Signal Processing for Multitrack Digital Data Storage

Thomas Conway, *Member, IEEE*, Richard Conway, *Member, IEEE*, and Sebastien Tosi

Electrical and Computer Engineering Department, University of Limerick, Limerick, Ireland

Using a multitrack format in both optical and magnetic data storage applications yields important improvements in system performance, including higher data density and higher data transfer rates. However, the full advantage in data density can be achieved only through the use of joint equalization and joint detection. This paper addresses the complexity of implementing these functions and proposes a transform domain equalization architecture and a reduced-complexity detection method based on a breadth first search of a one-dimensional time-varying representation of a two-dimensional target response. With this method, the complexity of joint equalization and joint detection is not unreasonably greater than that of existing single-track systems and should be feasible for implementation with emerging integrated circuit technologies.

***Index Terms*—Magnetic recording, multidimensional signal detection, optical data storage, partial response signaling, Viterbi detection.**

I. Introduction

THE rapid advance in data storage technologies, both magnetic and optical, have resulted in exponentially increasing data storage densities with hard disk drives approaching terabyte capacities. This has resulted from the improvements and innovations in magnetic read head technology and recording media as well as improvements in the signal processing employed. Similarly, the reductions in laser wavelength and improved optics have increased the capacity of optical storage systems. While the data rates of such systems have also increased rapidly, they have done so at a lower rate than capacity improvements. Hence, there is a need to achieve higher read and write speeds. One potential method of achieving such increases in transfer rate is to consider the use of multitrack storage, where data is written and read in parallel with N_R coherent tracks. Such a scheme has been demonstrated for optical storage [1], where data is written on a two-dimensional (2-D) hexagonal lattice consisting of seven coherent rows of data bits. In principle, a similar approach can be applied in the magnetic recording application.

The immediate benefit of such multitrack systems is in the potential data rate increase by a factor of N_R. This can be achieved with the application of N_R conventional read channels operating in parallel.

However, the existence of coherent bits in a multitrack format can be exploited [2]–[5]. Hence, considerable benefits can be obtained by application of joint signal processing techniques to a multitrack format, provided realistic complexity constraints on the signal processing can be achieved.

A. Benefits of Multitrack Recording

Most disk-based magnetic and optical data storage systems store data as concentric tracks or a continuous spiral, but require a guard band between the tracks to prevent the read transducer from responding to adjacent track data.

The use of a multitrack format allows for the possibility of avoiding the use of a guard band within the multitrack itself. This allows for the possibility of increased data density if joint detection of the data can be performed.

Many data storage systems require preamble or pilot patterns to be located at predetermined locations. These are used to acquire important channel information such as gain, offset, frequency, and phase control in a robust fashion. The use of a multitrack format could allow for increased efficiency in such format overhead. In addition, run length constraints for timing recovery purposes may be distributed across the N_R tracks potentially increasing code efficiency.

In the case of magnetic recording systems, the track following accuracy is dependent on the accuracy of the servo system, which is normally based on the use of dedicated servo wedges at predetermined locations on the disk platters. In between those wedges, the system is open loop. The use of a multitrack format in this case could allow for a "radial phase error" calculation that could compensate for track misregistration either through the servo actuator or directly through the application of radial shifting (interpolation) by the signal processing.

These potential advantages can only be attained if the signal processing required can be achieved with a reasonable level of complexity given the data rates involved. Hence, the development of read channel architectures that can achieve the required processing throughput are of interest. In this paper, Section II describes a basic architecture for a multitrack read channel and proposes efficient implementation methods for the key blocks. The key blocks required are a joint equalizer and a joint detector. Section III then illustrates the performance of such a multitrack read channel using real digitized data from a prototype multitrack optical disk system.

II. Read Channel Architecture

The purpose of a read channel system is to provide signal processing functionality that processes the read signals in a suitable manner such that the original stored data may be recovered with high reliability yet with reasonable implementation complexity.

By extension from the single track case, a general architecture is required to provide an equalization function, which filters the read signal, equalizing it to either a predefined or programmable partial response.

