justifies some predictions of the relative performance of the proposed system. Section 6 provides simulation results along with a discussion of the importance of those results, and Section 7 concludes the paper.

II. Arithmetic Coding And Continuous Error Detection

A. Arithmetic Coding

Arithmetic coding [12], [13] is a *source coding* algorithm. The algorithm asymptotically attains the theoretical entropy rate of a stationary source [12], i.e. it is a "perfect" lossless compression scheme. It is used to compress data by mapping the data-string to a real number in the interval [0,1) in a symbol-wise recursive fashion. *The resulting codeword is the binary representation of that number*. First, the [0,1) interval is partitioned into segments with each segment corresponding to exactly one symbol in the source-alphabet, and each segment's length proportional to the probability of occurrence of the corresponding symbol. Whenever a symbol is encoded, the segment corresponding to that symbol becomes the new code-space and is itself partitioned in the same manner as the [0,1) interval was partitioned at the beginning. This procedure is best clarified by an example.

Consider a source alphabet consisting of three symbols $\{a,b,c\}$ with the following probabilities given in binary representation:

$$p(a) = 0.011\,,\ p(b) = 0.011\,,\ p(c) = 0.01 \qquad (1)$$

Let us consider encoding a sequence *acb*… of these three symbols. Figure 1 illustrates the procedure. After encoding the first symbol, *a*, the code-space would be restricted to the interval [0, 0.011) in binary. Knowing that the first bit in the binary representation of the final range will have to be 0, the encoder can output 0 as the first bit as soon as *a* is encoded. The second symbol to be transmitted, *c*, confines the transmit sequence to the range

$$\big[0.011*(0.011 + 0.011),\ 0.011*1\big) = \big[0.01001,\ 0.011\big). \qquad (2)$$

0-7803-7206-9/01/$17.00 © 2001 IEEE

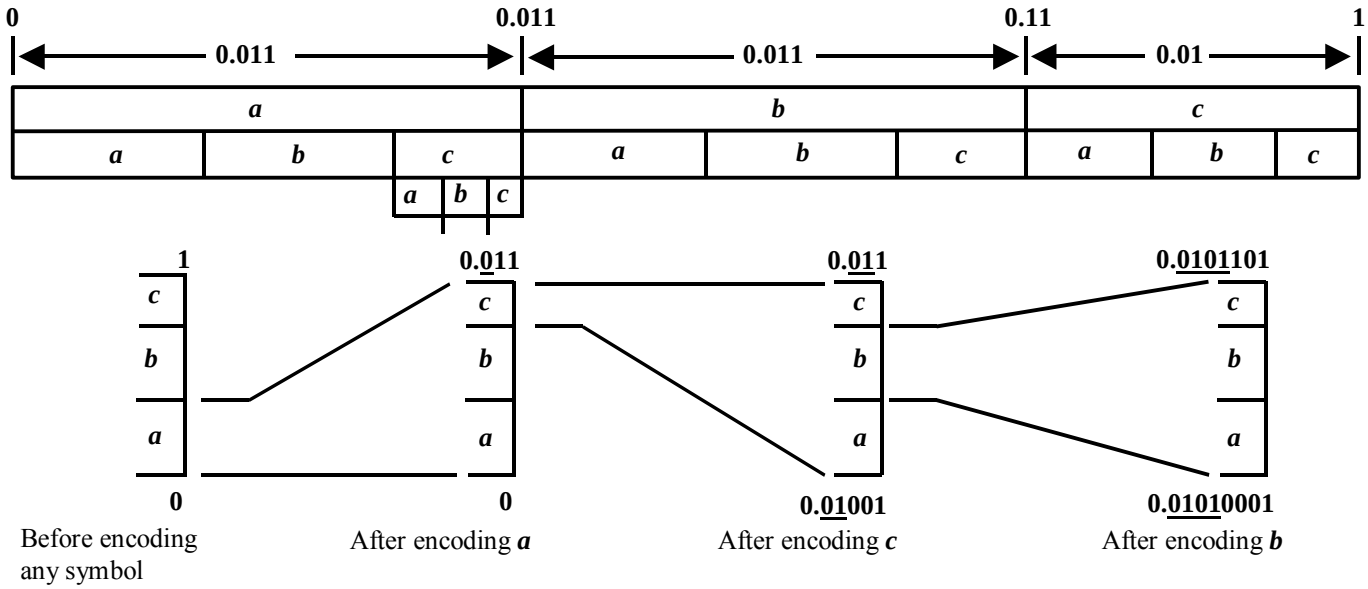

Figure 1. Arithmetic Coding Example

Now, the encoder knows that the second bit must be 1 and can immediately output that second bit. Each symbol encoded reduces the code-space. As the number of encoded symbols increases, we get a segment whose length is proportional to the probability of occurrence of that particular sequence of symbols.

