Effect of a Binary Symmetric Channel on the Synchronisation Recovery of Variable Length Codes

M. RAHMAN AND S. MISBAHUDDIN

Department of Electrical Engineering, King Fahd University of Petroleum & Minerals, Box 377, Dhahran 31261, Saudi Arabia

Optimal source codes are usually required to have variable length code words. When these code words are transmitted through a noisy channel, the decoder at the receiving end suffers loss of synchronisation caused by one or more erroneous received bits. The expected error span between loss and recovery of synchronisation is evaluated as a function of the channel cross over probability.

Received February 1988, revised August 1988.

1. INTRODUCTION

Variable length codes such as Huffman codes[1] are known to be minimum redundancy codes. By using variable length coding schemes, relatively fewer digits, on the average, are required to convey the same amount of information.[2] Digital data transfer rates are therefore increased. An inherent limitation is always associated with such coding schemes; this is the loss of synchronisation[3] of the decoding circuit due to single or multiple bit inversions.[4] Several researchers have investigated the synthesis of variable length codes having synchronising code words.[5,6,8] However, the effects of a Binary Symmetric Channel (BSC) on the synchronisation recovery process have not been examined. The work described in this paper evaluates the expected error span associated with decoding variable length codes transmitted through a BSC. Furthermore, the effects of crossover transition probability of a BSC on percentage loss of source characters are also investigated.

Example 1

We illustrate the loss of synchronisation with the following variable length code used in Ref. 7.

Table 1

Symbol	Probability	Codeword
A	0.4	00
B	0.2	01
C	0.2	10
D	0.1	110
E	0.1	111

Consider the sequence ABAEBCD corresponding to the bit stream 0001001110110110. Let us assume that the 3rd bit is inverted. It can be verified that the decoded sequence is given by ADBDDD. The decoder is out of synchronisation after the first character and incorrectly decodes the next 5 characters before regaining synchronisation. The average number of original source characters which are incorrectly decoded is defined as the error span. Maxted and Robinowitch[7] have developed a state space approach for evaluating the error span for a random single bit inversion. This approach is briefly described in the next section.

2. ERROR RECOVERY OF VARIABLE LENGTH CODES

2.1 Error state diagram

According to the approach described in Ref. 7, the synchronisation recovery process is described in terms of a finite number of decoder states. To begin with the decoder is in the INITIAL state in which no loss of synchronisation has yet occurred. When a codeword is received in which a single bit chosen at random has been inverted, the decoder may lose synchronisation, ending in one of the error states. The error states are defined as all the valid prefixes of all the codewords.

Obviously these prefixes cannot be codewords in a uniquely decodable code. The decoder may now go through several transitions between various error states before reaching SYNC state. To evaluate the expected error span, all transition probabilities, from initial state to any other state and then from any error state to any other state, are determined. These probabilities are summarised in a transition probability matrix. A state diagram is then obtained from this matrix. The gain along each edge is the probability of the state transition. Fig. 1 shows such a state diagram shown in Ref. 7 for the example given earlier. The indeterminate z in Fig. 1 corresponds to one source symbol.

$G(z)$, the gain of the state diagram is the generator function of the probability distribution of the error span, Ref. 7. Therefore, the expected error span defined as the average number of source characters lost due to loss of synchronisation, is given by:

$$E_S = G'(z) \quad \text{at} \quad z = 1.$$

This method of estimating the error span is inefficient in the sense that the state diagram for a source with a large alphabet would be complex and calculation of the transfer function of such a state diagram will be tedious.