Digital Object Identifier 10.1109/TMAG.2005.845394

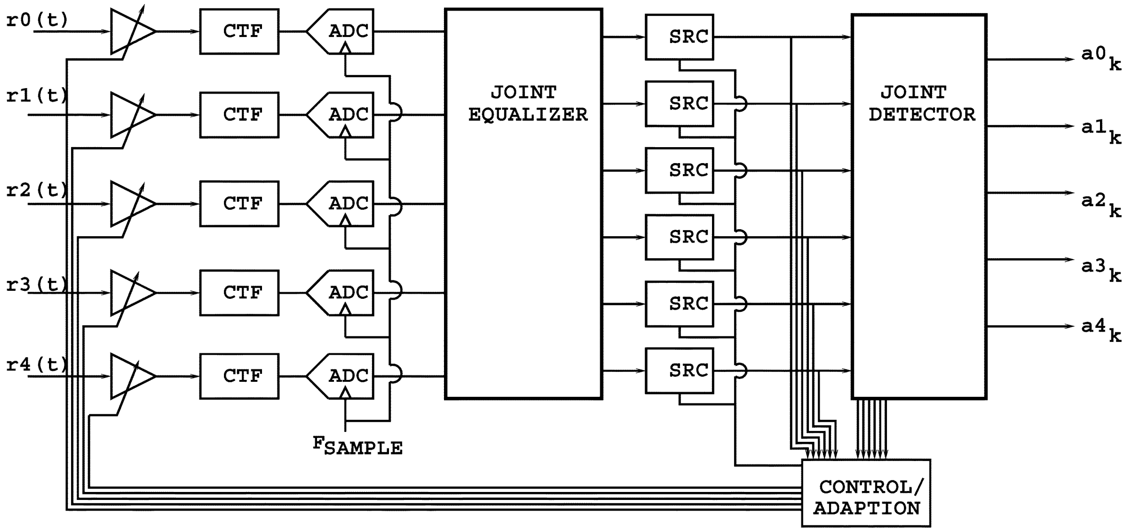

Fig. 1. Multitrack read channel.

For a multitrack system, the normal read channel functions such as automatic gain control and offset control are also required. In the case of nonlinear transducer readouts, some nonlinearity compensation may be applied after which the signal is low-pass filtered and sampled. These can all operate on each row independently and hence can be identical to existing one-dimensional (1-D) read channel implementations.

One function required specifically for the multitrack case is the alignment of the multitrack signals to account for the separation in space of the individual track transducers. This may be done explicitly as in [1], however, as it just requires a shift in time, it can be readily achieved as part of the equalization function.

Fig. 1 shows the basic multitrack read channel architecture for high-density storage. The continuous time filters (CTFs), analog-to-digital converters (ADCs), and sample rate converters (SRCs) for interpolated timing recovery can be conventional. The key blocks that require consideration are the joint equalizer and the joint detector. These are inherently more complex than the 1-D case as both must deal with intersymbol interference (ISI) of a fundamentally 2-D nature.

A. Joint Equalization

The requirement for joint equalization can be complex. Assuming that the equalization is required to shape the read signals to a predefined partial response, it is required to perform a 2-D convolution. In general, the number of equalizer outputs N_E may be different than the number of input rows. The output of the joint equalizer for row $s \in \{0, 1, \ldots, N_E - 1\}$ at time index k can be calculated as

$$y_s(k) = \sum_{r=0}^{N_R-1} \sum_{q=0}^{L-1} h_{r,q}^s \times x_r(k-q) \tag{1}$$

where the equalizer input from row $r \in \{0, 1, \ldots, N_R - 1\}$ is $x_r(k)$ and the equalizer coefficients for the output row s are $h_{r,q}^s$ with $q \in \{0, 1, \ldots, L-1\}$ and L being the span of the equalizer in the tangential axis. N_R is the number of input rows which is assumed to be equal to the number of read transducers.

In this case, the full joint equalizer requires LN_RN_E multiplications to generate N_E output samples in each time period. Each set of N_E output samples results in N_R decoded bits; therefore, the number of multiplications per decoded output bit (N_M) is

$$N_M = LN_E. \tag{2}$$

This compares to L multiplications per decoded bit for a single track format read channel. For significant ISI in the radial direction, the computational complexity of such a joint equalizer may be considerable.

1) Transform Domain Equalization: It is proposed in this paper to perform equalization using a transform filtering approach. As is well known, filtering in a transform domain can be efficient, as convolution in the time domain becomes multiplication in the transform domain. While initially a 2-D transform might be expected, it is important to note that the filter response $h_{r,q}^s$ in (1) is potentially different for each output (with s denoting the output row). This occurs because the read signal comes from an array of read transducers which may not all have an identical response. For example, in the optical case [1], the multiple read laser beams are generated from a single laser source using a diffraction grating and, thus, will not all be identical. Furthermore, any differential time alignment between the read transducers (due to their spatial separation) will require different filter responses for each output channel.

Each equalizer may appear as a 2-D equalizer, due to the 2-D nature of the data samples and filter coefficients. In the general sense of 2-D filters, the 2-D input is filtered to obtain a 2-D output. However, here, the 2-D input is filtered to obtain a 1-D output $y_s(k)$ for each of the N_E equalizers.

This has implications for considering transform approaches for the problem. A 2-D transform can map the input into a 2-D transform domain with filtering done in the transformed 2-D domain. Applying an inverse transform will involve using all points in the 2-D domain.

