Maximum likelihood syndrome decoding of linear block codes

Eindhoven University of Technology
Netherlands

Department of Electrical Engineering

MAXIMUM LIKELIHOOD SYNDROME DECODING OF LINEAR BLOCK CODES

by

A.J. Vinck

TECHNISCHE HOGESCHOOL EINDHOVEN
NEDERLAND
AFDELING DER ELEKTROTECHNIEK
VAKGROEP INFORMATIE- EN COMMUNICATIE
THEORIE EI

EINDHOVEN UNIVERSITY OF TECHNOLOGY
THE NETHERLANDS
DEPARTMENT OF ELECTRICAL ENGINEERING
GROUP INFORMATION- AND COMMUNICATION
THEORY EI

MAXIMUM LIKELIHOOD SYNDROME DECODING OF LINEAR BLOCK CODES

by

A.J. Vinck.

April 1978

78 E 84
ISBN 90 6144 08 X

MAXIMUM LIKELIHOOD SYNDROME DECODING OF LINEAR BLOCK CODES.

ABSTRACT

We describe a method for maximum likelihood syndrome decoding of linear block codes, with hard- as well as with soft decisions. Also upperbounds are presented concerning the complexity of a syndrome decoder.

1. INTRODUCTION

In [1], Wolf introduces an algorithm for decoding linear block codes using channel measurement information. Wolf's decoder is based on the Viterbi algorithm. Viterbi like syndrome decoding [2,3,4] is an alternate to the classical Viterbi decoding algorithm for convolutional codes.

It will be shown that Viterbi like syndrome decoding can also be used to decode block codes with hard- as well as soft decisions (i.e. using channel measurement information). Before giving more details, we first review some basic principles of linear block codes.

Let G be a kxn-matrix with rank k. Then G can be used as a generator matrix for an (n,k) block code. Given the message k-vector $\underline{m}$, the codeword n-vector $\underline{c}$ is given by

$$\underline{c} = \underline{m}\, G \ . \qquad (1$$

Let H be the generator of the dual code, i.e. $G\, H^T = \underline{0}$. Then an n-vector $\underline{c}$ is a codeword if

$$\underline{c}\, H^T = \underline{0} \ . \qquad (2$$

The matrix H^T is refered to as the parity check matrix. An equivalent generator matrix G' can be obtained from G by elementary row operations and column permutations. We are interested in the systematic generator G' ,

$$G' = \begin{bmatrix} 1 & 0 & \cdots & 0 & p_{1,1} & p_{1,2} & \cdots & p_{1,n-k} \\ 0 & 1 & \cdots & 0 & p_{2,1} & p_{2,2} & \cdots & p_{2,n-k} \\ \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & \cdots & 1 & p_{k,1} & p_{k,2} & \cdots & p_{k,n-k} \end{bmatrix} \tag{3}$$

and the associated parity check matrix, H'^T,

$$H'^T = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n-k} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n-k} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ p_{k,1} & p_{k,2} & \cdots & p_{k,n-k} \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & 1 \end{bmatrix} . \tag{4}$$

Now, suppose we transmit a codevector $\underline{c}$. The received vector $\underline{r} = \underline{c} + \underline{e}$, where $\underline{e}$ is an n-vector representing the noise by two-level quantization of the output.

Now we form the (n-k)-syndromevector $\underline{s}$, where

$$\underline{s} = (\underline{c} + \underline{e})H^T = \underline{e}\, H^T . \tag{5}$$

The task of the decoder is to determine the most likely noisevector that caused the syndromevector $\underline{s}$. This is the subject of the following section.

II. SYNDROME FORMER AND DECODING ALGORITHM

To form the syndromes, mentioned in the previous section, we reorganize the incoming data stream by putting (n-k-1) all zero n-vectors between two successive received vectors. The syndromes can then be calculated with the circuit of Fig. 1.

