Error Rate Prediction
in
Digital Magnetic Recording

Error Rate Prediction
in
Digital Magnetic Recording

Charles R. Bond

Contents

List of Figures

Abstract

A technique for predicting error rate in digital recording systems is presented. It is based on the following observations and assumptions:

- Data are recovered in the time domain, and time domain models are adequate to describe and quantify errors,
- Data recovery involves peak sensing and bit decisions made in a fixed timing window,
- Read head signals can be modeled by the superposition of isolated pulses,
- Errors (incorrect bit decisions) are caused by two dominant mechanisms, both of which produce time-base errors, (1) peak shift due to random disturbances (bit jitter), and (2) peak shift due to pulse crowding (intersymbol interference).
- Real world timing error distributions are neither unimodal nor bimodal, but are multimodal with a fixed, code-dependent, number of modes.

The analysis of BER (Bit Error Rate) in terms of calculable or measurable noise factors has been adequately documented in the literature, and we will draw on previous work throughout this paper. Our contribution is in the quantification of peak shift caused by pulse crowding, and the integration of its effects with the code dependencies and noise models appropriate to a given system.

We introduce the concept of an *interval pair matrix* which enables us to analyze codes for their influence on peak shift. This matrix is a tool which can be used for the enumeration of timing distribution modes within the recovery window and for computing the statistical weight to be associated with each mode. From this, and noise-induced error distributions derived by other means, we will develop a theoretical framework for predicting error rate and performing time interval analyses.

Introduction

In a practical digital magnetic recording system the maximum bit densities will be limited because of intersymbol interference (pulse crowding) and random noise.[1] The effect of pulse crowding can be understood by considering an isolated flux change or transition on a magnetic tape. Suppose this reference transition lies between adjacent neighboring transitions of opposite polarity. If the distance from the reference to its neighbors is great, the influence of the neighbors on the reference transition is small. A peak sensing readback device would see no detectable change in the amplitude or peak position of the head signal compared with a completely isolated transition. On the other hand, as the neighboring transitions are moved more and more closely to the reference, measurable interference will occur. This interference causes distortion which is a complicated, non-linear function of the distance from the reference to each of its neighbors.

The distortion will manifest itself as apparent changes in the amplitude and location of the reference transition. The closer an interfering signal is to it, the more its amplitude is reduced and its apparent position is shifted. The shifting, commonly called peak shift, is directed away from the interfering signal and ultimately limits the maximum practical recording densities. In peak position sensing systems, this shift generally has a greater influence on the error rate than does the amplitude loss.

There are multiple contributors to intersymbol interference. But for the recording systems which concern us, the effects can be adequately explained as a consequence of the linear superposition of isolated pulses. At higher densities there are non-linear effects associated with writing that complicate the picture considerably, and we will focus our attention on the simpler case.

For a given head/media interface, a specific amount of peak shift will be associated with fixed distances from the reference peak to each of its neighbors. These two intervals, in effect, define the reference peak. That is, for every transition interval pair, there will be a characteristic shift of the enclosed transition. We can associate this peak shift with the particular interval pair that produced it and analyze the distribution of intervals to provide estimates of the combined effect of peak shift on data recovery.

A good understanding of the relationship between peak position sensing errors and the encoding method used is crucial in the development of high performance recording systems. But the literature is largely silent on quantitative techniques for deriving accurate error rate predictions where peak shift is a dominant source of error. One reason is that the code studies generally rely on communications theory to provide tools for error rate analysis and approach

[1] In some digital systems there may be other factors which dominate, but these are the major factors in the magnetic recording systems used today.

the subject from the point of view of channel packing, bandwidth, Signal to Noise Ratio (SNR), etc. Generally, frequency domain models are employed and there is no clear link between the results of the analysis and the problems of data recovery in a timing window. Some recording signal studies, on the other hand, deal with pulse shapes and approach the subject from the point of view of isolated pulses or dipulses (dipoles), magnetic vectors, flux densities, SNR, etc. But they stop short of fully relating timing error distributions to the peak shift modes associated with the particular encoding scheme in use.

Both of these approaches yield results which are useful to the system designer and analyst, but on balance they leave the question of real world error rate performance largely unanswered. Consider the following questions:

- If the packing density is increased by 5%, how much will the BER change?
- How much of an improvement in BER will we get by using write pre-compensation for certain patterns on certain tracks (if any)?
- If the media resolution is increased by a fixed percentage, what change in BER will occur?
- Will a change from MFM to M^2FM code cause the error rate to go up? down? — how much?
- Will enlarging the data recovery window improve or degrade the error rate?
- Is it possible that a slightly shifted window will give better performance than a centered one?
- How can we accelerate a BER test and extrapolate the results correctly?

These are all legitimate questions which are difficult or impossible to answer using frequency domain models, and current time domain analysis techniques only provide partial answers to some of them. In order to address these questions directly, our approach is to develop models which provide accurate predictions of fractional peak shift under typical data recovery conditions, and to analyze the actual recording codes used in digital systems in terms of their relation to these models. For example, once we have created an acceptable pulse model for a given application, we will derive a generating function which produces fractional peak shift curves for all combinations of intervals. We will also determine the allowable interval pairs for a given encoding method and the probability of occurrence of each pair. We can then calculate the actual peak shift values (modes) which will be produced by the allowable time intervals and determine the correct statistical weighting for each. With these, we can construct accurate error distribution curves for the given system and encoding method. These curves allow us to compute the expected error rate of the system for any specified

recovery window under a variety of conditions, given only a single estimate or measure of the timing error due to random noise.

Having tools of this sort available allows us to predict a number of useful quantities. Among them are:

1. The expected average error rate for a system processing a random data stream,
2. The difference in error rates for different encoding schemes in a given system,
3. The error rate improvement or degradation resulting from changes in the PW_{50} of the head signal,
4. The quantitative effect of write pre-compensation on error rate,
5. The influence of different pulse models on error rate,
6. The effect of the recovery window on the error rate, and the changes introduced by window sizing or shifting.

In addition, we will be prepared to determine such things as the optimum write pre-compensation for specific systems, and the highest packing densities consistent with a pre-determined error rate ceiling. We will also confirm that a slightly shifted window may give better overall perfomance than a centered one for certain codes. It will be seen that the composite timing error distribution permits the system designer, for the first time, to accurately extrapolate error rate performance from measurements made using narrow recovery windows.

The first part of this paper deals with the analysis and quantification of timing errors due to pulse crowding. Several models are developed and graphs are provided to assist in computing fractional peak shift under a variety of conditions. The second part deals with the special analysis of encoding methods. The analysis aims at creating a weighted distribution function for use in constructing the previously mentioned error distribution curves. Together, these provide a consistent set of tools to aid in the development or enhancement of high performance digital recording systems.

Chapter 1

Pulse Analysis

In this chapter we will deal with the construction of realistic models for quantifying the effects of interacting pulses. The models can be refined from simple to complex in a series of steps in order to improve the accuracy. This approach has the advantage that future enhancements can be easily incorporated in the basic modeling process.

The simplified models used as a starting point do not in any way imply limitations in the methods developed. They are simply convenient approximations which can be analyzed easily and upon which more complex models may be built.

1.1 Isolated Pulses and Superposition

We begin with a classic illustration of the effects of pulse interaction. In Figure 1.1 we show the combined effect of the superposition of two isolated pulses.

This figure is an accurate representation of two Gaussian waveforms of opposite polarity and their sum. The equations for these pulses are

$$p_1 = \beta e^{-\frac{(t+h)^2}{\alpha^2}} \tag{1.1}$$

$$p_2 = -\beta e^{-\frac{(t-h)^2}{\alpha^2}} \tag{1.2}$$

where $h = \mathrm{PW}_{50}/2$ and $\alpha = 0.3003$. For this figure the two isolated pulses, p_1 and p_2, represented by dashed lines, have their PW_{50} and separation equal to 0.5 in. Note that the isolated pulses, for which $\beta = 1.2$ in, have been offset from the horizontal centerline for clarity, but the sum is properly situated.

A close look at the figure reveals discernable, but slight, peak shift. In fact, the fractional peak shift[1] for this case is 0.0166, or 1.66% of PW_{50}. Such

[1] In our usage, fractional peak shift is ΔP = peak shift/PW_{50}.

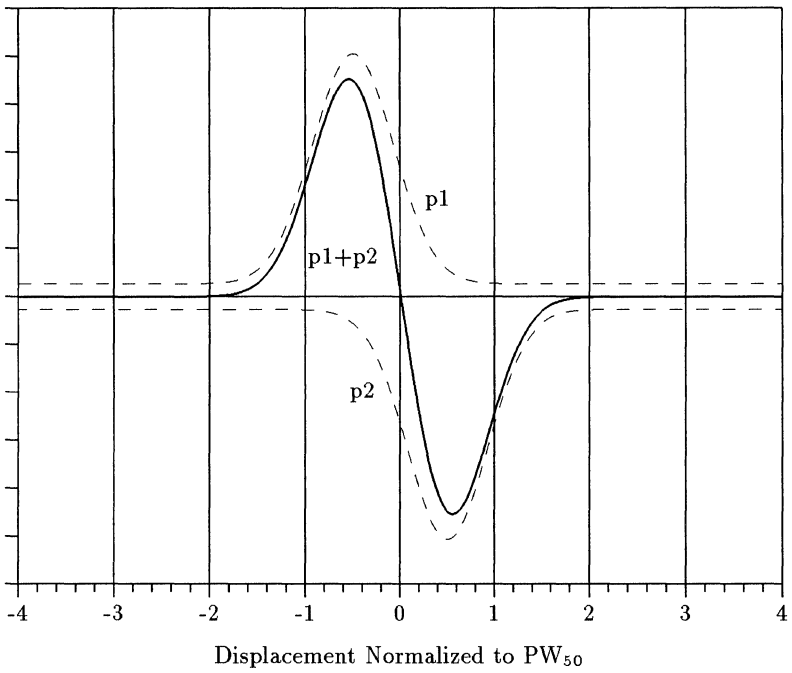

Figure 1.1: Superposition of Gaussian Pulses.

shifting is a direct consequence of the superposition, and not of any special properties of the medium. Peak shift will appear whenever a waveform with distinct maxima or mimima is summed with any other signal having non-zero slope at the extrema.

