

THIS SHEET HAS BEEN  
 REPLACED



This invention relates to apparatus and methods for improving data detection reliability in a coded signal processing channel by determining and setting optimal values for threshold constants constituting decision boundaries for interpreting sample values representative of coded binary data.

The following commonly-assigned prior art references are considered to be the most pertinent to the present invention:

1. U.S. Patent 4,945,538, granted July 31, 1990, entitled "Method and Apparatus for Processing Sample Values in a Coded Signal Processing Channel."
2. Allowed copending application U.S. Serial No. 07/920,027, filed July 27, 1992, entitled "High Data Rate Decoding Method for Coded Signal Processing Channel."
3. Allowed copending application U.S. Serial No. 07/526,878, filed May 22, 1990, entitled "Eight-Sample Look-Ahead for Coded Signal Processing Channels."
4. A. Patel, "A New Digital Signal Processing Channel For Data Storage Products," IEEE Trans. on Magnetics, Vol. 27, pp. 4579-4584, published November, 1991.

Each of the References 1-3 discloses a (1,7) ML decoder which uses a differing number of look-ahead digital sample values derived from an analog read signal to execute several decisions for implementing a maximum likelihood (ML) decision. An ML decision involves comparing each of several linear functions  $H$  of the sample values  $y$  to the value of a threshold constant constituting a ML



decision boundary related to that linear function. The linear functions and decision boundaries are chosen to minimize the overall mean square error in a ML sequence detection process using five, six, or eight look-ahead sample values of the read signal, as taught in References 1, 2, and 3, respectively.

Each Reference 1-3 discloses equations by which the decoder can compute nominal theoretical values of a set of boundary threshold constants from information about the shape of the analog pulse that is read back from the analog pulse transmitted over the (1,7) ML channel.

These references describe how to calculate and set at nominal theoretical values the test boundary threshold constants of look-ahead algorithms in order to accommodate variations in pulse shape when the pulse shapes are described in a simple form by six variables; i.e., when the read response to the positive and negative magnetic transitions are the pulses with the sample values  $\dots, 0, \alpha_1, \beta_1, \gamma_1, 0, \dots$  and  $\dots, 0, \alpha_2, \beta_2, \gamma_2, 0, \dots$  respectively. They also state that the thresholds may be optimized empirically in response to the boundary crossings in actual test data, but do not disclose how.

Moreover, the boundary threshold constants, as computed by these equations, are generated from the readback pulse corresponding to a single transition. This assumes that the positive or negative readback pulse has a single form which can be described by three non-zero sample values. However, these computed thresholds may not be optimal because in reality signal processing channels are subject to nonlinearity, misequalization, and pulse asymmetry, in addition to pattern-dependent noise.

Reference 4 describes a five-sample look-ahead algorithm, including a geometrical interpretation of the decision algorithm. Boundary constants are determined from formulae related to the pulse shape response for a single magnetic transition, as in References 1-3.

None of these references discloses or suggests (i) determining



nominal values of the threshold constants from a statistically significant number of readback pulse transitions that account for nonlinearity, misequalization, pulse asymmetry and pattern-dependent noise, and (ii) treating as potential errors in the detected data those which are too close to a selected threshold boundary, and automatically adjusting each threshold boundary so that a substantially equal number of error events occur at each side thereof.

There is a need for apparatus and methods for (i) determining for each particular magnetic recording device optimal value settings for the decision-boundary-constituting threshold constants for a (1,7) ML channel despite signal and noise anomalies in the readback pulse, (ii) establishing the decision boundary from a series of magnetic recording transitions instead of by artificially superpositioning the pulse shape corresponding to a single magnetic recording transition, and (iii) setting these boundary threshold constants at their optimal values by use of a predetermined sequence of recorded data or by use of any data with or without comparison to known data subject to signal and noise anomalies.

The present invention has the object to overcome the above drawbacks.

A method is provided for improving data detection reliability in a coded signal processing channel in which digital electrical signals representing digital sample values are generated from analog signals read by a transducer from digital data recorded on a data storage medium, as described in claim 1.

A plurality of threshold constants whose values define decision boundaries are used in determining whether detected data corresponding to a coded sequence of a runlength limited (RLL) code is a "1" or a "0". If desired, boundary adjustment may be performed dynamically during a time-out between consecutive read operations or during a read operation.

To determine an initial value for the threshold constant for a preselected decision boundary, two known preselected data



patterns of the digital sample values are identified that are determined to be closest above and closest below, respectively, said boundary. Actual values of linear functions of the digital sample values corresponding to multiple occurrences of these two data patterns are preferably collected on a decision boundary plot to permit visualization of the decision process. Clusters are formed of actual values of the one data pattern above and the other data pattern below the decision boundary. A focal point is determined for each of the two clusters and the threshold constant for the preselected decision boundary is established substantially midway between such focal points, provided the actual values in both of said clusters exceeds a preselected sufficient number, and then loaded in a boundary register.

After the initial value of the threshold constant for the selected boundary has been determined, it is adjusted to an optimal value by either of two methods. In each method, separate counters count as potential error events in the detected data those which are determined to be both above and below a selected one of the decision boundaries as a result of a "1" being potentially misdetected as a "0" (corresponding to an N pattern being potentially misinterpreted as an M pattern), and vice versa. The selected decision boundary is adjusted in a direction toward providing a substantially equal number of potential error events at each side of that boundary. A multiplexor enables selection of the particular boundary to be evaluated and adjusted.

In one method, an error margin distance (MD) is computed which represents the difference between a boundary threshold constant and H outputs representing actual values of the aforesaid linear functions. When this distance MD is small (within  $\pm m$  units), it is considered a potential error event. This distance MD, when compared to zero, indicates whether the value of H is above or below the decision boundary.

In the other method, while a sequence of digital sample values of the analog signals corresponding to known recorded data is generated, a selected one of the decision boundaries is



intentionally moved upward  $m$  units and then downward  $m$  units from an initial preselected position to determine the number of potential error events on each side and to determine if and how the boundary should be adjusted.

Fig. 1 is a schematic diagram of a coded digital signal processing channel, as known in the prior art.

Fig. 2 depicts ideal write and read waveforms for (1,7) ML coded data and read sample values, as heretofore known.

Fig. 3 is a boundary plot diagram for one type of peak position check for state 000, as known in the prior art.

Fig. 4 is a boundary plot diagram for a modified peak position check for state 000, as known in the prior art.

Fig. 5 is a boundary plot diagram for a baseline check for state 000, as known in the prior art.

Figs. 6A and 6B, when taken together, constitute a schematic diagram of a decoder apparatus embodying the invention and comprising a six-sample look-ahead feature and means for setting decision boundaries at optimal values in the coded digital signal processing channel of Fig. 1.

Figs. 7A and 7B, when taken together, constitute a schematic diagram that depicts a modification of the decoder apparatus illustrated in Figs. 6A and 6B to show a variation of the manner in which the invention may be implemented.

Fig. 1 depicts, as known in the prior art, a coded digital signal processing channel, preferably an extended partial response maximum likelihood (EPRML) channel which is characterized by the polynomial  $(1-D)(1+D)^2$ . An analog signal is read by a transducer 8 from a magnetic recording medium, such as a disk 9, in a digital storage device. This read signal corresponds to a binary data sequence coded with the (1,7) run length limited (RLL) code during a write operation. The read signal passes from line 10 through a



preamplifier 11 with automatic gain control (AGC) and a low bandpass filter 12. The output of filter 12 is supplied to a phase-locked clock 13 comprising a phase-locked loop timing recovery circuit including a variable frequency oscillator (VFO).

Filter 12 and equalizer 22 filter the read signal so that the response to a single magnetic transition is a pulse given by the sample values ..., 0, 2, 4, 2, 0, ... . This results in written transitions of magnitude 2 between +1 and -1 levels of magnetic saturation, and read response is normalized by a scaling factor to correspond thereto.

The coded analog read signal output from filter 12 is supplied to a delay means 20. Delay means 20 provides delay alignment between the analog signal and clock signal inputs to an analog-to-digital converter (ADC) 21. ADC 21 converts the analog input signal into digital sample values, such as  $y_0 \dots y_5$ , at successive clock times, where  $y_0$  corresponds to the current clock cycle. These digital sample values, optionally equalized by an equalizer 22, are then passed to a decoder 23. Positive and negative boundary threshold constants are applied via lines 24, 25, respectively, to decoder 23. Decoder 23 applies a decoding algorithm to the digital sample values to provide a coded binary data output in line 26 and, where appropriate, a pointer flag in line 27, at successive clock times under control of clock 13.

