XEROX

Century Data Systems
A Xerox Company

"EVALUATION OF DISK DRIVES BY MARGIN ANALYSIS"

Nigel D Mackintosh, B Sc, Ph D *

Presented at:

International Peripherals and Software Exposition
Anaheim Convention Center
Anaheim, California
October 1 1982

* Century Data Systems, P O Box 3056, Anaheim, CA 92803-3056

EVALUATION OF DISK DRIVES BY MARGIN ANALYSIS

Nigel D Mackintosh, B.Sc., Ph.D.*

Introduction

This paper addresses one particular aspect of disk drive testing or evaluation, that of checking the error rate. The main reason for having a disk drive is that you can store information on it at one time and then, sometime later, you can hopefully recover that same information without error. So it is important not only to the end user of the disk drive but also to the systems integrator and even to the disk drive manufacturer to actually check the error rate performance of the drive. Let's review a typical disk drive specification, shown in Figure 1, and consider how we might check each item in the specification.

Disk Drive Specifications

Capacity: this might be something that you need to check, but once you have verified the accuracy of the statement there's not really much point in continually checking it for each disk drive of that type, as it's fixed by many of the other well-defined parameters of the drive. Similarly with the *data rate* - you obviously want to check if the controller can, in fact, handle the data rate, but again it is very unlikely that it will change from one disk drive to another of the same type from the same vendor.

Access time, however, is different - it is fairly dependent on the components in the drive (both mechanical and electrical) and, therefore, this is something you might want to check for each drive, particularly if you are the manufacturer of the drive. However, track to track access time may be only 5ms, average access time 50ms, and settling time 5ms. These are all fairly short times, and you don't need to check too many of these to be happy with the statistical average. The most lengthy test might be to check the average access time of 50ms: if you were to do 1000 of these and then take an average, you would have a test of 50 second duration, which is no problem.

Latency: again this is a fairly fixed parameter, just dependent on the rotational speed, and is very easily checked. *Lineal recording density*: this is usually specified, but to the drive user it's actually fairly irrelevant, as long as the error rates are acceptable, and similarly with *track density*. *Number of cylinders*, however, is fairly critical, but again it's one of those parameters that, once checked, isn't going to vary from drive to drive. *Recording method*, like the BPI, isn't of too much significance to the drive user.

*Manager of Recording Systems, Century Data Systems, P O Box 3056, Anaheim, CA 92803-3056. TEL : (714) 999-2552.

TYPICAL DISK DRIVE SPECIFICATION

* CAPACITY (unformatted)	80 MBytes
* DATA RATE	7.68 Mbits/sec
* ACCESS TIME – track to track	5 msec
– average	50 msec
– settling time	5 msec
* LATENCY (average)	8.3 msec
* ROTATIONAL SPEED	3600 RPM ±3%
* LINEAL RECORDING DENSITY	7,500 BPI
* TRACK DENSITY	700 TPI
* NUMBER of CYLINDERS	823
* RECORDING METHOD	MFM
* SEEK ERROR-RATE	1 in 10^6 seeks
* ERROR-RATE (soft)	1 in 10^{10} bits

Fig. 1

The *seek error rate* may be defined typically as less than one seek error in 10^6 seeks. 10^6 random seeks would take several hours to perform, and this test should really be repeated several times for a better average. However, experience has shown that the causes of seek errors usually have more of a catastrophic effect than a continuous one, meaning that usually a given drive either fails the seek error-rate specification miserably, or it exceeds it comfortably. This means that, in practice, a short test for this parameter is sufficient.

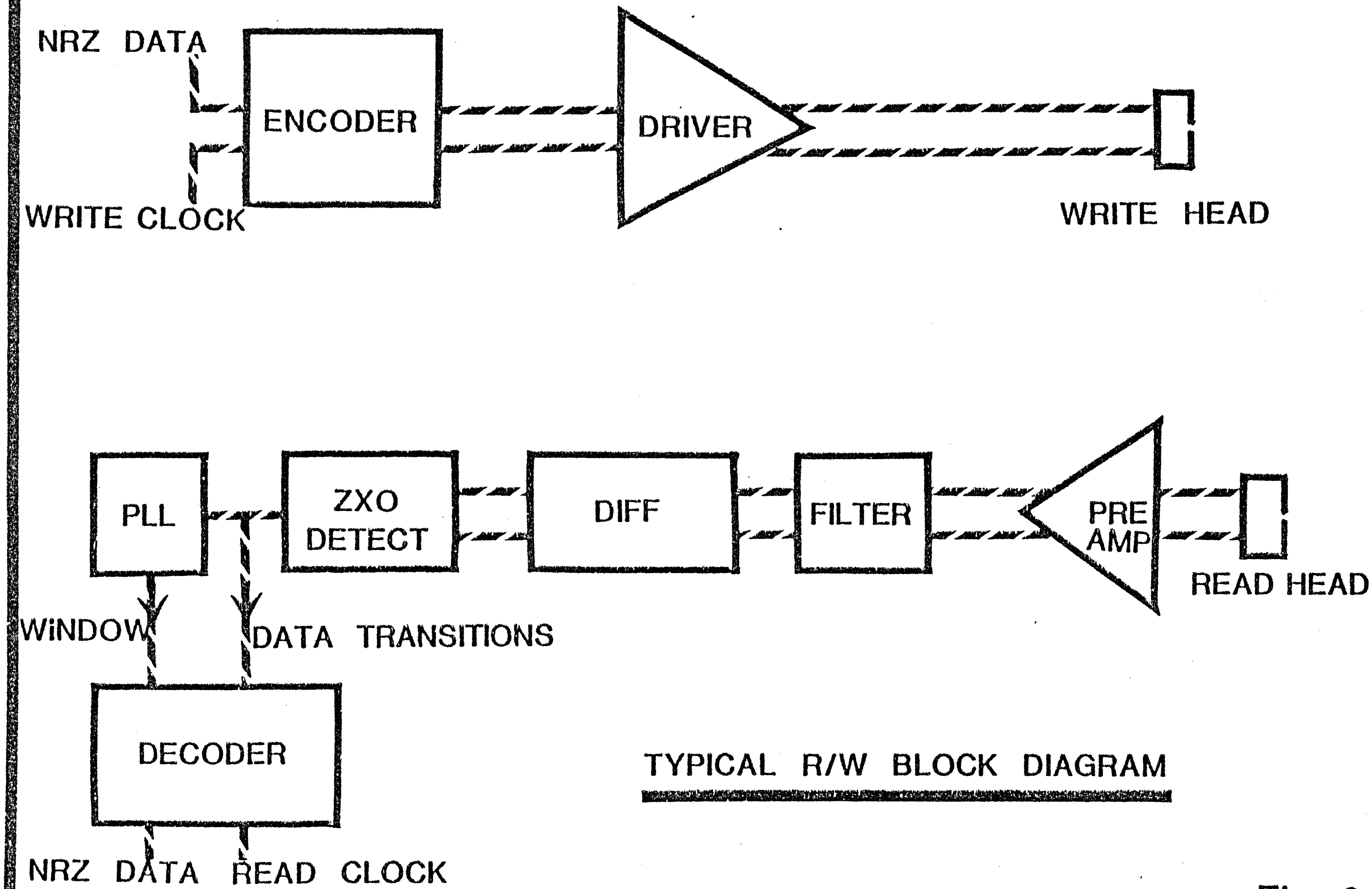
Then we have the *bit error rate*, sometimes called the raw or soft error rate, typically specified as better than one error in 10^{10} bits. Let's calculate how long it might take to test this particular parameter.

The average access time is 50 ms. Let's suppose one block of data is 256 bytes on average. Therefore, the average transfer rate during random seeks is 256 x 8 bits every 50 ms, which is 40 kilobits per second. Now, if we are checking an error rate specification of 1 in 10^{10} bits, we should statistically check for 10 times that many errors, and then average them. In other words, we should really check to see that we have less than 10 errors in 10^{11} bits transferred, so 10^{11} bits takes 10^{11}/40k seconds, which is approximately 2.5 x 10^6 seconds, or 30 days! So this is no mean feat. Just to check that one drive is operating within its bit error rate specification is going to take us 30 days. This is trouble enough to do just for one drive, maybe if you were comparing drives from different vendors for example, but for the manufacturer to do this for each drive he produces is obviously uneconomic. So the subject of this paper is various techniques for circumventing this problem of having to fully test the soft error rate specification on a disk drive. As an introduction to that, let's review what is really happening when we have an error in recovered data.

Data Recovery

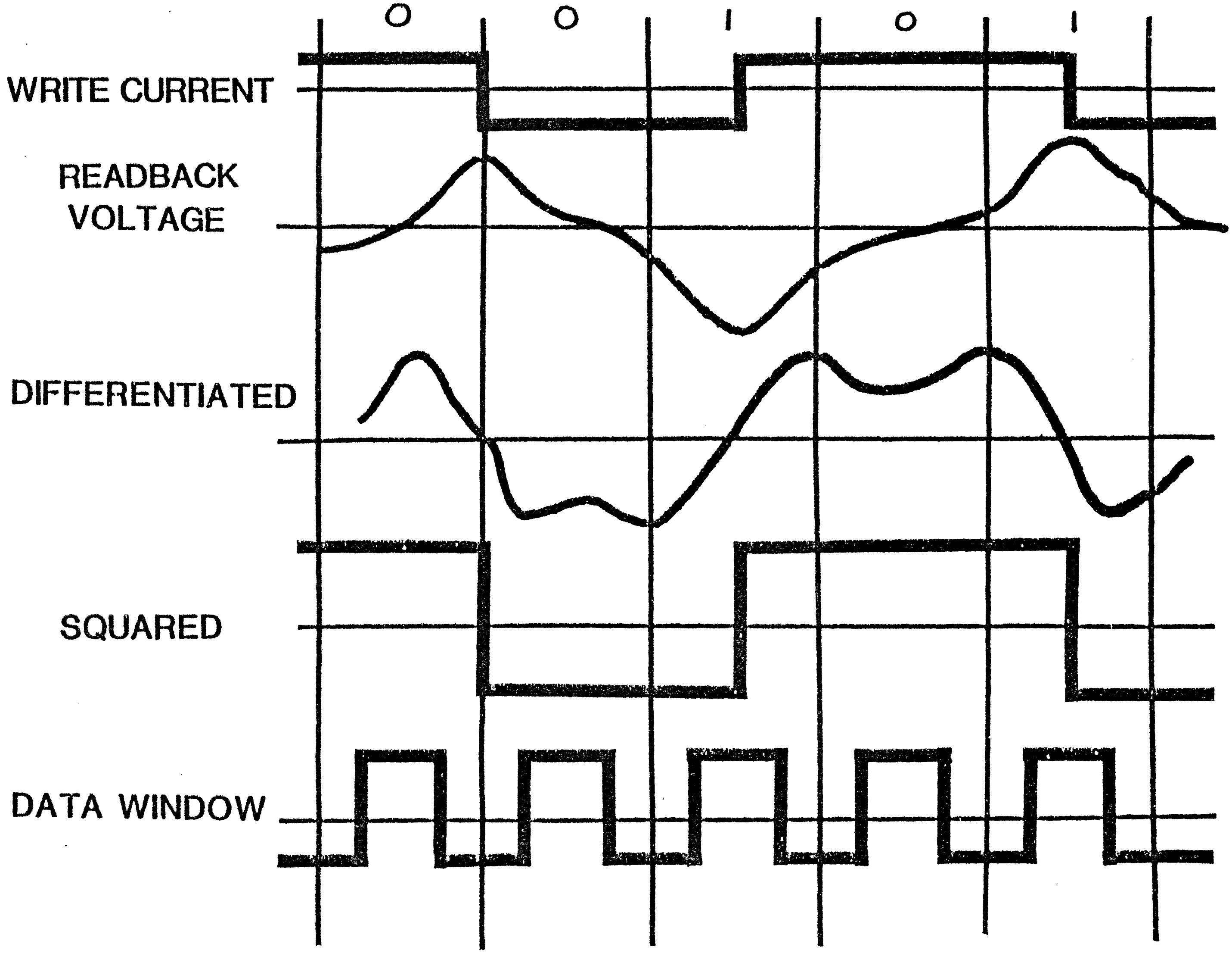
Figure 2 shows a typical read/write block diagram. The incoming NRZ data and write clock pass through an encoder, and the write driver, and are then presented to the write head for the appropriate flux transitions to be written on the disk. When we want to recover that data, the flux transitions excite the read head, and the resulting voltage is then passed through a preamplifier, a filter, a differentiator and a zero crossover detector, to form the 'data transitions'. At this point the data is split: it passes first into a phase lock loop. This creates a data window which is then used to clock the data transitions into a decoder, so that we can recover the NRZ data and send it, together with a read clock, to the controller.

