Implementation and Analysis of the EEPR4 Channel

Amir Hesam Salavati
E-mail: hesam.salavati@epfl.ch
Supervisor: Prof. Amin Shokrollahi
E-mail: amin.shokrollahi@epfl.ch
Algorithmics Laboratory (ALGO), Ecole Polytechnique Federale de Lausanne (EPFL)

***Abstract*—Magnetic recording systems have a widespread use in storing our data and information. As the amount of data to be stored is constantly growing and the size of consumer electronic devices is decreasing, it is necessary to enhance the performance of magnetic recording systems and increase their recording density. Since the performance and recording density is limited due to noise and ISI, we have to design better error correcting codes, pre-coders, equalizers and detectors to improve such systems.**

In order to investigate the performance of the newly designed techniques and compare them with existing approaches, it is necessary to conduct large scale simulations. In this report, we discuss the details of a simulator to implement a magnetic recording system. In particular, we address different types of signaling methods, specially the EEPR4 signaling.

***Index Terms*—EEPR4 signaling, Inter symbol interference, Magnetic recording system, Partial response signaling**

I. Introduction

Magnetic recording systems play an important role in our everyday life. They make magnetic storage possible which has enabled us to store our digital data on the hard drives of our personal computers. In fact, magnetic recording systems are a crucial part of current computers and many other storage devices.

As consumer electronic devices are constantly becoming smaller and smaller and the amount of data to be stored is growing larger and larger, it is necessary to have smaller magnetic recording systems as well. In order to store larger amounts of data on smaller devices, we need to increase recording density of the storage system. However, the magnetic recording channel is not a memory-less channel which means we have to handle inter symbol interference as well as noise. In fact, the recording density is limited due to these factors, namely ISI and noise.

Therefore, to increase recording density, we need to design methods to overcome ISI issues or even take advantage of them. Partial Response (PR) signaling is a means of limiting ISI by introducing some controlled amount of correlation in input data to neutralized effects of memory in channel. As a result, PR signaling helps us increase recording density. There are actually a number of ways to increase recording density: with coding based on time or amplitude redundancy, or both. Since the Magnetic Recording Channel (MRC) is peak-amplitude limited, partial response signaling is used to increase coding densities via amplitude redundancy. The fact that Class IV of Partial Response (PR4) signal is closely matched to that of readback pulse decreases noise enhancement in the equalization. Therefore, further increase in recording density is achievable using PR4 signaling [17].

In this report, we describe a simulator for implementation of the magnetic recording channel. The overall goal of designing such a simulator is to test the performance of new error correction codes for magnetic recording systems and compare them with existing codes.

The rest of this report is organized as follows: in section II, we briefly review the structure of a magnetic recording system. In section III we consider the partial response signaling in more details. Section IV describes readback channel in a magnetic recording systems. In particular, we explain the discrete model we have used in the simulator. We will discuss the implementation details in section V. Finally, in section VI we conclude the report and suggest some future improvements for the simulator.

II. Magnetic Recording Systems

A magnetic recording system is composed of two main parts: the recorder and the reader. Recorder is responsible for transforming the electrical input into magnetic output and store them on the magnetic medium. Stored data is then read back by the reader using a magnetic head. During the readback process, the direction of magnetization is translated into electrical pulses so that it can be processed later. In the following two subsections, we introduce these two components in more details.

A. Recorder

Figure 1 illustrates the model for recording part of a magnetic recording system [21]. According to this model, data bits are first encoded according to an error correction code in the channel encoder block. Then, the encoded bits are again encoded, this time based on a specifically designed modulation codes such that synchronization and bit detection is facilitated during the read back operation. [15]. This goal is

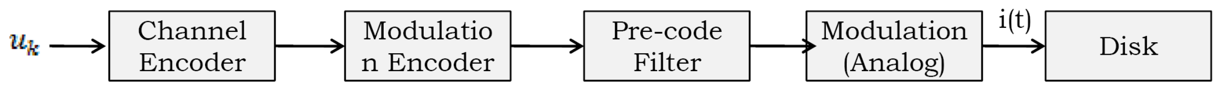

Fig. 1. The recorder part of a magnetic recording system

achieved by adding extra transitions to the modulated sequence and limit the number of consecutive ones or zeros. Moreover, modulation codes reduce the effect of minimum distance error events, which is the major parameter in the determining the performance of the detector. Furthermore, as the recording density increases, the role of modulation codes become more and more significant [2].

Modulation codes are also called Maximum Transition Run (MTR) codes. An MTR code is identified with two parameters: the maximum number consecutive transitions, t, and the maximum number of consecutive zeros, k. We show and MTR code with such properties by $MTR(t, k)$.

The modulated sequence is then pre-coded before being recorded on the disk in order to overcome the effects inter symbolic interference during the read back process. Roughly speaking, the pre-coder is an FIR filter whose transfer function is the inverse of that of the channel, i.e. if the channel transfer function is $H(D)$, then the transfer function of the pre-coder is $1/H(D)$. We will discuss the pre-coder more in a later section.

There are two standard methods for recording data in a magnetic recording system: Non Return to Zero (NRZ) and Non Return to Zero Inverse (NRZI). In the NRZ approach, one magnetization direction indicates zero and the reverse indicates one, which is analogous to the electrical case where voltage is used to determine a bit: a +1 voltage represents one and a -1 (or zero) voltage indicates zero. In contrast, the NRZI technique uses transition in magnetization direction (or similarly in voltage amplitude) to represent ”1” while zero is indicated by no change in the direction of the magnetization (or voltage sign).

