SOFT-DECISION THRESHOLD DECODING OF CONVOLUTIONAL CODES

R.M.F. Goodman*, B.Sc., Ph.D.

W.H. Ng*, M.S.E.E.

Summary

Existing majority-decision threshold decoders have so far been limited to hard-decision decoding only, with a consequent loss in coding gain. In this paper a new method for implementing soft-decision majority threshold decoding of convolutional codes is introduced and explained. The method is illustrated by describing soft-decision decoders for a simple random error-correcting code, and also for a more complex diffuse random-and-burst convolutional coding scheme.

1. Introduction

Binary convolutional codes have been shown to exhibit extremely good error-control properties under both Gaussian and burst noise conditions. In the case of the additive white Gaussian channel, there are several powerful convolutional decoding schemes (sequential decoding, Viterbi decoding) that yield high coding gains (5dB at a sink bit error rate of 10↑-5). Unfortunately, the hardware complexity of such schemes is high, as the decoders are essentially large special-purpose computers. In addition, the burst-noise performance of these powerful schemes tends to be disappointing in comparison with convolutional code systems designed specifically for burst-error correction.

The system designer is therefore often interested in convolutional decoding schemes that sacrifice a few dB of coding gain in order to achieve low hardware complexity with reasonably good burst and random error performance. For example, on the H.F. radio channel. Threshold decoding is one method of achieving this aim.

Majority-decision threshold decoding (ref.1), is in terms of hardware, one of the simplest convolutional decoding schemes possible, and is applicable to a wide range of time-varying and fading channels. However, because the scheme is not optimum, some coding gain is lost.

In this paper we present a soft-decision majority threshold decoding scheme that improves on the performance achievable with existing hard-decision decoders, thereby making up some of the lost coding gain, whilst still retaining the inherent hardware simplicity of threshold decoding. It has been shown (ref.2) that the maximum increase in coding gain that can be achieved by using soft-decision is about 2 dB for infinite-level quantisation, and that the degradation involved in using equal-spacing 8-level quantisation (as asummed in this paper) is only 0.2 dB. We therefore expect a maximum improvement of about 1.8 dB for soft-decision majority threshold decoding when compared with existing hard-decision decoders.

In this paper we firstly outline hard-decision majority threshold decoding and then introduce our soft-decision scheme using a simple constraint length 2 code as an example. Next we describe our general method for soft-decision decoding of multiple error-correcting codes, using a diffuse random-and-burst error-correcting scheme as an example.

* Department of Electronic Engineering, University of Hull.

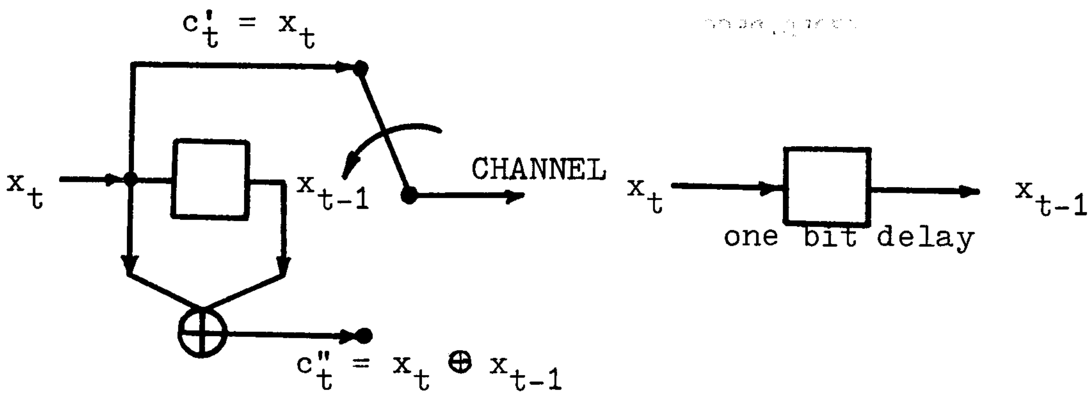


Fig. 1 - A simple convolutional encoder where $\underset{\sim}{g}$ = 11 01

2. Hard-decision majority threshold decoding

A single-generator systematic convolutional code is one in which each information digit is encoded into V code digits (giving a message throughput rate of 1/V), the first of which is the unchanged information digit. In general, such a code is generated by a K segment generator sequence $g = g(1)\ g(2)\ g(4)\ \ldots\ g(2^{K-1})$, where K is the constraint length of the code in segments, and each segment contains V digits. For simplicity, we restrict our discussion in this paper to rate one-half codes.