A more efficient approach using only 1-D transforms is now described, whereby a 2-D transform is not needed. The filter output $y_s(k)$ in (1) can be written as

$$y_s(k) = \sum_{r=0}^{N_R-1} z_s(r,k) \tag{3}$$

where $z_s(r,k) = \sum_{q=0}^{L-1} h^s_{r,q} x_r(k-q)$, which is a linear 1-D convolution. Each of these 1-D convolutions can be carried out using a transform technique. The summation of the convolutions in (3) can be carried out in the transform domain. This has the significant advantages of

- requiring N_R 1-D forward transforms, instead of one 2-D forward transform;
- requiring N_E 1-D inverse transforms, instead of N_E 2-D inverse transforms;
- using less multiplications in the 1-D transform domain than in the 2-D transform domain when used to implement linear convolution.

Fig. 2 shows an equalizer using the proposed 1-D transform approach using N points. The term $H^s(i)$ represents the N transformed coefficient values for row i of the equalizer s. The forward transform of the input rows only occurs once. The boxed section of Fig. 2 is repeated for each of the N_E equalizers.

In general, the number of computations required will depend on the details of the implementation; however, some insight can be obtained by considering some example calculations. Consider the use of N point transforms requiring $(N/2)\log_2(N)$ multiplications (assuming N is a power of 2 and using fast transform methods [6]). An equalizer support region of length L in the tangential direction with all N_R rows of data in the radial directions is assumed.

The transform domain equalizer requires N_R forward transforms and N_E inverse transforms. The convolution operation becomes multiplication in the transform domain and, hence, each block of outputs requires NN_RN_E multiplications.

Thus, for processing of a complete block, the total number of multiplications is

$$(N_R+N_E)(N/2)\log_2 N + NN_RN_E. \tag{4}$$

However, each block processed results in $N-L+1$ columns of data samples using overlap add or overlap save processing. Each column of data consists of N_E samples. Assuming that each N_E output samples results in N_R decoded bits, the number of multiplications per decoded bit N'_M is

$$N'_M = \frac{(N_R+N_E)(N/2)\log_2(N) + NN_RN_E}{N_R(N-L+1)}. \tag{5}$$

For example, consider the case of $L=9, N=16, N_R=7$, and $N_E=8$. In this case, the time domain equalizer requires 72 multiplications per decoded bit, the transform domain equalizer requires less than 25 multiplications per decoded bit, while a 1-D system requires nine multiplications. Taking $N=64$ would result in less than 17 multiplications per decoded bit although with increased latency. In practice, the multiplications in the transform domain equalizer will be more complex than those of the time domain equalizer. However, the forward and reverse transforms use constant coefficient multiplication which sometimes can be judiciously chosen to reduce complexities.

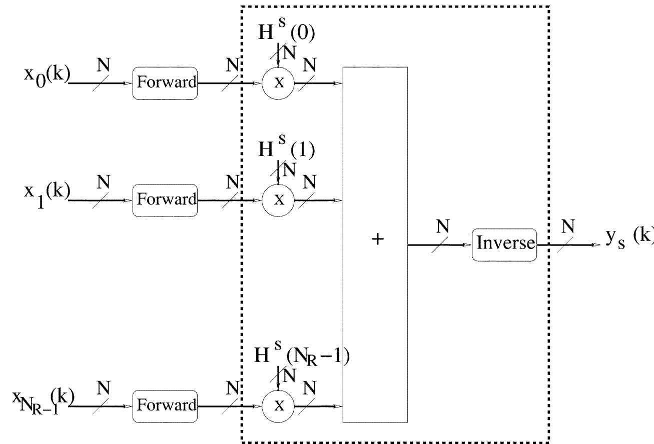

Fig. 2. Single equalizer using transform approach.

While the discrete Fourier transform may be considered for the transforms, it may not be the most efficient. For example, Section III-A shows that number theoretic transforms (NTTs) may be used effectively.

While the joint equalizer is more complex, the use of the transform domain equalizer brings the computational complexity significantly down to a level that should be feasible with near future integrated circuit (IC) processes. Overall, a complete detailed very large scale integration (VLSI) implementation would be required to assess the exact complexity; however, the transform domain equalizer has the potential for a significant reduction in the complexity of joint equalization in this application.

B. Joint Detection

The second area of complexity relates to the desirability of joint detection. While a number of methods have been developed, many rely on multiple iterations which often require the generation and passing of soft decision information [7].

An alternative is proposed here where the 2-D channel is modeled as a time-varying 1-D channel. In this case, the application of a tree search type algorithm such as the M-algorithm can provide good detection performance with reasonable implementation complexity.