Figure 3 explains this in more detail. Here I have defined five bits of data, the pattern being 00101. On the top, we have the write current according to MFM rules. MFM is one of the most common recording codes used today and is defined such that a data 'one' causes a change in the write current in the center of the bit cell. 'Zeros' do not cause a change in the write current except when there are two in succession, which produces a change in write current at the junction of the bit cells. So, whenever we have a change in the write current in the top waveform, we cause a flux transition, or a change in magnetization direction, on the disk. On readback then, each one of these flux transitions causes a pulse. The second waveform shows



TYPICAL R/W BLOCK DIAGRAM

Fig. 2



TYPICAL R/W WAVEFORMS

Fig. 3

a pulse in the readback for each transition in the original write current. In our quest to increase packing densities as much as possible on disk drives (and, therefore, the capacity of them), we find that we no longer see these pulses well defined and isolated, but in fact they tend to merge as shown here.

Remember that the peak of the pulse is really what we are trying to detect, as this represents the flux-transition on the disk, which in turn defines what data was written. The best way to detect peaks is to differentiate, so the third waveform shows the derivative of the readback voltage, and it is apparent that we now have zero crossings where we had peaks in the readback voltage, so it is now a simple matter to square this up and we have essentially recovered the original write current. From this we can deduce the data pattern that was written. We do this by determining whether these data transitions are occurring in the center of a bit cell or on the junction of the bit cells, so that we can determine whether each transition represents a one or a pair of zeros. The purpose of the phase lock loop in the read/write chain was, in fact, to provide us with a signal which delineates the center of a bit cell from the edges of the bit cell, and so the waveform at the bottom shows this signal, known as the read clock or data window. We might call the high portions of this waveform the 'ones window' because, if a transition occurs in the readback while this waveform is high, we know it is a 'one' that was recorded. It's common practice, actually, to only look for the ones transitions because if you look for one of those and there isn't one, you know by default that the data was in fact a zero.

Now, unfortunately, things are not quite as simple as I've made out here. Many things in the real world will cause the transitions that we recover not to be perfectly located either in the center or at the ends of the bit cell, as appropriate. For instance, there will be noise picked up from the disk along with the read signal. There will be noise introduced by the preamplifier. Interference of the readback pulses themselves will cause the waveform to distort and introduce what we call peak shift . The phase lock loop won't be perfect and might introduce some timing jitter. The delay through the read amplifier might not be constant for all frequencies, and this causes distortion. Additionally, we may get interference from previous data incompletely overwritten, or from data on adjacent tracks.

Transition Distribution Plots

Figure 4 shows the net result of all this imperfection. The horizontal scale here represents the data window for a disk drive with a bit cell of 100ns, using MFM code. The data window is thus 50ns in width. The vertical scale represents the probability density of any given transition occurring at any point in this data window, and we can see from the plot that the vast majority of them do in fact land perfectly in the center of the data window as we would expect. However, some of them fall just to the sides of the center of the data window, and as we go out more and more from the center of the data window we find that less and less of them occur. In fact what we have here is basically a gaussian (or 'normal') distribution, because the gaussian noise introduced by the disk and the preamplifier is usually the dominant cause of timing errors. On this linear probability scale used here, it would appear that there is very little chance of a transition falling outside the data window, and therefore there is very little chance of data being misread. However, the error

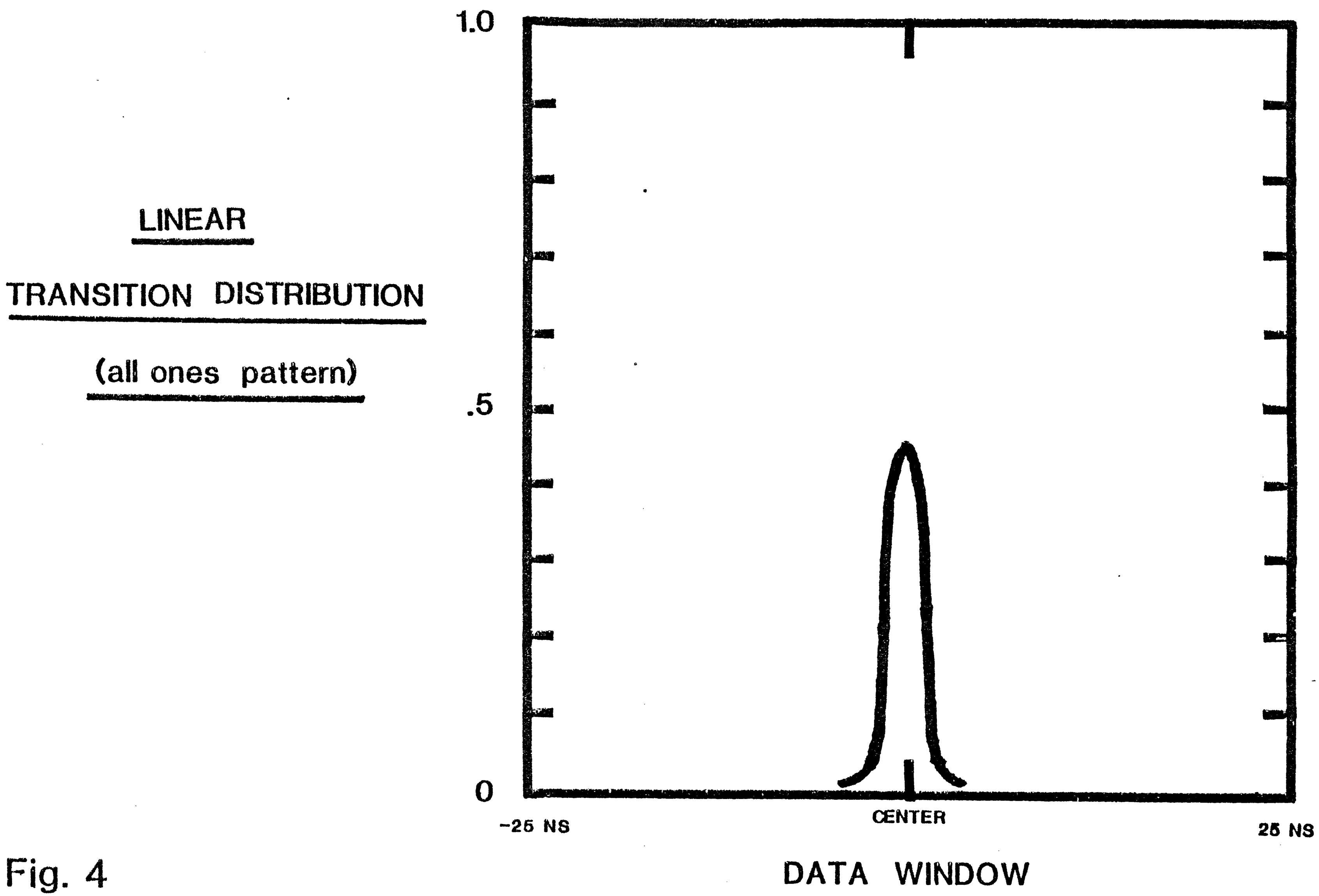

LINEAR
TRANSITION DISTRIBUTION
(all ones pattern)
1.0
.5
0
-25 NS
CENTER
25 NS
DATA WINDOW

Fig. 4

rate specification is only one error in every 10^{10} bits, or a probability of 10^{-10}, and we cannot see this very easily on this linear scale.

So Figure 5 shows exactly the same distribution with the vertical scale this time being logarithmic, and covering the range from 10^0 (or unity) down to 10^{-10}. Now we have a much clearer picture of the transition distribution. We can see that once in every 10^{10} bits a transition will actually occur 10ns away from the center of the window on each side. However, this still means that we have 15ns of margin on each side before we should get a data error. The pattern I used for this particular test was an all-ones pattern, in other words MFM 111----11.

Figure 6 shows the same type of distribution, for the same pattern, but from a different head with a different signal to noise ratio. This could be caused by either a different set of electronics, more noise in the preamplifier, a different radius, or just a lower output head. We can now see a much wider distribution, and in fact some transitions are occurring over 15ns away from the center, so that we might now define our margin at the 10^{-10} level to be about 8ns on each side. Notice how the slope of the curve has been affected by the reduced signal to noise ratio. We now have a more gradual slope on the sides of the curve.