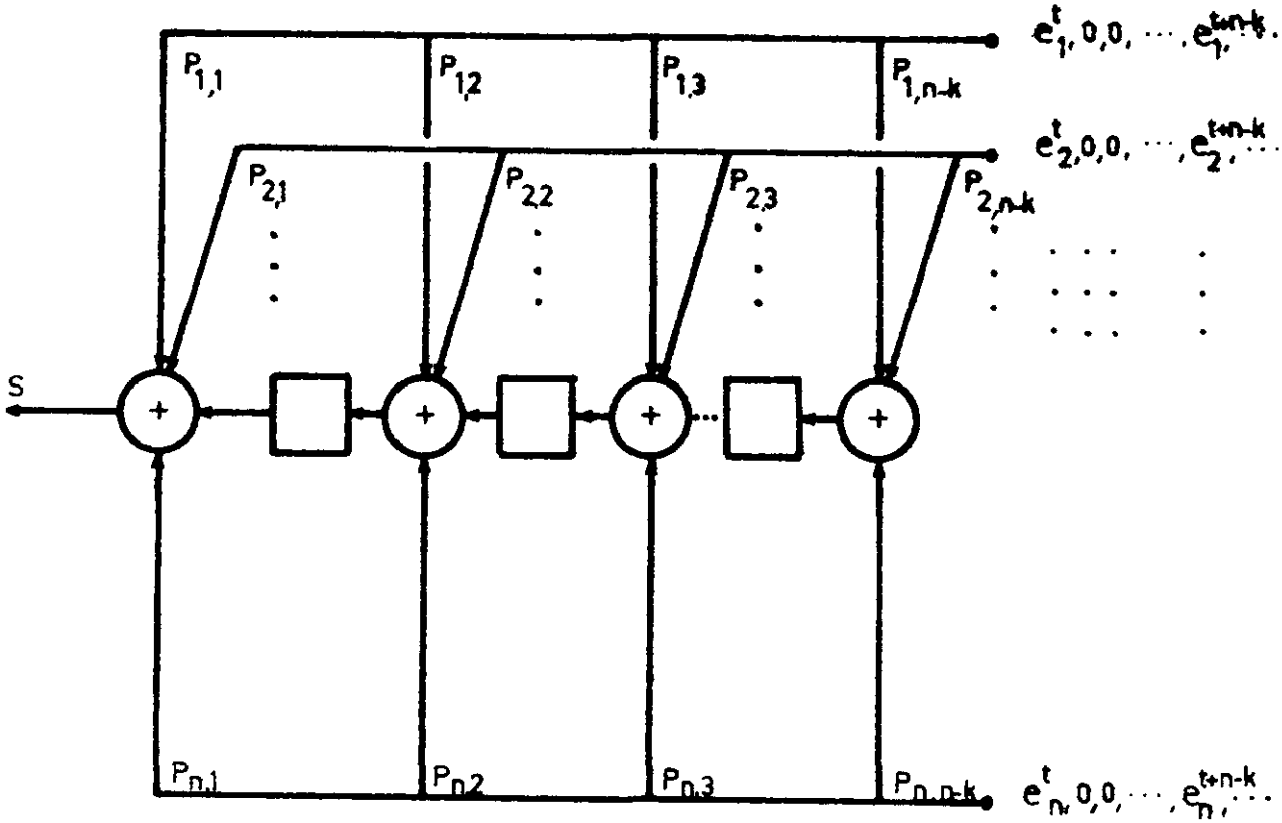

Fig. 1. Adjoint obvious realization of H^T .

This realization is called the adjoint obvious realization [3]. As the syndrome s only depends on the noise vectors, we can omit the code-vector contribution.

The syndrome former of Fig. 1 has 2^{n-k-1} states. Before entering an

$\underline{e}$-vector into the circuit it is in the all zero state, and after (n-k-1) shifts it has returned to the all zero state. It is possible that two or more noisevectors give rise to the same syndrome sequence. A maximum likelihood decoder is to determine the noise vector $\underline{\hat{e}}$ of minimum Hamming weight that may be a cause of the observed syndrome sequence. As the operation of the decoder can be described in terms of the syndrome former state space, we now introduce some convenient [4] notations. States are denoted by lower case greek letters with a subscript, e.g.

$$\tau_1 \triangleq [s_1, s_2, \ldots, s_{n-k-1}] \quad \text{, and its left shifts}$$
$$\tau_2 \triangleq [s_2, s_3, \ldots, s_{n-k-1}, 0] \quad \text{, and so on.}$$

The states generated by the system are defined as

$$\alpha_1 \triangleq [p_{1,2}, p_{1,3}, \ldots, p_{1,n-k-1}] \quad ,$$
$$\alpha_2 \triangleq [p_{2,2}, p_{2,3}, \ldots, p_{2,n-k-1}] \quad , \ \ldots \ ,$$
$$\alpha_n \triangleq [p_{n,2}, p_{n,3}, \ldots, p_{n,n-k-1}] \quad .$$