The optimum detection of data in the presence of ISI can be achieved with maximum likelihood detection through the use of the Viterbi algorithm. In the case of 2-D ISI, the Viterbi algorithm can also be used but its implementation complexity can be prohibitive. Consider a 2-D extension of the classical PR4 response modeled with a side reading factor α. This response is denoted as the H_{2PR4} response with sampled impulse response of

$$H_{\mathrm{2PR4}} = [\alpha,1,\alpha]^T[1,0,-1] = \begin{bmatrix} \alpha & 0 & -\alpha \\ 1 & 0 & -1 \\ \alpha & 0 & -\alpha \end{bmatrix}. \tag{6}$$

This H_{2PR4} response has a 2-D support region of three rows in the radial direction and three samples in the tangential direction. Fig. 3 shows the basis of a state machine to model this

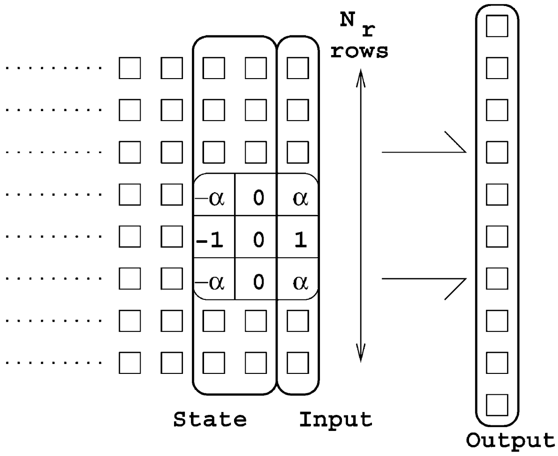

Fig. 3. Full 2-D state machine for 2-D channel.

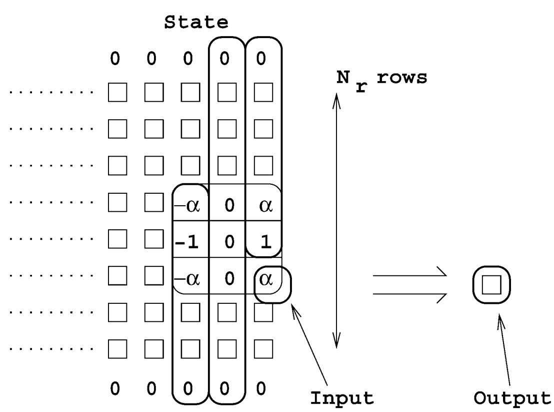

Fig. 4. Time-varying state machine for 2-D channel.

partial response channel. The state of the channel can be represented as two columns of binary data and one column representing the channel input. This information is sufficient to generate one column ($N_R + 2$ samples) of channel outputs.

However, this would result in a Viterbi detector with 2^{2N_R} states each with 2^{N_R} branches. For example, with a modest $N_R = 7$, this would result in excess of 16 384 states each with 128 branches. This is far too complex to implement at the type of data rates required. Hence, a reduction in the complexity of the detector is required. A reduced state version of a multitrack detector is reported in [8], though it is only considered for a two-track system.

C. M-Algorithm With Time-Varying Trellis

The first modification to make is to model the channel as a time-varying finite state machine which generates a single output at a time. Fig. 4 shows the state information required. In this case, the implicit zeros in the guard band are explicitly shown as they represent important state information. With this model, the channel state is represented with $2N_R + 2$ bits plus the four zero values.

With this channel model, the channel can in fact be viewed as a 1-D channel with a time-varying response. This can be written as a partial response in terms of a unit delay D as

$$H_{2PR4}(D) = \alpha + D + \alpha D^2 - \alpha D^{2N_R+4} - D^{2N_R+5} - \alpha D^{2N_R+6} \quad (7)$$

and noting the time-varying nature by which two zeros occur after every N_R channel input bits. A full Viterbi detector for this channel requires 2^{2N_R+2} states, which is still prohibitively large. However, each state has only two branches and the time-varying nature can be handled by simply extending the trellis with a single branch for each of the forced zeros. This time-varying property of the trellis is important as it represents key boundary information and failure to use this information results in a loss in detection performance.

In order to make such a detector structure practical, a reduction in the number of states is required. This can be achieved through the use of a tree search algorithm such as the M-algorithm [9]. In effect, this algorithm performs a tree search operation by retaining only M states at each stage in the trellis. Each of the M states are extended to produce $2M$ candidates of which the best M are retained. In this way, the complexity of the algorithm is related to the parameter M which can be chosen to trade off complexity against the detection performance.

D. Detector Complexity

The operation of the M-algorithm requires the storage of M path metrics and their corresponding survivor paths. With each new received symbol, these need to be expanded to $2M$ paths. The main complexity factor is then the need to select the best M metrics [10].