Figure 7 shows the distribution for what we call a 'peak shift' pattern, or an MFM 110110 repeating pattern. In this pattern, the two transitions in the pair tend to oppose each other and push each other apart, such that they have a fixed shift in their peaks, one to the left and one to the right, and we can actually see this now in the resulting transition distribution. Remember what this graph is showing us. Essentially what we have done is transferred an enormous number of bits, maybe 10^{11} bits, and we have looked at each one in turn and determined where exactly in the data window it fell. We have then plotted each one, and we have ended up with this distribution showing that many of the transitions fell 7ns to the left of the center and many of them also fell 7ns to the right of the center. This is obviously the amount of the peak shift introduced by this particular pattern. What we have then is the normal gaussian distribution about each of these two nominal positions. The net effect is a bi-modal distribution clearly showing the amount of the peak shift, as well as the signal to noise ratio (shown by the slope of the curve).

Notice that the basic distribution here corresponds to Figure 5, showing this to be the head with the good signal to noise ratio. If you can picture what would happen with the distribution of Figure 6 (the wide one) under these conditions of peak shift, there would probably be no margin at all left, whereas here we still have about 7ns of margin on each side, at the 10^{-10} level. If you were to transfer a random set of data, instead of these fixed patterns, you would get the summation of many different transition distribution plots, and, as shown in Figure 8, the net effect is a wide, flat-topped distribution, which is essentially limited on the edges by the worst case peak-shift plot.

Now that we understand what's actually happening within the data window on recovery, let's recap. Ideally we want all the recovered data transitions to fall in the middle of the data window. We can actually tolerate them moving about within this

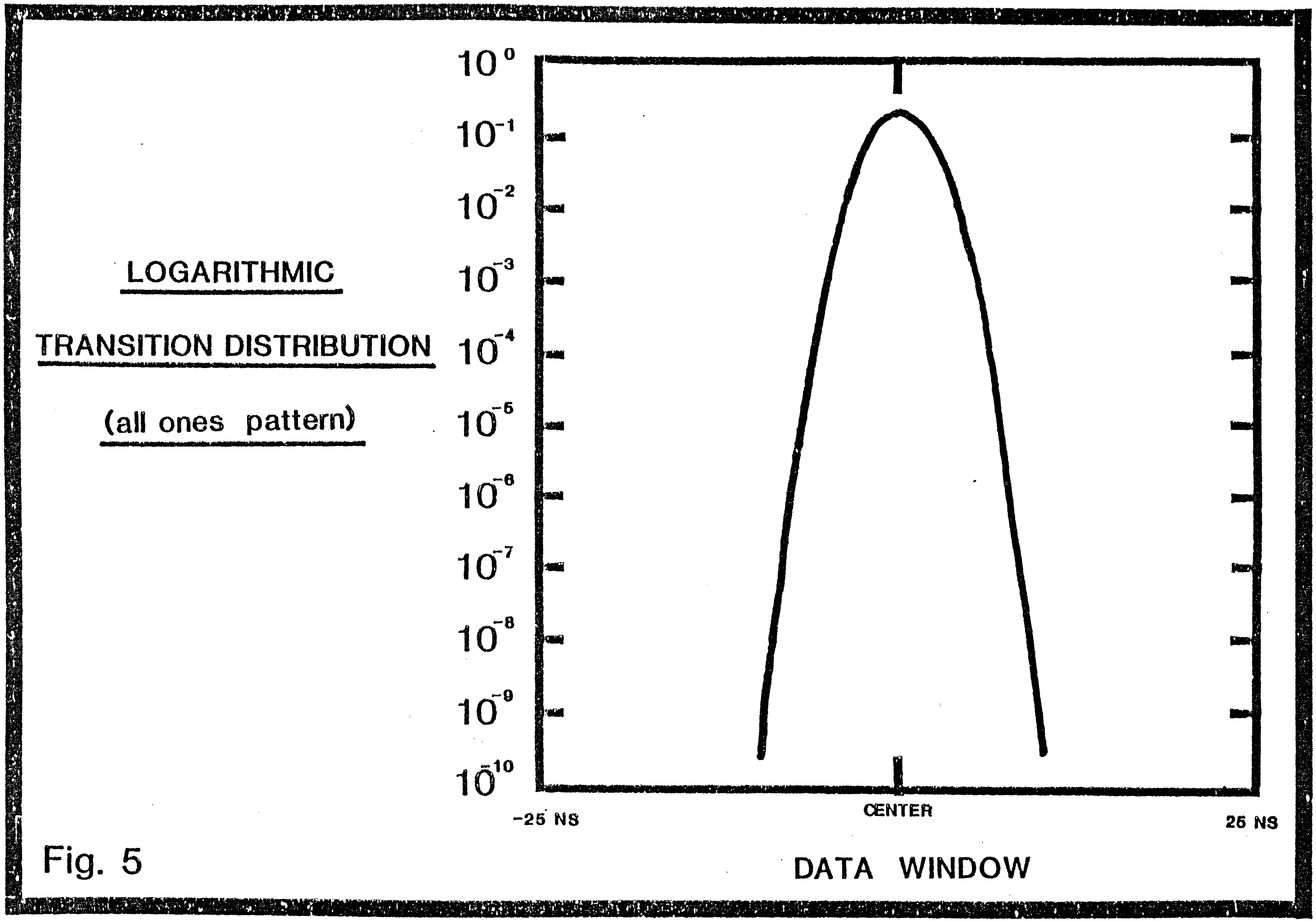

LOGARITHMIC
TRANSITION DISTRIBUTION
(all ones pattern)
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
-25 NS
CENTER
25 NS
DATA WINDOW

Fig. 5

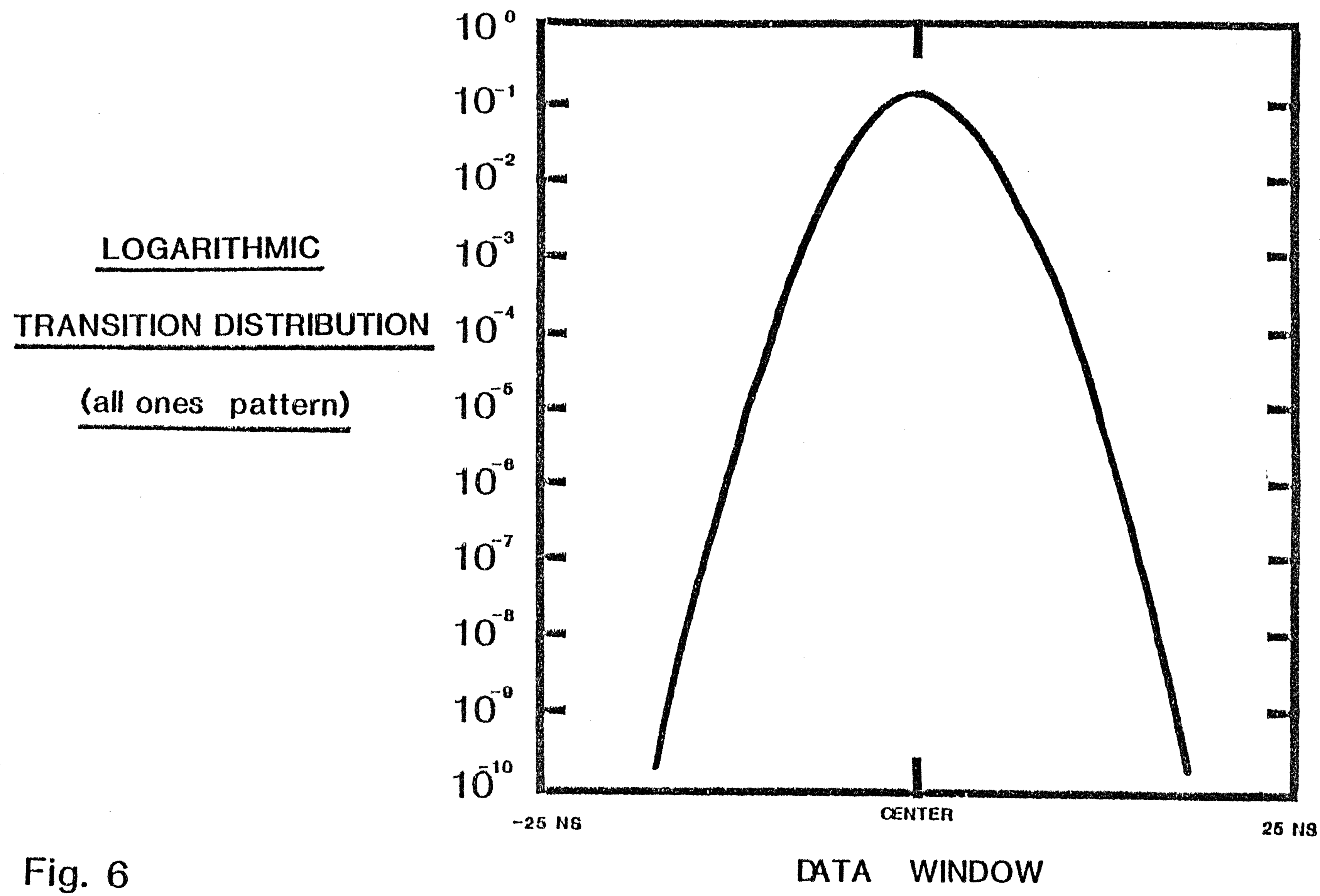

LOGARITHMIC
TRANSITION DISTRIBUTION
(all ones pattern)
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
10^-9
10^-10
-25 NS
CENTER
25 NS
DATA WINDOW

Fig. 6

LOGARITHMIC

TRANSITION DISTRIBUTION

(peak shift pattern)

