This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2005 proceedings.

Optimal Precompensation for Partial Erasure and Nonlinear Transition Shift in Magnetic Recording using Dynamic Programming

Fabian Lim* and Aleksandar Kavčić†
*Department of Electrical and Computer Engineering, National University of Singapore
10 Kent Ridge Crescent, Singapore 119260
*Data Storage Institute, 5 Engineering Drive 1, Singapore 117608
Email : fabian_lim@dsi.a-star.edu.sg
†Division of Engineering and Applied Sciences, Harvard University
33 Oxford Street, Cambridge, MA 02138
Email : kavcic@deas.harvard.edu

***Abstract*— When precompensating for partial erasure (PE) together with non-linear transition shift (NLTS) in saturation magnetic recording, it is possible to offset written transitions from the sampling intervals to obtain a more 'linearized' readback signal. The premise of this paper is to show how to compute these optimal precompensation values using dynamic programming, given known functional forms of the PE and NLTS functions.**

I. Introduction

At high recording densities in magnetic recording, the readback signal suffers from nonlinearities. Currently, possibly the best known method to deal with the nonlinearities is to precompensate for them during data writing. The initial train of thought was to compensate only for nonlinear transition shift (NLTS). The amount of precompensation was obtained by various nonlinearity measurement methods such as methods using psuedorandom sequences [1] and harmonic elimination methods [2]. These methods were further extended to take into account partial erasure (PE) and head saturation effects to increase the accuracy of the measurements [3].

Past precompensation methods do not show how to optimally precompensate for nonlinearities. Most common approaches are to parameterize the precompensation value and use empirical optimization procedures to minimize the bit error rate. In this paper, we seek to derive optimal precompensation of NLTS and PE. The optimal solution allows us to formulate the limits of such a precompensation scheme, and to compare suboptimal solutions to the optimal one. A cost function based on the mean-square error (MSE) between the nonlinear signal and the linear signal is formulated[1]. We show how we can solve for the optimal precompensation that minimizes this MSE using dynamic programming.

Notation: In this paper, vectors are denoted as $\mathbf{A}_1^n = [A_1, A_2, ..., A_n]$. The probability of an event A is denoted by $\Pr\{A\}$, and the expected value of the random variable B is denoted as $E\{B\}$.

Organization: We set up the optimization problem in Section II, where we formally define the nonlinear model and the cost function. In Section III, we formulate our optimization problem into an infinite-horizon stochastic control problem, which is solved using dynamic programming. The algorithm that finds the optimal solution turns out to be computationally expensive, hence we go on to develop a suboptimal solution, which is explained in Section IV. In Section V, using computer simulations, we show that the suboptimal solution achieves near-optimal performance. Finally, we conclude in Section VI.

II. Optimization Problem

A. The Nonlinear Model

In the literature, models exist that capture nonlinear effects in magnetic recording. Some examples of these models are the signal-dependent autoregressive channel model [4], the transition zig-zag model [5], the position jitter width variation model [6], the microtrack model [7] and the Volterra series expansion model [8].

We formulate the simplest channel model that facilitates our purpose. We first define 4 important sequences b_n, c_n, Δ_n and x_n. The written data is in the form of a signed transition sequence $b_n \in \{2, -2, 0\}$ denoting positive, negative, and no transitions, respectively. We will abuse terminology and refer to b_n's as signal *bits*. Let T denote the symbol interval. The amount of NLTS affecting the nth transition is defined as $\Delta_n T$, and is precompensated by offsetting the nth write current transition by $c_n T$ from the sampling instant nT. Finally, the transition position $x_n T = \Delta_n T + c_n T$ is the net shift due to the precompensation and the NLTS effect. Δ_n, c_n and x_n lie in the continuous interval $(1, -1)$ (normalized by the symbol interval). We adopt the convention where $\Delta_n, c_n > 0$ constitutes a time advance and $\Delta_n, c_n < 0$ a time delay.

Normalized NLTS is defined as

$$\Delta_n \quad \triangleq \quad \Delta(\mathbf{b}_{n-L}^{n}, \mathbf{x}_{n-L}^{n-1}, c_n), \qquad (1)$$

This work was supported by a grant from the Information Storage Industry Consortium (INSIC).