This can be achieved by sorting the $2M$ metrics and selecting the top M outputs from the sorter. The complexity of the sorting operation depends on its hardware implementation. One efficient method for hardware-based sorting is the use of Batcher's merge sort algorithm [11]. Using this algorithm, the number of compare select operations to sort $2M$ numbers is

$$2M\left(1 - \frac{1}{2M} + \frac{(\log_2 2M)(\log_2 2M - 1)}{4}\right). \quad (8)$$

Thus, the M-algorithm would require $2M$ path metric additions and a $2M$ sorting operation per decoded symbol. Fig. 5 shows the complexity in terms of adder/comparator operations per bit for the M-algorithm as a function of M. This is an upper bound as the M-algorithm only requires sorting to proceed until the best M metrics are identified. The best M metrics (and the worst M metrics) do not actually need to be ordered. The complexity of 8, 16, 32, and 64 state Viterbi detectors are also shown for comparison. If a 32-state Viterbi detector is considered feasible at present [12], then a detector with $M = 8$ should be readily implementable. Detectors with $M = 16$ should be feasible in the near future with the current rapid scaling of VLSI processes [12]. It may also be possible to implement the detector more efficiently based on the use of selection rather than full sorting as reported in [13].

The requirement for complete sorting inside the recursive loop of the algorithm means that the detector would need to be

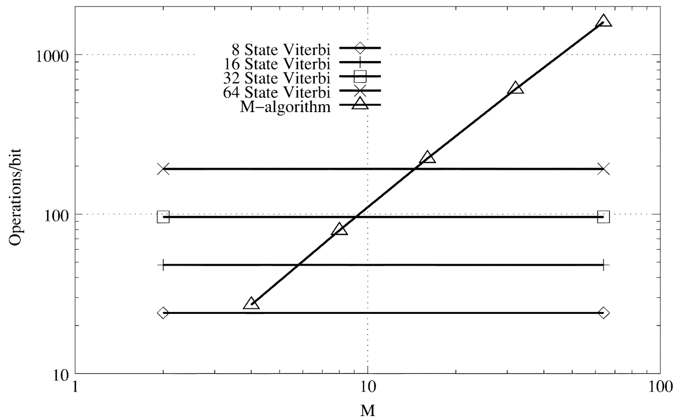

Fig. 5. Complexity of M-algorithm.

highly pipelined and operated in a windowed mode to achieve the data throughput required for data storage applications. The rapid scaling of VLSI capabilities should allow for the practical implementation of such a parallel decoding structure [14].

E. Detector Performance

The performance of this detection strategy is evaluated through simulations for the H_{2PR4} response and Fig. 6 shows the simulated performance of the H_{2PR4} detector for an eight-row multitrack as a function of M, the number of states retained by the M-algorithm. The simulations are performed with an ideal channel model with white Gaussian noise added. The figure shows the loss with respect to the maximum likelihood detector $(M = 2^{2N_R+2})$ for $\alpha \in \{0.25, 0.5, 1.0\}$. As would be expected, the loss reduces with increasing M. However, with M as low as 16, performance within 0.5 dB of the full detector can be achieved and the performance is relatively insensitive to the value of α. A more realistic channel model with data dependent media noise, or digitized data from a real head/media combination, would be required to fully verify the potential detection performance of such a target response in the magnetic recording application. However, Section III-B presents results using digitized data from a prototype multitrack optical player and disks.

III. Example System

While multitrack read heads for hard disk applications are not currently available, an experimental multitrack optical system has been developed based on a seven-track multirow format using a hexagonal lattice [1]. For this work, the prototype disk has a seven-row multitrack format with data written as the presence or absence of pits on a 165-nm hexagonal lattice. Such parameters correspond to a density of $\approx$1.4$\times$ the Blu-ray disk format or about 35 GB per 12-cm optical disk.

A parallel read channel has been developed for this system and its performance verified through the use of a software model with digitized data from the prototype optical player and disks. The fact that the data is stored on a hexagonal lattice rather than a rectangular lattice does not alter the fundamental receiver structure.

While a general response detector is proposed in [1], the use of a simplified partial response target is proposed here to achieve a more modest implementation complexity. Partial responses for hexagonal optical channels are considered in [15]. For the prototype optical disks available, the tribinary partial response was selected due to its good performance and ease of detector implementation. The tribinary partial response has the response

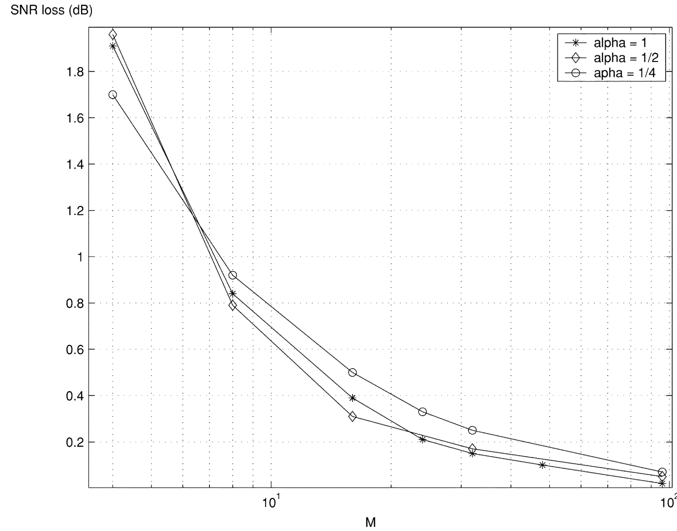

Fig. 6. Performance of H_{2PR4} target with eight rows.