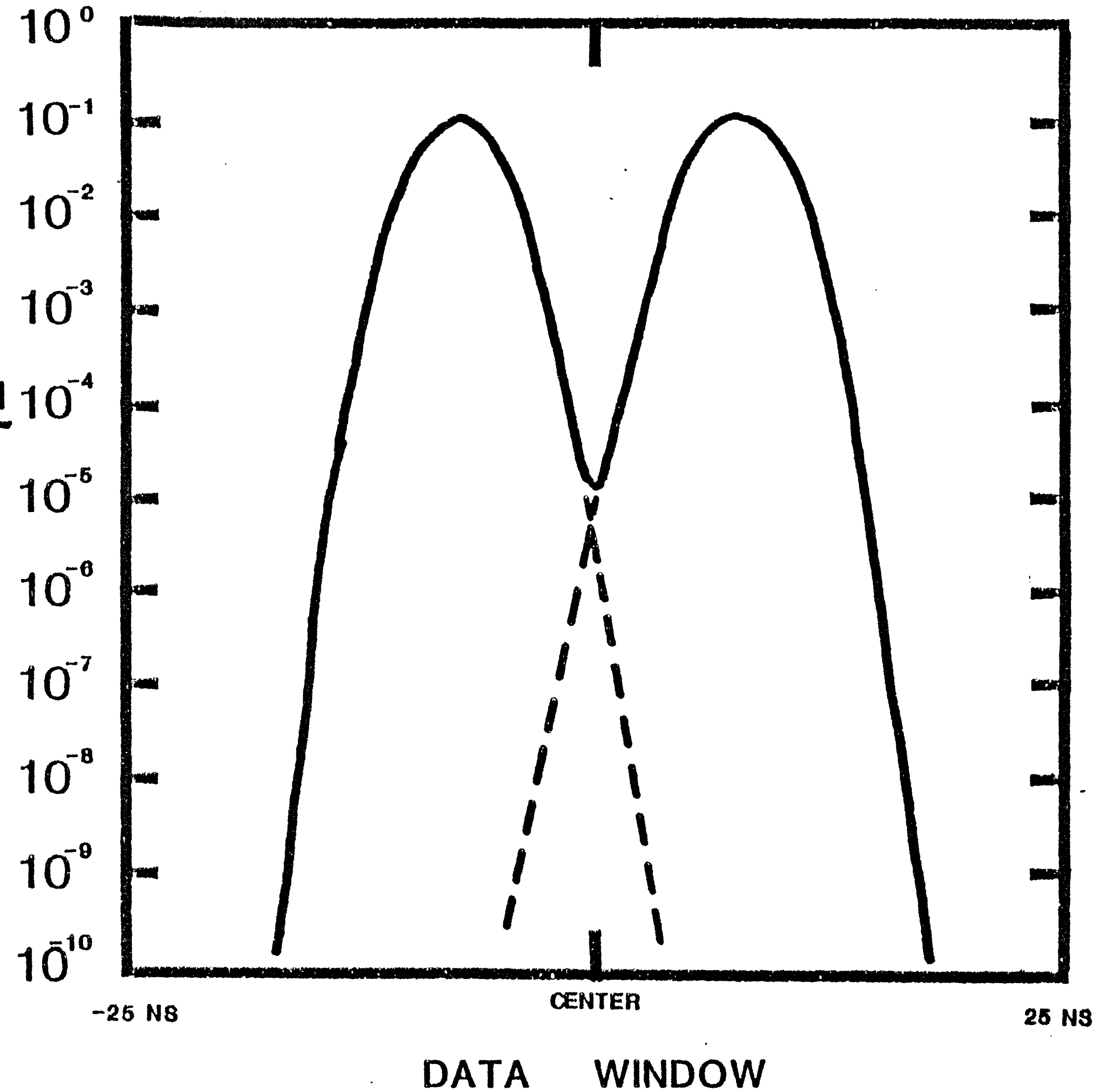


Fig. 7

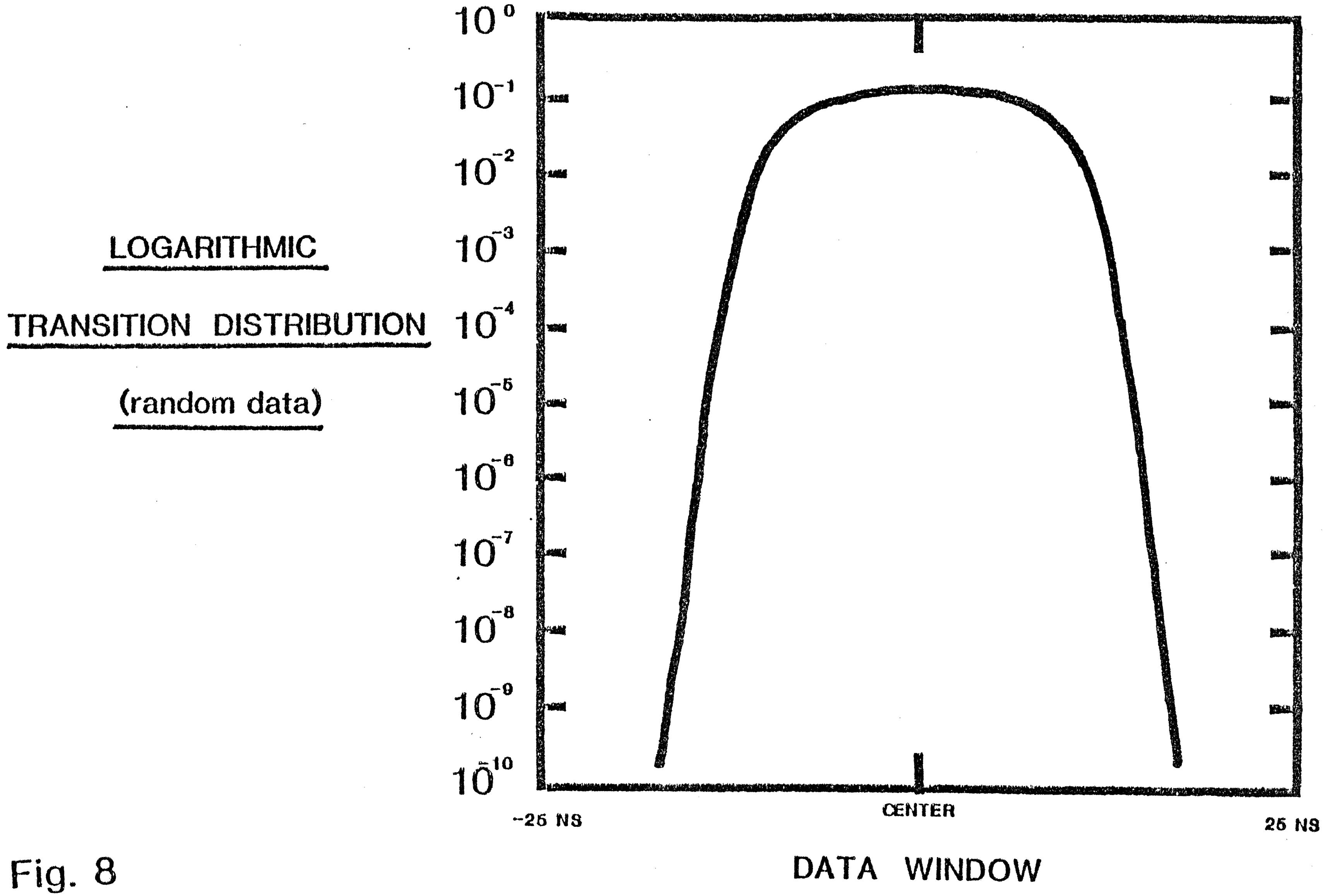

LOGARITHMIC
TRANSITION DISTRIBUTION
(random data)
10⁰
10⁻¹
10⁻²
10⁻³
10⁻⁴
10⁻⁵
10⁻⁶
10⁻⁷
10⁻⁸
10⁻⁹
10⁻¹⁰
-25 NS
CENTER
25 NS
DATA WINDOW

Fig. 8

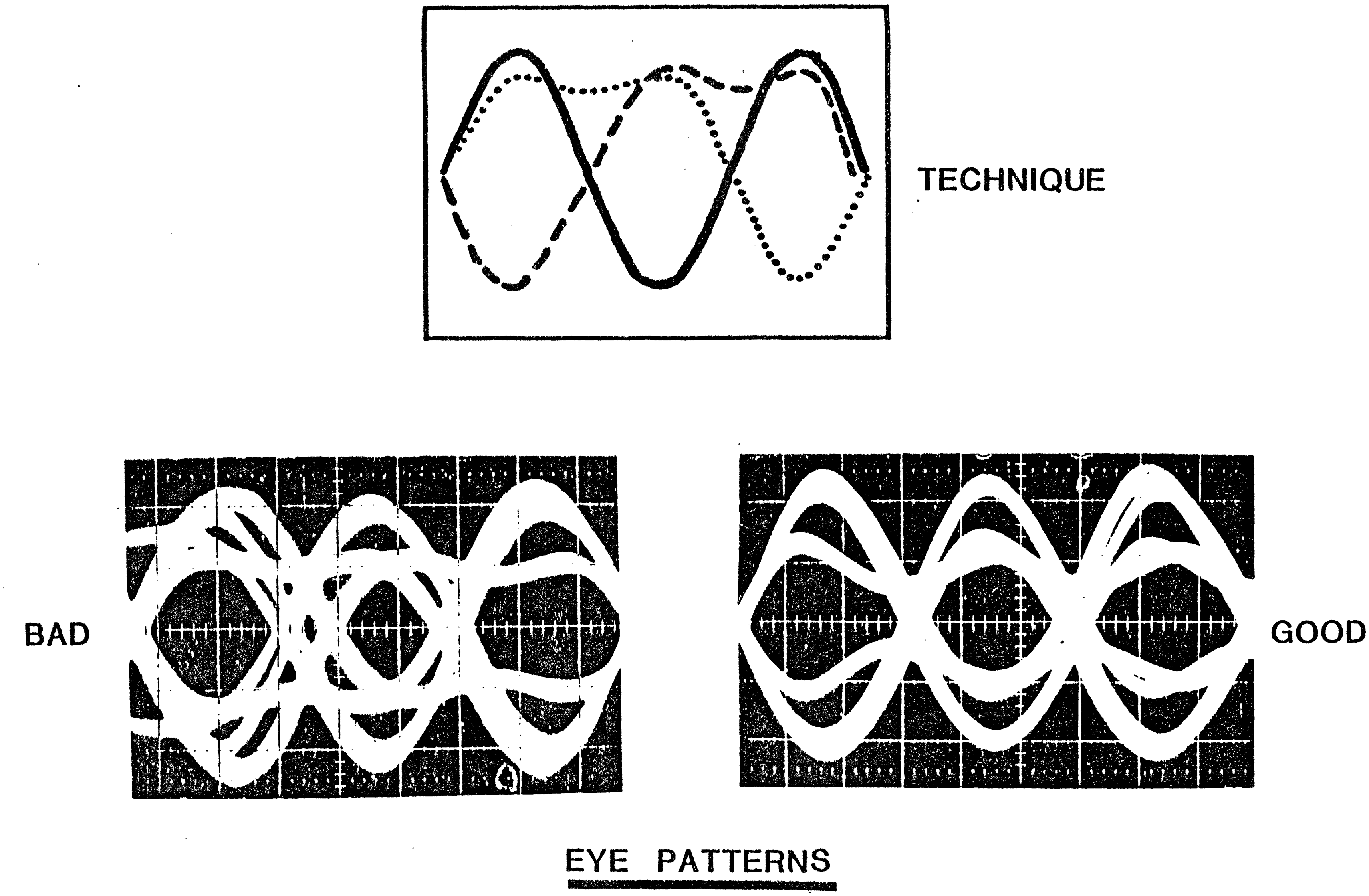


Fig. 9

window as long as they don't jump outside the window, because that would cause an error. So we would like to see them keeping away from the edges of the window, and therefore the more margin we have in one of these plots at the edge of the window, the happier we will feel, and the more easily we will meet the error rate specification.

