

Viterbi Detector for Non-Markov Recording Channels

Sam Gratrix¹, Robert Jackson², Tom Parnell², and Oleg Zaboronski²

¹Arithmatica LTD, Warwick CV35 7LS, U.K.

²Arithmatica LTD and Department of Mathematics, University of Warwick, Coventry CV4 7AL, U.K.

Channel noise in modern perpendicular channels is well approximated by a linear jitter model. In this paper, we propose a new type of Viterbi detector matched to such a channel model. The differentiating feature of this detector is recursive evaluation of whitened noise strengths for survivor paths using the on-the-fly Cholesky factorization of path-dependent correlation matrices. We study the performance of the detector using the simplest example of low-density channels with intersymbol interference length equal to three. We find that at BER = 10^{-4} , the new detector outperforms the data-dependent AR-detector of the comparable complexity by about 0.9 dB.

Index Terms—Cholesky factorization, data-dependent channel noise, perpendicular magnetic recording, recursive whitening.

I. INTRODUCTION

THE past two decades have seen a dramatic increase in the density of data stored on magnetic media. For example, in the period between 1994 and 2004, the amount of information stored per unit area of magnetic media has increased by approximately two orders of magnitude [1]. Such a spectacular capacity improvement of magnetic hard drives would not be possible without the ever increasing sophistication of signal detection schemes used to extract user data from the noisy analog signal produced by the read head. Over the years, the density increase brought about the effects of intersymbol interference in the 1980s, channel noise correlations in 1990s, data dependence and non-Gaussianity of noise in the beginning of this century. Accordingly, the data detection schemes employed by read channels has evolved from peak detection to partial-response maximum-likelihood detection based on the Viterbi algorithm to noise-predictive detection schemes to data-dependent Gauss–Markov detectors; see [2] for a review. The spectacular success enjoyed by read channel technology up to date depended, therefore, on the ability of read channel designers to build high-throughput data detectors matched to increasingly complex probabilistic models of channel noise while adhering to strict area and power requirements imposed on the read channel.

Today, the necessity to match data detectors to the realistic channel model is more relevant than ever: the disk drive industry is currently switching to iterative error correction schemes based on low-density parity check codes [3]. As a result, modern data detectors are required to output the estimates of the *probabilities*¹ of written data bits rather than the estimates of the bits themselves [1]. Clearly, the quality of these probability estimates is dependent on the quality of the probabilistic model used within the detector.

We therefore arrive at the fundamental contradiction of read channel design: on the one hand, there is a requirement to increase the complexity of data detectors as the complexity of channel noise increases with stored data density. On the other

hand, there is tremendous pressure to keep the power consumption and the die size of read channel low.

To illustrate the complexity penalty to data detectors due to complexity of channel noise, consider the Gauss–Markov channel model [4]. This is a de-facto standard model of correlated data-dependent channel noise which proved to be extremely accurate for today’s longitudinal and perpendicular channels. To our best knowledge, all current generation data detectors are based on the Gauss–Markov approximation of channel noise. According to this model, the channel noise sample at a given time is represented as the sum of white noise component and the linear combination of L previous noise samples. The strength of white noise component and the coefficients of the linear combination depend on a pattern of $D + 1$ bits including the current bit. The model is therefore characterized by two structural parameters—Markov length L and data-dependent length D . The class of detector matched to such a model is essentially a noise-predictive Viterbi detector with an L -tap whitening filter assigned to each branch of the trellis; see [2, Ch. 33] for details. As the Gauss–Markov model describes a data-dependent autoregressive random process, we will refer to the corresponding detectors as AR-detectors. It is easy to see the number of trellis states of the AR-detector 2^{L+D} . It is expected that both the Markov length L and D of the appropriate Gauss–Markov model must grow linearly with linear channel density; see Section IV for more details. Therefore, the size of the optimal AR-detector for the channel will grow exponentially with linear channel density.

Does this mean that in the future it will be necessary to sacrifice the performance of data detectors due to the complexity restrictions? In this paper, we use a simple linear model of perpendicular read channel to argue that the large values of L and D needed for an accurate description of channel noise simply reflect the non-Markov nature of the latter. We then derive a Viterbi-type detector for the non-Markov channel model and compare its performance against the performance of AR-detectors for various values of Markov and data-dependent lengths.

The idea of deriving the data detector from the first principles given the channel is certainly not new. As is well known, up to 90% of media noise in high-density perpendicular recording channels is due to transition jitter; see e.g., [2] and [5]. Therefore, it is reasonable to attempt creating a data-dependent *jitter* detector. These attempts have followed the following two directions so far: by discretizing the (continuous) set of all pos-

Digital Object Identifier 10.1109/TMAG.2007.912836

¹Log-likelihood ratios, to be exact.

sible values of position jitter, it is possible to create a trellis for joint jitter-data detection using the maximum likelihood principle, [6], [7]. The resulting trellis was referred to as a “time trellis” in [6]. Excellent performance results of such detectors have been reported (compared with AR-detectors for synthetic channel simulations). This, however, is not the direction we wish to pursue. First, the “timing trellis” approach requires the introduction of quantization parameter for the jitter distribution. This parameter does not have an obvious interpretation in terms of the original channel, but the number of states of the “timing” trellis diverges in the limit of zero quantization. Second, it is not clear to us how to modify the approach to account for other sources of data-dependent noise in the perpendicular channel, such as pulse (breathing) jitter and data-erasure noise. See [8] for a brief description of pulse jitter noise. Pulse jitter may account for just 10% of media noise, but it has a drastic effect on the performance of AR-detectors, as we have discovered in experiments using our industrial partners’ channel models.