Let us consider a rate one-half systematic code with constraint length K=2 segments, to review the basic hard-decision majority threshold decoding technique. The encoder for this single code is shown in Fig.1, and consists of only a single one-bit delay element and a single modulo-2 adder (exclusive-OR gate). Given a sequence of information digits $x = \ldots\ x_{t-1}\ x_t\ x_{t+1}\ \ldots$, where t denotes the time unit of the information digit x_t, each information digit is encoded into two code digits c'_t and c''_t. $c'_t = x_t$ is the unaltered information digit x_t, and $c''_t = x_{t-1} \oplus x_t$ is a parity check sum based on the present information digit x_t and the K-1 = 1 previous information digits. For serial transmission the coded digits are sent to the channel in order $c'_t\ c''_t$ by appropriate action of the switch. The encoder/décoder configuration for this code is shown in Fig.2. On the left of the diagram, the information digit x_t is encoded into c'_t and c''_t; in the middle, two noise digits n'_t and n''_t corrupt the coded digits c'_t and c''_t respectively; on the right is the decoder which realises the (hard-decision) single-error-correction capability of the code. The decoding action is explained with reference to the six points, a, b, c, S_1, S_2, and $\tilde{n}'_{t-1}$. The six points are interpreted as follows:

$$a = x_t \oplus n'_t$$

$$b = x_{t-1} \oplus n'_{t-1}$$

$$c = x_t \oplus x_{t-1} \oplus n''_t$$

$$S_1 = a \oplus b \oplus c = (x_t \oplus n'_t) \oplus (x_{t-1} \oplus n'_{t-1}) \oplus (x_t \oplus x_{t-1} \oplus n''_t)$$

$$S_2 = (x_{t-1} \oplus n'_{t-1}) \oplus (x_{t-2} \oplus n'_{t-2}) \oplus (x_{t-1} \oplus x_{t-2} \oplus n''_{t-1})$$

$$\tilde{n}'_{t-1} = \begin{cases} 1 \text{ if } S_1 = S_2 = 1 \\ 0 \text{ otherwise} \end{cases}$$

by cancelling information digits, S_1 and S_2 become:

$$\begin{aligned} S_1 &= n'_t \oplus n''_t \oplus n'_{t-1} \\ S_2 &= \qquad\qquad n'_{t-1} \oplus n''_{t-1} \oplus n'_{t-2} \end{aligned} \qquad \ldots\ldots (1)$$

and it can be seen that the two parity check equations S_1, S_2 are orthogonal on the noise digit n'_{t-1}. Thus if a single error occurs anywhere in the 5 digit span covered by the orthogonal check sums, the only case when $S_1 = S_2 = 1$ is when $n'_{t-1} = 1$. In the decoder, the AND gate sends an estimate $\tilde{n}'_{t-1}$ of n'_{t-1} to cancel the noise digit n'_{t-1} from the received digit $(x_{t-1} \oplus n'_{t-1})$, and thus produce an estimate $\tilde{x}_{t-1}$ of the transmitted digit x_{t-1}. From equation (1) it can be seen that if more than one error occurs in the 5 digit span covered by $\{S_1, S_2\}$, then the error correction capability of the code is exceeded and the decoded digit $\tilde{x}_{t-1}$ may be in error.

The decoder described above can be improved by the use of feedback. This is because if we are concerned with decoding x_{t-1} at the present moment, then x_{t-2} has already been decoded. We therefore have available an estimate of the noise digit n'_{t-2} before we decode x_{t-1}. Therefore S_2