Margin Analysis

Now lets look at four different ways of using this information to estimate the normal error rate without having to wait 30 days for the answer :

(a) Eye Pattern

The first method is called the eye pattern technique. In this method we use an oscilliscope to look at the differentiated readback signal. We trigger the scope from the zero crossings and examine the subsequent zero crossings. In Figure 9 I have superimposed three different patterns of bits which might occur when transferring random data. On a drive with good margins i.e., the transitions are occurring fairly accurately and consistently at a certain point in time, we see the right hand waveform, where the 'eyes', if you will, are very large. However, for the drive with bad margins, shown on the left, the succeeding zero crossings are occurring at differing points in time. The net effect is that the 'eyes' are rather closed. This is by far the simplest technique I will describe, as well as the least accurate, but it does give a good, quick method of analyzing a drive's performance without resorting to any fancy equipment whatsoever.

(b) Window Sliding

Figure 10 shows again the relationship between the data window and the data transitions, as well as a typical transition distribution plot for a peak shift pattern. We can see from the plot where exactly the transitions are falling within the data window. The probability of any transitions occurring right at the edges of the window is extremely small; in fact the probability is much less than 10^{-10}. In other words, if we were to transfer 10^{10} bits of data we would not expect to see any bits at all falling outside the window. But what happens if we keep everything else the same, but we slide the data window along slightly? Here I've shown it slid along by 15ns to the right, and now it is apparent that many transitions will fall outside the left hand edge of the window. In fact the probability from the plot is roughly 10^{-2}; in other words, for every 100 bits transferred, you would expect one to fall outside the window, on average.

What we have then is a technique for artifically increasing the error rate. The way in which you might use this technique, therefore, is as follows: if the position of the data window is easily alterable in the disk drive, perhaps by a potentiometer, set the disk drive transferring

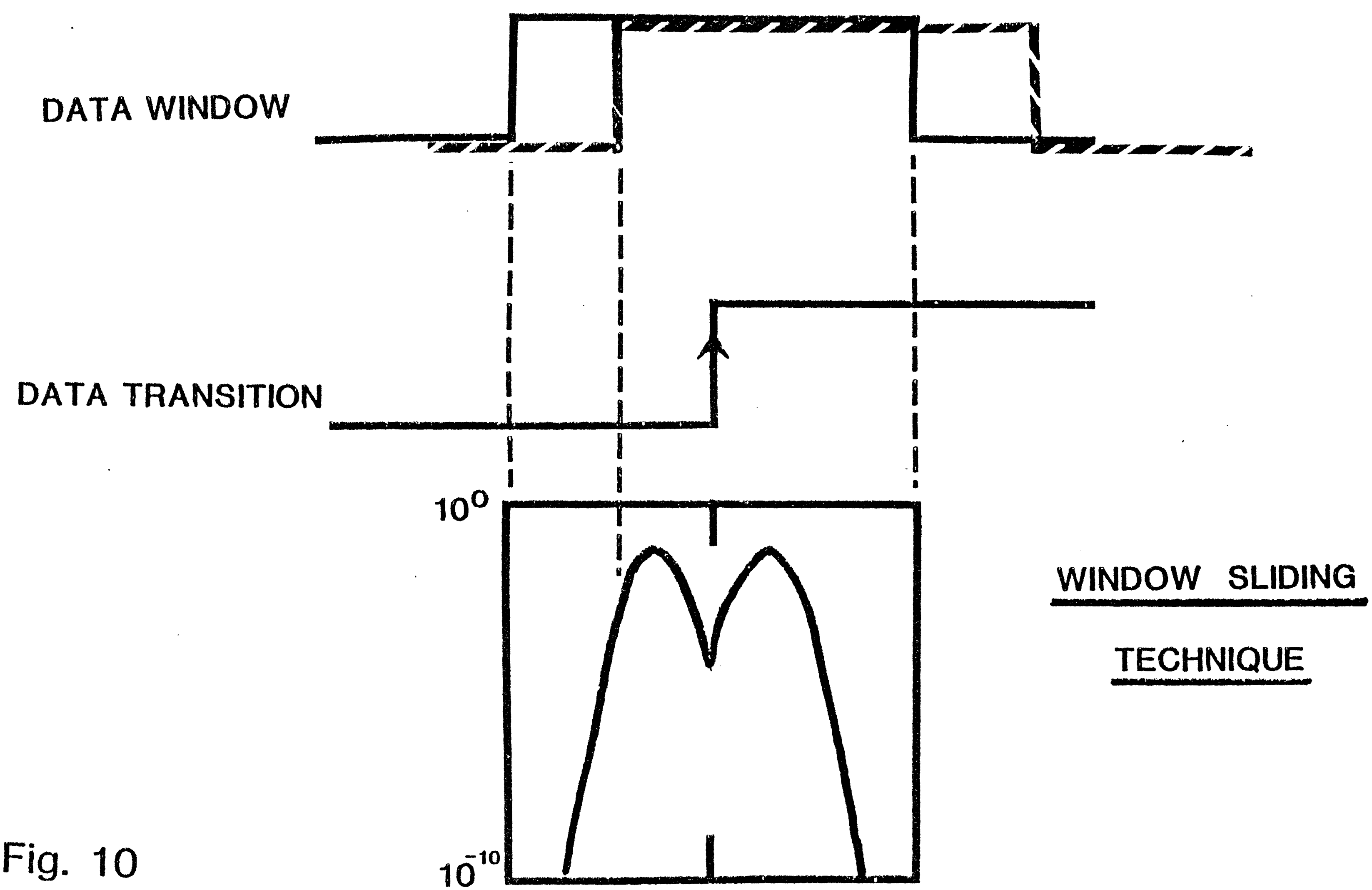

DATA WINDOW
DATA TRANSITION
10^0
10^-10
WINDOW SLIDING
TECHNIQUE

Fig. 10

data continually and then adjust the potentiometer and slide the window along until a fair number of errors occurs, for example one a second. Depending upon the particular test in use, you might be just sitting on one track transferring data, or perhaps randomly seeking. This might represent an error rate of one in 10^6 or 10^7, or a probability of 10^{-6} or 10^{-7}. By measuring how far you have slid the window, using an oscilloscope for example, you would then know how much margin this disk drive has at the particular error rate that has resulted. For instance you might now be able to say that this drive has a margin of 10ns on the left hand side, at an error rate of 10^{-7}. Knowing the typical slope of the transition distribution plot, you could then extrapolate and deduce that, at the 10^{-10} error rate, this drive has about 8ns of margin. The window could then be slid in the opposite direction and a similar calculation performed.

This technique is slightly more sophisticated than the previous one described, in that it does give you a quantitative measure of the margins. However, it is not always possible to slide the window in a disk drive and, even when it is, you are never sure how you are disturbing the operation of the drive by sliding the window backwards and forwards. After all, the drive has been designed, hopefully, for the data transitions to fall in the center of the data window. If you artificially alter this relationship, then, depending on the exact circuit details, it is possible that you are interfering with the normal operation of the phase lock loop, for example, and this could mean that the data you obtain is invalid.

Nevertheless, the technique is a useful one, and in fact by moving the data window in increments and measuring the error rate each time, it is possible to generate a transition distribution plot similar to the one shown here. However, this method is not capable of producing the exact one I've shown, because once you got to the error rate corresponding to the peaks in this plot and shifted the window further and further from that point, the error rate would not in fact get better, as suggested in this plot, since the peak of the distribution will remain the dominant cause of errors. As I mentioned, it is possible to design the disk drive to accommodate this window sliding technique of margin analysis. However, it may prove uneconomic, or it may be unsuitable for other reasons. In this case, the technique can still be used, but it must be done off-line, as will be described next.

(c) Window Board

Recalling Figure 2, the typical read/write block diagram, we see that the data window and the data transitions go into the decoder. If these two signals can be brought off the board, as shown in Figure 11, then we can create a dedicated variable-window decoder, specifically for the purpose of doing margin analysis. As this same board is going to be used to test all the disk drives, it can be as complicated and expensive as required. The controller that is handling this test knows to ignore the read data and read clock being sent from the drive's normal read channel, and instead it takes them from this variable-window decoder. The whole technique can indeed be automated by the controller, so that it slides the windows in increments and measures the error rate in each case. This exact technique has in fact been used by Century Data Systems on all its products for several years to insure that all products shipped meet an acceptable margin.

One slight disadvantage of this technique (which is shared by the other techniques I will describe) is that the normal decoder is not being used, which means that any imperfections in it are not being tested, and, as a corollary to this, the circuitry that is being used to take the two signals off the board may introduce its own imperfections. Remember that what we are measuring essentially is the phase relationship between the data window and the data transitions. As I mentioned earlier, however, this technique does not give the complete picture of the transition distribution. So, lastly, I will describe what may be the ideal method of margin analysis.

(d) Transition Distribution Analysis

If we look at a typical transition distribution plot, such as Figure 7, consider how we might obtain such a plot. Suppose we divide the data window into 1ns increments - in this case there would be 50 of them. We might consider these as 50 buckets or bins. Let's say we then transfer 10^{11} bits of data and, everytime a transition occurs, we see which bin it falls in, and we add one to the count for that bin. At the end of the test, we would find that all the bins towards the center of the data window were quite full, and, in fact, in this example, they would each contain a count of about 10^{10}. The bins towards the edges of the window, however, wouldn't have seen much action. Clearly, if we plotted a simple histogram based on the contents of each one of these bins, we would then produce the plot as shown, and we would plainly see the peaks in the distribution.

There are actually many different ways of implementing this particular technique, as described in Reference 4. I have designed and constructed one of these 'passive' margin analyzers, as I call them, and use it regularly in the design of disk drives in my work. As an example of the enormous use of such a tool in the Engineering phase of a new product, the final figures will show some miscellaneous plots I have obtained.