Fig. 2 depicts the write and read waveforms and associated sequences of data and sample values clocked by clock 13 in a noise-free environment. As so depicted, the sample values range through the discrete set  $\{-4, -2, 0, +2, +4\}$ . Actual samples, however, as noted earlier, will not be exactly equal to the ideal patterns of the read signal because of signal anomalies due to nonlinearity, misequalization, and/or pulse asymmetry, as well as additive noise, and thus will range through various noninteger values around the integers in the discrete set. In general,  $y = x + e$  where  $x$  is exactly equal to one of the ideal patterns, and  $e$  is the error in the received signal. The (1,7) ML decoding algorithm decides which of the ideal patterns is closest to the



actual received pattern.

At any point in time, the (1,7) ML channel will be in one of six states: 000, 100, 111, 011, 001, or 110. States 100, 000, and 001 correspond to detection of a positive read pulse and have mirror image (i.e., +/- sign) symmetry with states 011, 111, and 110, respectively, that correspond to detection of a negative read pulse. The current state is determined by past decoding decisions, and the current decoding decision determines which state comes next. Two of these states (001 and 110) are transitional only as the decision is automatic because the (1,7) code constraint implies that state 001 must be followed by state 011 and similarly state 110 must be followed by state 100. At each of the remaining four states, based on the sampled read signal, the channel must make a ML decision regarding the next state.

As in Reference 2, each decision of the (1,7) ML algorithm requires six linear functions,  $H_1, H_2, H_3, H_4, H_5, H_6$ , of the sample values  $Y_0, Y_1, Y_2, Y_3, Y_4, Y_5$ , to be computed and compared to six thresholds A, B, X, Y, Z and W as established by the boundary threshold constants. The A and B boundaries are for a baseline check, and the X, Y, Z and W boundaries are for a peak position check, as shown in Table 1.

Table 1, and all other tables, are included in an Appendix hereto which is made part hereof. Note that  $H_3$  is compared to two different boundary thresholds X and Y. There are four sets (corresponding to four decision-making states), each consisting of six boundary thresholds, resulting in a total of 24 boundary threshold constants. The set used for each ML decision depends on the current state. An error is associated with the value of at least one of the functions  $H_i$  falling on the wrong side of its boundary threshold.

It will thus be apparent that the probability of decoding errors is controlled by the established values of the boundary threshold constants.

Assume now, for sake of illustration, that the current state is 000. Decoding will be based on the next six values of the



sampled read signal:  $y^-(y_0, y_1, y_2, y_3, y_4, y_5)$ . The 24 possible six-sample patterns of possible sequences for the ideal read signal corresponding to 24 possible binary data patterns for state 000 are set forth in Table 2.

The decision process can be represented geometrically by the graphs shown in Figs. 3, 4, and 5. Each graph contains a line defining a boundary line between N patterns above the line and M patterns below the line. The points, as plotted in Figs. 3, 4 and 5, correspond to the ideal patterns. The location of each boundary line segment is determined by a corresponding boundary threshold constant. For example, the Y boundary, as shown in Figs. 3 and 4, is a segment of the line  $H_3(y^-) = KY$ , where KY is the threshold value. The functions  $H_i(y^-)$  are computed and plotted on the appropriate graphs. If a point falls on the wrong side of the boundary, an error occurs. Thus, if the point corresponding to  $y^-$  is closest to one of the N patterns, then  $y_0$  is decoded as a 2, and the next state is 001. If it is closest to one of the M patterns, then  $y_0$  is decoded as a 0, and the next state is 000.

As illustrated in Figs. 6A and 6B, the apparatus embodying the invention comprises a decoder 300 which replaces the prior art decoder 23 depicted in Fig. 1 and constitutes an improvement over the decoders disclosed in Reference 2. The following description has been limited to that considered necessary for an understanding of the embodiments of the present invention. For additional background description, the reader is referred to Reference 2.

In Figs. 6A, 6B and 7A, 7B, ck will be used to designate the clock input from a common source that is required in order to provide the necessary timing for the various registers hereinafter described.

According to the invention, and in the manner hereinafter described, each of the 24 boundary threshold constants of the (1,7) ML decoder 300 is determined and separately and independently optimized to minimize the possibility of decoding errors. Two types of decoding errors can occur: (1) misdecoding an N pattern as an M pattern, which causes a "1" to be misdetected as a "0" in



THIS SHEET HAS BEEN  
REPLACED

the detected (1,7) sequence; and (2) misdecoding an M pattern as an N pattern, which causes a "0" to be misdetected as a "1" in the detected (1,7) sequence. Adjusting the boundary threshold decreases the likelihood of one type of error but increases the likelihood of the other type of error.

Table 3 shows the six different types of boundary thresholds (A, B, X, Y, Z, W) that are to be optimized for the four different states (000, 100, 111, 011) for a total of 24 boundary types.

In the term  $KW2^+$ ,  $W2$  denotes a boundary that corresponds to the state 000 or 111; the superscript '+' clarifies that the data pattern is generated by a positive read phase and therefore  $W2^+$  identifies the state as 000 rather than 111; and prefix K in  $KW2^+$  denotes that the boundary has a preselected, but adjustable, constant value. Since each boundary threshold is optimized separately and independently, the methods and apparatus for optimizing the threshold value  $KW2^+$  of only one boundary, the  $W2^+$  boundary, will be described with the understanding that all other boundaries are optimized in a similar manner. Hereafter, the "+" superscript will be dropped for simplification; and this boundary will hereinafter be referred to simply as  $W2$  and its threshold value as  $KW2$ .

## I

### Use of Focal Points of Data Pattern Clouds for Initial Boundary Determination

This Section I describes how an initial value for a decision-boundary-constituting threshold constant for a (1,7) ML channel may be determined without requiring pulse sampling and calculation of a nominal value based upon the pulse shape of the read response to a single magnetic recording transition. It requires use of a known sequence of detected data.

In Figs. 4 and 5, for simplification, a given data pattern (eg., N2) has been shown as a single point. However, in actuality, due to noise, misequalization and other factors, a given data

This Sheet Has Been Replaced



pattern will fall at differing distances from its ideal (noise-free) position. Thus the cluster of points generated by the actual values for a given data pattern will hereinafter be referred to as a "cloud." Each cluster of points or cloud is resolved into a respective single point hereinafter referred to as the "focal point." This focal point may, for example, be determined as the mean or average of the actual values of points in each cluster, or a selected subset of the actual values of points in each cluster, to implement this method in the manner now to be described.

According to the invention, the value of the threshold constant for a selected decision boundary for each associated pair of patterns (eg., N2 and M8a) is determined from the focal points (such as the mean values) of the corresponding clouds using software. More specifically, a (1,7) sequence  $D^* = D_1, D_2, \dots, D_n$  (such as of 512 bits) corresponding to some pseudo-random data is written to a data storage disk; e.g., data shown below in Section II hereof in hexadecimal notation. The detected sequence  $d^* = d_1, d_2, \dots, d_n$  is obtained, as explained in connection with Figs. 1 and 2, from digital sample values  $y^* = (y_1, y_2, \dots, y_n)$  generated from the analog read signal. The pseudo-random sequence should be preceded and followed by a one byte pad of zeroes, which are not decoded.

Now, using  $d^*$  and  $y^*$ , the state vector  $s^* = (s_1, s_2, \dots, s_2)$  is computed, where  $s_i$  is the state  $a_i b_i c_i$  of the channel at the time that the  $i^{th}$  sample is being decoded, assuming that prior decoding decisions were correct.

Next, the average value of each function  $H_1 \dots H_5$  in the left column of Table 1 is computed for each pattern or set of patterns in the columns on the right. For example,  $\langle H_1(N1_s, N2_s) \rangle$  is equal to the average value of  $H_1(y_j)$  - where  $y_j = (y_j, y_{j+1}, y_{j+2}, y_{j+3}, y_{j+4}, y_{j+5})$ , and the average is over all  $y_j$  for which  $s_j = s$  (one of the four states 000, 011, 100, and 111) and the ideal pattern is  $N1_s$  or  $N2_s$ . The occurrences of these values of  $H_1(y_j)$  are counted by suitable means (not shown) and, if desired, displayed on a screen to show the sample size for each cloud or set of clouds, to alert the user if there is an insufficient number of



samples to determine any threshold.

