9-1975

An Optimum Symbol-by Symbol decoding rule for linear codes

Carlos R.P. Hartmann
Syracuse University, chartman@syr.edu

Luther D. Rudolph
Syracuse University

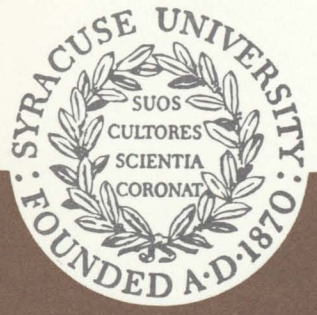
3-75

AN OPTIMUM SYMBOL-BY-SYMBOL DECODING RULE FOR LINEAR CODES

C. R. P. Hartmann

L. D. Rudolph

September, 1975



SYSTEMS AND INFORMATION SCIENCE
SYRACUSE UNIVERSITY

AN OPTIMUM SYMBOL-BY-SYMBOL DECODING RULE FOR LINEAR CODES

C. R. P. Hartmann

L. D. Rudolph

This work was supported by National Science Foundation Grant ENG 75-07709 and Rome Air Development Center Contract F30602-72-C-0360.

Systems and Information Science

Syracuse University

Syracuse, New York 13210

(315) 423-2368

ABSTRACT

A decoding rule is presented which minimizes the probability of symbol error over a time-discrete memoryless channel for any linear error-correcting code when the code words are equiprobable. The complexity of this rule varies inversely with code rate, making the technique particularly attractive for high rate codes. Examples are given for both block and convolutional codes.

ACKNOWLEDGEMENT

The authors wish to acknowledge the contribution to this work of Aileen McLoughlin, who provided a key to the discovery of the main result of this paper, and to Professors Kishan Mehrotra and Harry Schwarzlander for a number of helpful discussions on probability theory.

I. INTRODUCTION

In recent years there has been a growing interest in "soft decision" decoding schemes for error-correcting codes. The intent is to avoid, in part or in whole, the degradation of communication system performance which results when symbol-by-symbol "hard decision" quantization precedes decoding. The two best known techniques, which are optimum in the sense that they minimize the probability of word error for any time-discrete memoryless channel when the code words are equiprobable, are correlation decoding of block codes and Viterbi decoding of trellis codes [1]. Although in practice correlation and Viterbi decoding are usually used in conjunction with linear codes, neither technique makes any essential use of the linear property. Both techniques are exhaustive in that the received word is compared with every word in the code. For this reason, these techniques may be used only with codes having a small number of code words, i.e. low rate codes or middle-to-high rate codes with short block or constraint lengths.

In this paper we present a new decoding rule which is, in a way, the dual of correlation/Viterbi decoding in the case of linear codes. This rule is also optimum, but in the sense that it minimizes the probability of symbol error for any time-discrete memoryless channel when the code words are equiprobable, and makes essential use of the linear property.

It is also exhaustive, but in the sense that every word in the dual code is used in the decoding process. This means that in practice this decoding rule can be used only with codes whose dual code has a small number of code words, i.e. high rate codes or low-to-middle rate codes with short block or constraint lengths.

In Section II, we present the decoding rule and prove that it is optimum. Although perhaps not immediately obvious from the concise treatment given there, the decoding rule is a form of threshold decoding [2]. This is easily seen from the examples in Section III where the actual form of the decoding rule in the binary case is illustrated for both block and convolutional codes. Section IV contains a discussion of results.

II. THE DECODING RULE

For convenience, we present the decoding rule for linear block codes. The extension to convolutional codes is immediate and will be obvious from the examples in Section III.

Let $\underset{\sim}{c} = (c_0, c_1, \ldots, c_{n-1})$ denote any code word of an (n,k) linear block code C over GF(p) and $\underset{\sim}{c}'_j = (c'_{j0}, c'_{j1}, \ldots, c'_{j,n-1})$ the jth code word of the (n,n-k) dual code C'. A code word $\underset{\sim}{c}$ is transmitted over a time-discrete memoryless channel with output alphabet B. The received word is denoted by $\underset{\sim}{r} = (r_0, r_1, \ldots, r_{n-1})$, $r_j \in B$. The decoding problem is: given $\underset{\sim}{r}$, compute an estimate $\hat{c}_m$ of the transmitted code symbol c_m in such a way that the probability that $\hat{c}_m = c_m$ is maximized. Other notation: $\omega \equiv \exp[2\pi\sqrt{-1}/p]$ (primitive complex pth root of unity); $\delta_{ij} = 1$ if $i = j$ and 0 otherwise; Pr(x) is the probability of x and Pr(x|y) is the probability of x given y. Unless otherwise stated, the elements of GF(p) are taken to be the integers 0,1,...,p-1 and all arithmetic operations are performed in the field of complex numbers.