Typical Transition Distribution Plots

Figure 12 shows a normal peak shift distribution (solid line). The peak shift is well defined (approximately 10ns on each side), and we can see from the slope of the fall-off that the signal-to-noise ratio is fairly good. This situation lends itself ideally to

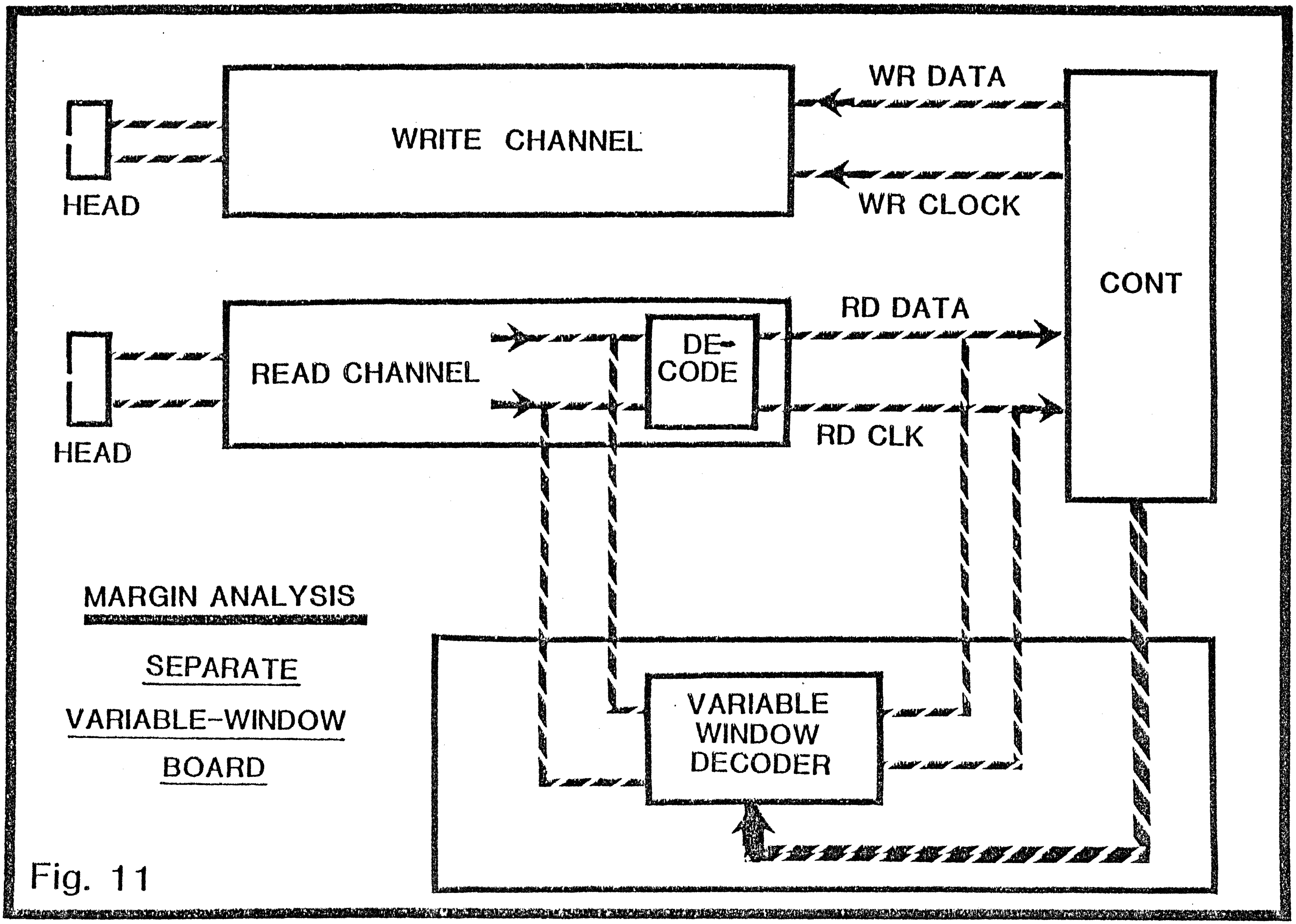

WRITE CHANNEL
WR DATA
WR CLOCK
HEAD
CONT
READ CHANNEL
DE-
CODE
RD DATA
RD CLK
HEAD
MARGIN ANALYSIS
SEPARATE
VARIABLE-WINDOW
BOARD
VARIABLE
WINDOW
DECODER

Fig. 11

SUCCESSFUL

PULSE SLIMMING

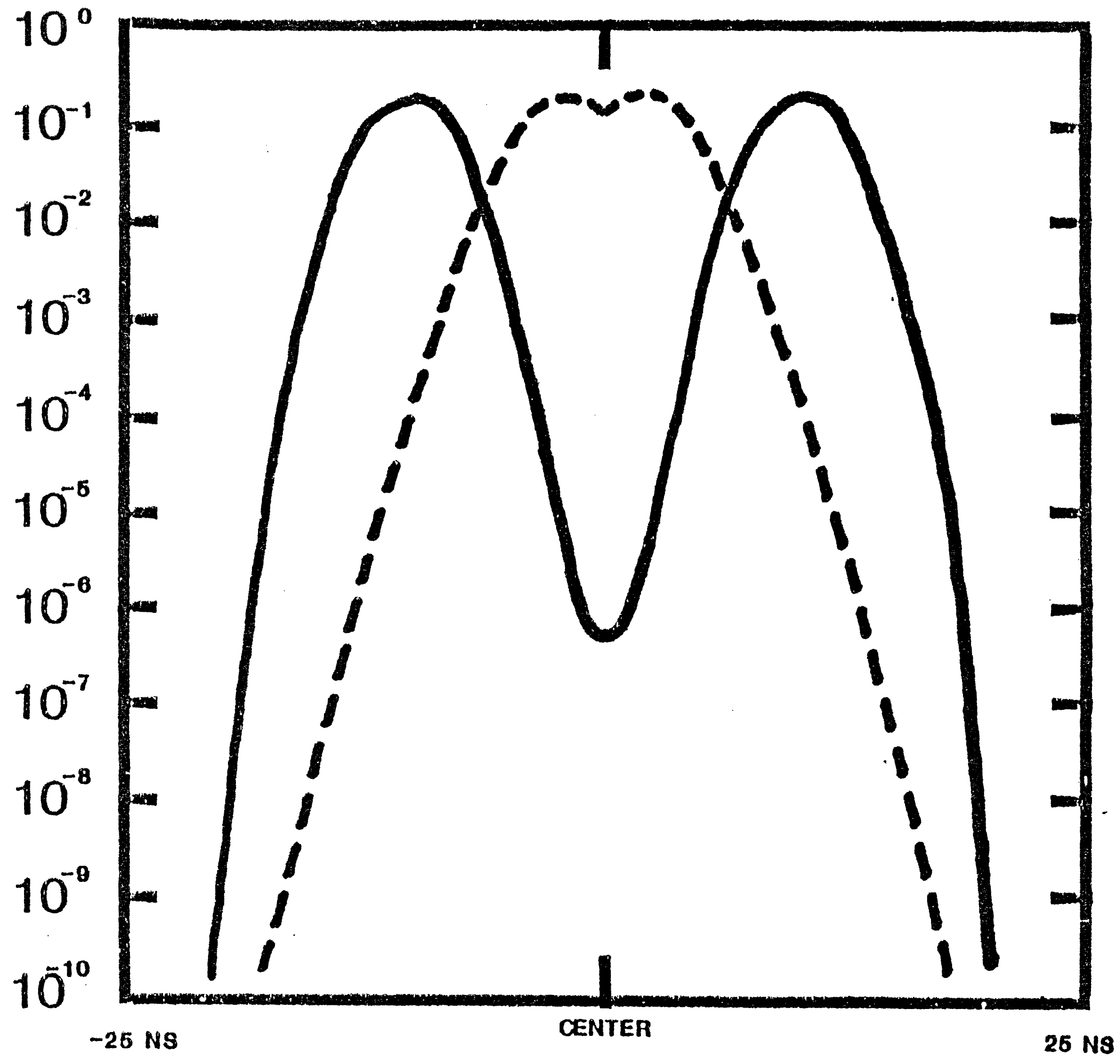


Fig. 12

pulse-slimming (see reference 2). Basically, what happens in pulse-slimming is that the peak shift is improved (reduced) and the signal-to-noise ratio is degraded, so the second plot shows that the peak shift has been reduced to approximately 2.5ns on each side, as compared to the original 10ns, but the slope of the fall-off is now less steep, meaning we have indeed degraded the signal-to-noise ratio. However, the net effect of this at the 10^{-10} level is an increase in the margin from 5ns on each side to approximately 8ns, and therefore the technique has been successful.

Figure 13 shows unsuccessful application of pulse-slimming. Here we have started out with only 5ns of peak-shift on each side and a fairly poor signal-to-noise ratio, producing the same result as before in that the margin at the 10^{-10} level is 5ns on each side. However, the pulse-slimming has reduced the peak shift from 5ns to about 2ns, and has again degraded the signal-to-noise ratio. However, the overall result is a loss in margin at the 10^{-10} level, from 5ns to 3ns.

Figure 14 shows how the margin analyzer can be used to check the centering of the transitions within the data window. Here again we have a peak-shift pattern, and in the solid plot the margin on the left hand side of the window is 5ns, while on the right hand side of the window it is almost 15ns. Clearly the 5ns would be the weak link in the chain, i.e., the main contributor to error-rate. When correctly centered as shown in the dotted plot, we have about 8ns of margin on each side, and this is clearly the optimum case.

Figure 15 shows how one must be careful to avoid end effects when using this technique. The distribution is supposedly for an all-ones pattern, but we can see that, appended to it, there is a smaller similar distribution offset from the center by about 9ns. This was caused by the first 'one' in the data: it did not have a preamble of all ones in front of it, but a preamble of zeros, so it did not see a symmetrical pattern on each side of it. The net result was peak-shift of about 9ns to the left on that one particular bit. So one bit in the 256 transferred in this particular test shows up with peak-shift, and with a probability of roughly one in 250; in other words, about two decades below the normal plot.

Figure 16 again demonstrates end defects ,where we have a peak-shift pattern, but again one of the end bits does not find itself in a peak-shifted situation. This time it finds itself surrounded by a symmetrical pattern on each side, and therefore it does not suffer any peak-shift, and appears in the plot as a uni-modal distribution, right in the center of the data window.

Finally, Figure 17 shows how off-track pick-up can affect the margins. The solid plot shows a peak-shift pattern with the heads set on track, and we have a margin of about 8ns on each side. However the head is then moved off the track by about 300 microinches into an interfering pattern, so the net effect is a degraded signal-to-noise ratio, and in this case we are now left with a margin of only 2ns on each side.

