Dynamic Threshold Detection Based on Pearson Distance Detection

Schouhamer Immink, Kees A.; Cai, Kui; Weber, Jos H.

DOI
10.1109/TCOMM.2018.2814618

Publication date
2018

Document Version
Accepted author manuscript

Published in
IEEE Transactions on Communications

Dynamic Threshold Detection Based on Pearson Distance Detection *†‡§

Kees A. Schouhamer Immink, Kui Cai, and Jos H. Weber

January 5, 2018

Abstract

We consider the transmission and storage of encoded strings of symbols over a noisy channel, where dynamic threshold detection is proposed for achieving resilience against unknown scaling and offset of the received signal. We derive simple rules for dynamically estimating the unknown scale (gain) and offset. The estimates of the actual gain and offset so obtained are used to adjust the threshold levels or to re-scale the received signal within its regular range. Then, the re-scaled signal, brought into its standard range, can be forwarded to the final detection/decoding system, where optimum use can be made of the distance properties of the code by applying, for example, the Chase algorithm. A worked example of a spin-torque transfer magnetic random access memory (STT-MRAM) with an application to an extended (72, 64) Hamming code is described, where the retrieved signal is perturbed by additive Gaussian noise and unknown gain or offset.

*Kees A. Schouhamer Immink is with Turing Machines Inc, Willemskade 15d, 3016 DK Rotterdam, The Netherlands. E-mail: immink@turing-machines.com.

†Kui Cai is with Singapore University of Technology and Design (SUTD), 8 Somapah Rd, 487372, Singapore. E-mail: cai_kui@sutd.edu.sg.

‡Jos Weber is with Delft University of Technology, Delft, The Netherlands. E-mail: j.h.weber@tudelft.nl.

§This work is supported by Singapore Ministry of Education Academic Research Fund Tier 2 MOE2016-T2-2-054

1 Introduction

In mass data storage devices, the user data are translated into physical features that can be either electronic, magnetic, optical, or of other nature. Due to process variations, the magnitude of the physical effect may deviate from the nominal values, which may affect the reliable read-out of the data. We may distinguish between two stochastic effects that determine the process variations. On the one hand, we have the unpredictable stochastic process variations, and on the other hand, we may observe long-term effects, also stochastic, due to various physical effects. For example, in non-volatile memories (NVMs), such as floating gate memories, the data is represented by stored charge. The stored charge can leak away from the floating gate through the gate oxide or through the dielectric. The amount of leakage depends on various physical parameters, for example, the device temperature, the magnitude of the charge, the quality of the gate oxide or dielectric, and the time elapsed between writing and reading the data.

Spin-torque transfer magnetic random access memory (STT-MRAM) [1] is another type of emerging NVMs with nanosecond reading/writing speed, virtually unlimited endurance, and zero standby power. In STT-MRAM, the binary input user data is stored as the two resistance states of a memory cell. Process variation causes a wide distribution of both the low and high resistance states, and the overlapping between the two distributions results in read errors. Furthermore, it has been observed that with the increase of temperature, the low resistance hardly changes, while the high resistance decreases, leading to a drift of the high resistance to the low resistance [2], which may lead to a serious degradation of the data reliability for conventional detection.

The probability distribution of the recorded features changes over time, and specifically the mean and the variance of the distribution may change. The long-term effects are hard to predict as they depend on, for example, the (average) temperature of the storage device. An increase of the variance over time may be seen as an increase of the noise level of the storage channel, and it has a bearing on the detection quality. The mean offsets can be estimated using an aging model, but, clearly, the offset depends on unpredictable parameters such as temperature, humidity, etc, so that the prediction is inaccurate. Various techniques have been advocated to improve the detector resilience in case of channel mismatch when the mean and the variance of the recorded features distribution have changed.

For example, estimation of the unknown offsets may be achieved by using reference cells, i.e., redundant cells with known stored data. The method is often considered too expensive in terms of redundancy, and alternative methods with lower redundancy have been sought for.

Also, coding techniques can be applied to alleviate the detection in case of channel mismatch. Specifically *balanced codes* [3], [4], [5] and *composition check codes* [6], [7] preferably in conjunction with Slepian's optimal detection [8] have been shown to offer solace in the face of channel mismatch. These coding methods are often considered too expensive in terms of coding hardware and redundancy when high-speed applications are considered.

Immink and Weber [9] advocated detectors that use the Pearson distance instead of the traditional Euclidean distance as a measure of similarity. The authors assume that the offset is constant (uniform) for all symbols in the codeword. In [10], it is assumed that the offset varies linearly over the codeword symbols, where the slope of the offset is unknown. The error performance of Pearson-distance-based detectors is intrinsically resistant to both offset and gain mismatch.

Although minimum Pearson distance detection restores the error performance loss due to channel mismatch without too much redundant overhead, it is, however, an important open problem to optimally combine it with error correcting codes. Source data are usually encoded to improve the error reliability, which means that the codewords have good (Hamming) distance properties using structures such as, for example, Hamming or BCH codes. Exhaustive optimal detection of such codes is usually an impracticality as it requires the distance comparison of all valid codewords. The celebrated Chase algorithm [11] has been recommended as it enables the trading of decoder complexity versus error performance of conventional error correcting codes. The Chase algorithm makes preliminary hard decisions of reliable symbols based on a given threshold level. The Chase algorithm reduces the exhaustive search of all symbols in the codeword to only a small number of unreliable symbols. In case of channel mismatch, however, due to incorrectly tuned threshold levels, the hard decisions made are unreliable, and the Chase algorithm fails to deliver reliable detection.

In this paper, we present new dynamic threshold detection techniques used to estimate the channel's unknown gain and offset. The estimates of the actual gain and offset so obtained are used to scale the received signal or to dynamically adjust the threshold levels on a word-by-word basis. Then, the corrected signal, brought into its standard range, can be forwarded to

the final detection/decoding system, where optimum use can be made of the distance properties of the code.