The syndrome digit s, and the new state τ_1, see Fig. 1, are completely determined by the present state τ_1, and the noise vector $[e_1, e_2, \ldots, e_n]$.

$$\tau_1 \xrightarrow{[e_1, e_2, \ldots, e_n]} \begin{cases} \tau_1 = \tau_2 + e_1\alpha_1 + e_2\alpha_2 + \ldots + e_n\alpha_n \\ s = s_1 + e_1p_{1,1} + e_2p_{2,1} + \ldots + e_np_{n,1} \end{cases}$$

Note that only on time $t = t_0 + i(n-k)$, $i=0,1,2,\ldots$, there is a possible nonzero input vector $\underline{e}$.

As an example take the (7,4) Hamming code with parity check matrix

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} .$$

Fig. 2 gives the syndrome former, and Fig. 3 the corresponding state

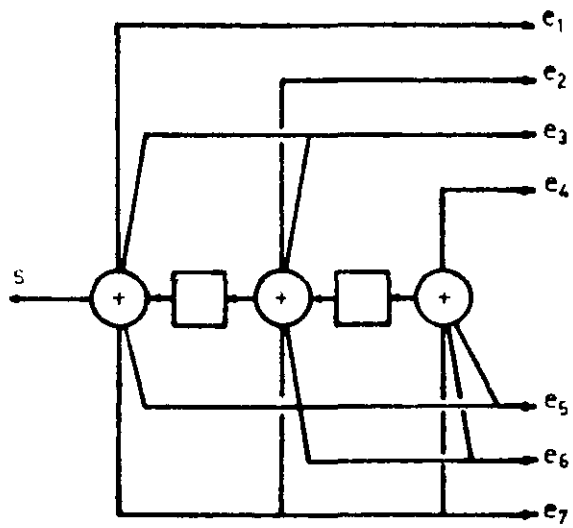

Fig. 2. Syndrome former for binary (7,4) Hamming code.

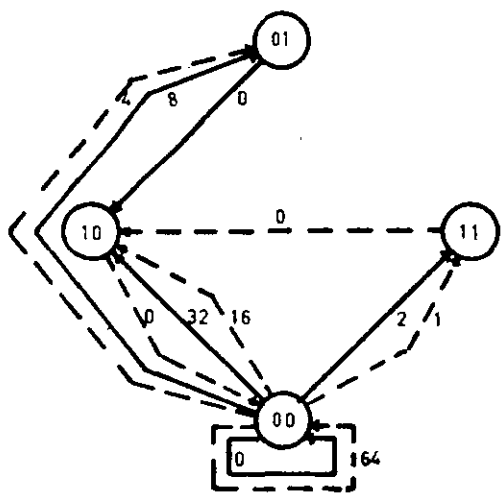

Fig. 3. State diagram of the syndrome former of Fig. 2.

diagram. Solid lines correspond to a syndrome digit s=0, and dashed lines to a syndrome digit s=1. The decimal values indicated along the edges represent the noise vector, to be interpreted as a binary number. Note that noise vectors of weight greater than one have been omitted in the state diagram. These noise vectors correspond with incorrectable error patterns !

Edges emerging from any state except for the all zero state have Hamming weight zero. This is a consequence of the construction of our input sequence, i.e. any noise vector being followed by (n-k-1) all zero vectors. The decoder has to determine the state sequence that corresponds to a noise vector estimate of minimum Hamming weight that may be a possible cause of the syndrome sequence. Therefore, we associate with each state a metric- and a pathregister. With each edge pointing to a state, we associate an edge metric. For hard decisions, the edge metric is the Hamming weight of the noise vector indicated along the edge. Normally, for independent reception, one would take the loglikelihood function of the noise vector. The metricregister contains the minimum weight of a path to this particular state. The pathregister contains the corresponding survivor noise vector. For each syndrome digit, calculate the survivor for each possible state, and update the metric- and pathregisters. This in done (n-k) times, starting in state zero and ending in state zero. States that have no paths leading to the all zero state within (n-k-1) steps and along zero noise vector edges can be deleted. Note that it is possible to only have one state remaining before completing (n-k-1) steps in the algorithm. In this case we are able to make an early decision. The algorithm above can also be used with real valued metrics, i.e. soft decisions, as will

be discussed in section V.