[1] One should note though that minimizing the MSE does not gaurantee a minimum bit error rate.

i.e., it depends on the past L bits, the positions of the past L transitions $\mathbf{x}_{n-L}^{n-1}$ and the current precompensation c_n.

For a written transition $b_n \neq 0$, partial erasure (PE) is defined as the amplitude attenuation (normalized to 1)

$$\gamma_n \quad \triangleq \quad \gamma_n \left(\mathbf{b}_{n-1}^{n+1}, \mathbf{x}_{n-1}^{n+1} \right). \tag{2}$$

If there are no neighboring transitions to a transition at time n, the partial erasure amplitude is $\gamma_n = 1$. The amount of amplitude attenuation $\gamma_n \leq 1$ depends on the presence and distance of the immediate (preceding/following) transition from the transition at time n. Note that our PE model does not include pulse broadening effects.

The sampled, nonlinear readback signal is written as

$$\begin{aligned} z_n &= \sum_{k=-I_1}^{I_2} b_{n-k}\gamma_{n-k} h\left(kT + \Delta_{n-k}T + c_{n-k}T\right) + v_n \\ &= \sum_{k=-I_1}^{I_2} b_{n-k}\gamma_{n-k} h\left(kT + x_{n-k}T\right) + v_n, \end{aligned} \tag{3}$$

where $h(t)$ is the continuous-time transition response and v_n is additive white Gaussian noise (AWGN) with variance σ_v^2. For practical purposes, we assume $h(t)$ to be of finite support. The summation limits I_1 and I_2 are the anticausal and causal intersymbol interference (ISI) lengths, respectively.

B. Optimality Criterion

We define the squared error signal as the square of the difference between the nonlinear readback signal and the linear readback signal, given as

$$e_n^2 \quad \triangleq \quad \left(z_n - \sum_{k=-I_1}^{I_2} b_{n-k} h_k \right)^2, \tag{4}$$

where $h_k \triangleq h(t)|_{t=kT}$. The cost function $\mathcal{C}$ is the sum of all the mean-squared error (MSE) samples for *all* sampling instants,

$$\mathcal{C} \quad = \quad E\left\{ \sum_{n=-I_1}^{N-1+I_2} e_n^2 \right\}. \tag{5}$$

Here, N is the sector length. We will use the notation $\mathbf{c}_0^{*N-1}$ for the optimal sequence of precompensation values $\mathbf{c}_0^{N-1}$.

III. Dynamic Programming

We propose to re-formulate the optimization problem into an *infinite-horizon stochastic control* problem, and solve it using *dynamic programming* [9].

A. Finite-State Machine Model (FSM)

We manipulate the problem into a form suitable for defining a finite-state machine (FSM) through the following two steps.

1) The PE amplitude loss γ_n is dependent on the transition positions $\mathbf{x}_{n-1}^{n+1}$. Hence by utilizing equations (3) and (4), we can write e_n^2 as a function of a neighborhood of written bits $\mathbf{b}_{n-I_2-1}^{n+I_1+1}$, and transition positions $\mathbf{x}_{n-I_2-1}^{n+I_1+1}$

$$e_n^2 \quad = \quad e_n^2(\mathbf{b}_{n-I_2-1}^{n+I_1+1}, \mathbf{x}_{n-I_2-1}^{n+I_1+1}). \tag{6}$$

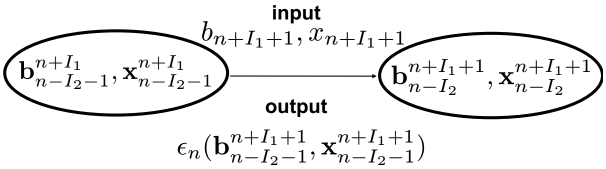

input
b_{n+I_1+1}, x_{n+I_1+1}
$\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1}$
$\mathbf{b}_{n-I_2}^{n+I_1+1}, \mathbf{x}_{n-I_2}^{n+I_1+1}$
output
$\epsilon_n(\mathbf{b}_{n-I_2-1}^{n+I_1+1}, \mathbf{x}_{n-I_2-1}^{n+I_1+1})$

Fig. 1. This Figure shows the Finite-State Machine (FSM) Model.