Error propagation is the main disadvantage of NRZ method. When a bit is misread in the NRZ format, the error propagates through the rest of the sequence. This issue is remedied to some extent in NRZI approach since NRZI method acts as a pre-coder for the PR channel[1]. In other words, the pre-coder used in the PR channel to overcome ISI does exactly the same job as using NRZI method to store data on the magnetic recording medium [9].

Pre-coding and designing good error correction codes are crucial for magnetic recording systems since the readback channel is a partial response channel. Therefore, special pre-coding techniques are need to reduce the effects of ISI.

B. *Reader*

Figure 2 shows the reader part of the system. A magnetic head is used to read the stored data on the magnetic disk and translate magnetization direction into electrical voltage. The magnetic head introduces electrical and media noises to the readback sequence. The noisy sequence is then passed through a low pass filter (LPF) so that the unnecessary high frequency parts are omitted. The filtered data is the equalized using a FIR digital filter. FIR filter is optional an depends on the ways systems is designed. For instance, a possible option is to merge the equalizer into the maximum likelihood detector. The type of FIR filter depends on the signaling method and the ML detector used, i.e. PR4, Extended Partial Response Class IV (EPR4) or Enhanced Extended Partial Response Class IV (EEPR4), which we discuss in the next section.

After equalization, a maximum likelihood detector is used to find out the most likely transmitted sequence over the noisy channel. This sequence is then demodulated and decoded to find the best estimation of the stored bit, $\hat{u}_k$. If $\hat{u}_k \neq u_k$ then an error has occurred. Because magnetic recording channel is a partial response channel, we can apply general precoding methods widely used in communication systems to overcome error propagation in the read-back process. Kobayashi and Tang were the first to note the applications of partial response signaling to overcome ISI and increase recording densities [9].

Since then, several experimental works have shown the superiority of PR4 signaling in magnetic recording systems to traditional method of peak detection [19], [14]. The improvement in performance is mainly a result of the bandwidth efficiency of PR4 signaling, the similarity between the spectrum of a typical readback pulse and that of the PR4 signaling and maximum likelihood detection. At higher densities, however, the similarity is becomes less and, therefore, the performance of PR4 signaling decreases [17].

In the next section, we discuss the principles of the partial response signaling in more details. In a later section, we investigate the magnetic recording channels and its relation to partial response channels.

[1]Please note that by using the term ”PR channel” we refer the channels where we need PR signaling to overcome the effects of ISI.

Fig. 2. The reader part of a magnetic recording system

III. Partial Response Channel

In a normal channel, the current output of the channel depends only on the transmitted input plus some noise. In other words, adjacent transmitted bits do not affect each other.

In many real world systems, however, things are not that simple as the current output of the channel not only depends on the transmitted input bit, but also on the ”reminiscents” of the previously transmitted bits. In other words, adjacent transmitted bits affect each other and create correlations in the output [10]. This situation arises because the shape of response of the system is not a rectangular pulse as in the ideal case. It is usually a continuous non-rectangular pulse which has non-zero values outside the time slot corresponding to a single bit (see figure 3). As a result, we will have interference among symbols which makes retrieving the transmitted data more difficult. With some abuse of notations, we call the channels with ISI Partial Response Channels (PRC).

A PRC is fully described via its transfer function [9] which relates current output to transmitted bits up to this moment, as shown in equation (1).

$$Y(D) = X(D)H(D) \tag{1}$$

In equation (1), D is the delay operator, Y and X are the output and input of the channel, respectively, and given by $Y(D) = \sum_{k=0}^{\infty} y_k D^k$ and $X(D) = \sum_{k=0}^{\infty} x_k D^k$. Here, $\{x_1, \dots, x_k, \dots\}$ is the input sequence, $\{y_1, \dots, y_k, \dots\}$ is the output sequence. $H(D)$ is the channel's transfer function and is shown in equation (2).

$$H(D) = \sum_{i=0}^{N} h_i D^i \tag{2}$$

In the above equations, h_i's are integers depending on the model used for the channel. Note that while input is a binary sequence, the output $(\{y_1, \dots, y_k, \dots\})$ is non-binary. The number of output levels depends on the channel coefficients, h_i.

An important issue in a PRC is the model used to approximate the behavior of the channel. This model affects the pre-coder, the equalizer and the maximum likelihood decoder in the readback process. Generally speaking, there are four widely used models: PR4, EPR4, EEPR4 and generalized PR [13], [8]. Generalized PR channel is described by the equation $H(D) = (1-D)(1+D)(1+c_1D+c_2D^2+\dots)$. PR4, EPR4 and EEPR4 channels are described by the equations $H(D) = (1-D)(1+D)$, $H(D) = (1-D)(1+D)^2$ and $H(D) = (1-D)(1+D)^3$, respectively.