2.2 Conditions for synchronisation recovery

The expected error span concept is meaningful if and only if the decoder can recover synchronisation eventually, i.e. the probability of being in the SYNC state in k steps or less should converge to unity as $k \to \infty$. Not all variable length codes will automatically resynchronise after an initial loss of synchronisation. Here we briefly discuss the necessary and sufficient conditions for recovery, in terms of the properties of the transition probability matrix T, defined as follows:

$$T = \begin{bmatrix} t_{11}\, t_{12} & \cdot\cdot & t_{1N} \\ \cdot & & \\ \cdot & & \\ t_{N1}\, t_{N2} & \cdot\cdot & t_{NN} \end{bmatrix},$$

where $t_{i,j}$ is the probability of going from state i to state j. There are a total of N states, including the initial state $I = 1$ and the final synch state $S = N$. From an examination of Fig. 1, we have

$$t_{i,1} = 0, \quad i = 1, 2, .., N, \tag{1}$$

$$t_{N,j} = 0, \quad j = 1, 2, .., N-1, \tag{2}$$

$$t_{N,N} = 1, \tag{3}$$

$$\sum_{j=1}^{N} t_{ij} = 1, \quad i = 1, 2, .., N. \tag{4}$$

Equation (4) states that the sum of the transition probabilities from any state must be unity. It is possible that t_{iN} may be equal to zero for some values of i, i.e. The SYNC state cannot be reached directly from state i. However, re-synchronisation is possible in a statistical sense as long as the SYNC state can eventually be reached from any state after a finite number of transitions. Therefore, a necessary and sufficient condition for resynchronisation can be stated in the following Lemma.

Lemma 1

Consider the matrix T^r obtained by successive multiplication with T with itself. If for a finite value of r, T^r contains non-zero entries in the last column, then the receiver will recover synchronisation, in a statistical sense.

For the example given in Fig. 1, the smallest value of r is 1 which implies that the SYNC state can be reached from any state in one step. We now prove the following lemma which will be required to show that the numerical algorithm given later will converge.

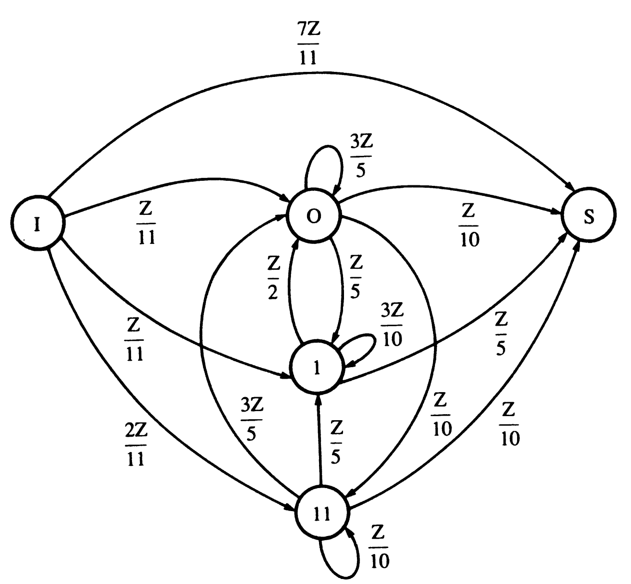

Figure 1. Error state diagram for example 1.

Lemma 2

Consider $t^k_{1,N}$, the last element of the first row of T^k. If the matrix T satisfies the conditions stated in Lemma 1, then

$$\underset{k\to\infty}{Lt}\; t^k(1, N) = 1.$$

Proof $\qquad T^k = T^{k-r}T^r.$

Therefore, $\qquad t^k_{i,N} = \sum_{j=1}^{N} t^{k-r}_{1,j}\, t^r_{j,N}$

$$= t^{k-r}_{1,N} + \sum_{j=1}^{N-1} t^{k-r}_{1,j}\, t^r_{j,N},$$

Since $\qquad t^r_{N,N} = 1.$

The second term is strictly non-zero since $t^r_{j,N} > 0$ for all j, where r is the smallest integer satisfying lemma 1. Thus,

$$t^k_{1,N} > t^{k-r}_{1,N}, \quad k > r.$$

Therefore $t^k_{1,N}$ is monotonically increasing with respect to k.

