Department of

ELECTRICAL ENGINEERING

UNIVERSITY OF NOTRE DAME, NOTRE DAME, INDIANA

A UNIFIED MARKOVIAN ANALYSIS OF DECODERS FOR CONVOLUTIONAL CODES

by

Thomas N. Morrissey, Jr.

October 24, 1968

Technical Report No. EE-687

Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

This work was supported by the National Aeronautics and Space Administration (NASA Grant 15-004-026) in liaison with the Flight Data Systems Branch of the Goddard Space Flight Center.

The first five chapters of this report are essentially Notre Dame Technical Report No. EE-682, "Analysis of Decoders For Convolutional Codes by Stochastic Sequential Machine Methods."

ACKNOWLEDGMENT

I am greatly indebted to Professor James L. Massey for the educational experience received while working under him as a thesis student, and for his many helpful suggestions pertaining to the reported research.

I would also like to acknowledge the contribution of Professor Celso de Renna e Souza, who first suggested the stochastic sequential machine approach for calculating P_{FD}.

I wish to thank the National Aeronautics and Space Administration for its support of my last two years of doctoral studies.

To Mrs. C. A. Barnhart I am also grateful for typing the first half of this report.

ABSTRACT

Previous analyses of the error probability for feedback decoding of convolutional codes have focused almost exclusively upon the syndrome portion of the decoder, the contents of which are statistically dependent upon the infinite past due to the feedback of previous decoding estimates. This viewpoint makes exact error probability calculations intractable. In this paper, however, the entire decoder is modeled as an autonomous stochastic sequential machine and finite Markov chain theory applied in order to obtain a precise expression for $P_{FD}(u)$, the probability of error associated with the feedback decoding of the u^{th} subblock of information digits, thus circumventing the problems imposed by the dependencies on the infinite past. The analysis technique developed here applies to any syndrome feedback decoder for a systematic, rate $R = \frac{K_o}{N_o}$ convolutional code of memory order m over GF(2), used for transmission over a binary symmetric channel.

The limit of $P_{FD}(u)$ as u tends to infinity, when the limit exists, is termed P_{FD}, the steady-state probability of error of feedback decoding. Sufficient conditions on decoders are given in order for P_{FD} to exist, and two classes of minimum distance decoders exhibited which meet these sufficient conditions.

P_{FD} is calculated for a particular simple example and found to satisfy $P_{FD} < P_{DD}$, $0 < p < \frac{1}{2}$, where P_{DD} is the

probability of error associated with definite (i.e., feedback free) decoding of the same code, and p is the transition probability of the binary symmetric channel.

The stochastic sequential machine approach is also used in order to calculate the probability of error associated with semi-definite decoding, a decoding technique intermediate to feedback and definite decoding. Sufficient conditions are given for $P_{SDD}(k)$, the probability of error of a k stage semi-definite decoder, to tend to P_{FD} as k tends to infinity. Also, examples are exhibited for which there exists a particular value of k, k_α, such that $P_{SDD}(k_\alpha)$ is strictly less than both P_{FD} and P_{DD}, indicating that semi-definite decoding may be of some practical value.

$P_{FD}(u)$ is found to satisfy $P_{MLGD} \leq P_{FD}(u)$, u = 0,1,2,--, where P_{MLGD} is the probability of error associated with a "maximum likelihood genie decoder" for the same code. Also, for a particular class of codes and feedback decoding rules, the relationship $P_{FD} < P_{DD}$ is established in the limit as the binary symmetric channel transition probability p approaches zero.

Viterbi decoders for convolutional codes associated with non-finite encoding trees are modeled as finite autonomous stochastic sequential machines for the purpose of calculating the probability of error per information digit. Steady-state state occupancy probabilities and thus steady-

state probability of decoding error are found to exist for such decoders for all convolutional codes. Also, it is demonstrated that even though a maximum likelihood *sequence* of information digits is estimated by a Viterbi decoder, the probability of error per information digit is not necessarily minimized. Finally, for a restricted range of code rates, it is proved that there exist convolutional codes such that the Viterbi decoding probability of error per information digit decreases exponentially with increasing encoding constraint length, thus demonstrating that Viterbi decoders need not be periodically resynchronized in order to obtain "good" decoding performance.

TABLE OF CONTENTS

ACKNOWLEDGMENT i

ABSTRACT ii

CHAPTER I INTRODUCTION 1

CHAPTER II EXAMPLE OF STOCHASTIC SEQUENTIAL MACHINE APPROACH 6

CHAPTER III GENERAL STOCHASTIC MACHINE APPROACH 11

CHAPTER IV EXISTENCE OF STEADY-STATE STATE OCCUPANCY PROBABILITIES FOR CERTAIN CLASSES OF DECODERS 20

CHAPTER V APPLICATION OF STOCHASTIC SEQUENTIAL MACHINE THEORY TO SEMI-DEFINITE DECODING 31

CHAPTER VI BOUNDS ON P_{FD} FOR SPECIAL CASES 43

CHAPTER VII APPLICATION OF STOCHASTIC SEQUENTIAL MACHINE THEORY TO VITERBI DECODING 55

CHAPTER VIII SUMMARY, CONCLUSIONS, AND SUGGESTIONS FOR FURTHER RESEARCH 85

REFERENCES 92

FIGURES 94

A UNIFIED MARKOVIAN ANALYSIS
OF DECODERS FOR CONVOLUTIONAL CODES ★

by

Thomas N. Morrissey, Jr.

I. Introduction

A diagram of a general, binary, systematic, rate $R = \frac{1}{2}$ convolutional coding and decoding scheme at time u + m is shown in Figure 1.

The digits i(t), p(t), e(t), ξ(t), and s(t) represent the information, parity, information error, parity error, and syndrome input digits respectively at time t, t = 0, 1, 2, ..., $\underline{\sigma}(t) \triangleq (\sigma_0(t), \sigma_1(t) \cdots, \sigma_{m-1}(t))$ is the m dimensional column vector representing the state of the syndrome register at time t, and $e^*(u) = f(\underline{\sigma}(u+m), s(u+m))$ is the estimate of e(u) formed at time u+m, where the decoding function f is designed to provide a "reasonable" estimate of e(u) based upon $\underline{\sigma}(u+m)$ and s(u+m).

In the sequel column vectors will be denoted by parentheses () and row vectors by brackets [].

All digits are elements of GF(2), the finite field of two elements, and all operations are assumed to be carried out in this field. The plus and minus signs in Fig. 1 represent identical operations in GF(2), and are distinguished here to emphasize the operation of the decoding process.

★ This work was supported by the National Aeronautics and Space Administration (NASA Grant NGR 15-004-026) in liaison with the Flight Data Systems Branch of the Goddard Space Flight Center, and forms a dissertation to be submitted in partial fulfillment of the requirements for the doctor's degree in electrical engineering at the University of Notre Dame, Notre Dame, Indiana.

The noise digits are assumed to be statistically independent, each having probability p of being a "1" and probability 1-p of being a "0". This noise source model is equivalent, with respect to the transmission of the information and parity digits, to a memoryless binary symmetric channel (BSC) with crossover probability p. Given that a "0" or "1" is transmitted over the BSC, the digit is received incorrectly with probability p, the probability that a noise source in Fig. 1 will generate a "1" and complement the corresponding transmitted binary digit.

Since the information sequence appears unaltered among the transmitted sequences, the code is said to be in systematic form. The remaining transmitted sequence is called the parity sequence.

The syndrome input digits are formed by passing the received information sequence through a replica of the discrete linear filter used to form the parity sequence at the encoder, and subtracting from the result the received parity sequence. Since the system is linear with respect to the formation of the syndrome input digits, this sequence is equal to the sum of the components at the input of the syndrome register resulting from the information and noise sources. However, the component due to the information source stream is formed by passing the information sequence through two identical discrete linear filters in parallel and subtracting the output of one from the other. Hence this component is zero, and the syndrome state and input are independent of the information stream and depend only on the noise digits.

Since the code is in systematic form, $e^*(u)$ is subtracted from the output of the encoder replica in the decoder in order to form $i^*(u) = i(u) + (e(u) - e^*(u))$, the estimate at time u+m of the information digit at time u. Thus the associated probability of error $Pr(i^*(u) \neq i(u))$ is equal to $Pr(e^*(u) \neq e(u))$, and is independent of the information sequence since the syndrome state and input used to form $e^*(u)$ are functions only of the channel errors.

Three modes of decoder operation, depicted by the position of the switch, are shown in Fig. 1.

The position labeled DD, in which the syndrome register is not modified, corresponds to the decoding scheme known as <u>definite decoding</u> [1]. For definite decoding,

$$\begin{aligned}
\sigma_o(u+m) &= s(u) = g_m e(u-m) + \cdots + g_1 e(u-1) + g_o e(u) && -\xi(u) \\
\sigma_1(u+m) &= s(u+1) = g_m e(u-m+1) + \cdots + g_1 e(u) + g_o e(u+1) && -\xi(u+1) \\
&\vdots && \vdots \\
\sigma_{m-1}(u+m) &= s(u+m-1) = g_m e(u-1) + g_{m-1} e(u) + \cdots + g_o e(u+m-1) && -\xi(u+m-1) \\
s(u+m) &= g_m e(u) + \cdots + g_o e(u+m) && -\xi(u+m)
\end{aligned} \qquad (1)$$

and thus a total of 3m+2 statistically independent noise bits affect the estimate $e^*(u)$.

The definite decoding probability of error is defined to be $P_{DD}(u) \triangleq Pr(e^*(u) \neq e(u)/_{DD\ mode})$. For $u \geq m$ the probability distribution for $(e(u-m), \cdots, e(u), \cdots, e(u+m), \xi(u), \cdots \xi(u+m)$, and therefore $P_{DD}(u)$, is independent of u. Hence the steady-state decoding probability of error for definite decoding, P_{DD}, is simply equal to $P_{DD}(m)$.

Since in definite decoding $e(u-m), \cdots, e(u-1)$ affect the estimate $e^*(u)$ but have already been estimated by the decoder, presumably with low probability of an incorrect decision, it would seem reasonable to use $e^*(u-m), \cdots, e^*(u-1)$ in an attempt to cancel the effect of $e(u-m), \cdots, e(u-1)$ in $(\underline{\sigma}(u+m), s(u+m))$. It is useful to assume that the true values of $e(u-m), \cdots e(u-1)$

have been estimated, in which case the syndrome state and input are functions only of 2m+2 statistically independent noise digits, a situation which intuitively is superior to definite decoding. This is done in the decoding position labeled GD, or "<u>genie decoding</u>" after Wozencraft and Jacobs [2], in which use is made of a genie who corrects the estimated error bits, if necessary, and feeds them back in order to form the modified syndrome equations

$$
\begin{aligned}
\sigma_o(u+m) &= g_o e(u) && -\xi(u) \\
\sigma_1(u+m) &= g_1 e(u) + g_o e(u+1) && -\xi(u+1) \\
&\;\;\vdots && \;\;\vdots \\
\sigma_{m-1}(u+m) &= g_{m-1} e(u) + \text{--------} + g_o e(u+m-1) && -\xi(u+m-1) \\
s(u+m) &= g_m e(u) + \text{----------} + g_1 e(u+m-1) + g_o e(u+m) && -\xi(u+m).
\end{aligned}
\qquad (2)
$$

Since genies are rumored to be quite scarce, the genie decoding mode is used merely as a handy conceptual tool for calculating an upper bound on system performance. The probability of error associated with genie decoding, $P_{GD}(u)$, is defined to be $Pr(e^*(u) \neq e(u)/_{GD\ mode})$. Since this probability is independent of u for $u \geq 0$, the steady-state error probability for genie decoding, P_{GD}, is simple $P_{GD}(0)$.

In the mode of operation labeled FD for <u>feedback decoding</u>, error estimates are fed back to modify the syndrome without the services of a genie. The resulting equations are

$$
\begin{aligned}
\sigma_o(u+m) &= g_m(e(u-m)-e^*(u-m)) + \text{---} + g_1(e(u-1)-e^*(u-1)) + g_o e(u) && -\xi(u) \\
\sigma_1(u+m) &= g_m(e(u-m+1)-e^*(u-m+1)) + \text{---} + g_1 e(u) + g_o e(u+1) && -\xi(u+1) \\
&\;\;\vdots && \;\;\vdots \\
\sigma_{m-1}(u+m) &= g_m(e(u-1)-e^*(u-1)) + g_{m-1} e(u) + \text{---} + g_o e(u+m-1) && -\xi(u+m-1) \\
s(u+m) &= g_m e(u) + \text{-----} + g_o e(u+m) && -\xi(u+m).
\end{aligned}
\qquad (3)
$$

As long as no decoding mistakes have been made, feedback decoding coincides with genie decoding; however, decoding errors in feedback decoding affect the syndrome as would a pattern of channel errors, and could result in further decoding mistakes even in the absence of additional channel noise [3].

The probability of error associated with feedback decoding, $P_{FD}(u)$, is the quantity $Pr(e^*(u) \neq e(u)_{/FD\ mode})$. The quantity $\lim_{u\to\infty} P_{FD}(u)$, when it exists, is denoted as P_{FD}, the steady-state probability of error of feedback decoding.

$[e^*(u)]_{FD}$, $[e^*(u)]_{DD}$, and $[e^*(u)]_{GD}$ will be used to denote the estimates made by the feedback, definite, and genie decoding operations respectively when it is necessary to distinguish which decoding method is being used.

For both definite and genie decoding the syndrome state and input are functions only of statistically independent channel error digits over a finite span, and $Pr(e^*(u) \neq e(u))$ may be easily calculated in principle by summing the probabilities of those error patterns for which $e^*(u) \neq e(u)$.

However, for feedback decoding, the estimated digits $e^*(u-m)$,---,$e^*(u-1)$ affecting $\underline{\sigma}(u+m)$ are dependent among themselves and upon $e(u-m)$,---,$e(u)$,------,$e(u+m-1)$, $\xi(u)$,---$\xi(u+m-1)$, the channel noise digits affecting the syndrome state. Alternatively, the syndrome state and input are dependent on the entire past history of the error sequences, i.e., upon $\xi(u+m)$, $\xi(u+m-1)$,------,$\xi(0)$, $e(u+m)$, $e(u+m-1)$,---,$e(0)$. Thus if attention is focused exclusively upon the syndrome equations for the purpose of calculating the probability of error, the complexity of the calculation necessary in order to calculate $P_{FD}(u)$ exactly grows exponentially with increasing time u.

In this paper the decoder is modeled as an autonomous stochastic sequential machine [4,5] and finite Markov chain theory applied in order to calculate the feedback decoding probability of error exactly. This approach can best be illustrated by an example.

II. EXAMPLE OF STOCHASTIC SEQUENTIAL MACHINE APPROACH

Consider the special case ($m=1$, $g_o=1$, $g_1=1$, $f(\sigma_o(u),s(u))=\sigma_o(u)\cdot s(u)$) illustrated in Fig. 2 of the general systematic rate $R = \frac{1}{2}$, binary convolutional coding and decoding scheme in Fig. 1.

Since with syndrome decoding the probability of error is independent of the information stream, as was stated in the previous section, the all-zero information sequence may be assumed without loss of generality. Thus for the purpose of calculating $\Pr(i^*(u) \neq i(u)) = \Pr(e^*(u) \neq e(u))$, the encoder portion of Fig. 2 may be ignored and the system modeled as in Fig. 3.

When the switch is in the DD position

$$\begin{aligned}[e^*(u)]_{DD} &= s(u+1)\cdot\sigma_o(u+1) = s(u+1)\cdot s(u)\\ &= (e(u)+e(u+1)-\xi(u+1))\cdot(e(u-1)+e(u)-\xi(u))\\ &= e(u)+e(u)[e(u-1)-\xi(u)+e(u+1)-\xi(u+1)]\\ &\quad +e(u+1)e(u-1)-e(u+1)\xi(u)-\xi(u+1)e(u-1)+\xi(u+1)\xi(u)\end{aligned}$$

and thus $P_{DD} = \Pr([e^*(u)]_{DD} \neq e(u))$

$$= \Pr[e(u)\{e(u-1)-\xi(u)+e(u+1)-\xi(u+1)\}+e(u+1)e(u-1)-e(u+1)\xi(u)-\xi(u+1)e(u-1)+\xi(u+1)\xi(u)=1].$$

Since the error digits in this expression are statistically independent, each having probability p of being a "1", P_{DD} may be easily calculated to be

$$P_{DD} = 8p^2(1-p)^3 + 4p^3(1-p)^2 + 4p^4(1-p). \qquad (4)$$

When the switch is in the GD position

$$\begin{aligned}[e^*(u)]_{GD} &= s(u+1)\cdot\sigma_o(u+1)\\ &= (e(u)+e(u+1)-\xi(u+1))\cdot(e(u)-\xi(u))\\ &= e(u)+e(u)[e(u+1)-\xi(u+1)-\xi(u)]-e(u+1)\xi(u)+\xi(u+1)\cdot\xi(u)\end{aligned}$$

and $P_{GD} = \Pr([e^*(u)]_{GD} \neq e(u))$

$$= \Pr[e(u)\{e(u+1)-\xi(u+1)-\xi(u)\}-e(u+1)\xi(u)+\xi(u+1)\cdot\xi(u) = 1].$$

The error digits here also are statistically independent, and P_{GD} may be computed to be

$$P_{GD} = 5p^2(1-p)^2 + 2p^3(1-p) + p^4. \qquad (5)$$

However, for feedback decoding

$$[e^*(u)]_{FD} = s(u+1)\cdot\sigma_o(u+1)$$

$$= (e(u)+e(u+1)-\xi(u+1))\cdot(e(u-1)-[e^*(u-1)]_{FD}+e(u)-\xi(u))$$

$$= e(u)+[e(u-1)-[e^*(u-1)]_{FD}][e(u)+e(u+1)-\xi(u+1)]$$

$$+e(u)e(u+1))-e(u)\xi(u+1))-e(u)\xi(u)-e(u+1)\cdot\xi(u)+\xi(u)\xi(u+1),$$

and $P_{FD}(u) = Pr([e^*(u)]_{FD} \neq e(u))$

$$= Pr[\{e(u-1)-[e^*(u-1)]_{FD}\}\{e(u)+e(u+1)-\xi(u+1)\}+e(u)e(u+1)-e(u)\xi(u+1)$$

$$-e(u)\xi(u)-e(u+1)\xi(u)+\xi(u)\xi(u+1)=1].$$

But $[e^*(u-1)]_{FD}$ depends upon $e(u)$, $\xi(u)$ and the previous error digits $e(u-1),---,e(0)$ and $\xi(u-1),---\xi(0)$. Therefore the procedure for calculating $Pr([e^*(u)]_{FD} \neq e(u))$ by means of summing the probabilities of the decoding-error-causing error patterns as was done for definite and genie decoding is extremely unwieldy and, for all practical purposes, impossible for large u.

However this problem of infinite past history dependence is circumvented by the representation of the (feedback) decoder of Fig. 3 as a four state sequential machine with two dimensional random input vector $(e(u+1),\xi(u+1))$ and single output $e^*(u)$ at time u+1. The four states are denoted as $q_o=00$, $q_1=01$, $q_2=10$, and $q_3=11$, where the first digit represents the content of the buffer and the second the content of the syndrome register.

The decoder then may be modelled as a four-state autonomous stochastic machine in which the states are the same as the original sequential machine, and the transition probabilities between pairs of states are the sums of the probabilities of those input noise vectors that take the machine from the first state to the second. The state-behavior of the four-state autonomous stochastic machine is thus described by the following Markov transition matrix:

$$\pi = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} = \begin{bmatrix} (1-p)^2 & p(1-p) & p^2 & p(1-p) \\ 1-p & 0 & p & 0 \\ p(1-p) & (1-p)^2 & p(1-p) & p^2 \\ 1-p & 0 & p & 0 \end{bmatrix},$$

where P_{ij} is the conditional probability that the system will be in state q_j at time u+1 given that it is in state q_i at time u.

As an example of how the P_{ij}'s were calculated for this decoder, assume that the system is in state q_1=01 at time u. Now if e(u)=0 and $\xi(u)$=0, or e(u)=0 and $\xi(u)$=1, the state at time u+1 will be q_o=00; and if e(u)=1 and $\xi(u)$=0, or e(u)=1 and $\xi(u)$=1, the state at time u+1 will be q_2=10.

Therefore, $P_{11}=0$, $P_{13}=0$,

$$P_{10}=Pr[e(u)=0,\xi(u)=0]+Pr[e(u)=0,\xi(u)=1]=(1-p)^2+p(1-p)=1-p,$$

and

$$P_{12}=Pr[e(u)=1,\xi(u)=0]+Pr[e(u)=1,\xi(u)=1]=p(1-p)+p^2=p.$$

The fact that e(u) and $\xi(u)$ are statistically independent of one another and of the assumed state at time u makes the calculation of the P_{ij}'s straightforward. The actual state at time u accounts for all the past history of the error sequences that is relevant to future operation of the decoder.

The probabilistic transition diagram for this decoder is shown in Fig. 4.

Let $\underline{W}(u) \triangleq [W_o(u),W_1(u),W_2(u),W_3(u)]$ be the state probability vector at time u; i.e., $W_i(u)$ is the probability that the system is in state q_i at time u.

Now $\underline{W}(u) = \underline{W}(0)\pi^u$, with $\underline{W}(0) \triangleq [1000]$ since a feedback decoder is by definition in the all-zero state at the beginning of the decoding process.

With each state q_i is associated a probability of error Pq_i equal to the sum of the probabilities of those noise inputs $(e(u+m), \xi(u+m)) = (e(u+1), \xi(u+1))$ that cause $e^*(u) \neq e(u)$, given that the system is in state q_i at time $u+m=u+1$. For the example of Fig. 3, these quantities are readily found to be

$$Pq_0 = 0$$
$$Pq_1 = 2p(1-p)$$
$$Pq_2 = 1$$
$$Pq_3 = 2p(1-p).$$

As an example of the above calculations, if the system is in state $q_3=11$ at time $u+1$, and $e(u+1)=0$ and $\xi(u+1)=0$, or $e(u+1)=1$ and $\xi(u+1)=1$, then $e^*(u)=e(u)=1$. On the other hand, if $e(u+1)=0$ and $\xi(u+1)=1$, or $e(u+1)=1$ and $\xi(u+1)=0$, then $0=e^*(u) \neq e(u) = 1$, and an incorrect decision is made. Thus $Pq_3 = Pr[e(u+1)=0,\ \xi(u+1)=1] + Pr[e(u+1)=1,\ \xi(u+1)=0] = 2p(1-p)$.

$P_{FD}(u)$ may thus be written as

$$P_{FD}(u) = \sum_{i=0}^{3} W_i(u+1)Pq_i. \tag{6}$$

A <u>sufficient</u> condition for P_{FD} to exist is that $\lim_{u\to\infty} \underline{W}(u) = \underline{W} \triangleq [W_0, W_1, W_2, W_3]$, the steady-state decoder state probability vector, with W_i equal to the steady-state probability of state q_i. If this is the case then $P_{FD} \triangleq \lim_{u\to\infty} P_{FD}(u) = \lim_{u\to\infty} \sum_{i=0}^{3} W_i(u+1)Pq_i = \sum_{i=0}^{3} W_i Pq_i$. $\underline{W}$ does in fact exist for this example (see Theorem 1 in the next section) and may be calculated by solving $\underline{W}\pi = \underline{W}$ for $\underline{W}$ with the added constraint $\sum_{i=0}^{3} W_i = 1$. The desired solution is

$$W_0 = \frac{1-2p+3p^2-2p^3}{1+3p^2-2p^3} \qquad W_2 = \frac{3p^2-2p^3}{1+3p^2-2p^3}$$
$$W_1 = \frac{p-3p^3+2p^4}{1+3p^2-2p^3} \qquad W_3 = \frac{p-3p^2+5p^3-2p^4}{1+3p^2-2p^3} \tag{7}$$

and thus $P_{FD} = \sum_{i=0}^{3} W_i P q_i = \frac{7p^2 - 12p^3 + 10p^4 - 4p^5}{1 + 3p^2 - 2p^3}$. (8)

P_{GD}, P_{DD}, and P_{FD} for this example are sketched in Fig. 5 as a function of p. Note that for small p, the asymptotic values $5p^2$, $7p^2$, and $8p^2$ are approached by P_{GD}, P_{FD}, and P_{DD} respectively.

It can be shown that $P_{GD} < P_{FD} < P_{DD}$ for $0 < p < \frac{1}{2}$, which illustrates the <u>global</u> (i.e., all $p < \frac{1}{2}$) <u>superiority</u> of feedback decoding over definite decoding for this example. This is the first instance known that P_{FD} has been calculated exactly and compared with P_{DD}, although intuitively one suspects that in general $P_{FD} < P_{DD}$ for sufficiently small p.

III. GENERAL STOCHASTIC MACHINE APPROACH

More generally, consider the arbitrary syndrome feedback decoder shown in Fig. 6 at time u+m for an $R = \frac{K_o}{N_o}$ binary systematic convolutional code with memory order m. For full details of the general syndrome feedback decoder, see the literature [6].

The decoder is described as follows: $\underline{S}(u+m) = G'\underline{E}(u+m)$, where $\underline{S}(u+m) \triangleq (S^{(K_o+1)}(u+m),---,S^{(N_o)}(u+m))$ is the syndrome input at time u+m. And $\underline{E}(u+m) \triangleq (\underline{e}(u):---:\underline{e}(u+m))$, where $\underline{e}(i) \triangleq (e^{(1)}(i),---, e^{(No)}(i))$, is the N_o dimensional random input column vector at time i with components which are statistically independent and have probability p of being a "1", $0<p<\frac{1}{2}$. The superscripts above the error and syndrome digits refer to the particular input sequence and syndrome register respectively of these digits. The matrix G' is given by

$$G' \triangleq [G'_m : G'_{m-1} :---: G'_o]$$

where

$$G'_i \triangleq [G_i : I_{N_o-K_o}],\ i = o$$

$$G'_i \triangleq [G_i : \bar{0}\ ],\ i = 1, 2, --, m.$$

and

$$G_i \triangleq \begin{bmatrix} g_{i(1)}^{(K_o+1)} & ---- & g_{i(K_o)}^{(K_o+1)} \\ \vdots & & \vdots \\ g_{i(1)}^{(N_o)} & ------ & g_{i(K_o)}^{(N_o)} \end{bmatrix} \quad i = 0, 1, --, m. \quad (9)$$

$I_{N_o-K_o}$ is the $(N_o-K_o) \times (N_o-K_o)$ identity matrix, $\bar{0}$ is the $(N_o-K_o) \times (N_o-K_o)$ all zero matrix, and $g_{i(j)}^{(k)}$ is the component of the response on the k^{th} output line of the encoder at time i due to an excitation (a "1") on the j^{th} input line at time 0. $j = 1, 2, ---, K_o$. $k = K_o+1, K_o+2,---, N_o$.

Now the operation of the syndrome register is described by

$$\underline{\sigma}(u+1) = A\,\underline{\sigma}(u) + B\underline{S}(u) + G_{FB}\,\underline{e}^*_I(u-m). \quad (10)$$

$$u = o, 1, 2,---, \quad \underline{\sigma}(0) \triangleq \underline{0},$$

where $\underline{\sigma}(u) \triangleq (\underline{\sigma}_0(u) : \underline{\sigma}_1(u) : --- : \underline{\sigma}_{m-1}(u))$,

with $\underline{\sigma}_i(u) \triangleq (\sigma_i^{(K_0+1)}(u), \sigma_i^{(K_0+2)}(u), ---, \sigma_i^{(N_0)}(u))$,

is the $m(N_0-K_0)$ dimensional column vector representing the syndrome state at time u, and

$$A \triangleq \begin{bmatrix} \tilde{0} & I_{N_0-K_0} & \tilde{0} & - & - & \tilde{0} \\ \tilde{0} & \tilde{0} & I_{N_0-K_0} & & & \tilde{0} \\ | & | & & \ddots & & I_{N_0-K_0} \\ \tilde{0} & \tilde{0} & - & - & - & \tilde{0} \end{bmatrix} \tag{11}$$

$$B \triangleq \begin{bmatrix} \tilde{0} \\ \tilde{0} \\ | \\ \tilde{0} \\ I_{N_0-K_0} \end{bmatrix} \tag{12}$$

$$G_{FB} \triangleq \begin{bmatrix} G_1 \\ \cdot\cdot \\ G_2 \\ \cdot\cdot \\ | \\ \cdot\cdot \\ G_m \end{bmatrix}. \tag{13}$$

$\underline{e}^*_I(u)$, the estimate at time u+m of the first K_0 components (i.e., the information components) of $\underline{e}(u)$, is given by $\underline{e}^*_I(u) = \underline{f}(\underline{\sigma}(u+m) : \underline{S}(u+m))$, where $\underline{f}$ is a boolean function specified by the decoding algorithm. $\underline{\sigma}(u)$ may also be written as

$$\underline{\sigma}(u) = A^u \underline{\sigma}(0) + \sum_{i=0}^{u-1} A^i [B\underline{S}(u-1-i) + G_{FB}\underline{f}(\underline{\sigma}(u-1-i):\underline{S}(u-1-i))],$$

$$u = 1, 2, ---, \qquad \underline{\sigma}(0) \triangleq \underline{0} . \tag{14}$$

The estimates $\underline{e}^*_I(-m)$, $\underline{e}^*_I(-m+1)$, --, $\underline{e}^*_I(-1)$ above, although corresponding to no actual error digits since the first such digit is received at time u = 0, nevertheless must be considered in the syndrome state equations since the feedback loop of the decoder is assumed to be closed for all time; i.e., the feedback decoder is insensitive to the starting time.