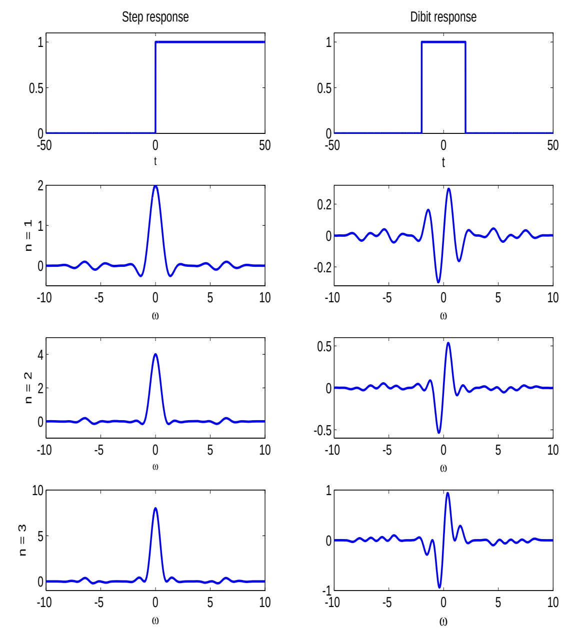

Fig. 3. One-sided and dibit response for PR4, EPR4 and EEPR4 channels (n=1,n=2 and n=3). The figure is driven by replacing the delay element D with e^{-jwt}.

Note that the overall transfer function of partial response channel is composed of two parts: $(1-D)$ and $(1+D)^n$. The first component, i.e. $(1-D)$, corresponds to the transfer function of the writer part. The (non-zero) coefficients of D in the second component, i.e, $(1+D)^n$, represent sample values of the step response of the read-back channel at the sampling instants. When multiplied by $(1-D)$, the step response is transformed into a pulse (dibit response). This is shown in figure 3 for $n=1$, $n=2$ and $n=3$. Note that for large n, $(1+D)^n$ has a Gaussian shape.

Choosing polynomials with higher degrees as the signaling model have a number of advantages [16]. First of all, a larger n is equivalent to having more samples per clock period. Hence, the recording density could be increased. Moreover, as n grows, the number of output levels will increase. For example, the number of levels for PR4 channel is three, i.e.

$y_k \in \{-1, 0, 1\}$. For EPR4 and EEPR4 channels, this number is five and seven, respectively.

Greater number of output levels translates into increased amount of redundancy in the system. The additional redundancy facilitates the decoding process which makes achieving higher coding densities possible. However, as just mentioned, decoding complexity will grow exponentially with n in the maximum likelihood detector.

Another advantage comes from the shape of the response shown in figure 3. As n increases, the pulse shape becomes smoother and have less oscillations. Therefore, equalization becomes less critical for EPR4 and EEPR4 compared to PR4.

However, these advantages are not for free. The price we have to pay is the increased complexity in the ML detector and circuit design for the hardware. Moreover, since the number of output levels becomes larger as n grows, we must have Analog to Digital Converters (ADCs) with higher resolutions.

Therefore, there is tradeoff between coding density and decoding complexity. There exist some methods for determining the best value for n based on channel response, date rate and noise properties [5]. Moreover, as shown in [17], having larger n requires higher input SNRs at the decoder to achieve the same error rate. Therefore, in our implementation, we have only considered the three mostly used channels, i.e. PR4, EPR4 and EEPR4 channels.

Equation (1) clearly indicated the correlation between output bits which results in ISI. In order to obtain the transmitted bits, we have to apply the filter $1/H(D)$ to the received sequence, $Y(D)$. If the received sequence contains error, i.e. $\hat{Y}(D) \neq Y(D)$, then even if we can implement $1/H(D)$ exactly, we will have error propagation because of the correlations in the sequence.

However, by precoding the transmitted bits we can overcome error propagation problem. For instance, if we precode the data bits using the filter $C(D) = [1/H(D)]_{mod\ 2}$, i.e. $X(D) = [U(D)/H(D)]_{mod\ 2}$, and then transmit $X(D)$ instead of $U(D)$, the received bits in the ideal case would be $Y(D) = [U(D)]_{mod\ 2}$ [9].

IV. The Readback Channel

In the readback channel, magnetization direction on the magnetic disk is translated into electrical voltage. The readback voltage, $e(t)$, is related to the magnetized direction, $m(t)$, according to the relationship shown in equation (3) [6]. In this equation, $h(t)$ represents the readback channel step function and " $*$ " indicates convolution operation.

$$e(t) = h(t) * \frac{dm(t)}{dt} \tag{3}$$

Magnetization direction is a function of the current used in writing data bits on the magnetic medium, $i(t)$. In practice, it can be assumed that $m(t)$ is directly proportional to $i(t)$ [9]. Therefore, given the input sequence x_k and assuming NRZ method is used for simplicity, the writing current is given by the equation 4.

$$i(t) = \sum_k x_k \Pi_T(t - kT) \tag{4}$$

where $\Pi_T(t)$ is the pulse with duration T:

$$\Pi_T(t) = \begin{cases} 1, & \text{if } 0 \leqslant t < T; \\ 0, & \text{otherwise} \end{cases}$$

Based on equations (3) and (4), the readback voltage is obtained according to equation (5).

$$e(t) = h(t) * \frac{di(t)}{dt} = 2\sum_{k=0}^{\infty} y_k h(t - kT) \tag{5}$$

in which we have:

$$y_k = \begin{cases} x_k - x_{k+1}, & \text{if } k \geqslant 1; \\ x_0, & \text{if } k = 0 \end{cases}$$

Moreover, it is known that $h(t) = ae^{-bt^2}$ can be used as a good approximation for $h(t)$ in the above equation [6].