Conclusion

In conclusion, several different types of margin analyzers have been described, and it is clear that even a simple type can give very useful information about the operation

UNSUCCESSFUL

PULSE SLIMMING

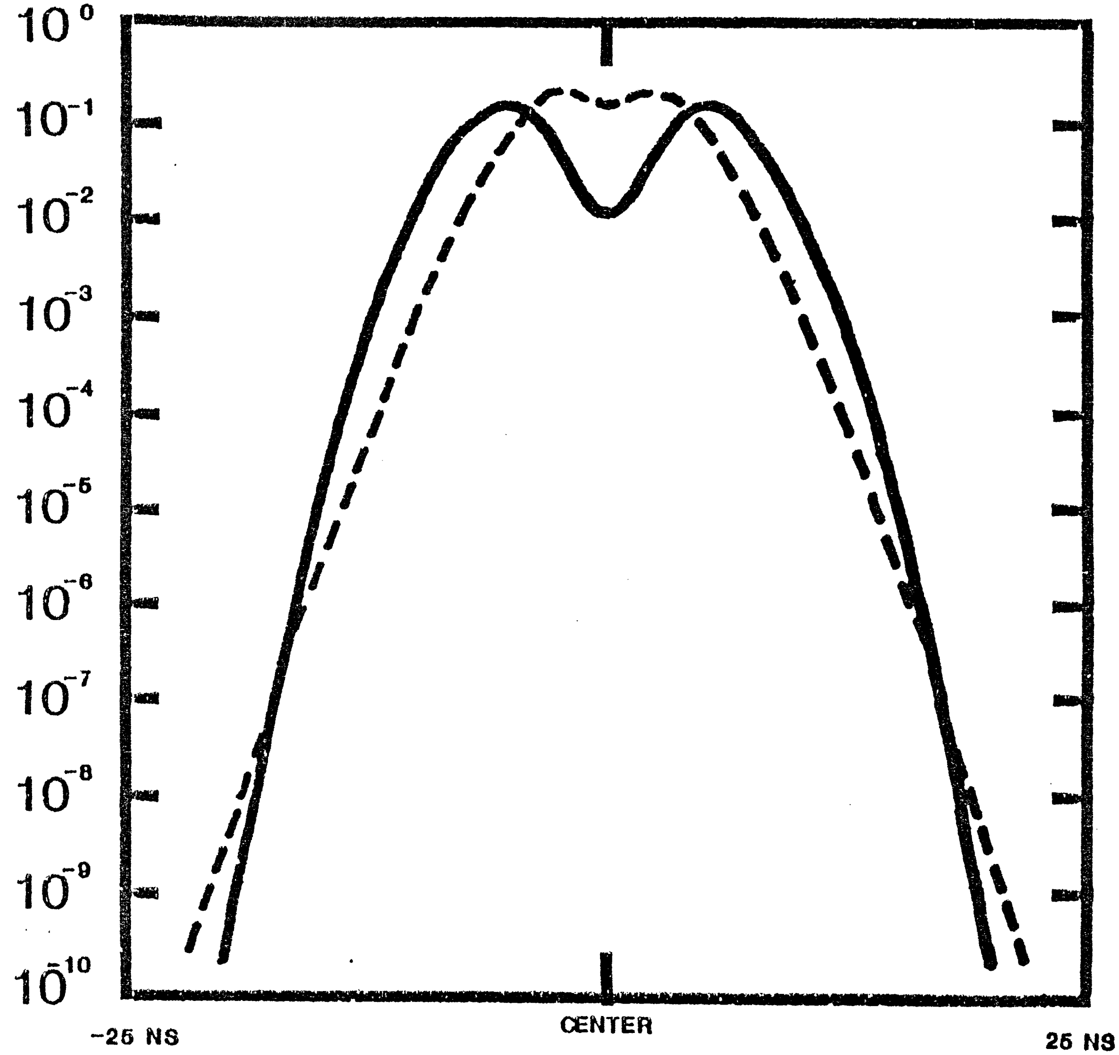


Fig. 13

INCORRECT CENTERING

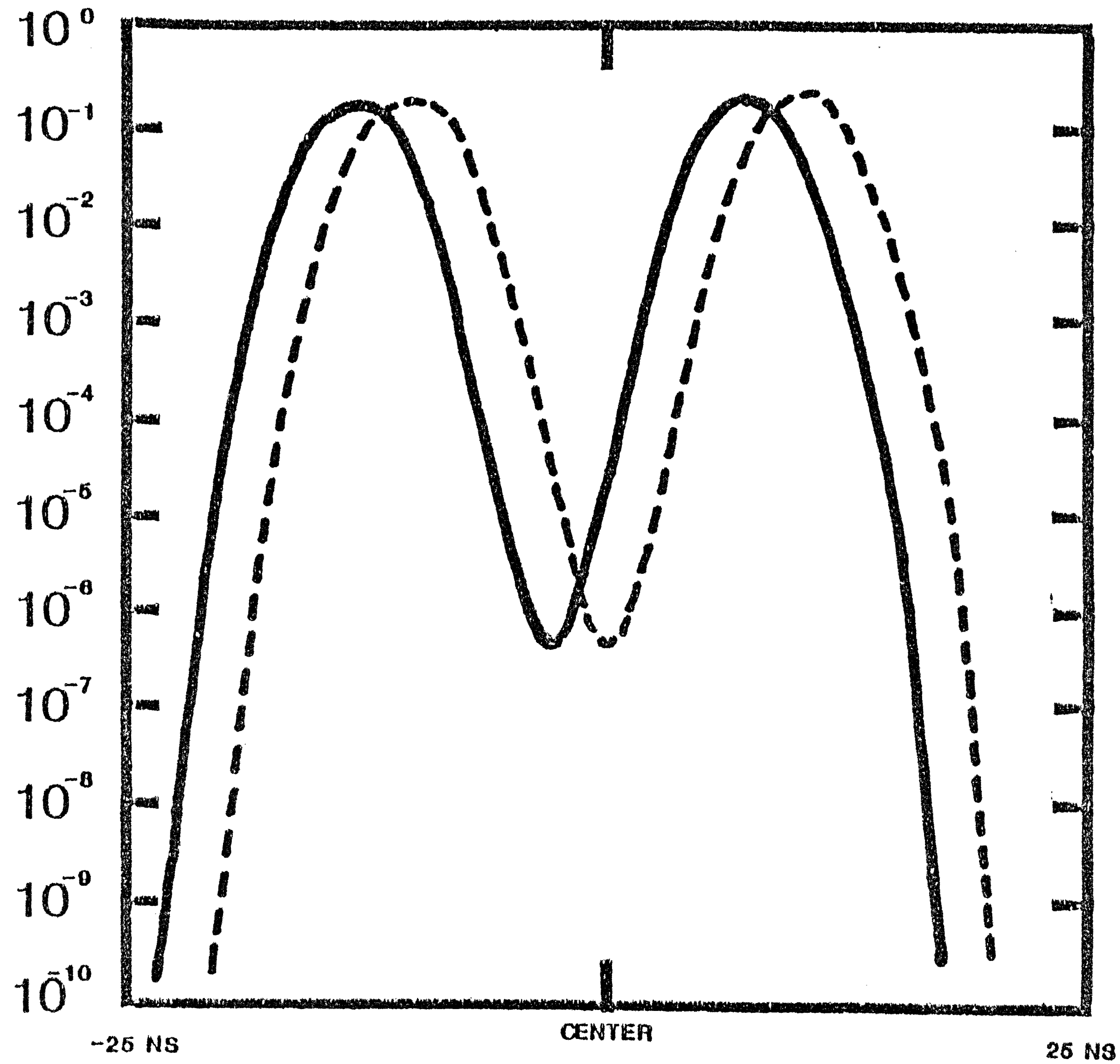


Fig. 14

END EFFECTS

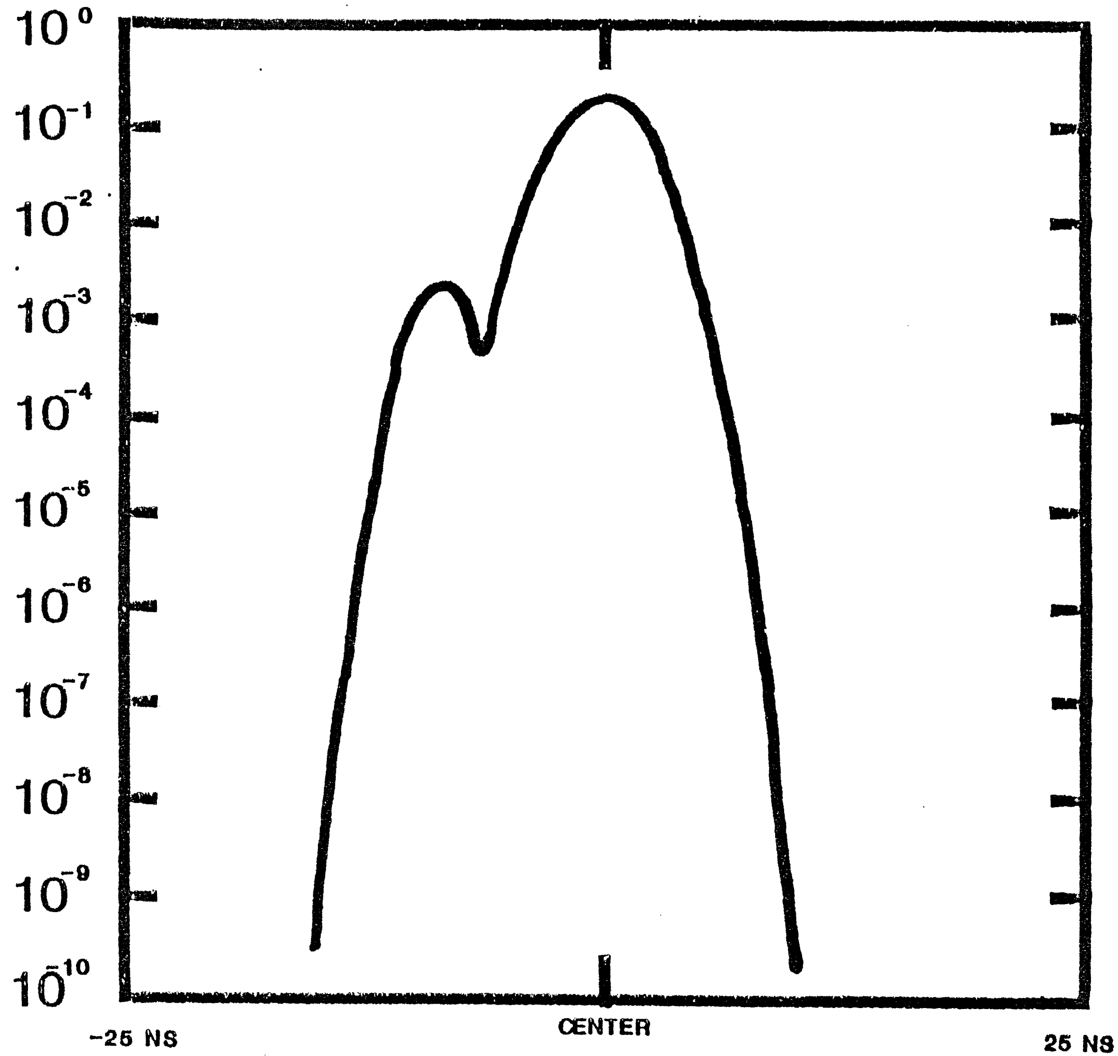


Fig. 15

END EFFECTS

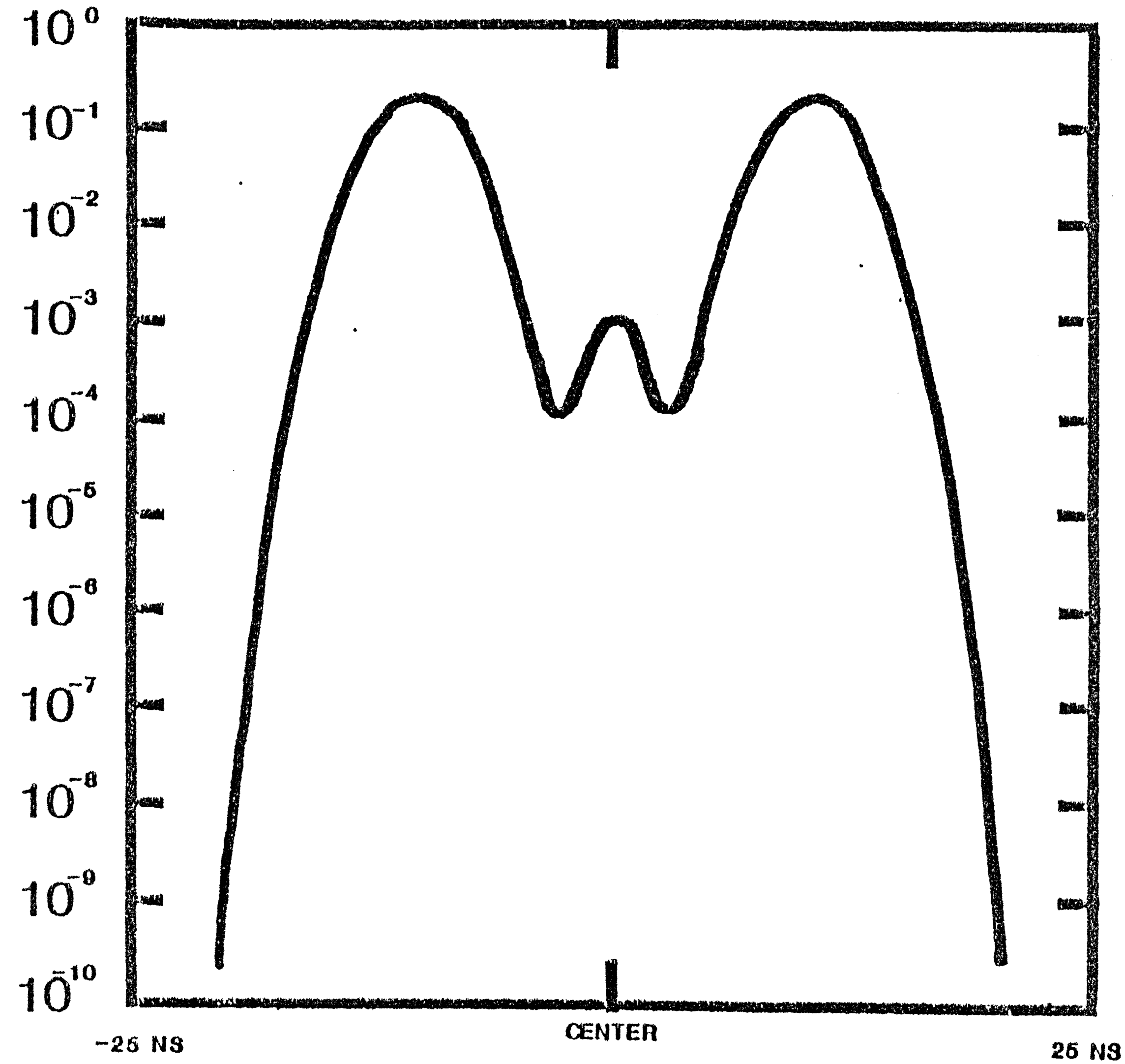
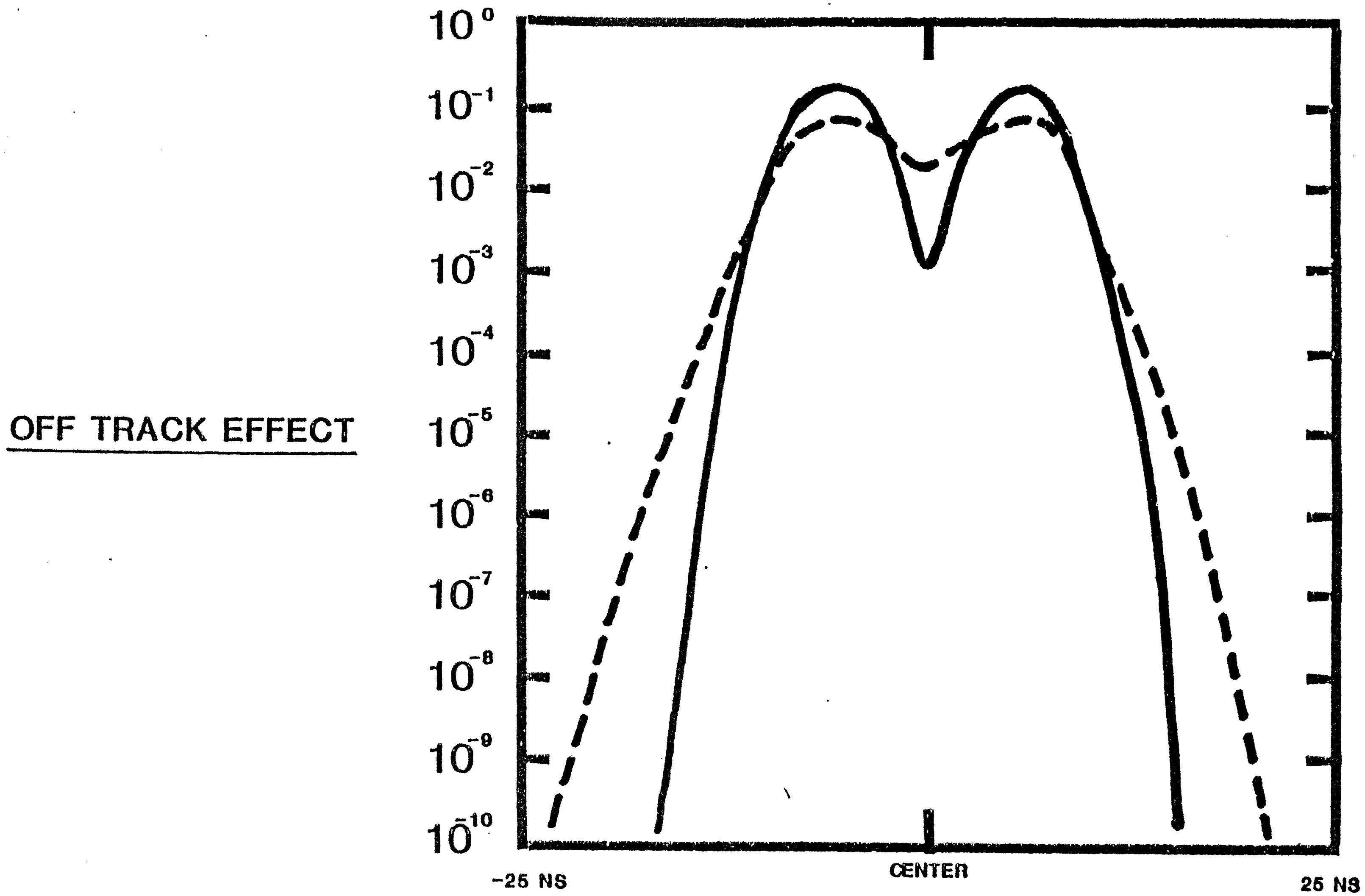


Fig. 16



OFF TRACK EFFECT
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
-25 NS
CENTER
25 NS

Fig. 17

of a disk drive, while one of the more elaborate types described here can be one of the most useful tools available to both the drive designer and the drive evaluator.

References

The references that follow discuss not only margin analysis, but also read/write technology. Ref. 1, although primarily comparing all the recording codes known to man, is also a useful reference on read/write design in general. Its partner, Ref. 2, as well as continuing the theme of Ref. 1, also describes in detail the application of pulse-slimming to data recovery. Ref. 3 describes what you would expect your transition distribution plots to look like, and why. Ref. 4 describes some implementation techniques for the passive type of margin analyzer (the transition distribution analyzer). Ref. 5 gives some useful quantitative data on interpreting the transition distribution plots, and relating them to the signal-to-noise ratio that exists in the system.