$$\begin{bmatrix} & 1 & \\ 1 & & 1 \end{bmatrix} \tag{9}$$

on a hexagonal lattice. Fig. 7 shows the basic read channel architecture. The low-pass filtered and 1.25 $\times$ sampled input data is passed through a transform domain equalizer. Note that there are seven rows of data per multitrack and seven read spots in the optical pickup unit; however, there are eight equalizer outputs to account for the radial dispersion implied in the tribinary partial response. A single common timing phase error with eight sample rate converters generates eight symbol rate data samples. The equalizer handles fractional misalignment of the read spots.

The eight equalized samples are sent to a joint detector for decoding of the final bit decisions. However, preliminary threshold decisions are generated with low latency for control loop decision directed algorithms.

A. Transform Equalizer Using NTTs

While any transformation that provides the convolution property can be employed (such as the fast Fourier transform), the use of number theoretic transforms (NTTs) are considered here due to the moderate dynamic range and short transform lengths required. These are integer-based transforms with the forward and inverse transforms being defined as

$$X_n = \left(\sum_{k=0}^{N-1} x_k \beta^{nk} \right) \bmod M \tag{10}$$

and

$$x_k = \left(N^{-1} \sum_{n=0}^{N-1} X_n \beta^{-nk} \right) \bmod M \tag{11}$$

with the values for the transform length N, the system modulus M, and the element β all carefully chosen to ensure existence

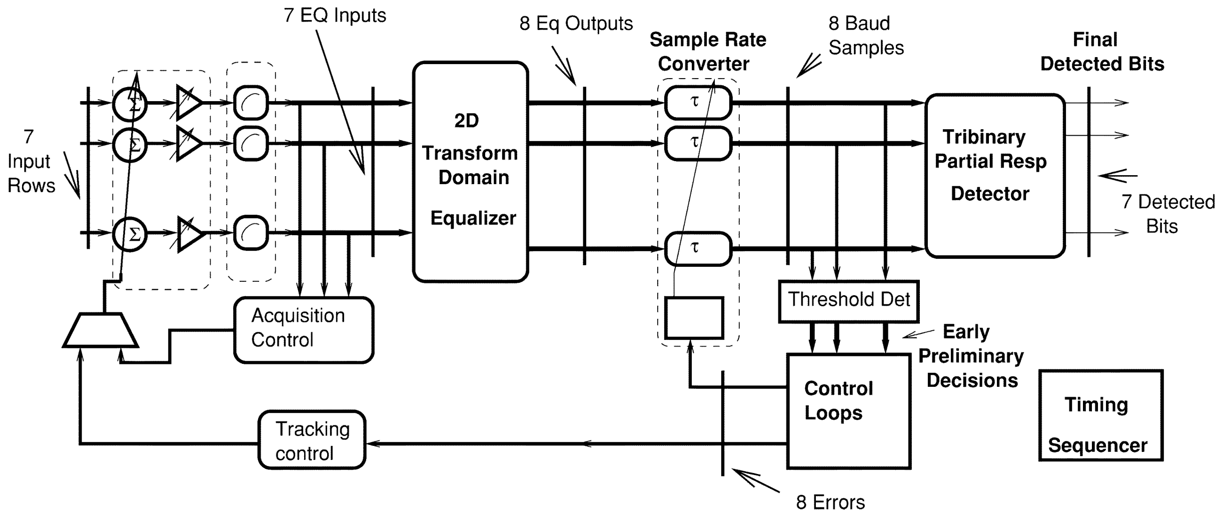

Fig. 7. Multitrack optical channel architecture.

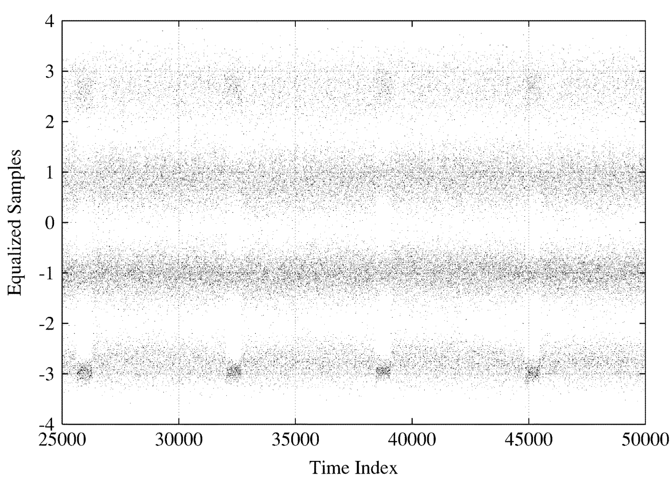

Fig. 8. Equalized data from prototype optics and disk.