If the channel step function satisfies the Nyquist ISI criterion, as shown in equation (6), then the sampled value of the readback voltage gives us the received bits, i.e. $e(nT) = 2y_n$.

$$h(nT) = \begin{cases} 1, & \text{if } n = 0; \\ 0, & \text{if } n \neq 0 \end{cases} \tag{6}$$

In fact, PR signaling introduces nulls at Nyquist frequencies to achieve the bandpass spectrum [17]. By appropriately equalizing the readback signal we can fulfill the spectral requirements over a range of recording densities. However, as recording density is increased further beyond this range, noise enhancement and low readback SNR (because of the peak power limitation of the channel) results in low input SNR at detector which reduces performance below acceptable thresholds.

We now focus on the details of the discrete model for the readback channel.

A. Channel Model

As mentioned before, the readback channel is a partial response noisy channel. The amount of noise and ISI depends on various parameters such as magnetic materials, read head and recording density [21].

There are two major source of noise in the readback channel: electrical noise and media noise. The former, which is caused by the read head and pre-amplifier [21], could be modeled as an AWGN noise. The latter, the media noise, is the dominating type of noise in high recording densities and is caused by the granularity of the magnetic material [4]. An important point about the media noise is that it is generated during signal transitions. In fact, it is some times called transition noise. Therefore, media noise not only depends on the magnetic material, but also on the stored sequence and the transitions in it.

Media noise itself is divided into three sub-categories [3]: position jitter, width jitter and partial erasures. While the model mentioned in [3] is exact, it is too complex. As a result, some simplified versions of this model are used in practice , [7], [11], [20].

The model considered in this report is the one given in [12]. Denote the single side response of a bit during transition by $s(t)$. Equation (7) describes the behavior of $s(t)$ [1], [6].

$$s(t) = A\tanh(\log 3t)/T_{50} \tag{7}$$

where A is the amplitude of the pulse and T_{50} is the duration between the times pulse reaches $A/2$ from $-A/2$. Then, the dibit response of one bit during transition, indicated by $h(t)$, is given by equation (8).

$$h(t) = s(t) - s(t - T_s) \tag{8}$$

Where T_s is the sampling period.

In absence of the media noise, the output of the readback channel is simply composed of the combination of dibit responses, as shown in equation (9).

$$y(t) = \sum_{k=0}^{\infty} x_k h(t - kT_s) + N(t) \tag{9}$$

where $N(t)$ is the white Gaussian noise.

However, ignoring media noise is too simplistic as it is dominating source of noise in high recording densities. In practical models, media noise is reduced to position jitter. In this model, the position of a pulse varies according to jitter parameter Δ which is random variable with Gaussian distribution, i.e. $\Delta \sim N(0, \sigma_j^2)$.

Therefore, a more accurate version for channel output is given by equation (10).

$$y(t) = \sum_{k=0}^{\infty} x_k h(t - kT_s + \Delta) + N(t) \tag{10}$$

Using Taylor expansion, we can simplify equation (10) into (11).

$$y(t) = \sum_{k=0}^{\infty} x_k (h(t - kT_s) + \Delta\frac{dg(t - kT_s)}{dt}) + N(t) \tag{11}$$

An accurate model of the channel that accounts for both electrical and media noises is shown in figure 4.

In this figure, u is the stored bit, x is the output of the transition detector and y is the channel output bit. Transition response is given by h_k^s in equation (12b). The index k here denotes the effects of the transition response on the k^{th} output bit after the transition. This is a direct result of the convolution in equation (3). For instance, if a transition occurs in the i^{th} position of the stored sequence, u, the output sequence would be $y_k = x_i h_k^s, i \leqslant k = 0 \leqslant N - i$, where N is the length of the sequence.

Electrical noise is taken care of in n_k which passes through a shaping function, h_k^n, as shown in equation (12a). Position jitter and width jitter are considered as well and are given in equations (12c) and (12d). These two noises depend on two zero mean Gaussian random variables, j_k and w_k, which determine the variance of position and width jitter.