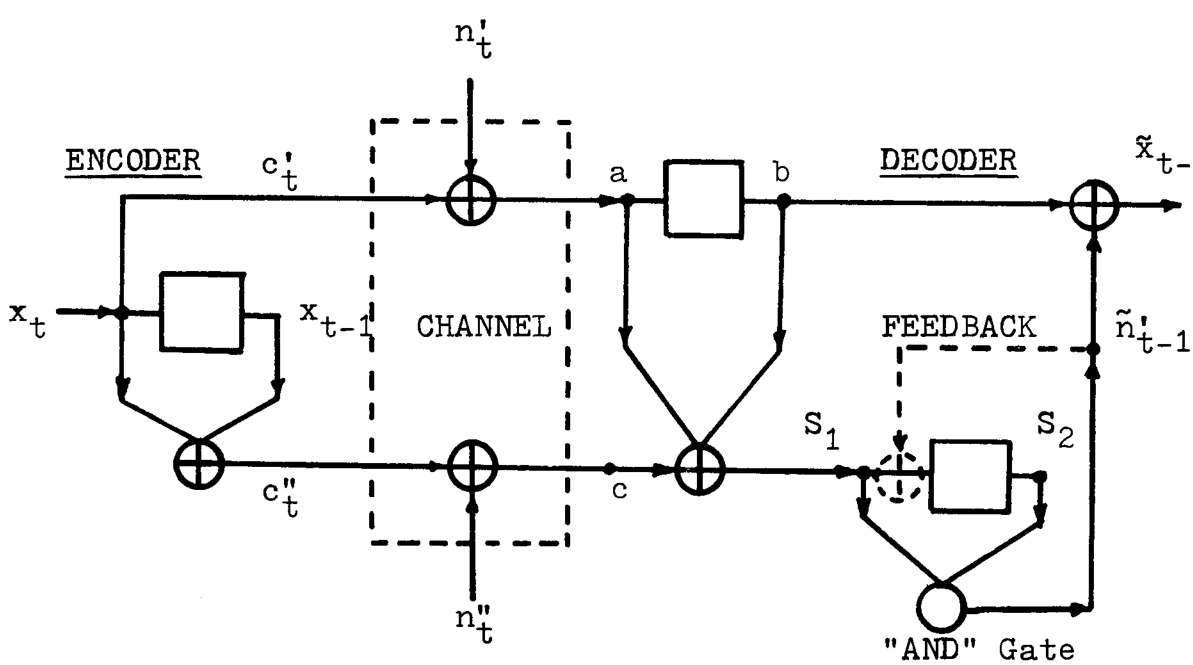


Fig.2 - A simple hard-decision majority threshold encoder/decoder

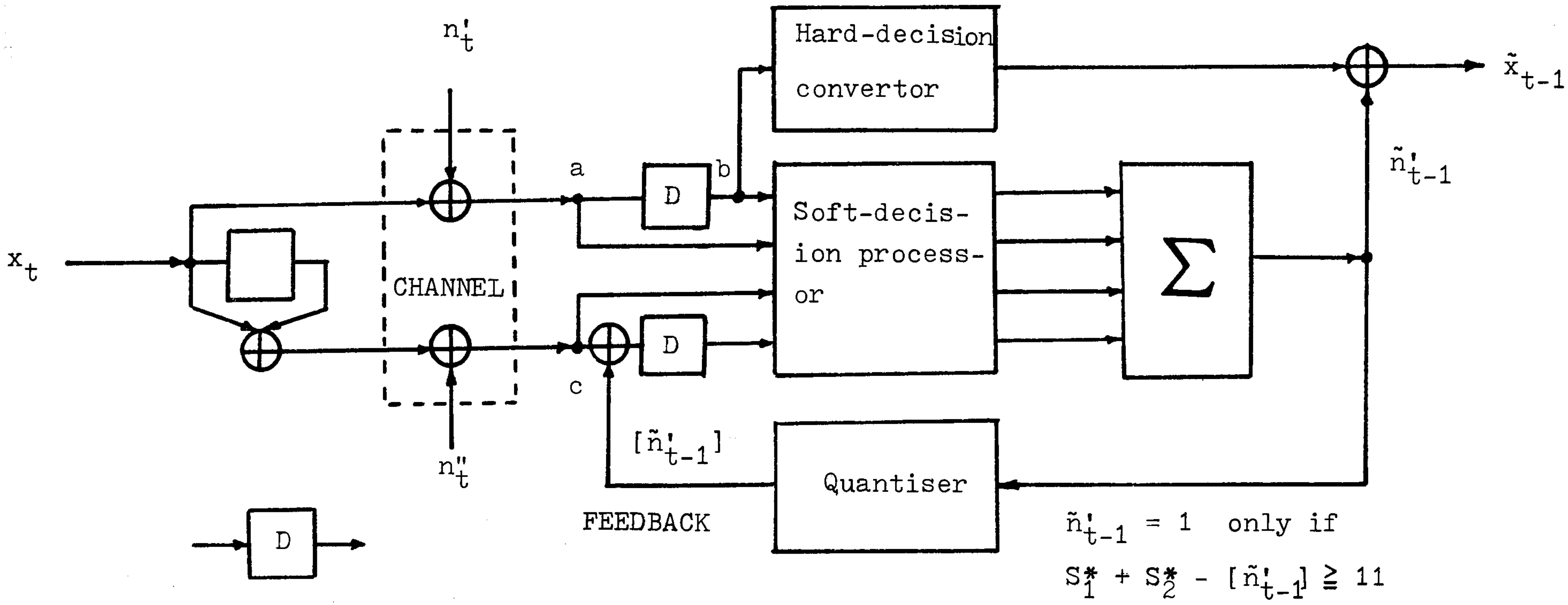
can be simplified by feeding back $\tilde{n}'_{t-2}$ to cancel n'_{t-2} in equation (1). We may then replace S_2 with $\tilde{S}_2 = S_2 \oplus n'_{t-2} = n'_{t-1} \oplus n''_{t-1} \oplus n'_{t-2} \oplus \tilde{n}'_{t-2}$. If the estimate $\tilde{n}'_{t-2}$ is correct, that is $\tilde{n}'_{t-2} = n'_{t-2}$, then $\tilde{S}_2 = n'_{t-1} \oplus n''_{t-1}$. This means that provided the previously decoded digit was correct, the decoder check sums $\{S_1, \tilde{S}_2\}$ only span 4 digits, and can therefore correct a single error anywhere in 4 digits as opposed to 5 digits in the previous case. A decoder that makes use of past decisions to simplify S_2 to $\tilde{S}_2$ is called a feedback decoder, whilst a decoder that does not use past decisions is called a definite decoder.