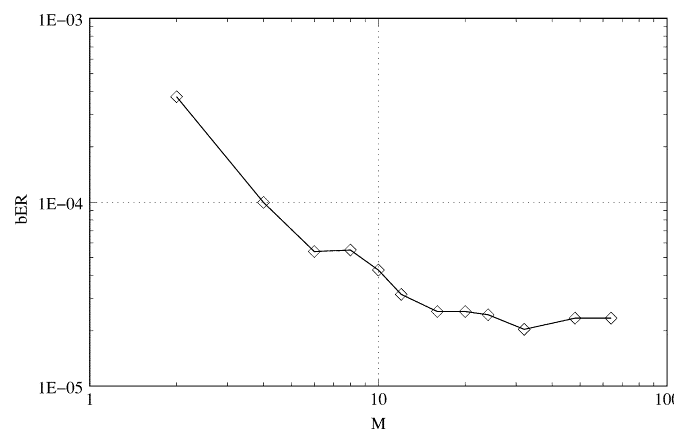

Fig. 9. BER performance versus detector parameter M.

of the transform. More detail on the use of such transforms can be found in [6], [16], and [17].

For the optical channel application, M is chosen to be 135 439 giving a 17-bit dynamic range. This value is a composite number equal to $17 \times 31 \times 257$, thereby allowing the computations to be performed in three parallel channels with the outputs combined using the Chinese remainder theorem (CRT). The moduli 17 and 257 are Fermat primes enabling simple modulo arithmetic and the use of the Cooley–Tukey algorithm for efficient implementation of transform length 2^4. The associated values for β are 3 and 2, respectively. The modulus 31 is a Mersenne prime and also facilitates easy modulo arithmetic, but requires the use of the Good–Thomas algorithm for efficient implementation of the transform length 3×5. β is 7 for this modulus. The modified overlap technique for NTTs proposed in [18] can be used to combine the results of the transform of length 16 with the transform of length 15.

In order to present a realistic comparison with the time domain approach, the complexity of both methods is measured by the number of full adders (FAs) required per output bit. This approach takes account of the different cost of multiplications in the transform domain. The time domain convolution requires 5742 FAs per output bit. This is based on using 8-bit data and coefficients. The transform domain approach using the moduli $\{17, 31, 257\}$ requires 1683 FAs per output bit, which is over a factor of 3 less. These complexity costs include all additions, constant multiplications and general multiplications. The single modulus $M = 2^{16} + 1 = 65\,537$ with $N = 16$ could be used instead of the three smaller moduli, but the larger cost of the multiplier (i.e., 16 bit $\times$ 16 bit) results in a cost of 4242 FAs per output bit. Using smaller moduli and the proposed technique in [18], a significant complexity reduction is achievable even with a short transform length.

Fig. 8 shows the sample rate converter output with digitized data from the prototype optics and disk with an NTT equalizer. The equalized outputs for each of the eight equalizer outputs are shown superimposed and the four-level equalized signal is clearly seen.

B. Tribinary Detector

The detector is implemented using the M-algorithm with a 1-D time-varying trellis modeling the 2-D channel partial response as described in Section II-C. The bit-error-rate (BER) performance using digitized data from prototype disks is shown in Fig. 9. For proper operation of higher level error control coding (usually based on interleaved Reed–Solomon codes), a raw BER of 1×10^{-4} is required. Thus, proper operation can be achieved for values of M as low as 6, which should be readily implementable. Essentially optimum detection (for that partial response) can be achieved for $M \approx 16$.

With $M = 6$, the number of additions and comparisons for a full sorting operation is less than 51, which is good considering that 2-D joint detection is being performed.

IV. Conclusion

In this paper, the use of a parallel read channel architecture is considered. Such a system can achieve higher data transfer rates and potentially higher data storage densities if joint equalization and detection can be practically achieved.

The use of a transform domain equalizer based on 1-D transforms is proposed and shown to achieve a high computational efficiency. While any transform with the convolution property could be chosen, it is shown that integer-based number theoretic transforms can perform well in this application.

Joint detection is accomplished through the use of the M-algorithm on a 1-D time-varying channel model of the 2-D target response. Such a detector can provide good performance with practical complexity.

A software model of a read channel incorporating these proposals is presented and shown to operate successfully on digitized data from a prototype multitrack optical disk system and achieve the desired BER.

The achievable increase in storage density for the magnetic recording channel is difficult to quantify and requires an in-depth study. In particular, the tradeoff between linear density and track pitch would need to be carefully evaluated. Furthermore, the actual channel response and the effect of read impairments in such a system would need to be considered. However, the optical system with the same optics as used in a 25-GB Blu-ray disk system showed proper operation at 35 GB. This was based on real digitized data from a prototype player and gives a good indication of the potential data density gains achievable.