We set the scene in Section 2 with preliminaries and a description of the mismatched channel model. In Section 3, we analyze the case where it is assumed that only the offset is unknown and the gain is known. In Section 4, we discuss the general case, where both gain and offset are unknown. In Section 5, we study the principal case of our paper, where it is assumed that an error correcting code is applied to improve the error performance of the channel. We start by showing that channel mismatch has a detrimental effect on the error performance of the extended Hamming code decoded by a Chase decoder. We show that the presented dynamic threshold detector (DTD) restores the error performance close to the situation with a well-informed receiver. Section 6 concludes the paper.

2 Preliminaries and channel model

We consider a communication codebook, $S \subseteq \mathcal{Q}^n$, of selected codewords $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ over the binary alphabet $\mathcal{Q} = \{0, 1\}$, where n, the *length* of $\boldsymbol{x}$, is a positive integer. The codeword, $\boldsymbol{x} \in S$, is translated into physical features, where logical '0's are written at an average (physical) level b_0 and the logical '1's are written at an average (physical) level $1 + b_1$, where b_0 and $b_1 \in \mathbb{R}$. Both b_0 and b_1 are average deviations, or 'offsets', from the nominal levels, and are relatively small with respect to the assumed unity difference (or amplitude) between the two physical signal levels. The offsets b_0 and b_1 may be different for each codeword, but do not vary within a codeword. For unambiguous detection, the average of the physical levels associated with the logical '0's, b_0, is assumed to be less than that associated with the '1's, $1+b_1$. In other words, we have the premise

$$b_0 < 1 + b_1. \tag{1}$$

Assume a codeword, $\boldsymbol{x}$, is sent. The symbols of the received vector $\boldsymbol{r} = (r_1, \ldots, r_n)$ are distorted by additive noise and given by

$$r_i = x_i + f(x_i; b_0, b_1) + \nu_i, \tag{2}$$

where we define the *switch function*

$$f(x; b_0, b_1) = (1 - x)b_0 + xb_1,$$

and $x \in \{0, 1\}$ is a dummy integer. We assume that the received vector, $\boldsymbol{r}$, is corrupted by additive Gaussian noise $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_n)$, where $\nu_i \in \mathbb{R}$ are zero-mean independent and identically distributed (i.i.d) noise samples with normal distribution $\mathcal{N}(0, \sigma^2)$. The quantity $\sigma^2 \in \mathbb{R}$ denotes the noise variance. We may rewrite (2) and obtain

$$r_i = ax_i + b + \nu_i, \tag{3}$$

where

$$b = b_0 \text{ and } a = 1 + b_1 - b_0. \tag{4}$$

The mean levels, b_0 and b_1, may slowly vary (drift) in time due to charge leakage or temperature change. As a result, the coefficient, $a = 1 + b_1 - b_0$, usually called the *gain* of the channel, and the offset, $b = b_0$, are both unknown to sender and receiver. From the premise (1) we simply have $a > 0$. Note that in [9] the authors study a slightly different channel model, $r_i = a(x_i + \nu_i) + b$, where also the noise component, ν_i, is scaled with the gain a.

We start, in the next section, with the simplest case, namely the offset only case, $a = 1$.

3 Offset-only case

In the offset-only case, $b_0 = b_1 = b$ and $a = 1$, we simply have

$$r_i = x_i + b + \nu_i, \tag{5}$$

where the quantity, b, is an unknown (to both sender and receiver) offset. For detection in the above offset-only situation, Immink and Weber [9] proposed the modified Pearson distance instead of the Euclidean distance between the received vector $\boldsymbol{r}$ and a candidate codeword $\hat{\boldsymbol{x}} \in S$. The modified Pearson distance is defined by

$$\delta(\boldsymbol{r}, \hat{\boldsymbol{x}}) = \sum_{i=1}^{n} (r_i - \hat{x}_i + \overline{\hat{x}})^2, \tag{6}$$

where we define the mean of an n-vector of reals $\boldsymbol{z}$ by

$$\overline{z} = \frac{1}{n} \sum_{i=1}^{n} z_i. \tag{7}$$

For clerical convenience we drop the variable $\boldsymbol{r}$ in (6). A minimum Pearson distance detector operates in the same way as the traditional minimum Euclidean detector, that is, it outputs the codeword $\boldsymbol{x}_{\mathrm{o}}$ 'closest', as measured in terms of Pearson distance, to the received vector, $\boldsymbol{r}$, or in other words

$$\boldsymbol{x}_{\mathrm{o}} = \arg\min_{\hat{\boldsymbol{x}} \in S} \delta(\hat{\boldsymbol{x}}). \tag{8}$$

Immink and Weber showed that the error performance of the above detection rule is independent of the unknown offset b. The evaluation of (8) is in principle an exhaustive search for finding $\boldsymbol{x}_{\mathrm{o}}$, but for a structured codebook, S, the search is much less complex. We proceed our discussion with the definition of a useful concept.

Let S_w denote the set of codewords of weight w, that is,

$$S_w = \{\boldsymbol{x} \in \mathcal{Q}^n : \sum_{i=1}^{n} x_i = w\},\ w = 0, \ldots, n.$$

A set S_w is often called a *constant weight code* of weight w. We study examples, where the codebook, S, is the union of $|V|$ constant weight codes defined by

$$S = \bigcup_{w \in V} S_w, \tag{9}$$

where the *index set* $V \subseteq \{0, 1, \ldots, n\}$.