Then, the thresholds are computed as the averages of the appropriate clouds, as follows:

$$\begin{aligned} KA_s &= [H_1(N1_s, N1_s) + H_1(M1_s, M2_s, M3_s)] / 2. \\ KB_s &= [H_2(N4_s, N5_s) + H_2(M1_s, M2_s)] / 2. \\ KX_s &= [H_3(N1_s, N2_s) + H_3(M6_s, M7_s)] / 2. \\ KY_s &= [H_3(N3_s) + H_3(M8_s)] / 2. \\ KZ_s &= [H_4(N1_s) + H_4(M8_s)] / 2. \\ KW_s &= [H_5(N2_s) + H_4(M8A_s)] / 2. \end{aligned}$$

Table 4 sets forth an APL program listing to perform the calculations described above. D17 is the binary vector of data which was written to the disk, and Y is the corresponding analog readback signal. Y and D17 have above been referred to as  $y^-$  and  $d^-$ , respectively. This program will compute all 24 thresholds and store the results in two APL vectors KPOS and KNEG, defined as follows:

KPOS = (A<sub>000</sub>, B<sub>000</sub>, X<sub>000</sub>, Y<sub>000</sub>, Z<sub>000</sub>, W<sub>000</sub>, A<sub>100</sub>, B<sub>100</sub>, X<sub>100</sub>, Y<sub>100</sub>, Z<sub>100</sub>, W<sub>100</sub>)  
KNEG = (A<sub>111</sub>, B<sub>111</sub>, X<sub>111</sub>, Y<sub>111</sub>, Z<sub>111</sub>, W<sub>111</sub>, A<sub>011</sub>, B<sub>011</sub>, X<sub>011</sub>, Y<sub>011</sub>, Z<sub>011</sub>, W<sub>011</sub>)

The program contains one loop, in lines 12-46, which is executed once for each of the four states, 000, 100, 111, 011, with the variable T used to indicate the state for which thresholds are being computed.

Lines 11, 14 and 22-25 compute the variable E, which contains the locations, within D17, where the channel is in state T. Lines 16-20 compute the H<sub>i</sub> functions, as defined in Table 7, for every six-tuple (Y<sub>j</sub>, Y<sub>j+1</sub>, Y<sub>j+2</sub>, Y<sub>j+3</sub>, Y<sub>j+4</sub>, Y<sub>j+5</sub>), within the vector Y.

Lines 27-36 compute the locations, within D17, where the channel is in state T, and the next six data points form one of the patterns indicated in the comment on the right side of the line. In addition, the sample size for each pattern or set of patterns is collected in the variable S.

Lines 40-45 compute the thresholds for state T, using the equations given in Section 1 of Table 5. Each threshold constant preferably is initially determined by use of the focal point method



just described; or, if desired, it may be calculated and set at a nominal theoretical value as taught in the prior art. The value or setting of the threshold constant is then optimized.

Two methods for adjusting the boundary thresholds and the decoder apparatus for implementing these methods will now be described. Each is based upon the unique relationship between the boundary settings and potential error events in the (1,7) ML channel. In each case, in accordance with the invention, the boundary threshold is set at a point where potential error events in each direction, upward and downward, are substantially balanced.

## II

### A. Use of Error Margin Measurement for Boundary Adjustment

Referring now to Fig. 6A, inputs  $H_1-H_6$  represent precomputed linear functions of digital sample values  $y_0-y_5$  corresponding to multiple occurrences of one preselected N data pattern and another preselected M data pattern. They preferably are generated in precisely the same manner as shown and described in connection with Fig. 4 of Reference 2. Hence, the manner in which they are generated will not here be shown or described. Note that input  $H_6$  for computing the boundary P for the phase check is not relevant to the present invention.

As in Reference 2, six sign change blocks 302-307 change the sign of the incoming binary number  $H_1-H_6$  whenever the  $c_0$  bit in a state register 310 has the value "1", which corresponds to states 001 and 111 in the negative phase of the sampled read signal. The outputs  $H_1'-H_6'$  from the respective sign change blocks 302-307 and also the initial boundary threshold constants  $KA_1...KW_2$  (previously loaded via bus IB into boundary-threshold-value register 308) are fed in parallel to a set of subtractors. Subtractors 312-324 compute differences between an  $H'$  output and a related boundary threshold prefixed K to provide an output prefixed D representing an error margin distance MD. This distance MD is fed to a related one of the comparators 330-342 which, by comparing the MD to zero,



determines whether the value of  $H$  is above or below the threshold boundary.

For example, the output from subtractor 312, which is  $DA_1 = H_1 - KA_1$ , is fed to comparator 330; and comparator 330 compares this to zero to generate the Boolean value  $A_1 = (DA_1 \leq 0)$ , which denotes whether the function  $H_1$  is above or below the decision boundary. This result is the same as that achieved in the decoder shown in Fig. 4 of Reference 2; and hence the intermediate logic and output logic, as set forth herein in Table 5, is identical with that shown in Table 4 of Reference 2. However, the present method has the advantage that the output from the subtractor 312 can be used as a measure of available margin to a potential error event, called margin distance MD.

The outputs from the comparators 330-342 are values "1" or "0", which, together with the state values  $a_0, b_0, c_0$  from state register 310, are fed to the logic block 349 for implementing the logic shown in Table 5.

The logic of block 349 provides the detected data  $d_0$  corresponding to state  $a_0 b_0 c_0$  and the erasure pointer  $P$  in accordance with the logic of Section 1 of Table 5. Block 349 also creates the state values  $a_1 b_1 c_1$  for the next state in accordance with the logic of Section 3 of Table 5 including the intermediate logic of Section 2 thereof. This state value is supplied to state register 310 to become the current state value  $a_0 b_0 c_0$  for the next clock cycle.

The error margin distances MDs are available for all boundaries as outputs of the subtractors 312-324. Multiplexor (MUX) 345 responds to a boundary select signal in line 346 from a test controller 347 to select and output to line 348 one of the error margin distances  $DA_1 \dots DW_2$ , to be examined for a boundary threshold of interest.

Assuming the error margin distance MD to be selected is  $DW_2$  (the difference output of subtractor 324), MUX 345 will output  $DW_2$  for the boundary  $W_2$  to line 348. The outputs  $b_1, d_1, DW_2$  of Fig. 6A are used to implement the boundary adjustment process. Note



that Fig 6B depicts the state of the system five clock cycles later, relative to Fig. 6A, and so these outputs are labeled  $b_6$ ,  $d_6$  and  $DW2_6$ , respectively, in Fig. 6B. The  $d$  and  $b$  outputs, which are binary, are fed into a seven-bit shift register 350 and a five-bit shift register 352, respectively. The  $DW2$  values, which are numerical values, such as in bytes, are fed into a five-byte shift register 354.

The outputs from comparators 356 and 358, in combination with the logic of Tables 6 and 7, generate, at each clock cycle, four binary values,  $D_{above}$ ,  $D_{below}$ ,  $W_{above}$  and  $W_{below}$  which are stored in single-bit registers similarly designated. These binary values are used to increment a set of five counters, denoted  $T_A$ ,  $T_B$ ,  $M_A$ ,  $M_B$  and  $C$ , respectively. Initially each of these counters is set to zero by a signal in reset line  $R$ . Then they are updated at each cycle.

Comparators 356 and 358 compare the error margin distance value  $DW2_1$  to a predetermined constant  $m$  to determine whether it is within  $m$  units above or below the boundary, respectively. The binary values  $D_{above}$  and  $D_{below}$  are computed as

$$D_{above} = \begin{cases} 1 & \text{if } DW2_1 \leq m \\ 0 & \text{otherwise} \end{cases}$$

$$D_{below} = \begin{cases} 1 & \text{if } -DW2_1 < m \\ 0 & \text{otherwise} \end{cases}$$

It is also necessary to process the values of  $d_1$  and  $b_1$  which are combined in block 351 in one logical expression to indicate if the system was in the desired state (here assumed as 000) while  $d_1$  was being detected. At the same time, the values of  $d_0 \dots d_5$  are concurrently combined in block 351 into two different logical expressions which respectively indicate each occurrence of the one  $N$  data pattern ( $N2$ ) nearest above and the one  $M$  data pattern ( $M8a$ ) nearest below the  $W2$  boundary irrespective of actual distance from said boundary.



THIS SHEET HAS BEEN  
REPLACED

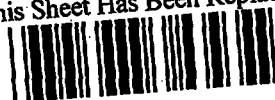
The logical expressions to be used for each state and for the pattern boundaries are given in Tables 6 and 7, respectively, wherein  $\emptyset$  indicates a "don't care". Note that the nearest patterns at each boundary are unique, except in the case of the B boundary (see Fig. 5) which has more than one equally near patterns N4, N5, and N3 above the boundary, corresponding to 0, 1, 0, 1, 0,  $\emptyset$  and 0, 1, 0, 0, 1, 0, respectively.

$W_{\text{above}}$  will be a logical "1" whenever the selected data pattern (in this case, N<sub>2</sub>) occurs at a given state (in this case, assumed 000) irrespective of how far above the W2 boundary. Similarly,  $W_{\text{below}}$  will be a logical "1" whenever the selected data pattern (in this case, M8a) occurs at a given state (herein 000) irrespective of how far below the W2 boundary.

If  $W_{\text{above}} = 1$ ,  $T_A$  counter will be incremented one unit. The  $T_A$  counter counts up the total number of occurrences of that data pattern (herein N<sub>2</sub>) which is nearest above (irrespective of how far above) the W2 boundary, until a preselected number N of such patterns has occurred. If  $W_{\text{above}} = 1$  and also  $D_{\text{above}} = 1$ , while the count in  $T_A$  counter is less than N, the AND gate 360 will be enabled and cause the  $M_A$  counter to be incremented. This counter accumulates the number of times that the corresponding computed decision function (in this case,  $H_5$ ) is above and very close to (i.e., within m units of) the W2 boundary until the count in the  $T_A$  counter reaches N and thereby disables the AND gate 360.