A curve which relates the pulse spacing to peak shift is shown in Figure 1.2. Here we see that the peak locations read from the media will differ from the recorded transitions locations by amounts which vary with pulse spacing and shape. We have plotted the curves of dipole peak shift for Lorentzian and Gaussian pulse models, both of which will be discussed later.

One interesting feature of these curves is that the limit case of peak shift for zero spacing is finite and nonzero. Of course, the superposition model is no longer valid at extremely close spacing,[2] but the limit cases for idealized models are important for their analytic value. Some additional discussion on those limits can be found in the appendix.

We will construct the pulse models and develop refinements on the premise that superposition produces acceptable composite waveforms from isolated pulses

[2]It is well known that superposition fails to predict actual signal waveforms at very high packing densities. But the major modeling deficiencies at typical densities can be attributed to problems with the individual pulse models rather than with superposition.

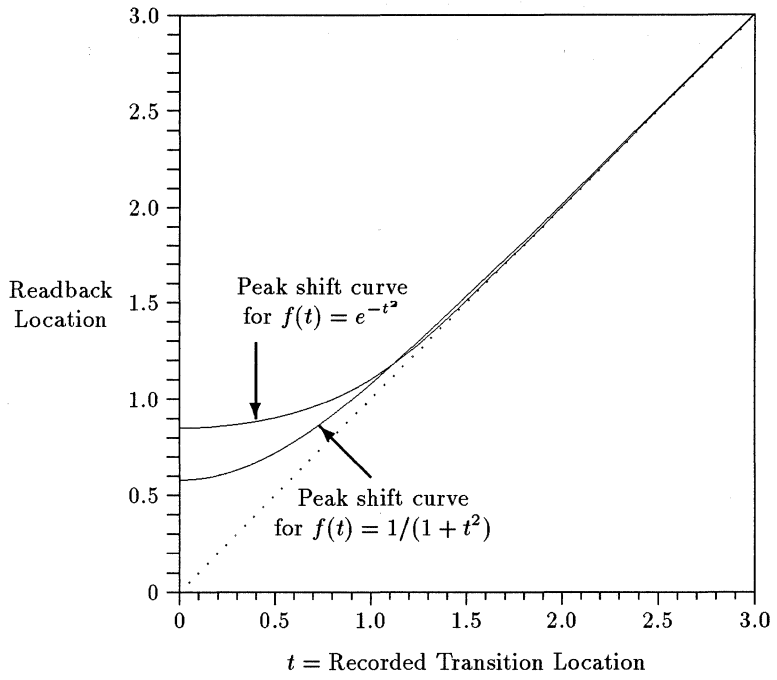

Figure 1.2: Curves of Dipole Peak Shift

for the packing densities which concern us.

An obvious concern in the construction of pulse models involves the presence or absence of lateral symmetry in the pulse models. The simplest models, which occupy much or our attention here, are symmetrical with respect to their local maxima or minima. But for typical isolated pulses there is a mild to very pronounced asymmetry. This asymmetry, which can be partially or completed accounted for by physical phenomena occurring in the media or at the head/disk interface, can cause an asymmetrical peak shift. We will deal with the asymmetrical pulse models after we complete the method for the symmetrical case.

1.2 A Generalized Symmetric Pulse Model

A generalized isolated pulse function which includes Lorentzian and Gaussian as special cases can be defined:

$$\mathcal{P}(t,n) = \left(\sum_{k=0}^{n} \frac{t^{2k}}{k!} \right)^{-1} \tag{1.3}$$

where the Lorentzian case, $\mathcal{P}(t,1)$, expands to

$$\mathcal{P}(t,1) = \frac{1}{1+t^2} \tag{1.4}$$

and the Gaussian case, $\mathcal{P}(t,\infty)$, expands to

$$\lim_{n\to\infty} \mathcal{P}(t,n) = \lim_{n\to\infty} \frac{1}{1+t^2+t^4/2!+t^6/3!+\cdots+t^{2n}/n!} = e^{-t^2} \tag{1.5}$$

This function, which is simply a truncated exponential, can be viewed as a generator for a family of impulses which bridge the gap between the far-spacing (Lorentzian) and near-spacing (Gaussian) models. The pulses can be used as direct models or building blocks in the construction of more complex models.

A plot of several members of the family is shown in Figure 1.3. Here the pulses have been normalized in amplitude and PW_{50} so that their shapes can be compared.

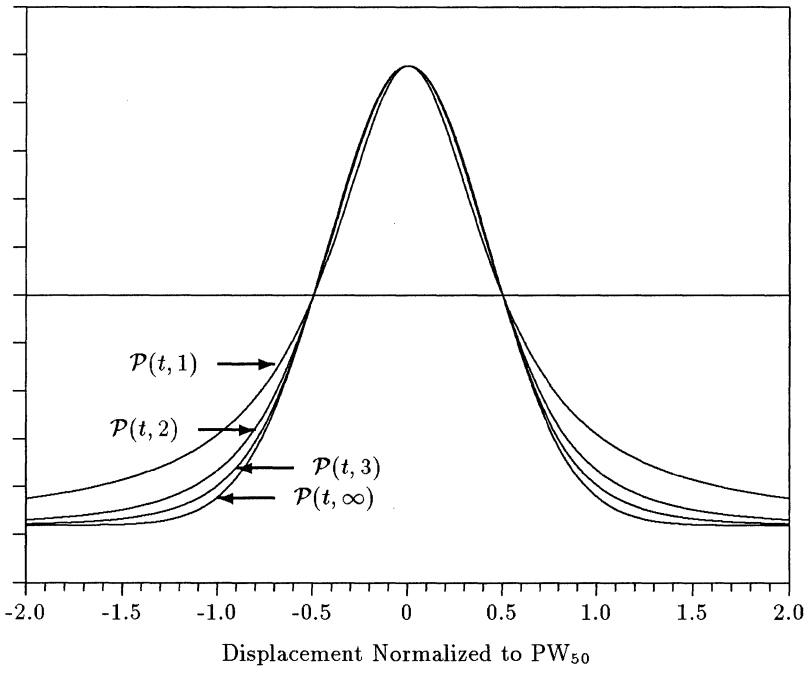

Figure 1.3: Normalized Pulses.

Shifted pulses can be expressed by an horizontal offset parameter h. For example, one way of defining a Lorentzian dipole is:

$$\mathcal{P}(t+h,1) - \mathcal{P}(t-h,1).$$

where the pulses are spaced $2h$ apart.

We will make use of a tripole function

$$\mathcal{P}(t,n) - \mathcal{P}(t+h,n) + \mathcal{P}(t+T,n)$$

where $0 \leq h \leq T$ in the construction of peak shift curves.

By generating arbitrarily long sequences of pulses it is possible to overcome the errors caused by edge effects which limit the usefulness of simple dipole models, *i.e.*, we can 'look into' a signal stream and analyze the pulse interactions which occur there, *i.e.* in context.

Some of the attractive features of the generalized function are,

- It is a function of time, which is a prerequisite for performing signal analysis in the time domain,
- It is mathematically simple, but can be used to construct more complex models by shifting, scaling, summation, differentiation, etc.,
- Unlike piecewise models, it generates entire functions,
- Since it includes commonly accepted models as special cases, it permits us to draw on the previous work in developing our methods.

A minor weakness is that the pulse function, as defined, describes symmetric pulses and leaves the pronounced asymmetry of isolated pulses in certain systems unaccounted for. However, we will find that asymmetric pulses can be modeled by summing scaled derivatives of a pulse with the pulse itself, as has been described elsewhere in the study of vector magnetization. A good compromise between complexity and accuracy can be achieved by this means.

Some of the properties of $\mathcal{P}(t,n)$ can be found in Appendix A.

1.2.1 The Lorentzian Model

The Lorentzian impulse $\mathcal{P}(t,1)$ is one of the most commonly used approximations to an isolated pulse. It is the simplest of these, and can be related directly to a correspondingly simple model of the readback process. It occurs in a context where the magnitude of the flux traversed by a sensor can be approximated by an arctangent function, whose derivative

$$\frac{d\arctan(t)}{dt} = \frac{1}{1+t^2}$$

is the sensor output. Because $\mathcal{P}(t,1)$ has the widest skirts (slowest decay) of the members of the pulse family, it is often referred to as a 'far spacing' model. The spreading or smearing is more pronounced as the distance between the sensor and the flux source is increased. The Lorentzian pulse is often the model of choice for flying head systems.

Although the Lorentzian model is simple, it may have limited use when accurate models are required. Nevertheless, it makes an excellent starting point for approximation methods, and can be improved by means which will be developed later.

Curves for computing peak shift with this model will be found in the appendix.

1.2.2 The Gaussian Model

The Gaussian Pulse function $\mathcal{P}(t,\infty) = e^{-t^2}$, sometimes referred to as the 'near spacing' model, is another often used approximation to the isolated pulse. Like its sister functions it has the combined merits of simplicity and reasonable fit for many systems. For contact recording systems with lateral symmetry in the pulse shape this is often the model of choice.

$\mathcal{P}(t,\infty)$ represents a limit case for the generalized function and has the most rapidly decaying skirts. We would expect that the influence of such a pulse on distant neighbors will be less than for a $\mathcal{P}(t,1)$ pulse, but for close spacing the influence may be greater. This phenomena can be observed in the curves for dipole peak shift. (See Figure 1.2.)