After working out (6), we obtain

$$\delta(\hat{\boldsymbol{x}}) = \sum_{i=1}^{n} (r_i - \hat{x}_i)^2 + n\overline{\hat{x}}(2\overline{r} - \overline{\hat{x}}), \tag{10}$$

where the first term is the square of the Euclidean distance between $\boldsymbol{r}$ and $\hat{\boldsymbol{x}}$, and the second term, $n\overline{\hat{x}}(2\overline{r} - \overline{\hat{x}})$, makes the distance measure, $\delta(\hat{\boldsymbol{x}})$, independent of the unknown offset b. The exhaustive search (8) can be simplified by the following observations. The decoder hypothesizes that $\boldsymbol{x} \in S_w$. Then we have

$$\delta(\hat{\boldsymbol{x}} \in S_w) = \sum_{i=1}^{n} (r_i - \hat{x}_i)^2 + w\left(2\overline{r} - \frac{w}{n}\right). \tag{11}$$

Since (8) is a minimization process, we may delete irrelevant (scaling) constants, and obtain

$$\begin{aligned}\delta(\hat{\boldsymbol{x}} \in S_w) &= \sum_{i=1}^{n} r_i^2 - 2\sum_{i=1}^{n} \hat{x}_i r_i + \sum_{i=1}^{n} \hat{x}_i^2 \\ &+ w\left(2\overline{r} - \frac{w}{n}\right) \\ &\equiv w\left(1 + 2\overline{r} - \frac{w}{n}\right) - 2\sum_{i=1}^{n} \hat{x}_i r_i. \end{aligned} \tag{12}$$

The symbol $\equiv$ is used to denote equivalence of the expressions (11) and (12) deleting (scaling) constants irrelevant to the minimization operation defined in (8). Note that the term

$$w\left(1 + 2\overline{r} - \frac{w}{n}\right)$$

depends on the number of '1's, w, of $\hat{\boldsymbol{x}}$ and, thus, not on the specific positions of the '1's of $\hat{\boldsymbol{x}}$. The only degree of freedom the detector has for minimizing $\delta(\hat{\boldsymbol{x}} \in S_w)$ is permuting the symbols in $\hat{\boldsymbol{x}}$ for maximizing the inner product $\sum_{i=1}^{n} \hat{x}_i r_i$. Slepian [8] showed that the inner product $\sum_{i=1}^{n} \hat{x}_i r_i$, $\hat{\boldsymbol{x}} \in S_w$, is maximized by pairing the largest symbol of $\boldsymbol{r}$ with the largest symbol of $\hat{\boldsymbol{x}}$, the second largest symbol of $\boldsymbol{r}$ with the second largest symbol of $\hat{\boldsymbol{x}}$, etc.

To that end, the n received symbols, r_i, are sorted, largest to smallest, in the same way as taught in Slepians prior art. Let $(r'_1, r'_2, \ldots, r'_n)$ be a permutation of the received vector $(r_1, r_2, \ldots, r_n)$ such that $r'_1 \geq r'_2 \geq \ldots \geq r'_n$. Then, since the w largest received symbols, r'_i, $1 \leq i \leq w$, are paired with '1's (and the smallest symbols r'_i, $w+1 \leq i \leq n$ with '0's), we obtain

$$\begin{aligned}\delta_w &= w\left(1 + 2\overline{r} - \frac{w}{n}\right) - 2\sum_{i=1}^{w} r'_i \\ &= \sum_{i=1}^{w}\left(-2(r'_i - \overline{r}) + \frac{n+1-2i}{n}\right), \end{aligned} \tag{13}$$

where for convenience we use the short-hand notation

$$\delta_w = \min_{\hat{\boldsymbol{x}}} \delta(\hat{\boldsymbol{x}} \in S_w).$$

Since, as is immediate from (13), $\delta_0 = \delta_n = 0$, the detector cannot distinguish between the all-'0' or the all-'1' codewords. For enabling unique detection one of the two (or both) codewords must be barred from the code book S. In other words, either $V \subseteq \{1, \ldots, n\}$ or $V \subseteq \{0, 1, \ldots, n-1\}$. Such constrained codes, S, called *Pearson codes*, have been described in [9]. In order to reduce computational load, we may rewrite (13) in recursive form, and obtain for $1 \leq w \leq n$,

$$\delta_w = \delta_{w-1} - 2(r'_w - \bar{r}) + \frac{n + 1 - 2w}{n}, \tag{14}$$

where we initialize with $\delta_0 = 0$. The value $w \in V$ that minimizes δ_w is denoted by $\hat{w}$, or

$$\hat{w} = \arg\min_{w \in V} \delta_w. \tag{15}$$

Once we have obtained $\hat{w}$, we may obtain an estimate of the sent codeword, $\boldsymbol{x}$, by applying Slepian's algorithm, and, subsequently we find an estimate of the offset, b. The estimate of the offset, denoted by $\hat{b}$, is obtained by averaging (5), or

$$\hat{b} = \frac{1}{n} \sum_{i=1}^{n} (r_i - \hat{x}_i) = \bar{r} - \frac{\hat{w}}{n}. \tag{16}$$

The retrieved vector, $\boldsymbol{r}$, is re-scaled by subtracting the estimated offset, $\hat{b}$, so that

$$\hat{r}_i = r_i - \hat{b} = r_i - \left(\bar{r} - \frac{\hat{w}}{n} \right), \ 1 \leq i \leq n, \tag{17}$$

where $\hat{\boldsymbol{r}}$ denotes the corrected vector. Note that we can, instead of re-scaling the received signal as done above, adjust the threshold levels used in a Chase decoder to discriminate between reliable and unreliable symbols. For asymptotically small noise variance, σ^2, we may assume with high probability that $\hat{w} = \sum x_i$, so that the variance of the offset estimate, $\hat{b}$, can be approximated by

$$\mathrm{E}[(b - \hat{b})^2] \approx \frac{\sigma^2}{n}, \ \sigma \ll 1, \tag{18}$$

where $E[]$ denotes the expectancy operator. The next example illustrates the detection algorithm.