The decoder, upon receiving bits, maps them to a segment of the number line (just as was done by the encoder) and retrieves the original data. As more bits arrive at the receiver, more data-symbols can be decoded. In this example, if the first three bits received were 010, then the value of the fractional number received cannot exceed 0.011, and so the decoder knows that the first information symbol was *a* since the interval [0,0.011) corresponds to the symbol *a*. Thus, both the encoder and decoder process data *continuously* (i.e. symbol-by-symbol or bit-by-bit) instead of having to wait for large blocks of data to be input.

B. Continuous Error Detection (CED)

Arithmetic coding has been shown to be the optimal algorithm for data compression [12]; however, as with most compression algorithms, a single bit error can be catastrophic because synchronization is lost. In arithmetic coding, if a single bit is lost or corrupted, synchronization is lost, and all the subsequent bits will be mapped to an incorrect portion of the number line resulting in erroneous decoding of the subsequent data. It is precisely this property that has been proposed for use in error-detecting channel codes [14].

Consider again the three-symbol alphabet, $\{a,b,c\}$, example given above with the code-space [0,1) partitioned as above, only this time consider a source which never produces symbol *c*. Should the decoder ever receive a bit-stream which corresponds to a *c* at the encoder, it will *know* that an error has occurred somewhere in the bit-stream. The CED scheme employed in this paper works exactly the same way. User bits are arithmetically encoded onto the [0,1) code space, but with a certain portion (of length ε) of the code space reserved for a forbidden symbol. This forbidden symbol is never encoded, so that whenever the decoder processes a received bit-stream that corresponds to an occurrence of the forbidden symbol, the decoder knows that an error has occurred during transmission.

The advantage of CED over commonly used error-detection block codes such as CRCs is that CED need not wait for an entire (often very large) block to be received before it can detect an error in transmission. The CED decoder processes each bit as it arrives, and each bit can potentially indicate that an error has occurred *somewhere* in the transmission, though it cannot point out *exactly* where the error has occurred. The question is, "will the decoder surely detect an error, and how long after the first bit-error will the system detect it?" There is no deterministic answer to this question, but a probabilistic answer can be given. As soon as the first error occurs, synchronization is lost, and each subsequent bit will point to an erroneous position on the number line. This new position is well modeled as being uniform on the current interval. Thus, each bit (corrupted or not) after the first bit-error will point to the forbidden symbol (i.e. the decoder will detect that an error has occurred) with probability ε. So, each bit in a CED-encoded stream serves as a check for each previous bit. Let us define the random variable *Y* to be the number of bits it takes to detect a bit-error after it occurs, then *Y* can be modeled as having a geometric distribution:

$$P(Y=k)=(1-\varepsilon)^{k-1}\varepsilon, \quad \forall\ k\in\{1,2,3...\} \tag{3}$$

Thus, the probability that an error in transmission is detected converges to 1 with every subsequently received bit, and the probability that the error is detected within *k* subsequent bits is $1-(1-\varepsilon)^k$. This model has been shown to be accurate in

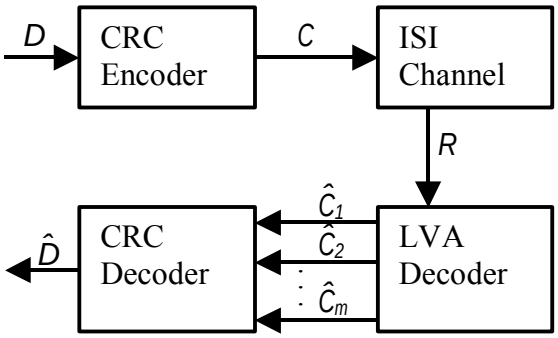

Figure 2. Conventional LVA System

[11] and will be used in later analyses of the system's performance.