DECODING RULE:

Set $\hat{c}_m = s$, where $s \in GF(p)$ maximizes the expression

$$A_m(s) = \sum_{t=0}^{p-1} \omega^{-st} \sum_{j=1}^{p^{n-k}} \left[\prod_{\ell=0}^{n-1} \sum_{i=0}^{p-1} \omega^{i(c'_{j\ell} + t\delta_{m\ell})} \Pr(r_\ell | i) \right] . \quad (1)$$

Theorem: Decoding Rule (1) maximizes the probability that $\hat{c}_m = c_m$.

(Proof) We must show that choosing s to maximize $A_m(s)$ is equivalent to maximizing the probability that $c_m = s$ given the received word $\underset{\sim}{r}$. We do this directly by showing that $A_m(s) = \lambda \Pr(c_m = S|\underset{\sim}{r})$, where λ is a positive constant which is independent of s. We first note that the expression in the brackets on the RHS of (1), which is in product-of-sums form, can be rewritten in the sum-of-products form

$$\sum_{v_0=0}^{p-1} \sum_{v_1=0}^{p-1} \cdots \sum_{v_{n-1}=0}^{p-1} \prod_{\ell=0}^{n-1} \omega^{v_\ell (c'_{j\ell} + t\delta_{m\ell})} \Pr(r_\ell | v_\ell) \tag{2}$$

where $\underset{\sim}{v} = (v_0, v_1, \ldots, v_{n-1})$ is any element of V_n, the vector space of all n-tuples over GF(p). Expressing (2) in vector notation and substituting in (1) yields

$$A_m(s) = \sum_{t=0}^{p-1} \omega^{-st} \sum_{j=1}^{p^{n-k}} \left[\sum_{\underset{\sim}{v} \in V_n} \omega^{\underset{\sim}{v} \cdot \underset{\sim}{c}'_j} \omega^{t(\underset{\sim}{v} \cdot \underset{\sim}{e}_m)} \Pr(\underset{\sim}{r}|\underset{\sim}{v}) \right]$$

$$= \sum_{t=0}^{p-1} \omega^{-st} \sum_{\underset{\sim}{v} \in V_n} \omega^{t(\underset{\sim}{v} \cdot \underset{\sim}{e}_m)} \Pr(\underset{\sim}{r}|\underset{\sim}{v}) \sum_{j=1}^{p^{n-k}} \omega^{\underset{\sim}{v} \cdot \underset{\sim}{c}'_j} , \tag{3}$$

where $\underset{\sim}{e}_m = (\delta_{m0}, \delta_{m1}, \ldots, \delta_{m,n-1})$ is the vector with 1 in the m^{th} position and 0 elsewhere. By the orthogonality properties of group characters [3,4] we know that

$$\sum_{j=1}^{p^{n-k}} \omega^{\underset{\sim}{v} \cdot \underset{\sim}{c}'_j} = \begin{cases} p^{n-k} \text{ if } \underset{\sim}{v} \in C \\ 0 \text{ otherwise} \end{cases} . \tag{4}$$

Applying (4) to (3) gives

$$A_m(s) = p^{n-k} \sum_{t=0}^{p-1} \omega^{-st} \sum_{\underset{\sim}{c}\in C} \omega^{t(\underset{\sim}{c}\cdot\underset{\sim}{e}_m)} \Pr(\underset{\sim}{r}|\underset{\sim}{c})$$

$$= p^{n-k} \sum_{\underset{\sim}{c}\in C} \Pr(\underset{\sim}{r}|\underset{\sim}{c}) \sum_{t=0}^{p-1} \omega^{t(\underset{\sim}{c}\cdot\underset{\sim}{e}_m - s)} \quad . \qquad (5)$$