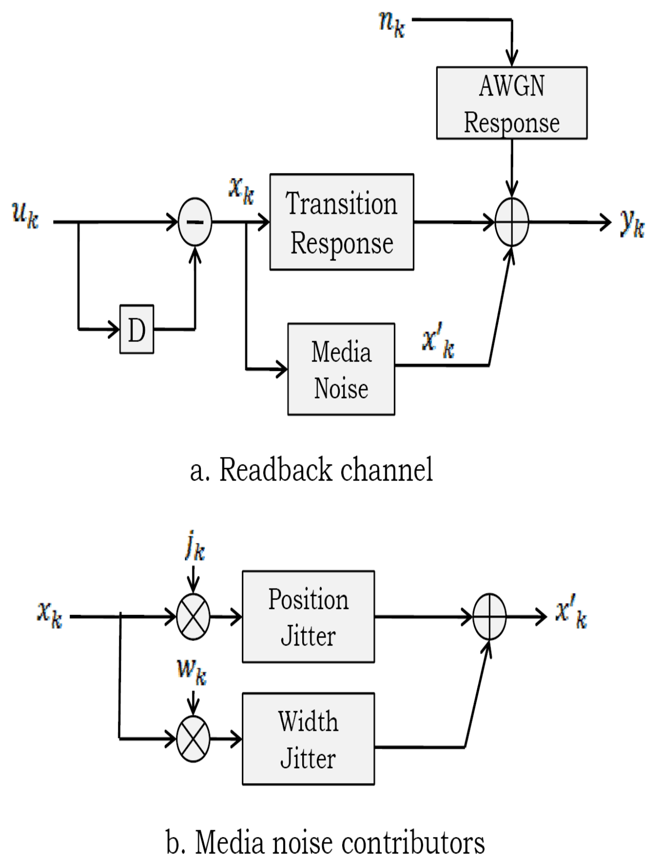

Fig. 4. Detailed model of the readback channel

$$\sqrt{\frac{E_t D_s}{2\pi} tanh(\frac{D_s \pi}{2})}\frac{k + D_s/2}{(D_s/2)^2 + k^2} \tag{12a}$$

$$h_k^s = \frac{E_t D_s^2}{D_s^2 + k^2} \tag{12b}$$

$$h_k^j = \frac{-2kE_t D_s^3}{(D_s^2 + k^2)^2} \tag{12c}$$

$$h_k^w = \frac{-E_t D_s^2 (D_s^2 - k^2)}{(D_s^2 + k^2)^2} \tag{12d}$$

where

$$\begin{aligned} N_\alpha &= E_t 10^{-SNRdB/10} \\ N_0 &= (1 - \alpha) N_\alpha \\ M &= \alpha N_\alpha \\ M_j &= (1 - \lambda) M \\ M_w &= \lambda M \end{aligned}$$

Parameter	Symbol	Command line syntax and available options
Sequence length	N	-f N
Number of iterations	R	-n R
Channel type	-	-t PR4/EPR4/EEPR4
Modulation type	-	-r 4_5/6_7/16_17
Recording density	D_s	-d D_s
Signal to Noise Ratio (SNR)	SNR_{dB}	-s SNR_{dB}
Percentage of media noise	α	-a α , $0 \le \alpha \le 1$
Percentage of width jitter	λ	-l λ , $0 \le \lambda \le 1$

Fig. 5. Input parameters of the simulator

$$
\begin{aligned}
\sigma_n &= \sqrt{N_0/2} \\
\sigma_j &= \sqrt{M_j/(4E_t)} \\
\sigma_w &= \sqrt{M_w/(4E_t)}
\end{aligned}
$$

In the above equations, D_s is recording density. α determines the fraction of noise power due to media noise. The contribution percentage of width jitter in total noise is governed by the factor λ. E_t is the transition power which is usually normalized to one. The variance of AWGN, position jitter and width jitter are shown by σ_n, σ_j and σ_w, respectively.

V. Implementation Details

In this section, we discuss the simulator in more details. The block diagram of the simulator is essentially the same as the one shown in figures 1 and 2. There is a slight variation from this figure in the reader file as we have merged the equalizer into the ML detector. Moreover, in the writer part, precorder is actually a block to transform NRZ into NRZI format.

The input parameters of the simulator are shown in table 5. As shown in the table, the simulator gets the length of the sequence which is going to be simulated. Moreover, it can be told the number of times it has to repeat the simulations for the given sequence length via the "number of iterations" parameter. For example, one can specify the input length to be 10^5 and the iteration number to be 10. This is *almost* like to perform the simulation for a sequence with length 10^6 [2].

In addition to the length and the number of iterations, the user can specify the type of channel going to be used from the three available options, i.e. PR4, EPR4 and EEPR4. Furthermore, the type of modulation code can be determined as well. We have implemented three standard modulation codes, namely, codes with rates $4/5$, $6/7$ and $16/17$. Note that since these are block codes, the length of the input sequence must be divisible by the length of the data word in the modulation code.

Channel parameters can also be specified for the simulator. These parameters include the recording density, D_s, SNR in dB form, the percentage of noise due to media noise, α, and the fraction of media noise due to width jitter, λ.

After determination of the parameters, the simulator starts by generating a random binary sequence according to the uniform distribution. The generated random bits are then modulated using modulation codes. We have implemented three types of modulation codes with rates 4/5, 6/7 [2] and 16/17 [18].

Data word	Code word	Data word	Code word	Data word	Code word	Data word	Code word
0000	10000	0100	00100	1000	01000	1100	01100
0001	00001	0101	00101	1001	01001	1101	01101
0010	00010	0110	00110	1010	01010	1110	10100
0011	10001	0111	10110	1011	10010	1111	10101

Fig. 6. Mapping between data and code words for the 4/5 code [2].

The code with rate 4/5 is an MTR(2,6) code, i.e. it limits the number of consecutive 1's to two and the maximum number of consecutive 0's to six. Note that these constraints are also considered for neighboring codewords. In other words, not only three consecutive ones are avoided in a single codeword, but also it can not occur in neighboring codewords. For instance, the pattern $"01011; 10001"$ is not acceptable. Table 6 shows the mapping between input sequence and relative codeword [2].