III. List Viterbi Algorithm (LVA)

LVA [15] is an extension of the regular Viterbi algorithm. Unlike the Viterbi algorithm which finds the most likely (ML) path to each state at each stage of the trellis and outputs the single ML path through the state-space trellis, LVA tracks a list of *m* ML paths to each state at each stage of the trellis and outputs an ordered list of *m* ML paths. The paths are ordered by their path metrics. The path with the lowest path metric is considered to be the most likely path, the path with the second lowest path-metric is considered to be the second most likely path and so on. Conventional LVA implementations, shown in Figure 2, use an error-detection code such as a CRC. The data first has some CRC parity-check bits appended to it, it is then sent through the ISI channel, and the LVA-decoder outputs an ordered list of ML received bit-sequences. The parity bits of the most likely bit sequence are first checked to see if any errors are contained within the sequence. If no errors are detected, that sequence is declared to be correct. If an error is detected in the most likely sequence, the second most likely sequence is then checked for errors, and so on. The list of paths has finite size, usually about 3-9, and if all of the paths in the list contain errors, the decoder either asks for retransmission or outputs the path with the lowest path-metric and flags it.

IV. System Design

The proposed system is shown in Figure 3. Simulations were run only on the inner code, although the results do have an implication on the possible design of the outer Reed-Solomon (RS) code.

The system operates as follows: user data is first encoded by an outer RS code, and the output of the RS encoder is then CED encoded one sector (4096 bits) at a time. The amount of redundancy introduced by the CED encoder can easily be set to any positive rational number by varying ε. In the simulations presented here, this redundancy was set to 3% because that is comparable to the redundancy added for parity-checks by [7] and [8] or inner FEC codes by [9]. The bit-stream at the output of the CED encoder is fed into an MTR encoder. The output of the MTR encoder is then passed through the Lorentzian channel, noise is added and the resulting waveform is equalized to the EEPR4 target and sampled. This sampled stream is then input to the LVA detector, which keeps a list of ML paths through the trellis and works in conjunction with the MTR and CED decoders to prune out any paths that violate the MTR or CED constraints. In Figure 3, the LVA, MTR and CED decoders are shown as separate blocks to point out the fact that three types of signal processing are being done, but they are done concurrently. Thus, the components within the dashed box labeled "LVA-MTR-CED" can be thought of as a single decoder in practice. In the end, the ML path that satisfies the MTR and CED constraints is sent to the RS decoder for further processing.

For comparison, a similar system setup was also used to simulate the performance of a magnetic recording system with a conventional LVA decoder. The only difference between the system shown in Figure 3 and the conventional LVA system is that the CED encoder and decoder were replaced by a CRC encoder and decoder. In that system, the LVA decoder works in conjunction with the MTR decoder to find the ML paths through the trellis and then outputs that list of paths to the CRC decoder. Simulating that system makes it possible to discriminate between gains provided by keeping a list of several paths through the trellis and gains provided by using CED for early error detection.

Both of these systems were compared to the EEPRML decoding technique in which a regular, single-path Viterbi decoder is used with no error detection codes. For reference, simulations were run for both EEPRML with and without MTR coding.

V. Expected Performance

There are two types of errors that can occur in the output of the LVA-MTR-CED decoder. *Type 1* occurs when the ML path through the LVA trellis contains an error that is not detected in time, i.e. by the time the CED decoder detects an error in the ML path, the correct path may already have been discarded from the finite list of ML paths tracked by the LVA. The probability of this type of error is negligible at high signal-to-noise ratios (SNR), but it can dominate the performance of low-SNR systems. *Type 2* occurs when the ML path through the LVA trellis contains an error that goes completely undetected by the CED decoder. Since each bit in a CED-encoded stream serves as a check for each prior bit, as discussed in Section II.B, this type of error is highly unlikely to occur early in the bit-stream. For example, let p_b denote the bit error rate (BER) of the single-path Viterbi decoder. This is the probability that the ML path of the LVA decoder (which would also be the output of a single-path Viterbi decoder) contains an error in any given bit position. Then from equation (3), one can see that the probability that an error occurs in the ML path in the 500th bit position from the end of the sector and goes undetected by the LVA-CED-MTR decoder, when $\varepsilon = 3\%$, is $p_b(1-0.03)^{500} = p_b 2*10^{-7}$, and the probability that an undetected error occurs in a bit-