Then the same method will be followed for that data pattern (in this case, M8a) which is nearest below (irrespective of how far below) the decision boundary (in this case, W2). The  $T_B$  and  $M_B$  counters will be incremented in a similar manner. If  $W_{\text{below}} = 1$ , then the  $T_B$  counter will be incremented one unit, up to the maximum value of N. This counter will count up the total number of occurrences of the data pattern which is nearest below the W2 boundary, until N actual values such patterns have occurred. If  $W_{\text{below}} = 1$  and also  $D_{\text{below}} = 1$ , while the count in  $T_B$  counter is less than N, the AND gate 362 will be enabled and cause the  $M_B$  counter to be incremented. This counter accumulates the number of times

This Sheet Has Been Replaced



that the actual values of the corresponding computed decision function ( $H_5$ ) is below and very close to (i.e., within  $m$  units of) the W2 boundary.

Meanwhile, the C counter will be decremented if  $W_{\text{above}} = 1$  and  $D_{\text{above}} = 1$  while  $T_A < N$ , and incremented if  $W_{\text{below}} = 1$  and  $D_{\text{below}} = 1$  while  $T_B < N$ . Therefore it will accumulate the difference between the number of points below and above the W2 boundary, counting only those points within  $m$  units of the W2 boundary. If the current value KW2 of the threshold of the W2 boundary is lower than the optimal value, then the C counter will tend to increment, while if KW2 is above the optimal value, the C counter will tend to decrement.

If and when the count in both the  $T_A$  and the  $T_B$  counter reaches  $N$ , their outputs will drop causing inverters 364 and 366 to condition the circuitry 376 to examine the value in the C counter. The value in the C counter will determine whether to adjust the boundary threshold. It will be adjusted upward if the count is at least equal to a preselected number  $S$ , or downward if it is less than or equal to  $-S$ , or adjustment will stop if the count is between  $+S$  and  $-S$ , using the adjustment decision rule shown in Table 8. MUX 370 (Fig. 6A), meantime, will have been conditioned by the signal in line 346 to feed the current threshold value KW2 to adder 372. The other input to adder 372 from line 374 will be a "1", "0", or "-1" as determined by the boundary adjustment decision logic circuitry 376 (Fig. 6B) which implements the decisions specified in Table 8. The value of boundary threshold constant KW2 in the appropriate boundary threshold value register 308 will be updated by the output of adder 372.

Meanwhile, if the count in either counter  $M_A$  or  $M_B$  exceeds a preselected number  $Q < N$ , then circuit 361 or 363, respectively, will provide an output U. This output indicates the device being tested is unacceptable because too many (more than  $Q$ ) actual values fall within  $m$  units above or below the W2 boundary--a condition which can occur even if the difference count in counter C is acceptable and the counts in either  $T_A$  or  $T_B$  have not reached  $N$ .



If the boundary threshold value is changed, the  $T_A$ ,  $T_B$ ,  $M_A$ ,  $M_B$  and C counters will be reset to zero by a signal on line R, and the process will be repeated to determine whether further changes should be made to the threshold value KW2 of the W2 boundary. At some point, when the boundary threshold value has reached an optimal state, the C counter will tend to neither increase nor decrease very much, and the  $T_A$  and  $T_B$  counters will both reach N before the C counter reaches a preselected number S or -S, where  $S < N$ . At this point, the boundary will be deemed optimal, and the adjustment process will stop.

This approach seeks to balance the respective error margins for data patterns above and below each boundary. For this process to work, the data sequence being read must contain the relevant data patterns in sufficient frequency that the  $T_A$  and  $T_B$  counters will both reach N in a reasonable time. For this example, at state 000 indicated by  $d_1b_1=00$ , the patterns N2 and M8 given by  $d_0d_1d_2d_3d_4d_5=010001$  and  $d_0d_1d_2d_3d_4d_5=001010$  must both occur frequently in the data pattern.

Instead of using the inputs  $d_1 \dots d_6$ , representing the detected data, for identifying the states and the patterns as set forth in Tables 6 and 7, the inputs  $d_1 \dots d_6$  may be replaced by a stream of known data  $D_1 \dots D_6$  that corresponds to this detected data. While this has the advantage that the inputs will be known to be correct, it requires that a known pattern be used in the boundary-adjustment process. It requires additional hardware, such as a pseudo-random pattern generator 380, to be switched by switch 382, to provide this data stream during the read operation for boundary adjustment. The following 32-byte pattern (in hexadecimal notation) is an example of a short pattern which, if written repeatedly along a data track, provides sufficient frequency of patterns for adjusting each of the 24 boundary values.

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| 4515 | 1455 | 528A | 2492 | 0841 | 08A2 | 8411 | 2494 |
| 40AA | A821 | 2882 | 0820 | 4545 | 4A24 | 8422 | 8449 |

The boundary adjustment process is initiated by an enable or



power-on signal in line E. This signal can be generated at a selected time during final test of a disk drive. Alternatively, it may be generated dynamically while the disk drive is in use by a customer during a read operation or during a time out between consecutive read operations.

When boundary adjustment has been completed, the contents of the counters can be used to determine the error margin of the device under test. The quantities  $M_A/N$  and  $M_B/N$  give estimates of the probability of points falling within  $m$  units of the boundary, on either side. For a good device,  $M_A/N$  and  $M_B/N$  will both be small. If either  $M_A/N$  or  $M_B/N$  is larger than some predetermined probability  $P$ , where  $P=Q/N$ , then the error margin for that device will be considered too small, and the device should be rejected or repaired. This technique can be used at the time of manufacture or by a customer during actual operation.

#### B. Use of Error Mode Analysis for Boundary Adjustment

Figs. 7A and 7B, taken together, illustrate a decoder 300&prime. which is a variation of the decoding apparatus and method depicted in Figs. 6A and 6B for implementing the invention to optimize the W2 boundary. Identical blocks in the two sets of figures have been designated with the same reference numerals. As in Section IIA hereof, the initial values of the threshold-boundary constants are loaded into the appropriate ones of registers 308 via bus IB.

This approach differs from that described in Section IIA in that instead of requiring direct calculation of the error margin  $DW2=H_5-KW2$ , the boundary threshold is moved upward  $m$  units and then downward  $m$  units from  $KW2$ . While moved upward, counter  $M_A$  counts the number of times the value of the linear function  $H_5$  is at or below boundary  $KW2+m$ ; and later while moved downward, counter  $M_B$  counts the number of times the value of  $H_5$  is above the boundary  $KW2-m$ . Since error margin calculation is not required in this



method, subtractors 312-324 and comparators 330-342 may be combined into a single set of comparators, such as taught in Fig. 4 of Reference 2.

This method requires that the correct value of the detected data sequence  $\{d_i\}$  be known in advance. This known data sequence will be designated  $\{D_i\}$  to distinguish it from  $\{d_i\}$ . When  $\{D_i\}$  is known, it is possible to construct the phase polarity sequence  $\{B_i\}$ . The sequence  $\{B_i\}$  represents the signal phase at the  $i$ th clock cycle according to the convention:

$$B_i = \begin{cases} 0 & \text{if phase is positive} \\ 1 & \text{if phase is negative} \end{cases}$$

and is computed recursively in logic circuitry of 392 and stored in shift register 393 by:

$$B_i + 1 = [y_i \geq 0] D_i + B_i \overline{D_i} .$$

The comparators 356, 358 of Fig. 6B are replaced in Fig. 7B with AND gates 395 and 396, which have as inputs indicators of the current boundary offset,  $+m$  or  $-m$ , together with logic expressions for the detected data above and below the preselected W2 decision boundary. An offset select circuit 398 will be conditioned by select line 399 to provide a signal to the AND gate 395 or 396 depending upon whether the offset  $+m$  or  $-m$ , respectively, has been selected.

First, the W2 boundary is selected by MUX 370. The value of its boundary threshold KW2 provides one of the two inputs to adder 372. The other input comes from MUX 389 and is selected via line 399. The W2 boundary is moved upward by  $m$  units from its initial value KW2 by adding at 372 the value  $+m$  from MUX 389 and loading the new value of KW2 in the appropriate register 308 via bus IB.

The detected data  $d_6$  from the decoder is fed into the shift register 350; and the corresponding known data  $D_6$ , provided either



by pattern generator 380 (as in Fig. 6A) or by a source external to the hardware, is fed into shift register 391. Block 392 evaluates the next signal phase using the current signal phase  $B_5$  from shift register 393, the known data  $D_5$ , and the sign bit of the sample value  $Y_5$ .