In general, if it is possible to form a set of 2e parity check equations which are orthogonal on a specified noise digit, then it is possible to build a hard-decision majority threshold decoder which can correct any combinations of e or fewer errors over one constraint span. Figure 3 shows the encoder/decoder arrangement for a triple error-correcting rate one-half (24,12) majority decoder which has K=12, and an effective constraint length of 22 digits within which 3 or fewer errors can be corrected. This decoder can achieve a coding gain of 1.85 dB at a sink bit error rate of 10↑-5 on the binary symmetric channel (which is comparable to the (23,12) perfect Golay code), and can be built with only 16 standard integrated circuits (which is much less than that required to decode the Golay code).

3. Soft-decision majority threshold deocding

In this section we introduce our new method for soft-decision majority threshold decoding. Our basic approach is to derive a modified set of orthogonal check sums $\{S_i\}$ which can be used to estimate each noise digit in the soft-decision sense.

Firstly, let us assume that each received digit is quantised into $Q = 8$ levels, and can therefore be expressed as a 3 digit binary number, or the BCD equivalent. For example, [000] = 0, [001] = 1, [010] = 2, ...[111] = 7. The x_t are therefore expressed as [000] when $x_t = 0$, or [111] when $x_t = 1$, in the soft-decision sense. The noise digits are expressed in a similar manner but can take any intermediate value between 0 and 7, that is, $0 = [000] \leq [n'_{t-j}] \leq [111] = 7$, where the square brackets indicate a quantised or soft-decision noise digit. Note that the most significant digit of a quantised digit is the hard decision digit itself. For example $[n'_{t-j}] = [010]$ implies $n'_{t-j} = 0$, and $[n'_{t-j}] = [110]$ implies $n'_{t-j} = 1$. Let us define d_h to be the hard-decision minimum distance between the two halves of the initial code tree. The guaranteed error-correcting capability of the code over K segments is then e_h digits where e_h is the largest integer satisfying $e_h \leq (d_h - 1)/2$. The simple code used in section 2 has $d_h = 3$, and is therefore a single error-correcting code. In the soft decision sense, the minimum distance of a code is given by



One soft-decision bit delay

Fig. 4 - Soft-decision majority decoder modified from Fig. 2

$d_s = (Q-1) \times d_h$, and its error correction capability is e_s soft-decision digits, where e_s is the largest integer satisfying $e_s \leq (d_s - 1)/2$. The simple example code therefore has $d_s = (8-1) \times d_h = 21$, and $e_s = 10$. Therefore, the relative error-correction capability

$$\frac{\text{(error-correction capability)}}{(Q-1) \times \text{(No. of coded digits involved in the decoding decision)}}$$

is increased from $[(d_h-1)/2]/4 = 0.25$ for hard-decision, to $[(d_s-1)/2]/(7\times4) = 0.36$ for soft decision.

We may now write the soft-decision orthogonal check sums for our example code by modifying equation (1) as follows:

$$\begin{aligned} S_1^* &= [n'_t] + [n''_t] + [n'_{t-1}] \\ S_2^* &= \qquad\qquad\quad [n'_{t-1}] + [n''_{t-1}], \end{aligned} \qquad \ldots\ldots(2)$$

where feedback has been used to produce S_2^*. We may then let the estimated noise digit $\tilde{n}'_{t-1} = 0$ if $S_1^* + S_2^* - [n'_{t-1}] \leq (d_s-1)/2 = 10$; or let $\tilde{n}'_{t-1} = 1$ if $S_1^* + S_2^* - [n'_{t-1}] > 10$. That is, we assume that the estimated noise digit $\tilde{n}'_{t-1}$ is 1 only when the 4 soft-decision noise digits involved in the check sums have a total soft-decision weight of greater than $(d_s-1)/2 = 10$. Before we can implement this scheme, however, we need to derive the actual soft-decision levels of each of the 4 noise digits involved in the decoding equations (2). At this point the fundamental difference between hard and soft decision is revealed. This is that we cannot <u>directly</u> obtain the quantised noise digits, because any decoder can only quantise received digits and not noise digits. The noise digits therefore have to be obtained by a process of estimation as follows.