But the sum on the far RHS of (5) vanishes unless $\underset{\sim}{c}\cdot\underset{\sim}{e}_m - s = 0$. Hence,

$$A_m(s) = p^{n-k+1} \sum_{\underset{\sim}{c}\in C, c_m = s} \Pr(\underset{\sim}{r}|\underset{\sim}{c})$$

$$= p^{n-k+1} \sum_{\underset{\sim}{c}\in C, c_m = s} \Pr(\underset{\sim}{c}|\underset{\sim}{r})\,(\Pr(\underset{\sim}{r})/\Pr(\underset{\sim}{c})) \quad . \qquad (6)$$

Finally, since the code words of C are equiprobable, $\Pr(\underset{\sim}{c}) = p^{-k}$ and (6) becomes

$$A_m(s) = p^{n+1} \Pr(\underset{\sim}{r}) \sum_{\underset{\sim}{c}\in C, c_m = s} \Pr(\underset{\sim}{c}|\underset{\sim}{r})$$

$$= p^{n+1} \Pr(\underset{\sim}{r}) \Pr(c_m = s|\underset{\sim}{r}) \quad . \qquad \text{Q.E.D.}$$

As one might expect, the decoding rule takes a simple form in the binary case: set $\hat{c}_m = 0$ if $A_m(0) > A_m(1)$ and $\hat{c}_m = 1$ otherwise. It is more convenient however to state the rule in terms of the <u>likelihood ratio</u> $\phi_m = \Pr(r_m|1)/\Pr(r_m|0)$.

Substituting the RHS of (1) into the inequality $A_m(0) > A_m(1)$ yields

$$\sum_{t=0}^{1} \sum_{j=1}^{2^{n-k}} \prod_{\ell=0}^{n-1} \sum_{i=0}^{1} (-1)^{i(c'_{j\ell}+t\delta_{m\ell})} \Pr(r_\ell|i) >$$

$$\sum_{t=0}^{1} (-1)^t \sum_{j=1}^{2^{n-k}} \prod_{\ell=0}^{n-1} \sum_{i=0}^{1} (-1)^{i(c'_{j\ell}+t\delta_{m\ell})} \Pr(r_\ell|i) ,$$

or

$$\sum_{j=1}^{2^{n-k}} \prod_{\ell=0}^{n-1} \left[\Pr(r_\ell|0)+(-1)^{c'_{j\ell}+\delta_{m\ell}} \Pr(r_\ell|1) \right] > 0 . \tag{7}$$

Dividing both sides of (7) by $\prod_{\ell=0}^{n-1} \Pr(r_\ell|0)$ and using the definition of the likelihood ratio, we have

$$\sum_{j=1}^{2^{n-k}} \prod_{\ell=0}^{n-1} \left[1+\phi_\ell(-1)^{c'_{j\ell}+\delta_{m\ell}} \right] > 0 . \tag{8}$$

Then dividing both sides of (8) by the positive quantity

$$\prod_{\ell=0}^{n-1} \left[1+\phi_\ell \right] ,$$

$$\sum_{j=1}^{2^{n-k}} \prod_{\ell=0}^{n-1} \frac{1+\phi_\ell(-1)^{c'_{j\ell}+\delta_{m\ell}}}{1+\phi_\ell} > 0 .$$

Finally, using the identity

$$\frac{1+\phi_\ell(-1)^{c'_{j\ell}+\delta_{m\ell}}}{1+\phi_\ell} = \left(\frac{1-\phi_\ell}{1+\phi_\ell} \right)^{c'_{j\ell}\oplus\delta_{m\ell}}$$

where ' $\oplus$ ' denotes modulo 2 addition, we obtain the

BINARY DECODING RULE:

$$\underline{\text{Set } \hat{c}_m = 0 \text{ if}}$$

$$\sum_{j=1}^{2^{n-k}} \prod_{\ell=0}^{n-1} \left(\frac{1-\phi_\ell}{1+\phi_\ell} \right)^{c'_{j\ell} \oplus \delta_{m\ell}} > 0 \qquad (9)$$