Curves for computing peak shift for this model are also included in the appendix.

1.3 The Peak Shift Curves

At this point, we will explain the process used to derive the curves for determining peak shift. Our goal is to be able to calculate the fractional peak shift given a pulse model, a value for PW_{50}, and a pair of transition intervals. Of course, for this section we are using a simple Lorentzian pulse model, and this will be our initial choice for explaining the method. The values for PW_{50} can vary over some, as yet unspecified, range and we need to provide for all practical cases. Interval pairs will be selected from considerations of the code in use. (See later sections in this paper.) We will use typical cases in the examples.

The waveform shown in Figure 1.4 is the sum of three isolated pulses which define an interval pair. We would like to plot the fractional peak shift of the enclosed pulse as a continuous function of the distance across the entire interval. For this example, the entire interval spans $4 \times PW_{50}$ and the middle pulse peak is at approximately two-thirds of this total distance. We say 'approximately' even though the curve was produced with the implied transition at exactly two-thirds of the distance because peak shift displacement has moved the peak to the left slightly.

For the conditions in the example — $\mathcal{P}(t,1)$ pulse with the total interval equal to $4 \times PW_{50}$ — we can construct a curve which plots fractional peak shift against fractional interval spacing. This enables us to determine the peak shift

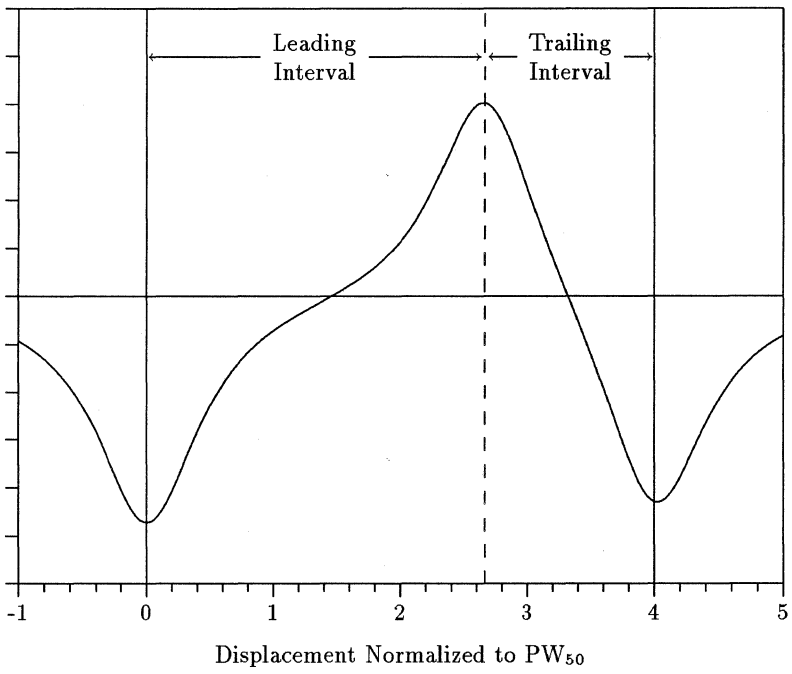

Figure 1.4: Superposition of Three Pulses.

or timing error for any division along this particular interval. A family of curves which plots fractional peak shift against interval span for various ratios of PW_{50} to total interval appears in Figure 1.5.[3]

1.3.1 Computing Peak Shift

With these curves we can calculate peak shift as follows:

1. Determine the PW_{50} of a single isolated pulse for the system under consideration.
2. Select the encoding method to be used.
3. Set the maximum allowable packing density[4] or, equivalently, the bit rate,
4. For each of the interval pairs associated with the code, compute the sum of the leading and trailing intervals,

[3]For the example above we desire a curve for PW_{50}/(total interval) = 0.25, so we would have to interpolate between the curves labelled 0.2 and 0.3.

[4]For rotating media this is cylinder dependent.

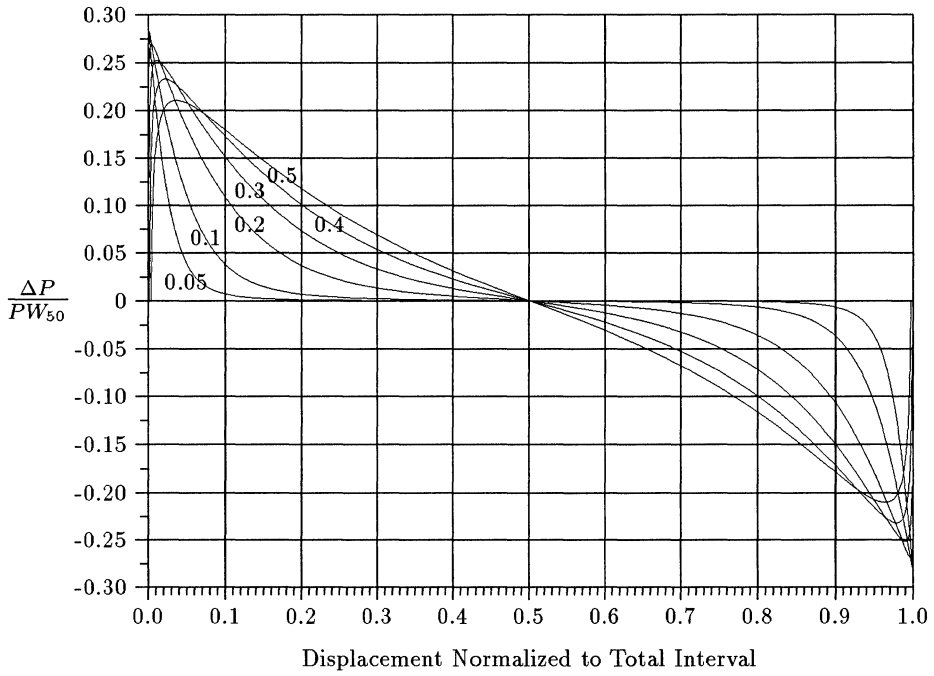

Figure 1.5: Curves of $\mathcal{P}(n,1)$ Tripole Peak Shift

5. Locate the curve which corresponds to the ratio of PW_{50} to the total interval,

6. Move along the horizontal axis to the point which divides the total interval in the same ratio as the leading and trailing intervals,

7. Locate the intersection of this coordinate with the selected curve,

8. Read the fractional peak shift from the vertical axis scale,

9. Convert to absolute peak shift, *i.e.*, multiply by PW_{50}.

As an example of the use of the graph, consider a high density $5\frac{1}{4}$ in. floppy disk system using MFM code. We have chosen the $\mathcal{P}(t,1)$ pulse model and the PW_{50} is measured on a chosen track to be 1.8 μsec. The bit-cell spacing is 2 μsec and we would like to determine the peak shift for the 'worst case' situation consisting of a 2 μsec interval followed by a 4 μsec interval. (An experienced recording technologist would expect a very low peak shift for a system with these values.)

First we compute the total interval, $I = 6\,\mu$sec. Then the ratio of PW_{50} to this interval is $PW_{50}/I = 0.3$. This identifies the curve we will use for determining the peak shift. The ratio of the leading interval to the total interval is 1/3, so that we will read the fractional peak shift from Figure 1.5 at the horizontal coordinate 0.33. The fractional peak shift is 0.025 which, on multiplication by PW_{50} converts to 45 nsec.

Reading high precision values from the accompanying curves in Figure 1.5 is not possible. These curves, although correct, are provided here as examples only. Complete sets of curves for computing peak shift with high accuracy for a wide variety of pulse models and spacings are included in the appendix.

1.4 Asymmetric Pulse Models

In many digital systems there is a pronounced lateral asymmetry in the observed isolated pulses. This asymmetry generally involves visible differences in the leading and trailing edges of the pulse and generally involves a sharpening of the leading edge and spreading of the trailing edge. A purely longitudinal model does not show this phenomena, but considerations of the actual shape of the recorded transition zone provide some explanation. In real media the transition zone is tilted or curved in a manner consistent with the penetration of a magnetizing field. These curved zones contain a perpendicular element as well as horizontal one, and can be approximated with a model which contains some portion of the derivative of the horizontal field.

For example, we can plot a series of models based on the following modified Lorentzian pulse

$$f(t) = \frac{1}{1+t^2} + \frac{\alpha t}{(1+t^2)^2} \tag{1.6}$$

with variable parameter 'α'.

A few of these are shown in Figure 1.6. The case $\alpha = 0$, of course, produces the symmetric Lorentzian.

Graphs of many of the pulse models and information about their equations can be found in the appendices.

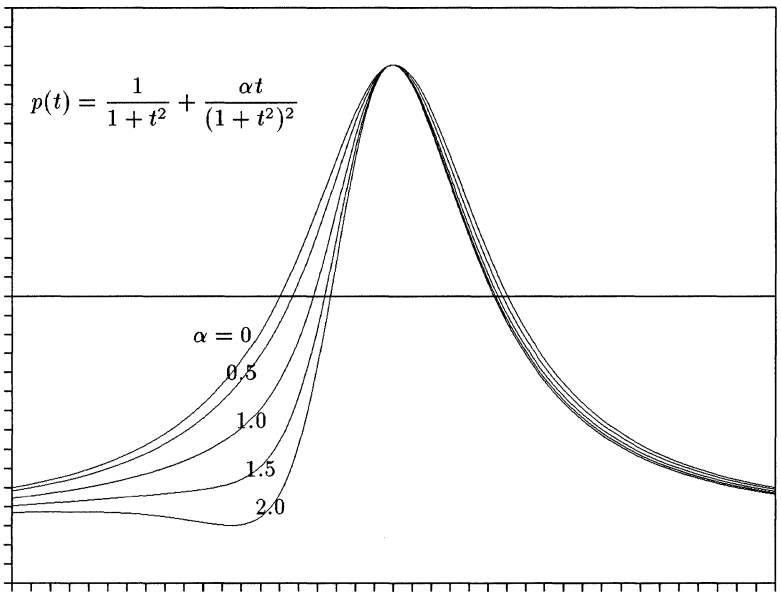

Figure 1.6: Asymmetric Isolated Pulses

Chapter 2

Code Analysis

The integration of the characteristic peak shift curves with code related peak interval distributions is central to the purpose of this paper. Clearly, the marriage of code dependent signal characteristics with shape dependent modeling has long been a partially realized objective in accepted test methodology. Witness the importance of selecting 'worst case' patterns from the repertoire of a recording code for simplified window margin analysis or bit error rate testing. In a real sense, the methods of this paper extend the simplified worst case concept to encompass 'realistic case' situations, providing the means to quantify all cases.