There is an alternative approach to signal detection for jitter-dominated channels pioneered in [9]. It is based on the remark that even if 100% of channel noise is due to position jitter, the standard deviation of the latter is still much smaller than the width of a transition region. Therefore, position jitter can be well approximated by an additive data-dependent conditionally Gaussian noise with parameters directly related to the properties of the recording channel; see [2] and Section II for details of such a derivation. Such a model is often referred to as the first-order jitter model. This model is intrinsically non-Markov: noise is represented by a regressive (“moving average”), rather than autoregressive stochastic process. Moreover, as we will show in Section IV, a faithful representation of the first-order jitter in the limit of strong media noise in terms of the Gauss–Markov model requires $L = \infty$. In [9], a detector is derived for the first-order jitter noise model under an extra assumption that the correlation matrix of noise is approximately diagonal. For high recording densities, the correlations are strong and the diagonal assumption breaks down. In this paper, we derive the detector for the first-order jitter model under the assumption that the correlation matrix of noise is banded. This assumption is always satisfied for as long as the assumption of finite intersymbol interference length holds. It is also important to stress that the detector described in the paper is applicable to any channel noise characterized by a banded correlation matrix including pulse jitter and erasure noise.

The rest of the paper is organized as follows. In Section II, we reproduce the familiar derivation of the linear jitter model presented in [9], and [2, Ch. 33]. In Section III, we explain the recursive whitening algorithm of the derived noise process based on on-the-fly Cholesky decomposition. In Section IV, we discuss conditions upon which the Gauss–Markov model can be derived from the linear jitter model. We also present the results of jitter noise whitening experiments using the recursive and the AR whitening algorithms. In Section V, we show that the non-linear Cholesky recursion can be linearized if jitter noise has a tridiagonal correlation matrix. This linearization is based on the celebrated relation between composition of rational transformations of complex plane and multiplication of matrices. In Section VI, we describe the sequence detector matched to the linear jitter model–data dependent regressive Viterbi detector and present the simulation results for its error-correction per-

formance. The Appendix contains MATLAB programs used to obtain experimental results reported in the paper.

II. DERIVATION OF THE THE LINEAR JITTER MODEL

The ideal read-back waveform generated by the read channel with isolated impulse response function $h(t)$ is

$$r(t) = \sum_{k=-\infty}^{\infty} a_k h(t - Tk) \quad (1)$$

where T is the bit period, $a_k = (1/2)(x_k - x_{k-1})$ is the transition signature, and $x_k \in \{-1, 1\}$ is the magnetic orientation in response to the written data bit $b_k \in \{0, 1\}$.

For an ideal perpendicular channel, $h(t)$ is an odd function satisfying $h(t) \xrightarrow{t \rightarrow \pm\infty} \pm 1$. For definiteness, we will assume the error function shape of the impulse response

$$h(t) = \text{erf}(t/W) \quad (2)$$

where W is the transition half-width. A computation shows that $W = (PW50/\sqrt{\ln(16)})T$, where $PW50$ is normalized linear channel density and T is the sampling period. We will use a system of units such that $T = 1$.

Assume that we sample $r(t)$ at the points $t = T/2 + mT$, where $m \in \mathbf{Z}$. Assume also the boundary conditions $x_{\pm\infty} = -1$. Then

$$r(T/2 + mT) = \sum_{k \geq 0} (a_{m-k} - a_{m+k+1})h(T/2 + kT). \quad (3)$$

Using the fact that

$$\sum_{k \geq 0} a_{m-k} = x_m/2 - x_{-\infty}/2 = x_m/2 + 1/2$$

and

$$-\sum_{k \geq 0} a_{m+k+1} = x_m/2 - x_{\infty}/2 = x_m/2 + 1/2$$

(the “telescopic” sums), we find that the above expression can be rewritten as follows:

$$r(T/2 + mT) = \sum_{k \geq 0} (a_{m-k} - a_{m+k+1}) \times (h(T/2 + kT) - 1) + x_m + 1. \quad (4)$$

The terms under the summation sign on the right-hand side decay as k increases, $\lim_{k \rightarrow \infty} (h(T/2 + kT) - 1) = 0$. If

$$\frac{1 - h(3T/2)}{1 - h(T/2)} \ll 1 \quad (5)$$

it is sufficient to retain just the first ($k = 0$) term in the sum and obtain the shape of channel intersymbol interference (ISI) of length three

$$r(T/2 + mT) = 1 + \frac{gx_{m-1}}{2} + (1 - g)x_m + \frac{gx_{m+1}}{2} \quad (6)$$

where $g = 1 - h(T/2)$. Using expression (2) for the channel response, it is easy to check that the value of the left-hand side of (5) for $PW50 = 1.5$ is approximately 0.04. Therefore, for linear densities $PW50 = 1.5$ or lower, the ISI target (6) provides a good discrete model for the channel response.

Our next task is to derive the linear model for channel noise. To simplify the presentation, we neglect the effects of pulse jitter and consider a channel with electronic noise and position jitter only

$$r_m \equiv r(T/2 + mT) = \sum_{k=-\infty}^{\infty} a_k \times h\left((m-k)T + \sigma_j N_k^{(1)}\right) + \sigma_e N_m^{(2)} \quad (7)$$

where $\{N_k^{(1)}, N_k^{(2)}\}_{k \in \mathbb{Z}}$ are independent standard Gaussian variables; σ_j and σ_e are standard deviations of jitter noise and electronics noise correspondingly.

If $\sigma_j \ll W$, jitter noise can be accurately approximated by an additive conditionally Gaussian noise. Applying Taylor expansion to the channel response function, one gets

$$h(t - Tk + \sigma_j N_k^{(1)}) \approx h(t - Tk) + \dot{h}(t - Tk)\sigma_j N_k^{(1)}. \quad (8)$$

Such an approximation is accurate at the sampling point $t = T/2$ if

$$\mathbf{E} \left[\frac{h\left(T/2 + \sigma_j N_0^{(1)}\right) - h(T/2) - \dot{h}(T/2)\sigma_j N_0^{(1)}}{h(T/2)} \right] \ll 1 \quad (9)$$

where the average is taken over noise sequences. To check if (9) is satisfied, we need to know the value of σ_j for the realistic values of media-to-electronic noise mixture. This can be calculated within the linear noise model. The validity of the latter will therefore be verified in posterior.