The problem now becomes to find the optimal transition position sequence $\mathbf{x}_0^{*N-1}$ that minimizes the cost function $\mathcal{C}$ in (5). Once $\mathbf{x}_0^{*N-1}$ is obtained, we can then compute $\mathbf{c}_0^{*N-1}$ using the knowledge of the NLTS function (1). Notice that we have expressed the squared-error in (6) as a function of a fixed length neighborhood of transition positions x_n. Had we chosen to express the squared-error as a function of precompensation values c_n, we could not confine the dependence to a fixed length window of precompensation values c_n.

2) We quantize the transition positions x_n to a finite number of values between -1 and 1.

With these modifications, we are now ready to define a finite-state machine (FSM) model.

Definition : [Finite-State Machine (FSM) Model]

State : The state is defined by $(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1})$.
Input : The input pair is $(b_{n+I_1+1}, x_{n+I_1+1})$.

a) The sequence b_n is random, with transition probability $\Pr\left\{ b_{n+I_1+1} \middle| \mathbf{b}_{n-I_2-1}^{n+I_1} \right\}$.
b) The sequence x_n is the *control* [9], meaning that the user can choose this value freely.

Output : The output is defined as

$$\begin{aligned} \epsilon_n &\triangleq \epsilon_n(\mathbf{b}_{n-I_2-1}^{n+I_1+1}, \mathbf{x}_{n-I_2-1}^{n+I_1+1}) \\ &= E\left\{ e_n^2(\mathbf{b}_{n-I_2-1}^{n+I_1+1}, \mathbf{x}_{n-I_2-1}^{n+I_1+1}) \middle| \mathbf{b}_{n-I_2-1}^{n+I_1+1} \right\}. \end{aligned} \tag{7}$$

This is the expectation of the squared-error sample e_n^2 over the noise sample v_n, given the signal pattern $\mathbf{b}_{n-I_2-1}^{n+I_1+1}$.

The transition probability is $\Pr\left\{ b_{n+I_1+1} \neq 0 \middle| \mathbf{b}_{n-I_2-1}^{n+I_1} \right\}$, and an absent transition occurs with probability $\Pr\left\{ b_{n+I_1+1} = 0 \middle| \mathbf{b}_{n-I_2-1}^{n+I_1} \right\}$. These values are assumed to be known a priori, and depend on the modulation applied to the signal bits. Figure 1 illustrates the FSM Model.

B. Infinite-Horizon Dynamic Programming

We observe that the output of the FSM ϵ_n, as given in equation (7), is determined by the current FSM state $(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1})$, the incoming bit b_{n+I_1+1}, and the control for the incoming bit x_{n+I_1+1}. At each state $(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1})$, we can observe the bit b_{n+I_1+1} and choose x_{n+I_1+1} to control the output ϵ_n. In dynamic programming terminology [9], the choice of control x_n is determined by the *policy* μ_n at time n, and we can write

$$x_{n+I_1+1} = \mu_{n+I_1+1}\left(\left(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1}\right), b_{n+I_1+1}\right).$$

Hence, the policy μ_{n+I_1+1} makes a decision on the value of x_{n+I_1+1}, given the state $(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1})$ and the incoming bit b_{n+I_1+1}. We minimize the MSE cost function $\mathcal{C}$ given in equation (5) over all policies $\mu_0^{N-1} = [\mu_0, \mu_1, ..., \mu_{N-1}]$.

We use the discounted future costs technique [9] to solve the precompensation problem. We freely choose the discounting factor α, where α must satisfy the condition $0 < \alpha < 1$. We denote the *optimum cost-to-go function* [9] at time n by $J_n^*(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1})$, which satisfies

$$\begin{aligned} & J_n^*(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1}) \\ &= \min_{\mu_{n+I_1+1}^{\infty}} E\left\{\sum_{k=n}^{\infty} \alpha^{k-n} e_k^2(\mathbf{b}_{k-I_2-1}^{k+I_1+1}, \mathbf{x}_{k-I_2-1}^{k+I_1+1}) \middle| \mathbf{b}_{n-I_2-1}^{n+I_1}\right\} \\ &= \min_{\mu_{n+I_1+1}^{\infty}} E\left\{\sum_{k=n}^{\infty} \alpha^{k-n} \epsilon_n(\mathbf{b}_{k-I_2-1}^{k+I_1+1}, \mathbf{x}_{k-I_2-1}^{k+I_1+1}) \middle| \mathbf{b}_{n-I_2-1}^{n+I_1}\right\}. \end{aligned} \tag{8}$$