In addition to codewords shown in table 6, there might be two minute adjustments needed during the implementation since we may end up having more than six consecutive zeros if we do the coding according to table 6. For instance, if we have the input sequence "0000;0001" the output sequence would be "10000;00001", which has eight consecutive ones. In order to resolve this issue, whenever the last bit of the previous codeword is zero and we have one of the two input sequences "0001" or "0010", we replace the first two bits of the relative codeword with "1" instead of "0". In other words, we will have "11001" and "11010" in the output instead of "00001" and "00010" if the last bit of the previous codeword is zero [2]. It is easy to verify that after having this small adjustment, all the required constraints are met.

The $6/7$ code is an $MTR(2, 8)$ code, i.e. the number of consecutive ones and zeros are limited to 2 and 8, respectively. Table 7 indicates the mapping between data and code words for the "6/7" code[3].

Please note that similar to the "4/5" code, some small adjustments are needed here as well. First of all, we may encounter cases of having three consecutive ones if we do the coding according to table 7. For instance, we may have "....001;110...." or "....101;110....". In such cases, we simply replace the former sequence by "....011;010...." and the latter

[2]There is a slight difference between these two situations since the simulation restarts at the beginning of each iteration. However, for large lengths, these situations are almost the same for practical purposes.

[3]This table is slightly different from the one presented in [2] as parts of the third and fourth columns in table II of [2] are the same and redundant due to a typo. The necessary corrections are made here though.

Data word	Code word	Data word	Code word	Data word	Code word	Data word	Code word
000000	0001000	010000	0101000	100000	1001000	110000	1101000
000001	0000001	010001	0100001	100001	1000001	110001	1100001
000010	0000010	010010	0100010	100010	1000010	110010	1100010
000011	0001001	010011	0101001	100011	1001001	110011	1101001
000100	0000100	010100	0100100	100100	1000100	110100	1100100
000101	0000101	010101	0100101	100101	1000101	110101	1100101
000110	0000110	010110	0100110	100110	1000110	110110	1100110
000111	0001010	010111	0101010	100111	1001010	110111	1101010
001000	0011000	011000	0110001	101000	1011000	111000	0001100
001001	0010001	011001	0010000	101001	1010001	111001	0001101
001010	0010010	011010	0100000	101010	1010010	111010	0101100
001011	0011001	011011	0110000	101011	1011001	111011	0101101
001100	0010100	011100	1000001	101100	1010100	111100	1001100
001101	0010101	011101	1010000	101101	1010101	111101	1001101
001110	0010110	011110	1100000	101110	1010110	111110	0110100
001111	0011010	011111	0110010	101111	1011010	111111	0110101

Fig. 7. Mapping between data and code words for the 6/7 code [2].

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	122	89	8A	8B	123	8D	8E	8F	322	A9	AA	AB	323	AD	AE	AF
01	129	C9	CA	CB	A5	CD	CE	CF	329	E9	EA	EB	2A5	ED	EE	EF
02	12A	189	18A	18B	A6	18D	18E	18F	32A	1A9	1AA	1AB	2A6	1AD	1AE	1AF
03	12B	1C9	1CA	1CB	A7	1CD	1CE	1CF	32B	1E9	1EA	1EB	2A7	1ED	1EE	1EF
04	162	91	92	93	163	95	96	97	362	99	9A	9B	363	9D	9E	9F
05	12D	B1	B2	B3	E5	B5	B6	B7	32D	B9	BA	BB	2E5	BD	BE	BF
06	12E	D1	D2	D3	E6	D5	D6	D7	32E	D9	DA	DB	2E6	DD	DE	DF
07	12F	F1	F2	F3	E7	F5	F6	F7	32F	F9	FA	FB	2E7	FD	FE	FF
08	1A2	111	112	113	1A3	115	116	117	3A2	119	11A	11B	3A3	11D	11E	11F
09	149	131	132	133	1A5	135	136	137	349	139	13A	13B	3A5	13D	13E	13F
0A	14A	151	152	153	1A6	155	156	157	34A	159	15A	15B	3A6	15D	15E	15F
0B	14B	171	172	173	1A7	175	176	177	34B	179	17A	17B	3A7	17D	17E	17F
0C	1E2	191	192	193	1E3	195	196	197	3E2	199	19A	19B	3E3	19D	19E	19F
0D	14D	1B1	1B2	1B3	1E5	1B5	1B6	1B7	34D	1B9	1BA	1BB	3E5	1BD	1BE	1BF
0E	14E	1D1	1D2	1D3	1E6	1D5	1D6	1D7	34E	1D9	1DA	1DB	3E6	1DD	1DE	1DF
0F	14F	1F1	1F2	1F3	1E7	1F5	1F6	1F7	34F	1F9	1FA	1FB	3E7	1FD	1FE	1FF
10	222	289	28A	28B	223	28D	28E	28F	226	2A9	2AA	2AB	227	2AD	2AE	2AF
11	169	2C9	2CA	2CB	125	2CD	2CE	2CF	369	2E9	2EA	2EB	325	2ED	2EE	2EF
12	16A	389	38A	38B	126	38D	38E	38F	36A	3A9	3AA	3AB	326	3AD	3AE	3AF
13	16B	3C9	3CA	3CB	127	3CD	3CE	3CF	36B	3E9	3EA	3EB	327	3ED	3EE	3EF
14	232	291	292	293	233	295	296	297	236	299	29A	29B	237	29D	29E	29F
15	16D	2B1	2B2	2B3	165	2B5	2B6	2B7	36D	2B9	2BA	2BB	365	2BD	2BE	2BF
16	16E	2D1	2D2	2D3	166	2D5	2D6	2D7	36E	2D9	2DA	2DB	366	2DD	2DE	2DF
17	16F	2F1	2F2	2F3	167	2F5	2F6	2F7	36F	2F9	2FA	2FB	367	2FD	2FE	2FF
18	262	311	312	313	263	315	316	317	266	319	31A	31B	267	31D	31E	31F
19	22A	331	332	333	22B	335	336	337	22E	339	33A	33B	22F	33D	33E	33F
1A	23A	351	352	353	23B	355	356	357	23E	359	35A	35B	23F	35D	35E	35F
1B	24A	371	372	373	24B	375	376	377	24E	379	37A	37B	24F	37D	37E	37F
1C	272	391	392	393	273	395	396	397	276	399	39A	39B	277	39D	39E	39F
1D	25A	3B1	3B2	3B3	25B	3B5	3B6	3B7	25E	3B9	3BA	3BB	25F	3BD	3BE	3BF
1E	26A	3D1	3D2	3D3	26B	3D5	3D6	3D7	26E	3D9	3DA	3DB	26F	3DD	3DE	3DF
1F	27A	3F1	3F2	3F3	27B	3F5	3F6	3F7	27E	3F9	3FA	3FB	27F	3FD	3FE	3FF