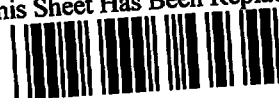
Circuit 397 executes the logic of Tables 9 and 10 for detecting an occurrence of the known N2 data pattern  $D_0D_1D_2D_3D_4D_5=010001$  at state 000, indicating that the known pattern is above the W2 boundary. This provides a logical "1" input to the register  $W_{above}$ . If, simultaneously, the detected data is the M8 pattern,  $d_0d_1d_2d_3d_4d_5=001010$ , then the logic of Table 10 detects this as a detected pattern below the W2 boundary and AND gate 395 provides a logical "1" input to the register  $D_{above}$ . The components shown below the  $W_{above}$ ,  $W_{below}$ ,  $D_{above}$  and  $D_{below}$  registers are identical to the ones in Fig. 6B and function in the same way.

As in Section IIA, counter  $T_A$  counts occurrences of the N2 data pattern  $D_0D_1D_2D_3D_4D_5=010001$  (i.e., where  $W_{above}=1$ ), counter  $M_A$  occurrences of potential errors indicated by the downward crossing of the moved W2 boundary when the detected pattern is the M8 pattern  $d_0d_1d_2d_3d_4d_5=001010$  (where  $D_{above}=1$ ), and counter C counts down by the number of these downward crossing errors. (The reader is referred to Section IIA for a detailed description of the counters and other identical blocks.) When counter  $T_A$  reaches N, the W2 boundary is reloaded in register 308 to its initial value  $KW_2$  via bus IB.

Next, the W2 boundary is moved downward by m units from its initial value  $KW_2$  by adding to its initial value at 372 the value -m selected through multiplexer 389, and loading this new value of  $KW_2$  into the appropriate one of the registers 308.

Detected data  $d_6$  from the decoder is fed into shift register 350 and the corresponding known data  $D_6$  is fed into shift register 391. Block 392 evaluates the next signal phase using the current signal phase  $B_5$  from shift register 393, the known data  $D_5$ , and the sign bit of the sample value  $Y_1$ .

Circuit 397 executes the logic of Tables 9 and 10 to detect



an occurrence of the known M8 data pattern  $D_0D_1D_2D_3D_4D_5=001010$  at state 000, indicating that the known pattern is below the W2 boundary. This provides a logical "1" input to the register  $W_{\text{below}}$ .

If, simultaneously, the detected data is the N2 pattern  $d_0d_1d_2d_3d_4d_5=010001$ , then the logic of Table 10 detects this as a pattern above the W2 boundary and AND gate 396 provides a logical "1" input to register  $D_{\text{below}}$ .

As described in connection with Section IIA, counter  $T_8$  counts occurrences of the M8 data pattern  $D_0D_1D_2D_3D_4D_5=001010$  (i.e., where  $W_{\text{below}}=1$ ), counter  $M_8$  counts occurrences of apparent errors indicated by downward crossing of the W2 boundary when the detected pattern is the N2 pattern  $d_0d_1d_2d_3d_4d_5=010001$  (i.e., where  $D_{\text{below}}=1$ ), and counter C counts up by the number of upward crossing errors. When counter  $T_8$  reaches N, the W2 boundary is reset to its initial value KW2.

The final result in counter C in conjunction with Boundary Adjustment Decision Logic of Table 8 generates one of three values: +1, -1, or 0. This value is added to the current KW2 boundary value at adder 372 using the MUX 389 and MUX 370.

The above process is repeated recursively until the stop condition as described in Table 8 occurs, denoting that optimization for the W2 boundary has been achieved.

The process, as described for the W2 boundary, is repeated to optimize the other boundaries in a similar manner. The boundary to be optimized is selected by the boundary select controller 347, which conditions MUX 370. Also a signal in line 346 causes the logic from Tables 9 and 10 to provide appropriate logic expressions corresponding to the selected boundary to the registers  $W_{\text{above}}$ ,  $W_{\text{below}}$ ,  $D_{\text{above}}$  and  $D_{\text{below}}$ .

For sake of simplification, in the embodiments as disclosed, it has been assumed that the objective is to balance the potential error events above and below each decision boundary. However, if desired, the objective may be modified to bias the potential error events toward one side of the decision boundary, such as to favor detection of a "1" as opposed to a "0". In such



THIS SHEET HAS BEEN  
REPLACED

case, the values of  $+m$  and  $-m$  would be replaced by unequal values  $+m_1$  and  $-m_2$ , respectively. Similarly, the values of the preselected numbers  $+S$  and  $-S$  for the C counter may also be replaced by unequal values  $+S_1$  and  $-S_2$ , respectively, to bias the adjustment of the decision boundary to favor detection of one preselected data pattern versus the other. Likewise, the preselected value of  $N$  for counter  $T_A$  and that for the counter  $T_B$  may be replaced by constants  $N_1$  and  $N_2$ , respectively, to maximize use of available data where one preselected data pattern is significantly more prevalent than the other preselected data pattern. Hence, in the claims, unless otherwise stated, use of  $m_1$ ,  $m_2$  and  $+S_1$ ,  $-S_2$  and  $N_1$ ,  $N_2$  are intended generically to cover either equal or unequal values for these respective constants.

While the invention has been particularly shown and described with reference to preferred embodiments thereof, it will be understood by those skilled in the art that the foregoing and other changes may be made therein without departing from the spirit, scope and teaching of the invention. Accordingly, the invention herein disclosed is to be considered merely as illustrative and limited in scope only as specified in the claims.



THIS SHEET HAS BEEN  
REPLACED

## Appendix

| Table 1. Sample Functions              |           |            |            |
|----------------------------------------|-----------|------------|------------|
| Precomputed Functions of Sample Values | Threshold | N Patterns | M Patterns |
| <b>Baseline Check</b>                  |           |            |            |
| $H_1 = (y_0 + 2y_1 + y_2)$             | KA        | N1, N2     | M1, M2, M3 |
| $H_2 = (y_0 + 2y_1 + y_2 - y_3)$       | KB        | N4, N5     | M1, M2     |
| <b>Peak Position Check</b>             |           |            |            |
| $H_3 = (y_0 + y_1) - (y_2 + y_3)$      | KX        | N1, N2     | M6, M7     |
|                                        | KY        | N3         | M8         |
| $H_4 = y_0 + y_1 - y_2 + 2y_4$         | KZ        | N1         | M8         |
| $H_5 = y_0 + y_1 - y_2 + y_4 - y_5$    | KW        | N2         | M8A        |

This Sheet Has Been Replaced



THIS SHEET HAS BEEN  
REPLACED

| Table 2. Possible Sequences - State 000 - 6-Sample Look-Ahead |                     |    |    |    |    |    |    |                                |    |    |    |    |    |
|---------------------------------------------------------------|---------------------|----|----|----|----|----|----|--------------------------------|----|----|----|----|----|
| Data<br>Pattern                                               | Binary Data Pattern |    |    |    |    |    |    | Sample Values of Analog Signal |    |    |    |    |    |
|                                                               | d0                  | d1 | d2 | d3 | d4 | d5 | d6 | y0                             | y1 | y2 | y3 | y4 | y5 |
| M1a                                                           | 0                   | 0  | 0  | 0  | 0  | 0  | 0  | 0                              | 0  | 0  | 0  | 0  | 0  |
| M1b                                                           | 0                   | 0  | 0  | 0  | 0  | 0  | 1  | 0                              | 0  | 0  | 0  | 0  | 2  |
| M2                                                            | 0                   | 0  | 0  | 0  | 0  | 1  | 0  | 0                              | 0  | 0  | 0  | 2  | 4  |
| M3a                                                           | 0                   | 0  | 0  | 0  | 1  | 0  | 0  | 0                              | 0  | 0  | 2  | 4  | 2  |
| M3b                                                           | 0                   | 0  | 0  | 0  | 1  | 0  | 1  | 0                              | 0  | 0  | 2  | 4  | 0  |
| M4a                                                           | 0                   | 0  | 0  | 1  | 0  | 0  | 0  | 0                              | 0  | 2  | 4  | 2  | 0  |
| M4b                                                           | 0                   | 0  | 0  | 1  | 0  | 0  | 1  | 0                              | 0  | 2  | 4  | 2  | -2 |
| M5                                                            | 0                   | 0  | 0  | 1  | 0  | 1  | 0  | 0                              | 0  | 2  | 4  | 0  | -4 |
| M6a                                                           | 0                   | 0  | 1  | 0  | 0  | 0  | 0  | 0                              | 2  | 4  | 2  | 0  | 0  |
| M6b                                                           | 0                   | 0  | 1  | 0  | 0  | 0  | 1  | 0                              | 2  | 4  | 2  | 0  | -2 |
| M7                                                            | 0                   | 0  | 1  | 0  | 0  | 1  | 0  | 0                              | 2  | 4  | 2  | -2 | -4 |
| M8a                                                           | 0                   | 0  | 1  | 0  | 1  | 0  | 0  | 0                              | 2  | 4  | 0  | -4 | -2 |
| M8b                                                           | 0                   | 0  | 1  | 0  | 1  | 0  | 1  | 0                              | 2  | 4  | 0  | -4 | 0  |
|                                                               |                     |    |    |    |    |    |    |                                |    |    |    |    |    |
| N1a                                                           | 0                   | 1  | 0  | 0  | 0  | 0  | 0  | 2                              | 4  | 2  | 0  | 0  | 0  |
| N1b                                                           | 0                   | 1  | 0  | 0  | 0  | 0  | 1  | 2                              | 4  | 2  | 0  | 0  | -2 |
| N2                                                            | 0                   | 1  | 0  | 0  | 0  | 1  | 0  | 2                              | 4  | 2  | 0  | -2 | -4 |
| N3a                                                           | 0                   | 1  | 0  | 0  | 1  | 0  | 0  | 2                              | 4  | 2  | -2 | -4 | -2 |
| N3b                                                           | 0                   | 1  | 0  | 0  | 1  | 0  | 1  | 2                              | 4  | 2  | -2 | -4 | 0  |
| N4a                                                           | 0                   | 1  | 0  | 1  | 0  | 0  | 0  | 2                              | 4  | 0  | -4 | -2 | 0  |
| N4b                                                           | 0                   | 1  | 0  | 1  | 0  | 0  | 1  | 2                              | 4  | 0  | -4 | -2 | 2  |
| N5                                                            | 0                   | 1  | 0  | 1  | 0  | 1  | 0  | 2                              | 4  | 0  | -4 | 0  | -4 |