Another way of implementing the syndrome former is suggested by (4). If the parity check matrix is in the form given in (4), we can reorganize the incomming data stream as follows. The first (k+1) digits are offered to the syndrome former realized according to the matrix H^T. The last (n-k-1) digits are buffered, and each step one digit of this buffer is offered to the syndrome former together with k zero's. Where the zero's enter the syndrome former on the top k entries. The state space again has dimension (n-k-1). The syndrome digit s, and the new state are now determined according to

$$\tau_1 \xrightarrow{[e_1, e_2, \ldots, e_k]} \begin{cases} \tau_1 = \tau_2 + e_1\alpha_1 + e_2\alpha_2 + \ldots + e_k\alpha_k \\ s = s_1 + e_1 h_{1,1} + e_2 h_{2,1} + \ldots + \\ \qquad + e_k h_{k,1} + e_{k+1} \end{cases}$$

in the first step, and according to

$$\tau_1 \longrightarrow \begin{cases} \tau_1 = \tau_2 \\ s = s_1 + e_{k+i} \end{cases}$$

each next step i , $2 \leq i \leq n-k-1$.

Note that for this realization there are only 2^k possible states in the first step. The decoding algorithm is similar to the one previous described.

III. DECODING OF CYCLIC CODES

The generator matrix of an (n,k) cyclic code consists of k rows that are right shifts of the top row. The parity matrix H is an (n-k)xn matrix with the same property. As with convolutional codes, we can use a shift register to form the codewords. Fig. 4 gives a possible implementation.

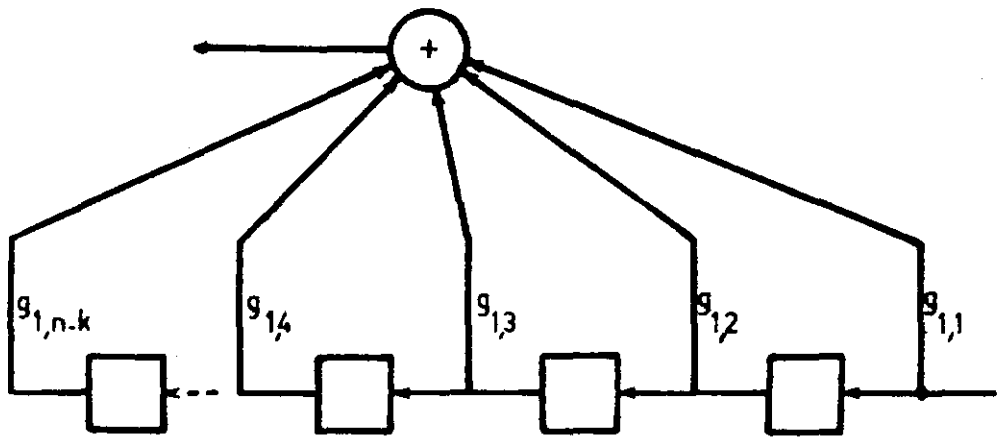

Fig. 4. Implementation of a cyclic encoder.

The k message digits are followed by (n-k) zero's. The connections correspond to the elements of the first row of G. To see how the syndrome forming procedure works, we proceed with a (7,3) cyclic code as an example. Let

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}, \quad \text{and } G = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

For this code, the syndrome former can be implemented as in Fig. 5.

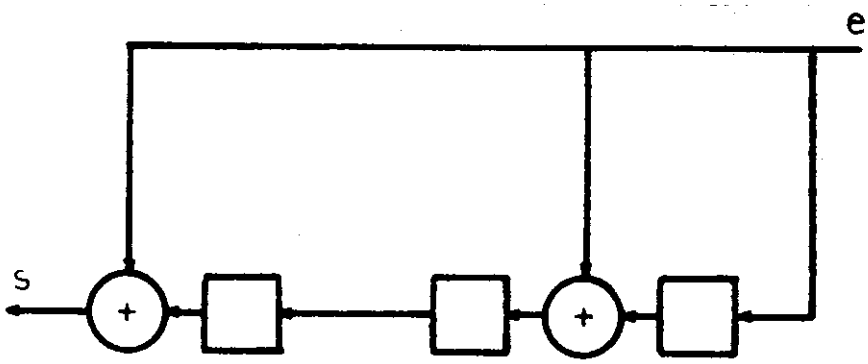

Fig. 5. Syndrome former for a (7,3) cyclic code.