Fig. 8. Mapping between data and code words for the 16/17 code [18].

one with ”....011;001....” [2] [4]. This adjustment makes sure that the number of transitions is less than or equal to two.

Another necessary adjustment is for the cases where we have ”....000,000....”. In these cases, the last two bits of the previous codeword are replaced with ones, i.e. we will have ”....011,000....”. Having performed this operation, we ensure that the number of consecutive zeros is limited to 8 which happens in ”1000000,001....” and ”....100,0000001” [2].

The third implemented modulation code is an MTR(6,6) code with rate 16/17 [18]. In this code, the coding is performed over the first nine bits of the data words. The resulted ten coded bits are then mixed with the last seven bits of the data word to obtain a 17-bit codeword. Table 8 illustrates the mapping between the first nine bits of the data word and the relative codewords. Values in the table are shown in hexadecimal format for convenience.

All three modulation codes have been implemented based on Look Up Tables (LUTs) and boolean expressions. Using LUTs makes the simulator faster compared to using the boolean expressions. However, since the output of the ML detector is not necessarily a valid codeword, demodulating such sequences is not possible if we use LUTs as the received codeword does not match any of the entries. Therefore, in deriving the results shown in the next section, we have used boolean expressions both in modulator and demodulator.

After performing the modulation, the modulated sequence is transmitted over the channel where media and electrical noise are added. The specifications of different noise types are determined according to input parameters and the aforementioned equations.

The output of the readback channel is then equalized and denoised by the ML detector. We have used the Viterbi algorithm to implement the maximum likelihood detector. The structure of the Viterbi decoder and the number of states depend on the channel type. For PR4, EPR4 and EEPR4 channels, we have four, eight and sixteen states, respectively.

Finally, the equalized and denoised sequence is demodulated to get an estimate of what has been transmitted. If the received bit is different from the one transmitted, i.e. $\hat{u}_k \neq u_k$ then an error has been occurred. These erroneous bits are summed and used to calculate the bit error rate of the system. The BER is returned by the simulator as the output.

[4]By verifying these two replacement with the reference [2] one notices that in that paper, authors have replaced ”....001;110....” with ”....011;001....” and ”....101;110....” with ”....011;010....”, which is the reverse of what we have presented here. However, according the logical expressions in the same paper, this way of replacement is not correct and the correct form is the one we mentioned above.

A. Complexity Analysis

The time taken by the modulation encoders and decoders is essentially linear in the input length, N. o is the complexity of the pre-coder. The channel part of the simulator takes $O(N^2)$ amount of time because of the convolution operation. The Viterbi algorithm in the ML detector is $\Theta(2^{2n}N)$, where n is the number of states in the model (4 for PR4, 8 for EPR4 and 16 for EEPR4).

Please note that we have ignored the complexity of error correcting encoder and decoder as the main focus of this report is on the implementation of the magnetic recording channel.

We can enhance the simulator and its speed by applying two modifications. The first modification should be made in the channel block where we have implemented a discrete convolution. Since the channel response is a decaying function, after a number of time slots its magnitude becomes negligible and we can ignore it in practice. Therefore, by assuming the channel response to be zero after a (fixed) number of time slots, we can implement the convolution in $O(N)$ instead of $O(N^2)$, which makes the simulator much faster. However, this improvement is accomplished at the expense of less accuracy in the channel model.

Further improvements in speed and space can be achieved by modifying the structure of the Viterbi decoder. In its standard format, we have to fill all the elements of a $2^n \times N$ matrix and then trace back to deduce the most likely transmitted sequence. However, due to spacial characteristics of modulation codes, it is not necessary to wait until the end of the filling procedure to calculate the most likely sequence of N bits. After a number of steps, we can decide the first m bits of the estimated sequence as all the alternative paths vanish after a certain number of steps. Therefore, instead of filling a $2^n \times N$ matrix, we need to find the elements of a $2^n \times f(m)$ matrix. Here, $f(m)$ is a function of m and depends on the type of modulation code we use. Having made these two modifications, we can reduce the running time of the simulator from $O(N^2)$ to $O(N)$.