Example 1 *Let $n = 6$, $\boldsymbol{x} = (110010)$, $\sigma = 0.125$, and offset $b = 0.2$. The received word is $\boldsymbol{r}$ = (1.194, 1.233, -0.024, 0.331, 1.402, 0.263), and after*

sorting we have $\boldsymbol{r}' = (1.402, 1.233, 1.194, 0.331, 0.263, -0.024)$. *We simply find* $\overline{r} = 0.733$. *The next table shows* δ_w *versus* w *using (14).*

w	r'_w	δ_w
1	1.402	-0.505
2	1.233	-1.005
3	1.194	-1.761
4	0.331	-1.123
5	0.263	-0.682
6	-0.024	0.000

We find $\hat{w} = 3$. *The estimated offset equals* $\hat{b} = \overline{r} - \hat{w}/n = 0.733 - 3/6 = 0.233$. ■

Example 2 *Let,* S, n *even, be the union of two constant weight codes, that is,*

$$S = S_{w_0} \cup S_{w_1}, \tag{19}$$

where $w_0 = \frac{n}{2} - 1$ *and* $w_1 = \frac{n}{2} + 1$. *We find from (13) that*

$$\delta_{w_0} = -2\sum_{i=1}^{w_0} r'_i + w_0\left(1 + 2\overline{r} - \frac{w_0}{n}\right)$$

and

$$\delta_{w_1} = -2\sum_{i=1}^{w_1} r'_i + w_1\left(1 + 2\overline{r} - \frac{w_1}{n}\right),$$

so that

$$\delta_{w_1} - \delta_{w_0} = -2(r'_{\frac{n}{2}} + r'_{\frac{n}{2}+1}) + 4\overline{r}. \tag{20}$$

We define the median *of the received vector,* $\tilde{r}$, *as the average of the two middle values (n even) [12], that is,*

$$\tilde{r} = \frac{1}{2}(r'_{\frac{n}{2}} + r'_{\frac{n}{2}+1}). \tag{21}$$

The receiver decides that $\hat{w} = w_1$ *if*

$$\delta_{w_1} - \delta_{w_0} < 0,$$

or, equivalently, if

$$\tilde{r} > \overline{r}. \tag{22}$$

■

In the next section, we take a look at the general case where we face both gain and offset mismatch, $a \neq 1$ and $b \neq 0$.

4 Pearson distance detection

We consider the general situation as in (3) where the symbols of the received vector $\boldsymbol{r} = (r_1, \ldots, r_n)$ are given by

$$r_i = ax_i + \nu_i + b, \tag{23}$$

where both quantities a, $a > 0$, and b are unknown. Immink and Weber proposed the Pearson distance as an alternative to the Euclidean distance in case the receiver is ignorant of the actual channel's gain and offset [9]. The Pearson distance between the n-vectors $\boldsymbol{r}$ and $\hat{\boldsymbol{x}}$ is defined by

$$\delta_{\mathrm{p}}(\hat{\boldsymbol{x}}) = 1 - \rho_{\boldsymbol{r},\hat{\boldsymbol{x}}}, \tag{24}$$

where

$$\rho_{\boldsymbol{r},\hat{\boldsymbol{x}}} = \frac{\sum_{i=1}^{n}(r_i - \overline{r})(\hat{x}_i - \overline{\hat{x}})}{\sigma_r \sigma_{\hat{x}}} \tag{25}$$

is the *Pearson correlation coefficient*. The (unnormalized) variance of the vector $\boldsymbol{z}$ is defined by

$$\sigma_z^2 = \sum_{i=1}^{n}(z_i - \overline{z})^2. \tag{26}$$

A minimum Pearson distance detector operates in the same way as the minimum Euclidean detector, that is, it outputs the codeword $\boldsymbol{x}_{\mathrm{o}}$ 'closest', as measured in terms of Pearson distance, to the received vector, or in other words

$$\boldsymbol{x}_{\mathrm{o}} = \arg\min_{\hat{\boldsymbol{x}} \in S} \delta_{\mathrm{p}}(\hat{\boldsymbol{x}}). \tag{27}$$

The minimum Pearson distance detector estimates the sent codeword $\boldsymbol{x}$, and implicitly it offers an estimate of the gain, a, and offset, b, using (23). We start by evaluating (24) and (27). Since (27) is a minimization process, we may delete irrelevant (scaling) constants, and obtain

$$\delta_{\mathrm{p}}(\hat{\boldsymbol{x}}) \equiv -\frac{1}{\sigma_{\hat{x}}} \sum_{i=1}^{n} r_i(\hat{x}_i - \overline{\hat{x}}). \tag{28}$$

As in the previous section, we consider a code $S = \bigcup_{w \in V} S_w$, where the *index set* $V \subseteq \{0, 1, \ldots, n\}$. Let $\hat{\boldsymbol{x}} \in S_w$, then

$$\delta_{\mathrm{p}}(\hat{\boldsymbol{x}} \in S_w) \equiv \frac{1}{\sqrt{w - \frac{w^2}{n}}} \left[w\overline{r} - \sum_{i=1}^{n} r_i \hat{x}_i \right]. \tag{29}$$

Note that $\delta_{\mathrm{p}}(\hat{\boldsymbol{x}} \in S_w)$ is undefined for $w = 0$ and $w = n$, and we must bar both the all-'0' and all-'1' words from S for unique detection. Clearly, $V \subseteq \{1, \ldots, n-1\}$.