THIS SHEET HAS BEEN  
REPLACED

| Table 3. Classification of (1,7) ML Boundaries |               |             |             |             |             |             |
|------------------------------------------------|---------------|-------------|-------------|-------------|-------------|-------------|
| State                                          | Boundary Type |             |             |             |             |             |
|                                                | A             | B           | X           | Y           | Z           | W           |
| 000                                            | <i>KA2+</i>   | <i>KB2+</i> | <i>KX2+</i> | <i>KY2+</i> | <i>KZ2+</i> | <i>KW2+</i> |
| 100                                            | <i>KA1+</i>   | <i>KB1+</i> | <i>KX1+</i> | <i>KY1+</i> | <i>KZ1+</i> | <i>KW1+</i> |
| 111                                            | <i>KA2-</i>   | <i>KB2-</i> | <i>KX2-</i> | <i>KY2-</i> | <i>KZ2-</i> | <i>KW2-</i> |
| 011                                            | <i>KA1-</i>   | <i>KB1-</i> | <i>KX1-</i> | <i>KY1-</i> | <i>KZ1-</i> | <i>KW1-</i> |

This Sheet Has Been Replaced



Table 4. Program Listing for BOUNDARY

```

V Y BOUNDARY43 D17:010:I:SS
[ 1] *****
[ 2] a Inputs:      Y = Sequence of Analog Sample Values
[ 3] a              D17 = Corresponding Data Sequence
[ 4] a Outputs:    KPOS = Boundaries for Positive States 000 and 100
[ 5] a              KNEG = Boundaries for Negative States 111 and 011
[ 6] a              IIO+1      a Set Index Origin
[ 7] a              KI=(4.6)00      a To accumulate thresholds
[ 8] a              SS=(4.6.2)00    a To accumulate sample sizes
[ 9] a *****Compute State 000, 110, 111, 011 Thresholds when I=1,2,3,4
[10] I=0
[11] DNRZ=DNRZ+0<+/(X(D17ADNRZ+D17)) a Indicator of Positive Phase
[12] AGAIN:I=I+1      a Loop on State
[13] a(I=3)/Y+Y      a Reverse sign for Negative Phase
[14] a(I=3)/DNRZ-DNRZ a Indicator of Negative Phase
[15] ***** Precomputed Functions of Sample Values
[16] H1=Y+(2*10Y)+20Y
[17] H2=H1-(30Y)
[18] H3=Y+(10Y)-(20Y)+30Y
[19] H4=H3+(30Y)+2*(40Y)
[20] H5=H4-(50Y)+(40Y)
[21] ***** E contains all locations of State I within D17
[22] E=DNRZ[E]/E+1+(+D17)11+/(1+D17)
[23] a(I<1 3)/E+DNRZ[E]/E+1+(+D17)11+/(1+D17)
[24] E=(D17[E])/E      a Eliminate Transition States: D17[0]=1
[25] E=(E(D17)-6)/E    a Truncate last 6 samples
[26] *****E01jk... = Locations of Data Pattern (0,1,j,k....)
[27] S=S+E01001+(D17[E+1]AD17[E+4])/E      a N3
[28] S=S+E0101+(D17[E+1]AD17[E+3])/E      a N4,N5
[29] S=S+E01000+(D17[E+1]A(D17[E+3])A(D17[E+4])/E      a N1,N2
[30] S=S+E010001+(D17[E+1]A(D17[E+3])AD17[E+5])/E      a N2
[31] S=S+E010000+(D17[E+1]A(D17[E+3])A(D17[E+4])A(D17[E+5])/E a N1
[32] S=S+E00101+(D17[E+2]AD17[E+4])/E      a N8
[33] S=S+E0010100+(D17[E+2]AD17[E+4]A(D17[E+6])/E      a N8A
[34] S=S+E00100+(D17[E+2]A(D17[E+4])/E      a N6, N7
[35] S=S+E0000+(D17[E+1]A(D17[E+2])A(D17[E+3])/E      a N1,N2,N3
[36] S=S+E00000+(D17[E+1]A(D17[E+2])A(D17[E+3])A(D17[E+4])/E a N1,N2
[37] SS[I:1]+S[3 2 3 1 5 4]      a N Sample sizes
[38] SS[I:2]+S[9 10 8 6 6 7]      a N Sample sizes
[39] *****KI[I:J] is average of cloud means closest to (I,J) Boundary
[40] KI[I:1]+(((+H1[E01000])E01000)+(H1[E0000])E0000)+2
[41] KI[I:2]+(((+H2[E0101])E0101)+(H2[E00000])E00000)+2
[42] KI[I:3]+(((+H3[E01000])E01000)+(H3[E00100])E00100)+2
[43] KI[I:4]+(((+H3[E01001])E01001)+(H3[E00101])E00101)+2
[44] KI[I:5]+(((+H4[E010000])E010000)+(H4[E00101])E00101)+2
[45] KI[I:6]+(((+H5[E010001])E010001)+(H5[E0010100])E0010100)+2
[46] +(I<4)/AGAIN ***** Repeat for four states
[47] KPOS=L.5+KI[1:1].KI[2:]      a Round to nearest integer
[48] KNEG=L.5+KI[3:1].KI[4:]
[49] END:SHOTRESH /      a Display thresholds and Sample Sizes

```



THIS SHEET HAS BEEN  
REPLACED

Table 5. Intermediate Logic and Output Logic

1. Decoded Data and Pointer

$$d_0 = b_0 \bar{c}_0 + \bar{b}_0 c_0$$

$$P = p \bar{b}_0 \bar{c}_0 + p' b_0 c_0$$

2. Intermediate Logic

$$R_1 = A_1 B_1 + X_1 + Y_1 Z_1 W_1 \quad R'_1 = A'_1 B'_1 + X_1 + Y_1 Z_1 W_1$$

$$R_2 = A_2 B_2 + X_2 + Y_2 Z_2 W_2 \quad R'_2 = A'_2 B'_2 + X_2 + Y_2 Z_2 W_2$$

3. Next State

$$a_1 = p \bar{b}_0 \bar{c}_0 + b_0 \bar{c}_0 + \bar{p} b_0 c_0$$

$$b_1 = p \bar{b}_0 \bar{c}_0 + \bar{p} c_0 + \bar{b}_0 c_0$$

$$c_1 = p \bar{b}_0 + \bar{b}_0 c_0 + \bar{R}_1 \bar{b}_0 a_0 + \bar{R}_2 \bar{a}_0 \bar{c}_0 + \bar{p} R'_1 \bar{a}_0 b_0 + \bar{p} R'_2 a_0 c_0$$

This Sheet Has Been Replaced



THIS SHEET HAS BEEN  
REPLACED

| Table 6. State at Time of Detecting $d_1$ |       |                                 |       |
|-------------------------------------------|-------|---------------------------------|-------|
| $d_1$                                     | $b_1$ | Logic Expression                | State |
| 0                                         | 0     | $\overline{d_1} \overline{b_1}$ | 000   |
| 1                                         | 0     | $d_1 \overline{b_1}$            | 100   |
| 0                                         | 1     | $\overline{d_1} b_1$            | 111   |
| 1                                         | 1     | $d_1 b_1$                       | 011   |