Let $\underline{e}_I(u) \triangleq (e^{(1)}(u), e^{(2)}(u), \text{---}, e^{(K_o)}(u))$ denote the error pattern at time u in the information positions. Thus the vector $(\underline{e}_I(u) : \underline{e}_I(u+1) : \text{---} : \underline{e}_I(u+m-1) : \underline{g}(u+m))$ represents a possible state of the decoder; and since there are 2^{mN_o} distinct vectors of this form there are 2^{mN_o} possible decoder states. However, since the registers in a feedback decoder are initially filled with zeroes, only the set of states Q_{RO} reachable from the all-zero state need be considered. Thus the system may be represented as an r state Markov chain ($r = \#Q_{RO} \leq 2^{mN_o}$) with transition probabilities from a given state determined by that state and the probability distribution of the noise input vector.

Associated with each state $q_j \in Q_{RO}$ is a probability of error, P_{q_j}, which is the probability that $\underline{e}(u+m)$ is such that $\underline{e}^*_I(u) \neq \underline{e}_I(u)$, given that the system is in state q_j at time u+m.

If $W_j(u+m)$ is the probability that the decoder is in state q_j at time u+m, j = 0, 1,---, r-1, then $P_{FD}(u) \triangleq Pr(\underline{e}^*_I(u) \neq \underline{e}_I(u))$ is given by

$$P_{FD}(u) = \sum_{j=0}^{r-1} W_j(u+m)\, P_{q_j}, \qquad (15)$$

where

$$[W_o(u+m), W_1(u+m), \text{---}, W_{r-1}(u+m)] \triangleq \underline{W}(u+m) = \underline{W}(0) \cdot \pi^{u+m}. \qquad (16)$$

π is the rxr Markov transition matrix associated with the decoder, and $\underline{W}(0) \triangleq [W_0(0), W_1(0), \text{---}, W_{r-1}(0)] = [100\text{---}00]$, with q_o taken as the all-zero decoder state, is the initial probability distribution of the feedback decoder.

If steady-state probabilities exist for the Markov chain, then $\lim_{u \to \infty} \underline{W}(u) = \underline{W} \overset{\Delta}{=} [W_o, W_1, --, W_{r-1}]$, where W_j is the steady-state probability that the system is in state q_j. For this case,

$$P_{FD} = \lim_{u \to \infty} P_{FD}(u) = \sum_{j=0}^{r-1} W^{(j)} P_{q_j}, \tag{17}$$

the steady-state probability of error of feedback decoding.

Note that for large memory m and/or block length N_o this approach for calculating P_{FD} is impractical because the number of states which must be considered grows exponentially with mN_o. The Markov chains, however, are "loosely connected" since, there being only 2^{N_o} possible input vectors, each state can make a transition into at most 2^{N_o} other states.

The condition for existence of steady-state probabilities for a Markov chain is given by the following theorem [7].

Theorem 1. Let π be an r by r transition matrix associated with a finite Markov chain. Then steady-state probabilities exist if and only if there exists an integer $N, 1 \leq N \leq r-1$, such that π^N has a positive column, that is, a column all of whose elements are > 0.

The above condition is equivalent to the existence of a state q_j and a positive integer N such that starting from any initial state, q_j can be reached (with nonzero probability) is exactly N steps. This condition when applied to feedback decoders results immediately in the following corollary.

Corollary 1.1 For a syndrome feedback decoder for an $R = \frac{K_o}{N_o}$ systematic convolutional code with memory order m, a sufficient condition for steady-state state occupancy probabilities to exist is that there exists some positive integer N_{q_o} such that the all-zero state q_o is reachable from any initial state $q_j \in Q_{RO}$ in exactly N_{q_o} steps.

Now assume p = 0 and that the decoder is in the all-zero state q_o at time u+m. Now $p = 0 \Rightarrow \underline{S}(u+m) = \underline{0}$ and $\underline{E}(u+m) = \underline{0}$; $\underline{W}(u+m) = (10---0) \Rightarrow \underline{\sigma}(u+m) = \underline{0}$; and if correct estimates are made by a decoder for this noiseless case (all decoders in this paper are assumed to have this property), $\underline{e}^*_I(u) = \underline{f}(\underline{\sigma}(u+m) : \underline{S}(u+m)) = \underline{f}(\underline{0}) = \underline{0}$, and thus a transition from the all-zero state into itself results from the all-zero noise input vector. Thus if N_{q_o} is the maximum number of transitions, over all $q_j \varepsilon Q_{RO}$, required in order to reach q_o, and if state q_k can be driven to q_o in $N'_{q_o} \leq N_{q_o}$ steps, then q_k can be driven to q_o in exactly N_{q_o} steps by first taking q_k to q_o in N'_{q_o} steps and then driving q_o into itself $N_{q_o} - N'_{q_o}$ times. Hence the following corollary is obtained.

<u>Corollary 1.2</u> For a syndrome feedback decoder for a rate $R = \frac{K_o}{N_o}$ systematic convolutional code with memory order m, which decodes correctly in the absence of channel errors, a sufficient condition for steady-state state occupancy probabilities to exist is that the all-zero state q_o be reachable from any initial state $q_j \varepsilon Q_{RO}$.

Since the binary symmetric channel is assumed for this development, the random input vector $\underline{e}(u)$ assumes each of its 2^{N_o} possible values with nonzero probability at each time unit u. Thus if a sequence of inputs is found to drive the decoder from q_j to q_o, that sequence of inputs has nonzero probability.

By inspection, the K_o buffer registers which form the encoder replica may be driven to the all-zero state from any other buffer state by the shifting of m successive zeroes into each. And the inputs to the N_o-K_o syndrome registers may be controlled, independently of the states and inputs of the buffer registers, by proper choice of the N_o-K_o parity error sequences. The following theorem may thus be stated:

Theorem 2. For a syndrome feedback decoder for an $R = \frac{K_o}{N_o}$ systematic convolutional code with memory order m, which decodes correctly in the absence of channel errors, steady-state state occupancy probabilities exist if the nonlinear feedback shift register composed of the $N_o - K_o$ syndrome registers can be driven from any syndrome state $\underline{\sigma}_i$ reachable from $\underline{\sigma}_o = \underline{0}$ back to the $\underline{\sigma}_o$ state by means of a suitable choice of syndrome inputs.

A shift register is said to be "driven stable" [8] if any state reachable from $\underline{\sigma}_o = \underline{0}$ can be driven autonomously back into the $\underline{\sigma}_o$ state. With this definition the following corollary to Theorem 2 is immediate:

Corollary 2.1 If the syndrome register portion of a syndrome feedback decoder for an $R = \frac{K_o}{N_o}$, systematic convolutional code with memory order m, which decodes correctly in the absence of channel errors, is "driven stable", then steady-state state occupancy probabilities exist for the decoder.

The converse to this corollary is not necessarily true.

It might be noted that if a transition from any state to any other state is possible due to a malfunction in circuitry, no matter how small the probability of such a malfunction, then steady-state probabilities exist for the decoder.

The sufficient condition that the syndrome portion of the decoder be capable of being driven from any reachable state back to $\underline{\sigma}_o = \underline{0}$ is not necessary in order for $\underline{W}$ and P_{FD} to exist, as is shown by the following example.

Consider the $R = \frac{1}{2}$ systematic convolutional decoder shown in Fig. 7, with $f(\sigma_o(u+1), S(u+1)) = \overline{S(u+1)} \cdot \sigma_o(u+1)$. With $q_o = 00$, $q_1 = 10$, $q_2 = 01$, $q_3 = 11$, $Q_{RO} = \{q_0 q_1 q_2 q_3\}$, the associated Markov matrix is

$$\pi = \begin{bmatrix} (1-p)^2 & p^2 & p(1-p) & p(1-p) \\ p(1-p) & p(1-p) & (1-p)^2 & p^2 \\ 0 & 0 & 1-p & p \\ 0 & 0 & 1-p & p \end{bmatrix}.$$

Since column 3 of π is nonzero, steady-state state occupancy probabilities exist from Theorem 1 with N = 1, and $\underline{W}$ and P_{FD} may be easily calculated to be $\underline{W}$ = [0 0 1-p p] and $P_{FD} = p^2 + (1-p)^2$. However, states q_2 and q_3 , reachable from q_o , cannot be driven back to $\underline{\sigma}_o = 0$, and thus the sufficient condition given in Theorem 2 for $\underline{W}$ and thus P_{FD} to exist does not hold.

The state q_o is here a "transient state" in that it has zero steady-state probability ($W_o = 0$). In general, if steady-state probabilities exist for the decoder states and yet $q_o \triangleq 0$ is not reachable from $q_j \in Q_{RO}$, it follows that q_o has zero steady-state probability and hence $\lim_{p \to o} P_{FD} \geqq \frac{1}{m}$ since at least one decoding error must be made each m time units or else the syndrome register would clear itself.

As an example of a decoder for which neither steady-state state occupancy probabilities nor P_{FD} exists, consider the R = $\frac{1}{2}$ systematic decoder with $f(\underline{\sigma}(u+2), S(u+2)) = S(u+2) \cdot \sigma_1(u+2) \cup \overline{S(u+2)} \cdot \sigma_o(u+2)$, where "$\cup$" denotes "inclusive or", shown in Fig. 8.

The state diagram of the syndrome register portion of the decoder is shown in Fig. 9, with the first digit of the 2-tuple state representation representing $\sigma_i(\ )$ and the second $\sigma_o(\ )$. S refers to the syndrome input which causes the syndrome state transition.

For this example there exists no syndrome state $\underline{\sigma}_i$ for which there exists an N such that every syndrome state reachable from $\underline{\sigma}_o = \underline{0}$ (including $\underline{\sigma}_i$) can be driven to $\underline{\sigma}_i$ in exactly N steps. (It is assumed that

$p \neq 0$, in which case $\underline{\sigma}_o$, $\underline{\sigma}_1$ and $\underline{\sigma}_2$ are all reachable from $\underline{\sigma}_o$). Clearly, therefore, there exists no decoder state $q_i \epsilon Q_{RO}$ to which every state ϵQ_{RO} can be driven in exactly N steps, and thus no power of the associated Markov matrix π has a nonzero column. Thus from Theorem 1 the conclusion is made that steady-state state occupancy probabilities do not exist for this decoder.

The existence of P_{FD} for this example must still be investigated, since the existence of $\underline{W}$ is merely a sufficient condition for the existence of P_{FD}.

The cyclic operation of the syndrome register is begun at time u^*, where u^* is the time instant such that $S(i) = 0$, $i = 0,1,\ldots,u^*-1$, and $S(u^*) = 1$.

Now the quantity $Pr(u^* \geq \alpha)$ is conservatively bounded as $Pr(u^* \geq \alpha) \leq (1-p)^{\alpha}$ $\alpha = 0,1,2,---,0<p<\frac{1}{2}$, which implies that $\lim_{\alpha\to\infty} Pr(u^* \geq \alpha) = 0$, and u^* may be assumed to be finite.

For $u \geq u^* + 3$ and $\underline{\sigma}(u+2) = \underline{\sigma}_1 \overset{\Delta}{=} (10)$, $e^*(u) = S(u+2) = e(u) + e(u+2) + \xi(u+2)$ and $P_{FD}(u) = 2p(1-p)$. However, if $\underline{\sigma}(u+2) = \underline{\sigma}_2 \overset{\Delta}{=} (01)$ for $u \geq u^*+3$, then $e^*(u) = \overline{S(u+1)} = 1+e(u) + e(u+2) + \xi(u+2)$, and $P_{FD}(u) = p^2+(1-p)^2$. Therefore for $u \geq u^*+3$, $P_{FD}(u) = Pr(u^* \text{ is even})\ 2p(1-p) + Pr(u^* \text{ is odd})\ [p^2+(1-p)^2]$, u even and $P_{FD}(u) = Pr(u^* \text{ is even})\ [p^2+(1-p)^2] + Pr(u^* \text{ is odd}) \cdot 2p(1-p)$, u odd. Thus $\lim_{u\to\infty} P_{FD}(u) = P_{FD}$ if and only if $Pr(u^* \text{ is even}) = Pr(u^* \text{ is odd})$, which intuitively does not seem true for all p.

More specifically, for $p = 0.4$, $Pr(u^*=0) = 2p(1-p) = .48$ and $Pr(u^*=2) = 2p(1-p)^6 + 3p^3(1-p)^4 + 2p^5(1-p)^2 + p^7 = .0712$. Thus $Pr(u^* \text{ is even}) \geq Pr(u^*=0) + Pr(u^*=2) = .5512$, $Pr(u^* \text{ is even}) \neq Pr(u^* \text{ is odd})$, and P_{FD} does not exist for this example for $p = 0.4$.

However, in the limit as $p\to 0$, terms of order p^2 may be neglected and thus $Pr(u^*=\alpha) = 2p(1-2p)^{\alpha}$. Therefore $Pr(u^* \text{is even}) = \sum_{i=0}^{\infty} 2p(1-2p)^{2i} = \frac{2p}{1-(1-2p)^2} = \frac{2p}{1-1+4p-4p^2} = \frac{2p}{4p} = \frac{1}{2} = Pr(u^* \text{is odd})$, and

$$P_{FD} = \frac{1}{2}[2p(1-p)] + \frac{1}{2}[p^2 + (1-p)^2] = \frac{1}{2}.$$

Thus in the limit as $p \to 0$, P_{FD} exists even though $\lim_{u \to \infty} \underline{W}(u)$ does not.(For $p = 0$, however, $\lim_{u \to \infty} \underline{W}(u) = \underline{W} = [100---0]$ and $P_{FD} = 0$ since the decoder never leaves the all-zero state).

IV. EXISTENCE OF STEADY-STATE STATE OCCUPANCY PROBABILITIES FOR CERTAIN CLASSES OF DECODERS

Definition 1. A "Quasi-Maximum Likelihood Decoder" (QMLD) is a feedback decoder for a binary, rate $R = \frac{K_o}{N_o}$, systematic convolutional code with memory order m which operates in the following manner:

Conceptually, the QMLD first determines (one of) the most likely error pattern $\underline{\hat{E}}(u+m)$ consistent with the syndrome state and input at time u+m, assuming past decisions have been correctly made, and estimates $\underline{e}^*_I(u)$ as the first K_o components of the first subblock of $\underline{\hat{E}}(u+m)$. Ties among the most probable error patterns are resolved on the basis of the fewest number of errors in the leading positions of $\underline{\hat{E}}(u+m)$.

More precisely, let $\underline{Y}(u+m) \triangleq (\underline{\sigma}(u+m) : \underline{S}(u+m))$ be the $(N_o-K_o)(m+1)$ dimensional column vector representing the syndrome state and input at time u+m. Also define $E[\underline{Y}]$ as the set of error vectors consistent with $\underline{Y}$; i.e., $E[\underline{Y}(u+m)]$ is the set of error patterns $\underline{E}_i(u+m)$ which give $\underline{Y}(u+m) \triangleq (\underline{\sigma}(u+m) : \underline{S}(u+m))$ given that $\underline{e}^*_I(j) = \underline{e}_I(j)$, j = u-1, u-2,---, u-m. In other words $\underline{E}_i(u+m) \in E[\underline{Y}(u+m)] \Longleftrightarrow \underline{Y}(u+m) = H\underline{E}_i(u+m)$, where

$$H = \begin{bmatrix} G'_0 & "0" & "0" & \cdots & "0" \\ G'_1 & G'_0 & & & \vdots \\ \vdots & \ddots & \ddots & & \vdots \\ G'_{m-1} & & \ddots & \ddots & "0" \\ G'_m & G'_{m-1} & \cdots & G'_1 & G'_0 \end{bmatrix}, \quad (18)$$

the G'_i's are defined as in the previous section, "0" is the $(N_o-K_o)XN_o$ all zero matrix, and $\underline{E}_i(u+m) \triangleq (\underline{e}_i(u) : \underline{e}_i(u+1) : \text{---} : \underline{e}_i(u+m))$, with $\underline{e}_i(j) \triangleq (e_i^{(1)}(j), e_i^{(2)}(j), \text{---}, e_i^{(N_o)}(j))$, is any $(m+1)N_o$ dimensional error vector.

Since H is an $(m+1)(N_o-K_o) \times (m+1)N_o$ matrix with rank equal to $(m+1)(N_o-K_o)$, it has a null space of dimension $K_o(m+1)$, and hence $\#E[\underline{Y}(u+m)] = 2^{(m+1)K_o}$ for every $\underline{Y}(u+m)$. Note though that $\underline{E}(u+m)$, the actual noise vector at time u+m, may not be an element of $E[\underline{Y}(u+m)]$ if the previous m error estimates $\underline{e}^*_I(u-1),---,\underline{e}^*_I(u-m)$ have not all been made correctly.)

Now let $\hat{\underline{E}}(u+m) \triangleq (\hat{\underline{e}}_{u+m}(u) : \hat{\underline{e}}_{u+m}(u+1):---:\hat{\underline{e}}_{u+m}(u+m))$ with $\hat{\underline{e}}_{u+m}(i) \triangleq (\hat{e}^{(1)}_{u+m}(i), \hat{e}^{(2)}_{u+m}(i),---,\hat{e}^{(N_o)}_{u+m}(i))$, be the minimum weight error pattern which is an element of $E[\underline{Y}(u+m)]$. In the event more than one such minimum weight error patterns exist, $\hat{\underline{E}}(u+m)$ is determined as follows. Given $\underline{E}_A(u+m)$ and $\underline{E}_B(u+m)$, $\underline{E}_A(u+m) \neq \underline{E}_B(u+m)$, as two minimum weight error patterns consistent with $\underline{Y}(u+m)$, let j be the smallest integer for which $\underline{e}_A(u+j) \neq \underline{e}_B(u+j)$, $j\varepsilon\{0,1,---,m\}$, and q the smallest integer for which $e_A^{(q)}(u+j) \neq e_B^{(q)}(u+j)$, $q\ \varepsilon\{1,2,---,N_o\}$. Describe the case $e_A^{(q)}(u+j) = 1$, and $e_B^{(q)}(u+j) = 0$ by $\underline{E}_A(u+m) > \underline{E}_B(u+m)$, and the case $e_A^{(q)}(u+j) = 0$ and $e_B^{(q)}(u+j) = 1$ by $\underline{E}_B(u+m) > \underline{E}_A(u+m)$. Thus if $\underline{E}_i(u+m)$, $i = 1,2,---,p$ are all minimum weight error patterns consistent with $\underline{Y}(u+m)$, $\hat{\underline{E}}(u+m) \triangleq \underline{E}_\ell(u+m)$, where $\underline{E}_\ell(u+m) > \underline{E}_j(u+m)$, $j = 1,2,---,p$, $j \neq \ell$.

Therefore, given $\underline{Y}(u+m)$, $\hat{\underline{E}}(u+m)$ as defined above is determined by the decoder, and the desired estimate $\underline{e}^*_I(u)$ chosen as $\underline{e}^*_I(u) \triangleq (\hat{e}^{(1)}_{u+m}(u), \hat{e}^{(2)}_{u+m}(u),---,\hat{e}^{(K_o)}_{u+m}(u))$. This completes the description of the QMLD.

A QMLD correctly decodes in the absence of channel errors, since $\hat{\underline{E}}(u+m) = \underline{0}$ for $\underline{Y}(u+m) = \underline{0}$.

As an example of a QMLD, consider the decoder of Fig. 3. Here

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

and $\underline{Y}(u+1) = \begin{bmatrix} \sigma_o(u+1) \\ s(u+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \hat{e}^{(1)}_{u+1}(u) \\ \hat{e}^{(2)}_{u+1}(u) \\ \hat{e}^{(1)}_{u+1}(u+1) \\ \hat{e}^{(2)}_{u+1}(u+1) \end{bmatrix}$.

For $\underline{Y}(u+1) = (00)$, $E[(00)] = \{(0000), (1110), (1101), (0011)\}$,

$\hat{\underline{E}}(u+1) = (0000)$, and $e^*(u) = \hat{e}^{(1)}_{u+1}(u) = 0$;

for $\underline{Y}(u+1) = (01)$, $E[(01)] = \{(0010), (0001), (1100), (1011)\}$,

$\hat{\underline{E}}(u+1) = (0010)$, and $e^*(u) = \hat{e}^{(1)}_{u+1}(u) = 0$;

(Note here that two consistent error patterns are of minimum weight, namely (0010) and (0001). However (0010) > (0001) by the ordering relation ">" defined above, and thus $\hat{\underline{E}}(u+1) = (0010)$.)

for $\underline{Y}(u+m) = (10)$, $E[(10)] = \{(1010), (1001), (0100), (0111)\}$,

$\hat{\underline{E}}(u+1) = (0100)$, and $e^*(u) = \hat{e}^{(1)}_{u+1}(u) = 0$; and for $\underline{Y}(u+1) = (11)$,

$E[(11)] = \{(1000), (1011), (0110), (0101)\}$,

$\hat{\underline{E}}(u+1) = (1000)$, and $e^*(u) = \hat{e}^{(1)}_{u+1}(u) = 1$.

Contrasted to the QMLD, a "Maximum Likelihood Decoder" (MLD) is a feedback decoder which operates as follows. Given $\underline{Y}(u+m)$, let $e_I^{B_j}[\underline{Y}(u+m)]$ be defined as the subset of $E[\underline{Y}(u+m)]$ for which the vectors have $\underline{e}_I^{B_j}(u)$ as the first K_o components, $j = 1,2,---,t \leq 2^{K_o}$. Now $Pr(e_I^{B_j}[\underline{Y}(u+m)])$, the probability of the subset $e_I^{B_j}[\underline{Y}(u+m)]$, is defined as

$$Pr(e_I^{B_j}[\underline{Y}(u+m)]) = \sum_{\underline{E}_i(u+m) \in e_I^{B_j}[\underline{Y}(u+m)]} Pr(\underline{E}_i(u+m) \qquad (19)$$

where Σ is over all $\underline{E}_i(u+m) \in e_I^{B_j}[\underline{Y}(u+m)]$ and

where $$Pr(\underline{E}_i(u+m)) = p^{W(\underline{E}_i(u+m))}(1-p)^{N_o(m+1) - W(\underline{E}_i(u+m))} \qquad (20)$$

is the apriori probability of receiving $\underline{E}_i(u+m)$, $W(\underline{E}_i(u+m))$ is the (Hamming) weight of $\underline{E}_i(u+m)$, and p is the transition probability of the binary symmetric channel. Now $\underline{e}^*_I(u)$, the estimate corresponding to $\underline{Y}(u+m)$, is chosen to be an $\underline{e}_I^{B_i}(u)$ for which

$$Pr(e_I^{B_i}[\underline{Y}(u+m)]) \geq Pr(e_I^{B_j}[\underline{Y}(u+m)]), \quad j = 1, 2, ---, t$$
$$i \epsilon \{1, 2, --- , t\}.$$

Thus an MLD chooses $\underline{e}_I^*(u)$ as (one of) the most likely $\underline{e}_I^{B_j}(u)$('s) corresponding to $\underline{Y}(u+m)$, while a QMLD determines (one of) the most likely error vector(s) consistent with $\underline{Y}(u+m)$, i.e., $\hat{\underline{E}}(u+m)$, and chooses $\underline{e}_I^*(u)$ as the first K_o components of $\hat{\underline{E}}(u+m)$.

Note that for the case of only one minimum weight error pattern consistent with $\underline{Y}(u+m)$, $\underline{e}_I^*(u)$ is the same for both an MLD and a QMLD for sufficiently small p.

It is also interesting to note that the decoder of Fig. 3 is an MLD as well as a QMLD for all $p < \frac{1}{2}$.

It should be pointed out that the decoding algorithm of an MLD is a true maximum likelihood decoding algorithm for a genie decoder since the algorithm is predicated upon perfect removal of $\underline{e}(u-1)$,---,$\underline{e}(u-m)$ from $\underline{\sigma}(u+m)$.

That an MLD may not always put out the true maximum likelihood estimate in the feedback decoding mode because of past decoding errors is demonstrated by the decoder of Fig. 3. For this decoder let p^* be defined as the value of the BSC transition probability such that

$$\begin{aligned} P_{FD} &< p \quad , \quad 0<p<p^* \\ P_{FD} &= p \quad , \quad p = p^* \\ P_{FD} &> p \quad , \quad p^*<p<\tfrac{1}{2}, \end{aligned} \qquad (21)$$

(p^*, the real root of $1-5p+5p^2-2p^3 = 0$, satisfies $.25<p^*<.3$.) Now for $p<p^*$ it can be shown that the maximum likelihood error estimate conditioned on the syndrome state and input $\underline{Y}(u+1)$ is always put out by the MLD of Fig. 3 in steady-state. However for $p^*<p<\frac{1}{2}$, again in steady-state operation and with $f(\sigma_o(u+1), S(u+1)) = \sigma_o(u+1)\cdot S(u+1)$ as before, the most likely estimate is $e^*(u) = 0$ for any $\underline{Y}(u+1)$. However, since $f(11) = 1$ the MLD estimates $e^*(u) = 1$ whenever $\underline{Y}(u+1) = (11)$, so that this decoder does not always make the true maximum likelihood decision for this range of p.

To see this more precisely, let $\underline{Y}(u+1) = (\sigma_o(u+1), S(u+1)) = (11)$, and assume that the system is in steady-state. Now

$$\Pr(e(u) = 1/\sigma_o(u+1) = 1,\ S(u+1) = 1)$$

$$= \Pr(e(u)=1,\ \sigma_o(u+1) = 0/\sigma_o(u+1) = 1,\ S(u+1) = 1) +$$

$$+ \Pr(e(u) = 1,\ \sigma_o(u+1) = 1/\sigma_o(u+1) = 1,\ S(u+1) = 1)$$

$$= 0 + \Pr(e(u) = 1,\ \sigma_o(u+1) = 1/\sigma_o(u+1) = 1,\ S(u+1) = 1)$$

$$= \frac{\Pr(\sigma_o(u+1)=1, S(u+1)=1/e(u)=1, \sigma_o(u+1)=1)\ \Pr(e(u)=1, \sigma_o(u+1)=1)}{\Pr(\sigma_o(u+1) = 1,\ S(u+1) = 1)}$$

$$= \frac{[p^2 + (1-p)^2]\, P_{q_3}}{\Pr(\sigma_o(u+1) = 1, S(u+1) = 1)} = \frac{[p^2+(1-p)^2][p-3p^2+5p^3-2p^4]}{\Pr(\sigma_o(u+1) = 1, S(u+1) = 1)[1+3p^2-2p^3]}. \quad (22)$$

And similarly, $\Pr(e(u) = 0/\sigma_o(u+1)=1, S(u+1)=1) = \dfrac{2p(1-p)[p-3p^3+2p^4]}{\Pr(\sigma_o(u+1)=1, S(u+1)=1)[1+3p^2-2p^3]}$, (23

It can be readily shown that

$$\Pr(e(u) = 1/\sigma_o(u+1)=1, S(u+1)=1) > \Pr(e(u)=0/\sigma_o(u+1)=1, S(u+1)=1),\ p<p^*$$

$$\Pr(e(u)=1/\sigma_o(u+1)=1, S(u+1)=1) = \Pr(e(u)=0/\sigma_o(u+1)=1, S(u+1)=1),\ p=p^*$$

$$\Pr(e(u)=1/\sigma_o(u+1)=1, S(u+1)=1) < \Pr(e(u)=0/\sigma_o(u+1)=1, S(u+1)=1),\ p>p^*$$

The following definition is prompted by this discussion.

Definition 2: A "True Maximum Likelihood Decoder" (TMLD) for a systematic, $R = \frac{K_o}{N_o}$ convolutional code of memory order m is a decoder of the general form of Fig. 6 for which the decoding function $\underline{f}(\underline{\sigma}(u):\underline{S}(u))$ is such that P_{FD} (exists and) is a minimum.

It can be shown (by exhaustion) that for a decoder of the form of Fig. 3, $f(\sigma_o(u),S(u))$ for a TMLD is given by

$$f(\sigma_o(u),S(u)) = \sigma_o(u).S(u), \qquad 0 \le p < p^*$$

$$f(\sigma_o(u),S(u)) = \sigma_o(u).S(u) \text{ or } 0, \qquad p = p^*$$

$$f(\sigma_o(u),S(u)) = 0 , \qquad p^* < p \le \tfrac{1}{2}$$

in which case

$$P_{FD} = \frac{7p^2-12p^3+10p^4-4p^5}{1+3p^2-2p^3}, \qquad 0 \le p < p^*$$

$$P_{FD} = p, \qquad p^* \le p \le \tfrac{1}{2}. \tag{24}$$

For definite decoding or genie decoding, the probability distribution of $\underline{\sigma}(u)$ does not depend on the decoding function $\underline{f}$, and thus the determination of the $\underline{f}$ for which $Pr(\underline{e}_I^*(u) \neq \underline{e}_I(u))$ is a minimum is relatively straightforward.