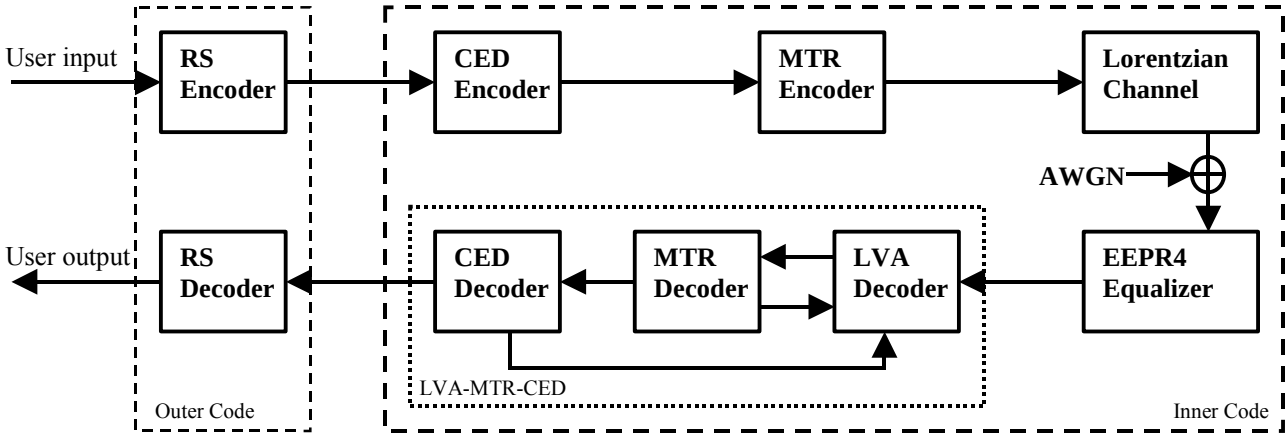

Figure 3. Proposed LVA-CED system for magnetic recording

position prior to that decays exponentially. Thus, errors that do occur in this LVA-CED scheme will be restricted to the end of the sector with high probability, allowing for outer FEC codes that can focus on correcting errors at the end of the sector.

VI. Simulation Results

Figure 4 shows the simulated relative performances, at a user density of 3.0, of the EEPRML system with and without MTR coding, the LVA with CRC and MTR system, and the proposed LVA with CED and MTR system. At high SNR, no gains were observed by increasing the number of paths kept by the LVA decoders from 6 to 7; therefore, the list size used for both the LVA-CED system and LVA-CRC system was 6.

As can be seen from Figure 4, at low SNR, the LVA-CED system performs considerably worse than both the LVA-CRC system and both EEPRML systems. This is to be expected since, at low SNR, errors of Type 1 do occur, and they are not limited to the end of the sector. However, at high SNR, the LVA-CED system provides two important benefits:

1. At a BER of $2\text{x}10^{-6}$, it gives a 2dB improvement over EEPRML with MTR and a 1dB improvement over an LVA-CRC based system.
2. Error occurrences are probabilistically confined to the end of each sector.

The first point shows that, while some performance gains are obtained by keeping a larger list size through the Viterbi trellis, those gains can be further increased by being able to detect errors early in the bit-stream and immediately discarding the incorrect paths. These gains are expected to be even greater at higher SNR.

The second point may be the biggest advantage offered by the LVA-CED system since it allows for the design of outer FEC codes that focus on correcting errors in those bytes. This phenomenon is obtained only in the LVA-CED system and is the result of CED's ability to detect errors early by using subsequent bits to check each previous bit in the stream. This advantage could become even greater if sector sizes could be extended beyond 512 user bytes because only the last several bytes of each sector, *regardless of sector-size*, are likely to contain errors.