For each received block of n digits, fill the stages of the shift-register with the first (k-1) digits. Then, as $k < n-k$, the contribution to the syndrome output of the codeword is equal to zero, and this is also true for the next (n-k-1) left shifts. Hence, the (n-k) syndrome digits corresponding with the received word are completely determined by the noise contribution to the codeword. The codeword can thus be deleted in the decoding and syndrome forming procedure. The above syndrome forming method can always be used for $k \leq (n-k)$. Like Wolf [1], we can construct a trellis to be used in the decoding procedure.

Note that the syndrome sequence can also be calculated [6] by deviding the received data stream, considered as a polynomial, by the generator. This procedure leads to an alternative trellis decoder.

IV. COMPLEXITY OF THE SYNDROME DECODER

The major functions to be performed by the decoder are

1) the computation of the new state metrics, and

2) the determination of the survivor sequence.

For each state in the trellis, or state diagram, one has to perform these functions. Hence, reducing the number of states is equivalent to reducing the complexity of the decoder. Suppose we have chosen for the implementation of Fig. 2. With 2^n possible noise inputs. we can go to atmost 2^{n-k-1} different states. We will now show that this number of states can be reduced, depending on the error correcting capabilities of the code.

Let d_{min} be defined as the minimum Hamming distance between any two codewords. Then, if we want to correct $d_{min/2}$ or fewer channel errors, the decoder must distinguish between no more then

$$M_{d_{min}} = \sum_{i=0}^{d_{min}/2} \binom{n}{i} \tag{6}$$

error patterns in the first step. However, we have 2^{n-k-1} different states at our disposal. Hence, a decodable error pattern must be included in the first transition to the 2^{n-k-1} states. We can reduce our decoder if

$$M_{d_{min}} < 2^{n-k-1} \quad . \tag{7}$$

For large values of n, this means that reduction is possible if

$$H\left(\frac{d_{min}}{2n}\right) < 1 - \frac{k}{n} \quad . \tag{8}$$

The Hamming bound however tells us that there are no codes for which $H\left(\frac{d_{min}}{2n}\right) > 1 - \frac{k}{n}$. Hence, in almost all practical cases, we can reduce the complexity of the decoder.

For values of $k < n-k$, we are able to derive a simular result. Here the maximum number of different paths in the first step is equal to 2^k. If we want to correct a maximum of $d_{min/2}$ channel errors, reduction is possible if

$$2^k > \sum_{i=0}^{d_{min/2}} \binom{k}{i} \quad , \tag{9}$$

or equivalently, if

$$\frac{d_{min}}{2k} < \frac{1}{2} \tag{1}$$

or

$$\frac{d_{min}}{2n} < \frac{1}{2}\,\frac{k}{n} \quad ,$$

which is an analogue to the Plotkin upper bound.

Note that the above decoders are no longer ML decoders.

V. QUANTIZATION OF THE RECEIVED CODE SYMBOLS

As pointed out in [7], 180° binary phase shift keying (BPSK) in combination with coding is an efficient way of communication over Gaussian channels. Quantization of the demodulated received code symbols, facilitates digital processing at the decoder. When 8-level quantization is used, about 0.25 dB in received signal to noise ratio is lost, compared with infinitely fine quantization. Hence, further quantization is questionable. With 2-level (binary) quantization the loss in SNR is roughly 2 dB. Fig. 6 shows the quantization schemes for 2, 4 and 8 levels, where +1

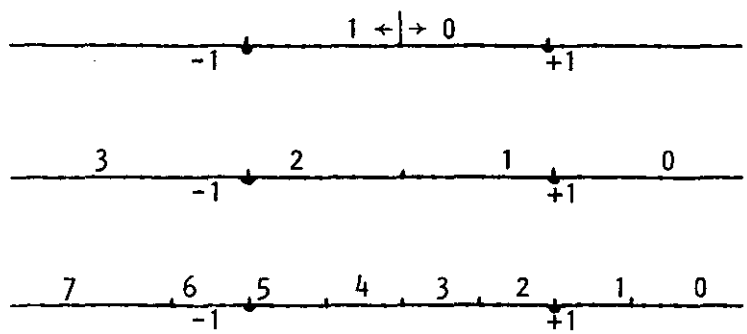

Fig. 6. Quantization scheme for 2, 4 and 8 levels.