However, for feedback decoding the probability distribution of $\underline{\sigma}(u)$ is a function of $\underline{f}$, and thus the determination of $\underline{f}$ for a TMLD is more involved. Moreover, even if the maximum likelihood estimate $\underline{e}_I^*(u)$ for each $\underline{Y}(u+m)$ is always put out in steady-state for a feedback decoder with a specified $\underline{f}$, this is not sufficient to insure that the decoder is a TMLD, as the following example shows.

For the decoder of the form of Fig. 3 but with $e^*(u) = f(\sigma_o(u+1),S(u+1))=0$, it can be shown that in steady-state the state distribution is such that $e^*(u) = 0$ is the most likely error estimate for any syndrome state and input for $\hat{p} \le p \le \frac{1}{2}$. $\hat{p}$, the real root of $1-6p+8p^2-4p^3 = 0$, satisfies $\hat{p} < p^*$.

However, as stated earlier, a feedback decoder of the form of Fig. 3 with $f(\sigma_o(u),S(u)) = \sigma_o(u).S(u)$ is a TMLD for $p < p^*$, which implies that the above feedback decoder with $f(\sigma_o(u),S(u)) = 0$ for $\hat{p} \le p < p^*$ is not a TMLD, even though the maximum likelihood error estimate is put out for any $\underline{Y}(u+1)$ in steady-state.

Intuitively, decoders such as the one above for which the maximum likelihood estimate is a function of p are more difficult to analyze than maximum likelihood decoders for block codes, for which the maximum likelihood decoding function is not a function of p for $0<p<\frac{1}{2}$.

<u>Lemma 1.</u> For a syndrome feedback decoder for an $R = \frac{K_o}{N_o}$, binary, systematic convolutional code of memory order m, $N \geq m$, where N is the smallest power of the Markov transition matrix π associated with the decoder for which there exists a non-zero column.

<u>Proof:</u> Assume $N<m$. Then there exists a decoder state $q_j \in Q_{RO}$ which is reachable from every state $q_i \in Q_{RO}$ in exactly $N<m$ steps. Let $\underline{E}_j^{(1)} \triangleq [e_{j,m}^{(1)},\text{---},e_{j,2}^{(1)},e_{j,1}^{(1)}]$ be the row vector denoting the contents of the first buffer register when the decoder is in state q_j. But any state $q_k \in Q_{RO}$ with $\underline{E}_k^{(1)} \triangleq [e_{k,m}^{(1)},\text{---},e_{k,1}^{(1)}] = [\overline{e_{j,1}^{(1)}}, \overline{e_{j,1}^{(1)}}, \text{---},\overline{e_{j,1}^{(1)}}]$, where $\overline{e_{j,1}^{(1)}}$ is the binary complement of $e_{j,1}^{(1)}$, cannot be driven to q_j in fewer than m steps, contradicting the assumption $N<m$. Hence $N \geq m$ and the lemma is proved.

<u>Theorem 3.</u> A QMLD for a systematic, rate $R = \frac{K_o}{N_o}$, binary convolutional code has steady-state state occupancy probabilities. Moreover N=m, where N is the smallest integer for which π^N has a positive column and π is the Markov transition matrix associated with the decoder.

<u>Proof:</u> From Lemma 1, $N \geq m$. Thus from Theorem 2 of the previous section, since a QMLD correctly decodes in the absence of channel errors, it is sufficient to demonstrate that any state of the syndrome register can be driven to the $\underline{\sigma}_o = \underline{0}$ state in a number of steps $N_{q_o} \leq m$. (The buffer registers can be cleared in at most m steps simultaneously.)

For $\hat{\underline{E}}(u+m) \triangleq (\hat{\underline{e}}_{u+m}(u) : \hat{\underline{e}}_{u+m}(u+1) :\text{---}: \hat{\underline{e}}_{u+m}(u+m))$

associated with $\underline{Y}(u+m)$, let

$$\theta\underline{\hat{E}}(u+m) \triangleq (\underline{\hat{e}}_{u+m}(u+1) : \underline{\hat{e}}_{u+m}(u+2):---:\underline{\hat{e}}_{u+m}(u+m) : \underline{0}), \qquad (25)$$

where $\underline{0}$ is the N_o dimensional all-zero column vector. Also let

$$\gamma\underline{\hat{E}}(u+m) \triangleq (\underline{\hat{e}}_{u+m-1}(u) : \underline{\hat{e}}_{u+m}(u): \underline{\hat{e}}_{u+m}(u+1) :---: \underline{\hat{e}}_{u+m}(u+m-1)), \qquad (26)$$

where $\underline{\hat{e}}_{u+m-1}(u)$ is the first block of $\underline{\hat{E}}(u+m-1)$, the estimated error pattern corresponding to $\underline{Y}(u+m-1)$ at time u+m-1. Note that

$$\theta\gamma\underline{\hat{E}}(u+m) \triangleq \theta(\gamma\underline{\hat{E}}(u+m)) = (\underline{\hat{e}}_{u+m}(u):---:\underline{\hat{e}}_{u+m}(u+m-1) : \underline{0}). \qquad (27)$$

Starting with $\underline{\sigma}(u+m)$ at time u+m, let the syndrome input vector $\underline{S}(u+m)$ be chosen such that $\underline{Y}(u+m) = H(\theta\underline{\hat{E}}(u+m-1))$, i.e., $\underline{S}(u+m) = G'\theta\underline{\hat{E}}(u+m-1)$, and assume that $\underline{\hat{E}}(u+m) \neq \theta\underline{\hat{E}}(u+m-1)$.

From the equations of operation of the syndrome (10), it can be shown that $\gamma\underline{\hat{E}}(u+m) \varepsilon E[\underline{Y}(u+m-1)]$; i.e., $\underline{Y}(u+m-1) = H(\gamma\underline{\hat{E}}(u+m))$. Now $W(\underline{\hat{E}}(u+m-1) \leq W(\gamma\underline{\hat{E}}(u+m)$ since $\underline{\hat{E}}(u+m-1)$ was the error estimate at time u+m-1. And since $\underline{\hat{E}}(u+m-1)$ and $\gamma\underline{\hat{E}}(u+m)$ have identical first blocks (by definition of the operator γ),

$$W(\theta\underline{\hat{E}}(u+m-1)) \leq W(\theta\gamma\,\underline{\hat{E}}(u+m)) \leq W(\underline{\hat{E}}(u+m)) \leq W(\theta\underline{\hat{E}}(u+m-1)) \qquad (28)$$

$$\Rightarrow W(\theta\underline{\hat{E}}(u+m-1)) = W(\theta\gamma\,\underline{\hat{E}}(u+m)) = W(\underline{\hat{E}}(u+m)). \qquad (29)$$

But

$$W(\theta\gamma\,\underline{\hat{E}}(u+m)) = W(\underline{\hat{E}}(u+m)) \Rightarrow \theta\gamma\underline{\hat{E}}(u+m) = \underline{\hat{E}}(u+m), \qquad (30)$$

and

$$W(\theta\underline{\hat{E}}(u+m-1)) = W(\theta\gamma\,\underline{\hat{E}}(u+m)) \Rightarrow W(\underline{\hat{E}}(u+m-1)) = W(\gamma\,\underline{\hat{E}}(u+m)). \qquad (31)$$

Now $W(\underline{\hat{E}}(u+m-1)) = W(\gamma\underline{\hat{E}}(u+m)) \Rightarrow$ the ordering relation ">" is defined between the two vectors, and $\underline{\hat{E}}(u+m-1) > \gamma\,\underline{\hat{E}}(u+m)$ since $\underline{\hat{E}}(u+m-1)$ was the estimate at time u+m-1. (Note that $\underline{\hat{E}}(u+m-1) = \gamma\,\underline{\hat{E}}(u+m)$ would contradict the assumption that $\underline{\hat{E}}(u+m) \neq \theta\underline{\hat{E}}(u+m-1)$, since $\theta\gamma\,\underline{\hat{E}}(u+m) = \underline{\hat{E}}(u+m)$.) But $\underline{\hat{E}}(u+m-1)$

$$> \gamma\,\underline{\hat{E}}(u+m) \Rightarrow \theta\underline{\hat{E}}(u+m-1) > \theta\gamma\,\underline{\hat{E}}(u+m) = \underline{\hat{E}}(u+m). \qquad (32)$$

However, this is impossible since $\underline{\hat{E}}(u+m)$ is by definition the estimated error pattern at time u+m. Thus the only possible conclusion is that

$\hat{\underline{E}}(u+m) = \theta \hat{\underline{E}}(u+m-1)$.

By continuing to choose new syndrome input vectors which are consistent with the θ shift of the previously estimated error pattern (i.e., $\underline{S}(u+m+i) = G'(\theta^{i+1}\hat{\underline{E}}(u+m-1)$, $i = 1, 2, ---, m)$, a $\underline{Y}(u+2m)$ can be obtained such that $\underline{Y}(u+2m) = H(\theta^{m+1}\hat{\underline{E}}(u+m-1))$, where $\theta^{m+1}\hat{\underline{E}}(u+m-1) \overset{\Delta}{=} \overbrace{\theta \cdots \theta}^{m+1} \hat{\underline{E}}(u+m-1)$.

But $\theta^{m+1}\hat{\underline{E}}(u+m-1) = \underline{0} \Rightarrow \underline{Y}(u+2m) = \underline{0} \Rightarrow \underline{\sigma}(u+2m) = \underline{0}$. (33)

Thus the syndrome state can be driven to $\underline{\sigma}_o = \underline{0}$ in $N_{q_o} \leq m$ steps and the theorem is proved.

Definition 3. For a syndrome feedback decoder for a rate $R = \frac{K_o}{N_o}$, systematic convolutional code with syndrome state and input at time u+m given by $\underline{Y}(u+m)$, the error pattern $\underline{E}_\alpha(u+m) \overset{\Delta}{=} (\underline{e}_\alpha(u) : \underline{e}_\alpha(u+1) :---: \underline{e}_\alpha(u+m))$ is said to be a "fully correctable" error pattern if

i) $\underline{e}_I^*(u) = \underline{e}_{\alpha I}(u)$, and

ii) when the subsequent syndrome inputs are chosen as

$$\underline{S}(u+m+i) = G'(\theta^i \underline{E}_\alpha(u+m)),$$

then $\underline{e}_I^*(u+i) = \underline{e}_{\alpha I}(u+i)$, $i = 1, 2, ---, m$.

Note that whether $\underline{E}_\alpha(u+m)$ is fully correctable or not depends on both the code and the decoding rule. Also note that $\underline{E}_\alpha(u+m)$ need not be $\underline{E}(u+m)$, the actual error pattern at time u+m. However, if $\underline{E}(u+m)$ is "fully correctable", then choosing $\underline{S}(u+m+i) = G'(\theta^i\underline{E}(u+m))$, $i = 1, 2, ---, m$, is equivalent to choosing $\underline{e}(u+m+j) = \underline{0}$, $j = 1, 2, ---, m$, at the decoder input.

For $\underline{Y}(u+m)$ consistent with a correctable error pattern $\underline{E}_\alpha(u+m)$, the syndrome may be driven to $\underline{\sigma}(u+2m+1) = \underline{0}$ by the choice of $\underline{S}(u+m+i) = G'(\theta^i\underline{E}_\alpha(u+m))$, $i = 1, 2, ---, m+1$, since for this case $\underline{Y}(u+2m+1) = H(\theta^{m+1}\underline{E}_\alpha(u+m)) = \underline{0}$. For the QMLD, every $\underline{Y}(u+m)$ is consistent with a "fully correctable" error pattern, namely $\hat{\underline{E}}(u+m)$, and for this reason the syndrome portion of the decoder is capable of being driven to $\underline{\sigma}(u+2m+1) = \underline{0}$.

With the above as motivation, the following theorem is stated.

Theorem 4. For a syndrome feedback decoder for an $R = \frac{K_o}{N_o}$ systematic convolutional code, which correctly decodes in the absence of channel errors, steady-state state occupancy probabilities exist if every syndrome state $\underline{\sigma}_i(u+m)$ reachable from $\underline{\sigma}(0) \triangleq \underline{0}$ can be driven to a state $\underline{\sigma}_j(u+m+\delta_i)$ for which there exists vectors $\underline{S}(u+m+\delta_i)$ and $\underline{E}_j(u+m+\delta_i)$ such that $\underline{E}_j(u+m+\delta_i)$ is a "fully correctable" error pattern relative to $\underline{Y}(u+m+\delta_i)$.

Proof: Given the arbitrary syndrome state $\underline{\sigma}_i(u+m)$, let $\underline{\sigma}_i(u+m)$ be driven to $\underline{\sigma}_j(u+m+\delta_i)$ by means of a suitable choice of inputs. Then let

$$\underline{S}(u+m+\delta_i+k) = H(\theta^k \underline{E}_j(u+m+\delta_i)),\ k = 1, 2, ---, m+1, \qquad (35)$$

be chosen as syndrome inputs in order to obtain $\underline{\sigma}(u+2m+1+\delta_i) = \underline{0}$.

Corollary 4.1 A syndrome feedback decoder for a systematic, $R = \frac{K_o}{N_o}$ convolutional code which correctly decodes any pattern of B or fewer errors over a $(m+1)N_o$ bit constraint length, and which puts out $\underline{e}_I^*(u) = \underline{0}$ as an estimate whenever $\underline{Y}(u+m)$ is consistent with no $\underline{E}_i(u+m)$ with $W(\underline{E}_i(u+m)) \leq B$, has steady-state state occupancy probabilities with $m \leq N \leq 2m+1$.

Proof: From Lemma 1, $m \leq N$. Now given $\underline{\sigma}(u+m)$, let $\underline{S}(u+m+j) = \underline{0}$, $j = 0, 1, ---, i$, be chosen as syndrome inputs until a $\underline{Y}(u+m+i)$, $i \in \{0, 1, ---, m\}$, is obtained for which there exists an $\underline{E}_\alpha(u+m+i)$ such that $\underline{E}_\alpha(u+m+i) \in E[\underline{Y}(u+m+i)]$ and $W(\underline{E}_\alpha(u+m+i)) \leq B$, or until m+1 such zero syndrome inputs have been fed in. If such a $\underline{Y}(u+m+i)$ exists, then from Theorem 4 the syndrome registers may be driven to $\underline{\sigma}(u+2m+1+i) = \underline{0}$ since $\underline{E}_{\alpha}(u+m+i)$ is a "fully correctable" error pattern. However, if no such $\underline{Y}(u+m+i)$ exists, then $\underline{\sigma}(u+2m) = \underline{0}$ since the syndrome registers will have been driven autonomously for m consecutive time units while feeding back the all-zero vector as an estimate.

For $B = \left\lfloor \frac{d_{FD}\min-1}{2} \right\rfloor$, where $d_{FD}\min$ is the feedback decoding minimum distance and $\lfloor\ \rfloor$ denotes the "greatest integer less than or equal to", the above corollary gives a class of minimum distance feedback decoders for which steady-state state occupancy probabilities exist.

V. APPLICATION OF STOCHASTIC SEQUENTIAL MACHINE THEORY TO SEMI-DEFINITE DECODING

The concept of semi-definite decoding (SDD) was introduced by Massey and investigated experimentally by Frasco [9] as a decoding technique intermediate to feedback and definite decoding.

A diagram of a general semi-definite decoder of order k for a binary $R = \frac{1}{2}$ systematic convolutional code with memory order m is shown at time u+m in Fig. 10. The first intermediate decision relevant to the estimation of e(u), $e^{*1}(u-k+1)$, is made on the basis of the unmodified syndrome bits S(u-k+m+1),---,S(u-k+1) and is thus equivalent to a definite decoding decision. $e^{*1}(u-k+1)$ is then used ("fed back") in order to modify the syndrome bits S(u-k+m+2),---,S(u-k+2) and the second intermediate decision $e^{*2}(u-k+2)$ formed on the basis of these modified syndrome digits by means of the same decoding function f. In general, the i^{th} intermediate decision $e^{*i}(u-k+i)$, $i \in \{1,2,---,k\}$, is made on the basis of S(u-k+m+i), ---, S(u-k+i) (with the same f) after appropriate modification by $e^{*i-1}(u-k+i-1)$,---,$e^{*i-m}(u-k+i-m)$ if $i > m$, or by $e^{*i-1}(u-k+i-1)$,---$e^{*1}(u-k+1)$ if $i \leq m$. The desired estimate of e(u) is then chosen as $[e^*(u)]_{SDD} \triangleq e^{*k}(u)$. Hence $[e^*(u)]_{SDD}$ is the k^{th} intermediate estimate formed at time u+m, and is seen to be a function of only a finite number of syndrome bits, namely S(u+m), S(u+m-1), ---,S(u-k+1). At the next instant of time, u+m+1, k new intermediate estimates $e^{*1}(u+1-k+1)$, $e^{*2}(u+1-k+2)$,---,$e^{*k}(u+1)$ are formed and the estimate $[e^*(u+1)]_{SDD} \triangleq e^{*k}(u+1)$ of e(u+1) thus made from S(u+m+1),-----,S(u-k+2) at time u+m+1 in exactly the same manner as was $[e^*(u)]_{SDD}$

from S(u+m),---,S(u-k+1) at time u+m.

More rigorously, in a semi-definite decoder of order k for a binary $R = \frac{K_o}{N_o}$ systematic convolutional code with memory order m, the estimate $[\underline{e}^*_I(u)]_{SDD}$ of $\underline{e}_I(u)$ is made at time u+m on the basis of $\underline{S}(u+m)$, $\underline{S}(u+m-1)$,---,$\underline{S}(u-k+1)$ in the following manner. Given the decoding function $\underline{f}$, the first intermediate decision $\underline{e}^{*1}_I(u-k+1)$ is made as

$$\underline{e}^{*1}_I(u-k+1) = \underline{f}\,(\underline{\sigma}_1(u+m-k+1) : \underline{S}(u+m-k+1)), \tag{36}$$

where

$$\underline{\sigma}_1(u+m-k+1) \triangleq (\underline{S}(u-k+1) : \underline{S}(u-k+2) :---: \underline{S}(u-k+m)). \tag{37}$$

The next k-1 intermediate decisions are then made as

$$\underline{e}^{*i}_I(u-k+i) = \underline{f}(\underline{\sigma}_i(u+m-k+i) : \underline{S}(u+m-k+i)), \tag{38}$$

where

$$\underline{\sigma}_i(u+m-k+i) = A\underline{\sigma}_{i-1}(u+m-k+i-1) + B\underline{S}(u+m-k+i-1) + G_{FB}\underline{e}^{*i-1}_I(u-k+i-1), \tag{39}$$

i = 2, 3,---,k, and where A, B, and G_{FB} are defined in equations (11), (12), and (13) in section III. The desired estimate $[\underline{e}^*_I(u)]_{SDD}$ is then chosen as

$$[\underline{e}^*_I(u)]_{SDD} \triangleq \underline{e}^{*k}_I(u). \tag{40}$$

This decoding method is similar to feedback decoding in that it utilizes some previous decoding decisions in arriving at an estimate. However, infinite error propagation is avoided as in definite decoding since each estimate is a function of only a finite number of syndrome bits, and hence a finite number of channel noise bits.

Let

$$P_{SDD}(u,k) \triangleq Pr([\underline{e}^*_I(u)]_{k\ \text{stage SDD}} \neq \underline{e}_I(u)) = Pr(\underline{e}^{*k}_I(u) \neq \underline{e}_I(u)) \tag{41}$$

and

$$P_{SDD}(k) \triangleq P_{SDD}(u,k)/u \geq k+m-1. \tag{42}$$

For $u \geq k+m-1$, the probability distribution of $(\underline{e}(u-m-k+1) :---: \underline{e}(u+m): :\underline{S}(u-k+1):---:\underline{S}(u+m))$, and thus $P_{SDD}(u,k)$, is independent of u and the second definition is therefore meaningful as the steady-state error probability for a k-stage semi-definite decoder; i.e.

$$P_{SDD}(k) = \lim_{u\to\infty} P_{SDD}(u,k). \tag{43}$$

Note that although this latter limit always exists, $\lim_{k\to\infty} \lim_{u\to\infty} P_{SDD}(u,k)$ may not exist.

For k = 1, the semi-definite decoder reduces to a definite decoder since

$$[\underline{e}_I^*(u)]_{1\text{stage SDD}} = \underline{e}_I^{*1} = \underline{f}(\underline{S}(u) :---: \underline{S}(u+m)) = [\underline{e}_I^*(u)]_{DD}. \tag{44}$$

Writing the equations of semi-definite decoding operation in recursion relation form yields

$$\begin{aligned}\underline{\sigma}_i(u+m-k+i) &= A^{i-1}\,\underline{\sigma}_1(u+m-k+1)\\ &+ \sum_{j=0}^{i-2} A^j[B\underline{S}(u+m-k+i-1-j) + G_{FB}\underline{f}(\underline{\sigma}_{i-1-j}(u+m-k+i-1-j) : \underline{S}(u+m-k+i-1-j))]\end{aligned}$$

$$i = 2,3,---,k. \tag{45}$$

These equations may also be written as

$$\begin{aligned}\underline{\sigma}_i(u+m-k+i) &= A^{i-\alpha}\underline{\sigma}_\alpha(u+m-k+\alpha)\\ &+ \sum_{j=0}^{i-1-\alpha} A^j[B\underline{S}(u+m-k+i-1-j) + G_{FB}\underline{f}(\underline{\sigma}_{i-1-j}(u+m-k+i-1-j):\underline{S}(u+m-k+i-1-j))]\end{aligned}$$

$$i = \alpha+1,\ \alpha+2,---,k,\ \alpha\ \varepsilon(1,2,---,k-1). \tag{46}$$

(For $\alpha = 1$, equations (46) reduce to equations (45).)

For the purpose of analyzing $P_{SDD}(u,k)$ it is appropriate to introduce the following definition.

Definition 4: For a semi-definite decoder for an $R = \frac{K_o}{N_o}$ systematic convolutional code with memory order m, the "equivalent feedback decoder" (EFD) is a syndrome feedback decoder for the same code with the same decoding function $\underline{f}$ as the semi-definite decoder.

With this definition, comparing the set of recursion equations (46) for a semi-definite decoder with a similar set for its corresponding "equivalent feedback decoder", namely

$$\underline{\sigma}(i) = A^{i-\alpha}\underline{\sigma}(\alpha) + \sum_{j=0}^{i-1-\alpha} A^{j}[B\underline{S}(i-1-j) + G_{FB}\underline{f}(\underline{\sigma}(i-1-j) : \underline{S}(i-1-j))]$$

$$i = \alpha+1, \quad \alpha+2, ---, \qquad (47)$$

it can be seen that if

$$\underline{\sigma}(u+m-k+\alpha) = \underline{\sigma}_{\alpha}(u+m-k+\alpha)$$

(where $\underline{\sigma}(i)$ refers to a state of the EFD and $\underline{\sigma}_i(u+m-k+i)$ an intermediate "state" of the semi-definite decoder), and if the same syndrome input vectors $\underline{S}(u+m-k+\alpha)$, $\underline{S}(u+m-k+\alpha+1)$,---,$\underline{S}(u+m)$ are input to both the semi-definite decoder and its "equivalent feedback decoder", then

$$\underline{\sigma}(u+m-k+\alpha+i) = \underline{\sigma}_{\alpha+i}(u+m-k+\alpha+i) \qquad (48)$$

and

$$[\underline{e}^{*}_{I}(u-k+\alpha+i)]_{EFD} = \underline{e}^{*\,\alpha+i}_{I}(u-k+\alpha+i) \qquad (49)$$

$$i = 0,\ 1,\ ---,\ k-\alpha,$$

which implies that $[\underline{e}^{*}_{I}(u)]_{EFD} = \underline{e}^{*k}_{I}(u) \overset{\Delta}{=} [\underline{e}^{*}_{I}(u)]_{SDD}$. (50)

If $\alpha = k-u-m$ with $k>u+m$ for a semi-definite decoder, then

$$\underline{\sigma}_{\alpha}(u+m-k+\alpha) = \underline{\sigma}_{\alpha}(0) = A^{\alpha-1}\underline{\sigma}_{1}(u+m-k+1)$$
$$+ \sum_{j=0}^{\alpha-2} A^{j}[B\underline{S}(u+m-k+i-1-j) + G_{FB}\underline{f}(\underline{\sigma}_{i-1-j}(u+m-k+i-1-j):\underline{S}(u+m-k+i-1-j))] \qquad (51)$$

where

$$\underline{\sigma}_{1}(u+m-k+1) \overset{\Delta}{=} (\underline{S}(u-k+1):---:\underline{S}(u-k+m)). \qquad (52)$$

Obviously $\underline{\sigma}_{\alpha}(0) = \underline{0}$, since $\underline{f}(\underline{0})=\underline{0}$ and $\underline{S}(B)=\underline{0}$ for $B<0$. Thus from the discussion in the previous paragraph, if the same syndrome vectors $\underline{S}(0)$, $\underline{S}(1)$,---,$\underline{S}(u+m)$ are input both to the semi-definite decoder and its "equivalent feedback decoder", and with $\underline{\sigma}(0) = \underline{0}$ for the EFD, $[\underline{e}_I^*(u)]_{EFD} = [\underline{e}_I^*(u)]_{SDD}$ for $k>u+m$, and for all such u the probability of error associated with a semi-definite decoder is equal to that of its "equivalent feedback decoder" under normal feedback decoding conditons (i.e., $\underline{\sigma}(0) = \underline{0}$). This may be stated mathematically as

$$\lim_{k\to\infty} P_{SDD}(u,k) = P_{FD}(u). \tag{53}$$

Note that $\lim_{u\to\infty} \lim_{k\to\infty} P_{SDD}(u,k) = \lim_{u\to\infty} P_{FD}(u)$ exists if and only if P_{FD} exists for the EFD.

Thus semi-definite decoding is seen to provide a spectrum of decoding techniques intermediate to definite deocding ($k=1$) and feedback decoding ($k\to\infty$ and $u<\infty$).

It is of more practical interest, however, to calculate the quantity $P_{SDD}(k) \triangleq P_{SDD}(u,k)/u\geq k+m-1 = \lim_{u\to\infty} P_{SDD}(u,k)$ which is determined from the probability distribution of the vector $(\underline{e}(u-m-k+1):--- :\underline{e}(u+m):\underline{S}(u-k+1):---:\underline{S}(u+m))$, $u\geq k+m-1$. The following lemma gives an expression for $P_{SDD}(k)$.

<u>Lemma 2.</u> $P_{SDD}(k)$, the "steady-state" probability of error associated with a semi-definite decoder of order k for a rate $R = \frac{K_o}{N_o}$, binary, systematic convolutional code of memory order m, is given by $P_{SDD}(k) = \Pr([\underline{e}_I^*(u)]_{EFD} \neq \underline{e}_I^*(u))$, $u\geq k+m-1$, where $[\underline{e}_I^*(u)]_{EFD}$ is the estimate formed by an "equivalent feedback decoder" at time u+m whose syndrome state is $\underline{\sigma}(u-k+m+1) \triangleq (\underline{S}(u-k+1):---:\underline{S}(u-k+m))$ at time u-k+m+1, and which is fed the syndrome inputs $\underline{S}(u-k+m+1),\underline{S}(u-k+m+2),---,\underline{S}(u+m)$ at times u-k+m+1,--- --,u+m respectively.

<u>Proof:</u> From the previous discussion in which the operation of a semi-definite decoder was compared to that of its "equivalent feedback decoder" (Eqs. (48), (49), and (50), it is clear that $[\underline{e}_I^*(u)]_{EFD} = [\underline{e}_I^*(u)]_{SDD}$,

and thus

$P_{SDD}(k) \triangleq Pr([\underline{e}_I^{*}(u)]_{k \text{ stage SDD}} \neq \underline{e}_I(u)) = Pr([\underline{e}_I^{*}(u)]_{EFD} \neq \underline{e}_I(u))$.

Equivalently, $P_{SDD}(k) = Pr([\underline{e}_I^{*}(u)]_{EFD} \neq \underline{e}_I^{*}(u))$, where $[\underline{e}_I^{*}(u)]_{EFD}$ is the estimate at time u+m of an "equivalent feedback decoder" which is in the decoder state given by $(\underline{e}_I(u-k+1):---:\underline{e}_I(u-k+m):\underline{S}(u-k+1):---:\underline{S}(u-k+m))$ at time u-k+m+1, and which is fed the error input vectors $\underline{e}(u-k+m+1)$,-----,$\underline{e}(u+m)$ at times u-k+m+1,---,u+m respectively.

It should be noted that each of the 2^{mN_o} possible values of $(\underline{e}_I(u-k+1):----:\underline{e}_I(u-k+m):\underline{S}(u-k+1):---:\underline{S}(u-k+m))$ occurs with non-zero probability, since each syndrome digit which is a part of the initial state is an additive function of a unique parity error digit. Thus all possible "initial" states must be considered for the "equivalent feedback decoder" in the calculation, even though all may not be reachable in normal feedback decoding operation (i.e., $\underline{\alpha}(0) = \underline{0}$).