$$\underline{\text{and } \hat{c}_m = 1 \text{ otherwise.}}$$

We remark that up to this point we have ignored the question of how one retrieves the decoded information symbols from the code word estimate $\hat{\underset{\sim}{c}}$. This could be a problem because, when a symbol-by-symbol decoding rule is used, $\hat{\underset{\sim}{c}}$ is not in general a code word. In the case of block codes, we could insist that the code be systematic without loss of generality, but there might be some objection to this restriction in the case of convolutional codes. As it turns out, this is not a problem since our decoding rule is easily modified to produce estimates of the information symbols directly if need be. Simply note that every information symbol a_m can be expressed as a linear combination, over GF(p), of code word symbols c_m, i.e. $a_m = \sum_\ell b_{m\ell} c_\ell$, $b_{m\ell} \in$ GF(p), and that the proof of the theorem goes through intact if we substitute $\widehat{\sum_\ell b_{m\ell} c_\ell}$ for $\hat{c}_m$ and $b_{m\ell}$ for $\delta_{m\ell}$ in (1).

III. EXAMPLES

(a) (7,4) Hamming code

We will illustrate the decoding rule for the received symbol r_0. Since the (7,4) code is cyclic, $r_1,\ldots,r_6$ are decoded simply by cyclically permuting the received word $\underline{r}$ in the buffer store.

Binary Decoding Rule (9) in this case becomes

$$\hat{c}_0 = 0 \text{ iff } \sum_{j=1}^{8} \prod_{\ell=0}^{6} \left(\frac{1-\phi_\ell}{1+\phi_\ell} \right)^{c'_{j\ell} \oplus \delta_{0\ell}} > 0 \ . \tag{10}$$

The parity check matrix H of the (7,4) code and its row space C' are shown below.

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{matrix} (a) \\ (b) \\ (c) \end{matrix}$$

$$C': \quad \begin{array}{ccccccc} c_0 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & (a) \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & (b) \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & (a \oplus b) \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & (c) \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & (a \oplus c) \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & (b \oplus c) \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & (a \oplus b \oplus c) \end{array} \tag{11}$$

Let $\rho_\ell = (1-\phi_\ell)/(1+\phi_\ell)$. Then substituting (11) into (10) gives

$$\hat{c}_0 = 0 \text{ iff } \rho_0 + \rho_1\rho_2\rho_4 + \rho_2\rho_5\rho_6 + \rho_1\rho_3\rho_6 + \rho_3\rho_4\rho_5 + \rho_0\rho_1\rho_2\rho_3\rho_5 + \rho_0\rho_2\rho_3\rho_4\rho_6 + \rho_0\rho_1\rho_4\rho_5\rho_6 > 0 \ . \tag{12}$$

The decoder configuration corresponding to (12) is shown in Figure 1.

The knowledgable reader will immediately recognize the similarity between the decoder of Figure 1 and a one-step majority decoder using nonorthogonal parity checks [5]. And in fact if the "soft decision" function $(1-\phi(x)/(1+\phi(x))$ were replaced by the "hard decision" function $f(x) = -1$ if $x > \frac{1}{2}$ and $+1$ otherwise, and the last three parity checks in the decoder were deleted, the resulting circuit would be mathematically equivalent to a conventional one-step majority decoder. Parity checks in the circuit of Figure 1 would be computed by taking products of +1's and -1's, rather than by taking modulo 2 sums of 0's and 1's as would be the case in a conventional digital decoding circuit.

(b) (4,3,3) convolutional code

We now illustrate the decoding rule for the received symbol r_0 using an $(n_0,k_0,m) = (4,3,3)$ convolutional code (from Peterson and Weldon [6], page 395).

Binary Decoding Rule (9) in this case becomes

$$\hat{c}_0 = 0 \text{ iff } \sum_{j=1}^{\infty} \prod_{\ell=0}^{\infty} \left(\frac{1-\phi_\ell}{1+\phi_\ell} \right)^{c'_{j\ell} \oplus \delta_{0\ell}} > 0 \ . \tag{13}$$

Of course, there are only a finite number of nonzero terms in (13), the number depending upon the length of the transmitted code sequence. The initial portions of the parity check matrix H of the (4,3,3) code and its row space C' are shown below.