THIS SHEET HAS BEEN  
REPLACED

| Table 7. Pattern Identification Logic |                                                |                               |                                                |                           |
|---------------------------------------|------------------------------------------------|-------------------------------|------------------------------------------------|---------------------------|
| Boundary                              | Nearest Above the Boundary                     |                               | Nearest Below the Boundary                     |                           |
|                                       | Data Pattern<br>$d_0, d_1, d_2, d_3, d_4, d_5$ | Logic Expression              | Data Pattern<br>$d_0, d_1, d_2, d_3, d_4, d_5$ | Logic Expression          |
| A                                     | 0, 1, 0, 0, $\emptyset$ , $\emptyset$          | $d_1 \bar{d}_3$               | 0, 0, 0, 0, $\emptyset$ , $\emptyset$          | $\bar{d}_0 d_1 d_2 d_3$   |
| B                                     | 0, 1, 0, 1, 0, $\emptyset$                     | $d_1(d_3 + d_4)$              | 0, 0, 0, 0, $\emptyset$ , $\emptyset$          | $\bar{d}_0 d_1 d_2 d_3$   |
|                                       | 0, 1, 0, 0, 1, 0                               |                               |                                                |                           |
| X                                     | 0, 1, 0, 0, 0, $\emptyset$                     | $d_1 \bar{d}_3 \bar{d}_4$     | 0, 0, 1, 0, 0, $\emptyset$                     | $\bar{d}_1 d_2 \bar{d}_4$ |
| Y                                     | 0, 1, 0, 0, 1, 0                               | $d_1 d_4$                     | 0, 0, 1, 0, 1, 0                               | $\bar{d}_0 d_2 d_4$       |
| Z                                     | 0, 1, 0, 0, 0, 0                               | $d_1 \bar{d}_3 d_4 \bar{d}_5$ | 0, 0, 1, 0, 1, 0                               | $\bar{d}_0 d_2 d_4$       |
| W                                     | 0, 1, 0, 0, 0, 1                               | $d_1 \bar{d}_3 d_5$           | 0, 0, 1, 0, 1, 0                               | $\bar{d}_0 d_2 d_4$       |



THIS SHEET HAS BEEN  
REPLACED

| Table 8. Boundary Adjustment Decision Table |             |               |                                |
|---------------------------------------------|-------------|---------------|--------------------------------|
| Counter Values                              |             |               | Change In Boundary<br>Constant |
| $T_A$                                       | $T_B$       | $C$           |                                |
| $< N$                                       | $\emptyset$ | $\emptyset$   | 0                              |
| $\emptyset$                                 | $< N$       | $\emptyset$   | 0                              |
| $N$                                         | $N$         | $\geq +S$     | add 1                          |
| $N$                                         | $N$         | $\leq -S$     | subtract 1                     |
| $N$                                         | $N$         | $-S < 0 < +S$ | stop adjustments               |



THIS SHEET HAS BEEN  
REPLACED

| Table 9. State at Time of Apparent Error |       |                                   |       |
|------------------------------------------|-------|-----------------------------------|-------|
| $D_{-1}$                                 | $B_1$ | Logic Expression                  | State |
| 0                                        | 0     | $\overline{D_{-1}}\overline{B_1}$ | 000   |
| 1                                        | 0     | $D_{-1}\overline{B_1}$            | 100   |
| 0                                        | 1     | $\overline{D_{-1}}B_1$            | 111   |
| 1                                        | 1     | $D_{-1}B_1$                       | 011   |

This Sheet Has Been Replaced



THIS SHEET HAS BEEN  
REPLACED

| Table 10. Boundary Crossing Detection Logic                 |                                                               |                                                             |                                                               |                                    |
|-------------------------------------------------------------|---------------------------------------------------------------|-------------------------------------------------------------|---------------------------------------------------------------|------------------------------------|
| Expected<br>Pattern Below<br>$D_0, D_1, D_2, D_3, D_4, D_5$ | Logic<br>Expression                                           | Received<br>Pattern Above<br>$d_0, d_1, d_2, d_3, d_4, d_5$ | Logic<br>Expression                                           | Boundary<br>(Direction<br>Crossed) |
| 0, 0, 0, 0, 0, 0                                            | $\overline{D_0} \overline{D_1} \overline{D_2} \overline{D_3}$ | 0, 1, 0, 0, 0, 0                                            | $d_1 \overline{d_3}$                                          | A (up)                             |
| 0, 0, 0, 0, 0, 0                                            | $\overline{D_0} \overline{D_1} \overline{D_2} \overline{D_3}$ | 0, 1, 0, 1, 0, 0<br>0, 1, 0, 0, 1, 0                        | $d_1(d_3 + d_4)$                                              | B (up)                             |
| 0, 0, 1, 0, 0, 0                                            | $\overline{D_0} \overline{D_2} \overline{D_4}$                | 0, 1, 0, 0, 0, 0                                            | $d_1 \overline{d_3} \overline{d_4}$                           | X (up)                             |
| 0, 0, 1, 0, 1, 0                                            | $\overline{D_0} \overline{D_2} D_4$                           | 0, 1, 0, 0, 1, 0                                            | $d_1 \overline{d_3} d_4$                                      | Y (up)                             |
| 0, 0, 1, 0, 1, 0                                            | $\overline{D_0} \overline{D_2} D_4$                           | 0, 1, 0, 0, 0, 0                                            | $d_1 \overline{d_3} \overline{d_4} \overline{d_5}$            | Z (up)                             |
| 0, 0, 1, 0, 1, 0                                            | $\overline{D_0} \overline{D_2} D_4$                           | 0, 1, 0, 0, 0, 1                                            | $d_1 \overline{d_3} d_5$                                      | W (up)                             |
| Expected<br>Pattern Above<br>$D_0, D_1, D_2, D_3, D_4, D_5$ | Logic<br>Expression                                           | Received<br>Pattern Below<br>$d_0, d_1, d_2, d_3, d_4, d_5$ | Logic<br>Expression                                           | Boundary<br>(Direction<br>Crossed) |
| 0, 1, 0, 0, 0, 0                                            | $D_1 \overline{D_3}$                                          | 0, 0, 0, 0, 0, 0                                            | $\overline{d_0} \overline{d_1} \overline{d_2} \overline{d_3}$ | A (down)                           |
| 0, 1, 0, 1, 0, 0<br>0, 1, 0, 0, 1, 0                        | $D_1(D_3 + D_4)$                                              | 0, 0, 0, 0, 0, 0                                            | $\overline{d_0} \overline{d_1} \overline{d_2} \overline{d_3}$ | B (down)                           |
| 0, 1, 0, 0, 0, 0                                            | $D_1 \overline{D_3} \overline{D_4}$                           | 0, 0, 1, 0, 0, 0                                            | $\overline{d_0} \overline{d_2} \overline{d_4}$                | X (down)                           |
| 0, 1, 0, 0, 1, 0                                            | $D_1 \overline{D_3} D_4$                                      | 0, 0, 1, 0, 1, 0                                            | $\overline{d_0} \overline{d_2} d_4$                           | Y (down)                           |
| 0, 1, 0, 0, 0, 0                                            | $D_1 \overline{D_3} \overline{D_4} \overline{D_5}$            | 0, 0, 1, 0, 1, 0                                            | $\overline{d_0} \overline{d_2} d_4$                           | Z (down)                           |
| 0, 1, 0, 0, 0, 1                                            | $D_1 \overline{D_3} D_5$                                      | 0, 0, 1, 0, 1, 0                                            | $\overline{d_0} \overline{d_2} d_4$                           | W (down)                           |

This Sheet Has Been Replaced





### Claims

1. A method for improving data detection reliability in a coded signal processing channel in which digital electrical signals representing digital sample values are generated from analog signals read by a transducer from digital data recorded on a data storage medium, comprising the steps of:

providing a plurality of threshold constants whose values define decision boundaries for use in determining whether detected data corresponding to a coded sequence of a runlength limited (RLL) code is a "1" or a "0";

collecting actual values of one of a plurality of linear functions of the digital sample values of the analog data corresponding to multiple occurrences of two preselected data patterns that are determined to be closest above and closest below, respectively, a preselected decision boundary for forming corresponding clusters of such actual values above and below the decision boundary;

determining a focal point for each cluster; and

establishing the threshold constant for said preselected decision boundary at a value substantially midway between said focal points.

2. The method of claim 1, wherein the threshold constant is established only if the number of actual values in both of said clusters exceeds a preselected number.