The following formulation is helpful in calculating $P_{SDD}(k)$: Let Q be the set of the 2^{mN_o} possible states associated with the "equivalent feedback decoder" of a semi-definite decoder for a binary, $R = \frac{K_o}{N_o}$ systematic convolutional code with memory order m, and let Q be written as the union of its disconnected submachines. That is, $Q = Q_o \cup Q_1 \cup --- \cup Q_{n-1}$, where $Q_j = \{q_{j1}, q_{j2}, ---, q_{jr_j}\}$ and the q_{ij}'s represent the states of the decoder, $i = 1,2,---,r_j$, $j = 0, 1,---,n-1$. That Q_i and Q_j are disconnected, $i \neq j$, implies that q_{ik} cannot be driven into q_{jf}, $k = 1, 2, ---, r_i$, $f = 1, 2, ---, r_j$, $i,j = 0, 1, ---, n-1$, $i \neq j$.

Denote π_o, π_1, ---, π_{n-1}, as the Markov transition matrices associated with Q_o, Q_1,---,Q_{n-1} respectively. Also denote

$\underline{W}^{Q_j}(u) \triangleq [W_1^{Q_j}(u), W_2^{Q_j}(u), ---, W_{r_j}^{Q_j}(u)]$ as the state probability vector at time u, conditioned on the event that the "equivalent feedback decoder" began operation at time zero in a state $q_{jf} \epsilon Q_j$, and

$P_o(Q_j)$ as the probability of this conditioning event, $f = 1, 2, ---, r_j$, $j = 0, 1, ---, n-1$.

With this formulation, $P_{SDD}(k)$ may be written as

$$P_{SDD}(k) = \sum_{j=0}^{n-1} P_o(Q_j) \sum_{i-1}^{r_j} W_i^{Q_j}(k-1)\, P_{q_{ji}}, \tag{54}$$

where

$$\underline{W}^{Q_j}(k-1) = \underline{W}^{Q_j}(0)\, \pi_j^{k-1}, \tag{55}$$

$P_{q_{ji}}$ is the probability of error associated with state q_{ji},

$$P_o(Q_j) = Pr((\underline{e}_I(u-k+1):---:\underline{e}_I(u-k+m):\underline{S}(u-k+1):---:\underline{S}(u-k+m)) \varepsilon\, Q_j), \tag{56}$$

and

$$W_i^{Q_j}(0)=Pr((\underline{e}_I(u-k+1):---:\underline{S}(u-k+m))=q_{ji}/(\underline{e}_I(u-k+1):---:\underline{S}(u-k+m)) \,\varepsilon\, Q_j) \tag{57}$$

$\underline{u \geq} k+m-1$, $i = 1, 2, ---, r_j$, $j = 0, 1, ---, n-1$.

Note that if steady-state probabilities exist for $\pi_o, \pi_1, ---, \pi_{n-1}$, then $\lim_{k \to \infty} P_{SDD}(k) = \lim_{k \to \infty} \lim_{u \to \infty} P_{SDD}(u,k)$ exists and will be denoted $P_{SDD}(\infty)$; i.e., one can speak of the error probability of "long" semi-definite decoders.

As an example of a calculation of $P_{SDD}(k)$ consider the semi-definite decoding scheme for the $R = \frac{1}{2}$ systematic binary convolutional code with $m = 1$, $g_{0(1)}^{(2)} = 1$, $g_{1(1)}^{(2)} = 1$, for which the decoding function is the "exclusive or" gate; i.e., $f(\sigma_o(u), S(u)) = \sigma_o(u) \oplus S(u)$. The "equivalent feedback decoder" is shown in Fig. 11, where the switch in the feedback loop is closed at time u-k+2 after S(u-k+1) enters the syndrome register, and for which $P_{SDD}(k) = P_{FD}(u)$.

For this example, $Q=Q_o \cup Q_1, Q_0=\{q_{01},q_{02}\}, Q_1=\{q_{11},q_{12}\}$, $n=2, r_1=2, r_2=2$, $q_{01}=00, q_{02}=10, q_{11}=01, q_{12}=11$, where the first digit represents the content of the buffer register and the second the content of the syndrome register. The error probabilities associated with the states are readily calculated to be

$$
\begin{aligned}
Pq_{01} &= 2p(1-p) \\
Pq_{02} &= 2p(1-p) \\
Pq_{11} &= p^2+(1-p)^2 \\
Pq_{12} &= p^2+(1-p)^2,
\end{aligned}
\tag{58}
$$

$$P_0(Q_0) = Pr(S(u-k+1) = 0) = (1-p)^3 + 3p^2(1-p), \tag{59}$$

$$P_0(Q_1) = Pr(S(u-k+1) = 1) = 3p(1-p)^2 + p^3, \tag{60}$$

$$\pi_0 = \begin{bmatrix} 1-p & p \\ 1-p & p \end{bmatrix}, \qquad \pi_1 = \begin{bmatrix} 1-p & p \\ 1-p & p \end{bmatrix} \tag{61}$$

$$\underline{W}^{Q_0}(k-1) = \underline{W}^{Q_0}(0)\ \pi_0^{\ k-1} = [1-p,p], \quad k=2,3,4,---, \tag{62}$$

$$\underline{W}^{Q_1}(k-1) = \underline{W}^{Q_1}(0)\ \pi_1^{\ k-1} = [1-p,p], \quad k=2,3,4,---, \tag{63}$$

$$
\begin{aligned}
P_{SDD}(k) &= [(1-p)^3+3p^2(1-p)][(1-p)2p(1-p)+p\ 2p(1-p)] \\
&\quad + [3p(1-p)^2 + p^3][(1-p)\{p^2 + (1-p)^2\} + p\{p^2+(1-p)^2\}] \\
&= 5p - 20p^2 + 40p^3 - 40p^4 + 16p^5, \quad k=2,3,4,---,
\end{aligned}
\tag{64}
$$

and $P_{SDD}(1) = P_{DD} = 4p - 12p^2 + 16p^3 - 8p^4$.

On the other hand, when $P_{FD}(u)$ is calculated for this coding scheme for normal feedback decoding operation (i.e., the switch is closed for all time and $\underline{\sigma}(0) = \underline{0}$), only π_0 is considered since the decoder is initially in the all-zero state. (For this example, Q_σ, the set of states connected with the all-zero state, is equal to Q_{RO}, the set of states reachable from the all-zero state, although this is not true in general.)

The feedback decoding probability of error is then

$$P_{FD}(u) = (1-p)2p(1-p) + p2p(1-p) = 2p(1-p). \qquad (65)$$

Here no transient terms are present, and for all $u, P_{FD}(u) = P_{FD}$, the steady-state probability of error for feedback decoding.

Thus for this example, both $\lim_{k\to\infty} \lim_{u\to\infty} P_{SDD}(u,k)$ and $\lim_{u\to\infty} \lim_{k\to\infty} P_{SDD}(u,k)$ exist, and

$$\lim_{k\to\infty} \lim_{u\to\infty} P_{SDD}(u,k) > \lim_{u\to\infty} \lim_{k\to\infty} P_{SDD}(u,k) = P_{FD}, <0<p<\frac{1}{2}. \qquad (66)$$

The question arises as to the conditions under which $\lim_{k\to\infty} P_{SDD}(k) \triangleq$ $= \lim_{k\to\infty} \lim_{u\to\infty} P_{SDD}(u,k) = \lim_{u\to\infty} \lim_{k\to\infty} P_{SDD}(u,k) \triangleq \lim_{u\to\infty} P_{FD}(u) = P_{FD}$ for a semi-definite decoder, where $P_{FD}(u)$ refers to the probability of error of its "equivalent feedback decoder" under normal feedback decoding conditions (i.e., $\underline{\alpha}(0) = \underline{0}$). Sufficient conditions for this occurance are given by the following theorem.

Theorem 5. For a semi-definite decoder and its "equivalent feedback decoder" for an $R = \frac{K_o}{N_o}$ binary systematic convolutional code with memory order m, if steady-state probabilities exist for π, the Markov matrix associated with Q, then

$$\lim_{k\to\infty} P_{SDD}(k) \triangleq \lim_{k\to\infty} \lim_{u\to\infty} P_{SDD}(u,k) = \lim_{u\to\infty} \lim_{k\to\infty} P_{SDD}(u,k) \triangleq \lim_{u\to\infty} P_{FD}(u) = P_{FD} \qquad (67)$$

Proof: Since steady-state probabilities exist for π, $Q=Q_o$ and $P_{SDD}(k) = \sum_{i=1}^{2^{mN_o}} W_i^{Q_o}(k-1)\, P_{q_{oi}}$ with

$$\underline{W}^{Q_o}(k-1) \triangleq [W_1^{Q_o}(k-1), W_2^{Q_o}(k-1), ---, W_{2^{mN_o}}^{Q_o}(k-1)]. \text{ Now}$$

$$\lim_{k\to\infty} \underline{W}^{Q_o}(k-1) = \lim \underline{W}^{Q_o}(0)\, \pi_o^{\,k-1} = \underline{W}^{Q_o} \triangleq [W_1^{Q_o}, ---, W_{2^{mN_o}}^{Q_o}] \qquad (68)$$

independently of the choice of $\underline{W}^{Q_o}(0)$, since steady-state probabilities exist for $\pi_o = \pi$, and thus $\lim_{k\to\infty} P_{SDD}(k) = \sum_{i=1}^{2^{mN_o}} W_i^{Q_o} P_{q_{oi}}$. But since $\lim_{k\to\infty} \underline{W}^{Q_o}(0)\, \pi_o^{\,k-1} = \underline{W}^{Q_o}$ independently of the choice of the initial probability distribution $\underline{W}^{Q_o}(0)$, if any state q_{oi} has a non-zero steady-state probability (i.e., $W_i^{Q_o} \neq 0$), that state is reachable from every state $q_{oj} \in Q_o$. In particular, every state with non-zero steady-state probability is reachable from the all-zero state; or, conversely,

$$q_{oi} \notin Q_{RO} \Rightarrow W_i^{Q_o} = 0. \tag{69}$$

Thus

$$\lim_{k\to\infty} P_{SDD}(k) = \sum_{i=1}^{2^{mN_o}} W_i^{Q_o} Pq_{oi} = \sum_{i=1}^{r} W_i^{Q_{RO}} P_{q_{r_{oi}}} \triangleq P_{FD} \triangleq \lim_{u\to\infty} P_{FD}(u), \tag{70}$$

and the theorem is proved.

A code which meets the conditions of Theorem 5 is that of the example in Section II, with the decoding function f given by $f(\sigma_o(u),S(u)) = \sigma_o(u)\cdot S(u)$ for both the feedback and semi-definite decoding schemes. The example in this section with $f(\sigma_o(u),S(u)) = \sigma_o(u) \oplus S(u)$ does not meet the condition of the theorem since $Q \neq Q_\sigma$, and for this case $\lim_{k\to\infty} P_{SDD}(k) \neq P_{FD}$.

An important point which might be raised is that of the shape of the $P_{SDD}(k)$ vs. k curve. That is, is there an intermediate value of k, say k_α, for which $P_{SDD}(k_\alpha) < P_{FD}$ and $P_{SDD}(k_\alpha) < P_{DD}$; i.e., is semi-definite decoding ever strictly superior to definite decoding and feedback decoding?

An example which illustrates this eventuality is that of section II, with $f(\sigma_o(u),S(u)) = \sigma_o(u)\cdot S(u)$ and $p = 0.4$.

For this case

$$\pi_o = \pi = \begin{bmatrix} .36 & .24 & .16 & .24 \\ .6 & 0 & .4 & 0 \\ .24 & .36 & .24 & .16 \\ .6 & 0 & .4 & 0 \end{bmatrix} \tag{71}$$

and

$$\pi^{k-1} = \begin{bmatrix} .408 & .192 & .26 & .14 \\ .408 & .192 & .26 & .14 \\ .408 & .192 & .26 & .14 \\ .408 & .192 & .26 & .14 \end{bmatrix} + \begin{bmatrix} 0 & .4 & 0 & -.4 \\ 0 & .4 & 0 & -.4 \\ 0 & -.6 & 0 & .6 \\ 0 & -.6 & 0 & .6 \end{bmatrix} \delta(k-1) +$$

$$+(.0967)^{k-1} \begin{bmatrix} .416 & -.416 & -.387 & .387 \\ -.0187 & .0187 & .0176 & -.0176 \\ -.627 & +.627 & .587 & -.587 \\ -.0187 & .0187 & .0176 & -.0176 \end{bmatrix} + (-.495)^{k-1} \begin{bmatrix} .176 & -.176 & .128 & -.128 \\ -.39 & .39 & -.278 & .278 \\ .218 & -.218 & .153 & -.153 \\ -.39 & .39 & -.278 & .278 \end{bmatrix} \quad (72)$$

where

$$\delta(k-1) = \begin{matrix} 1 & k = 1 \\ 0 & k \neq 1 \end{matrix} \quad (73)$$

$$\underline{W}^{Q_o}(k-1) = \underline{W}^{Q_o}(0)\, \pi_o^{\,k-1} \quad (74)$$

$$\underline{W}^{Q_o}(0) = [.312\ .288\ .192\ .208] \quad (75)$$

and

$$P_{SDD}(k) = .48W_2^{\,Q_o}(k-1) + W_3^{\,Q_o}(k-1) + .48W_4^{\,Q_o}(k-1) \quad (76)$$

$$\Rightarrow P_{SDD}(k) = .419 + .0107\,(-.495)^{k-1} - .00076\,(.0967)^{k-1}. \quad (77)$$

Note that

$$\lim_{k\to\infty} P_{SDD}(k) = .419 = P_{FD}, \quad (78)$$

$$P_{SDD}(1) = .429 = P_{DD}, \quad (79)$$

and

$$\min_{\forall k} P_{SDD}(k) = P_{SDD}(2) = .414 \begin{matrix} < P_{FD} \\ < P_{DD} \end{matrix}. \quad (80)$$

For normal feedback decoding operation, $\underline{W}(0) = [1000]$ and

$$P_{FD}(u) = .48\,W_2(u+1) + W_3(u+1) + .48W_4(u+1) \quad (81)$$

$$= .419 + (.0967)^{u+1}(-.401) + (-.495)^{u+1}(-.018). \quad (82)$$

Note that

$$\lim_{u\to\infty} P_{FD}(u) = .419 = P_{FD}, \quad (83)$$

$$P_{FD}(-1) = 0, \quad (84)$$

and

$$P_{FD}(0) = .389 = P_{GD}. \quad (85)$$

$P_{SDD}(k)$ for the same code and decoder but with $p = .015$ was also calculated and found to be

$$P_{SDD}(k) = 1.54 \times 10^{-3} + 8.95 \times 10^{-4}(.1121)^{k-1} - 6.92 \times 10^{-4}(-.1271)^{k-1}. \qquad (86)$$

For this case

$$\min_{k} P_{SDD}(k) = P_{SDD}(5) \begin{array}{l} < P_{FD} \\ < P_{DD} \end{array}, \qquad (87)$$

although the difference $(P_{FD} - P_{SDD}(5))$ is negligible for all practical purposes.

The fact that it is possible for $P_{SDD}(k_\alpha)$ to be less than both P_{DD} and P_{FD} indicate that semi-definite decoding might be used to practical advantage. Moreover, semi-definite decoding has the further psychological advantage that infinite error propagation is impossible.

VI. BOUNDS ON P_{FD} FOR SPECIAL CASES

For the example of Chapter II it was shown that $P_{GD} < P_{FD} < P_{DD}$ for $0 < p < \frac{1}{2}$. This relationship between P_{GD}, P_{FD}, and P_{DD} does not hold in general, however, as the following two hypothetical examples illustrate.

Example 1: For the decoder of the form of Fig. 3 but with $f(\sigma_0(u), S(u)) = \sigma_0(u) \oplus S(u)$, $P_{FD} \approx 2p$ and $P_{GD} \approx 4p$ for p small, indicating that for this case $P_{GD} > P_{FD}$. In this case, the decoding function is so "bad" that accurate feedback decisions are deleterious.

Example 2: For the decoder of the form of Fig. 3 but with $f(\sigma_0(u), S(u)) = \sigma_0(u)$, $P_{FD} = \frac{1}{2}$ and $P_{DD} = 2p(1-p)$, indicating that $P_{FD} > P_{DD}$ for $p < \frac{1}{2}$.

Obviously, therefore, relationships of the form $P_{GD} < P_{FD}$ and $P_{FD} < P_{DD}$ do not hold in general and can be verified only for certain classes of decoders. In order to prove one such "special case" relationship use is made of the Theorem of Irrelevance [10] of Wozencraft and Jacobs.

<u>Theorem of Irrelevance</u>: For the communication system depicted in Fig. 12, where $\{\underline{S}_i\}$ is a finite signal set, $\underline{S} \in \{\underline{S}_i\}$, $\underline{r}_1$ and $\underline{r}_2$ are vector outputs from the channel, and $\hat{\underline{S}}$ is the estimate of $\underline{S}$ made by the receiver, the optimum receiver may ignore $\underline{r}_2$ in forming $\hat{\underline{S}}$ if and only if $p_{\underline{r}_2/\underline{r}_1,\underline{S}} = p_{\underline{r}_2/\underline{r}_1}$.

An optimum receiver, of course, is one for which $Pr(\hat{\underline{S}} = \underline{S})$ is a maximum.

In relating the signaling scheme of Fig. 12 to our decoding problem, let us assume that for the above scheme the signal $\underline{S} \triangleq \underline{e}_I(u)$, and $\underline{r}_1$ and $\underline{r}_2$ are the $(u+m+1)(N_o-K_o)$ dimensional column vectors given by

$$\underline{r}_1 \triangleq \begin{bmatrix} G_o' & \text{"0"} & \cdots & \cdots & \text{"0"} \\ G_1' & G_o' & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ G_m' & & \ddots & \ddots & \text{"0"} \\ \text{"0"} & \ddots & & \ddots & \vdots \\ \vdots & & \ddots & & \vdots \\ \text{"0"} & \cdots \text{"0"} & G_m' & \cdots \; G_1' & G_o' \end{bmatrix} \begin{bmatrix} \underline{0} \\ \underline{0} \\ \vdots \\ \underline{0} \\ \underline{e}(u) \\ \underline{e}(u+1) \\ \vdots \\ \underline{e}(u+m) \end{bmatrix}, \qquad (88)$$

$$\underline{r}_2 \triangleq \begin{bmatrix} G_o' & \text{"0"} & \cdots & \cdots & \text{"0"} \\ G_1' & G_o' & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ G_m' & & \ddots & \ddots & \text{"0"} \\ \text{"0"} & \ddots & & \ddots & \vdots \\ \vdots & & \ddots & & \vdots \\ \text{"0"} & \cdots \text{"0"} & G_m' & \cdots \; G_1' & G_o' \end{bmatrix} \begin{bmatrix} \underline{e}(0) \\ \underline{e}(1) \\ \vdots \\ \underline{e}(u-1) \\ \underline{0} \\ \vdots \\ \underline{0} \end{bmatrix}, \qquad (89)$$

where "0" and the G_i''s are as defined in Chapter III and $\underline{0}$ is the N_o-dimensional all-zero column vector.

Now in forming the optimum estimate of $\underline{e}_I(u)$, i.e., $\underline{e}_I^*(u)_{OPT}$, the vector $\underline{r}_2$ may be ignored on the basis of the Theorem of Irrelevance since $p_{\underline{r}_2/\underline{r}_1,\underline{S}} = p_{\underline{r}_2/\underline{r}_1}$. And thus $\underline{e}_I^*(u)_{OPT}$ is that vector of maximum likelihood conditioned only upon reception of $\underline{r}_1$.

Definition 5: A Maximum Likelihood Genie Decoder (MLGD) for a binary, rate $R = \frac{K_o}{N_o}$, systematic convolutional code of memory order m is a decoder for which the decoding function $\underline{f}$ is the same as that of the MLD for the same code (The ML was defined in Chapter IV as a "feedback" decoder.) but which operates in the "genie-decoding" mode.

With this definition and with the observation that the first $u(N_o-K_o)$ components of $\underline{r}_1$ above are always zero and that the last $(m+1)(N_o-K_o)$ components correspond to the $\underline{Y}(u+m)$ which would be obtained at time u+m for genie decoding, it is clear that

$$\Pr(\underline{e}_I^*(u)_{OPT} \neq \underline{e}_I(u)) = P_{MLGD}, \qquad (90)$$

where P_{MLGD} is the probability of error associated with an MLGD for the binary, rate $R = \frac{K_o}{N_o}$, systematic convolutional code specified by the G_i''s, $i = 0,1,---,m$. Therefore it is impossible for any receiver, on the basis of $\underline{r}_1$ and $\underline{r}_2$, to estimate $\underline{e}_I(u)$ with a probability of error less than P_{MLGD}.

In particular, any receiver (or decoder) which adds $\underline{r}_1$ and $\underline{r}_2$ in order to obtain $\underline{r}_1 \oplus \underline{r}_2 = (\underline{S}(0):\underline{S}(1):---:\underline{S}(u+m))$, as shown in Fig. 13, has an associated probability of error, with respect to the estimation of $\underline{e}_I(u)$, greater than or equal to P_{MLGD}. We state this result as a theorem.

Theorem 6. For a binary, rate $R = \frac{K_o}{N_o}$, systematic convolutional code of memory order m, any method of processing the entire syndrome vector $(\underline{S}(0):\underline{S}(1):---\underline{S}(u+m))$ in order to estimate

$\underline{e}_I(u)$ has an associated probability of error satisfying $Pr(\underline{e}_I^*(u) \neq \underline{e}_I(u)) \geq P_{MLGD}$, where P_{MLGD} is the probability of error of a maximum likelihood genie decoder for the same code. In particular, for any feedback, definite, or semi-definite decoding scheme,

$$P_{MLGD} \leq \begin{cases} P_{FD}(u) & \\ P_{DD}(u) & , u = 0,1,2,--- . \\ P_{SDD}(u,k) & , k = 1,2,--- \end{cases} \qquad (91)$$

With respect to Fig. 12, if $\underline{S} \triangleq \underline{e}_I(u)$ as before, but $\underline{r}_1 \triangleq (\underline{S}(u):\underline{S}(u+1):---:\underline{S}(u+m))$ and $\underline{r}_2 \triangleq (\underline{S}(0):\underline{S}(1):---:\underline{S}(u-1))$, then $p_{\underline{r}_2/\underline{r}_1,\underline{S}} \neq p_{\underline{r}_2/\underline{r}_1}$, and the vector $\underline{r}_2$ is not irrelevant to the optimum receiver. This is not surprising in view of the fact that if $\underline{r}_2$ were indeed irrelevant, the probability of error associated with semi-definite decoding, for $\underline{f}$ a maximum likelihood definite decoding function, would be minimum for k = 1, a fact known to be false from Chapter V.

We next prove a relationship involving P_{FD} and P_{DD} for a particular class of codes and range of channel transition probabilities. Several definitions and a lemma are first needed.

In feedback decoding, $\underline{i}(u)$ is decoded on the basis of the received vectors from time u to u+m under the assumption that the effects of $\underline{i}(u-m),---,\underline{i}(u-1)$ have been removed from these vectors by means of correct feedback. $\underline{i}(u) \triangleq$ $\triangleq [i^{(1)}(u),---,i^{(K_o)}(u)]$, where $i^{(j)}(u)$ is the j^{th} information digit entering the encoder at time u, $j = 1,2,---N_o$,

$u = 0,1,2,---$. Hence the following definition.

<u>Definition 6</u>: The <u>feedback decoding minimum distance</u> [11], d_{FD}, for a binary, rate $R = \frac{K_o}{N_o}$, systematic convolutional code of memory order m is defined as the minimum number of positions in which two encoded sequences with differing values of $\underline{i}(u)$ differ from time u to u+m, assuming that the effect of previous information digits has been removed.

Since the encoding process is linear, it follows that

$$d_{FD} = \min_{\substack{\underline{i}(u)\neq \underline{0} \\ \underline{i}(u-m)=---=\underline{i}(u-1)=\underline{0}}} W(\underline{T}(u)) \quad , \tag{92}$$

where

$$\underline{T}(u) \triangleq \left[\underline{i}(u-m) : --- : \underline{i}(u) : --- : \underline{i}(u+m)\right] \begin{bmatrix} \hat{G}_m & \text{"0"} & --- & \text{"0"} \\ | & \diagdown & \diagdown & | \\ \hat{G}_1 & & \diagdown & \text{"0"} \\ \hat{G}_o & \hat{G}_1 & --- & \hat{G}_m \\ \text{"0"} & \hat{G}_o & \diagdown & | \\ | & \diagdown & \diagdown & \hat{G}_1 \\ \text{"0"} & --- & \text{"0"} & \hat{G}_o \end{bmatrix} , \tag{93}$$

"0" is the $K_o X N_o$ all-zero matrix,

$\hat{G}_i \triangleq \left[I_{K_o} : G_i^T\right]$, $i = 0$

$\hat{G}_i \triangleq \left[0' : G_i^T\right]$, $i = 1,2,---,m$,

I_{K_o} is the $K_o X K_o$ identity matrix, $0'$ is the $K_o X K_o$ all-zero matrix, and G_i^T is the transpose of G_i, which is defined in Chapter III.

The feedback decoding minimum distance is well known to be equivalent to the minimum number of columns of the matrix H associated with the code (defined in Chapter IV), including at least one of the first K_o columns, which sum to the all-zero column [12] .

This definition is meaningful for feedback decoding in the sense that, with $C_E \triangleq \left\lfloor \frac{d_{FD}-1}{2} \right\rfloor$, where $\lfloor \quad \rfloor$ denotes the "largest integer less than or equal to" the enclosed quantity, two vectors $\underline{E}_i(u+m)$ and $\underline{E}_j(u+m)$ consistent with $\underline{Y}(u+m)$ and differing in their first K_o positions must satisfy $W(\underline{E}_i(u+m) \oplus \underline{E}_j(u+m)) \geq 2C_E+1$, since $\underline{E}_i(u+m) \oplus \underline{E}_j(u+m)$ is nonzero in at least one of its first K_o positions and is in the null space of H. Thus if $W(\underline{E}(u+m)) \leq C_E$ and $\underline{e}_I(u-m)$, ---, $\underline{e}_I(u-1)$ have all been estimated correctly, any other error vector consistent with $\underline{Y}(u+m)$ and differing from $\underline{E}(u+m)$ in the first K_o positions will have weight of at least C_E+1, and therefore $\underline{e}_I(u)$ will be estimated correctly by a decoder which chooses $\underline{e}^*_I(u)$ as the first K_o components of one of the minimum weight error vectors $\underline{E}_i(u+m)$ satisfying $\underline{Y}(u+m) = H\underline{E}_i(u+m)$.

In definite decoding, $\underline{i}(u)$ is decoded on the basis of the received digits from time u to u+m plus the received digits corresponding to information vectors $\underline{i}(u-m)$,--,$\underline{i}(u-1)$ which affect these digits. Hence $(m+1)N_o + mK_o$ digits, a number referred to as the <u>definite decoding constraint length</u>, affect the decoding of $\underline{i}(u)$, and we have the following definition.

<u>Definition 7</u>. For a binary, rate $R=\frac{K_o}{N_o}$, systematic convolutional code of memory order m, the <u>definite decoding minimum distance</u> [13] , d_{DD}, is defined as the minimum number of positions in which two encoded sequences with differing values of $\underline{i}(u)$ differ over a definite decoding constraint length.

Thus for definite decoding,

$$d_{DD} = \min_{\underline{i}(u)\neq\underline{0}} W(\underline{i}(u-m):\text{--}:\underline{i}(u-1):\underline{T}(u)) \tag{94}$$

The definite decoding minimum distance is well known [14] to be equivalent to the minimum number of columns of the matrix $\overline{H}$ associated with the code, including at least one of columns mN_o+1 through $mN+K_o$, which sum to the all-zero column, where

$$\overline{H} \triangleq \begin{bmatrix} G'_m & \text{—} & \text{—} & G'_1 & G'_o & \text{"0"} & \text{—} & \text{—} & \text{"0"} \\ \text{"0"} & \ddots & & \ddots & G'_1 & G'_o & \ddots & & \vdots \\ \vdots & & \ddots & & \ddots & \ddots & \ddots & & \text{"0"} \\ \text{"0"} & \text{—} & \text{—} & \text{"0"} & G'_m & \text{—} & \text{—} & G'_1 & G'_o \end{bmatrix}, \tag{95}$$

and "0" and the G'_i's are defined as in Chapter III.

This definition is analagous to that for d_{FD} in the sense that for $C_{\overline{E}} \triangleq \left\lfloor \frac{d_{DD}-1}{2} \right\rfloor$ and with $W(\underline{e}_I(u-m):\text{--}:\underline{e}_I(u-1):\underline{e}(u):\text{--}:\underline{e}(u+m)) \leq C_{\overline{E}}$, $\underline{e}_I(u)$ can always be correctly estimated by a definite decoder which decodes $\underline{e}_I(u)$ as the sub-block of digits of the error pattern with fewest "ones" consistent with the syndrome. ($\underline{e}_I(u-i)$, $i = 1,2,\text{--},m$, were used above in order to emphasize the fact that, since the code is in systematic form, the parity components of $\underline{e}(u-i)$, $i = 1,2,\text{--},m$, are not checked by $\overline{H}$ at time $u+m$.)