However, the sum of the elements of the first row (or any other row) must always be equal to unity according to equation (4). Therefore, increase of the last element of the first row, must be at the expense of the other elements in that row. Since successive values of $t^k_{1,N}$ must always be increasing, eventually this must converge to unity, as $k \to \infty$. Furthermore, all the other terms must converge to zero.

The implication of lemma 2 is that the probability of reaching sync state after an infinite number of transitions is unity, provided the conditions stipulated by lemma 1 are satisfied.

2.3 Extended error state diagram

We now present a numerical technique which uses the extended model of the state diagram. First, we look at a random single bit inversion and later a similar approach will be used for multiple bit inversions.

The error state diagram obtained by the transition probability matrix could be extended into a semi infinite state diagram. The error state diagram shown in Fig. 1 depicts the transition probabilities after one symbol. Now, if we wish to calculate the transition probabilities from one state to another state at the end of many symbols, we expand the state diagram in the time domain as shown in Fig. 2. Each row of states corresponds to a certain time, say the ith symbol time and from each of these states any of the other states may be reached with certain probability at the end of $(i+1)$th symbol time. It is possible that the decoder will switch between different error states and take a large number of steps to become synchronised.

From the semi infinite error state diagram it is obvious that the decoder of the variable length coding scheme can switch between the initial state I to any one of the n error states with some transition probability obtained from the transition probability matrix.

Let $P(\varepsilon_i, m)$ be the probability of being in error state ε_i in m steps, i.e. m symbols after loss of synchronisation and $P(S, m)$ is the probability of reaching the sync state in exactly m steps. The probabilities in the extended error state diagram can easily be calculated from the transition

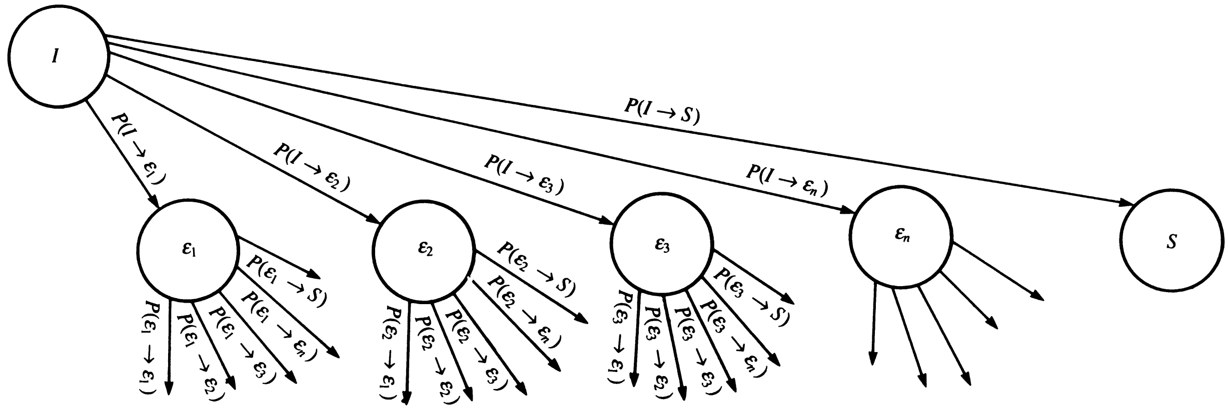

Figure 2. Segment of a service – infinite state diagram.

matrix. If $P(\varepsilon_k, m)$ is the probability of kth error state in mth step, it can be calculated from $(m-1)$th step probabilities as follows:

$$\left.\begin{aligned} P(\varepsilon_k, 1) &= P(I \rightarrow \varepsilon_k), \\ P(\varepsilon_k, m) &= \sum_{j=1}^{n} p(\varepsilon_j, m-1)\, P(\varepsilon_j \rightarrow \varepsilon_K), \quad m > 1, \end{aligned}\right\} \tag{5}$$

where $P(\varepsilon_j \rightarrow \varepsilon_k)$ is the transition probability from jth error state to the kth error state, n is the total number of valid error states of the variable length code.