We will present the fundamental concepts by way of examples using common encoding methods. Comparable analyses for other codes will be forthcoming from the author.

2.1 MFM Code

MFM[1] is one of the most common codes used in digital magnetic recording today. It has a higher information-per-transition density than FM and is one of the simplest of the self-clocking codes. With appropriate phase locking circuitry, an MFM encoded data stream can be recovered from a signal source with high accuracy and reliability, even in the presence of frequency drift from the source..

2.1.1 Description of MFM Code

The standard recording application of the code involves the generation of a reference clock which defines a bit-cell stream and the encoding of raw data at a rate of one bit per cell. The data is contained in the bit cell such that the following rules apply,

[1] Sometimes called 'Miller' code.

1. Each '**1**' in the data stream is encoded as a transition in the center of the corresponding bit cell,

2. Each '**0**' is encoded as a transition at the beginning (leading edge) of the bit cell, unless the previous bit cell contains a **1**,

3. If the current data bit is a **0** and the previous data bit was a **1**, no change of state occurs in the current bit cell.

Some of the time domain characteristics of the code are,

1. The shortest time interval between transitions is one bit cell,

2. The longest time interval between transitions is two bit cells,

3. The only other allowable interval is one and a half bit cells.

4. All combinations of the above intervals are allowed.[2]

Since a finite number of interval lengths are possible, we would expect there to be a characteristic distribution of these intervals and interval pairs for any random bit stream encoded by the above rules. A method for determining that distribution is the subject of the next section.

2.1.2 Interval Analysis

It is convenient to catalog all possible interval pairs for the code and identify the binary input code sequences which produce them. These pairs are shown in Figure 2.1. The interval pairs are divided in three groups according to the length of the leading interval. For MFM, this gives a group of combinations which start with a one bit cell interval, a one and a half bit cell interval, and a two bit cell interval.

Some explanation is in order concerning the binary codes which produce these interval pairs. In the figure, those leading intervals which begin with a **1** do not depend on previous data in order to occur. Those which begin with a **0**, on the other hand, do depend on the previous data bit. A transition at the leading edge of a bit cell can only occur if the previous data was a **0**. We have indicated the required previous condition by enclosing the necessary leading **0** in parentheses.

We would like to determine the relative frequency of occurrence of each interval pair. Our assumption that the input data stream (prior to encoding) is random simply means that each new bit is a 'coin toss', *i.e.*, a **1** or **0** is equally likely. For ASCII text and program code this isn't a bad assumption.

If the probability that the next bit is a **1** equals 1/2, then the probability that any specific 2-bit pattern will occur next is 1/4. Similarly, each 3-bit pattern

[2]This is *not* true with some codes.

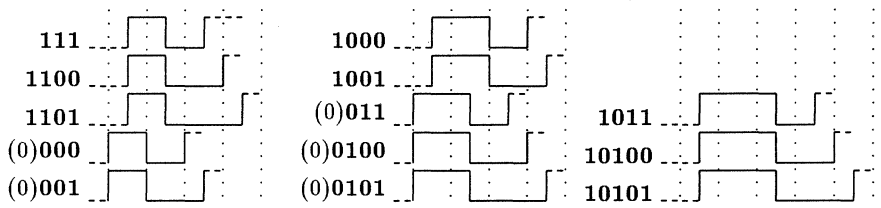

Figure 2.1: Allowable MFM Interval Pairs and Associated Codes

has a probability of 1/8, etc. Using this, we can compute the probabilities and relative frequencies of each of the interval pairs above. The result is shown in Figure 2.2.

2.1.3 The Interval Matrix

Leading Interval \ Trailing Interval	1	1.5	2
1	**3/16**	**1/8**	**1/16**
1.5	**1/8**	**3/32**	**1/32**
2	**1/16**	**1/32**	**1/32**

Figure 2.2: Interval Pair Probability Matrix

The matrix contains one cell for each allowable interval pair. It is organized as a two dimensional array with the row label identifying the present (leading) interval lengths and the column identifying the next (trailing) interval lengths. If, for example, the leading interval is two bit cells in length and the trailing interval is one and a half bit cells in length, then the probability for this combination is 1/32.

Entries in the table are computed by adding up the probabilities of individual cases for each combination. For example, to compute the probability that a one bit cell interval will be followed by a one bit cell interval, we see that there are two ways this can happen. The first way is by the occurrence of a **111** pattern and the second is by a **0000** pattern. Note that we must count the leading zero in this last case. Now the probability of a **111** pattern is 1/8 and that of a **0000**

is 1/16. Thus the total probability is 3/8.

It is important to keep in mind that the matrix shows the probability that a particular interval pair will occur following a reference event, and not the probability that a particular interval pair will be associated with a randomly selected transition. Specifically, the probability that a particular interval pair will occur is equal to the probability that a particular transition type will occur. The sum of the probabilities for all possible types is the transition rate for the code, which is 3/4 for MFM code.[3] The sum of the probabilities that certain interval pairs are associated with a given transition is, of course, unity.

As a quick check on our numbers, we can do as follows. Assume an arbitrarily long string of **1**s and **0**s. The length of this string is related to the individual intervals by $\mathcal{L} = \mathcal{L} \cdot 1 \cdot p_1 + \mathcal{L} \cdot 1.5 \cdot p_{1.5} + \mathcal{L} \cdot 2 \cdot p_2$. That is, it equals the sum of the products of the total length of the string times the probabilities of each interval type times its characteristic length. Then $\mathcal{L} = \mathcal{L}(3/8 + 3/8 + 1/4)$, as would be expected.

The matrix contains a lot of implicit information, some of which we will mention here. First, the sum of all the probabilities in the matrix should equal the transition rate for the code. This is because the interval pairs define transition types and the sum of all their probabilities is precisely the probability that a transition will appear at all. Second, the sums of the probabilities along individual rows is the probability that the corresponding leading interval will occur. Third, it is an easy matter to determine the probability of occurrence of interval pairs associated with 'worst case' peak shifts by adding the probability entries in those cells which represent the greatest difference in the leading and trailing intervals.

Some of the probability information required to complete these matrices can be obtained by considering the interval sequences as Markov chains. But note that the interval matrix is *not* the same as the customary Markov matrix for which the sum of the probabilities along each row would be unity.

2.1.4 Development of Peak Shift Modes

The matrix diagonal from upper left to lower right separates the matrix into a lower triangular part and an upper triangular part. Along the diagonal the leading and trailing intervals are equal. We would not expect any peak shift to occur for these cases.[4] The lower triangle consists of those interval pairs whose leading interval exceeds the trailing interval. Here we would expect the peaks of the enclosed transition to occur early. In contrast the interval pairs defined by the upper triangular part are associated with peaks which arrive late.

[3] The transition rate is the average number of transitions per bit (or bit cell). For MFM this can be determined by computing $t.r. = p_{1_c} + p_{0_c} \cdot p_{0_p}$ where p_{1_c} is the probability that the current bit is a **1** (1/2), p_{0_c} is the probability that the current bit is a **0** (1/2), and p_{0_p} is the probability that the previous bit was a **0** (1/2). Thus $t.r. = 1/2 + 1/4 = 3/4$.

[4] This remark only applies to symmetric pulse models.

The interval pair (or transition type) probabilities shown in Figure 2.2 can be used to construct a weighting function for the various modes of peak shift in an error distribution curve.[5] To pursue this line of investigation it is better to normalize the probability matrix so that we are looking at the probabilities that a given transition is associated with some interval pair. This matrix is shown in Figure 2.3. The relative values of the entries are the same as in Figure 2.2, but now the sum of the probabilities is unity. Normalizing is accomplished by multiplying each entry by the inverse transition rate (4/3) of the code.

		Trailing Interval		
		1	1.5	2
Leading Interval	1	**1/4**	**1/6**	**1/12**
	1.5	**1/6**	**3/24**	**1/24**
	2	**1/12**	**1/24**	**1/24**

Figure 2.3: Normalized Interval Pair Probability Matrix for MFM

From the material presented so far, it is evident that the probabilities (weights) of individual peak shift modes are code-dependent and will differ with differing encoding methods. Thus, the weighted distributions of peak shift modes in a random bit stream encoded by some other method may exhibit higher or lower overall error rates than will MFM, independent of the random system noise present.

We will adopt a special notation for referring to individual cells in the kind of matrix we are considering here. Each entry is defined by its coordinates in the matrix so that, for example, the probability that a given transition has a leading interval of 1.5 bit cells and a trailing interval of 2 bit cells will be designated by $M_{1.5,2} = 1/24$. The sum of the entries along the diagonal is $M_{1,1} + M_{1.5,1.5} + M_{2,2} = 5/12$. This is the probability that a given transition will exhibit zero peak shift. For the 'worst case' interval pair the probability is $M_{1,2} + M_{2,1} = 1/6$.