VI. Conclusion and Future Works

In this report we have described the details of a simulator for the magnetic recording systems. The simulator is designed for a wide variety of channels including PR4, EPR4 and EEPR4. It can also handle different types of modulation codes such as the standard MTR(2,6) [2], MTR(2,8) [2] and MTR(6,6) [18] codes with rates 4/5, 6/7 and 16/17 respectively.

Furthermore, the simulator is designed to be flexible in the way that each block is designed and implemented separately. Therefore, in order to have a different setting or model compared to the one mentioned in this report, one needs to modify the corresponding block or replace it with another model. Having different settings is possible as well just by adding or removing blocks.

In its current format, the model mentioned in the channel model used in our simulator needs some adjustments to achieve its best performance. Because the model we used was due to Oenning and Moon [12] and they have pursued a different approach than the one we followed in our implementation. Therefore, in order to adapt the model to our simulator, some adjustments are necessary either in the channel model or in the maximum likelihood detector.

VII. Acknowledgment

I would like to deeply thank Dr. Luoming Zhang for his great help and kindness during the process of implementing the simulator.

References

[1] H. Bertram, ”Theory of magnetic recording”, Cambridge University Press, 1994

[2] B. Brickner, J. Moon, ”Design of a rate 6/7 maximum transition run code”, IEEE Transactions On Magnetics, Vol. 33, No. 5, 1997.

[3] J. P. Caroselli, ”Modeling, analysis, and mitigation of medium noise in thin film magnetic recording channels,” PhD thesis, University of California at San Diego, 1998.

[4] R. D. Cideciyan, E. Eleftheriou, T. Mittelholzer, ”Perpendicular and longitudinal recording: a signal processing and coding perspective,” IEEE Transactions on Magnetics, Vol. 38, pp. 1698-1704, 2002

[5] G. Forney, ”Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference,” IEEE Transaction of Information Theory, Vol. 18,No. 3, pp. 363- 378, 1972.

[6] A. S. Hoagland, ”Digital magnetic recording”, John Wiley and Sons. Inc., New York, 1963.

[7] L. Inkyu, T. Yamauchi, J. M. Cioffi, ”Performance comparison of receivers in a simple partial erasure model,” IEEE Transactions on Magnetics, Vol. 30, pp. 1465-1469, 1994.

[8] E. Kretzmer, ”Generalization of a techinque for binary data communication,” IEEE Transactions on Communication Technology, Vol. 14, pp. 67-68, 1966.

[9] H. Kobayashi, D. T. Tang, ”Application of Partial response channel coding to magnetic recording systems” IBM J. Res. And Dev., Vol. 15, July 1970.

[10] A. Lender, ”Correlative Digital Communication Techniques,” IEEE Trans. Comm. Tech., Vol. 12, No. 4, pp. 128-135, 1964.

[11] S. Nair, H. Shafiee, and J. Moon, ”Modeling and simulation of advanced read channels,” IEEE Transactions on Magnetics, Vol. 29, pp. 4056-4058, 1993.

[12] T. Oenning, J. Moon, ”Modeling the lorentzian magnetic recording channel with transition noise”, IEEE Transactions on Magnetics, Vol. 37, No. 1, pp. 583-591, 2001.

[13] Y. Okamoto, H. Osawa, H. Saito, H. Muraoka, and Y. Nakamura, ”Performance of PRML systems in perpendicular magnetic recording channel with jitter-like noise,” Journal of Magnetism and Magnetic Materials, vol. 235, pp. 259-264, 2001.

[14] R. Price, J. Craig, H. Melbye, A. Perahia, ”An experimental, multi-level high density disk recording system,” IEEE Transaction on Magnetics, Vol. 14, No. 5, pp. 315-317, 1978

[15] P. H. Siege, J. K. Wolf, Modulation coding for information storage, IEEE Commun. Mag., vol. 29, no. 12, pp. 68-86, Dec. 1991.

[16] A. Taratorin, Characterization of magnetic recording system: A practical approach, Guzik Technical Enterprise, San Jose, CA, 1995

[17] H. K. Thapar, A. M. Pate, ”A Class of Partial Response Systems for Increasing Storage Density in Magnetic Recording”, IEEE Transactions On Magnetics, Vol. 23, Ni. 5, 1987.

[18] K. P. Tsang, ”Method and apparatus for implementing run length limited codes in partial response channels”, U.S. patent 5,537,112, July ,1996.

[19] R. W. Wood, D. A. Peterson, ”Viterbi Detection of Class IV Partial Response on a Magnetic Recording Channel,” IEEE Transaction on Communications, Vol. 34, No. 5, pp. 454-461, 1986.

[20] T. Yamauchi, J. M. Cioffi, ”A nonlinear model for thin film disk recording systems,” IEEE Transactions on Magnetics, vol. 29, pp. 3993-3995, 1993.

[21] L. Zhang, ”A study on readback channel of perpendicular magnetic recording system”, Technical report, Ecole Polytechnique Federale de Lausanne (EPFL), Sept. 2009.

[22] L. Zhang, ”Channel model in magnetic recording channel (MRC)”, Technical report, Ecole Polytechnique Federale de Lausanne (EPFL), Nov. 2009.