The second equality follows from (7). The cost-to-go function $J_n^*(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1})$ gives the minimum future cost that can be incurred from the state $(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1})$. This minimum cost is achieved using the optimum policy $\mu_{n+I_1+1}^{*\infty}$.

The optimal policy $\mu_{n+I_1+1}^*$ is found by solving Bellman's equation [9]. Bellman's equation is derived from the optimum cost-to-go function (8), and is written as

$$\begin{aligned} & J_n^*(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1}) = \\ & \qquad \min_{\mu_{k+I_1+1}} E\Big\{\epsilon_n(\mathbf{b}_{n-I_2-1}^{n+I_1+1}, \mathbf{x}_{n-I_2-1}^{n+I_1+1}) \\ & \qquad\qquad +\alpha J_{n+1}^*(\mathbf{b}_{n-I_2}^{n+I_1+1}, \mathbf{x}_{n-I_2}^{n+I_1+1})\Big\}. \end{aligned} \tag{9}$$

We see from equation (9) that if we know the cost incurred by proceeding optimally from the future state $J_{n+1}^*(\mathbf{b}_{n-I_2}^{n+I_1+1}, \mathbf{x}_{n-I_2}^{n+I_1+1})$, we can solve for $\mu_{n+I_1+1}^*$ given some present state $(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1})$.

It is well known in the dynamic programming literature [9] that there are various methods available to solve Bellman's equation. We shall elaborate on one such method known as value iteration. We initialize the values $\hat{J}_0(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1}) = 0$ for all states. Then, we update $\hat{J}_i(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1})$ for all states $(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1})$ using the value iteration equation

$$\begin{aligned} & \hat{J}_i(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1}) = \\ & \qquad \min_{\mu_{k+I_1+1}} E\Big\{\epsilon_n(\mathbf{b}_{n-I_2-1}^{n+I_1+1}, \mathbf{x}_{n-I_2-1}^{n+I_1+1}) \\ & \qquad\qquad +\alpha \hat{J}_{i-1}(\mathbf{b}_{n-I_2}^{n+I_1+1}, \mathbf{x}_{n-I_2}^{n+I_1+1})\Big\}. \end{aligned} \tag{10}$$

As $i \to \infty$, the value of $\hat{J}_i(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1})$ will converge to $J_n^*(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1})$ for all states $(\mathbf{b}_{n-I_2-1}^{n+I_1}, \mathbf{x}_{n-I_2-1}^{n+I_1})$, and we can obtain the optimal policy $\mu_{n+I_1+1}^*$.

We will call the solution to Bellman's equation (9) the optimal solution, although this is a slight abuse of terminology. The real optimal solution is the one that minimizes the MSE defined in (5). We note that because of the exponential discounting factor α, the solution of (9) does not exactly minimize the MSE. However, as the number of bits in a sector N goes to ∞ and if we let α approach 1 (from below), the solution of (9) converges to the MSE solution. In that sense, it is correct to call the solution of (9) the optimal solution.

C. Extracting Optimal Precompensation Values

The solution of Bellman's equation will give us the optimal policy μ^*, which in turn delivers the optimal transition position sequence $\mathbf{x}_0^{*N-1}$ for any written bit sequence $\mathbf{b}_0^{N-1}$. The task now is to extract the optimal precompensation value sequence c_n^* from the optimal transition positions $\mathbf{x}_0^{*N-1}$. Assuming that the NLTS functional form (1) is known, we can simply solve

$$x_n^* \quad = \quad \Delta(\mathbf{b}_{n-L}^n, \mathbf{x}_{n-L}^{*n-1}, c_n) + c_n, \tag{11}$$