Except for the inner product $\sum r_i \hat{x}_i$, the above expression depends on the number of '1's, w, of $\hat{\boldsymbol{x}}$ and, thus, not on the specific positions of the '1's of $\hat{\boldsymbol{x}}$. For maximizing the inner product $\sum r_i \hat{x}_i$ we must pair the w largest symbols r_i with the w 1's of $\hat{\boldsymbol{x}}$. Let $(r'_1, r'_2, \ldots, r'_n)$ be a permutation of the received vector $(r_1, r_2, \ldots, r_n)$ such that $r'_1 \geq r'_2 \geq \ldots \geq r'_n$. Since the w 1's are paired with the largest symbols, r'_i, $1 \leq i \leq w$, we have [8]

$$\delta_{\mathrm{p},w} = -\frac{1}{\sqrt{w - \frac{w^2}{n}}} \sum_{i=1}^{w} (r'_i - \overline{r}), \tag{30}$$

where $\delta_{\mathrm{p},w}$ is a short-hand notation of $\min_{\hat{\boldsymbol{x}}} \delta_{\mathrm{p}}(\hat{\boldsymbol{x}} \in S_w)$. The detector evaluates $\delta_{\mathrm{p},w}$ for all $w \in V$. Define

$$\hat{w} = \arg\min_{w \in V} \delta_{\mathrm{p},w}. \tag{31}$$

The decoder decides that the $\hat{w}$ largest received signal amplitudes, r'_i, $1 \leq i \leq \hat{w}$ are associated with a 'one', and $n - \hat{w}$ smallest received signal amplitudes, r'_i, $\hat{w} + 1 \leq i \leq n$ are associated with a 'zero'.

The estimates of the gain, $\hat{a}$, and offset, $\hat{b}$, of the received vector $\boldsymbol{r}$ are found by using (4). Let $\hat{b}_0$ and $\hat{b}_1$ denote the estimates of b_0 and b_1, respectively. Then we find

$$\hat{b}_0 = \frac{1}{n - \hat{w}} \sum_{i=\hat{w}+1}^{n} r'_i$$

and

$$\hat{b}_1 = -1 + \frac{1}{\hat{w}} \sum_{i=1}^{\hat{w}} r'_i,$$

so that, after using (4),

$$\hat{a} = 1 + \hat{b}_1 - \hat{b}_0 = \frac{1}{\hat{w}} \sum_{i=1}^{\hat{w}} r'_i - \frac{1}{n-\hat{w}} \sum_{i=\hat{w}+1}^{n} r'_i \tag{32}$$

and

$$\hat{b} = \hat{b}_0 = \frac{1}{n-\hat{w}} \sum_{i=\hat{w}+1}^{n} r'_i. \tag{33}$$

The normalized vector $\hat{\boldsymbol{r}}$ is found after scaling and offsetting with the estimated gain, $\hat{a}$, and offset, $\hat{b}$, that is,

$$\hat{r}_i = \frac{r_i - \hat{b}}{\hat{a}},\ 1 \le i \le n. \tag{34}$$

After the above normalization, the normalized vector, $\hat{\boldsymbol{r}}$, is corrected to its standard range, and may be forwarded to the second part of the decoder, where the vector is processed, decoded, and quantized.

The variance of the estimates $\hat{a}$ and $\hat{b}$ depend on the numbers of 1's and 0's in the sent codeword $\boldsymbol{x}$. For asymptotically small noise variance, σ^2, so that we may assume that with high probability $\hat{w} = \sum x_i$, the variance of the offset, b, denoted by $\sigma^2_{\hat{b},w}$, can be approximated by

$$\begin{aligned} \sigma^2_{\hat{b},w} &= \mathrm{E}[(b-\hat{b})^2] = E\left[\left(b - \frac{1}{n-\hat{w}} \sum_{i=\hat{w}+1}^{n} r'_i\right)^2\right] \\ &= \frac{1}{n-w}\sigma^2,\ \sigma \ll 1. \end{aligned} \tag{35}$$

Similarly, the variance of the estimate of the gain a, denoted by $\sigma^2_{\hat{a},w}$, is given by

$$\sigma^2_{\hat{a},w} = \mathrm{E}[(a-\hat{a})^2] = \frac{n}{w(n-w)}\sigma^2,\ \sigma \ll 1. \tag{36}$$

The above findings are intuitively appealing as they show that the quality of the estimate of the quantities, a and b, depends on the numbers, $n-w$ and w, of '0's and '1's in the sent codeword, respectively. We have verified the above estimator quality using computer simulations. Results of our simulations are collected in Table 1, where we assumed the case $\sigma = 0.1$ and $n = 6$. We are now considering the general case of uncoded i.i.d input data, so that the sent

Table 1: Simulations results of 10^5 samples for $\sigma = 0.1$ and $n = 6$. The values in parentheses are computed using (35) and (36), respectively.

w	$\sigma^2_{\hat{b},w}/\sigma^2$	$\sigma^2_{\hat{a},w}/\sigma^2$
1	0.201 (0.200)	1.201 (1.200)
2	0.250 (0.250)	0.745 (0.750)
3	0.333 (0.333)	0.668 (0.667)
4	0.497 (0.500)	0.751 (0.750)
5	1.011 (1.000)	1.198 (1.200)

codeword does not have a specified weight. The codeword's weight is in the range $\{1, \ldots, n-1\}$. For the i.i.d. case, the variance of the estimations $\hat{a}$ and $\hat{b}$, denoted by $\sigma^2_{\hat{a}}$ and $\sigma^2_{\hat{b}}$, can be found as the weighted average of $\sigma^2_{\hat{a},w}$ and $\sigma^2_{\hat{b},w}$, or

$$\sigma^2_{\hat{b}} = \frac{\sigma^2}{2^n - 2} \sum_{w=1}^{n-1} \binom{n}{w} \frac{1}{n-w} \tag{37}$$

and

$$\sigma^2_{\hat{a}} = \frac{\sigma^2}{2^n - 2} \sum_{w=1}^{n-1} \binom{n}{w} \frac{n}{w(n-w)}. \tag{38}$$

Results of computations and simulations are shown in Table 2, where we assumed the case $\sigma = 0.1$. We have computed the relative variance of the estimators $\sigma^2_{\hat{b}}/\sigma^2$ and $\sigma^2_{\hat{a}}/\sigma^2$ for different values of the noise level, σ, and observed that (37) and (38) are accurate up to a level where the detector is close to failure (word error rate > 0.1).

In the next section, we show results of computer simulations with the newly developed DTD algorithms applied to the decoding of an extended Hamming code.