To quantify this statement consider a sector of size s. Let the desired sector error rate be p_s. Since the LVA-CED scheme offers greater protection to bits earlier in the sector, it is possible to find a value n such that the probability of at least one error occurring within the first n bits is less than p_s. Define $\alpha := 1/(1-\varepsilon)$ and p_b as in Section V. Since this is not a binary symmetric channel, and the average error event is longer than one bit, we have $P(e_i \mid \textit{no errors before } i) < p_b$, where e_i is the event that an error occurred at bit position i of the output of the EEPRML inner decoder. Assuming that enough paths through the Viterbi trellis are tracked so that any error detected by the CED decoder can be corrected, we can use the following calculations to find the largest n such that the probability of even a single bit error in the first n bits is less than the desired p_s. This also gives the smallest value, s-n, of bits at the end of the sector where probability of error exceeds p_s.

$$P(\text{ at least one error in first } n \text{ bits of LVA-CED decoder output }) \le \sum_{i=0}^{n} P(e_i \mid \text{no errors before } i) * P(\text{error at } i \text{ goes undetected}) < \sum_{i=0}^{n} p_b (1-\varepsilon)^{s-i} = p_b \alpha^{-s} \sum_{i=0}^{n} \alpha^i =$$

$$= p_b \alpha^{-s} \frac{1-\alpha^{n+1}}{1-\alpha} \le p_s \qquad (4)$$

Thus,

$$\alpha^{n+1} \le 1 - \alpha^s (1-\alpha) \frac{p_s}{p_b} \qquad \text{(Note that } 1\text{-}\alpha \text{ is negative)} \qquad (5)$$

But for large values of s, the 1 on the right hand side of the inequality becomes negligible giving

$$n < (\log(p_s / p_b) + \log(\alpha - 1) + s\log(\alpha)) / \log(\alpha) - 1$$
$$= s + (\log(p_s / p_b) + \log(\alpha - 1)) / \log(\alpha) - 1 \qquad (6)$$

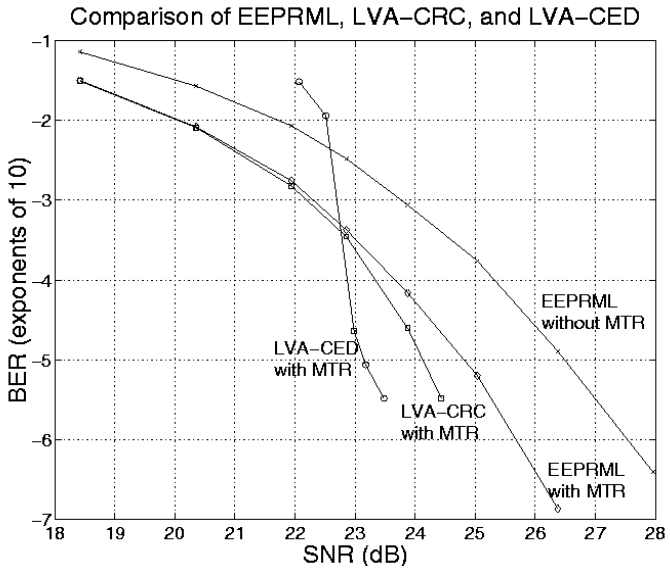

Figure 4. Performance comparison of EEPRML (with and without MTR), List Viterbi Decoding with CRC, and the proposed List Viterbi Decoding with Continuous Error Detection at a user density of 3.0.

If n is the bit-position such that the probability of any errors before the n^{th} bit is less than p_s, then the number of bits, s-n, at the end of the sector not offered this high level of protection by the LVA-CED system is constant with respect to s for large s:

$$s - n > (\log(p_s / p_b) + \log(\alpha - 1)) / \log(\alpha) + 1 \qquad (7)$$

Thus, as the sector size, s, increases the relative portion of bits which are not offered this high level of protection decreases to zero as 1/s. For p_b=10^{-7}, p_s=10^{-13}, ε=0.03, and s large (a thousand or more) the constant, s-n, is 569.

This localization of errors to the end of each sector allows for the design of outer FEC codes that specialize in correcting errors at the end of a sector. Thus, the LVA-CED system should give a lower sector error rate at a given inner-code-BER than systems in which the errors at the output of the inner code are uniformly distributed throughout the bit-stream.