Substituting expansion (8) into (7) and using the ISI target (6), we arrive at the desired linear channel model

$$r_m = \left(1 + \frac{gx_{m-1}}{2} + (1-g)x_m + \frac{gx_{m+1}}{2}\right) + \dot{h}(T/2)\sigma_j \left(a_m N_m^{(1)} + a_{m+1} N_{m+1}^{(1)}\right) + \sigma_e N_m^{(2)}. \quad (10)$$

The most important feature of the derived model is that the noise is *regressive* rather than *autoregressive*. As a result, the noise process is non-Markov. In particular, if jitter noise is strong compared with electronics noise, (10) cannot be well approximated by an autoregressive process with small Markov length; see Section IV for the quantification of this statement.

The parameter g of the channel model (10) is determined by pulse shape and channel density; see (6). For the error function pulse and $PW50 = 1.5$

$$g \approx 0.43. \quad (11)$$

Parameter σ_j can be either defined as the percentage of the pulsewidth or via the given fraction of total channel noise due to jitter. We will adopt the latter definition. In this case, notations can be simplified by redefining $\sigma_j \rightarrow \sigma_j/\dot{h}$. As a result, σ_j will refer from now on to the typical additive jitter noise strength rather than the typical position jitter

$$r_m = \left(1 + \frac{gx_{m-1}}{2} + (1-g)x_m + \frac{gx_{m+1}}{2}\right) + \sigma_j \left(a_m N_m^{(1)} + a_{m+1} N_{m+1}^{(1)}\right) + \sigma_e N_m^{(2)}. \quad (12)$$

Using (12) it is easy to compute the average signal power P_s , white noise power P_e , and jitter noise power P_j

$$P_s = (1-g)^2 + \frac{g^2}{2} \quad (13)$$

$$P_e = \sigma_e^2 \quad (14)$$

$$P_j = \sigma_j^2 \quad (15)$$

where we assumed that data sequence $\{\vec{x}\}$ consists of independent identically distributed uniform random variables sampled from the alphabet $\{-1, 1\}$. The parameters σ_e and σ_j of the model can be fixed by specifying signal-to-noise ratio SNR and the fraction α of channel noise power due to jitter noise

$$\frac{P_s}{P_e + P_j} = 10^{SNR/10}, \quad \frac{P_j}{P_j + P_e} = \alpha. \quad (16)$$

From (16) and (13), (14), (15), one can obtain expressions for σ_j and σ_e in terms of g , SNR and α

$$\sigma_e = \sqrt{(1-\alpha)P_s} 10^{-SNR/20} \quad (17)$$

$$\sigma_j = \sqrt{\alpha P_s} 10^{-SNR/20} \quad (18)$$

where signal power P_s is given by (13).

We are now ready to investigate the validity of the linear jitter model (10). Let us consider the following channel parameters: $PW50 = 1.5$, $\alpha = 0.9$. Evaluating the left-hand side of (9) numerically, we find

SNR, dB	Relative Taylor expansion error
10	-0.11
15	-0.071
20	-0.045

We conclude that for $PW50 = 1.5$, jitter mix 90% of jitter noise and the error function pulse, the nonlinear perpendicular channel model (7) is well approximated by (12) for $SNR \geq 10$ dB. This is the simplest jitter-dominated non-Markov channel model which we will use to illustrate the main ideas behind our jitter detector.

The MATLAB file generating a random data sequence and the corresponding noisy read-back according to (12) is presented in Appendix A.

III. WHITENING PROCESS FOR THE LINEAR JITTER MODEL

Channel noise corresponding to the linear model (12) is given by

$$n_m = \sigma_j \left(a_m N_m^{(1)} + a_{m-1} N_{m-1}^{(1)}\right) + \sigma_e N_m^{(2)} \quad (19)$$

where we relabeled noise variables $n_m \rightarrow n_{m+1}$ so that the model has a causal form. We would like to build a data detector based on the optimal whitening (equivalently, noise cancellation) algorithm for the noise process given in (19). The purpose of the current section is to construct such an algorithm.

Conditional on the data sequence \vec{x} , the noise process is correlated Gaussian with the tri-diagonal correlation matrix

$$\begin{aligned} R_k^{(0)}(\vec{x}) &\equiv \mathbf{E}(n_k^2|\vec{x}) = \sigma_e^2 + \sigma_j^2 \cdot (a_k^2 + a_{k-1}^2), \\ R_k^{(1)}(\vec{x}) &\equiv \mathbf{E}(n_k n_{k-1}|\vec{x}) = \sigma_j^2 a_{k-1}^2, \\ R_k^{(p)}(\vec{x}) &\equiv \mathbf{E}(n_k n_{k-p}|\vec{x}) = 0, \text{ for } p \geq 2. \end{aligned} \quad (20)$$

It is well known that such a process can be equivalently represented as the linear combination of two independent Gaussian processes using Cholesky decomposition, [10], [11]

$$n_k = \Sigma_k^{(0)}(\vec{x})\eta_k + \Sigma_k^{(1)}(\vec{x})\eta_{k-1} \quad (21)$$

where $\{\eta_k\}_{k \in \mathbf{Z}}$ is a sequence of i.i.d standard Gaussian variables ($N(0, 1)$). Coefficients $\Sigma_k^{(1)}(\vec{x})$ and $\Sigma_k^{(0)}(\vec{x})$ can be found from the requirement that the correlation matrix of the process (21) coincides with (20)

$$\begin{aligned} \Sigma_k^{(0)2} + \Sigma_k^{(1)2} &= R_k^{(0)}, \\ \Sigma_k^{(1)}\Sigma_{k-1}^{(0)} &= R_k^{(1)}. \end{aligned} \quad (22)$$

Let $\mathbf{R}(\vec{x})$ be the correlation matrix with matrix elements (20). Let $\mathbf{\Sigma}(\vec{x})$ be the upper triangular matrix elements with nonzero matrix elements $(\mathbf{\Sigma}(\vec{x}))_{kk} = \Sigma_k^{(0)}(\vec{x})$ and $(\mathbf{\Sigma}(\vec{x}))_{kk+1} = \Sigma_k^{(1)}(\vec{x})$. In matrix notation, relation (22) can be written as

$$\mathbf{R}(\vec{x}) = \mathbf{\Sigma}(\vec{x})\mathbf{\Sigma}^T(\vec{x}) \quad (23)$$

which states that the matrix $\mathbf{\Sigma}(\vec{x})$ is the Cholesky factor of data-dependent correlation matrix $\mathbf{R}(\vec{x})$.