5 Application to an extended Hamming code

Error correction is needed to guarantee excellent error performance over the memory's life span. To be compatible with the fast read access time of STT-MRAM, the error correction code adopted needs to have a low redundancy of around ten percent and it must have a short codeword length. A (71, 64)

Table 2: Simulations results of 10^5 samples for $\sigma = 0.1$. The values in parentheses are computed using (37) and (38), respectively.

n	$\sigma^2_{\hat{b}}/\sigma^2$	$\sigma^2_{\hat{a}}/\sigma^2$
8	0.297 (0.296)	0.5919 (0.5919)
16	0.135 (0.135)	0.2700 (0.2699)
32	0.064 (0.065)	0.1293 (0.1293)
64	0.031 (0.032)	0.0634 (0.0635)
128	0.017 (0.016)	0.0314 (0.0315)

regular Hamming code is used for Everspins 16 MB MRAM, where straightforward hard decision detection is used [13]. Cai and Immink [14] propose a (72, 64) extended Hamming code with a two-stage hybrid decoding algorithm that incorporates hard decision detection for the first-stage plus a Chase II decoder [11] for the second stage of the decoding routine.

In the next subsection, we show, using computer simulations, that the application of DTD in the above scenario offers resilience against unknown charge leakage or temperature change. We show results of computer simulations with the (72, 64) Hamming code, which is applied to a simple channel with additive noise.

5.1 Evaluation of the Hamming code

An $(n, n-r)$ Hamming code is characterized by two positive integer parameters, r and n, where the redundancy $r > 1$ is a design parameter and n, $n \leq 2^r - 1$ is the length of the code [13]. The payload is of length $n - r$. The minimum Hamming distance of a regular Hamming code equals $d_{\mathrm{H}} = 3$. An *extended* Hamming code is a regular $(n, n-r)$ Hamming code plus an overall parity check. The minimum Hamming distance of an extended Hamming code equals $d_{\mathrm{H}} = 4$.

The word error rate of binary words transmitted over an ideal, matched, channel, using a Hamming code under maximum likelihood soft decision decoding, denoted by $\mathrm{WER}_{\mathrm{H}}$, equals (union bound estimate)

$$\mathrm{WER}_{\mathrm{H}} \approx A_{\mathrm{H}}(n, r) Q\left(\frac{\sqrt{d_{\mathrm{H}}}}{2\sigma}\right), \ \sigma \ll 1, \tag{39}$$

where A_{H} denotes the average number of codewords at minimum distance d_{H}, and the Q-function is defined by

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{u^2}{2}} du. \tag{40}$$

For a regular Hamming code, we have

$$A_{\mathrm{H}}(n, r) = \frac{n(n-1)}{6}, \;\; n = 2^r - 1. \tag{41}$$

For a shortened Hamming code, $n < 2^r - 1$, since the weight distribution of many types of linear codes, including Hamming codes, is asymptotically binomial [15] for $n \gg 1$, we can use the approximation

$$A_{\mathrm{H}}(n, r) \approx \binom{n}{3} \frac{1}{2^r}, \tag{42}$$

and for an extended Hamming code (only even weights)

$$A_{\mathrm{H}}(n, r) \approx \binom{n}{4} \frac{1}{2^{r-1}}. \tag{43}$$

Exhaustive optimal detection of long Hamming codes, such as the extended $(72, 64)$ is an impracticality as it requires the distance comparison of 2^{64} valid codewords. Sub-optimal detection can be accomplished with, for example, the well-known Chase algorithm [11], [14].

The Chase algorithm selects T of the least reliable bits by selecting the symbols, r_i, having least absolute channel value with respect to the decision level. The remaining $n - T$ symbols, that is the most reliable ones, are quantized. Then, the T unreliable symbols are selected, using exhaustive search, in such a way that the word so obtained is a valid codeword of the Hamming code at hand and that the word minimizes the Euclidean distance to the received vector $\boldsymbol{r}$. The error performance of the Chase algorithm is worse than the counterpart error performance of the full-fledged maximum likelihood detector given by (39). The loss in performance depends on the parameter T.

As the parameter T determines the complexity of the search, it is usually quite small in practice. The majority of symbols are thus quantized using hard decision detection, where a pre-fixed threshold is used. The error performance of the Chase decoder depends therefore heavily on the accuracy of

the threshold with respect to mismatch of the gain and offset of the signal received. This means that the Chase decoder loses a major part of its error performance in case of channel mismatch.

Using computer simulations, we computed the error performance of the Chase decoder in the presence of offset or gain mismatch versus the noise level $-20\log_{10}\sigma$. We simulated the error performance of an extended $(72, 64)$ Hamming code decoded by the Chase decoder, where we selected, in Figure 1, the offset mismatch case, $a = 1$ and $b = 0.15$. Figure 2 shows the gain mismatch case, $a = 0.85$ and $b = 0$ ($b_0 = 0$, $b_1 = -0.15$). Both diagrams show the significant loss in performance due to channel mismatch. Combinations of offset and gain mismatch give similar devastating results [9]. The word error rate found by our simulations of the ideal channel (without mismatch), is quite close to the theoretical performance given by the union bound estimate (39) and (43).

In order to improve the detection quality, we applied DTD, as presented in the previous sections, followed by (standard) Chase decoding. Before discussing the simulation results, we note two observations. The all-'1' word is not a valid codeword, and the all-'0' word is a valid codeword of the $(72, 64)$ Hamming code. The probability of occurrence of the all-'0' word, assuming equiprobable codewords is $2^{-64} \approx 10^{-19}$, which is small enough to be ignored for most practical situations. The weight of a codeword of an extended Hamming code is even, so that the number of evaluations of δ_w or $\delta_{\mathrm{p},w}$, see (13) and (30), can be reduced.

Figure 1 shows the word error rate in case DTD is applied in the offset-only case, $a = 1$ and $b = 0.15$. We notice that DTD restores the error performance close to the error performance of the ideal offset-free situation.