The probability $P(S, k)$ of reaching the sync state in exactly k steps can be calculated from the $(k-1)$th state probabilities and their corresponding transition probabilities as follows:

$$\left.\begin{aligned} P(S, 1) &= P(I \rightarrow S) = t_{1,n}, \\ P(S, k) &= \sum_{j=1}^{n} p(\varepsilon_j, k-1)\, P(\varepsilon_j \rightarrow S), \quad k > 1. \end{aligned}\right\} \tag{6}$$

If the conditions for synchronisation recovery are satisfied as discussed in the earlier section, then

$$P(S) = \sum_{i=1}^{\infty} P(S, i) = \mathop{Lt}_{k \rightarrow \infty} t^k(1, N) = 1.$$

From a numerical point of view, the probabilities $P(S, k)$ may be computed up to r steps such that the following inequality is satisfied

$$1 - \sum_{k=1}^{r} P(S, k) \leqslant E, \quad E \ll 1. \tag{7}$$

In this work E was chosen to be 10^{-5}. Any further decrease in the order of E does not make any noticeable change in the value of the expected error span E_S, which can now be expressed as

$$E_S = \sum_{i=1}^{r} iP(S, i) \tag{8}$$

E_S is the number of characters, on the average, which will be 'lost' and replaced by other erroneous characters. In general the number of these erroneous characters will not be the same as the number of the lost characters.

The numerical method described above for calculating the expected error span has been tested for various variable length codes including variable length codes for the 26-letter English alphabet. The results are in close agreement with those obtained in Ref. 7 using the transfer function approach. This numerical approach will now be used to investigate the error recovery process when the codewords are transmitted through a BSC.

3. EFFECT OF A BSC ON ERROR RECOVERY

Multiple bit inversions commonly occur during data transmission through a noisy channel. In Section 2 we have examined the decoder's error recovery behaviour due to single bit inversion. The analysis of error recovery becomes more complicated when multiple bit inversions occur.

In the case of single bit inversion it has been observed that the decoder remains out of synchronisation and keeps producing invalid characters until resynchronisation is achieved. When data is transmitted through a noisy channel, a bit error will not necessarily initiate loss of synchronisation since the decoder may already be out of synchronisation. However, if an erroneous bit is encountered while the decoder is trying to regain synchronisation, the error recovery process will be affected. Error recovery may be accelerated or retarded depending on the location of bit inversions, the state of the decoder etc. In a BSC, every bit has a non-zero probability of being inverted. Therefore, if variable length code words are transmitted through a BSC, the loss of synchronisation at the receiver due to inversion of multiple bits will affect the error recovery process.

In this section, we extend the techniques of the previous section to incorporate the effect of multiple errors. The probabilities of such error patterns may be computed from the properties of the BSC and these values would be used to modify the transition probabilities of the extended state diagram of the error recovery process. The expected error span during any cycle of non-synchronisation is obtained as a function of p, the bit error probability of the BSC.

Finally, we derive an expression for calculating the percentage of source characters (symbols) lost (and replaced by erroneous characters) due to incorrect decoding.

We shall be using the following notations in the following discussions.

LOS = Loss of synchronisation.
ROS = Recovery of synchronisation.
E_S = Expected error span between LOS and ROS.
p = Bit error probability of the BSC.

3.1 Initial transition probabilities

Assume that at some point in the decoding process, an incorrect bit has caused loss of synchronisation. The decoder remains in LOS mode until the last bit of the decoded bit coincides with the last bit of the correct sequence. During the LOS cycle there may be additional bits in error but there is no new LOS cycle since the decoder is already in that mode. Let us assume we are in the initial state, i.e. there has been no inversion of any bit so far. Consider the first error bit in a certain code word which causes LOS. Obviously the decoder will now go to some error state. However, given that a corrupted received word has caused LOS, there may be one or more errors in that received word. Depending on the exact probabilities of these multiple error patterns, the transition probabilities of the state diagram are now modified in the following way.