Overall, it is shown that a parallel read channel architecture can be implemented with a computational complexity that should be feasible for implementation with near future IC processes.

Acknowledgment

This work is part of a European IST-Project called TwoDOS (IST-Project No. IST-2001-34168). All project members (Philips Research, University of Limerick, Eindhoven University of Technology, University of Lancaster, Philips ODTC, and HW Communications) are acknowledged for their valuable contributions to this work; in particular, the following: W. Coene, A. Immink, A. Hekstra, J. Bergmans, J. Riani, and S. van Beneden for discussions on the architecture. Also, thanks is due to A. van der Lee and C. Busch for providing digitized waveform from the prototype player and Philips ODTC and Sony Corp. for producing prototype disks.

The work of S. Tosi was supported by the Science Foundation Ireland (SFI) under Grant 02/IN.1/I40.

References

[1] A. H. J. Immink, W. M. J. Coene, A. M. Vander Lee, C. Busch, A. P. Hekstra, J. W. M. Bergmans, J. Riani, S. J. L. V. Beneden, and T. Conway, "Signal processing and coding for two-dimensional optical storage," in *Proc. GlobeCom*, 2003, GC16_a2.

[2] E. Soljanin and C. N. Georghiades, "Multihead detection for multitrack recording channels," *IEEE Trans. Inform. Theory*, vol. 44, no. 7, pp. 2988–2997, Nov. 1998.

[3] P. J. Davey, T. Donnelly, D. J. Mapps, and N. Darragh, "Two-dimensional coding for a multi-track recording system to combat inter-track interference," *IEEE Trans. Magn.*, vol. 34, no. 4, pp. 1949–1951, Jul. 1998.

[4] S. Gopalaswamy and B. V. K. V. Kumar, "Multi-channel equalization for PWM channels in magneto-optical recording," *IEEE Trans. Magn.*, vol. 33, no. 5, pp. 3265–3267, Sep. 1997.

[5] P. A. Voois and J. M. Cioffi, "Multichannel signal processing for multiple-head digital magnetic recording," *IEEE Trans. Magn.*, vol. 30, no. 6, pp. 5100–5114, Nov. 1994.

[6] R. E. Blahut, *Fast Algorithms for Digital Signal Processing*. Reading, MA: Addison-Wesley, 1985.

[7] X. Chen and K. M. Chugg, "Near-optimal data detection for two-dimensional ISI/AWGN channels using concatenated modeling and iterative algorithms," in *IEEE Int. Conf. Communications*, 1998, pp. 952–956.

[8] E. Kurtas, J. G. Proakis, and M. Salehi, "Reduced complexity maximum likelihood sequence estimation for multitrack high-density magnetic recording channels," *IEEE Trans. Magn.*, vol. 35, no. 4, pp. 2187–2193, Jul. 1999.

[9] G. J. Pottie and D. P. Taylor, "A comparison of reduced complexity decoding algorithms for trellis codes," *IEEE J. Sel. Areas Commun.*, vol. 7, no. 9, pp. 1369– 1380, Dec. 1989.

[10] P. A. Bengough and S. J. Simmons, "Sorting-based VLSI architectures for the M-algorithm and T-algorithm trellis decoders," *IEEE Trans. Commun.*, vol. 43, no. 234, pp. 514–522, Feb./Mar./Apr. 1995.

[11] K. E. Batcher, "Sorting networks and their applications," in *AFIPS Proc. Spring Joint Comput. Conf.*, 1968, pp. 307–314.

[12] R. Lynch, E. M. Kurtas, A. Kunznetsov, E. Yeo, and B. Nikolic, "The search for a practical iterative detector for magnetic recording," *IEEE Trans. Magn.*, vol. 40, no. 1, pp. 213–218, Jan. 2004.

[13] S. J. Simmons, "A nonsorting VLSI structure for implementing the (M, L) algorithm," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 6, pp. 538–546, Apr. 1988.

[14] G. Fettweis and H. Meyr, "High-speed parallel Viterbi decoding: Algorithm and VLSI-architecture," *IEEE Commun. Mag.*, vol. 29, no. 5, pp. 46–55, May 1991.

[15] T. Conway, "Partial response detection for multi-track optical data storage," *IEEE Trans. Commun.*, submitted for publication.

[16] J. H. McClellan and C. M. Rader, *Number Theory in Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[17] G. A. Jullien, "Number theoretic techniques in digital signal processing," in *Advances in Electronics and Electron Physics*. New York: Academic, 1991, vol. 80, ch. 2, pp. 63–163.

[18] R. Conway, "Modified overlap save technique for NTTs," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, submitted for publication.

Manuscript received September 28, 2004; revised January 29, 2005.