$$
H = \left|\begin{array}{lll}
1\ 1\ 1\ 1\ 0\ \ldots & & \text{(a)} \\
1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ \ldots & & \text{(b)} \\
1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ \ldots & & \text{(c)} \\
\vdots & & \vdots
\end{array}\right|
$$

$$
C': \quad \begin{array}{ll}
0\ \ldots & \\
1\ 1\ 1\ 1\ 0\ \ldots & \text{(a)} \\
1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ \ldots & \text{(b)} \\
0\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ \ldots & \text{(a}\oplus\text{b)} \\
1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ \ldots & \text{(c)} \\
0\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ \ldots & \text{(a}\oplus\text{c)} \\
0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ \ldots & \text{(b}\oplus\text{c)} \\
1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ \ldots & \text{(a}\oplus\text{b}\oplus\text{c)} \\
\vdots & \vdots
\end{array} \qquad (14)
$$

As before, let $\rho_\ell = (1-\phi_\ell)/(1+\phi_\ell)$. Then substituting (14) into (13) gives

$$\hat{c}_0 = 0 \text{ iff } \rho_0 + \rho_1\rho_2\rho_3 + \rho_2\rho_4\rho_5\rho_6\rho_7 + \rho_0\rho_1\rho_3\rho_4\rho_5\rho_6\rho_7 + \ldots > 0 \ . \qquad (15)$$

The decoding diagram corresponding to (15) is shown in Figure 2. This takes the form of a trellis diagram for the (4,1,3) dual code C' with the c'_{j0} positions in the branch labels complemented. (In general, to decode r_m the c'_{jm} positions would be complemented.) Note that the all-zero state acts as the accumulator for the terms of (15).

Since a different storage unit must be used for each symbol to be decoded, the amount of storage for this type of decoder grows linearly with the length of the transmitted code sequence. This is also true of a Viterbi decoder, which must keep track of its path-elimination decisions. Of course a Viterbi decoder for the (4,3,3) code would be considerably more complex, since the trellis would have 64 states instead of the 4 states of the decoder in Figure 2.

IV. DISCUSSION

We have presented a symbol-by-symbol decoding rule for linear codes which is optimum in that it minimizes the probability of symbol error on a time-discrete memoryless channel when the code words are equiprobable. A comment or two on the relationship of this technique to correlation/Viterbi decoding, which is optimum in that it minimizes the same channels, would seem to be in order.

First, although the performance of correlation/Viterbi decoding is inferior to the performance of the decoding rule presented here on a symbol-error basis, and vice versa on a word-error basis, some preliminary simulation results for the Gaussian channel suggest that the two approaches are very close in performance on either basis. Symbol-error-rate is generally considered to be a better measure of performance than word-error-rate, especially in the case of convolutional codes, and this would seem to give a slight edge to our decoding rule. On the other hand, correlation/Viterbi decoding is applicable to nonlinear as well as linear codes, which might be of some advantage. Our present feeling is that for all practical purposes the two approaches give essentially the same performance.

When we turn to the question of complexity, there is of course a radical difference between the two decoding techniques. Correlation/Viterbi decoding is only practical for low rate or short codes whereas our decoding rule is only practical for

high rate or short codes. We are fairly well convinced, and the reader may be able to convince himself by studying the examples in Section III, that the complexity of our decoding rule for an (n,k) linear code is comparable to the complexity of a correlation/Viterbi decoder for the (n,n-k) dual code. This is fairly easy to see in the case of linear block codes and not so obvious in the case of convolutional codes since there are so many options and programming tricks to be considered. The authors, however, are firm believers in the coding-complexity Folk Theorem: "The complexity of any operation involving a linear code is comparable to the complexity of essentially that same operation involving the dual code". (In fact, it was the unsatisfying lack of a decoding method for high rate linear codes that was "dual" to correlation/Viterbi decoding that motivated the research reported here.) If our intuition is correct, then our scheme and correlation/Viterbi decoding should be of about the same complexity for rate 1/2 codes.

Finally, we remark that the decoding rule presented here, where all words of the dual code are used in the decoding process and the "soft decision function" is the finite Fourier transform $\sum_{i=0}^{p-1} \omega^{ij} \Pr(r_\ell | i)$, is a very important but very special limiting case of a general approach to soft decision decoding which we discuss in a companion paper [7].

REFERENCES

1. A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," IEEE Trans. Inform. Theory, vol. IT-13, pp. 260-269, Apr. 1967.