Let X_m be the mth variable length code word generated by a transmitting system. Let L_m be its length and P_m be its a-priori probability, Let ϕ_{mk} be the probability of a specific k error pattern, given that a code word has been received incorrectly and it is given by:

$$\phi_{mk} = \frac{p^k(1-p)^{L_m-k}}{1-(1-p)^{L_m}}, \quad k = 1, 2, \ldots, L_m. \tag{9}$$

The number of k error patterns in L_m is determined by the binomial co-efficient given by:

$$\delta_{mk} = \binom{L_m}{k} = \frac{L_m!}{(L_m-k)!\,k!}.$$

An algorithm is presented here which generates the initial state transition table for the multiple bit inversion case.

Algorithm 1

(1) Choose $X_m, m = 1, 2, \ldots, r$.
(2) Generate δ_{mk} bit error pattern for $k = 1, 2, \ldots, L_m$.
(3) Determine probabilities $\phi_{mk} p_m$ for the bit pattern generated in step (1).
(4) Decode the bit pattern generated in step (2) and determine the next state.
(5) Repeat (1)–(4) for next X_m.
(6) End of algorithm.

The decoded bit pattern obtained from step (4) would either end in the Sync state after the decoder makes one incorrect decision or in one of the error states. An initial entry table for multiple bit inversion case is now generated.

The initial entry table is generated with the assumption that a codeword has been received incorrectly due to single or multiple bit errors. Subsequently there may be additional errors until ROS.

3.2 State transition probabilities

The state transitions and the corresponding probabilities, from any error state to any other error state, including itself and the sync state can be determined by the following algorithm.

Algorithm 2

(1) Start with the ith error state.
(2) Append the code word X_m with length L_m.
(3) Consider all possible δ_{mk} error patterns of length k, $k = 0, 1, \ldots, L_m$.
Decode the resulting sequence. Calculate the probability of each event and the next state.
(4) If all code words are exhausted go to 5, otherwise repeat 2 and 3 for the next code word.
(5) If all error states are exhausted go to 6, otherwise, repeat step 1–4.
(6) End of algorithm.

Example 2

This example illustrates the application of Algorithms 1 and 2. The variable length code used in this example is given in Example 1.

The valid error states for this code are 0, 1 and 11. Application of algorithms 1 and 2 yield the following probability transition matrix T (Table 2) for $p = 0.1$.

3.3 Expected error span for multiple bit inversions

Once the transition probability matrix for the given variable length codes has been generated, given that LOS has been initiated, the expected error span between LOS and ROS can be determined. The entries in the transition probability matrix will of course vary with variation in p. In other words as the BSC characteristics change the error span between LOS and ROS will also vary.

To determine the expected error span between LOS and ROS equations 5–8 may again be used to calculate $P(\varepsilon_k, m)\,P(S, m)$ and E_S. The only difference is that the transition probabilities $P(I \to \varepsilon_j)$ and $P(\varepsilon_j \to \varepsilon_k)$ are obtained from Algorithms 1 and 2. The error span for various codes was evaluated using this numerical approach for different values of p. The results are shown in Figs (3)–(5). Fig. 3 shows the error span for the code given in example 1, while Fig. 4 shows the results when the same source is encoded differently, e.g. A = 01, B = 00, C = 11, D = 100 and E = 101. Fig. 5 shows the expected error span of a variable length Huffman code for the 26-letter English alphabet, modelled as a zero-memory source with source probabilities given in Ref. 7.