Refer to Fig. 2. The soft-decision received digit at point a is $[x_t \oplus n'_t]$. As we use 8-level quantisation the information digit x_t is estimated as: let $\tilde{x}_t = 0$ if $[x_t \oplus n'_t] \leq 3$, or let $x_t = 1$ if $[x_t \oplus n'_t] \geq 4$. Then, the estimated noise digit $[\tilde{n}'_t]$ is derived from the following equations: $[\tilde{n}'_t] = [x_t \oplus n'_t]$ if $[x_t \oplus n'_t] \leq 3$, or $[\tilde{n}'_t] = [\tilde{x}_t] \oplus [x_t \oplus n'_t]$ if $[x_t \oplus n'_t] \geq 4$. In a similar manner, the received digit $[x_{t-1} \oplus n'_{t-1}]$ at point b in Fig.2 is used to give the estimate $\tilde{x}_{t-1}$, and $[\tilde{n}'_{t-1}]$ is derived from this estimate. From the received digit $[x_t \oplus x_{t-1} \oplus n''_t]$ at point c, and from the estimates of x_t and x_{t-1}, the estimate of n''_t is derived as: $[\tilde{n}''_t] = [x_t \oplus x_{t-1} \oplus n''_t] \oplus [\tilde{x}_t] \oplus [\tilde{x}_{t-1}]$. Therefore, the value of S_1^* can be calculated by taking the sum of the values for the three noise digits $[\tilde{n}'_t]$, $[\tilde{n}'_{t-1}]$ and $[\tilde{n}''_t]$. Similarly, the value of S_2^* can be obtained, and a decision on $\tilde{n}_{t-1}$ taken.

Fig. 4 shows the soft-decision threshold decoder for the rate one-half code used above, and the following example illustrates its operation.

Let us assume that $x_t = x_{t-1} = 0$, that the noise digits are $[n'_{t-1}] = [101]$, $[n''_{t-1}] = [100]$, $[n'_t] = [001]$, $[n''_t] = [000]$, and that the decoder has not accepted any previous decoding error. Note that as $n'_{t-1} = n''_{t-1} = 1$, a hard-decision decoder would decode $\tilde{x}_{t-1} = 1$ thus giving a decoding error. Using the following soft-decision procedure, however, x_{t-1} can be decoded correctly.

(1) Because the received digit $[x_t \oplus n'_t] = [001]$, we let $\tilde{x}_t = 0$ and $[\tilde{n}'_t] = [x_t \oplus n'_t] = [001]$.

(2) Because the received digit $[x_{t-1} \oplus n'_{t-1}] = [101]$, we let $\tilde{x}_{t-1} = 1$ and $[\tilde{n}'_{t-1}] = [\tilde{x}_{t-1}] \oplus [x_{t-1} \oplus n'_{t-1}] = [010]$

(3) Because the received digit $[x_t \oplus x_{t-1} \oplus n''_t] = [000]$, we let $[\tilde{n}''_t] = [\tilde{x}_t] \oplus [\tilde{x}_{t-1}] \oplus [x_t \oplus x_{t-1} \oplus n''_t] = [111]$.

Finally, $[\tilde{n}''_{t-1}] = [\tilde{x}_{t-1}] \oplus [\tilde{x}_{t-2}] \oplus [x_{t-1} \oplus x_{t-2} \oplus n''_{t-1}] = [\tilde{x}_{t-1}] \oplus [x_{t-1} \oplus n''_{t-1}] = [111] \oplus [100\ ] = [011]$, and all the noise digits are determined.

(5) By using ordinary addition:

$$S_1{*} = [\tilde{n}'_t] + [\tilde{n}'_{t-1}] + [\tilde{n}''_t] = [001] + [010] + [111] = 10,$$

and $S_2{*} = [\tilde{n}'_{t-1}] + [\tilde{n}''_{t-1}] = [010] + [011] = 5$

(6) The value of $S_1{*} + S_2{*} - [\tilde{n}'_{t-1}]$ is then $10 + 5 - 2 = 13$ which is greater than $(d_s - 1)/2 = 10$, and therefore indicates $\tilde{n}'_{t-1} = 1$. This however contradicts our assumption of $\tilde{x}_{t-1} = 1$ which gave $[\tilde{n}'_t] = [010]$. Thus $\tilde{x}_{t-1} = 0$ and the hard-decision received digit $x_{t-1} \oplus n_{t-1}$ is corrected by the modulo-2 addition of $\tilde{n}'_{t-1}$. Note that by assumming $\tilde{x}_{t-1} = 0$, and recalculating steps 2,3, and 4, we have $[\tilde{n}'_{t-1}] = [101]$ $[\tilde{n}''_t] = [000]$, and $[\tilde{n}''_{t-1}] = [100]$. Thus $S_1{*} = 6$, $S_2{*} = 9$, and $S_1{*} + S_2{*} - [\tilde{n}'_{t-1}] = 10$, there is no contradiction in $\tilde{n}'_{t-1}$, and therefore x_{t-1} is correctly decoded as $\tilde{x}_{t-1} = 0$.

For a simple soft-decision decoder described above the value of $S_1{*} + S_2{*} - [\tilde{n}'_{t-1}]$ can be calculated and the estimate of $\tilde{n}'_{t-1}$ indicates directly whether or not a contradiction exists in the original assumption $\tilde{x}_{t-1}$. With a multiple error-correcting code there are more than two orthogonal check sums and it is not convenient to directly derive the value of $\tilde{x}_{t-1}$. In these cases our method involves the decoder computing the value of $S_1{*} + S_2{*} - [\tilde{n}'_{t-1}]$ for both cases $\tilde{x}_{t-1} = 0$ and $\tilde{x}_{t-1} = 1$.