2. J. L. Massey, Threshold Decoding, Cambridge, Mass.: M.I.T. Press, 1963.

3. W. W. Peterson, Error-Correcting Codes, pp. 207-212, Cambridge, Mass.: M.I.T. Press, 1961.

4. L. H. Loomis, An Introduction to Abstract Harmonic Analysis. New York: Van Nostrand, 1953.

5. L. D. Rudolph, "A class of majority logic decodable codes," IEEE Trans. Inform. Theory (Corresp.), vol. IT-13, pp. 305-307, Apr. 1967.

6. W. W. Peterson and E. J. Weldon, Jr., Error-Correcting Codes, 2nd ed., Cambridge, Mass.: M.I.T. Press, 1972.

7. L. D. Rudolph and C. R. P. Hartmann, "Algebraic analog decoding," to be submitted to IEEE Trans. Inform. Theory.

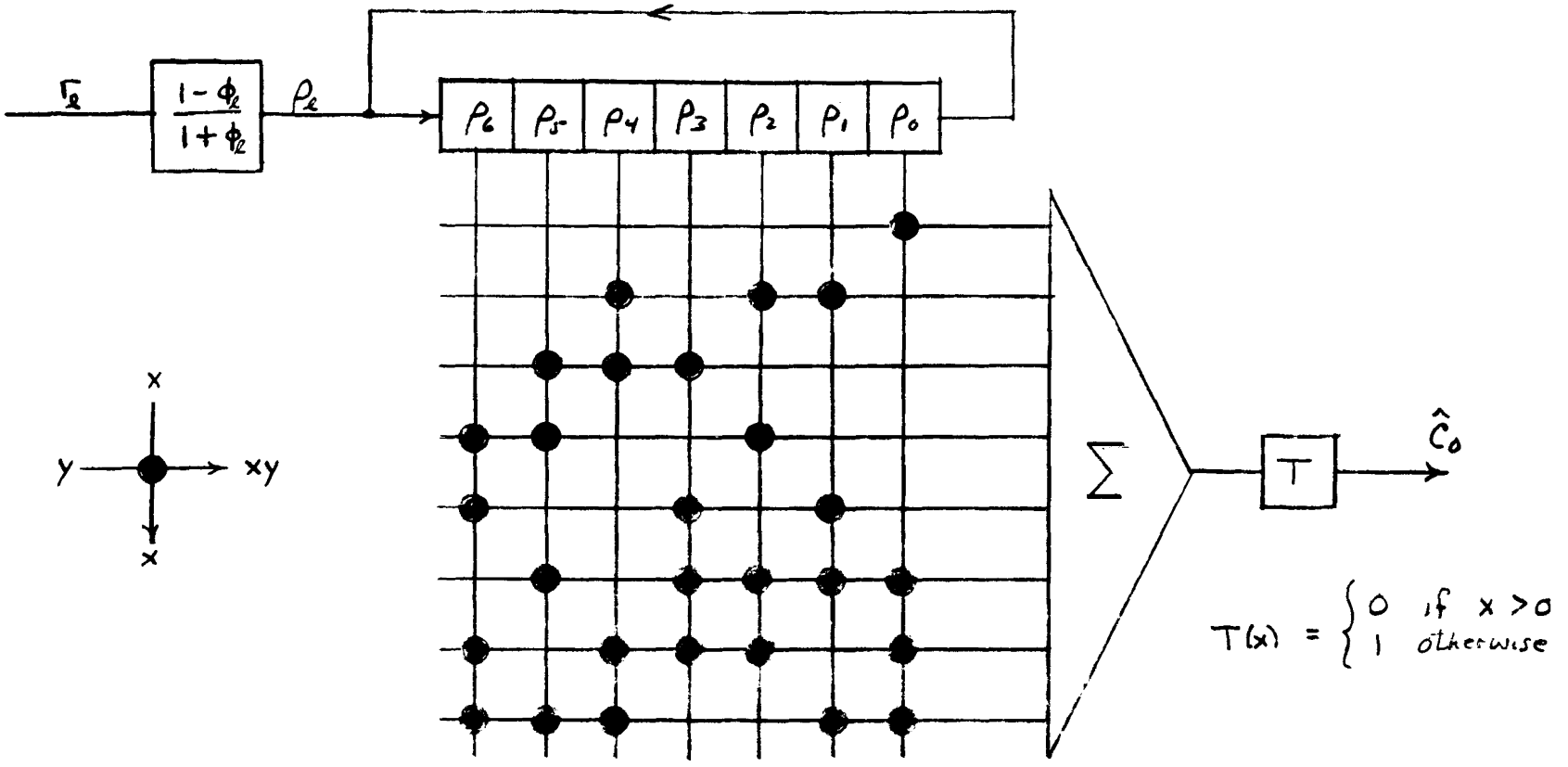


Figure 1. Decoder for the (7,4) code.