A close inspection of Figs 3 and 4 reveals that the

Table 2

		I (Initial)	0	1	11	S (Synch)
	I (Initial)	0	0.070110	0.070111	0.210526	0.649252
	0	0	0.560000	0.240000	0.090000	0.110000
T =	1	0	0.450000	0.290000	0.009000	0.251000
	11	0	0.560000	0.240000	0.090000	0.110000
	S (Synch)	0	0	0	0	1

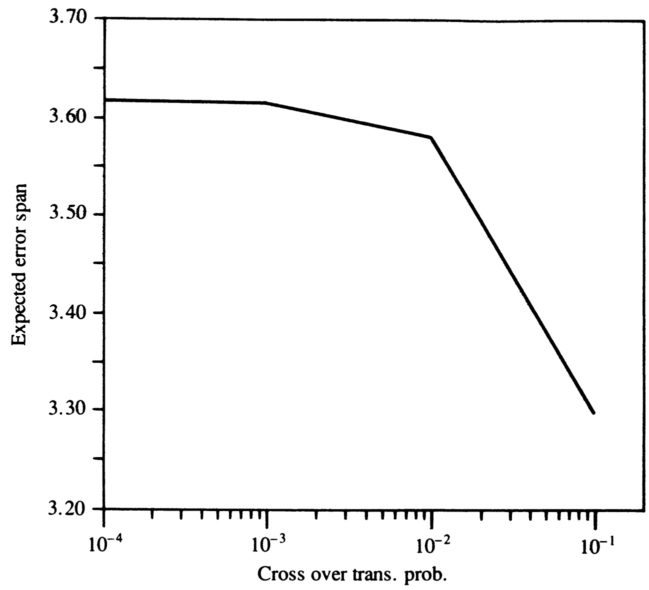

Figure 3. Expected error span vs. P.

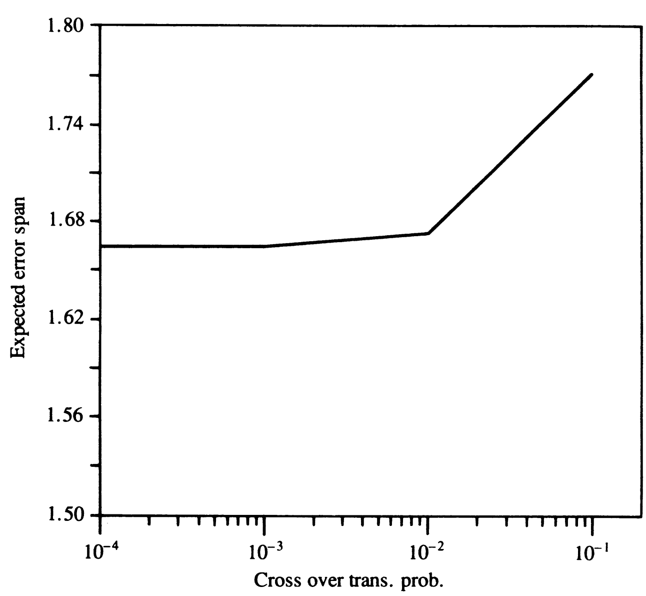

Figure 4. Expected error span vs. P.

expected error span is not a monotonically increasing or decreasing function of p. Each curve depends on the specific code and no general conclusions can be made about the exact relationship between E_S and p of the BSC.

4. EFFECT OF BSC ON PERCENTAGE LOSS OF CHARACTERS

Let us consider the transmission of variable length code words and the first occurrence of a received word being corrupted. The decoder loses synchronisation and begins producing erroneous characters. This is continued until the last bit of a decoded sequence coincides with the last bit of the correct code sequence. The situation between LOS and ROS is depicted in Fig. 6, which shows that on the average there are E_S characters within an error span.