The decoding decision is then based on which assumption gives the smaller value of $S_1{*} + S_2{*} - [\tilde{n}'_{t-1}]$. A more complex decoder illustrating this point is outlined in the next section.

4. Soft-decision diffuse threshold decoding

In general, most real channels are time-varying and subject to bursts of errors. Diffuse convolutional coding is one method of providing burst-and-random error-correction capability, and is most suitable for a channel in which the bursts are diffuse. That is, one in which (relatively) low density bursts occur on a noise background whose random error rate is relatively high.

Fig.5 shows a hard-decision diffuse convolutional coding system that uses majority-decision threshold decoding. The system is rate one-half, Δ-diffuse, and double error correcting. By assuming that all previous decoding decisions were correct, it is possible to form four check-sums orthogonal on the noise digit $n'_{t-3\Delta-1}$ as follows:

$$S_t = n'_t \oplus n'_{t-\Delta} \oplus n'_{t-2\Delta} \oplus n'_{t-3\Delta-1} \oplus n''_t$$

$$S_{t-1} \oplus S_{t-\Delta-1} = n'_{t-1} \oplus n'_{t-3\Delta-1} \oplus n''_{t-1} \oplus n''_{t-\Delta-1}$$

$$S_{t-2\Delta-1} = n'_{t-2\ -1} \oplus n'_{t-3\Delta-1} \oplus n''_{t-2\Delta-1}$$

$$S_{t-3\Delta-1} = n'_{t-3\ -1} \oplus n''_{t-3\ -1}$$

This code corrects any pattern of two or fewer scattered errors among the 11 received digits involved in the orthogonal check sums, as well as any burst of length b = 2Δ or less, given a guard space of 6Δ + 2 digits between bursts (ref.3).

Following the general procedure outlined in the last section we may now develop the soft-decision diffuse convolutional coding scheme, with some extra modifications. Our basic approach is to calculate the algebraic sum of four soft-decision noise sums which are orthogonal on the information digit $x_{t-3\Delta-1}$, for both the assumptions $\tilde{x}_{t-3\Delta-1} = 0$, and $\tilde{x}_{t-3\Delta-1} = 1$.

The algebraic sum of the noise sums is given by:

$$S_i = [\tilde{n}''_t] + [\tilde{n}''_{t-1} \oplus n''_{t-\Delta-1}] + [n''_{t-2\Delta-1}] + [n''_{t-3\Delta-1}],$$

where i = 0 if we let $\tilde{x}_{t-3\Delta-1} = 0$, and i = 1 if $\tilde{x}_{t-3\Delta-1} = 1$. The decoding decision depends on the value of S_o and S_1: We decode $x_{t-3\Delta-1} = 0$ if $S_o \leq S_1$ or $x_{t-3\Delta-1} = 1$ if $S_o \geq S_1$.

The soft-decision decoder for this code is shown in Fig. 6, and the decoding procedure is explained as follows:

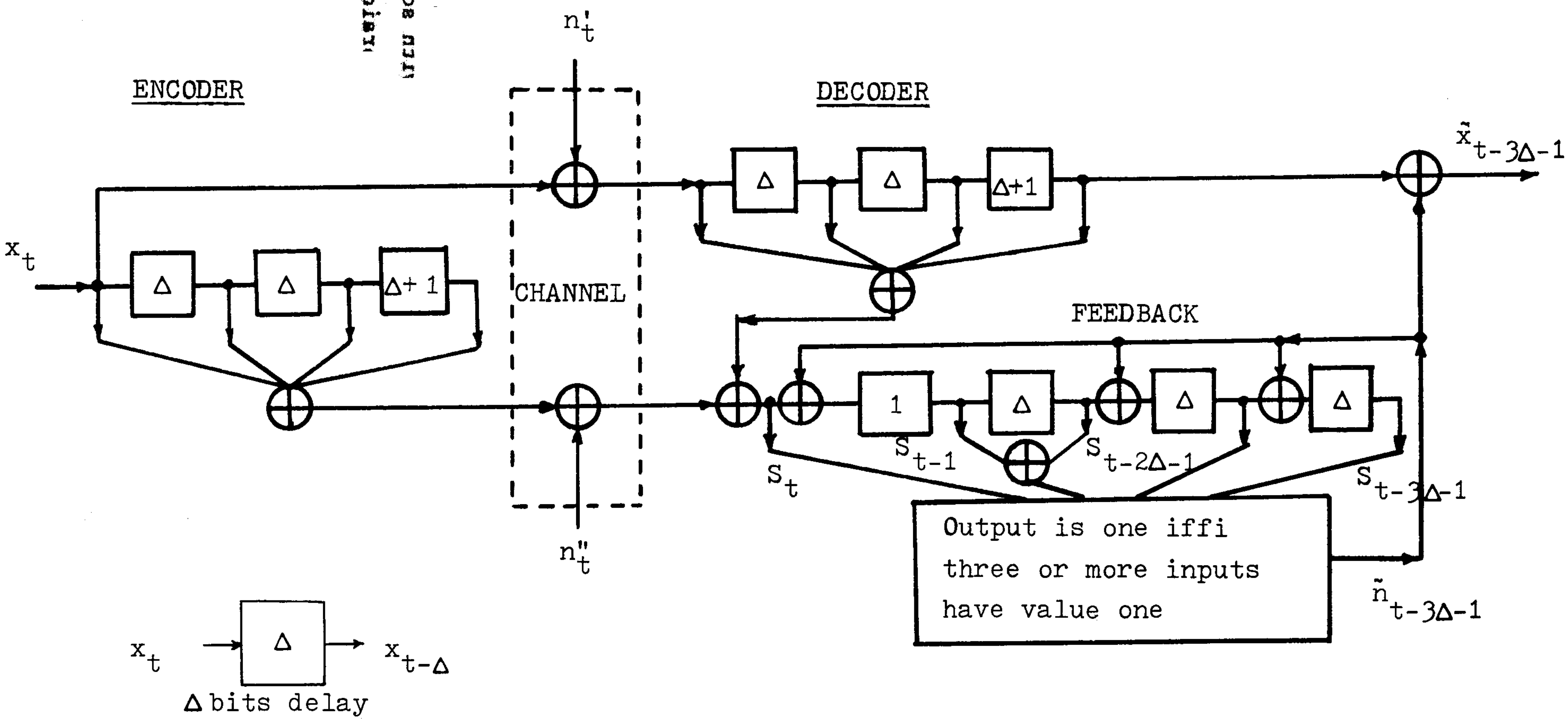


Fig. 5 - A hard-decision diffuse convolutional coding system.