Refer to Figure 2.4 to see how the peak shift modes are equated with the interval matrix. We have identified a total of seven peak shift modes including the zero shift mode. The modes are labelled from 0 (no peak shift) to ± 3 (maximum peak shift). Actually, we have used a progressive numbering scheme based on interval ratios decreasing (increasing) from 50% to minimum (maximum). Positive numbers identify late-occurring peaks, and negative numbers

[5] We will be concerned with constructing this distribution elsewhere in this paper.

identify early-occurring peaks. Hence, each cell in the matrix corresponds with a specific amount and direction of peak shift.

Since our chosen pulse model exhibits lateral symmetry, the peak shift magnitudes can be arranged in mirror image pairs. For example, modes 2 and -2 represent the same value of shift, but in opposite directions. For asymmetric pulse shapes all modes with the possible exception of mode 0 will have unique values.

Leading Interval \ Trailing Interval	1	1.5	2
1	**0**	**2**	**3**
1.5	**-2**	**0**	**1**
2	**-3**	**-1**	**0**

Figure 2.4: Peak Shift Mode Classification Matrix

Figure 2.5 shows the significance of the matrix in representing peak shift modes. For all codes, there will exist a matrix which conforms with this template. If, for a given code, the probability matrix can be constructed, then the problem of predicting BER for the storage and recovery of a stochastic variable is partially solved. What remains is to determine the magnitudes of the time displacements corresponding with each mode, and the characteristic noise induced time-base error distribution.

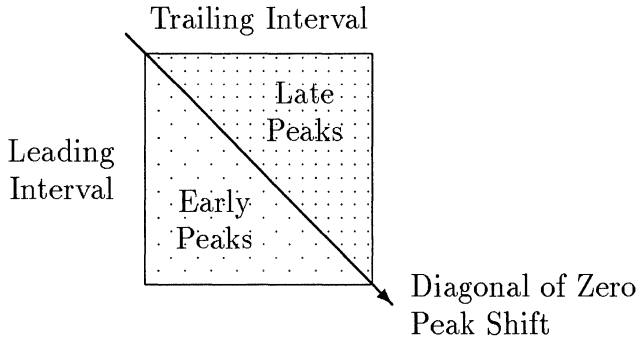

Figure 2.5: Generalized Matrix Template

Having developed the model related peak shift curves in the first part of this paper, and the code related interval matrix in this part, we are now able to

combine them to produce the weighted peak shift histograms required for error analysis. We will proceed with a number of example problems.

Chapter 3

Combining the Peak Shift Curves and Code Matrix

The example problems in this section will be used to clarify the concepts developed so far, and to explain the use of the curves. Our intent is to 'walk through' the construction of a complete error distribution curve which accurately represents the expected field performance of a digital magnetic recording system.

3.1 An Example of MFM Mode Analysis

In this example we will compute the peak shift displacements for all modes in a given problem environment, and construct a weighted error distribution diagram. We will approximate typical conditions for a double density, $5\frac{1}{4}$ in. IBM compatible floppy disk.

Let the model be a $\mathcal{P}(t, \infty)$ (Gaussian) pulse, and the nominal bit cell spacing is 4 μsec. On the cylinder of interest, the PW_{50} is 5 μsec.

Mode	L.I.	T.I.	I	L.I./I	PW/I	ΔP/PW	Peak Shift
1	6 μsec	8 μsec	14 μsec	0.43	0.36	0.015	76 nsec
2	4 μsec	6 μsec	10 μsec	0.4	0.5	0.061	305 nsec
3	4 μsec	8 μsec	12 μsec	0.33	0.42	0.081	401 nsec

Figure 3.1: Tabulation for Example Problem

We can tabulate the solution steps as shown in Figure 3.1, where *Mode* is the numbered entry from the Classification Matrix, *L.I.* is the Leading Interval, *T.I.*

is the Trailing Interval, I is the total Interval, $L.I./I$ is the ratio of the leading interval to the total interval, PW/I is the ratio of PW_{50} to the total interval, $\Delta P/PW$ is the fractional peak shift taken from the graphs, and *Peak Shift* is the unnormalized, absolute peak shift. $L.I./I$ is the measure along the horizontal axis and PW/I is the curve designator.

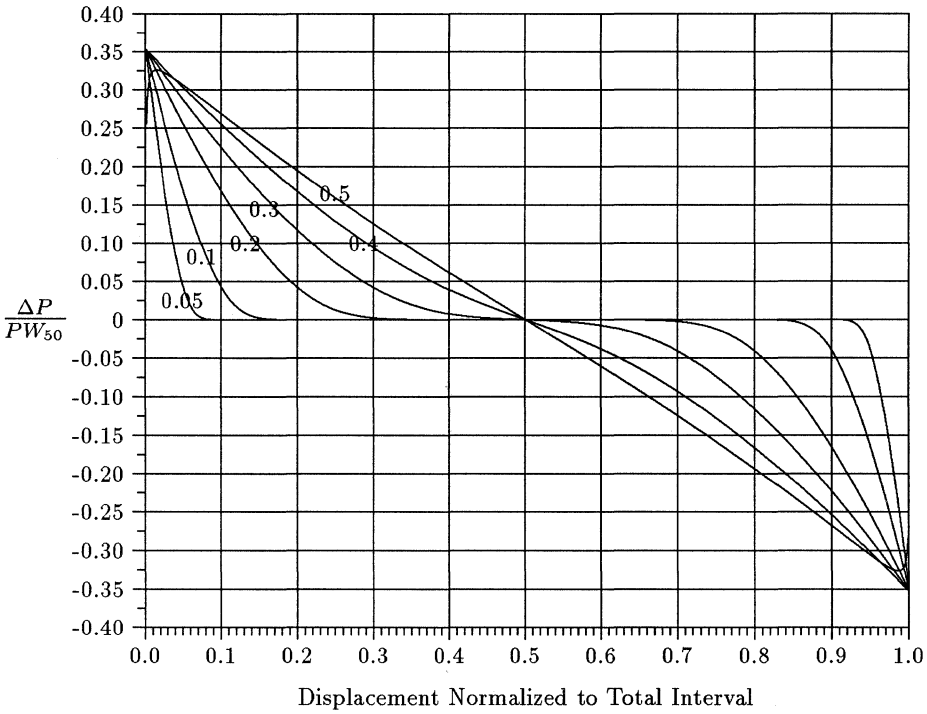

Figure 3.2: Curves of $\mathcal{P}(t, \infty)$ Tripole Peak Shift

Figure 3.2 can be used to visualize the curve set for $\mathcal{P}(t, \infty)$, but the more accurate curves in the appendix were used in finding the values for $\Delta P/PW$.

Since the pulse model is symmetrical we did not have to determine the early and late peak shift values separately, as only the sign is different. It is understood that the peaks are early $(-)$ if the leading interval exceeds the trailing interval and late $(+)$ otherwise.

We are now ready to construct the weighted peak shift mode diagram. Figure 3.3 shows a representation of the 2 μsec timing window within which data is to be recovered. We have assumed a 'standard' MFM window centrally placed within the reconstructed data bit cell. The relative amplitudes of the various modes are taken from the interval matrix, and the horizontal position is taken

from the tabulation in Figure 3.1. As will be seen, the size and location of the window are parameters which can be varied at will.

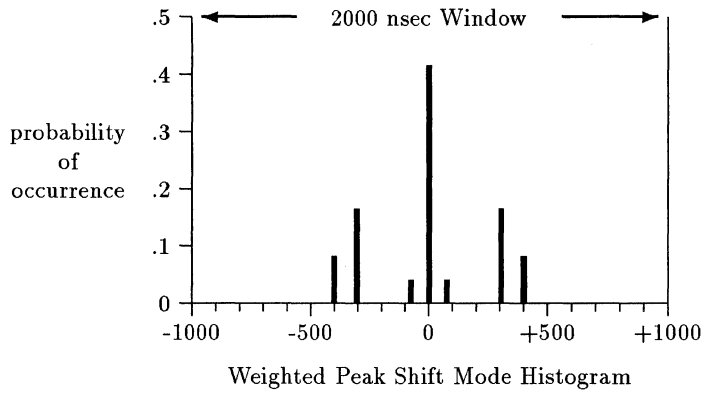

Figure 3.3: Window Diagram for Example MFM Problem

Figure 3.3 can be interpreted as the kind of hypothetical display which one might expect to see on a time interval analyzer (TIA) for an idealized recording system with zero bit jitter. The locations of the peak shift displacements are fixed and unvarying. They are purely a function of the isolated pulse shape, the packing density, and the encoding scheme used. The amplitudes of the peak shift modes are similarly fixed, and are purely a function of the encoding scheme used and the choice of a random variable as the signal source. For real world systems, the histogram has a similar, straightforward interpretation: it shows the *mean values* and *amplitudes* of a set of timing error distributions which fully describe the error performance of the system.

The weighted peak shift mode histogram serves as a substructure for the development of the complete timing error distribution curve. All that remains is to determine what the appropriate timing error distribution for the system under consideration looks like,[1] and to construct the composite function as shown in the next sections.

3.1.1 The Noise Induced Timing Error Distribution

Peak shift displacements are the points at which the mean values of the noise induced timing error distributions are located. These are measured from the zero peak shift reference point. For each of the modes the characteristic, possibly mode dependent, timing error distribution can be determined by controlled tests or can be assumed for the purpose of constraining other parameters.

[1] This can be done empirically by measurements on an existing system, or analytically for a theoretical system.

Such timing error distributions represent the sum of all random noise disturbances associated with the recovery process. These include electronic noise, variations in particulate distribution in the media, external fields, etc. Periodic or slowly varying disturbances such as the once-around envelope modulation or motor speed variations may or may not be problematic depending on the ability of the recovery electronics to reposition the timing window. Here we are separating out the disturbances which randomly offset the peak locations within a fixed timing window.