Relation (22) can be rewritten as the nonlinear recursion relation for $\Sigma_k^{(0)}(\vec{x})$ and $\Sigma_k^{(1)}(\vec{x})$

$$\begin{aligned} \Sigma_k^{(1)}(\vec{x}) &= \frac{R_k^{(1)}(\vec{x})}{\Sigma_{k-1}^{(0)}(\vec{x})}, \\ \Sigma_k^{(0)2}(\vec{x}) &= R_k^{(0)}(\vec{x}) - \frac{R_k^{(1)2}(\vec{x})}{\Sigma_{k-1}^{(0)2}(\vec{x})}. \end{aligned} \quad (24)$$

Formula (24) can be used to find all Σ 's recursively, but care has to be exercised to avoid instabilities in the limit $\sigma_e \rightarrow 0$. The important property of this recursion is that its coefficients $R_k^{(1)}$ and $R_k^{(0)}$ depend only on the two most recent transitions signatures a_k and a_{k-1} ; see (20). This enables one to implement Σ -recursion using a 4-state trellis, as we will explain in Section V.

We are ready to discuss the whitening of the channel noise (19). It is clear from (21) that the variable $\xi_k = \Sigma_k^{(0)}(\vec{x})\eta_k$ is decoupled from all past noise samples, i.e., $\mathbf{E}(\xi_k n_{i-p} | \vec{x}) = 0$ for any $p > 0$. Therefore, ξ_k is the whitened variable. It has the minimal variance possible, equal to $v_k(\vec{x}) = \Sigma_k^{(0)2}(\vec{x})$. Substituting (24) into (21), we obtain the following recursion relations for the whitened variable $\xi_k(\vec{x})$ and its variance $v_k(\vec{x})$:

$$\xi_k(\vec{x}) = n_k - \frac{R_k^{(1)}(\vec{x})}{v_{k-1}(\vec{x})}\xi_{k-1}(\vec{x}) \quad (25)$$

$$v_k(\vec{x}) = R_k^{(0)}(\vec{x}) - \frac{R_k^{(1)2}(\vec{x})}{v_{k-1}(\vec{x})} \quad (26)$$

where n_k is the estimate of the current noise sample. Compared with the whitening formulae for data-dependent auto-regressive noise process, two features of (25) and (26) are worth noting: First, the whitening filter (25) is of IIR type, rather than FIR as is the case for the AR noise process, [4]. Second, the coefficients of the whitening filter are not predefined, but computed on the fly via the separate recursion (26).

In short, an AR noise process is defined using an IIR filter and is whitened using an FIR filter. In contrast, channel noise (19) is defined using an FIR filter and is whitened using an IIR filter. It has been already noted for data-independent noise-predictive Viterbi detectors that an FIR whitening filter can be replaced with an IIR whitening filter with a small number of taps, [12], [13]. The above calculation shows that data-dependent IIR noise-canceling filters appear naturally for channel models characterized by *banded* correlation matrices. Due to data-dependence, the coefficients of these whitening filters have to be computed on the fly using Cholesky factorization recursion (26).

The differences between the noise model (19) and the AR model are significant. Even though correlation coefficients $R^{(0)}$ and $R^{(1)}$ entering the above whitening recursions depend only on the two latest transitions and the latest noise sample n_k , the recursive variables ξ_k and v_k depend on *all* past noise samples and all bits of the data sequence up to time k . Using the terminology of Gauss–Markov models, channel model (19) is characterized by an infinite Markov length L and infinite pattern length D . In the next section, we will show that an adequate approximation of the whitening process using an AR whitening formula with small values of L and D is only possible if jitter noise is much weaker than electronic noise.

Appendix B contains the MATLAB implementation of the whitening routine (25), (26).

IV. COMPARISON WITH GAUSS–MARKOV WHITENING ALGORITHM

Unrolling the recursive formula (26), one gets a representation for the variance of whitened noise samples as the continued fraction

$$v_k(\vec{x}) = R_k^{(0)} - \frac{R_k^{(1)2}}{R_{k-1}^{(0)} - \frac{R_{k-1}^{(1)2}}{R_{k-2}^{(0)} - \dots}} \quad (27)$$

This infinite fraction can be well approximated by a finite truncation if $R^{(0)2} \gg R^{(1)2}$, which is true if

$$\sigma_j \ll \sigma_e \quad (28)$$

see (20). In other words, the electronic noise must be much stronger than jitter noise for the truncation approximation to hold.

If (28) is valid and, consequently, v_k is well-approximated by (27) truncated at the L th level for some $L \geq 0$, v_k will depend only on $2 + L$ latest transitions. A more elaborate calculation shows that the L th truncation of $v_k(\vec{x})$ can be rewritten as the ratio of determinants of $(L+1) \times (L+1)$ correlation matrices²

$$v_k(\vec{x}) \approx \frac{\det(E(n_{k-i}n_{k-j}|\vec{x})_{0 \leq i, j \leq L})}{\det(E(n_{k-l}n_{k-m}|\vec{x})_{1 \leq l, m \leq L})} \quad (29)$$

Therefore, the L th truncation of v_k is given by the standard AR model formula with Markov length L and data-dependent length $L+2$ derived in ([4]). The relative error of such a truncation is $(\sigma_e/\sigma_j)^{(2L+1)}$. Keeping the same precision, the IIR

²To derive this fact, one should compare (27) with the expression for the determinant of a tri-diagonal matrix expressed in terms of determinants of tri-diagonal matrices of a smaller size.

whitening filter (25) can be approximated by the $(L + 1)$ -tap FIR whitening filter

$$\xi_k = n_k - \frac{R_k^{(1)}}{v_{k-1}} \left(n_{k-2} - \frac{R_{k-1}^{(1)}}{v_{k-2}} n_{k-3} + \dots \right) \quad (30)$$

the coefficients of which are also given by the standard AR formula for Markov length L .