Figure 2, the gain mismatch case, shows that the error performance with DTD (Curve 2) is worse than that of the ideal case, $a = 1$, without applying DTD (Curve 4). This can easily be understood: in case $a = 0.85$ ($b_0 = 0, b_1 = -0.15$), the average levels, b_0 and $1 + b_1$, of the recorded data, x_i, are closer to each other than in the ideal case, $a = 1$. Curve 3 shows that the error performance with DTD is close to the situation, where the receiver is informed about the actual gain, $a = 0.85$. This gives a fairer comparison, and we observe that the WER of DTD almost overlaps with the simulated, matched channel, performance. This demonstrates the efficacy of DTD for the case of $a = 0.85$ ($b_0 = 0, b_1 = -0.15$).

Figure 3 shows the WER as a function of the offset mismatch, b, where $a = 1$ and $-20\log\sigma = 15$, using a Chase decoder, $T = 4$. The error performance

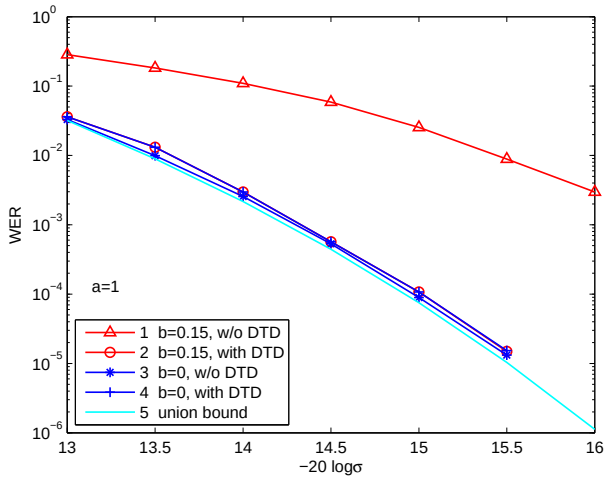

Figure 1: Word error rate (WER) of the extended $(72, 64)$ Hamming code with and without dynamic threshold detection (DTD), and with and without an offset, $b = 0.15$, using a Chase decoder, $T = 4$. The union bound estimate to the word error rate for the ideal channel, $a = 1$ and $b = 0$, given by (39), is plotted as a reference (Curve 5).

of the DTD is unaffected by the offset mismatch, b, and the error performance is close to the performance without mismatch. Figure 4 shows the WER as a function of b_1, where the gain $a = 1 + b_1$, $-20 \log \sigma = 15.5$, and $b_0 = 0$, using a Chase decoder, $T = 4$. Curve 3 shows the situation where the receiver is informed about the actual gain (no mismatch), and we infer that the error performances of a receiver of the matched channel and a receiver of the mismatched channel combined with DTD are very similar.

Above we have shown simulation results of dynamic threshold detection used in conjunction with an extended Hamming code and a Chase decoder. We remark that although in this paper we exemplify DTD detection on an extended Hamming code, the hybrid DTD/decoding algorithm is a general tool that can be applied to other (extended) BCH codes, LDPC, polar codes, etc., for applications in both data storage and transmission systems.

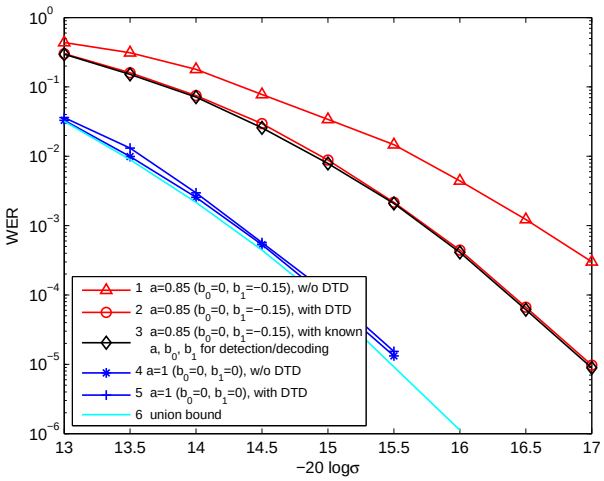

Figure 2: Word error rate (WER) of the extended $(72, 64)$ Hamming code with and without dynamic threshold detection (DTD), and with and without a gain mismatch, $a = 0.85$ ($b_0 = 0$, $b_1 = -0.15$), using a Chase decoder, $T = 4$. The union bound estimate, Curve 6, to the word error rate for the ideal channel, $a = 1$ and $b = 0$, given by (39), is plotted as a reference. Curves 2 and 3 show that the error performance with DTD is close to the situation, where the receiver is informed about the actual gain, $a = 0.85$.

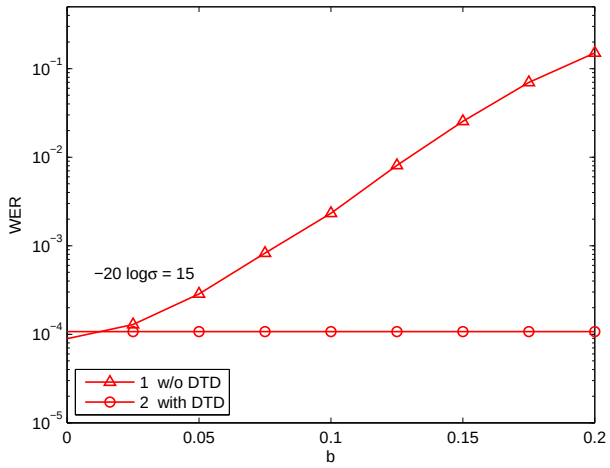

Figure 3: Word error rate (WER) of the extended $(72, 64)$ Hamming code with and without dynamic threshold detection (DTD), versus the offset mismatch b, where $a = 1$ and $-20 \log \sigma = 15$, using a Chase decoder, $T = 4$.