VII. Conclusion

This paper presents a system that employs the list Viterbi algorithm (LVA) along with arithmetic coding based continuous error detection (CED) for recording on the EEPR4 channel. First, the advantages of keeping more than one path through the Viterbi trellis are examined by implementing a conventional LVA decoder which uses a cyclic redundancy check (CRC) code to select the correct path from the list output by the LVA decoder. This LVA-CRC system was found to give a 1dB improvement over an MTR-coded EEPRML system at 2x10^{-6} BER. The LVA decoder was then implemented in conjunction with the CED decoder, which could do early error detection and eliminate incorrect paths early in the bit-stream allowing the LVA to use those resources to track other candidate paths. Adding this early error detection capability provided another 1dB gain at 2x10^{-6} BER and confined bit errors to the end of a sector. This localization of errors can be used to design outer FEC codes that specifically protect against errors in those bytes giving a lower sector error rate than in systems where the errors are not localized. Even for larger sector sizes the number of bytes at the end of the sector where probability of error exceeds any desired threshold remains constant.

It should be noted that the LVA-CED system could be applied to *any* ISI channel, and the benefits of early error detection and localization of errors would still be present. The results presented in this paper of applying the LVA-CED system to the EEPR4 channel with AWGN are just one example.

Acknowledgments

The authors would like to thank Anand Raghavan and Rohit Puri for useful discussions during the course of this project.

References

[1] R. Karabed, P. Siegel, "Coding for higher order partial response channels," *Proc. 1995 SPIE Int. Symposium on Voice, Video and Data Communications*, vol. 2605, pp. 115-126, Philadelphia, PA, Oct. 1995.

[2] J. Moon, B. Brickner, "Maximum transition run codes for data storage systems," *IEEE Trans. Magn.*, vol. 32, no. 5, Sept. 1996.

[3] W.G. Bliss, "An 8/9 rate time-varying trellis code for high density magnetic recording," *IEEE Trans. Magn.*, vol. 33, no.5, part.1, pp. 2746-2748, Sept. 1997.

[4] B.E. Moision, P.H. Siegel, E. Šoljanin, "Distance-enhancing codes for digital recording," *IEEE Trans. Magn.*, vol. 34, no. 1, pp. 69-74, Jan. 1998.

[5] R. Karabed, P.H. Siegel, E. Šoljanin, "Constrained coding for binary channels with high intersymbol interference," *IEEE Trans. Info. Theory*, vol. 45, no. 6, pp. 1777-1797, Sept. 1999.

[6] K. Knudson-Fitzpatrick, C.S. Modlin, "Time-varying MTR codes for high density magnetic recording," *1997 Global Telecomm. Conf. (GLOBECOM'97)*, Conference Record, pp. 1250-1253, Phoenix, AZ, Nov. 3-8, 1997.

[7] T. Conway, "A New target response with parity coding for high density magnetic recording channels," *IEEE Trans. Magn.*, vol. 34, no. 4, pp. 2382-2386, July 1998.

[8] J. Sonntag, B. Vasić, "Implementation and bench characterization of a read channel with parity check postprocessor," *Proc. TMRC'00*, Santa Clara, CA, pp. B1-B2, Aug. 14-16 2000.

[9] H. Sawaguchi, M. Kondou, N. Kobayashi, S. Mita, "Concatenated error correction coding for high-order PRML channels," *Proc. GLOBECOM'98*, Sydney, pp. 2694-2699, Nov. 1998.

[10] R. Anand, J. Chou, I. Kozintsev, K. Ramchandran, "Continuous error detection for reliable communication," submitted to *IEEE Trans. Comm.* July 1999.

[11] J. Chou, K. Ramchandran, "Arithmetic coding based continuous error detection for efficient ARQ-based image transmission," *IEEE J. Sel. Areas Comm.*, January 2000.

[12] G. Langdon, "An introduction to arithmetic coding," *IBM J. Res. Develop.*, vol. 28, pp. 135-149, March 1984.

[13] T. Bell, J. Cleary, I. Witten, *Text Compression*, Prentice Hall, 1990

[14] C. Boyd, J. Cleary, S. Irvine, I. Rinsma-Melchert, I. Witten, "Integrating error detection into arithmetic coding," *IEEE Trans. Comm.*, vol. 45, pp. 1-3, Jan. 1997.

[15] N. Seshadri, C. Sundberg, "List Viterbi decoding algorithms with applications," *IEEE Trans. Comm.*, vol. 42, pp. 313-323, April 1994.