for $c_n = c_n^*$. One way of obtaining this solution is to compute the value c_n^* only after the sequence $\mathbf{x}_0^{*n}$ is obtained. This method is clearly not suited for real-time applications as we need to solve for the optimal precompensation value c_n^* at each time instance n. Another method for obtaining c_n^* is to draw it from a look-up table. However, the size of this look-up table grows exponentially with N. This is because x_n^* is a function of $\mathbf{b}_0^{*n}$, and therefore by (11), the precompensation value c_n^* is also a function of $\mathbf{b}_0^n$. Hence, since the sector size N is typically large, storing the optimal precompensation values c_n^* in a look-up table is practically unfeasible.

Since the extraction of c_n^* is computationally very complex, we investigate suboptimal methods in the next section. Nevertheless, the optimal solution derived in this section is very useful because, even though we cannot efficiently find c_n^*, we can find x_n^*, and thus provide a point of comparison for all other suboptimal methods.

IV. Suboptimal Solution

In the previous section, we argued that x_n^* depends on $\mathbf{b}_0^n$. In practice, we know that bits in the distant past do not really affect the present transition position x_n^*. This suggests that we can truncate the dependence on past bits to some reasonable memory length τ. We cannot prove mathematically that this property holds, but by making this assumption we greatly reduce the look-up table size. We will support our assumption with simulation results shown later in the paper.

Assumption 1: The optimal transition position x_n^* at time n is only dependent on the present signal bit b_n and τ past signal bits $\mathbf{b}_{n-\tau}^{n-1}$.

If Assumption 1 held, it would imply that for a given signal pattern $\mathbf{b}_0^n$, the optimal transition position x_n^* would not depend on the values of bits b_k for $k < n - \tau$. Consequently, the optimal precompensation value c_n^* would depend only on the bits $\mathbf{b}_{n-\tau-L}^n$, which vastly reduces the look-up table size.

The idea behind our sub-optimal solution is therefore to limit the size of the look-up table to a length of $\tau + L + 1$ input bits. Thereby, the precompensation value $c_n(\mathbf{b}_{n-\tau-L}^n)$ is forced to be time-invariant, and can hence be written as

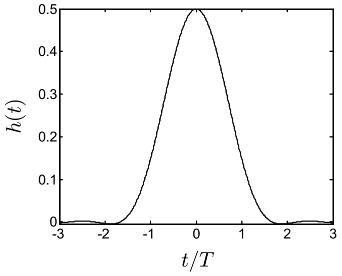


Fig. 2. Channel Transition Response $h(t)$.

$c_n(\mathbf{b}_{n-\tau-L}^n) = c(\mathbf{b}_{n-\tau-L}^n)$. The only remaining part is to determine the functional form of the function $c(\cdot)$. An easy way to solve this is to set $\mathbf{b}_{-\infty}^{-1} = \mathbf{0}$ and set

$$c(\mathbf{b}_0^{\tau+L}) \triangleq c^*_{\tau+L}(\mathbf{b}_0^{\tau+L}),$$

where $c^*_{\tau+L}(\mathbf{b}_0^{\tau+L})$ is the optimal precompensation computed in Section III-C for the bit pattern $\mathbf{b}_0^{\tau+L}$.

A. Error Propagation

For sake of completeness, we consider the possibility of error propagation if the following scenario happens. Let us consider that we make Assumption 1 and assume some value for τ, but in actuality the optimal precompensation sequence c_n^* depends on a *larger* past neighborhood of bits $\mathbf{b}_{n-\hat{\tau}-L}^n$, where $\hat{\tau} > \tau$. Now we want to examine what happens when we use precompensation values c'_n obtained by making Assumption 1 (suboptimal solution).

Let the sequence c_n^* denote the optimal precompensation sequence obtained using the methods described in Section III-C. Let us consider that we write bits $\mathbf{b}_{-\infty}^{\infty}$ twice, using precompensation sequences c_n^* and c'_n, respectively. Let us assume that $c_n^* \neq c'_n$ for $n \leq D$ and $c_n^* = c'_n$ for all $n > D$, where D is an arbitrarily chosen integer for the sake of the argument that follows. Intuition suggests that if $c_n^* = c'_n$ for $n > D$, we would expect that x_n^* should be equal to x'_n for n sufficiently larger than D. However this may not be the case (as we state below), i.e. the error made by choosing $c'_n \neq c_n^*$ for $n \leq D$ may propagate for indeces $n > D$.