If we are permitted to assume that timing error is normally distributed, we need only determine the standard deviation, variance or time scaling factor to completely specify the composite function.

We use the following form[2]

$$P(t) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{t} e^{-\frac{(t-m)^2}{2\sigma^2}} dt \tag{3.1}$$

where m is the mean value and σ is the standard deviation of the distribution. It is a standard practice to set $\sigma = 1$ to simplify the construction of tables, and to perform time scaling on t directly before insertion in the formula, as has been done in the discussion which follows.

We now wish to define a related function which represents the error rate for data recovery within the timing window. For our application, the probability that an event is improperly recovered is simply the probability that it falls outside the window, *i.e.*, the sum of the lower tail $P(t_l)$ and upper tail $Q(t_u) = 1 - P(t_u)$, where t_l and t_u are the lower and upper window edge locations, respectively. See Figure 3.4 for a simplified representation of an idealized, single mode distribution for a transition centered in a recovery window.

The error function of interest to us, $g(t, w[k], p[k])$, then is

$$g(t, w[k], p[k]) = \frac{w[k]}{sqrt2\pi} \left[\int_{-\infty}^{t_l} e^{-\frac{(t-p[k])^2}{2}} dt + \int_{t_u}^{\infty} e^{-\frac{(t-p[k])^2}{2}} dt \right] \tag{3.2}$$

where $w[k]$ the normalized weight for mode k, and $p[k]$ is the corresponding normalized peak shift.

Equation 3.2 is a generalized function which relates the peak shift displacements and weights to the window edge locations. Recall that σ is normalized to unity for this function.

3.1.2 The Composite Error Function

A composite error function can now be constructed as in Equation 3.3,

$$\mathcal{G}(t) = \sum_{k=-maxmode}^{+maxmode} g(t, w[k], p[k]) \tag{3.3}$$

[2]There are a variety of forms for this function and its relatives. We have adopted the conventions in *Handbook of Mathematical Functions*, by Abramowitz and Stegun.

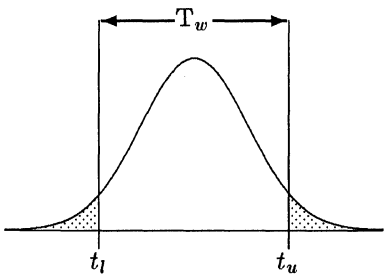

Figure 3.4: Unimodal Transition Error Density Function

where k ranges over all modes associated with the code under analysis. This, then, represents the error rate for the system. More accurately, it is the transition error rate and must be multiplied by the average number of bits per transition to yield the bit error rate.

The composite cumulative error distribution defined above, $\mathcal{G}(t)$ is a computable function of the weights, peak shifts, and time scaling factor. Hence, given the code, pulse model, and transition timing error distribution, it is possible to make an accurate determination of the system error rate. Furthermore, we can then quantify the impact of any combination of changes to the code, model, or intrinsic noise.

To complete the previous example, we have only to specify the transition error density function. Since we already have the mean locations and weights, the horizontal scaling factor is all that is needed.[3]

For our example we will assume that T_w is 10σ or 200ns. The formula is

$$\begin{aligned} \mathcal{G}(t) &= g(t, 0.083, -401) + g(t, 0.167, -305) + g(t, .0417, -76) && (3.4) \\ &\quad + g(t, 0.417, 0) + g(t, 0.0417, 76) + g(t, 0.167, 305 && (3.5) \\ &\quad + g(t, 0.083, 401) && (3.6) \end{aligned}$$

(complete the example)

3.2 M^{2}FM Code

Sometimes referred to as Modified-MFM, this code has a number of features to recommend it over ordinary MFM. The transition rate is 2/3, as opposed to 3/4 for MFM, and the relative frequency of the minimum interval is less.

The interval pair matrix for M^{2}FM is shown in Figure 3.5. The method for determining the probability entries is in the appendix.

[3] It is likely that the normal distribution will fail for transitions which are severely crowded on one side only, depending on the nature of the noise.

3.2.1 Interval Pair Matrix for M^{2}FM

Leading Interval \ Trailing Interval	1	1.5	2	2.5
1	3/16	3/32	3/32	0
1.5	1/16	1/8	5/64	3/64
2	3/32	5/64	1/16	1/64
2.5	1/32	1/64	1/64	0

Figure 3.5: Normalized Interval Pair Probability Matrix for M^{2}MF

Notice that there are two empty cells in the interval pair matrix. One if these is along the zero peak shift diagonal, but the other illustrates a startling asymmetry in the code. One of the modes which exists for early peaks is missing for late peaks. Furthermore, it is a 'worst case' mode, suggesting that the composite error distribution may be skewed, *i.e.*, the least error rate may not coincide with a centered data recovery window!

The peak shift classification matrix is shown in Figure 3.6.

Leading Interval \ Trailing Interval	1	1.5	2	2.5
1	0	3	5	6
1.5	−3	0	2	4
2	−5	−2	0	1
2.5	−6	−4	−1	0

Figure 3.6: Peak Shift Classification Matrix for M^{2}MF

3.2.2 Example Problem for M^{2}FM

For this example, let's assume a system with a 500 nsec bit cell and PW_{50} equal to 750 nsec. We will use the $\mathcal{P}(n, \infty)$ model, and choose a conventional 50% window of 250 nsec duration for data recovery. The peak shift window diagram is shown in Figure 3.7.

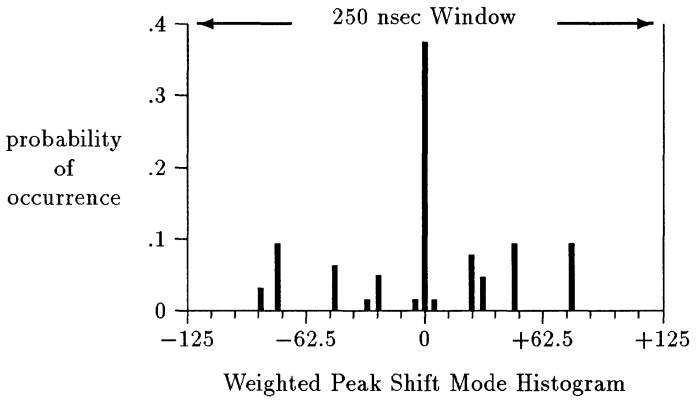

Figure 3.7: Window Diagram for Example M^{2}FM Problem

This histogram deserves a close examination. We have drawn our window diagrams so that the point of zero peak shift is placed exactly at the window center. But, for this case, if the dominant errors are due to peak shifts associated with 'worst case' patterns, it is certain that better performance (lower error rate) will result from shifting the window slightly to the left, so that it is earlier in time than a perfectly centered window. The optimum shift will, of course, depend on the shape of the composite timing error distribution curve.

One question that arises concerns the actual position of the window during the processing of a random data stream. If we assume the data are recovered with a PLL whose phase detector is linear for small differences between the reference frequency and the PLO, which will be true for most phase/frequency detectors, we can then compute a 'center of gravity' within the window toward which the PLL clock edge will converge. This *centroid* is the sum of the products of the peak shift mode weights by their corresponding displacements, *i.e.*, their moments about the mean position of the PLL reference,[4] and does not necessarily coincide with the point of zero peak shift or the center of the window.

For example, the computed average reference location for our previous system is −0.25 nsec; *i.e.*, to the left of the point of zero peak shift. A small

[4]The reason the PLL reference edge will converge to this point is that the PLL corrections are precisely these same weighted sums.

amount, to be sure, but at least it is in the right direction!

For uniform distributions around the modes, and assuming that the system is initially calibrated with a fixed frequency (zero peak shift) reference, the computed window centroid provides a reference point from which the detection window edges can be correctly located. Thus armed with a knowledge of the window placement and the peak shift weights and magnitudes we need only determine the noise induced timing error distributions to complete the error analysis — and this problem has been adequately addressed in the literature. (See references.)

(complete the example)

3.3 RLL 2,7 Code

The interval pair matrix for this code is shown in Figure 3.8.[5] This matrix has the rows and columns labeled according to the number of zeros between transitions, unlike that scheme used previously where the bit-cell length was used as the interval unit. Either method leads to the same matrix. Note that it is asymmetrical around the zero peak shift diagonal, and has an empty cell in one corner.

	Trailing Interval					
	2	3	4	5	6	7
2	.118	.0838	.0592	.0419	.0247	.00740
3	.0904	.0642	.0453	.0321	.0189	.00576
4	.0642	.0453	.0321	.0226	.0132	.00370
5	.0453	.0321	.0226	.0160	.00946	.00287
6	.0172	.0226	.0161	.0113	.00658	.00185
7	0	.00862	.00577	.00431	.00287	.00144

Figure 3.8: RLL 2,7 Interval Pair Matrix

The Mode Classification Matrix is shown in Figure 3.9. As before we have numbered the modes approximately in order of decreasing interval ratio. For this code some pairs have the same ratio, so we subordered by decreasing total interval size. This is somewhat arbitrary because the non-linear relation between

[5] The probability entries for the MFM and M^2FM case were derived analytically and verified with a computer simulation. For RLL 2,7 we have computed the probabilities directly from the simulation.

interval ratios, sizes, and resulting peak shift can cause the modes to 'cross over' each other in the recovery window. But the primary reason for enumerating modes is for identification purposes, so any consistent scheme will serve.