We conclude that the AR noise model can be systematically derived from the regressive noise model (19) as an expansion in the small parameter σ_e/σ_j . The problem is that for modern perpendicular recording channels this parameter is large rather than small: the typical contribution of jitter noise to the total noise power ranges from 70% to 90%, which implies $\sigma_j \gg \sigma_e$. Therefore, channel noise is not Markov and pattern dependence length is not finite. Consequently, AR whitening filters with small values of L and D are expected to be outperformed by the recursive whitening algorithm with (25), (26).³

This conclusion can be confirmed experimentally using the MATLAB file presented in Appendix B. We used linear jitter model (12) with $PW50 = 1.5$, $SNR = 10$ dB, and jitter-to-white noise power mixture $\alpha = 0.9$ to create the sequence (19) of noise samples of length $N = 10000$. We then whitened this sequence assuming the knowledge of the original data sequence in two different ways: using IIR regressive whitening (25), (26) and using FIR AR whitening with $L = 1$, $D = 3$. Note that the number of taps of the whitening filters is 2 in both cases. $D = 3$ is chosen to accommodate the dependence of AR whitened noise sample on the 4 most recent bits. We analyzed the following three noisy sequences numerically: the original noisy sequence, AR-whitened sequence, and finally the sequence whitened according to (25), (26). The measured variance of the latter is 0.7 dB less than the variance of the AR-whitened sequence and about 2.1 dB less than the variance of the original noise sequence. These numbers happen to be weakly dependent on SNR and fluctuate very slightly from experiment to experiment.

Fig. 1 illustrates another important property of the linear jitter channel: the infiniteness of the pattern length. To construct the red curve, we have created an array of all $L = 1$, $D = 3$ AR variances which occurred for a random data sequence of length $N = 10000$. We then ordered this array according to the magnitude of the variance and plotted the result as the function of the index of each variance in the ordered array. The result is a staircase with six levels. The patterns corresponding to each level is also shown. The fact that there are 6 levels is easy to explain: for $D = 3$ there are $2^4 = 16$ possible values of the AR variance. It follows from (29) with $L = 1$ and (20) that AR variances depend on the magnitude but not on the sign of transition signature. Therefore, in general, there are only 8 different values of the variance. In addition, if $a_{m-1} = 0$, the dependence of the AR variance on a_{m-2} drops out, which reduces the number of

³For the case of the channel with tri-diagonal correlation matrix we consider in this paper, AR whitening algorithm with large values of L and D performs much better than one expects basing on the value σ_e/σ_j : the fraction (27) is of a finite length N with probability, $1 - 2^{-N}$. As a result, an AR model with sufficiently large L , will predict variances incorrectly with exponentially small probability of error. The problem is that the corresponding data detector will be exponentially complex.

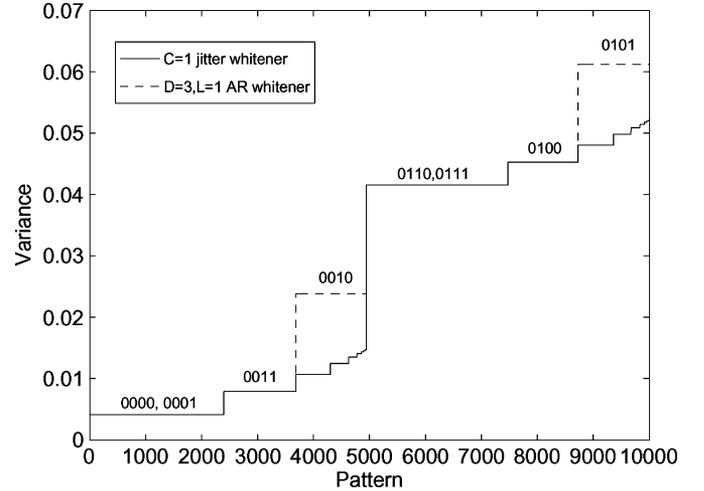


Fig. 1. The spectrum of variances of whitened variables.

distinct values to six. The length of each step is proportional to the probability of the occurrence of each pattern of length 4. In particular, degenerate steps corresponding to pairs of patterns are twice as long.

The black curve is constructed in the same way, but using all the values attained by the recursion (26) over 10000 iterations. The resulting plot suggests that the number of all possible values is infinite. This means that it is impossible to match the black curve using AR variance with finite pattern length D , as the latter would have at most 2^{D+1} values. Note that AR variance greatly exaggerates the value of whitened noise sample for the pattern 0101 which corresponds to the strongest transition noise. This is exactly the pattern for which detection errors occur with the greatest probability. Using recursion (25), (26) the strength of the whitened variable can be significantly reduced for this pattern compared with AR variance. This observation suggests that a detector based on whitening formula (25), (26) will significantly outperform the AR detector of a comparable complexity.

The MATLAB m-file used to generate curves of Fig. 1 and Fig. 2 is presented in Appendix B.

V. LINEAR REPRESENTATION OF Σ -RECURSION

The nonlinear recursion relation (26) can be linearized by linking the composition of (generalized) Möbius transformations to matrix multiplication. Let z be a complex variable. Consider the following transformation of the complex plane:

$$\phi_M(z) = \frac{az + b}{cz + d} \text{ for any } z \in \mathbf{C} \quad (31)$$

where M is a 2×2 complex matrix

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad (32)$$

such that $|c|^2 + |d|^2 \neq 0$. If matrix M is nondegenerate, the transformation (31) is referred to as the Möbius transformation, [14]. As we do not insist on the nondegeneracy of matrix M , we call (31) a generalized Möbius transformation. If matrices M_2, M_1 are such that the transformations $\phi_{M_2 \cdot M_1}, \phi_{M_2}, \phi_{M_1}$

are well defined, the following relation can be checked via an explicit calculation:

$$\phi_{M_2}(\phi_{M_1}(z)) = \phi_{M_2 M_1}(z). \quad (33)$$

The transformation (26) is the special case of the generalized Möbius transformation

$$v_k^{-1} = \phi_{M_k(\vec{x})}(v_{k-1}^{-1}) \quad (34)$$

where

$$4M_k(\vec{x}) = \begin{pmatrix} 0 & 1 \\ -R_k^{(1)2}(\vec{x}) & R_k^{(0)}(\vec{x}) \end{pmatrix}. \quad (35)$$