The average number of errors embedded within an error span is calculated by considering a long span of

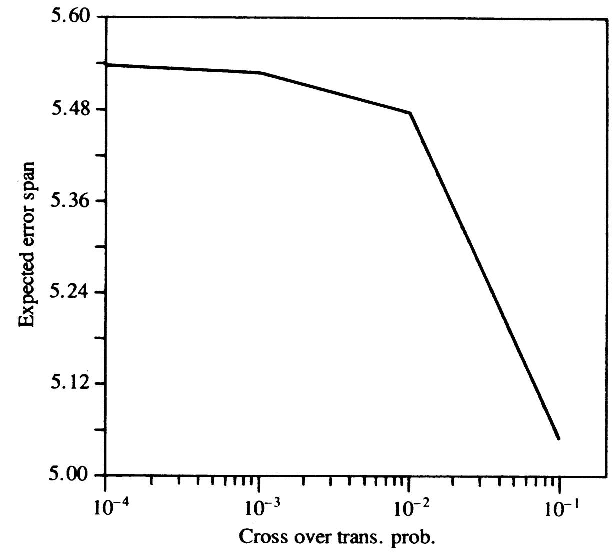

Figure 5. Expected error span vs. P.

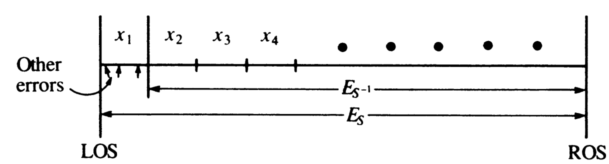

Figure 6. Error span.

received bits of length N. We define additional parameters in the following way:

N_e = Total number of errors within a span of N bits.

N_c = Total number of code words in this span.

N_S = Total number of errors which initiate LOS.

N_L = Total number of original source characters which are incorrectly decoded.

A_v = Average length of a given variable length code.

$$\eta = \frac{\text{number of incorrectly decoded source characters}}{\text{Total number of source characters}} \times 100.$$

From the law of large numbers we have

$$N_e = N^p. \tag{10}$$

In Fig. 6 it can be seen that LOS is initiated due to the first erroneous bit of the character X_1. During the span between LOS and ROS there may be additional bit inversions, but they do not initiate a new cycle of LOS and ROS. The total number of errors in each error span can be calculated as follows.

The total number of errors within the span of length $E_S - 1$ (i.e. excluding the first erroneous character) is given by the following equation.

$$N'_e = (E_S - 1) A_V p. \tag{11}$$

Therefore, the total number of errors in E_S is given by

$$N_{et} = (E_S - 1) A_v p + \sigma. \tag{12}$$

Where, σ is the average number of errors in the corrupted code word which has initiated LOS. This can be evaluated in the following way.

Let $X_1, X_2, \ldots, X_r$ be the variable length codewords. Let α_K^m denote the conditional probability of k errors in the mth character X_m (conditional on X_m being incorrectly received). By using Bayes' theorem the conditional probability of k errors in X_m given that error has occurred is given by:

$$\alpha_k^m = P(k \text{ errors in } X_m/\text{error}) = \frac{P(k \text{ errors in } X_m)}{P(\text{error in } X_m)}.$$

Therefore,

$$\alpha_k^m = \frac{\binom{L_m}{k} p^k (1-p)^{L_m - k}}{1-(1-p)^{L_m}}, \tag{13}$$

where $\binom{L_m}{k}$ is the number of different ways of making k errors in a length of L_m.

The average number of errors in X_m given that an error has occurred is therefore given by

$$\beta_m = \sum_{k=1}^{L_m} k \alpha_k^m. \tag{14}$$

The average number of errors in the first corrupted codeword can be evaluated by calculating the weighted average of β_m

$$\sigma = \sum_{m=1}^{r} P(X_m)\beta_m. \tag{15}$$

Now by simple counting arguments and using equations (10)–(15), Ref. 4, we may establish the following equations:

$$N_S = \frac{Np}{\sigma + (E_S - 1) A_v P} \tag{16}$$

$$N_L = \frac{NpE_S}{\sigma + (E_S - 1) A_v p} \tag{17}$$

and $$\eta = \frac{N_L}{N_c} \times 100 = \frac{pE_S A_v}{\sigma + (E_S - 1) A_v p} \times 100. \tag{18}$$

Figure (7) shows η vs. p for the code given in example 1. As the channel gets progressively worse with increase of p, the percentage of lost characters increases as would be expected.