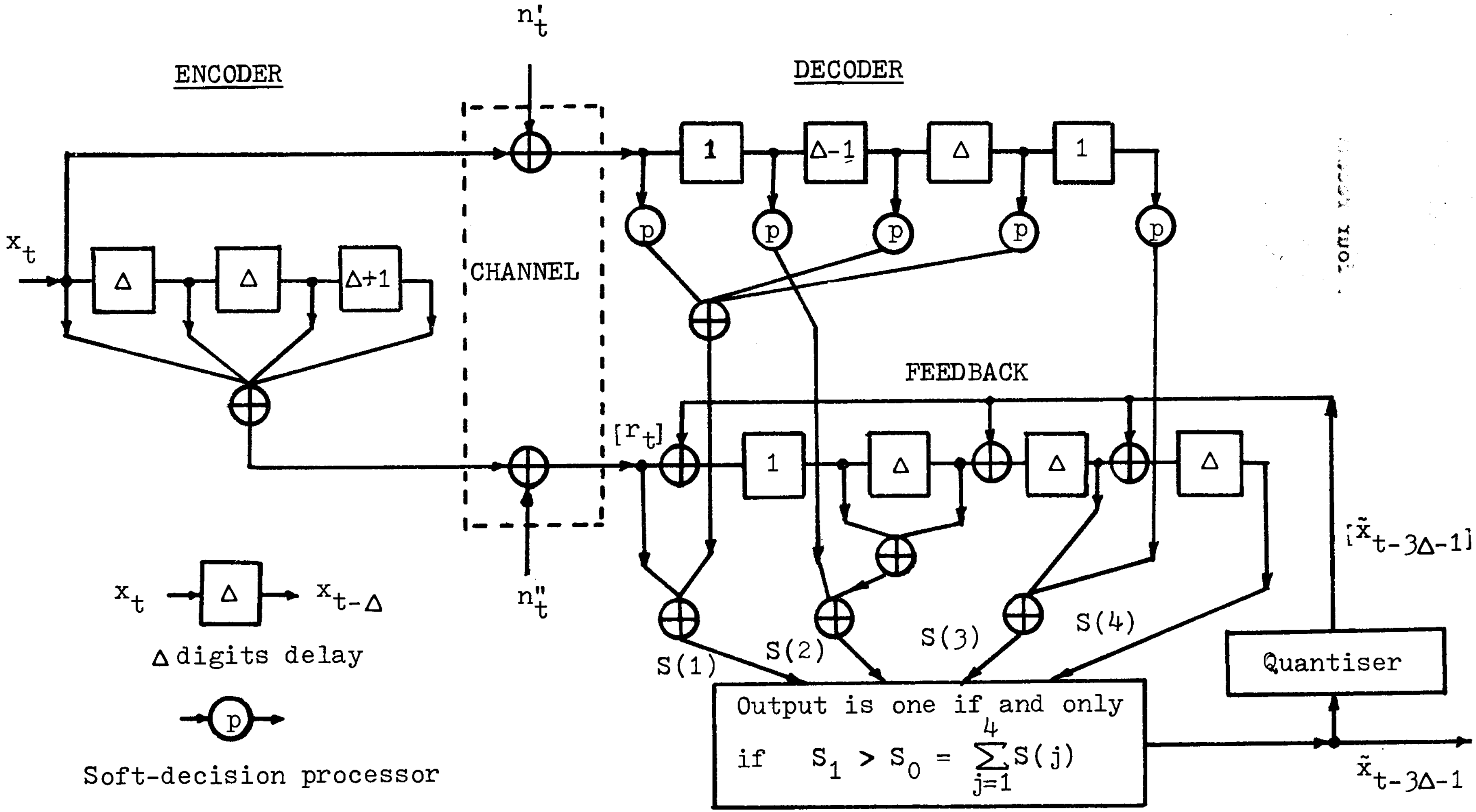


Fig. 6 - A soft-decision diffuse convolutional coding system.

(1) From the received digits $[x_t \oplus n'_t]$, $[x_{t-1} \oplus n'_{t-1}]$, $[x_{t-\Delta} \oplus n'_{t-\Delta}]$ and $[x_{t-2\Delta} \oplus n'_{t-2\Delta}]$, we estimate each corresponding information digit $\tilde{x}_t$, $\tilde{x}_{t-1}$, $\tilde{x}_{t-\Delta}$, and $\tilde{x}_{t-2\Delta}$, by taking the hard-decision estimate of each most significant bit.

(2) The value of S_o is found as follows. From the four received digit sums:

$$[r_t] = [x_t \oplus x_{t-\Delta} \oplus x_{t-2\Delta} \oplus x_{t-3\Delta-1} \oplus n''_t]$$

$$[r_{t-1}] = [x_{t-1} \oplus x_{t-3\Delta-1} \oplus n''_{t-1} \oplus n''_{t-\Delta-1}]$$

$$[r_{t-2\Delta-1}] = [x_{t-2\Delta-1} \oplus x_{t-3\Delta-1} \oplus x_{t-4\Delta-1} \oplus x_{t-5\Delta-2} \oplus n''_{t-2\Delta-1}]$$

$$[r_{t-3\Delta-1}] = [x_{t-3\Delta-1} \oplus x_{t-4\Delta-1} \oplus x_{t-5\Delta-1} \oplus x_{t-6\Delta-2} \oplus n''_{t-3\Delta-1}]$$

The previously estimated information digits $\tilde{x}_t$, $\tilde{x}_{t-1}$, $\tilde{x}_{t-\Delta}$, $\tilde{x}_{t-2\Delta}$;

and the previously decoded digits $\tilde{x}_{t-4\Delta-1}$, $\tilde{x}_{t-5\Delta-1}$, $\tilde{x}_{t-5\Delta-2}$, $\tilde{x}_{t-6\Delta-2}$: the following four noise digit sums can be derived.

$$S(1) = [r_t] \oplus [\tilde{x}_t \oplus \tilde{x}_{t-\Delta} \oplus \tilde{x}_{t-2\Delta}] = [\tilde{n}''_t]$$

$$S(2) = [r_{t-1}] \oplus [\tilde{x}_{t-1}] = [n''_{t-1} \oplus n''_{t-\Delta-1}]$$

$$S(3) = [r_{t-2\Delta-1}] \oplus [\tilde{x}_{t-2\Delta-1} \oplus \tilde{x}_{t-4\Delta-1} \oplus \tilde{x}_{t-5\Delta-2}] = [\tilde{n}''_{t-2\Delta-1}]$$

$$S(4) = [r_{t-3\Delta-1}] \oplus [\tilde{x}_{t-4\Delta-1} \oplus \tilde{x}_{t-5\Delta-1} \oplus \tilde{x}_{t-6\Delta-2}] = [\tilde{n}''_{t-3\Delta-1}]$$

Hence, $S_o = \Sigma\, S(j)$, for $j = 1$ to 4.

(3) The value of S_1 is given by $S_1 = \Sigma\, S(j) \oplus [111]$, for $j = 1$ to 4.

(4) We decode $\tilde{x}_{t-3\Delta-1} = 0$ if $S_o \leq S_1$, $x_{t-3\Delta-1} = 1$ if $S_o > S_1$.

5. Conclusions

In this paper we have introduced a new method for soft-decision majority threshold decoding of convolutional codes. The method consists of determining two check sum weights, and the decoding decision is based on the smaller of these two weights. The use of threshold decoding thus enables the advantage of increased coding gain to be realised without undue increase in complexity. Although two check sum weights have to be determined, as opposed to one with hard decision, the increase in decoding time is minimal as the two sums are obtained by merely inverting the estimated output digit.

6. References

1. Massey, J.L.: Threshold Decoding., M.I.T. Press, Cambridge, Mass., 1963.
2. Wozencraft, J.M., and Jacobs, I.M.: Principles of Communication Engineering., Wiley, New York, 1965.
3. Kohlenberg, A., and Forney, G.D. Jnr.: Convolutional coding for channels with memory, IEEE Trans., IT-14, 1968.