Using (33)

$$v_k^{-1} = \phi_{M_k}(v_{k-1}^{-1}) = \phi_{M_k}(\phi_{M_{k-1}}(v_{k-2}^{-1})) \quad (36)$$

$$= \phi_{M_k M_{k-1}}(v_{k-2}^{-1}) \dots \quad (37)$$

$$= \phi_{\mu_k}(v_0^{-1}) \quad (38)$$

where $v_0 = E(n_0^2|\vec{x})$ is the variance of the first noise sample coinciding with the variance of the first whitening sample; $\mu_k(\vec{x}) = M_k(\vec{x})M_{k-1}(\vec{x}) \dots M_1(\vec{x})$. Matrix $\mu_k(\vec{x})$ can be obtained from the *linear* matrix recursion

$$\begin{aligned} \mu_k &= M_k(\vec{x})\mu_{k-1}(\vec{x}), \quad k \geq 1, \\ \mu_0 &= \mathbf{I}_{2 \times 2}. \end{aligned} \quad (39)$$

Therefore, instead of running nonlinear recursion (26), the stability of which is difficult to analyze, one can run linear matrix recursion (39) and then calculate inverse variance v_k^{-1} via the Möbius transformation of v_0^{-1} ; see (38).

Given v_k^{-1} , the current whitening sample ξ_k and the branch metric BM_k can be computed using multiplications and a logarithmic lookup table only.

VI. DATA-DEPENDENT REGRESSIVE VITERBI DETECTOR

Data detector estimating the data sequence by whitening the regressive noise model (19) has four states. Essentially it is a data-dependent Viterbi sequence detector based matched to the regressive noise model (19). To make a contrast with more conventional AR data-dependent Viterbi detectors, we refer to it as data-dependent *regressive* Viterbi detector. It has a pair of whitening filters (25) and (26) attached to each branch. Each branch corresponds to three latest bits which allows one to determine the coefficients of the nonlinear $\Sigma^{(0)}$ -recursion and the tap of the IIR whitener. Each state of the detector is equipped with a path metric, the variance v and the estimate of the whitened noise sample ξ . Thus, it is possible to calculate the new whitened variable and its variance for each branch. The branch metric is given by the usual formula

$$BM = \frac{1}{v} \xi^2 + \ln(v). \quad (40)$$

Given branch metrics for each branch and path metrics for each state from which the branches originate, the usual ACS step is performed. The states inherit an updated path metric, the estimate of the whitened noise sample and its variance from the

survivors of the ACS step. In other words, the structure of the regressive detector is very similar to the that of noise predictive detector, which has the estimates of past noise samples and data patterns assigned to each state.⁴

We see that the detector is effectively building Cholesky factorization of all correlation matrices corresponding to survivor paths. Due to the fact that at each step of Viterbi recursion there are only 4 such paths, the factorization can be done without analyzing an exponential number of paths. One can perhaps notice some parallels between regressive Viterbi algorithm described above and arithmetic coding, [3].

The MATLAB file implementing regressive Viterbi detector is presented in Appendix C. Fig. 2 depicts the performance of the regressive detector, white noise detector, $L = 1, D = 2$ AR Viterbi detector with noise-predictive feedback loop and feedback-free $L = 3, D = 5$ AR detector. All three detectors assume the “natural target” (6) calculated for $PW50 = 1.5$. For our simulations, we chose jitter mix equal to 90%. The reason for this particular choice of the $L = 1, D = 2$ AR detector is that it has the same number of whitening filters as the regressive Viterbi detector and these filters have the same number of taps as the IIR filters used in the regressive detector. We observe that the 4-state regressive detector is about 0.9 dB better than the $L = 1, D = 2$ 4-state noise-predictive AR detector at $BER = 10^{-4}$ and over 2 dB better than the white noise Viterbi detector. We also see that the performance of the 4-state regressive detector matches the performance of a significantly more complex 32-state $L = 3, D = 5$ feedback-free AR detector. These results are consistent with the results of the whitening experiments for the data sequence described in Section II.⁵

VII. CONCLUSION

In this paper, we have derived a data detection algorithm for the simplest linear jitter channel model characterized by a tri-diagonal noise correlation matrix and a 3-tap ISI target. We have derived a recursive whitening algorithm based on the recursive Cholesky factorization of the regressive noise process describing linear jitter noise. We have compared the performance of the recursive whitening algorithm with $L = 1, D = 3$ whitening algorithm and found a 0.7 dB performance advantage of the former over the latter. We have found that for the tri-diagonal correlation matrices the nonlinear Cholesky decomposition recursion can be linearized using the correspondence between the composition of generalized Möbius transformations and multiplication of 2×2 complex matrices. Finally, we have described a 4-state data-dependent regressive Viterbi detector based on the recursive whitening algorithm and compared its performance with various AR detectors. Our main conclusion

⁴Let us stress that the described detector uses noise prediction and is therefore *suboptimal*. The corresponding optimal detector will have infinitely many states even for the simple linear noise model derived above.

⁵It must be said, however, that such a significant performance advantage over the white noise detector may be somewhat misleading: whereas the best performance of the regressive detector is already achieved for the “natural” channel target, the performance of the white noise detector can be significantly improved by equalizing the channel to a target corresponding to the maximal data-independent whitening of the channel noise.