As seen from equation (1), the value Δ_n depends on x_{n-1}. Using the relation $x_{n-1} = \Delta_{n-1} + c_{n-1}$, we see that x_{n-1} depends on Δ_{n-1} and c_{n-1}. If we continue this inductive argument, we conclude that Δ_n depends on $\mathbf{c}_0^n$. Now, since $\mathbf{c}_0^{*D} \neq \mathbf{c}_0^{\prime D}$, and since Δ_n depends on the entire vector $\mathbf{c}_0^n$, we conclude that $\Delta_n^* \neq \Delta'_n$. Consequently, we may get $x_n^* \neq x'_n$, for $n > D$, i.e. the error may propagate.

Fortunately, in a realistic scenario, it is observed that Δ_n depends on a short neighborhood of past precompensation values c_n. As our simulations show, at time instances $n >> D$, the transition position sequence x'_n will equal to the optimal transition position sequence x_n^* if τ is chosen adequately.

V. Computer Simulations

A. Channel characteristics

We assume the transition response of the channel $h(t)$ to be a truncated raised cosine pulse in the frequency domain, whose time-domain shape is shown in Figure 2. The causal and anticausal ISI lengths are chosen to be $I_2 = 1$ and $I_1 = 1$, respectively. This is of course an unrealistically short ISI length, but we purposely keep it short to make the tests simple. In real applications, if the ISI is large, we need to use equalizers to reduce the ISI, of course at the expense of coloring the noise.

To model NLTS, we use a functional form similar to equation (5) in [10]

$$\Delta(\mathbf{b}_{n-L}^n, \mathbf{x}_{n-L}^{n-1}, \mathbf{c}_{n-L}^n) = \sum_{j=1}^{L} \frac{-b_n b_{n-j} C_1}{4\left(j + x_{n-j} - c_n\right)^{C_2}}, \qquad (12)$$

where C_1 and C_2 are constants dependent on the physical parameters of the recording system. Here, we set them to $C_1 = 0.4$ and $C_2 = 2$. We set the NLTS past-bit dependence length to $L = 5$, see equation (1), because transitions more than 5 symbol intervals away do not seem to contribute much to NLTS under this model.

We chose the PE function (2) arbitrarily, but to resemble a plausible amplitude loss

$$\begin{aligned}\gamma(\mathbf{b}_{n-1}^{n+1}, \mathbf{x}_{n-1}^{n+1}) = \\ 1 - |b_{n-1}b_n/4| \left\{1 - \tanh[1 + x_{n-1} - x_n]\right\} \\ -|b_n b_{n+1}/4| \left\{1 - \tanh[1 + x_n - x_{n+1}]\right\}. \qquad (13)\end{aligned}$$

The term $|b_{n-1}b_n/4|$ equals 1 only if the pair (b_{n-1}, b_n) is non-zero (both are transitions), and equals 0 for all other cases. The similar reasoning applies to the other term $|b_n b_{n+1}/4|$. We need to point out that we must constrain $\gamma_n \geq 0$. We limit x_n to be quantized in intervals of 0.07, and in the range $-0.42 \leq x_n \leq 0.42$. Finally, we set the bit transition probabilities to $\Pr\left\{b_{n+I_1+1} \neq 0 | \mathbf{b}_{n-I_2-1}^{n+I_1}\right\} = \Pr\left\{b_{n+I_1+1} = 0 | \mathbf{b}_{n-I_2-1}^{n+I_1}\right\} = 0.5$.