By the nature of the definition of d_{FD} and d_{DD}, it is readily seen that $d_{FD} \geq d_{DD}$ and thus $C_E \geq C_{\overline{E}}$ for a given code.

<u>Definition 8</u>. For a feedback decoder for a binary, rate $R = \frac{K_o}{N_o}$, systematic convolutional code of memory order m, associated with state $q_i \in Q_{RO}$, let L_i be the smallest

integer such that inputs of $\underline{e}(u) = \underline{0}$, $\underline{e}(u+1) = \underline{0}$,----, $\underline{e}(u+L_1-1) = \underline{0}$ result in $q(u+L_1) = q_o$, the all-zero decoder state, when $q(u) = q_1$. (If no such L_1 exists then $L_1 \triangleq \infty$.) Then L, the propagation length of the feedback decoder, is defined as $L \triangleq \max_{q_1 \in Q_{R0}} L_1$.

This definition of propagation length is similar in form to those of Sullivan [15] and Robinson [16].

As was mentioned in Chapter I, with feedback decoding it is possible for decoding errors to trigger further decoding errors, even in the absence of additional channel noise. In fact, for certain codes and decoders it is possible for a finite number of channel errors to cause an infinite number of decoding errors, a phenomenon referred to as catastrophic error propagation [17]. The significance of finite error propagation length L is that with $L<\infty$, catastrophic error propagation is impossible because L successive all-zero error inputs always return the decoder to the all-zero state.

Lemma 3. Let X and Y be discrete random variables taking on nonnegative integer values and let $Pr(X \leq u) \leq Pr(Y \leq u)$, $u = 0,1,2,$---. Then $\overline{X} \geq \overline{Y}$, where $\overline{X}$ and $\overline{Y}$ denote the expected values of X and Y respectively.

Proof: $$\overline{X} - \overline{Y} = \sum_{u=0}^{\infty} uPr(X=u) - \sum_{u=0}^{\infty} uPr(Y=u) \qquad (96)$$

$$= \sum_{u=0}^{\infty} u\left[Pr(X\leq u)-Pr(X\leq u-1)\right] - \sum_{u=0}^{\infty} u\left[Pr(Y\leq u)-Pr(Y\leq u-1)\right] \qquad (97)$$

$$= \sum_{u=0}^{\infty} u\left[Pr(X\leq u)-Pr(Y\leq u)\right] - \sum_{u=0}^{\infty} u\left[Pr(X\leq u-1)-Pr(Y\leq u-1)\right] \qquad (98)$$

$$= \sum_{u=0}^{\infty} u\left[Pr(X\leq u)-Pr(Y\leq u)\right] - \sum_{u=0}^{\infty} (u+1)\left[Pr(X\leq u)-Pr(Y\leq u)\right] \qquad (99)$$

$$= -\sum_{u=0}^{\infty} (Pr(X\leq u)-Pr(Y\leq u)) = \sum_{u=0}^{\infty} (Pr(Y\leq u)-Pr(X\leq u)). \qquad (100)$$

Therefore if $Pr(Y\leq u) \geq Pr(X\leq u)$, $u = 0,1,2,$--, then $\bar{X} - \bar{Y} \geq 0$, or equivalently, $\bar{X} \geq \bar{Y}$, and the lemma is proved.

Theorem 7. For a binary, rate $R = \frac{K_o}{N_o}$, systematic convolutional code of memory order m, if $C_E \triangleq \left\lfloor\frac{d_{FD}-1}{2}\right\rfloor > C_{\bar{E}} \triangleq \left\lfloor\frac{d_{DD}-1}{2}\right\rfloor$, then provided that $L<\infty$, $P_{FD} < P_{DD}$ in the limit as the channel transition probability p approaches zero, where P_{FD} is the probability of error of a feedback decoding scheme which corrects up to the guaranteed error correcting limit of the code (i.e., which correctly estimates $\underline{e}_I(u)$ at time u+m if $W(\underline{E}(u+m)) < C_E$ and if $\underline{e}_I^*(u-i) = \underline{e}_I(u-i)$, i = 1,2,---,m.), and P_{DD} is the probability of error associated with any definite decoding scheme for the code.

Proof: We first obtain an upper bound on P_{FD}.

For the code in question, let $\underline{E}(u) \triangleq (\underline{e}(u-m);---\underline{e}(u))$ as in Chapter II, A(u) be defined as the event that $W(\underline{E}(u)) > C_E$, and $Pr(A) \triangleq Pr(A(u)/_{u\geq m})$, since for $u \geq m$, Pr(A(u)) is independent of u. Since $\underline{e}(-m) \triangleq \underline{e}(-m+1) \triangleq --- \triangleq \underline{e}(-1) \triangleq \underline{0}$, it is readily seen that Pr(A(u)) < Pr(A), u = 0,1,---,m-1.

With Er(u) defined as the event that $\underline{e}_I^*(u) \neq \underline{e}_I(u)$ for the given feedback decoder, assuming of course that the decoder is in the all-zero state at time zero, application of the union bound yields

$Pr(Er(0)UEr(1)U---UEr(u-1)) \leq Pr(A(0)UA(1)U---UA(u-1)) \leq$

$$\leq \sum_{i=0}^{u-1} \Pr(A(i)) < u\Pr(A) \leq \frac{u}{\lfloor 1/\Pr(A) \rfloor}, \quad u = 1,2,\text{---},\lfloor 1/\Pr(A) \rfloor,$$

$$\text{and } \Pr(Er(0) \cup Er(1) \cup \text{---} Er(u-1)) \leq 1, \quad u > \lfloor 1/\Pr(A) \rfloor. \tag{101}$$

Thus if X is defined as the random variable representing the number of correct estimates made by the feedback decoder before the first incorrect estimate, given that the decoder is initially in the all-zero state, then

$$\Pr(X \leq u) \leq \frac{u+1}{\lfloor 1/\Pr(A) \rfloor}, \quad u = 0,1,\text{---},\lfloor 1/\Pr(A) \rfloor - 1,$$
$$\leq 1, \quad u \geq \lfloor 1/\Pr(A) \rfloor. \tag{102}$$

Now if Y is defined on the sample space of the non-negative integers as the random variable with probability distribution function given by

$$\Pr(Y \leq u) = \frac{u+1}{\lfloor 1/\Pr(A) \rfloor}, \quad u = 0,1,\text{---},\lfloor 1/\Pr(A) \rfloor - 1,$$
$$= 1, \quad u \geq \lfloor 1/\Pr(A) \rfloor, \tag{103}$$

$$\text{then } \overline{Y} = \sum_{i=0}^{\lfloor 1/\Pr(A) \rfloor - 1} i \Big/ \lfloor 1/\Pr(A) \rfloor = \frac{\lfloor 1/\Pr(A) \rfloor - 1}{2}. \tag{104}$$

Therefore since $\Pr(X \leq u) \leq \Pr(Y \leq u)$, $u = 0,1,2,\text{---}$, $\overline{Y} \leq \overline{X}$ from Lemma 3, and hence the expected number of consecutive correct estimates made by the feedback decoder is lower bounded by

$$\frac{\lfloor 1/\Pr(A) \rfloor - 1}{2} = \overline{Y} \leq \overline{X}. \tag{105}$$

For the feedback decoder , we next define random variable Z_1 as the smallest positive integer such that $q(u+Z_1) = q_o$, the all-zero state, given that $q(u) = q_i \in Q_{RO}$. We also define random variable W as the smallest integer greater than or equal to the propagation length L for which $\underline{e}(u+W-L) = \underline{e}(u+W-L+1) = \text{---} = \underline{e}(u+W-1) = \underline{0}$. Thus $q(u+W) = q_o$ by definition of L, and hence $Pr(Z_i \leq k) \geq Pr(W \leq k)$, $k = 1,2,3,\text{--}$, which implies from Lemma 3 that $\overline{W} \geq \overline{Z}_i$, $\forall i$. Therefore, given an arbitrary state of the feedback decoder, the expected number of additional time units until the all-zero decoder state is reached is less than or equal to the expected number of time units until the all-zero state is reached by means of L successive zero inputs, which is equal to L in the limit as $p \to 0$.

Hence, in the limit as $p \to 0$, an upper bound on P_{FD} may be obtained by assuming that the expected number of consecutive correct decisions made by the feedback decoder, initially in the all-zero state q_o, is equal to $\overline{Y}$, and that the expected number of additional time units needed to return to q_o after a decoding error is equal to $\overline{W} = L$. Therefore associating decoding errors with the L time units following the initial error, the bound

$$P_{FD} \leq \frac{L+1}{L+1+\overline{Y}} < 2(L+1)Pr(A) = 2(L+1)\binom{mN_o+N_o}{C_E+1} p^{C_E+1} \qquad (106)$$

is obtained.

Next a simple lower bound on P_{DD} is obtained. A definite decoder cannot correct <u>every</u> error pattern of weight $C_{\overline{E}} + 1$, which implies that for any definite decoding scheme

$$P_{DD} \geq p^{C_{\overline{E}} + 1} \qquad (107)$$

in the limit as $p \to 0$.

Thus, since $C_E > C_{\overline{E}}$ by assumption, in the limit as $p \to 0$,

$$P_{FD} < 2(L+1) \binom{mN_o+N_o}{C_E+1} p^{C_E+1} < p^{C_{\overline{E}}+1} \leq P_{DD}, \qquad (108)$$

and the theorem is proved.

An example which meets the conditions of the theorem is the binary, $R = \frac{1}{2}$, systematic convolutional code of memory order m=5, with $g_{0(1)}^{(2)} = 1$, $g_{1(1)}^{(2)} = 0$, $g_{2(1)}^{(2)} = 0$, $g_{3(1)}^{(2)} = 1$, $g_{4(1)}^{(2)} = 1$, $g_{5(1)}^{(2)} = 1$, and which is feedback decoded by means of "majority threshold decoding" [18]. Here $d_{FD} = 5$, $C_E = 2$, $d_{DD} = 4$, $C_{\overline{E}} = 1$, and $L \leq 13$.

The results of this report thus far, although given for codes over GF(2), can easily be generalized to codes over GF(q) if the binary symmetric channel is replaced by a memoryless, time invariant, additive GF(q) noise source. Generalizations can also be made to non-systematic codes and non-syndrome-type decoding, but the details shall not be carried out here.

VII. APPLICATION OF STOCHASTIC SEQUENTIAL MACHINE THEORY TO VITERBI DECODING

Thus far we have considered only decoders of the fixed span variety, that is, decoders which form $\underline{i}^*(u)$ at time u+m on the basis of the received vectors $\underline{r}(0),\text{--},\underline{r}(u),\text{--},\underline{r}(u+m)$, where $\underline{r}(i) \triangleq (r^{(1)}(i), r^{(2)}(i), \text{---}, r^{(N_o)}(i))$, and $r^{(j)}(i)$ is the digit received at time i from the j^{th} channel, $j = 1,2,\text{--},N_o$, $i = 0,1,2,\text{---}$. The point is that a fixed span decoder never looks more than a definite distance into the future.

Contrasted to the fixed span decoders are the variable span decoders, for which the particular received vectors utilized in forming the estimate $\underline{i}^*(u)$ are not specified apriori. In fact, a variable span decoder may look an indefinite distance into the future before deciding upon an estimate of $\underline{i}(u)$.

For conceptual simplicity, the discussion in this section will be restricted to binary, rate $R = \frac{1}{N_o}$, systematic convolutional codes of memory order m. Generalizations can easily be made, however, to arbitrary rates $R = \frac{K_o}{N_o}$.

The input-output characteristics of a convolutional encoder may be conveniently represented by means of a code tree [19], a concept important both to the operation and analysis of variable span decoders. An example is shown in Fig. 14 of a code tree for the encoder for the rate

$R = \frac{1}{3}$, systematic convolutional code of memory m=2 with $g_{o(1)}^{(2)} = 1$, $g_{o(1)}^{(3)} = 1$, $g_{1(1)}^{(2)} = 1$, $g_{1(1)}^{(3)} = 1$, $g_{2(1)}^{(2)} = 0$, and $g_{2(1)}^{(3)} = 1$.

With the encoder initially filled with zeroes, the information digits i(0), i(1), ---, i(L-1) uniquely specify the transmitted blocks $\underline{t}(0)$, $\underline{t}(1)$,---, $\underline{t}(L+m-1)$ at the decoder output, given that the L information digits are followed by a sequence of m zeroes. $\underline{t}(i) \triangleq (t^{(1)}(i),---t^{(N_o)}(i))$, where $t^{(j)}(i)$ is the binary digit transmitted across the j^{th} channel at time i, $j = 1,2,--,N_o$, $i = 0,1,--$.

The correspondence of the encoder to the code tree representation is as follows. The sequence of information digits i(0), i(1),---, i(L-1) input to the encoder also uniquely specifies a path through the code tree, with the upper branch stemming from a node at depth u into the tree being taken if i(u) = 1 and the lower branch if i(u) = 0, $0 \leq u \leq L-1$. There is no more branching of the tree past depth L-1 but merely a sequence of m branches appended to each of the nodes at this depth, corresponding to the m successive zeroes fed into the encoder.

Now with each branch emanating from a node at depth u into the tree is associated an N_o-tuple of digits; these are the transmitted digits which would appear at the encoder output at time u if the information digits i(0), i(1),--, i(u) specifying the path up to and including the particular branch were input to the encoder, given of course that the encoder is initially filled with zeroes.

Thus there exists a one to one correspondence between sequences of information digits input to the encoder and paths through the code tree. And for each sequence of information digits i(0), i(1),---, i(L-1), the transmitted blocks at the encoder output $\underline{t}(0)$, $\underline{t}(1)$,---, $\underline{t}(L+m-1)$ are exactly those blocks of N_o-tuples encountered along the path through the tree specified by i(0), i(1),---, i(L-1).

Since the class of decoders which will shortly be described utilizes a replica of the code tree in order to decode the received sequence, it is necessary for the sake of clarity to reserve the notation i(0), i(1), i(2),--- for the actual sequence of information digits, and to denote various paths through the code tree in another manner. Hence we introduce the following notation.

Let the vector (a(0), a(1),---, a(u-1)), $a_i \in GF(2)$, represent the u-branch segment of the code tree which would be specified in the encoding process by the sequence of information digits i(0)=a(0), i(1)=a(1),---, i(u-1)=a(u-1).

Forney [20] has observed that for a node of the tree specified by (a(0), a(1),--, a(u-1)), since the digits associated with any branch emanating from that node are functions only of a(u-m), a(u-m+1),---, a(u-1), a(u), the tree may be condensed into the form of a "trellis" by superimposing the nodes at depth u specified by identical values of $a(u-m),---,a(u-1)$. The "trellis" or "condensed tree" for the code tree of Fig. 14 is shown in Fig. 15. No loss is incurred in this representation, since there is a one to one correspondence between paths through the tree and paths through the trellis, and since the same sequence of transmitted digits is encountered along paths through each which result from the same sequence of information digits.

Note that a trellis has 2^u nodes at depth u, $0 \leq u \leq m$, 2^m nodes for $m \leq u \leq L$, and 2^{L+m-u} nodes at depth u for $L \leq u \leq L+m$.

Let $n_h(u)$ refer to the h^{th} node at depth u into the trellis. We use the convention that the m binary digit representation of h specifies the path leading to $n_h(u)$. For example, for m=3, since $(6)_{10} = (110)_2$, $n_6(u)$ is the node of the trellis at depth u uniquely specified by the branches $a(u-3) = 1$, $a(u-2) = 1$, and $a(u-1) = 0$.

We now focus attention upon a particular (variable span) decoding method introduced by Viterbi [21], who demonstrated that the ensemble of convolutional codes when decoded by this technique obeys the random coding bound with an exponent E(R) superior to that of block codes.

First we let $\underline{a}_h(u) \triangleq (a_{h,u}(0), a_{h,u}(1),---, a_{h,u}(u-1))$ be defined as the maximum likelihood path from the origin of the trellis to node $n_h(u)$, conditioned upon reception of $\underline{R}(u) \triangleq (\underline{r}(0):\underline{r}(1):---:\underline{r}(u-1))$, with ties being decided in some arbitrary but consistent manner. But since the binary symmetric channel is assumed here, the maximum likelihood path $\underline{a}_i(u)$ to each node $n_i(u)$ is merely that path for which $D_i(u)$ is a maximum, where $D_i(u) \triangleq W(\underline{a}_i(u) \oplus \underline{R}(u))$, and W refers to the Hamming weight of its vector argument.

A Viterbi decoder performs an L-step operation in decoding an L branch trellis. At the 1^{st} step, the decoder considers $\underline{R}(m+1)$ and determines the maximum likelihood path from the origin to each of the 2^m nodes at depth m+1 into the trellis, i.e., $\underline{a}_i(m+1)$, $i = 0,1,---,2^m-1$, and discards the remaining 2^m possible paths. For example, for m=3 and for node $n_6(4)$, the decoder determines which of the two paths specified by (a(0), a(1), a(2), a(3)) = (0110) and (a(0), a(1), a(2), a(3)) = (1110) is closest in Hamming distance to $\underline{R}(4)$, and eliminates the other path from further consideration. Thus in the first step a Viterbi decoder makes 2^m comparisons involving 2 paths each and keeps the 2^m winning or "surviving" paths along with their respective $D_i(m+1)$'s.

At step 2, the Viterbi decoder determines the 2^m maximum likelihood paths from the origin to the nodes at depth m+2, conditioned upon $\underline{R}(m+2)$. Note, however, that node $n_j(m+2)$, where $(j)_{10} = (f_{m-1},---,f_1,f_o)_2$, $f_i \epsilon \{0,1\}$, can be reached only from nodes $n_k(m+1)$ and $n_s(m+1)$ with $a(m+1) = f_o$, where $(k)_{10} = (0,f_{m-1},---,f_1)_2$ and $(s)_{10} =$ $= (1,f_{m-1},---,f_1)_2$. Thus $\underline{a}_j(m+2)$ must pass through either $n_k(m+1)$ and coincide with $\underline{a}_k(m+1)$ from $n_k(m+1)$ to the origin or it must pass through $n_s(m+1)$ and coincide with

$\underline{a}_s(m+1)$ from $n_s(m+1)$ to the origin, since $\underline{a}_k(m+1)$ and $\underline{a}_s(m+1)$ are by definition the minimum weight paths from their respective nodes to the origin of the trellis. Hence all that is needed in determining $\underline{a}_j(m+2)$ is knowledge of $D_k(m+1)$ and $D_s(m+1)$ (determined from the previous step), along with the number of disagreements with $\underline{r}(m+1)$ of the branch from $n_k(m+1)$ to $n_j(m+2)$ and of the branch from $n_s(m+1)$ to $n_j(m+2)$.

In short, the Viterbi decoder at step 2 considers the 2^{m+1} paths given by $(a_{i,m+1}(0),---,a_{i,m+1}(m),0)$, $i = 0,1,---,2^m-1$, and $(a_{j,m+1}(0),---,a_{j,m+1}(m),1)$, $j = 0,1,---,2^m-1$, and from knowledge of $D_i(m+1)$, $i = 0,1,--,2^m-1$ and $\underline{r}(m+1)$, determines the 2^m surviving paths $\underline{a}_j(m+2)$ and their respective distances $D_j(m+2)$, $j = 0,1,---,2^m-1$.

In general, at the k^{th} step, $2 \leq k \leq L-m$, the decoder constructs the 2^{m+1} paths $(a_{i,m+k-1}(0),---,a_{i,m+k-1}(m+k-2),0)$, $i = 0,1,--,2^m-1$, and $(a_{j,m+k-1}(0),---,a_{j,m+k-1}(m+k-2),1)$, $j = 0,1,---,2^m-1$, from the 2^m survivors at the $(k-1)^{st}$ step, namely $\underline{a}_j(m+k-1)$, $j = 0,1,---,2^m-1$. It then, utilizing $\underline{r}(m+k-1)$ and $D_i(m+k-1)$, $i = 0,1,---,2^m-1$, determines by means of 2^m comparisons involving 2 paths each, the 2^m survivors at the k^{th} step, $\underline{a}_j(m+k)$, along with their respective distances $D_j(m+k)$, $j = 0,1,---,2^m-1$.

At each step k, $2 \leq k \leq L-m$, the number of competitive paths is increased by a factor of 2 by consideration of the 2^m surviving paths at the previous step both with a "0" branch and with a "1" branch appended; this number is then decreased by a factor of 2 as the appropriate comparisons are made, resulting in 2^m survivors at the end of the k^{th} step.

However for step k, $L-m < k \leq L$, since branching ceases at depth L, only "0" branches are appended to the surviving paths and thus the net number of survivors is decreased by a factor of 2 by the comparisons at each step. The end result is one surviving path at depth L+m, namely $\underline{a}_o(L+m) \triangleq$ $\triangleq (a_{o,L+m}(0), a_{o,L+m}(1), ---, a_{o,L+m}(L+m-1))$, which is the desired (estimated) path through the trellis (or tree). The sequence of estimated information digits is then given as $i^*(0) = a_{o,L+m}(0)$, $i^*(1) = a_{o,L+m}(1)$, ---, $i^*(L-1) =$ $= a_{o,L+m}(L-1)$.

Now $\underline{a}_o(L+m)$ is the maximum likelihood path through the trellis conditioned upon the received sequence $\underline{R}(L+m)$, since $\underline{a}_i(u)$ is by definition the maximum likelihood path from the origin to node $n_i(u)$, for all appropriate values of u and i.

It is interesting to note that any two paths which are compared in the process agree in the last m positions, for otherwise they would not correspond to the same node.

For the trellis of Fig. 15 and for the received vector $\underline{R}(6) = (101\ 100\ 101\ 110\ 101\ 010)$, it can easily be shown that

$\underline{a}_3(3) = (111)$, $D_3(3) = 1$

$\underline{a}_2(3) = (110)$, $D_2(3) = 4$

$\underline{a}_1(3) = (001)$, $D_1(3) = 4$

$$\underline{a}_o(3) = (100), \quad D_o(3) = 5$$

$$\begin{aligned} \underline{a}_3(4) &= (1111), & D_3(4) &= 3 \\ \underline{a}_2(4) &= (1110), & D_2(4) &= 2 \\ \underline{a}_1(4) &= (1101), & D_1(4) &= 4 \\ \underline{a}_o(4) &= (1100), & D_o(4) &= 7 \end{aligned} \qquad (109)$$

$$\underline{a}_2(5) = (11110), \quad D_2(5) = 6$$
$$\underline{a}_o(5) = (11100), \quad D_o(5) = 3$$

$$\underline{a}_o(6) = (111000), \; D_o(6) = 4.$$

Thus $\underline{a}_o(6) = (111000)$ is the maximum likelihood path through the code tree conditioned upon reception of $\underline{R}(6)$, and the corresponding sequence of information estimates is $i^*(0) = 1$, $i^*(1) = 1$, $i^*(2) = 1$, and $i^*(3) = 0$.

Now let us consider the application of the Viterbi decoding technique to a "semi-infinite trellis", that is, a trellis which is not terminated by a sequence of m successive zeroes but one which corresponds to the encoding of an indefinite number of information digits.

Definition 9: For a semi-infinite trellis, a hard decision $i^*(u)$ of $i(u)$ is said to be made by time (or depth) f, $f > u$, if and only if $a_{i,f}(u) = i^*(u)$, $i = 0,1,---,2^m-1$.

In the above example with $\underline{R}(6) = (101\ 100\ 101\ 110\ 101\ 010)$ for the trellis of Fig. 15, hard decisions $i^*(0) = 1$ and $i^*(1) = 1$ resulted immediately after $\underline{a}_i(4)$, $i = 0,1,2,3$, were determined. $i^*(0) = 1$ was made since $a_{o,4}(0) = a_{1,4}(0) = a_{2,4}(0) = a_{3,4}(0) = 1$, and $i^*(1) = 1$ resulted since

$a_{0,4}(1) = a_{1,4}(1) = a_{2,4}(1) = a_{3,4}(1) = 1.$

Note that hard decisions are irrevocable by the nature of the decoding process. That is, if $a_{i,f}(u) = i^*(u)$, $i = 0,1,---,2^m-1$, then $a_{i,f'}(u) = i^*(u)$, $i = 0,1,---,2^m-1$, for $f'>f$. (It will later be shown that a hard decision on $i(u)$ is made with probability 1 as the depth of operation of the Viterbi decoder into the trellis approaches infinity.)

Now even though a Viterbi decoder which operates on a semi-infinite trellis chooses the path through the trellis of smallest distance from the received vector for $0 \leq p < \frac{1}{2}$, the probability that this decoded path is the actual transmitted path approaches zero as the distance into the trellis approaches infinity, since it is inevitable that some decoding error will eventually be made. Therefore we choose as our criterion of goodness $Pr(i^*(u) \neq i(u))$ for arbitrary u, and not the probability that the entire path hypothesized by the Viterbi decoder is actually the correct (or transmitted) path.

Note that the vector $\underline{D}'(u) \triangleq (D_0'(u), D_1'(u), --, D_{2^m-1}'(u))$, where $D_i'(u) \triangleq D_i(u) - \min_D(u)$, $i = 0,1,--,2^m-1$, and $\min_D(u) \triangleq \min_{i=0,1,--,2^m-1} D_i(u)$, provides an adequate description of the "state" of a Viterbi decoder at depth u into the trellis, with respect to the formation of $i^*(u-1)$, the "hard decision" of $i(u-1)$. For $\underline{D}'(u)$, in conjunction with *future* received vectors $\underline{r}(u)$, $\underline{r}(u+1)$, $\underline{r}(u+2)$,---, is sufficient to determine the subset of nodes of the trellis at depth u through which pass $\underline{a}_i(f)$, $i = 0,1,---,2^m-1$, $\forall f > u$. And if for some $f^* > u$, all $\underline{a}_i(f^*)$ pass through a subset of $\{n_o(u), n_2(u),---, n_{2m-1}(u)\}$, then $i^*(u-1) = 0$; and if all $\underline{a}_i(f^*)$ pass through a subset of $\{n_1(u), n_3(u),---, n_{2m-1}(u)\}$, then $i^*(u-1) = 1$. Therefore, with respect to the calculation of the probability of error per information digit, $\underline{D}'(u)$ will be taken as the appropriate state (or state vector) at time u, and will henceforth be called the *relative distance vector*.

Lemma 4. For a Viterbi decoder for a binary, rate $R = \frac{1}{N_o}$, systematic convolutional code of memory order m, the number of distinct relative distance vectors at any time u, $\#\underline{D}'(u)$, is bounded by

$$\#\underline{D}'(u) \leq 2^m(1+d_S\max)^{2^m-1}, \qquad (110)$$

where

$$d_S\max \triangleq \max_{i(u-m),---,i(u+m-1)} W(\underline{T}_S(u)), \qquad (111)$$

$$\underline{T}_S(u) \triangleq \left[i(u-m),--,i(u),--,i(u+m-1)\right] \begin{bmatrix} \underline{g}_m & \underline{0} & \cdots & \underline{0} \\ \vdots & \ddots & \ddots & \vdots \\ \underline{g}_1 & & \ddots & \underline{0} \\ \underline{g}_o & \underline{g}_1 & \cdots & \underline{g}_m \\ \underline{0} & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \underline{g}_1 \\ \underline{0} & \cdots & \underline{0} & \underline{g}_o \end{bmatrix} \qquad (112)$$

and $\underline{g}_i \triangleq \left[g_{i(1)}^{(1)}, ---, g_{i(1)}^{(N_0)}\right], \quad i = 0,1,---,m.$ (113)

In other words, we want to show that the paths retained by the Viterbi decoder are never too greatly different, i.e., that the distance of none from the received vector exceeds that of the closest by more than $d_S max$.

Proof: Let $n_i(u)$, $i \in \{0,1,---,2^m-1\}$, be a node at depth u into the trellis for which $D_i(u) = \min.\{D_0(u),---,D_{2^m-1}(u)\}$ and hence for which $D_i'(u) = 0$. We now ask what is the maximum possible value of $D_j'(u)$ for an arbitrary node $n_j(u)$ at depth u. It is readily seen that there exists a path from node $n_j(u)$ at most m steps back through the trellis which intersects $\underline{a}_i(u)$, the minimum weight path from the origin to node $n_i(u)$. Now the distance between the path segments to nodes $n_i(u)$ and $n_j(u)$ from the point of intersection of $\underline{a}_i(u)$ and the path back from node $n_j(u)$ can be at most $d_S max$, which implies that there exists a path from the origin to node $n_j(u)$ which differs from the received sequence in at most $D_i(u) + d_S max$ positions. Hence $D_j'(u) \le D_i'(u)+d_S max =$ $= d_S max$, and thus $D_j'(u)$ can assume at most $d_S max + 1$ distinct values, i.e., $0,1,2,---,d_S max$. Hence the bound $\#\underline{D}'(u) \le 2^m(1+d_S max)^{2^m-1}$ follows, with the factor 2^m corresponding to the specification of a node $n_i(u)$ for which $D_i'(u) = 0$, and the factor $(1+d_S max)$ corresponding to the maximum number of values each of the remaining 2^m-1 nodes may assume. This completes the proof of the lemma.