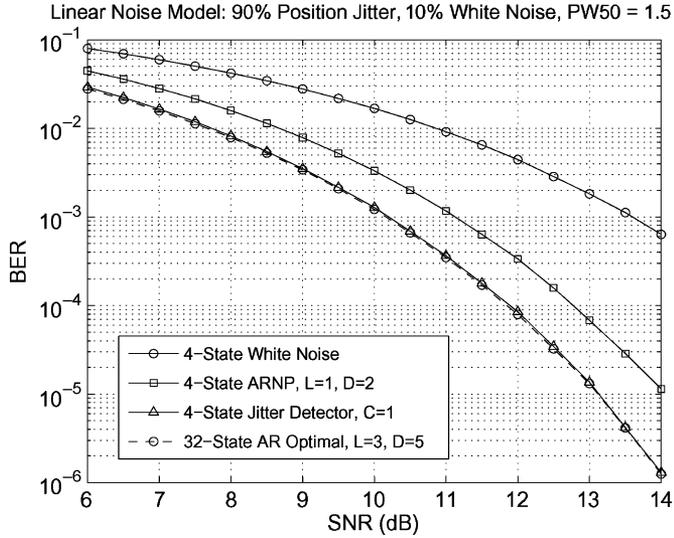


Fig. 2. The BER performance comparison for the regressive, AR, and white noise Viterbi detectors.

is that the regressive detector outperforms the comparable complexity $L = 1$, $D = 2$ AR detector with noise-predictive feedback by approximately 0.7 dB at $BER = 10^{-4}$. We also found that this 4-state detector matches the performance of the feedback-free 32-state $L = 3$, $D = 5$ AR detector.

In conclusion, we would like to stress that a modification of the recursive whitening algorithm discussed in this paper is applicable to noise models characterised by *banded* correlation matrices with an arbitrary number of nonzero diagonals. We believe that such correlation matrices provide a more adequate description of the channel noise than the AR noise model. The performance of the corresponding regressive detectors will be reported in a separate publication.

APPENDIX A

MATLAB IMPLEMENTATION OF THE NOISE MODEL

```
%Linear position jitter noise model
function [x,r,sw,sj,noise,G]...=
noise_model3 (N,SNR,rw,pw50);
%-----
%Arguments
%-----
%SNR—signal to noise ratio (dB);
%N—sector size (bits)
%rw—fraction of white noise power
%pw50—normalized linear density
% Calculate pulswidth:
w = pw50/sqrt(log(16));
% Calculate the target
g = 1 - erf(1/2/w); G = [g 2 * (1 - g) g];
% Signal power
```

```
%Relative power of jitter noiserj = 1 - rw;
ps = (1 - g)^2 + g^2/2;
%WN strength
sw = sqrt(rw * ps) * 10^(-SNR/20);
%JN strength
sj = sqrt(rj * ps) * 10^(-SNR/20);
%Generate data sequence
x = round(rand(1,N));
%Generate ideal sequence
I = conv(G,x); I = I(1:N);
%Generate transitions sequence
y = conv([1,-1],x); y = y(1:N);
%Generate noise sequence
temp = sj * y * randn(1,N);
noise = sw * randn(1,N) + temp + [0,temp(1:N-1)];
%Measure experimental SNR
SNR_e =
10 * log10((mean(I.^2)...-mean(I)^2)/mean(noise.^2))
%Generate noisy readback
r = I + noise;
end
```

APPENDIX B

MATLAB IMPLEMENTATION OF THE WHITENING ROUTINE

```
%Whitening filter
%Set channel parameters
N = 10 000; SNR = 10; rw = 0.1; pw50 = 1.5;
%Generate a noisy sector
function[x,r,sw,sj,noise]...
= noise_model3_1(N,SNR,rw,pw50);
%Generate transitions sequence
y = conv([1,-1],x); y = y(1:N);
%Maximal theoretical gain
delta = 10 * log10((sw^2 + sj^2)/(sw^2 + sj^2/2))
%Calculate correlation coefficients
R0 = sw^2 * ones(1,N)...+sj^2 * (y.^2 + [0,y(1:N-1)].^2);
R1 = sj^2 * [0,y(1:N-1)].^2;
%Compute variances of whitened variables
v = zeros(1,N); v_ar = zeros(1,N);
v(1) = R0(1); v_ar(1) = R0(1);
for i = 2 : N
    if y(i-1) == 0
        v(i) = R0(i);
    else
        v(i) = R0(i) - R1(i)^2/v(i-1);
    end
    v_ar(i) = (R0(i) * R0(i-1) - R1(i)^2)/R0(i-1);
end
```

```

end
delta_e = 10 * log10(mean(R0)/mean(v))
delta_e_ar = 10 * log10(mean(R0)/mean(v_ar))
%Compute whitened sequence
ksi = zeros(1,N); ksi_ar = ksi;
ksi(1) = noise(1); ksi_ar(1) = noise(1);
for
i = 2 : N
    if y(i - 1) == 0
        ksi(i) = noise(i);
    else
        ksi(i) = noise(i) - R1(i)/v(i - 1) * ksi(i - 1);
    end
    ksi_ar(i) = noise(i) - R1(i)/v(i - 1) * noise(i - 1);
end
delta_e2 = 10 * log10(mean(noise.^2)/mean(ksi.^2))
delta_e2_ar... = 10 * log10(mean(noise.^2)/mean(ksi_ar.^2))
subplot (2,1,1),
plot(1:1:N,noise, 'g-', 1:1:N,
ksi_ar, 'b-', 1:1:N,ksi, 'r-')
legend('Raw noise', 'AR-whitened noise',...
'Recursively whitened noise')
ylabel('Noise')
xlabel('Bit position')
    subplot(2,1,2),
    plot([1:N],sort(v), 'k', [1:N],sort(v_ar), 'r')
    legend('C = 1 jitter whitener',... 'D = 3,L =
1 AR whitener')
    ylabel('Variance')
    xlabel('Pattern')

```

APPENDIX C

MATLAB IMPLEMENTATION OF THE REGRESSIVE DETECTOR

```

%Jitter detector for
%c = 1 position jitter and white noise
function Err = jdpr1(G,x,r,sw,sj,N,WN);
%Choose the regime (WN = 1 => WNV, WN = 0 => JD)
%WN = 0;
%Generate noisy readback
%noise_model
%Fix the number of states
nst = 4;
%Generate survivors array
surv = zeros(nst,N);
%Generate path metrics array
PM = zeros(1,nst); PM_new = zeros(1,nst);