Trailing Interval

	2	3	4	5	6	7
2	0	7	11	13	14	15
3	−7	0	4	8	10	12
4	−11	−4	0	3	6	9
5	−13	−8	−3	0	2	5
6	−14	−10	−6	−2	0	1
7	−15	−12	−9	−5	−1	0

Figure 3.9: Mode Classification Matrix for RLL 2,7

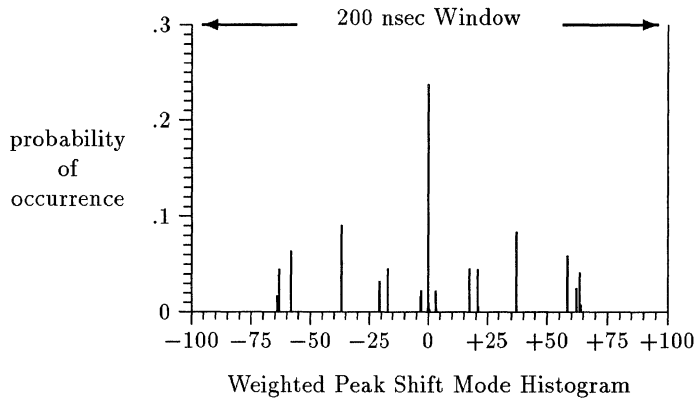

Figure 3.10: Window Diagram for Example RLL 2,7 Problem

(complete the example)

Appendices

Appendix A

Pulse Models

A.1 Generalized Pulse Function

The generalized symmetric pulse defined in the text

$$\mathcal{P}(t,n) = \left(\sum_{k=0}^{n} \frac{t^{2k}}{k!} \right)^{-1} \tag{A.1}$$

is the basis for developing dipole and tripole models. For example, $\mathcal{P}(t,\infty) = e^{-t^2}$ so that $\mathcal{P}(t-h,\infty) = e^{-(t-h)^2}$, where t is time and h is the time displacement of the pulse peak from $t = 0$. A Gaussian dipole can be expressed by $\mathcal{P}(t-h,\infty) - \mathcal{P}(t+h,\infty)$. With the selection of appropriate values for h we can express any pulse train in terms of sums and differences of $\mathcal{P}(t,n)$ functions.

We determine the peak locations of dipoles by differentiation, aided by a numerical computation. For the Gaussian case,

$$f(t,h) = \mathcal{P}(t+h,\infty) - \mathcal{P}(t-h,\infty) \tag{A.2}$$

$$f(t,h) = e^{-(t+h)^2} - e^{-(t-h)^2} \tag{A.3}$$

$$f'(t,h) = -2(t+h)e^{-(t+h)^2} + 2(t-h)e^{-(t-h)^2} \tag{A.4}$$

The zeros of of A.4, of course, correspond to the maxima and minima (peaks) of the dipole. Special numerical algorithms were developed for the solution of this and similar equations involving multiple pulses of various types. These algorithms were used in the construction of the curves used to determine peak shift for dipole and multipole signal models.

A.2 Limit Cases

A special case of (A.4) occurs when h approaches zero. In this case,

$$0 = (t-h) - (t+h)e^{-4th} \tag{A.5}$$

Then,

$$t-h = (t+h)\left[1-(4th)+\frac{(4th)^2}{2!}-\frac{(4th)^3}{3!}+\cdots\right] \tag{A.6}$$

$$t-h = t-t(4th)+t\frac{(4th)^2}{2!}-\cdots+h-h(4th)+h\frac{(4th)^2}{2!}-\cdots \tag{A.7}$$

which, on transposing and cancelling, reduces to

$$-2h = -4t^2h+\cdots(\text{second and higher order terms in 'h'}) \tag{A.8}$$

And, finally, the limiting case as h approaches zero

$$2t^2 = 1, \qquad \text{or} \quad t = \sqrt{\frac{1}{2}} \tag{A.9}$$

A similar development leads to the equations for other members of the $\mathcal{P}(t,n)$ family. For the Lorentzian dipole,

$$f(t) = \frac{1}{1+(t-h)^2}-\frac{1}{1+(t+h)^2} \tag{A.10}$$

$$f'(t) = \frac{-2(t-h)}{(1+(t-h)^2)^2}+\frac{2(t+h)}{(1+(t+h)^2)^2} \tag{A.11}$$

The limit case as h approaches zero leads to $3t^4+2t^2-1=0$, from which

$$t = \sqrt{\frac{1}{3}} \tag{A.12}$$

For the Gaussian model $\mathcal{P}(t,\infty)$ the limit is the positive real root of $t^2 = 1/2$. This is 0.707107. For the Lorentzian model $\mathcal{P}(t,1)$ the limit is the positive real root of $3t^4+2t^2-1$, which is 0.577350. For the model defined by $\mathcal{P}(t,2)$ it is the positive real root of $5t^{10}+19t^8+28t^6+16t^4-4t^2-4$. This equals 0.651219. For $\mathcal{P}(t,3)$ it is the positive real root of $7t^{16}+48t^{14}+169t^{12}+366t^{10}+510t^8+408t^6+108t^4-72t^2-72$. This equals 0.6903979.

A note on 'worst case' patterns — 6DB is NOT the worst case for peak shift! Note that 6DB refers to the constant repetition of the three-bit pattern '110'. 6DB is the hexadecimal pattern which represents the least common multiple of 3 with 4. A worse pattern is '101011', a six-bit pattern whose hex equivalent is CEB. In the 6DB pattern each transition is shifted away from its nearest

$g(t,h,n)$	$\lim_{h\to 0} t$
$n=0$	1
$n=1$	0.577
$n=2$	0.651
$n=3$	0.690
$n=\infty$	0.707

Figure A.1: Table of Peak Shift Limits

neighbor and the neighbor is itself shifted, slightly reducing its influence. With the CEB pattern shifted transitions are interlaced with unshifted transitions maximizing their effect. By extension, blocks of '$111\cdots$'s interlaced with blocks of '$01010\cdots$'s combinations produce the maximum displacement errors at the block boundaries. (Check this out with a simulation program!).