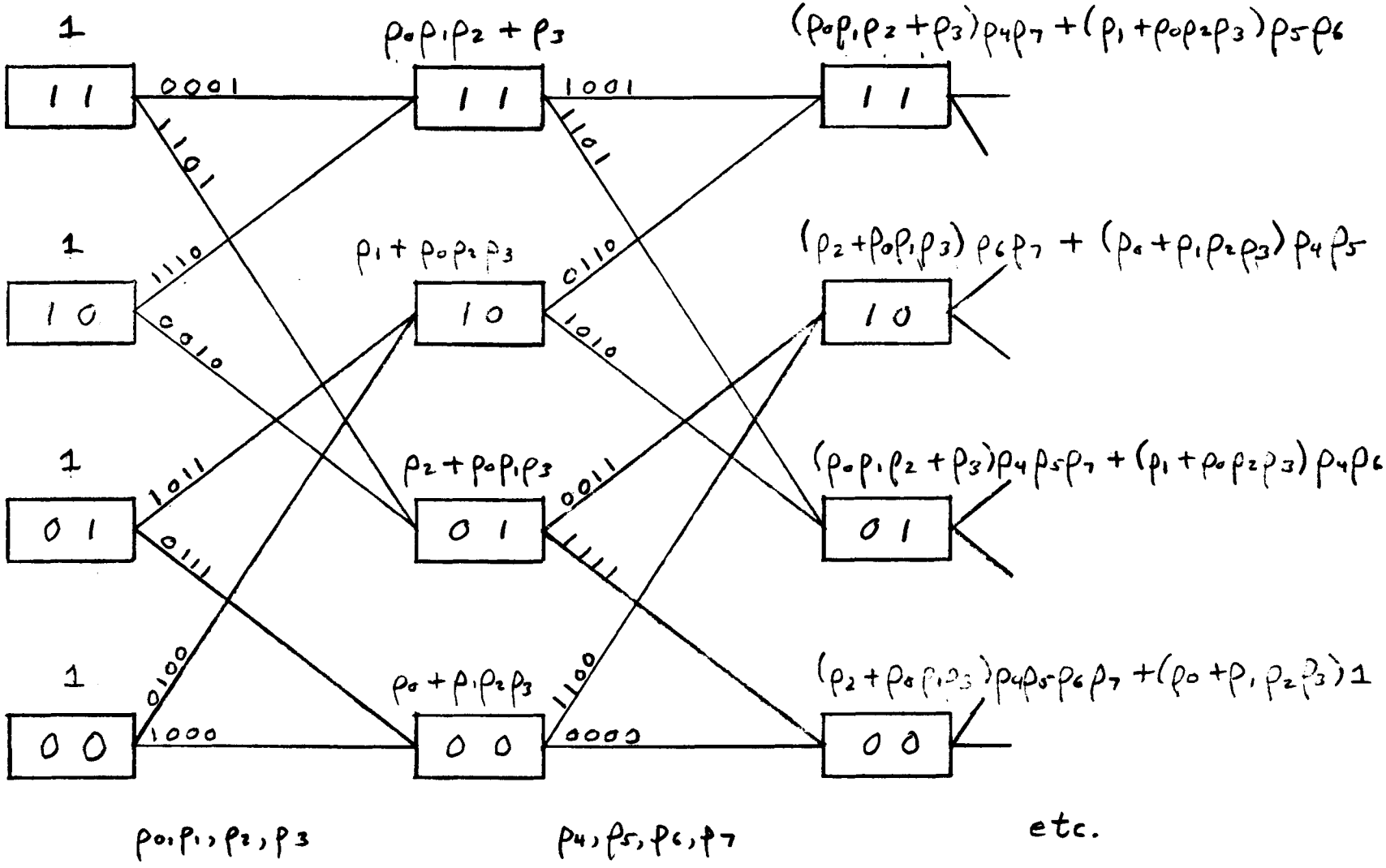


Figure 2. Decoder for the (4,3,3) code.