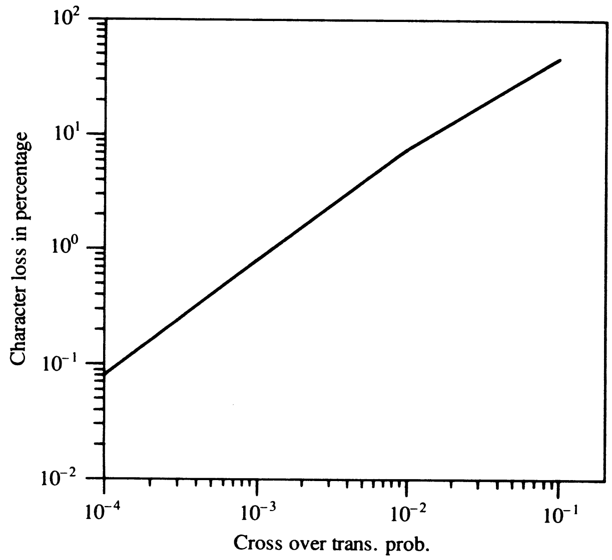

Figure 7. Percentage character loss vs. P.

5. CONCLUSION

Various researchers[5, 6, 8] have contributed to the synthesis of variable length codes having inherent error recovery properties. The problems associated with decoding such variable length codes after they have been transmitted over a BSC have received little attention. In this paper we have discussed the error recovery behaviour of the decoder of a receiving system of a digital communication system when variable length codes are transmitted through a Binary Symmetric Channel. The following concluding remarks may be made.

(*a*) The expected error span for a given variable length code for a single bit error may be determined by using the numerical approach developed in section 2. The numerical approach is desirable especially for the case of a large alphabet.

(*b*) The same approach can be used for evaluating the error span when the variable length codewords are transmitted through a BSC. However, the probability transition matrix has to be evaluated by taking into account that any transmitted bit may be inverted with probability p, as shown in section 3.

(*c*) The degradation in the system performance is caused by both channel errors and synchronisation errors. Furthermore, for a given code, synchronisation error measured in terms of E_S may increase or decrease as a function of p and depends on the specific set of codewords.

Acknowledgement

The authors would like to express their thanks to the reviewer for his helpful comments.

The authors would also like to express their appreciation to the King Fahd University of Petroleum & Minerals for its support of this research.

REFERENCES

1. D. Huffman, A method for construction of minimum redundancy codes. *Proceedings Institute Radio Engineers*, **40**, 1098–1101 (1952).
2. R. W. Hamming, *Coding and Information Theory*. Englewood Cliffs, NJ, Prentice Hall (1980).
3. J. J. Stiffler, *Theory of Synchronous Communication*. Prentice Hall, Englewood Cliffs, NJ (1971).
4. Syed Misbahuddin, 'An Investigation of Error Recovery of Variable Length Codes', *MS thesis*. Department of Electrical Engineering, King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia, January, 1988.
5. P. G. Neuman, On a class of efficient error limiting variable length codes. *IRE Transactions Information Theory*, **IT-8** (1962).
6. T. J. Ferguson and J. H. Robinowitz, Self Synchronizing Huffman Codes. *IEEE Transactions Information Theory*, **IT-30**, 687–693 (1984).
7. J. C. Maxted and J. H. Robinowitz, Error recovery of variable length codes. *IEEE Transactions Information Theory*, **IT-31, 6**, 7794–801 (1985).
8. E. N. Gilbert and E. F. Moore, Variable length binary encoding. *Bell systems Technology Journal*, **38**, 933–967 (1959).