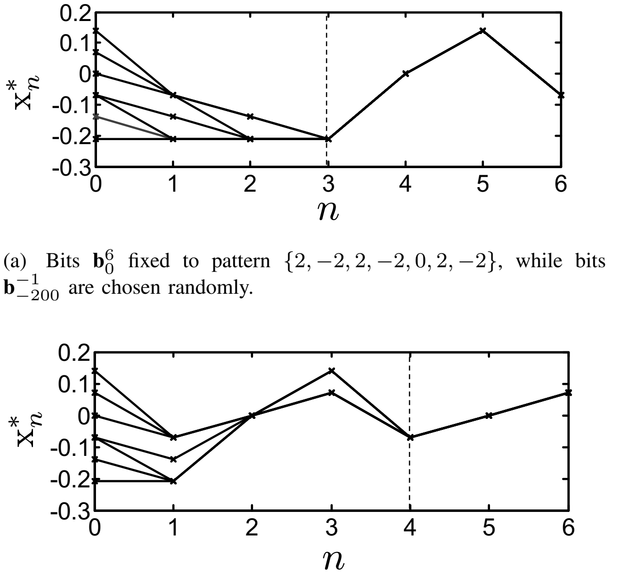
B. Testing the validity of Assumption 1

Figure 3 depicts the trajectories of the optimal sequences x_n^* computed for various bit patterns. When the discounting factor was chosen to be $\alpha = 0.9$, the Figure suggests that the memory of past signal bits in Assumption 1 should be chosen to as $\tau \geq 4$.

Next, we test how different choices for the memory constant τ affect the error between the optimal transition position x_n^* and the sub-optimal transition positions x'_n. Figure 4 reveals that for $\tau = 6$, the error is practically zero, and there is negligible error propagation.

C. Mean-squared error (MSE)

Finally, Figure 5 depicts the MSE performance of the sub-optimal solution. We evaluate the MSE cost function $\mathcal{C}$ in (5) using Monte Carlo methods for different choices for the memory constant τ. We observe that for a chosen value of α, the MSE error approaches the value $\mathcal{C}(\alpha)$, which is the MSE obtained using the optimal solution of Section III-B. For a sufficiently large memory constant τ and as α approaches 1



(a) Bits $\mathbf{b}_0^6$ fixed to pattern $\{2,-2,2,-2,0,2,-2\}$, while bits $\mathbf{b}_{-200}^{-1}$ are chosen randomly.

(b) Bits $\mathbf{b}_0^6$ fixed to pattern $\{2,-2,0,2,-2,0,2\}$, while bits $\mathbf{b}_{-200}^{-1}$ are chosen randomly.

Fig. 3. Here we fixed the pattern $\mathbf{b}_0^6$ and let the bits $\mathbf{b}_{-200}^{-1}$ be random. In Figure a), it takes 3 taps for all trajectories to converge, suggesting $\tau \geq 3$. In Figure b), it takes 4 taps for all trajectories to converge, suggesting $\tau \geq 4$. Taking the maximum of the two, we get $\tau \geq 4$.

in the limit, the MSE will approach the minimum MSE value indicated by the dotted line marked $\lim_{\alpha\to 1} \mathcal{C}(\alpha)$ in Figure 5.

VI. Conclusion

We showed, using dynamic programming, how to optimally precompensate for NLTS and PE, in order to minimize the MSE between the nonlinear readback signal and a linear one. The optimal solution serves as a reference point to which we can compare the performance of other suboptimal precompensation schemes.

We found the optimal solution to be complicated, and proposed a suboptimal solution that gives near optimal performance. The proposed suboptimal solution is found to have a much reduced complexity as compared to the optimal solution, and obtains near-optimal performance.

References

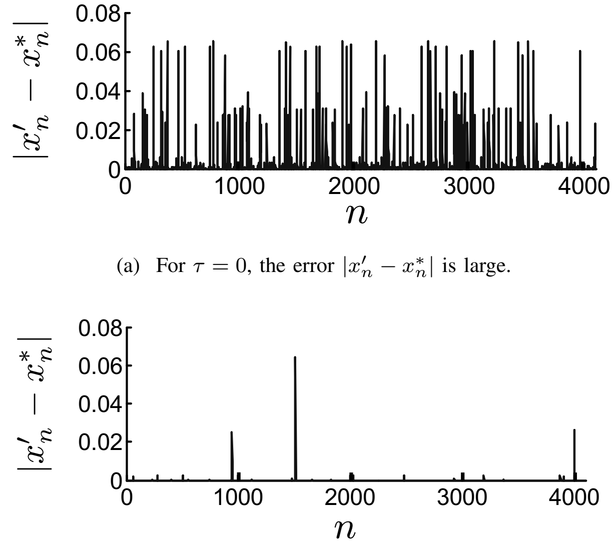
[1] R. W. D. Palmer, P. Ziperovich and T. Howell, “Identification of nonlinear write effects using psuedorandom sequences,” *IEEE Trans. on Magn.*, vol. 23, pp. 2377–2379, Sept. 1987.