The last two references describe actual products that you can buy. Ref. 6 again describes how useful the margin analyzer can be, and is meant as an introduction to the margin analyzer that you can actually buy from MTI. Ref. 7, from Kode Inc., again is based on the product that they manufacture.

1. *The Choice of a Recording Code*, N D Mackintosh, *The Radio and Electronic Engineer*, Vol. 50, No. 4, pp 177-193, April 1980.

2. *A Superposition-based Analysis of Pulse-slimming Techniques for Digital Recording*, N D Mackintosh, *The Radio and Electronic Engineer*, Vol. 50, No. 6, pp 307-314, June 1980.

3. *Effect of Bitshift Distribution on Error-rate in Magnetic Recording*, E R Katz and T G Campbell, *IEEE Transactions on Magnetics*, Vol. MAG-15, No. 3, pp 1050-1053, May 1979.

4. *A Margin Analyser for Disk and Tape Drives*, N D Mackintosh, *IEEE Transactions on Magnetics*, Vol. Mag-17, No. 6, pp 3349-3351, November 1981.

5. *On Noise in Digital Recording*, G F Hughes and R K Schmidt, *IEEE Transactions on Magnetics*, Vol. MAG-12, No. 6, pp 752-754, November 1976.

6. *Phase Margin Analysis*, M R Monett, Memory Technology Inc., 2589 Scott Boulevard, Santa Clara, California 95050. Telephone: (408) 727-2333.

7. *A Transition Density Analyzer*, R D Petit, Kode Inc., 2752 Walnut Street, Tustin, California 92680. Telephone: (714) 730-6901.