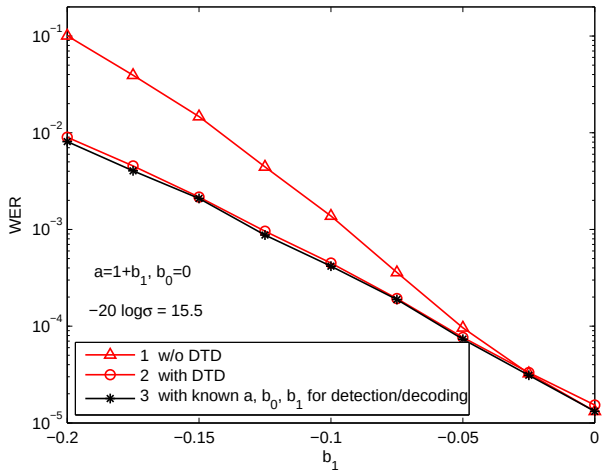

Figure 4: Word error rate (WER) of the extended $(72, 64)$ Hamming code with and without dynamic threshold detection (DTD), versus the gain mismatch $a = 1 + b_1$, $b_0 = 0$, where $-20 \log \sigma = 15.5$, using a Chase decoder, $T = 4$. Curve 3 shows the situation where the receiver is informed about the actual gain.

6 Conclusions

We have considered the transmission and storage of encoded strings of binary symbols over a storage or transmission channel, where a new dynamic threshold detection system has been presented, which is based on the Pearson distance. Dynamic threshold detection is used for achieving resilience against unknown signal-dependent offset and corruption with additive noise. We have presented two algorithms, namely a first one for estimating an unknown offset only and a second one for estimating both unknown offset and gain. As an example to assess the benefit of the new dynamic threshold detection, we have investigated the error performance of an extended $(72, 64)$ Hamming code using a Chase decoder. The Chase algorithm makes hard decisions of reliable symbols that are above or below a given threshold level. In case of channel mismatch, however, due to incorrectly tuned threshold levels, the hard decisions made are unreliable, and as a result the Chase algorithm fails. We have shown that the error performance of the extended Hamming code degrades significantly in the face of an unknown offset or gain mismatch. The presented threshold detector dynamically adjusts the threshold levels (or re-scales the received signal), and improves the error performance by estimating the unknown offset or gain, and restores the performance close to the performance without mismatch. A worked example of a Spin-torque transfer magnetic random access memory (STT-MRAM) with an application to an extended (72, 64) Hamming code has been described, where the retrieved signal is perturbed by additive Gaussian noise and unknown gain or offset.

References

[1] M. Hosomi, H. Yamagishi, T. Yamamoto, K. Bessho, Y. Higo, K. Yamane, H. Yamada, M. Shoji, H. Hachino, C. Fukumoto, H. Nagao, and H. Kano, "A novel nonvolatile memory with spin torque transfer magnetization switching: Spin-RAM," *Tech. Dig. Intl. Electron Devices Meeting (IEDM)*, Washington, USA, pp. 459-462, Dec. 2005.

[2] X. Kou, J. Schmalhorst, A. Thomas, and G. Reiss, "Temperature dependence of the resistance of magnetic tunnel junctions with MgO barrier, "*Appl. Phys. Lett.,* vol. 88, pp. 212-215, 2006.

[3] K. A. S. Immink, "Coding Schemes for Multi-Level Channels with Unknown Gain and/or Offset Using Balance and Energy constraints," pp. 709-713, IEEE International Symposium on Information Theory, (ISIT), Istanbul, July 2013.

[4] H. Zhou, A. Jiang, and J. Bruck, "Balanced Modulation for Nonvolatile Memories," arXiv:1209.0744, Sept. 2012.

[5] B. Peleato, R. Agarwal, J. M. Cioffi, M. Qin, Member, and P. H. Siegel, "Adaptive Read Thresholds for NAND Flash," *IEEE Transactions on Commun.,* vol. COM-63, pp. 3069-3081, Sept. 2015.

[6] F. Sala, R. Gabrys, and L. Dolecek, "Dynamic Threshold Schemes for Multi-Level Non-Volatile Memories," *IEEE Trans. on Commun.,* pp. 2624-2634, vol. COM-61, July 2013.

[7] K. A. S. Immink and K. Cai, "Composition Check Codes," *IEEE Trans. Inform. Theory,* vol. IT-64, pp. 249-256, Jan. 2018.

[8] D. Slepian, "Permutation Modulation," *Proc. IEEE,* vol. 53, pp. 228-236, March 1965.

[9] K. A. S. Immink and J. H. Weber, "Minimum Pearson Distance Detection for Multi-Level Channels with Gain and/or Offset Mismatch," *IEEE Trans. Inform. Theory,* vol. IT-60, pp. 5966-5974, Oct. 2014.

[10] K. A. S. Immink and V. Skachek, "Minimum Pearson Distance Detection Using Mass-Centered Codewords in the Presence of Unknown Varying Offset," *IEEE Journal on Selected Areas of Communications*, vol. 34, pp. 2510 - 2517, 2016.

[11] D. Chase, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information," *IEEE Trans. Inform. Theory,* vol. IT-18, pp. 170-179, Jan. 1972.

[12] R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*, 5th ed. New York: Macmillan, 1995.

[13] W. E. Ryan and S. Lin, *Channel Codes, Classical and Modern*, Cambridge University Press, 2009.

[14] K. Cai and K. A. S. Immink, "Cascaded Channel Model, Analysis, and Hybrid Decoding for Spin-Torque Transfer Magnetic Random Access Memory (STT-MRAM)," *IEEE Trans on Magn.*, vol. MAG-53, pp. 1-11, Nov. 2017.

[15] V. M. Sidel'nikov, "Weight spectrum of binary Bose-Chaudhuri-Hoquinghem codes," *Probl. Peredachi Inform.,* vol. 7, no. 1, pp. 14-22, Jan.-Mar. 1971.