Note that the bound on the number of relative distance vectors grows greater than exponentially, i.e., approximately as exp(exp(m)), with encoder memory m.

Thus a Viterbi decoder, for the purpose of analyzing the probability of error per information digit, may be represented as a finite Markov chain, with the states of the chain being denoted as $\underline{D}'_o$, $\underline{D}'_1$,---, $\underline{D}'_{\#\underline{D}'-1}$, $\#\underline{D}' \leq \leq 2^m(1+d_s\max)^{2^m-1}$. The all zero transmitted sequence is assumed, and thus the transition probability P_{ij} from state $\underline{D}'_i$ to state $\underline{D}'_j$ is given as the probability of those error patterns $\underline{e}(u)$ that take the decoder from state $\underline{D}'_i = \underline{D}'(u)$ to $\underline{D}'_j = \underline{D}'(u+1)$. Thus given the set of states with the associated transition probabilities, and a suitable initial condition, $Pr(\underline{D}'(u) = \underline{D}'_i)$ for all u and i can (at least in principle) be determined.

Associated with each state $\underline{D}'_i$ is a probability of error, $P_{\underline{D}'_i}$, which is the probability that $\underline{e}(u)$, $\underline{e}(u+1)$, $\underline{e}(u+2)$,---- is such that for some $f > u$, $\underline{a}_j(f)$, $j = 0,1,-----,2^m-1$, all pass through a subset of the nodes $\{n_1(u), n_3(u), ---, n_{2^m-1}(u)\}$ at depth u into the trellis, given that $\underline{D}'(u) = \underline{D}'_i$.

$$\text{Thus } P_V(u) \triangleq Pr(i^*(u) \neq i(u)/_{\text{Viterbi Decoding}}) = \sum_{i=0}^{\#\underline{D}'-1} Pr(\underline{D}'(u+1)=\underline{D}'_i)P_{\underline{D}'_i} \quad (114)$$

is the probability of error associated with a Viterbi decoder as a function of u, and

$$P_V \triangleq \lim_{u\to\infty} P_V(u) \tag{115}$$

is the steady-state Viterbi decoding probability of error. (It will be shown shortly that $\lim_{u\to\infty} \Pr(\underline{D}'(u)=\underline{D}_i')$ and hence $\lim_{u\to\infty} P_V(u)$ always exists for a Viterbi decoder.)

As an example of a calculation of P_V, let us consider the $R=\frac{1}{2}$, systematic convolutional code with $g_{o(1)}^{(2)} = 1$ and $g_{1(1)}^{(2)} = 1$, the semi-infinite trellis of which is shown in Fig. 16.

For this code, let $\underline{D}_o' = (02)$, $\underline{D}_1' = (01)$, $\underline{D}_2' = (00)$, $\underline{D}_3' = (10)$, and $\underline{D}_4' = (20)$. Thus the associated Markov transition matrix may be calculated to be

$$\Pi = \begin{bmatrix} (1-p)^2 & 0 & 2p(1-p) & 0 & p^2 \\ (1-p)^2 & 0 & 2p(1-p) & 0 & p^2 \\ 0 & 1-p & 0 & p & 0 \\ p(1-p) & 0 & 1-2p+2p^2 & 0 & p(1-p) \\ p(1-p) & 0 & 1-2p+2p^2 & 0 & p(1-p) \end{bmatrix} \tag{116}$$

and from Π the steady-state state distribution found as

$$\Pr(\underline{D}'=\underline{D}_o') = \frac{1-4p+8p^2-7p^3+2p^4}{1+3p^2-2p^3}$$

$$\Pr(\underline{D}'=\underline{D}_1') = \frac{2p-5p^2+5p^3-2p^4}{1+3p^2-2p^3}$$

$$\Pr(\underline{D}'=\underline{D}_2') = \frac{2p-3p^2+2p^3}{1+3p^2-2p^3} \tag{117}$$

$$\Pr(\underline{D}'=\underline{D}_3') = \frac{2p^2-3p^3+2p^4}{1+3p^2-2p^3}$$

$$\Pr(\underline{D}'=\underline{D}_4') = \frac{p^2+p^3-2p^4}{1+3p^2-2p^3} .$$

Before determining the probability of error associated with each state, it is necessary to consider a particular tie-breaking rule utilized in the calculations. As is shown in Fig. 17, assume $\underline{D}'(u) = (01)$ and $\underline{r}(u) = \underline{e}(u) = (00)$. Now although the hypothesized path segment to $n_o(u+1)$ at time u+1 can come only from $n_o(u)$, the path segment to $n_1(u+1)$ from $n_o(u)$ as well as $n_1(u)$ results in the same value of $D_1'(u+1)$, and hence each is equally likely. In this example the convention was used that in case of such a tie, the path which is more favorable to a "hard decision" is chosen, thus resulting here in the choice of the branch from $n_o(u)$ to $n_1(u+1)$.

Note that although tie-breaking rules affect the branch configurations which are related to the making of hard decisions, and hence the probability of error associated with each state, they do not affect the state of the system at any time, since the <u>state</u> at time u depends merely upon the relative number of agreements with the received vector of the paths leading to the 2^m nodes at depth u, and not upon the particular <u>configuration</u> of the surviving paths prior to time u.

Thus with the tie-breaking mechanism suitably defined for this example, the probability of error associated with each state may be easily calculated to be

$$\begin{aligned} P_{\underline{D}'_0} &= 0 \\ P_{\underline{D}'_1} &= 0 \\ P_{\underline{D}'_2} &= 2p(1-p) \\ P_{\underline{D}'_3} &= 1 \\ P_{\underline{D}'_4} &= 1 \ , \end{aligned} \tag{118}$$

and hence

$$P_V = \sum_{i=0}^{4} Pr(\underline{D}'=\underline{D}'_i)P_{\underline{D}'_i} = \frac{7p^2-12p^3+10p^4-4p^5}{1+3p^2-2p^3} . \tag{119}$$

A comparison of equations (8) and (119) reveals that $P_V = P_{FD}$, where P_{FD} is the steady-state probability of error associated with feedback decoding of the code in the above example. The fact that $P_V = P_{FD}$ for this case appears to be merely a property of the particular code employed, and is not believed to hold in general.

(In fact, it can be demonstrated for this example that the Viterbi decoder results in the same sequence of information estimates as does the feedback decoder, given that both decoders operate on the same received sequence.)

Another interesting property of the P_V of equation (119) is that $P_V > p$ for $p^* < p < \frac{1}{2}$, where p^* is defined as the real root of $1-5p+5p^2-2p^3 = 0$ (see Chapter IV, p.23). This indicates that for $p^* < p < \frac{1}{2}$, even though the path chosen by the Viterbi decoder is more (or at least as equally) likely than (as) any other path through the trellis, the Viterbi algorithm does not minimize the probability of error per information digit for this example and particular range of p. For given $\underline{r}(u) \triangleq (r^{(1)}(u), r^{(2)}(u))$, the decoding algorithm for this code with $i^*(u) = r^{(1)}(u) = i(u) + e(u)$ has $\Pr(i^*(u) \neq i(u)) = \Pr(e(u) = 1) = p < P_V$.

This result is not as surprising if the set of possible paths through the trellis is viewed as a set of messages to be transmitted as a block code. For even a maximum likelihood block decoding algorithm does not <u>necessarily</u> minimize the probability of error with respect to the information digits which may be associated with the set of messages.

<u>Theorem 8</u>. For a binary, rate $R = \frac{1}{N_o}$, systematic convolutional code of memory order m, steady-state state occupancy probabilities associated with a Viterbi decoder always exist with $N \leq 2C$, where N is the smallest power of the Markov transition matrix Π associated with the decoder for which there exists a nonzero column,

$$C \triangleq 3m \left\lceil \frac{d_U\max + d_D\max + 2d_{DD}}{d_{DD}} \right\rceil , \qquad (120)$$

d_{DD} is the definite decoding minimum distance of the code,

$$d_D\max \triangleq \max_{i(u-m),--,i(u-1)} W(\underline{T}_D) \qquad (121)$$

where $\underline{T}_D \triangleq [i(u-m),---,i(u-1)]$

$$\begin{bmatrix} g_m & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ g_2 & & & \\ g_1 & g_2 & \cdots & g_m \end{bmatrix}, \qquad (122)$$

$$d_U\max \triangleq \max_{i(u),---,i(u+m-1)} W(\underline{T}_U) \tag{123}$$

where
$$\underline{T}_U \triangleq \left[\underline{i}(u),---,i(u+m-1)\right] \begin{bmatrix} \underline{g}_o & \underline{g}_1 & --- & \underline{g}_{m-1} \\ \underline{0} & \ddots & \ddots & | \\ | & \ddots & \ddots & \underline{g}_1 \\ \underline{0} & --- & \underline{0} & \underline{g}_o \end{bmatrix}, \tag{124}$$

the $\underline{g}_i$'s, $i = 0,1,--,m$, are defined in equation (113), and $\lceil \quad \rceil$ denotes the "smallest integer greater than or equal to" the enclosed number.

Proof: From Theorem 1 of Chapter III, it suffices to show the existence of a state $\underline{D}_i'$ such that every state of the decoder can be driven to $\underline{D}_i'$ in exactly 2C steps.

First of all, assume that the decoder is in an arbitrary state $\underline{D}_q'$ at time u, i.e., $\underline{D}'(u) = \underline{D}_q'$, and let $\underline{r}(u) = \underline{r}(u+1) = ---- = \underline{r}(u+2C-1) = \underline{0}$.

Now assume that for some $j \in \{0,1,---,2^m-1\}$, there exists a path $\underline{a}_j(u+2C)$ such that $\underline{a}_j(u+2C)$ does not pass through node $n_o(u+C)$. It is not possible then for this path $\underline{a}_j(u+2C)$ to pass through both nodes $n_o(u+k)$ and $n_o(u+f)$, $0 \leq k < C$, $C < f \leq 2C$, since the optimum path from $n_o(u+k)$ to $n_o(u+f)$ passes through $n_o(u+k+i)$, $i = 0,1,---,f-k$. Thus $\underline{a}_j(u+2C)$ does not pass through a zero order node (i.e., a node reachable by means of m consecutive "zero" branch decisions) for at least C consecutive time units. However over this span at least one branch corresponding to an information digit of a "one" must be taken every m consecutive time units, or else a zero order node would be reached. Thus over this span, path $\underline{a}_j(u+2C)$ picks up a distance of at least d_{DD} every 3m time units, by nature of the definition of d_{DD}, and hence over this particular span of C time units gains a distance of at least

$$\frac{C-3m}{3m} d_{DD} \geq \frac{(d_U max+d_D max+2d_{DD})\, d_{DD}}{d_{DD}} - d_{DD} = d_U max+d_D max+d_{DD}. \quad (125)$$

However there exists a path through the trellis from the node which $\underline{a}_j(u+2C)$ passes through at depth u to $n_j(u+2C)$, formed by taking 2C-m consecutive "zero" branches to node $n_o(u+2C-m)$, and then by proceeding to node $n_j(u+2C)$ along the m branches specified by the binary representation of "j". This path picks up a distance of at most $d_D max$ from depth u to u+m, a distance of 0 from u+m to u+2C-m, and a distance of at most $d_U max$ from depth u+2C-m to u+2C, resulting in at most a total distance gain of $d_D max + d_U max$. But since this path intersects $\underline{a}_j(u+2C)$ at depth u and gains a smaller distance than $\underline{a}_j(u+2C)$ over the next 2C time units, $\underline{a}_j(u+2C)$ is not the "maximum likelihood" path to node $n_j(u+2C)$ and a contradiction is obtained. Thus the conclusion is reached that $\underline{a}_i(u+2C)$, $i = 0,1,---,2^m-1$, all pass through node $n_o(u+C)$.

Hence the state at time u+2C, $\underline{D}'(u+2C)$, is merely that vector obtained by considering the minimum weight path from node $n_o(u+C)$ to each of the nodes $n_o(u+2C)$, $n_1(u+2C)$,---, $n_{2^m-1}(u+2C)$, and hence is independent of the particular state at time u. We define this state as the zero state and denote it henceforth as $\underline{D}_o'$.

Thus every state can be driven to $\underline{D}_o'$ in exactly 2C transitions, and the theorem is proved.

It might be noted that Viterbi decoders are "stable" [22] since their state moves autonomously from any state to the zero state.

The zero state defined above for a Viterbi decoder is analagous to the zero state for a feedback decoder which correctly decodes in the absence of channel errors, in the sense that for each the zero state makes a transition into itself as a result of the all-zero input vector.

In order to verify this for a Viterbi decoder, it can be shown by arguments similar to those used in the proof of the above theorem that for $\underline{r}(u) = \underline{r}(u+1) = ---- = \underline{r}(u+2C) = \underline{0}$, paths $\underline{a}_j(u+2C+1)$, $j = 0,1,---,2^m-1$, all pass through node $n_o(u+C+1)$ (as well as through node $n_o(u+C)$). Thus the state at time u+2C+1 will be determined by the minimum weight paths from node $n_o(u+C+1)$ to nodes $n_o(u+2C+1)$, $n_1(u+2C+1)$, ---, $n_{2^m-1}(u+2C+1)$, and hence $\underline{D}'(u+2C+1) = \underline{D}_o'$, the "zero state." However $\underline{D}'(u+2C) = \underline{D}_o'$ also, and thus since $\underline{r}(u+2C) = \underline{0}$, $\underline{D}_o'$ is seen to make a transition into itself upon application of the input $\underline{0}$.

Next let us consider the making of "hard decisions" by a Viterbi decoder which operates upon a semi-infinite trellis.

Theorem 9. For a Viterbi decoder for a binary, rate $R = \frac{1}{N_o}$, systematic convolutional code of memory order m,

Pr(hard decision i*(u-1) is not made by time f) $\le$

$$\le \exp\left(-N_o(f-u)\left[T_p\left(\frac{d_{DD}}{6mN_o}\right) - H\left(\frac{d_{DD}}{6mN_o}\right)\right]\right), \qquad (126)$$

for $p < \frac{d_{DD}}{6mN_o}$ and f-u sufficiently large, where

$$T_p(x) \triangleq -x\ln(p) - (1-x)\ln(1-p), \qquad (127)$$

$$H(x) \triangleq -x\ln(x) - (1-x)\ln(1-x), \qquad (128)$$

d_{DD} is the definite decoding minimum distance of the code, u is any arbitrary but fixed distance into the trellis, and f is the depth at which the decoder is presently operating. Moreover, for arbitrary p,

$$\lim_{f\to\infty} \Pr(\text{hard decision } i^*(u-1) \text{ is made by time } f) = 1. \qquad (129)$$

Proof: Let $C_{u,f}$ be the random variable defined as

$$C_{u,f} \triangleq W(\underline{r}(u):\cdots:\underline{r}(f-1)) \; ; \qquad (130)$$

$C_{u,f}$, of course, is equal to the weight of the error pattern from time u to f, since the all-zero sequence is assumed to be transmitted.

Now assume that $\underline{a}_j(f)$ and $\underline{a}_i(f)$, $i \ne j$, $f > u$, are distinct at least from depth u to depth f of the trellis; i.e., they do not pass through a common node over this span. Therefore $d_{u,f}(\underline{a}_j(f),\underline{a}_i(f)) \ge \left\lfloor\frac{f-u}{3m}\right\rfloor d_{DD}$, where $d_{u,f}(\;,\;)$ denotes the Hamming distance between the segments of the two enclosed vectors from depth u to depth f of the trellis. We also note that $d_{u,f}(\underline{a}_j(f),\underline{R}(f)) + d_{u,f}(\underline{a}_i(f),\underline{R}(f)) \ge$ $\ge d_{u,f}(\underline{a}_j(f),\underline{a}_i(f))$ from the triangle inequality.

Now $d_{u,f}(\underline{a}_j(f),\underline{R}(f)) \leq d_{u,f}(\underline{a}_i(f),\underline{R}(f)) + 2d_S\max$, since from the proof of Lemma 4, $d(\underline{a}_j(f),\underline{R}(f)) \leq d_S\max +$ $+ d(\underline{a}_i(f),\underline{R}(f))$ and $d_{0,u}(\underline{a}_i(f),\underline{R}(f)) \leq d_{0,u}(\underline{a}_j(f),\underline{R}(f)) +$ $+ d_S\max$. Therefore $d_{u,f}(\underline{a}_j(f),\underline{R}(f)) + d_{u,f}(\underline{a}_i(f),\underline{R}(f)) \leq$ $\leq 2d_{u,f}(\underline{a}_i(f),\underline{R}(f)) + 2d_S\max$.

We also note that $d_{u,f}(\underline{a}_i(f),\underline{R}(f)) \leq d_{u,f}(\underline{T}_i(u,f),\underline{R}(f))$ by definition of $\underline{a}_i(f)$, where $\underline{T}_i(u,f)$ is the path from the node of the trellis which $\underline{a}_i(f)$ intersects at depth u, formed by taking f-u-m successive "zero" branches to n_o f-m) and then proceeding to node $n_i(f)$. Also, $d_{u,f}(\underline{T}_i(u,f),\underline{R}(f))$ $\leq C_{u,f} + 2mN_o$.

Thus in summary,

$$\left\lfloor\frac{f-u}{3m}\right\rfloor d_{DD} \leq d_{u,f}(\underline{a}_j(f),\underline{a}_i(f)) \leq d_{u,f}(\underline{a}_j(f),\underline{R}(f)) + d_{u,f}(\underline{a}_i(f),\underline{R}(f))$$
$$\leq 2d_{u,f}(\underline{a}_i(f),\underline{R}(f)) + 2d_S\max \leq 2d_{u,f}(\underline{T}_i(u,f),\underline{R}(f)) + 2d_S\max$$
$$\leq 2C_{u,f} + 4mN_o + 2d_S\max, \tag{131}$$

and therefore

$$C_{u,f} \geq \frac{f-u}{6m}d_{DD} - \frac{7}{2}mN_o. \tag{132}$$

Thus if <u>actually</u> $\frac{C_{u,f}}{N_o(f-u)} < \frac{d_{DD}}{6mN_o} - \frac{7m}{2(f-u)}$, a contradiction is obtained to the assumption that two arbitrary paths $\underline{a}_j(f)$ and $\underline{a}_i(f)$ are distinct back through depth u. And hence a hard decision (i.e., i*(u-1)) is assured at time f.

Therefore from the Chernoff bound [23] , for $p < \frac{d_{DD}}{6mN_o}$,

Pr(hard decision $i^*(u-1)$ is not made by time f) $\leq$

$$\leq \Pr\left(\frac{C_{u,f}}{N_o(f-u)} \geq \frac{d_{DD}}{6mN_o}\right) \leq \exp\left(-N_o(f-u)\left[T_p\left(\frac{d_{DD}}{6mN_o}\right)-H\left(\frac{d_{DD}}{6mN_o}\right)\right]\right), \quad (133)$$

for f-u sufficiently large such that $\frac{7m}{2(f-u)}$ is insignificant compared to $\frac{d_{DD}}{6mN_o}$.

Now since there exist sequences of error vectors which result in "hard decision" $i^*(u-1)$ (as well as $i^*(0)$,---, $i^*(u-2)$) being made by some future time f, even for arbitrary p the probability is "one" that one of these sequences will eventually occur, and thus

$\lim_{f\to\infty}$ Pr(hard decision $i^*(u-1)$ is made by time f) = 1,

completing the proof of the theorem.

It would be fortunate if a hard decision were in general "guaranteed" after a given number of additional time units, but the following counterexample demonstrates that such is not the case.

Consider the trellis for the binary, rate $R = \frac{1}{2}$, systematic convolutional code of memory order $m = 2n-1$, with $\underline{g}^{(2)}_{(1)} \triangleq (g^{(2)}_{o(1)},---,g^{(2)}_{m(1)}) \triangleq (\hat{\underline{g}}:\hat{\underline{g}})$, (or equivalently, $\hat{\underline{g}} \triangleq (g^{(2)}_{o(1)},---,g^{(2)}_{n(1)})$), $g^{(2)}_{o(1)} = 1$, and $W(\hat{\underline{g}}) \geq 2$, for the semi-infinite received vectors $\underline{r}^{(1)} \triangleq (r^{(1)}(0), r^{(1)}(1),--)$ $= \underline{0}$ and $\underline{r}^{(2)} \triangleq (r^{(2)}(0), r^{(2)}(1), --) = (\hat{\underline{g}}:\hat{\underline{g}}:\hat{\underline{g}}:---) + (100--0-)$.

Now at depth tn into the trellis, t an even integer,

$\underline{a}_o(tn)$ is such that $a_{o,tn}(i) = 1$, $i = 0, 2n, 4n, ---, (t-2)n$, and $a_{o,tn}(i) = 0$ for i not a multiple of $2n$. Such a path disagrees with the received vector only once every m+1 time units with the exception of 2 differences over the first m+1 time units. But at depth tn, for t an odd integer, $a_{o,tn}(i) = 1$, $i = n, 3n, 5n, ---, (t-2)n$, and $a_{o,tn}(i) = 0$ otherwise. This path disagrees with the received vector only once every m+1 time units after depth n, although before n it has a number of disagreements equal to $W(\hat{\underline{g}})-1$. (The (100--0---) portion of $\underline{r}^{(2)}$ forces the first n time units to adsorb the parity disagreements rather than any other such n time units, since $g_{o(1)}^{(2)} = 1$.)

Thus $a_{o,tn}(0) = 1$ for t even and $a_{o,tn}(0) = 0$ for t odd, indicating that a hard decision is never made on i(0) for this code and received pattern even as the depth into the trellis approaches infinity, and we have the desired counterexample.

Lemma 5. For a binary, rate $R = \frac{1}{N_o}$, systematic convolutional code of memory order m, a "hard decision" on the information digit at time u, i(u), cannot be made at least until the Viterbi decoder is operating at depth u+m+1 into the trellis.

Proof: A hard decision on i(u) implies that for some $f > u$, $a_{j,f}(u)$, $j = 0, 1, ---, 2^m-1$, all pass through nodes at depth u+1 corresponding to the same decision on i(u), i.e., through a subset of either $\{n_0(u+1), n_2(u+1), --, n_{2^m-2}(u+1)\}$ or

$\{n_1(u+1), n_3(u+1), ---, n_{2^m-1}(u+1)\}$. But paths restricted to passing through only one such subset cannot reach all 2^m nodes at a greater depth into the trellis for at least m more branches. Hence $f \geq m+u+1$ and the lemma is proved.

Even though the number of states necessary and sufficient for the calculation of P_V is finite, the number of states required in an actual implementation of a Viterbi decoder is finite if and only if the code is such that a hard decision is always made on i(u) at least by the time the decoder is operating at depth j+u into the trellis, for some j, $m+1 \leq j < \infty$. For a Viterbi decoder, in order to decode i(u), must keep track of the surviving paths in the future until such a time that all of the paths emanating from the nodes of the trellis at depth u "agree" on the same estimate of i(u).

We next show that, for a restricted range of rates, there exist convolutional codes such that the associated steady-state error probability of Viterbi decoding, P_V, decreases exponentially with encoding constraint length. First, however, a definition and two preliminary lemmas are needed.

Definition 10: The received vectors $\underline{r}(i)$, $\underline{r}(i+1)$,---, $\underline{r}(j-1)$ are said to be low weight received vectors if and only if $W(\underline{r}(q):\underline{r}(q+1):---:\underline{r}(q+m)) \leq \left\lfloor \frac{d_{DD}-1}{6} \right\rfloor$, q = i, i+1,---, j-1-m, where d_{DD} is the definite decoding minimum distance of the associated code.

Lemma 6. If the received vectors $\underline{r}(i)$, $\underline{r}(i+1)$,---, $\underline{r}(j-1)$ are low weight, then the path through the trellis from node $n_o(k)$ to $n_o(f)$ with the smallest number of disagreements with the received sequence is the all-zero path between nodes $n_o(k)$ and $n_o(f)$, $i \leq k < f \leq j$.

Proof: The lemma is trivially true for $f-k \leq m$, since for this case the only path from $n_o(k)$ to $n_o(f)$ is the all-zero path segment.

For $f-k \geq m+1$, it suffices to verify that $d_1(n_o(k), n_o(f)) > d_o(n_o(k), n_o(f))$, where $d_1(n_o(k), n_o(f))$ is defined as the minimum number of disagreements with a low weight received vector of any path from $n_o(k)$ to $n_o(f)$ of the form $a(k)=1$, $a(f-m-1)=1$, $a(f-m)=a(f-m+1)=---=a(f-1)=0$, and $d_o(n_o(k), n_o(f))$ is defined as the maximum number of possible disagreements of a low weight received vector with the all-zero path segment between $n_o(k)$ and $n_o(f)$.

Now a path from $n_o(k)$ to $n_o(f)$ with $a(k)=1$, $a(f-m-1)=1$, and $a(f-m)=---=a(f-1)=0$ has a weight of at least $d_{FD} \geq d_{DD}$ over the first m+1 branches, a weight of at least d_{DD} over the last 2m+1 branches, and a weight of at least d_{DD} over each consecutive string of 3m branches. And a low weight received vector has a weight of at most $\left\lceil \frac{f-k}{m+1} \right\rceil \frac{d_{DD}-1}{6}$ over f-k consecutive branches. Thus it can be easily verified that

$$d_1(n_o(k), n_o(f)) \geq \frac{d_{DD}+1}{2}, \quad m+1 \leq f-k \leq 3m+1$$

$$\geq A\frac{d_{DD}+1}{2}, \quad 3(A-1)m+2 \leq f-k \leq 3Am+1, \quad A = 2,3,4,--- \qquad (134)$$

And since

$$d_o(n_o(k),n_o(f)) \le \left\lceil \frac{f-k}{m+1} \right\rceil \frac{d_{DD}-1}{6} \quad , \tag{135}$$

it can be shown that

$$d_1(n_o(k),n_o(f)) > d_o(n_o(k),n_o(f)), \quad f-k = m+1,\ m+2,\ --. \tag{136}$$

Hence the optimum path between two zero order nodes, conditioned upon reception of a low weight received vector, is the all-zero path between the two nodes. This completes the proof of the lemma.

Lemma 7. If the received vectors $\underline{r}(u-F)$, $\underline{r}(u-F+1)$, ---, $\underline{r}(u)$, ---, $\underline{r}(u+F-1)$ are low weight, where $F \triangleq 3m(d_D\max + d_U\max + 1)$, then $\underline{a}_i(u+F)$, $i = 0,1,---,2^m-1$, all pass through $n_o(u)$ and hence $i^*(u-1) = 0$.

Proof: Assume $\underline{r}(u-F)$,---,$\underline{r}(u-F+1)$ are low weight and that $\underline{a}_j(u+F)$ does not pass through $n_o(u)$ for some $j \in \{0,1,--,2^m-1\}$. Now $\underline{a}_j(u+F)$ cannot pass through both nodes $n_o(k)$ and $n_o(f)$, $u-F \le k < u < f \le u+F$, because from Lemma 6 the optimum path from $n_o(k)$ to $n_o(f)$ passes through $n_o(u)$. Hence $\underline{a}_j(u+F)$ does not pass through a zero order node for at least F consecutive branches through the trellis (i.e., from depth u-F to u or from u to u+F, or both).

Without loss of generality, assuming that these F consecutive branches are from depth u to u+F, $\underline{a}_j(u+F)$ picks up a distance with respect to the received vector over this span of at least

$$\left\lfloor \frac{F}{3m} \right\rfloor d_{DD} - \left\lceil \frac{F}{m+1} \right\rceil \left\lfloor \frac{d_{DD}-1}{6} \right\rfloor \ge \frac{F}{3m} d_{DD} - \frac{F}{m} \frac{d_{DD}-1}{6} =$$

$$= (d_U max + d_D max + 1)d_{DD} - \tfrac{1}{2}(d_U max + d_D max + 1)(d_{DD}-1)$$
$$= \tfrac{1}{2}d_{DD}(d_U max + d_L max + 1) + \tfrac{1}{2}(d_U max + d_D max) + \tfrac{1}{2}. \qquad (137)$$

But the path to $n_j(u+F)$ from the node which $\underline{a}_j(u+F)$ intersects at depth u, formed by taking F-m successive "zero" branches and then proceeding from $n_o(u+F-m)$ to $n_j(u+F)$, picks up a distance with respect to the received path of at <u>most</u>

$$d_D max + d_U max + \left\lceil \frac{F}{m+1} \right\rceil \left\lfloor \frac{d_{DD}-1}{6} \right\rfloor \leq d_D max + d_U max + \frac{F}{m}\,\frac{d_{DD}-1}{6}$$
$$= d_D max + d_U max + \tfrac{1}{2}d_{DD}(d_U max + d_D max + 1) - \tfrac{1}{2}(d_D max + d_U max + 1)$$
$$= \tfrac{1}{2}d_{DD}(d_U max + d_D max + 1) + \tfrac{1}{2}(d_D max + d_U max) - \tfrac{1}{2}. \qquad (138)$$

This distance is thus less than that associated with $\underline{a}_j(u+F)$ over the same span and a contradiction is obtained. Hence $\underline{a}_i(u+F)$, $i = 0,1,---,2^m-1$, all pass through $n_o(u)$, $i^*(u-1) = 0$, and the lemma is proved.