corresponds with a code symbol 1, and -1 with a code symbol 0. The spacing in the above schemes can be shown to be almost optimum. The Gaussian channel with modulator and demodulator is then equivalent to a discrete channel with two inputs, and 2, 4 or 8 outputs, respectively. The channel transition probabilities are equal to the probabilities that a Gaussian random variable with variance $\sqrt{\frac{N_0}{2E}}$ and mean $\pm$ 1 lies in the intervals indicated in Fig. 6. The problem we are now faced with is the adjustment of the syndrome decoder, Take, for example,

a 4-level quantizer as indicated in Fig. 7.

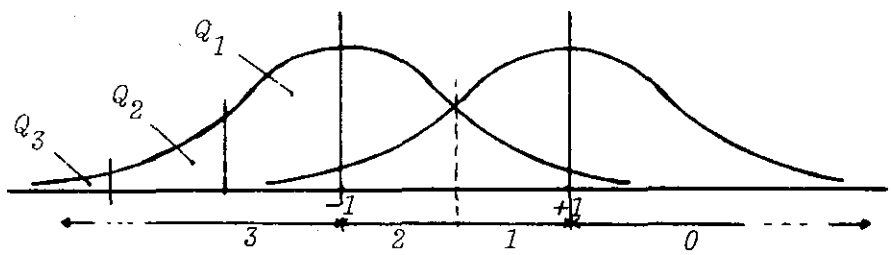

Fig. 7. Probability density function of the received signal.

Let a received signal lie in interval 2. The syndrome forming circuit only accepts the symbols 0 and 1. Hence, a binary quantizer is used to set the received signal equal to 0. Now there are two possibilities, the relevant noise bit could either be zero or one with probability $Pr(0) = Q_1$ and $Pr(1) = Q_2$, respectively. The same can be said about a received signal lying in interval 1. For the intervals 0 and 3, $Pr(0) = \frac{1}{2}$ and $Pr(1) = Q_3$. In fact, we only need the absolute value of the received signal to determine Pr (0) and Pr(1) and thus the egde metric. From simulations, it follows that the decoder is quite insensitive to egde metric quantization. Hence, use of integers instead of exact loglikelihoods gives a very small performance degradation. Fig. 8 shows a possible set of metrics for the case of 4-level quantization.

Hard quantized noise	Received quantized level			
	0	1	2	3
0	0	1	1	0
1	3	2	2	3

Fig. 8. Metric quantization scheme.

The decoder now looks for a path that minimizes the metric per state, and selects the path with minimum overall metric.

VI. CONCLUSION

A method, using the syndromes generated by the noise on the channel, is given to decode efficiently linear block codes. This method can be used with channel measurement information. Also, possibilities to reduce the complexity of the decoder are indicated. It is our feeling that this technique can also be used in syndrome decoding of convolutional codes, leading to a reduced state syndrome decoder. For the Viterbi decoder, this was tried without success, at least up to now. However, considering the noise in the state diagram is a more natural way of looking at this problem.

REFERENCES

[1] J. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis", IEEE Trans. Inform. Theory, vol. IT-24, pp. 76-80, January 1977.

[2] J.P.M. Schalkwijk, and A.J. Vinck, "Syndrome decoding of convolutional codes", IEEE Trans. on Communications, vol. COM-23, pp. 789-792, July 1975.

[3] J.P.M. Schalkwijk, and A.J. Vinck, "Syndrome decoding of binary rate-$\frac{1}{2}$ convolutional codes", IEEE Trans. on Communications, vol. COM-24, pp. 977-985, September 1976.

[4] J.P.M. Schalkwijk, A.J. Vinck, and K.A. Post, "Syndrome decoding of binary rate k/n convolutional codes", IEEE Trans. in Inform. Theory, to appear.

[5] G.D. Forney, Jr., "Convolutional codes I : Algebraic structure", IEEE Trans. Inform. Theory, vol. IT-16, pp. 720-738, November 1970.

[6] Peterson and Weldon, "Error correcting codes", M.I.T. Press 1972.

[7] J.A. Heller, and I.M. Jacobs, "Viterbi decoding for satellite and space communication", IEEE Trans. on Comm. Techn., vol. COM-18, pp. 835-848, October 1971.