[2] Y. Tang and C. Tsang, “A technique for measuring nonlinear bit shift,” *IEEE Trans. on Magn.*, vol. 27, pp. 5316–5318, Nov. 1991.

[3] F. et al., “Process for measuring nonlinear transition shift (NLTS) at high recording densities with a giant magnetoresistive (GMR) head,” Sept. 2004, patent No. 6,788,481.

[4] A. Kavcic and A. Patapoutian, “A signal-dependent autoregressive channel model,” *IEEE Trans. on Magn.*, vol. 35, pp. 2316–2318, Sept. 1999.

[5] A. Kavcic and J. Moura, “Statistical study of zig-zag transition boundaries in longitudinal digital magnetic recording,” *IEEE Trans. on Magn.*, vol. 33, pp. 4482–4491, Nov. 1997.



(a) For $\tau = 0$, the error $|x'_n - x^*_n|$ is large.

(b) For $\tau = 6$, the error $|x'_n - x^*_n|$ is practically zero, and there is negligible error propagation.

Fig. 4. We test how different values for the memory constant τ affect the error between the optimal transition positions x^*_n and the suboptimal transition positions x'_n.

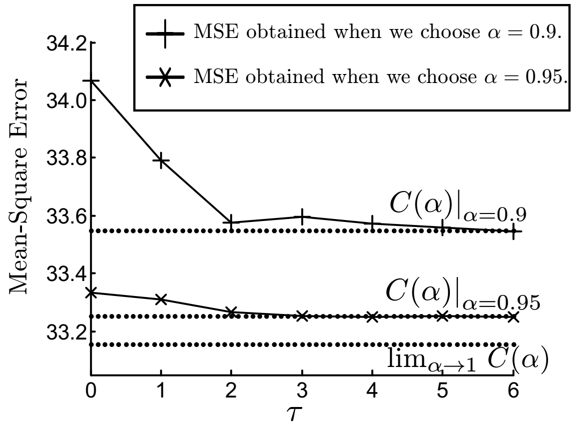


Fig. 5. The MSE performance of the suboptimal solution. When α was chosen to be $\alpha = 0.9$, the MSE approaches the optimal value $\mathcal{C}(\alpha)|_{\alpha=0.9}$ for $\tau \geq 6$. When α was chosen to be $\alpha = 0.95$, the MSE approaches the optimal value $\mathcal{C}(\alpha)|_{\alpha=0.95}$ for $\tau \geq 3$. As α approaches 1 in the limit, the MSE will approach the minimum MSE value indicated by $\lim_{\alpha\to 1} \mathcal{C}(\alpha)$. For comparison, if we set the precompensation value $c_n = -\Delta_n$ (i.e., aligning transitions with sample instances), the MSE is found to be 60.2.

[6] J. Moon, “Discrete-time modeling of transition-noise-dominant channels and study of detection performance,” *IEEE Trans. on Magn.*, vol. 27, pp. 4573–4578, Nov. 1991.

[7] J. Caroselli and J. Wolf, “Applications of a new simulation model for media noise limited magnetic recording channels,” *IEEE Trans. on Magn.*, vol. 32, pp. 3917–3919, Sept. 1996.

[8] R. Hermann, “Volterra modeling of digital magnetic saturation recording channels,” *IEEE Trans. on Magn.*, vol. 26, pp. 2125–2127, Sept. 1990.

[9] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Bellmont, MA: Athena Scientific, 2000.

[10] J. Caroselli, J. Fitzpatrick, and J. Wolf, “A simple model for transition interactions with the microtrack model,” in *Proc. IEEE International Conference on Communications (ICC’ 98)*, Atlanta, GA, June 1998, pp. 942–946.