```

```

%Generate the array of
%past whitened noise samples
ksi = zeros(1,nst);
ksi_new = zeros(1,nst);
%Generate the array of variances
var = zeros(1,nst);
for k = 0 : (nst - 1)
%Transitions signature
temp = [0,mod(double(dec2bin(k,2)),2)];
var(k + 1) = sw^2 + sj^2 * (temp(3) - temp(2))^2;
end
var_new = zeros(1,nst);
%Generate the array for detected data
xbar = zeros(1,N);
%Start Viterbi Algorithm
for i = 1 : N
for st = 0 : (nst - 1)
%Determine states feeding into state j
st0 = (st - mod(st,2))/2;
st1 = st0 + 2;
%Determine competing binary patterns
st0_bin = [0,mod(double(dec2bin(st,2)),2)];
st1_bin = [1,mod(double(dec2bin(st,2)),2)];
%Determine transition signatures
%for each of the binary patterns
y0 = st0_bin(2:3) - st0_bin(1:2);
y1 = st1_bin(2:3) - st1_bin(1:2);
%Calculate correlation
%coefficients for each contender
R00 = sw^2 + sj^2 * (y0(2)^2 + y0(1)^2);
R10 = sj^2 * y0(1)^2;
R01 = sw^2 + sj^2 * (y1(2)^2 + y1(1)^2);
R11 = sj^2 * y1(1)^2;
%Calculate noise estimates
%for each contender
nse0 = r(i) - dot(G,st0_bin);
nse1 = r(i) - dot(G,st1_bin);
%Calculate variances for each contender
if y0(1) == 0
var0 = R00;
else
var0 = R00 - R10^2/var(st0 + 1);
end
if y1(1) == 0
var1 = R01;

```

```

else
var1 = R01 - R11^2/var(st1 + 1);
end
%Calculate whitening samples
%for each contender
if y0(1) == 0
ksi0 = nse0;
else
ksi0 = nse0 - R10/var(st0 + 1) * ksi(st0 + 1);
end
if y1(1) == 0
ksi1 = nse1;
else
ksi1 = nse1 - R11/var(st1 + 1) * ksi(st1 + 1);
end
%Calculate path metric for each contender
if WN == 0
PM0 = PM(st0 + 1) + ksi0^2/var0 + log(var0);
PM1 = PM(st1 + 1) + ksi1^2/var1 + log(var1);
else
PM0 = PM(st0 + 1) + nse0^2;
PM1 = PM(st1 + 1) + nse1^2;
end
%Find the survivor
if PM0 > PM1
surv(st + 1, i) = 1;
PM_new(st + 1) = PM1;
var_new(st + 1) = var1;
ksi_new(st + 1) = ksi1;
else
surv(st + 1, i) = 0;
PM_new(st + 1) = PM0;
var_new(st + 1) = var0;
ksi_new(st + 1) = ksi0;
end
end
PM = PM_new;
var = var_new;
ksi = ksi_new;
end
%Determine the estimated correct

```

```

%signal via global traceback.
[PM min, endstate] = min(PM);
xbar(N - 1 : N) = ...mod(double(dec2bin(endstate - 1, 2)), 2);
for k = N - 2 : -1 : 1
xbar(k) = surv(endstate, k + 2);
endstate = 2 * xbar(k) + xbar(k + 1) + 1;
end
Err = sum(mod(x(100 : N - 100) + xbar(100 : N - 100), 2))

```

ACKNOWLEDGMENT

The authors are grateful to S. Bates, D. Wright, and A. Vityaev for useful discussions. The authors are also grateful to anonymous referees for useful suggestions and a constructive critique.

REFERENCES

- [1] T. Christensen, "Hard disk drive read channels a must for perpendicular recording," Planet Analogue, May 20, 2004 [Online]. Available: <http://www.planetanalogue.com/showArticle?.articleID=20900003>
- [2] B. Vasik and E. Kurtas, *The Handbook of Coding and Signal Processing for Recording Systems*. Boca Raton, FL: CRC, 2004.
- [3] D. Mackay, *Information Theory, Inference, and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
- [4] A. Kavcic and J. M. Moura, "The Viterbi algorithm and Markov noise memory," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 291–301, Jan. 2000.
- [5] A. Takeo, Y. Takahashi, Y. Tanaka, K. Miura, H. Muraoka, and Y. Nakamura, "Precise noise characterization of perpendicular recording media," *J. Appl. Phys.*, vol. 87, no. 9, pp. 4987–4989, 2000.
- [6] J. R. Barry, A. Kavčić, S. W. McLaughlin, A. Nayak, and W. Zeng, "Iterative time recovery," *IEEE Signal Process. Mag.*, pp. 89–102, Jan. 2004.
- [7] R. Negi and X. Zhang, "Joint estimation for detection in transition-noise jitter dominant perpendicular recording channels," *J. Appl. Phys.*, vol. 99, no. 8, pp. 08K505–08K505-3, 2006.
- [8] J. Moon, "Signal-to-noise ratio definition for magnetic recording channels with transition noise," *IEEE Trans. Magn.*, vol. 36, no. 5, pp. 3881–3883, Sep. 2000.
- [9] W. M. Radich, "Computation of Branch Metric Values in a Data Detector," U.S. Patent Application Publication No. US 2004/0268208 A1, Dec. 30, 2004.
- [10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: Johns Hopkins Univ. Press, sec. 4.2.
- [11] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1996, sec. 6.7.
- [12] J. D. Coker, E. Eleftheriou, R. L. Galbraith, and W. Hirt, "Noise-predictive maximum likelihood (NPML) detection," *IEEE Trans. Magn.*, vol. 34, no. 1, pp. 110–117, Jan. 1998.
- [13] A. Duell-Hallen and C. Heegard, "Delayed decision-feedback sequence estimation," *IEEE Trans. Commun.*, vol. 37, no. 6, pp. 428–436, Jun. 1989.
- [14] S. G. Krantz, "Möbius transformations 6.2.2," in *Handbook of Complex Variables*. Boston, MA: Birkhäuser, 1999, p. 81.