Therefore, if the all-zero transmitted sequence is assumed, the probability of error per information digit, $Pr(i^*(u-1) \neq i(u-1))$, may be upper bounded as

$$Pr(i^*(u-1) \neq i(u-1)) \leq Pr(W(\underline{e}(q):\underline{e}(q+1):---:\underline{e}(q+m)) > \left\lfloor \frac{d_{DD}-1}{6} \right\rfloor),$$
$$\text{for some } q \in \{u-F,\ u-F+1, ---, u+F-m-1\}. \qquad (139)$$

And once again utilizing the Chernoff Bound, the expression

$$Pr\left(\frac{W(e(q):---:e(q+m))}{N_{FD}} > \frac{\left\lfloor \frac{d_{DD}-1}{6} \right\rfloor}{N_{FD}}\right) \leq e^{-N_{FD}E_X}, \qquad (140)$$

for $p < \dfrac{\left\lfloor \frac{d_{DD}-1}{6} \right\rfloor}{N_{FD}}$, is obtained, where

$$E_X \triangleq T_p\left(\frac{\left\lfloor \frac{d_{DD}-1}{6} \right\rfloor}{N_{FD}}\right) - H\left(\frac{\left\lfloor \frac{d_{DD}-1}{6} \right\rfloor}{N_{FD}}\right), \qquad (141)$$

$T_p(x)$ and $H(x)$ are defined in equations (127) and (128), and $N_{FD} \triangleq (m+1)N_o$ is the feedback decoding constraint length (or the encoding constraint length).

Therefore, from equations (139), (140), the union bound, and the definition of F, the result

$$P_V \le (2F-m)e^{-N_{FD}E_X} < 12m^2 N_o e^{-N_{FD}E_X}, \qquad (142)$$

for $p < \frac{\left\lfloor \frac{d_{DD}-1}{6} \right\rfloor}{N_{FD}}$, is obtained.

For N_{DD} sufficiently large, where $N_{DD} = N_{FD}(1+R)$ is the definite decoding constraint length, Massey [24] has shown that there exist binary, systematic, rate $R = \frac{K_o}{N_o}$ convolutional codes of memory order m such that $\frac{d_{DD}}{N_{DD}} \ge H^{-1}(0.1\frac{1-R}{1+R})$. In the above paper it was conjectured that tighter bounding arguments could do away with the factor of 0.1 and the result $\frac{d_{DD}}{N_{DD}} \ge H^{-1}(\frac{1-R}{1+R})$ obtained, which is identical to the bound proved for periodic convolutional codes [25]. Assuming the truth of this conjecture and utilizing equation (142), the following theorem is obtained.

<u>Theorem 10</u>. For $p < Z(R) \triangleq \frac{1+R}{6}H^{-1}(\frac{1-R}{1+R})$, and for $N_{FD} \triangleq (m+1)N_o$ sufficiently large, there exist binary, rate $R = \frac{1}{N_o}$, systematic convolutional codes of memory order m such that the associated steady-state Viterbi decoding probability of error P_V is upper bounded by

$$P_V < 12m^2 N_o e^{-N_{FD} E_V(R)} \qquad (143)$$

where $$E_V(R) \triangleq T_p(Z(R)) - H(Z(R)) \qquad (144)$$

For $p = .015$, the rate R must satisfy $R < \frac{1}{2}$ in order for $p < Z(R)$ to hold; this range of rates is substantially less than channel capacity C, which here is equal to .887 bits/channel symbol. Also, $E_V(0) = .0703$, which compares unfavorably to $R_o = .683$, where R_o is the zero-rate random coding exponent for both convolutional and block codes [26] for the particular channel.

The value of Theorem 10, in spite of the above mentioned limitations on $E_V(R)$, lies in the demonstration that variable span decoders for convolutional codes can be operated without resynchronization and a steady-state error probability obtained which decreases exponentially with encoding constraint length.

It should be noted that a definite decoder which corrects every pattern of $\left\lfloor \frac{d_{DD}-1}{2} \right\rfloor$ or fewer errors over a definite decoding constraint length could achieve a performance similar to that guaranteed Viterbi decoders by equation (143), and for a higher range of rates. For here, p would only need satisfy $p < \frac{\left\lfloor \frac{d_{DD}-1}{2} \right\rfloor}{N_{DD}}$.

Viterbi decoders are rendered impractical because the complexity of their implementation grows greater than exponentially with encoder memory m. The most practical "variable span" decoding technique to date, sequential decoding, has a complexity which grows (approximately) only linearly with m [27].

Straightforward generalizations of the results of this section can be made to arbitrary rate $R = \frac{K_o}{N_o}$ convolutional codes, as was mentioned earlier. Generalizations can also be made to codes over GF(q) if the binary symmetric channel is replaced by a memoryless, time-invariant, additive GF(q) noise source.

VIII. SUMMARY, CONCLUSIONS, AND SUGGESTIONS FOR FURTHER RESEARCH

In Chapter I, the concepts of genie, definite, and feedback decoding of convolutional codes were introduced, and an explanation given as to why previously attempted analyses of the feedback decoding probability of error had proven fruitless.

In Chapter II, a new approach for calculating the feedback decoding error probability, in which the entire feedback decoder is modeled as a finite, autonomous stochastic sequential machine, was introduced via its application to a simple convolutional code and feedback decoding function. For this example it was found that $P_{GD} < P_{FD} < P_{DD}$, $0 < p < \frac{1}{2}$, where P_{GD}, P_{FD}, and P_{DD} are the error probabilities (steady-state) associated with genie, feedback, and definite decoding, respectively, for the particular code, and p is the transition probability of the binary symmetric channel. This is the first instance known that an exact expression has been obtained for P_{FD} for a non-trivial convolutional code and decoding rule.

In Chapter III, the stochastic sequential machine approach was formalized for feedback decoders for binary, rate $R = \frac{K_o}{N_o}$, systematic convolutional codes of memory order m, used for transmission over a binary symmetric channel, and a precise expression developed for $P_{FD}(u)$, the probability of error associated with the feedback decoding of the u^{th} subblock of information digits. Unfortunately, as was stated, the number of states of the machine, and hence the complexity of the computation required, grows exponentially with mN_o.

Also in Chapter III, sufficient conditions were given for the existence of a steady-state decoder state probability vector $\underline{W}$, and it was shown that the existence of $\underline{W}$ is a sufficient condition for the existence of $P_{FD} \triangleq \lim_{u \to \infty} P_{FD}(u)$. Examples were given for which P_{FD} exists but $\underline{W}$ does not, and for which neither $\underline{W}$ nor P_{FD} exists.

In Chapter IV, the Quasi-Maximum Likelihood Decoder (QMLD), Maximum Likelihood Decoder (MLD), and True Maximum Likelihood Decoder (TMLD), all feedback decoders, were defined. By means of an example it was demonstrated that, in the steady-state mode, the estimates made by an MLD are not <u>necessarily</u> the true "maximum likelihood" estimates, conditioned upon the syndrome state and input. Another example was used to show that a feedback decoder which puts out maximum likelihood estimates in steady-state is not <u>necessarily</u> a TMLD, that is, a feedback decoder for which P_{FD} is a minimum.

Also in Chapter IV, it was proved that steady-state state occupancy probabilities exist for a QMLD with N = m, where N is the smallest power of the Markov transition matrix associated with the decoder states for which there exists a nonzero column. Steady-state state occupancy probabilities were also found to exist for another class of feedback decoders, namely the decoders which correct up to the feedback decoding minimum distance of the code and estimate the all-zero vector whenever the syndrome state and input vector is consistent with no "minimum distance correctable" error pattern.

In Chapter V, the stochastic sequential machine approach was used in order to obtain an expression for $P_{SDD}(k)$, the probability of error of a k stage semi-definite decoder. The concept of an "equivalent feedback decoder" proved useful in the development and was found to provide insight into semi-definite decoding operation. A sufficient condition was obtained for $\lim_{k\to\infty} P_{SDD}(k) = P_{FD}$, and examples given both for which this event does and does not occur. Also, an example was exhibited for which there exists a particular value of k, k_q, such that $P_{SDD}(k_q)$ is strictly less than both the P_{FD} for a TMLD and the P_{DD} for a "maximum likelihood definite decoder" associated with the code, indicating that semi-definite decoding may be of practical importance.

In Chapter VI, it was shown that nothing in general can be said about the ordering of P_{GD}, P_{FD}, and P_{DD}, although some special case relationships were found among these quantities. By application of the Theorem of Irrelevance of Wozencraft and Jacobs, it was shown that any method of decoding the u^{th} subblock of information digits, in which the received vectors corresponding to times greater than u+m are not observed, has an associated probability of error which is lower bounded by P_{MLGD}, the probability of error of the "maximum likelihood genie decoder" for the same code. In particular, $P_{MLGD} \leq P_{FD}$ and $P_{MLGD} \leq P_{DD}$.

Also in Chapter VI, in the limit as the binary symmetric channel transition probability p approaches zero, the relationship $P_{FD} < P_{DD}$ was established for codes for which more errors are guaranteed correctable by the feedback decoding minimum distance than by the definite decoding minimum distance, and which are decoded by a particular class of feedback decoders. This class decodes up to the feedback decoding minimum distance of the code and exhibits finite error propagation length L.

In Chapter VII, Viterbi decoders were described in the framework of the "condensed code tree" or "code trellis" formulation of Forney. By proper definition of the "state" of the system, it was shown that a Viterbi decoder also could be modeled as a finite autonomous stochastic sequential machine for the purpose of calculating the probability of error per information digit, even for non-finite encoding trees. Here also, unfortunately, it was shown that the number of states of the machine grows extremely rapidly (greater than exponentially) with encoder memory m.

An example was given in which the above approach was illustrated, and in which it was demonstrated that, even though a maximum likelihood <u>sequence</u> of information digits is estimated by a Viterbi decoder, the probability of error per information digit is not necessarily minimized.

Steady-state state occupancy probabilities, and thus steady-state decoding probability of error, were found to exist for Viterbi decoders for every convolutional code, and a bound given on N, which here, as before, is the smallest integer power of the Markov transition matrix associated with the decoder states for which there exists a non-zero column. A natural definition of the zero-state was given, and this state, analagously to the zero-state of a feedback decoder which correctly decodes in the absence of channel errors, was found to make a transition into itself upon application of the all-zero input.

Also in Chapter VII, the concept of a "hard decision" was introduced, and a lower bound given on the number of time units that must elapse before such a hard decision is made by a Viterbi decoder. An example was given in which it was demonstrated that a corresponding upper bound does not exist for all codes, although it was shown that a hard decision is eventually made by a Viterbi decoder with probability 1.

Finally, in Chapter VII it was proved that, for a restricted range of rates, there exist convolutional codes such that the Viterbi decoding probability of error per information digit decreases exponentially with increasing encoding constraint length. Although this upper bound is substantially inferior to the random coding upper bounds for convolutional and block codes, its value lies in the demonstration that "variable span" decoders can be operated without resynchronization and still attain a low probability of error per information digit.

Unfortunately, the number of states required in a calculation of P_{FD}, $P_{SDD}(k)$, or P_V grows at least exponentially with encoder memory m, which rules out the results of this thesis as practical methods of calculating error probabilities for "real world" decoders. However, the results presented here do add much insight into feedback, semi-definite, and Viterbi decoding behavior, and serve as an analytical framework for further research in the area, which could possibly lead to very practical results.

One area of practical interest that might be investigated further is that concerned with a comparison of P_{FD} and P_{DD} for certain classes of codes and decoding functions. For example, does there exist a theorem of the form: "For 'good' codes and decoding functions $\underline{f}$, $P_{FD} < P_{DD}$, ---."?

Another interesting question concerns the determination of the $\underline{f}$ of a TMLD for a binary, rate $R = \frac{K_o}{N_o}$, systematic convolutional code of memory order m. This desired function $\underline{f}$ is the solution of a minimization problem in $2^{(m+1)(N_o-K_o)}$ variables, corresponding to the number of distinct possible syndrome vectors $\underline{Y}_1$ of the decoder, and can, at least in principle, be determined. However does there possibly exist a simpler method, perhaps one based upon convolutional code structure, of determining the optimum $\underline{f}$? Also, does there exist a simple test for proving that a given function $\underline{f}$ results in a TMLD? (It should be remembered that even if each estimate $\underline{e}_I^*(u)$ is maximum likelihood conditioned upon $\underline{Y}(u+m)$ in steady-state, the corresponding $\underline{f}$ is not necessarily that of a TMLD.)

It would be of interest to prove that the $\underline{f}$ of a TMLD is necessarily deterministic (or to produce an appropriate counterexample). That is, for $K_0 = 1$, if $Pr(f(\underline{Y}_i) = 0) = 1-w_i$ and $Pr(f(\underline{Y}_i) = 1) = w_i$, $0 \leq w_i \leq 1$, then is the f for the TMLD specified by $w_i \in \{0,1\}$, $\forall i$?

A determination of the conditions under which semi-definite decoding yields a smaller probability or error for a given code than either feedback or definite decoding would prove very useful.

Also, the problem of "hard decision" making by a Viterbi decoder should be investigated, in order to determine if there exist classes of codes for which a hard decision on i(u) is guaranteed after a certain finite number of additional time units.

REFERENCES

[1] J. P. Robinson, "Error Propagation and Definite Decoding of Recurrent Codes", Tech. Rept. No. 47, Princeton University, Princeton, N. J., June, 1965, p. 12.

[2] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, New York: Wiley, 1965, p. [illegible].

[3] D. D. Sullivan, "Control of Error Propagation in Convolutional Codes", Tech. Rept. No. EE-667, University of Notre Dame, Notre Dame, Indiana, November, 1966.

[4] J. W. Carlyle, "Equivalent Stochastic Sequential Machines", *Tech. Rept. Series 60*, Issue 415, University of California at Berkeley, also as "Reduced Forms for Stochastic Sequential Machines", *J. Math. Anal. and Appl.*, Vol. 7, 1963, pp. 167-175.

[5] C. R. Souza and R. J. Leake, "Stochastic Finite-State Systems: Four Models and Their Relationships", *Proc. 5th Annual Allerton Conference on Circuits & System Theory*, Monticello, Illinois, Oct., 1967.

[6] J. L. Massey, *Threshold Decoding*, The Technology Press of M.I.T., Cambridge, Mass., 1963, p. 55.

[7] R. Ash, *Information Theory*, New York: Wiley, 1965, p. 176.

[8] J. L. Massey and R. W. Liu, "Application of Lyapunov's Direct Method to the Error-Propagation Effect in Convolutional Codes," *IEEE Transactions of the Professional Technical Group on Information Theory*, Vol. IT-10, Num. 3, July, 1964.

[9] L. A. Frasco, "Analysis and Simulation of Semi-Definite Decoding", S. M. Thesis, M.I.T., Course VI, June, 1967.

[10] J. M. Wozencraft and I. M. Jacobs, op. cit., p. 220.

[11] J. L. Massey, "Some Algebraic and Distance Properties of Convolutional Codes", to appear in *Proceedings, Symposium on Error Correcting Codes*, sponsored by Math. Research Center, U.S. Army at the University of Wisconsin, May 6-8, 1968.

[12] J. P. Robinson, op. cit. p. 17.

[13] J. L. Massey, op. cit.

[14] J. P. Robinson, op. cit., p. 17.

[15] D. D. Sullivan, op. cit., p. 19.

[16] J. P. Robinson, op. cit., p. 8.

[17] J. L. Massey, "Catastrophic Error-Propagation in Convolutional Codes", Proceedings, Eleventh Midwest Symposium on Circuit Theory, University of Notre Dame, Notre Dame, Indiana, May 13-14, 1968.

[18] J. L. Massey, Threshold Decoding, The Technology Press of M.I.T., Cambridge, Mass., 1963, p. 6.

[19] J. M. Wozencraft and I. M. Jacobs, op. cit., p. 412.

[20] G. D. Forney, Private Communication, June, 1968.

[21] A. J. Viterbi, "Error Bounds For Convolutional Codes And An Asymptotically Optimum Decoding Algorithm", IEEE Transactions On Information Theory, Vol. IT-13, Num. 3, April, 1967.

[22] J. L. Massey and R. W. Liu, op. cit.

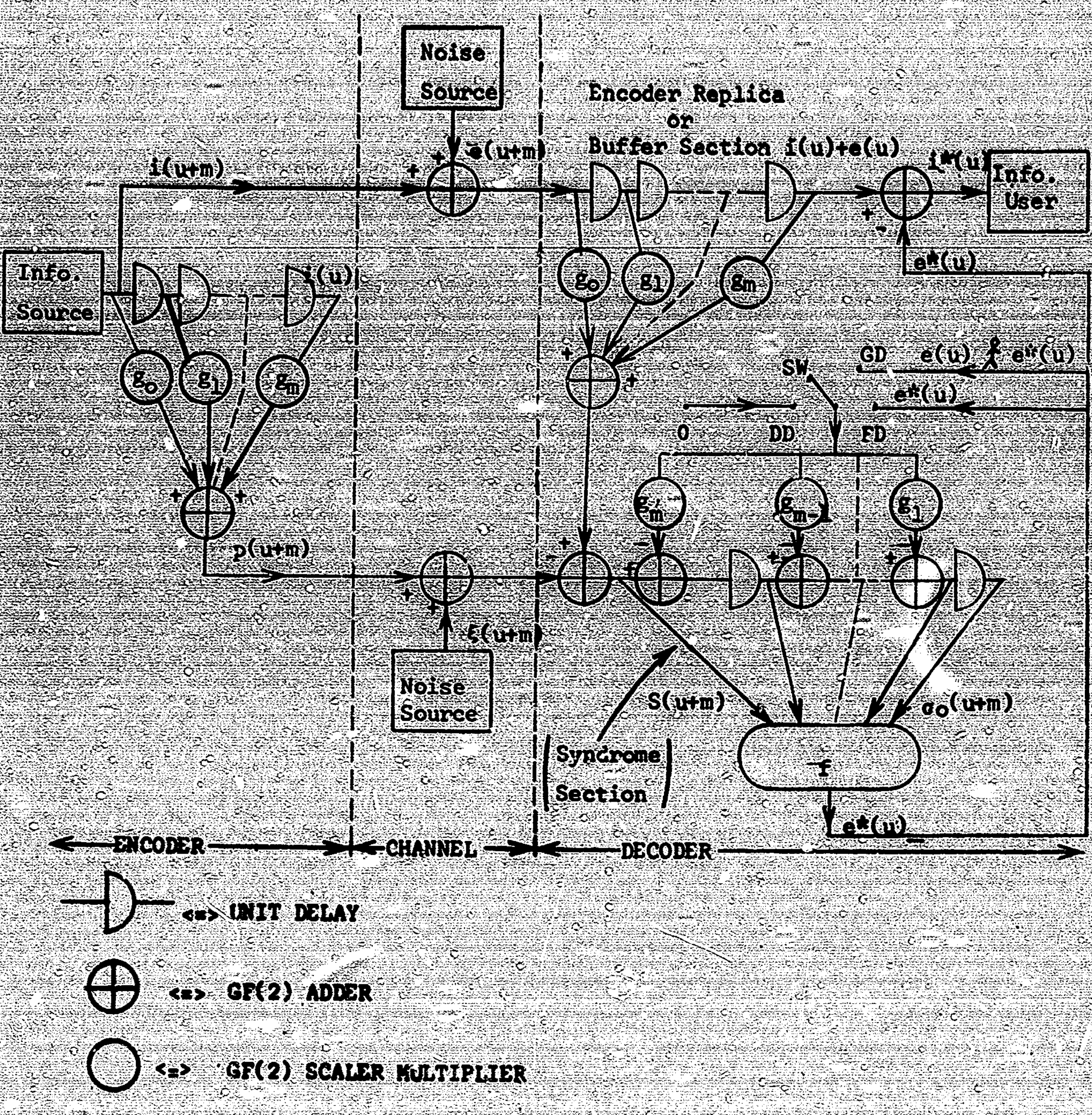
[23] J. M. Wozencraft and I. M. Jacobs, op. cit., p. 103.

[24] J. L. Massey, "Some Algebraic and Distance Properties of Convolutional Codes", to appear in Proceedings, Symposium on Error Correcting Codes, sponsored by Math. Research Center, U.S. Army at the University of Wisconsin, May 6-8, 1968.

[25] D. Costello, Private Communication, University of Notre Dame, Notre Dame, Indiana, 1968.

[26] A. J. Viterbi, op. cit., p. 267.

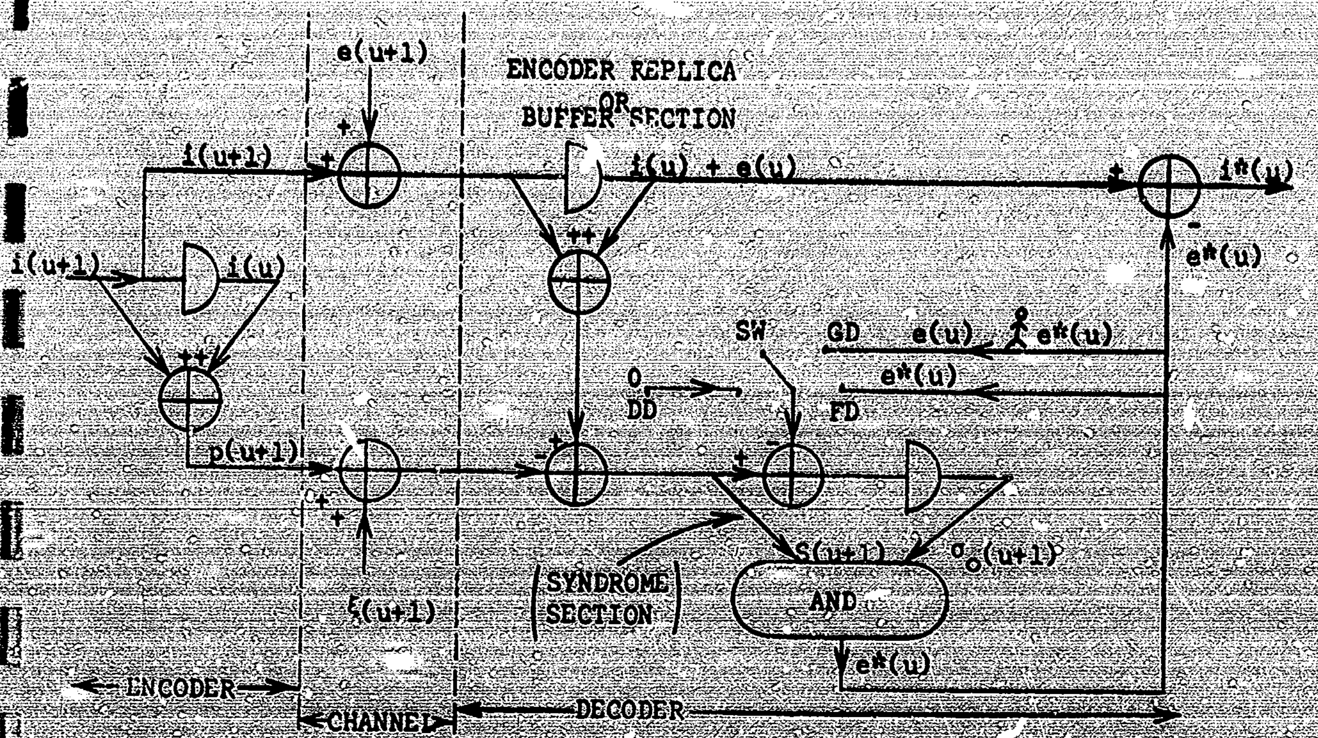
[27] R. M. Fano, "A Heuristic Discussion of Probabilistic Decoding", IEEE Transactions On Information Theory, Vol. IT-9, April, 1963, p. 74.



$g_i \; \varepsilon \; \{0,1\}$

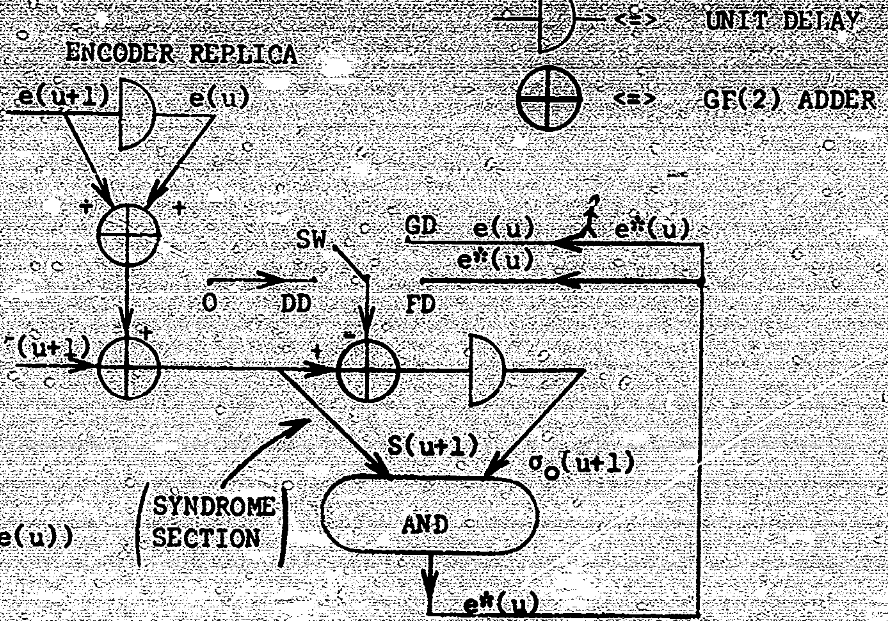
GENERAL, BINARY, $R = \frac{1}{2}$ ENCODING & DECODING SYSTEM

Fig. 1



BINARY, $R = \frac{1}{2}$ ENCODING and DECODING SYSTEM - SPECIAL CASE

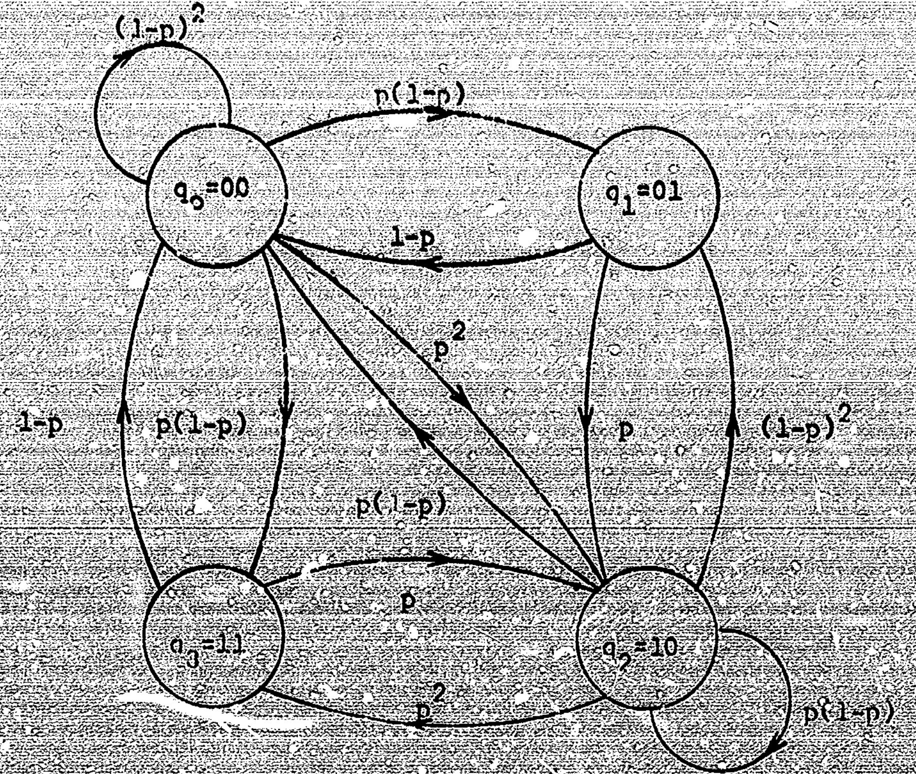
Fig. 2



Pr(ERROR) = Pr(e*(u)≠e(u))