Appendix B

Additional Graphs and Curves

B.1 Peak Shift Curves for $\mathcal{P}(t,1)$

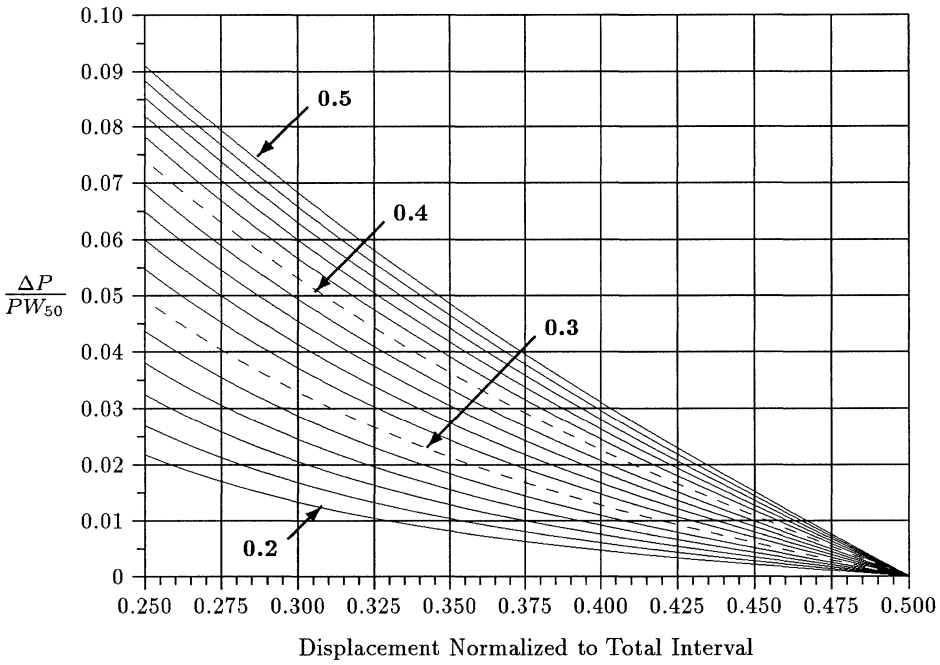

Figure B.1: Curves of $\mathcal{P}(t,1)$ Tripole Peak Shift, Group 1

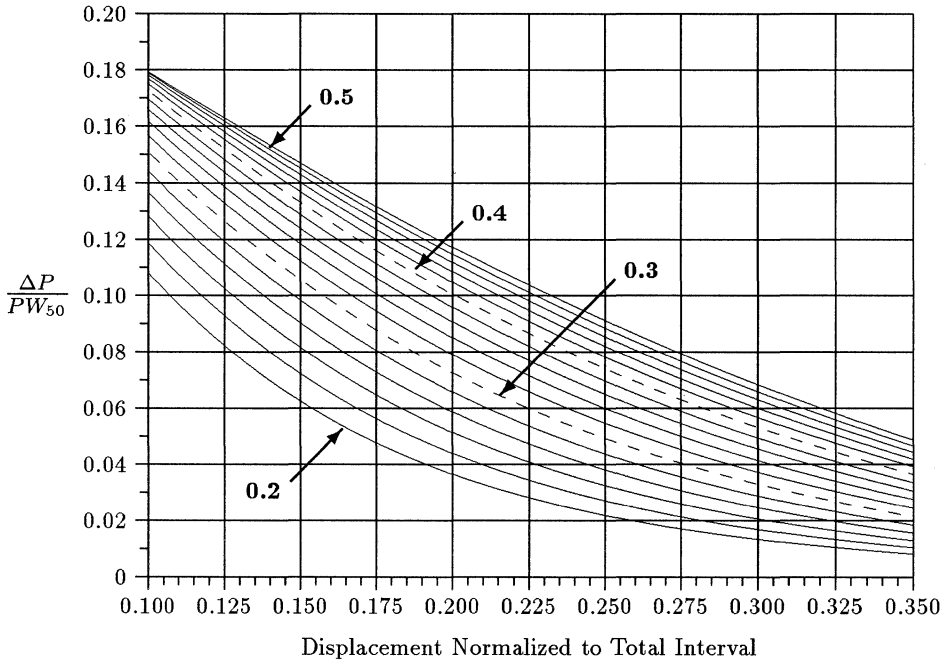

Figure B.2: Curves of $\mathcal{P}(t, 1)$ Tripole Peak Shift, Group 2

B.2 Peak Shift Curves for $\mathcal{P}(t, \infty)$

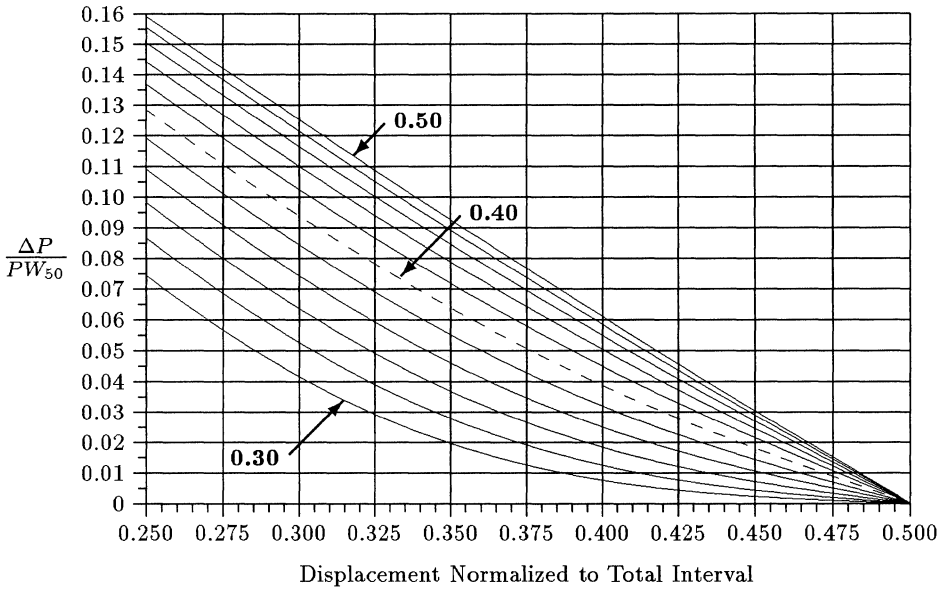

Figure B.3: Curves of $\mathcal{P}(t, \infty)$ Tripole Peak Shift, Group 1

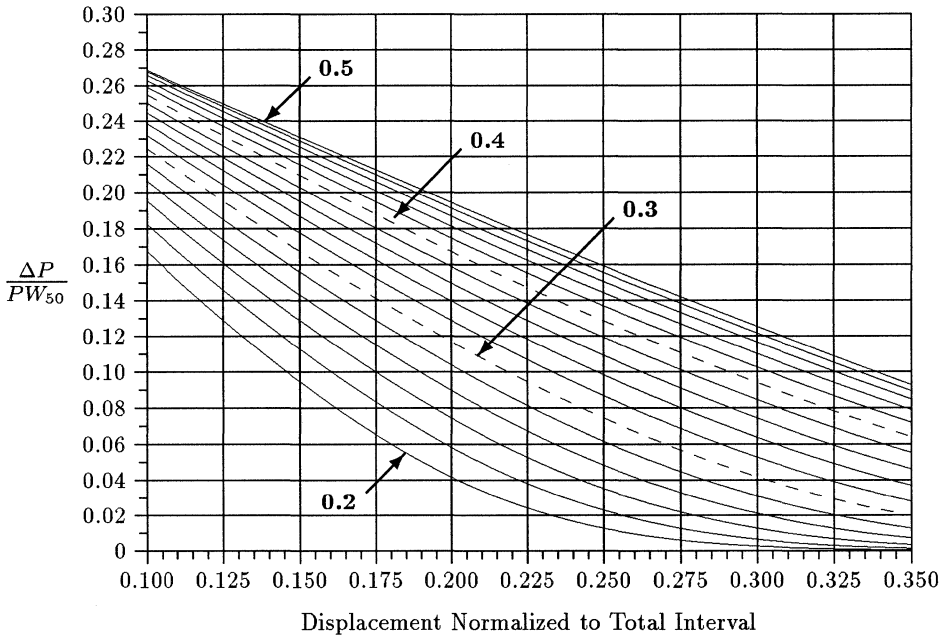

Figure B.4: Curves of $\mathcal{P}(t, \infty)$ Tripole Peak Shift, Group 2

B.3 Code and Model Specific Peak Shift Curves

The curves in this appendix show the values of peak shift for a variety of pulse models in standard encoding applications. Since each code has a characteristic set of allowable interval pairs there will be a finite set of peak shift modes. The values of peak shift for each mode have been plotted for increasing ratio of PW_{50} to the bit cell size. Peak shift magnitude is normalized to bit cell size for these curves.

To use the curves, determine the appropriate pulse model and code first. Given the PW_{50} of the isolated pulse on a cyllinder of interest, and the packing density (or bit rate) under consideration, the peak shift for each mode can be found on the corresponding mode curve. Simply find the value of $PW_{50}/Bitcell$ and look up the fractional peak shift for that curve. This can be converted to fractional displacement in the data recovery window by doubling the value on the graph.

Alternatively, the curves can be used to find the highest packing density consistent with a desired maximum peak shift. For this case simply reverse the process and start with the required limit on the vertical scale and look up the horizontal value on the mode curve of interest.

These curves are especially valuable for quantifying the worst case (maximum mode number) peak shifts, as the other modes are primarily of statistical interest.

Since the curves are computed using fixed interval pair ratios, they are not applicable for systems which use write pre-compensation. For those cases other curves in this paper are required.

B.3.1 Uncompensated Peak Shift Mode Curves for MFM

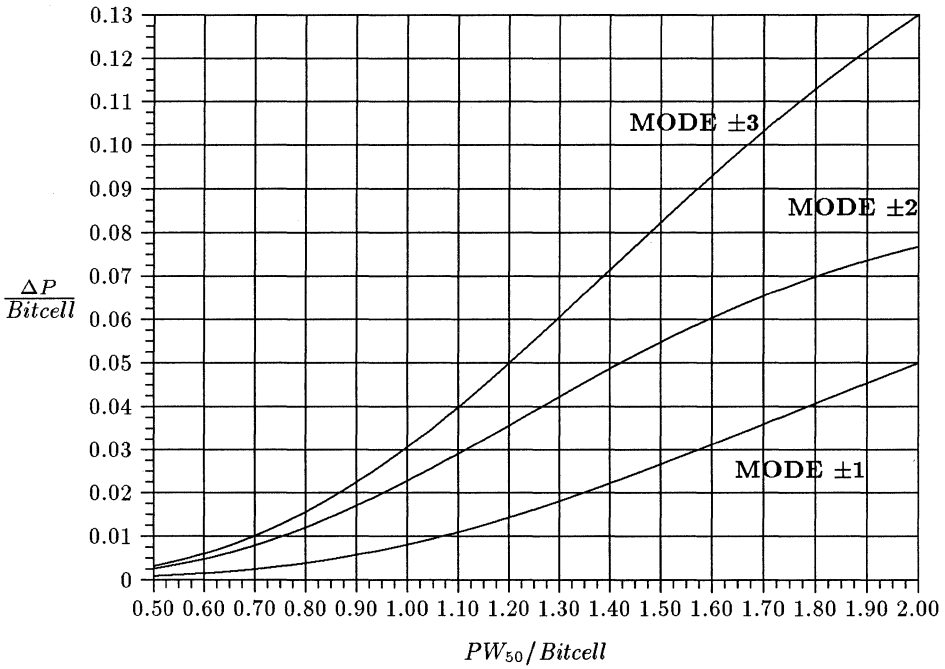

Figure B.5: Curves of $\mathcal{P}(t,1)$ Tripole Peak Shift for MFM

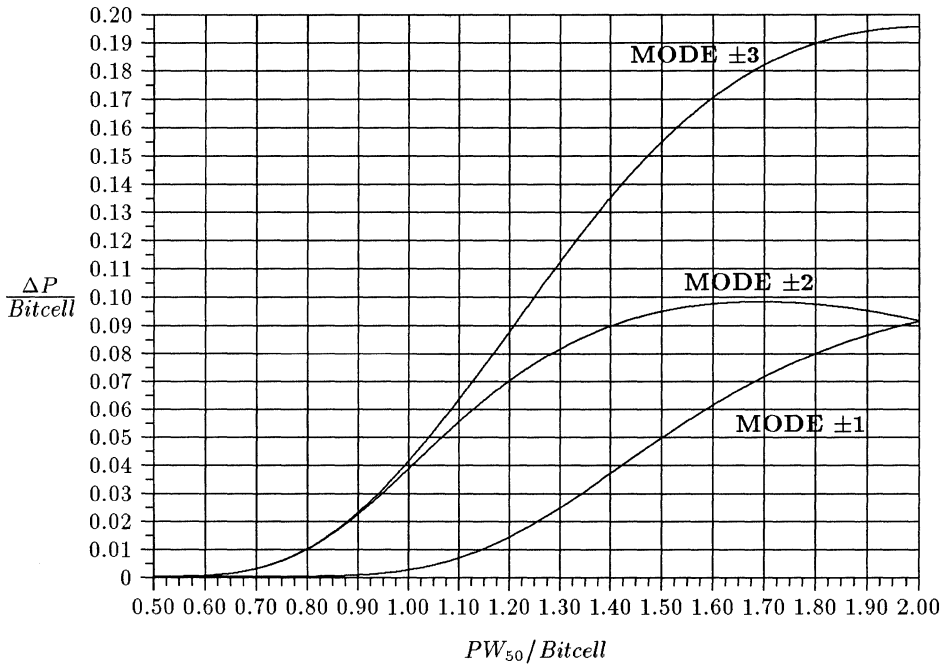

Figure B.6: Curves of $\mathcal{P}(t, \infty)$ Tripole Peak Shift for MFM