3. A method for improving data detection reliability in a coded signal processing channel in which digital electrical signals representing digital sample values are generated from analog signals read by a transducer from digital data recorded on a data





THIS SHEET HAS BEEN  
REPLACED

storage medium, comprising the steps of:

providing a plurality of threshold constants whose values define decision boundaries for use in determining whether detected data corresponding to a coded sequence of a runlength limited (RLL) code is a "1" or a "0";

determining potential error events in the detected data by counting in two separate counters numbers of times actual values of one of a plurality of linear functions of the digital sample values corresponding to one preselected data pattern are within  $m_1$  units above and of the digital sample values corresponding to another preselected data pattern are within  $m_2$  units below, respectively, a preselected one of said boundaries;

accumulating a difference count having a magnitude and sign denoting difference between counts in said separate counters; and

responsive to a predesignated event, adjusting the preselected decision boundary upwardly if the difference count at least equals  $+S_1$  or downwardly if the difference count is not greater than  $-S_2$ .

4. The method of claim 3, wherein said one preselected data pattern and another preselected data pattern are those closest above and closest below, respectively, said preselected boundary.

5. The method of claim 3, including the steps of:

establishing initial values for the threshold constants;

setting the threshold constants at said initial values; and  
then

optimizing the threshold constants by executing the counting and adjusting steps.





THIS SHEET HAS BEEN  
REPLACED

6. The method of claim 3, including the step of providing as the recorded data a stream of known digital data having a length and diversity that assures it is statistically reliable.

7. The method of claim 3, including executing the adjusting step during a read operation.

8. The method of claim 3, including executing the adjusting step during a time out between successive read operations.

9. The method of claim 3, including the steps of:

calculating an error margin distance representing the difference between actual values of said one linear function of the digital sample values and a corresponding one of the threshold constants; and

using said error margin distance corresponding to the preselected data patterns for determining whether each actual value is within  $m_1$  units above or  $m_2$  units below the preselected boundary.

10. The method of claim 3, wherein  $m_1 = m_2$ .

11. The method of claim 3, wherein said predesignated event is the counting of a total of at least  $N_1$  one preselected data patterns and at least  $N_2$  another preselected data patterns, irrespective of distance from the preselected boundary.

12. The method of claim 11, wherein  $N_1 = N_2$ .

13. The method of claim 11, wherein  $S_1 < N_1$  and  $S_2 < N_2$ .

14. Apparatus for improving data detection reliability in a coded signal processing channel in which digital electrical signals representing digital sample values are generated from analog



THIS SHEET HAS BEEN  
REPLACED

signals read by a transducer from digital data recorded on a data storage medium, comprising:

means (308) for storing a plurality of threshold constants whose values define decision boundaries for use in determining whether detected data corresponding to a coded sequence of a runlength limited (RLL) code is a "1" or a "0";

means for determining potential error events in the detected data (d), comprising two separate counters ( $M_A$ ,  $M_B$ ) for counting numbers of times actual values of one of a plurality of linear functions of the digital sample values corresponding to one preselected data pattern are within  $m_1$  units above and of the digital sample values corresponding to another preselected data pattern are within  $m_2$  units below, respectively, a preselected one of said boundaries;

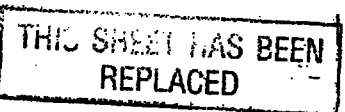
a difference counter (C) for accumulating a count having a magnitude and sign denoting difference between counts in said separate counters; and

circuitry (Table 8) responsive to the count in one of said separate counters exceeding a preselected number  $N_1$  and in the other separate counter exceeding a preselected number  $N_2$  a predesignated control event for adjusting the preselected decision boundary upwardly if the count in the difference counter at least equals  $+S_1$  or downwardly if the latter count is not greater than  $-S_2$ .

15. The apparatus of claim 14, wherein said one preselected data pattern and another preselected data pattern are those closest above and closest below, respectively, said preselected boundary.

16. The apparatus of claim 14, including means (380) for providing as the recorded data a stream of known digital data having a length





and diversity that assures it is statistically reliable.

17. The apparatus of claim 14, including two total counters ( $T_A$ ,  $T_B$ ) for counting, and initiating said predesignated control event upon occurrence of, both  $N_1$  one preselected data patterns and  $N_2$  another preselected data patterns, respectively, irrespective of how far from the preselected boundary.

18. The apparatus of claim 14, including:

means (324) for calculating an error margin distance representing the difference between actual values of said one linear function of the digital sample values and a corresponding one of the threshold constants;

means (356, 358) for comparing said error margin distance with  $+m_1$  and  $-m_2$  for determining whether each actual value is within  $m_1$  units above or  $m_2$  units below the preselected boundary; and

means (351, 360, 362) for gating said separate counters for counting only when said one and said another preselected data patterns occur, respectively.

19. The apparatus of claim 14 wherein  $m_1 = m_2$ .

20. The apparatus of claim 14 wherein  $N_1 = N_2$ .

21. The apparatus of claim 14 wherein  $S_1 = S_2$ .

22. Apparatus for improving data detection reliability in a coded signal processing channel in which digital electrical signals representing digital sample values are generated from analog signals read by a transducer from digital data recorded on a data storage medium, comprising:

means (308) for storing a plurality of threshold constants



THIS SHEET HAS BEEN  
REPLACED

whose values define decision boundaries for use in determining whether detected data corresponding to a coded sequence of a runlength limited (RLL) code is a "1" or a "0";

means (including 397,  $W_{above}$ ,  $W_{below}$ , 360, 362) for determining potential error events in the detected data;

means (including 399, 398, 395, 396) operative during generation of a sequence of digital sample values of the analog signals corresponding to known recorded data for intentionally moving a selected one of the decision boundaries upward  $m_1$  units and then downward  $m_2$  units from an initial value;

two separate counters ( $M_A$ ,  $M_B$ ) operative while the selected boundary is moved upward and then downward for counting numbers of times a preselected above data pattern is detected as a related below data pattern and the said related below data pattern is detected as the said above data pattern, respectively;

a difference counter (C) for accumulating a count having a magnitude and sign denoting difference between counts in said separate counters; and

circuitry (Table 8) responsive to a predesignated control event for adjusting the selected decision boundary upwardly if the count in the difference counter at least equals  $+S_1$  or downwardly if the latter count is not greater than  $-S_2$ .

23. Apparatus for improving data detection reliability in a coded signal processing channel in which digital electrical signals representing digital sample values are generated from analog signals read by a transducer from digital data recorded on a data storage medium, comprising:

means (308) for storing a plurality of threshold constants





THIS SHEET HAS BEEN  
REPLACED

whose values define decision boundaries for use in determining whether detected data corresponding to a coded sequence of a runlength limited (RLL) code is a "1" or a "0";

circuitry for determining potential error events in the detected data by generating, for a selected state, first and second outputs ( $W_{above}$ ,  $W_{below}$ ) corresponding to one preselected data pattern and another preselected related data pattern, respectively, above and below a preselected one of said decision boundaries;

first and second counters ( $T_A$ ,  $T_B$ ) for respectively counting up to preselected numbers  $N_1$  and  $N_2$  total occurrences of said one and another preselected data patterns above and below said preselected one boundary, respectively;

means (351,  $W_{above}$ ,  $W_{below}$ , 356,  $D_{above}$ , 358,  $D_{below}$ ) establishing error margin distances  $m_1$  units above and  $m_2$  units below, respectively, said preselected one boundary;

third and fourth counters ( $M_A$ ,  $M_B$ ) for counting numbers of times actual values of a linear function of digital sample values corresponding to said one and another preselected data patterns are respectively within  $m_1$  units and  $m_2$  units, respectively, of said preselected boundary;

a difference counter (C) for accumulating a count having a magnitude and sign denoting difference between counts in said third and fourth counters; and

circuit means, conditioned by said first and second counters counting to  $N_1$  and  $N_2$ , respectively, for adjusting the preselected one boundary one unit up or one unit down provided the count in the difference counter has reached number  $+S_1$  or  $-S_2$ , respectively, where  $S_1 < N_1$  and  $S_2 < N_2$ .

24. The apparatus of claim 23, said circuit means being responsive



THIS SHEET HAS BEEN  
REPLACED

to the counts in said first and second counters reaching  $N_1$  and  $N_2$ , respectively, before the count in the difference counter reaches  $+S_1$  or  $-S_2$  to terminate boundary adjustment, for thereby defining preselected error margins for said data patterns above and below said preselected boundary.

25. The apparatus of claim 23, including means for resetting all of said counters whenever the boundary is adjusted.

26. The apparatus of claim 23, including circuitry responsive to the count in either said third or fourth counter exceeding a preselected number  $Q$  less than  $N_1$  or  $N_2$  to produce a signal (U) indicating performance of the channel is unacceptable.

27. The apparatus of claim 23, including means providing a signal (E) for enabling each of said counters during reading of data from the storage medium to permit initiation of a boundary adjustment.

28. The apparatus of claim 23, comprising:

means providing a plurality of sets of preselected data patterns for which the actual values of the functions of digital sample values are to be measured against related decision boundaries; and

means including a multiplexor (MUX) for selecting one set of the data patterns at a time for measurement and boundary adjustment.