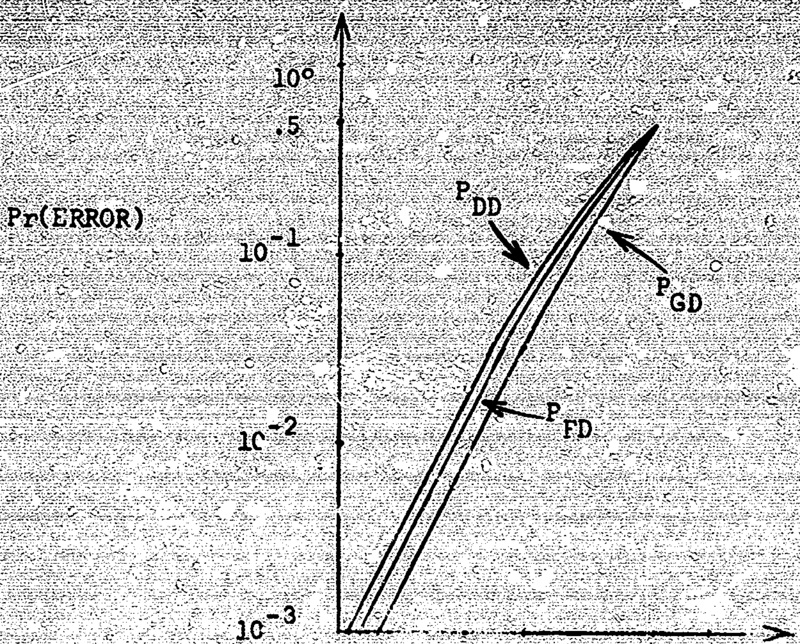
DECODER PORTION OF Fig. 2

Fig. 3



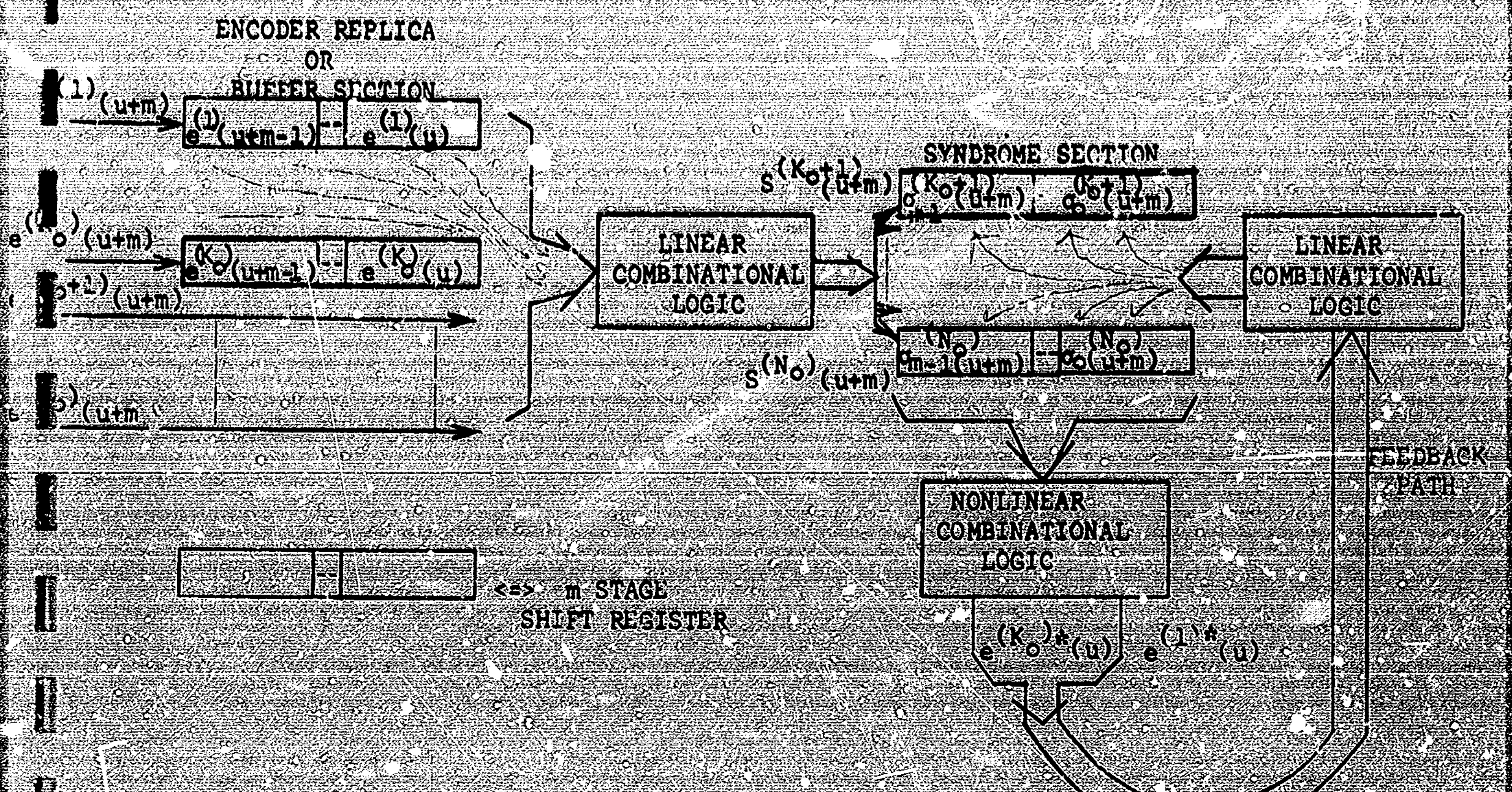
PROBABILISTIC STATE TRANSITION DIAGRAM FOR DECODER OF Fig. 3

Fig. 4



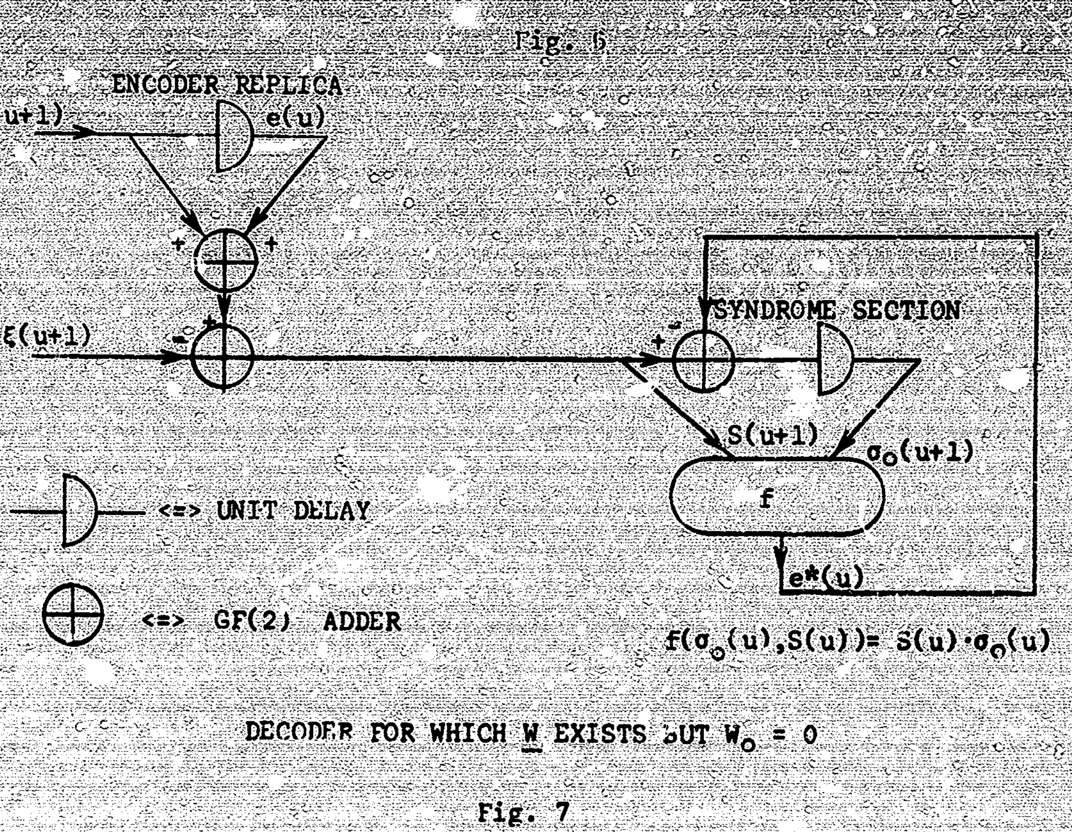
P_{GD}, P_{FD}, and P_{DD} vs. p FOR DECODER OF Fig. 3

Fig. 5



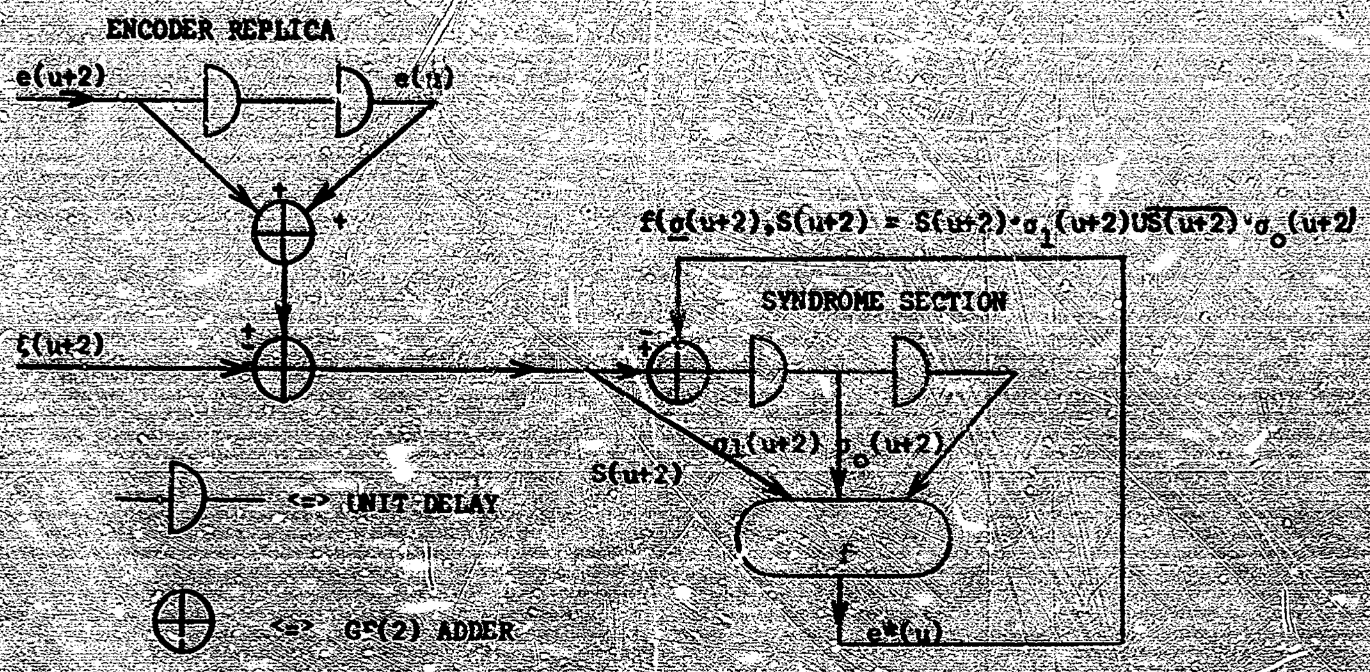
GENERAL, BINARY, $R = \frac{K_o}{N_o}$ SYNDROME FEEDBACK DECODER

Fig. 6



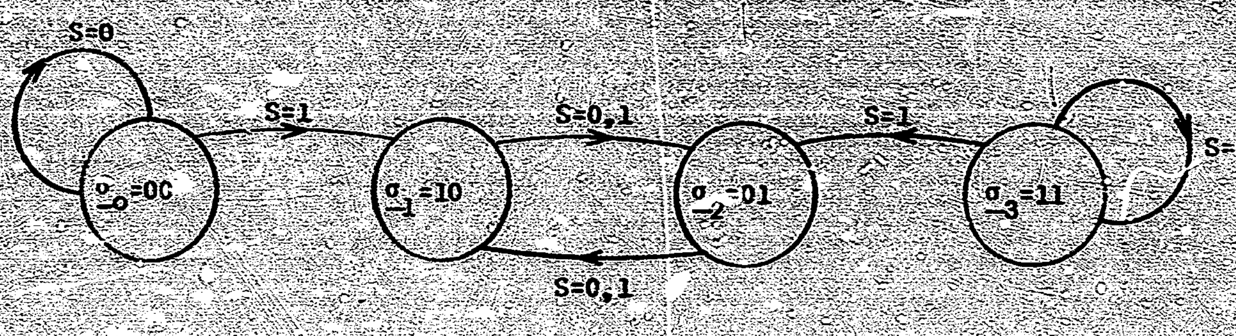
DECODER FOR WHICH $\underline{W}$ EXISTS BUT $W_o = 0$

Fig. 7



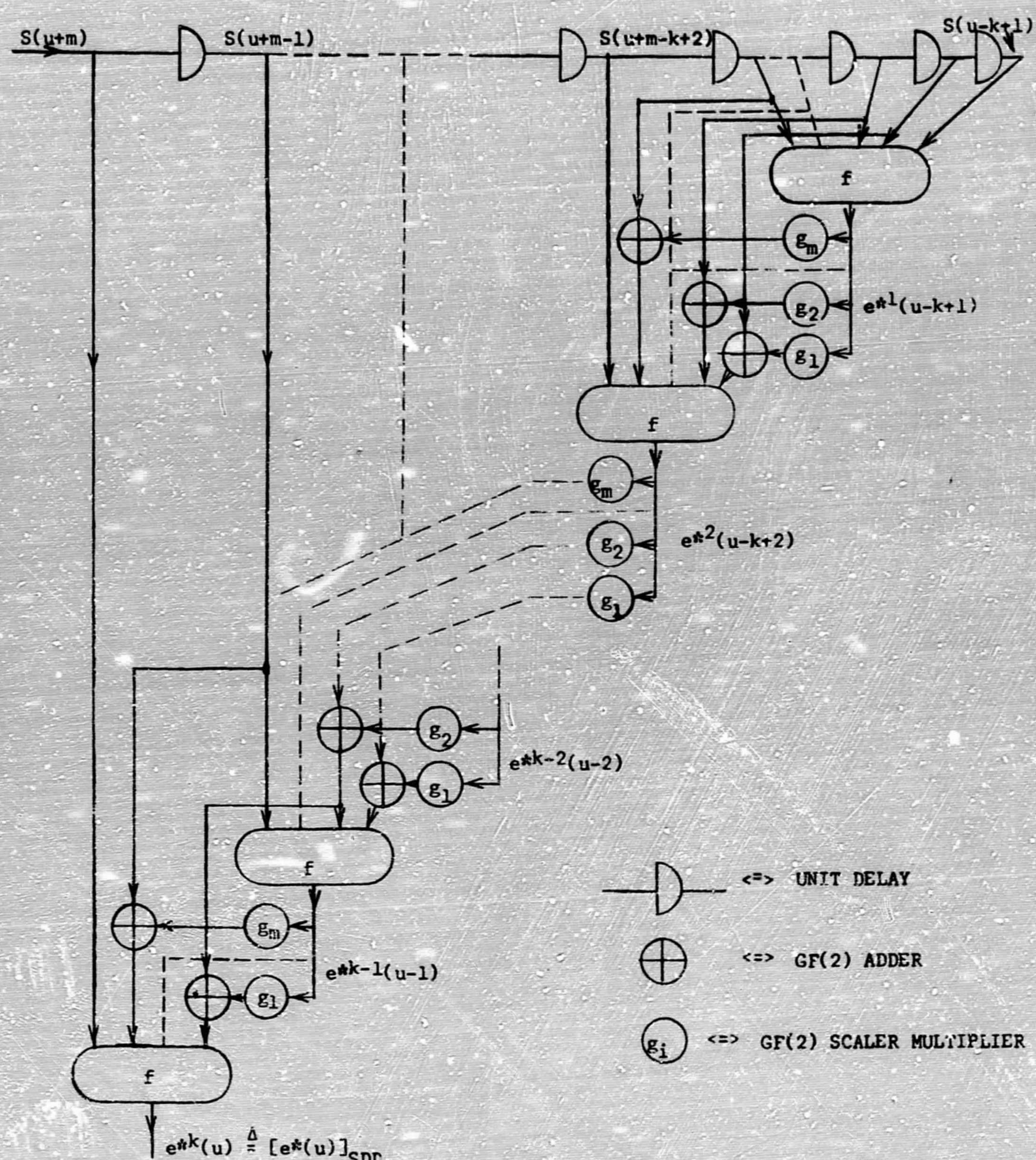
DECODER FOR WHICH NEITHER W NOR P_{FT} EXISTS

Fig. 8



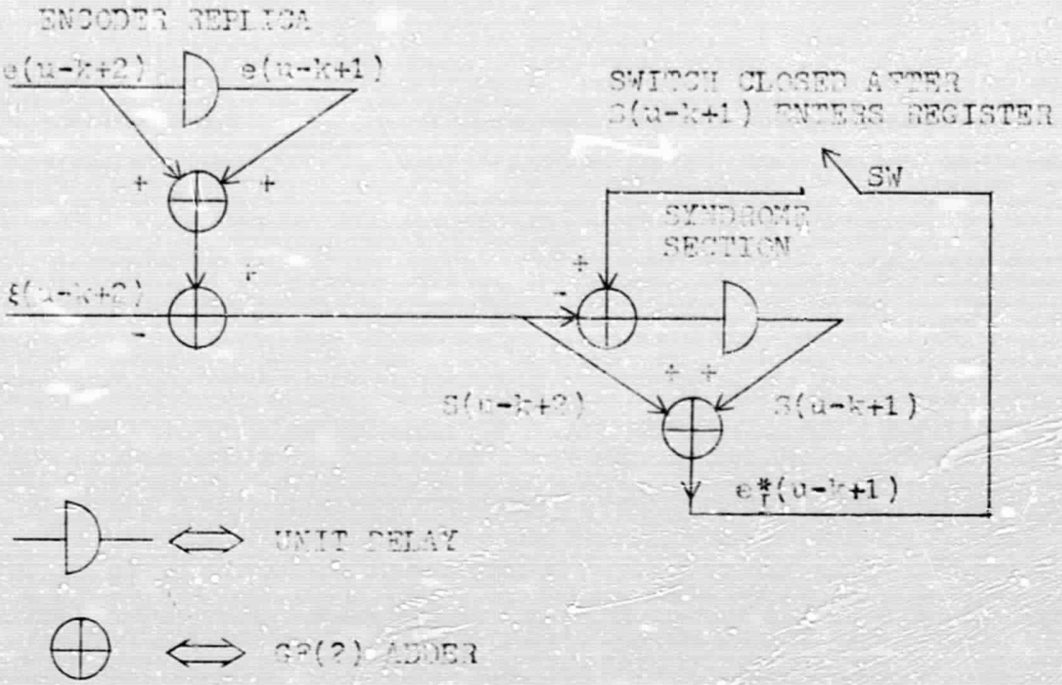
SYNDROME STATE DIAGRAM FOR DECODER OF Fig. 8

Fig. 9



GENERAL, SYSTEMATIC, BINARY, $R = \frac{1}{2}$ SEMI-DEFINITE DECODER OF ORDER k

Fig. 10



EQUIVALENT FEEDBACK DECODER

Fig. 11

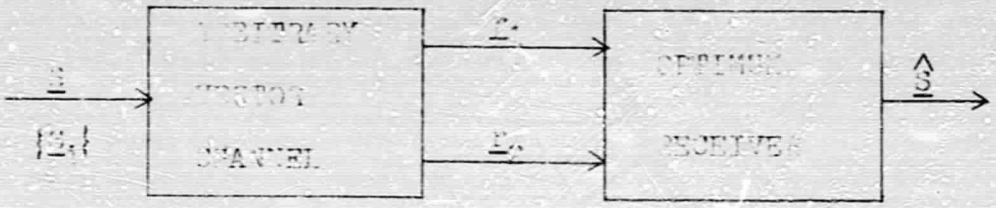
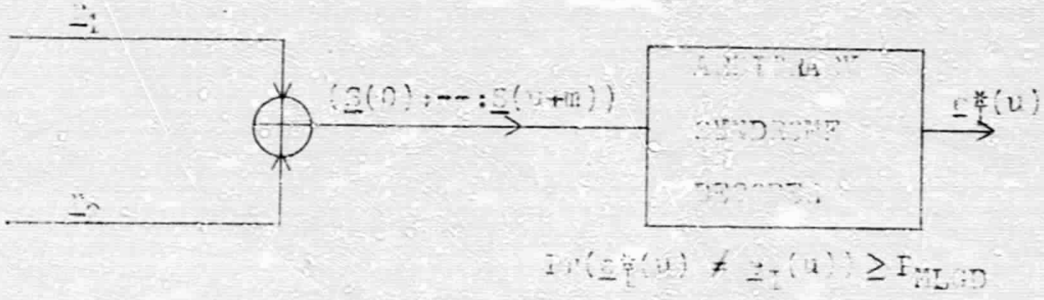


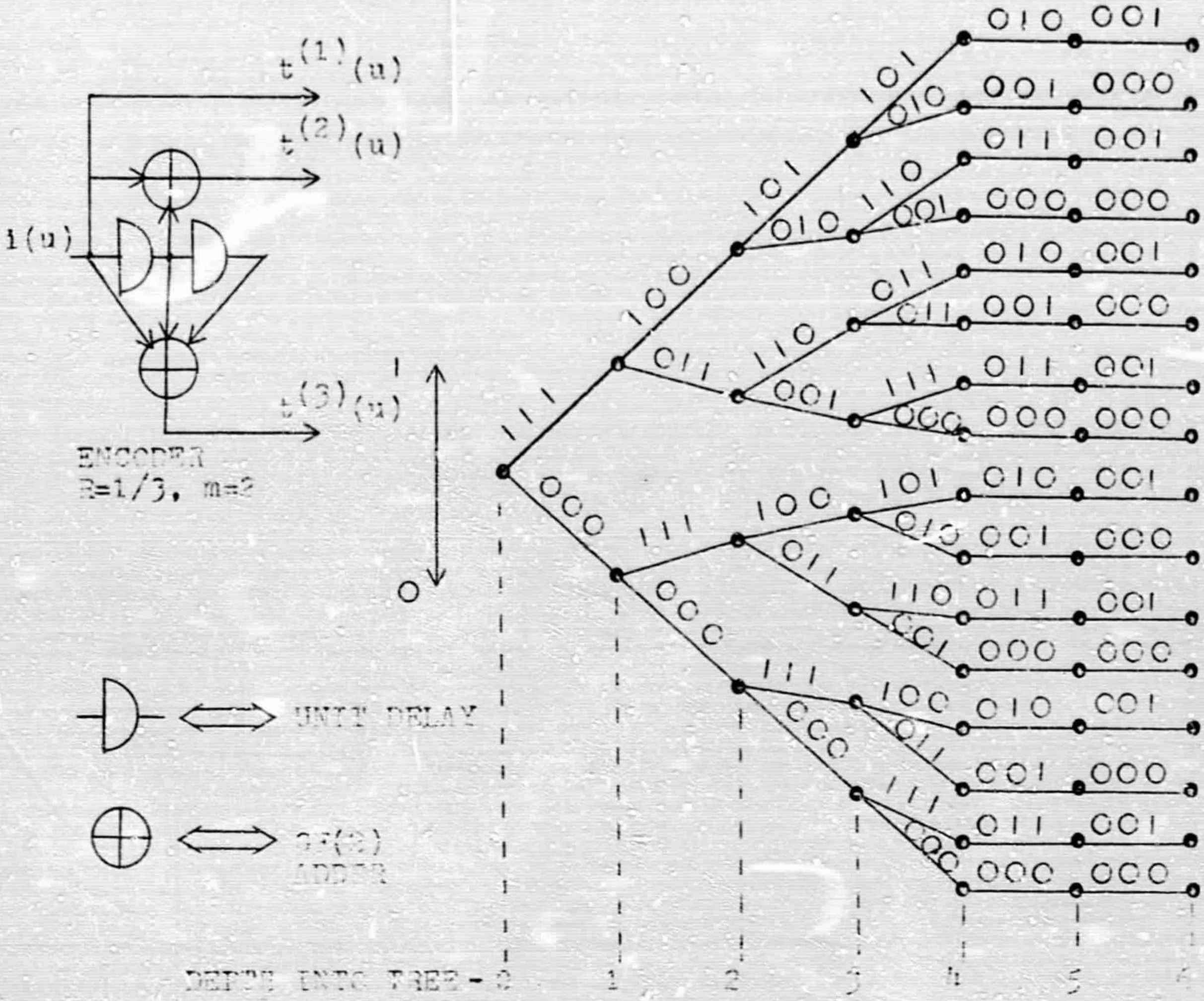
ILLUSTRATION OF THE THEOREM OF IRRELEVANCE

Fig. 12



$Pr(e_1^*(u) \neq e_1(u)) \geq P_{MLSD}$

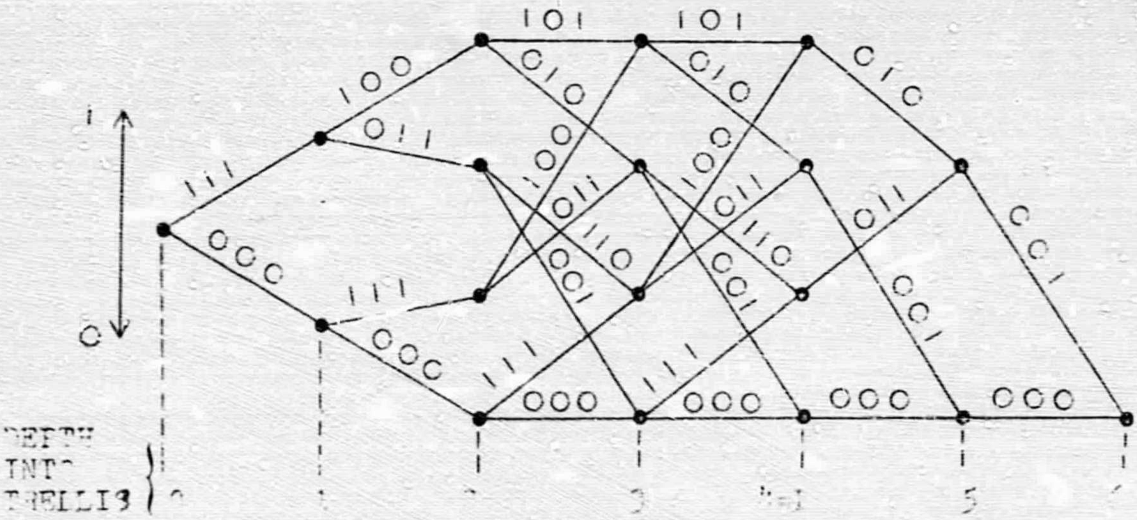
Fig. 13



CODE TREE, J = 4

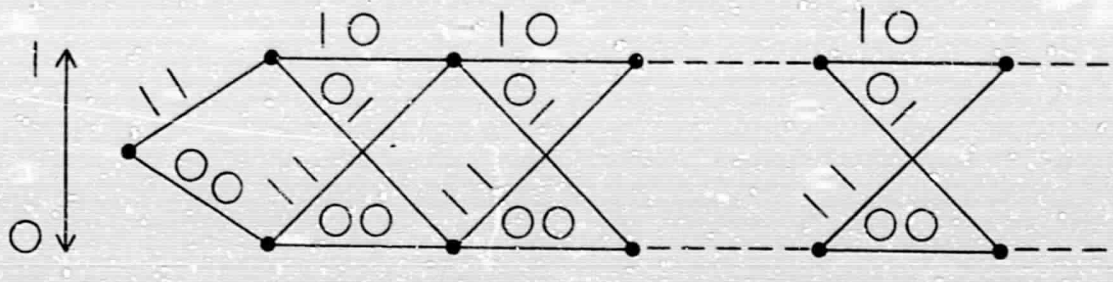
ENCODER WITH ASSOCIATED CODE TREE

Fig. 14



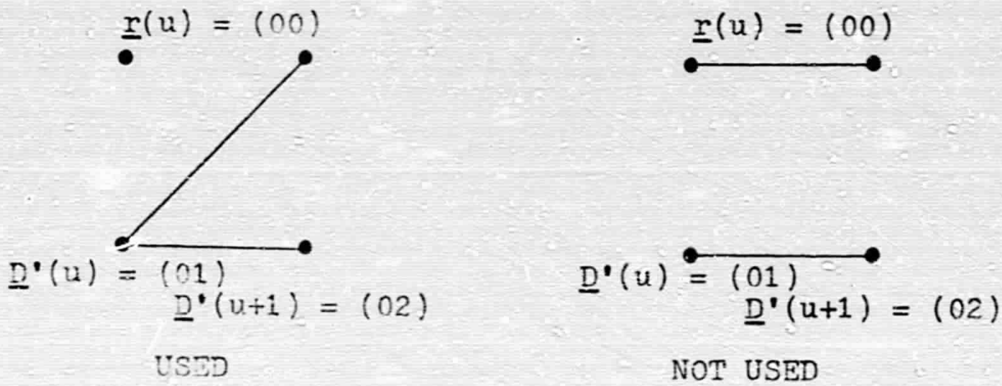
TRELLIS FOR ENCODER OF FIG. 14

Fig. 15



TRELLIS FOR BINARY, $R = \frac{1}{2}$, SYSTEMATIC CONVOLUTIONAL CODE WITH $m = 1$, $g_{o(1)}^{(2)} = 1$, AND $g_{1(1)}^{(2)} = 1$.

Fig. 16



TIE-BREAKING RULE USED BY VITERBI DECODER IN DECODING TRELLIS OF FIG, 16

Fig. 17