Capacity of Insertion and Deletion Channels

David Leigh

July 23, 2001

Abstract

Theoretical estimates for lower bounds on the capacity of various insertion and deletion channels are calculated by considering the rates achievable using convolutional codes and sequential decoding. It is found that these limits fall below the rates that can be achieved by the use of watermark codes over similar channels. The sequential decoding procedure used is shown to be suboptimal, but it is expected that a decoder which was able to take account of the phase of received and transmitted sequences would lead to an improved lower bound estimate for the capacity. However, it is unknown how this would compare to the capacity estimation that watermark codes give.

1 Introduction

The problem of reliable communication over a noisy channel can be broken down into several steps. The sender has information that they wish to convey to the receiver. To do this the data is first put into compressed binary form, reducing the problem to that of passing a string of ones and zeros over the channel and recovering it at the other end. A noisy channel will insert errors into this data so the received string differs from what was sent. In order to be able to recover the original data, the message is encoded, adding degeneracy to it. This added degeneracy increases the number of bits that need to be transmitted, and hence reduces the rate of communication possible over the channel. The method of encoding is chosen so that the distortion introduced by the noisy channel is small enough that the original data can be recovered. For a noisy channel, there exists a non zero rate of transmission called the channel capacity below which reliable communication is possible. Transmission at rates greater than the capacity will result in errors after decoding.

This project looks at theoretical methods for calculating the capacity for channels that introduce synchronisation errors. Synchronisation errors are insertions and deletions of symbols, meaning the received string can differ in length to the transmitted string. Synchronisation errors can occur in channels where the time of arrival of the transmitted bit is unknown, examples include serial lines where the clock speed of the transmitter may not be accurately known and hard disks, where the there may be variations in the rotation speed.

Previous work on channels with insertion and deletion errors using seqeuntial decoding has been done by Gallager [1] 1961 and Zigangirov [2] 1969, leading to some theoretical estimates of the channel capacity. This project looks closely at these limits. More recent work by Davey and Mackay [3] uses watermark codes to communicate over channels with synchronisation errors and have shown that these codes are able to reach rates above the bound on channel capacity obtained by Zigangirov, (fig 1).

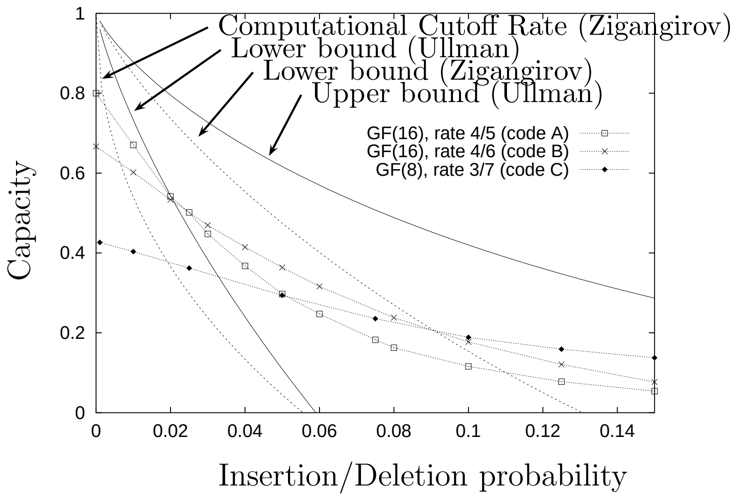

Figure 1: Comparison of estimates of lower bounds on channel for an insertion and deletion channel with zero probability of substitution. Taken from [3].

2 Background

2.1 Synchronisation channel models

The papers by Gallager, Zigangirov and Davey and Mackay all look at different types of insertion and deletion channel and these different channel models are now compared.

2.1.1 Gallager channel model

A bit of information passed across the channel emerges in one of four states. It is transmitted correctly with probability p_t, is deleted with probability p_d, undergoes a bit flip with probability p_s, and is replaced with two random bits with probability p_i. The probability of correct transmission is $p_t = 1 - p_i - p_s - p_d$.

2.1.2 Zigangirov channel model

The channel model considered by Zigangirov allows any number of insertions to occur in the gaps between each transmitted bit. The probability of no insertion is q_1, one insertion occurs with probability $q_1 p_1$, two insertions with probability $q_1 p_1^2$, three with

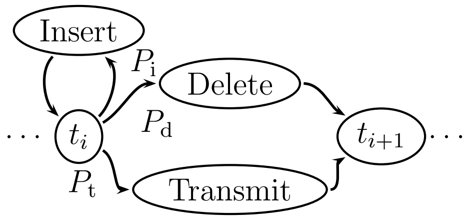

Figure 2: Insertion Deletion channel with probabilities P_i, P_d and P_t of insertions, deletions and transmissions. The 'Transmit' state makes a fraction P_s of substitution errors

probability $q_1p_1^3$ etc. Each inserted digit is either a zero or a one with equal probability. Summing these probabilities gives unity showing that p_1 is the probability of at least one insertion where $q_1 + p_1 = 1$.

As well as insertions occurring in the gaps between transmission, each bit sent over the channel will be deleted with probability p_2, the probability of no deletion is q_2 so that $q_2 + p_2 = 1$. In this model of the channel there is zero probability of a bit flip occurring. This is not a fundamental restriction, but is made to keep the analysis simpler.

2.1.3 Davey and Mackay channel model

This model is similar to that considered by Zigangirov in that any number of insertions between each bit is possible, however bit substitutions can also occur. The symbols to be sent over the channel are queued, waiting to be transmitted. For each channel use, one of three events occurs. With probability P_i a random bit is inserted into the received stream. With probability P_d the queued bit is deleted and with probability $P_t = (1 - P_d - P_i)$ the queued bit is transmitted. During this transmission a bit flip will occur with probability P_s. Figure 2 shows a graphical representation of this channel.

By setting P_s to zero it is possible to compare the probabilities of each outcome for this channel to those for the channel considered by Zigangirov, see table 1. These probabilities can be made equivalent by setting

$$p_1 = P_i \tag{1}$$
$$p_2 = P_d/(1 - P_i) \tag{2}$$

making the two channels identical.

2.2 Sequential Decoding and Tree Codes

As discussed, to reliably transfer data over a noisy channel it is necessary to encode it. The receiver then decodes the noisy version of the signal and recovers the original message. One type of code that can be used to do the encoding is a convolutional code [4]. Convolutional codes have a tree structure as shown in fig 3.

No. Insertions	No.Deletions	Prob (D and M)	Prob (Zig.)
0	0	P_t	q_1q_2
0	1	P_d	q_1p_2
1	0	P_iP_t	$q_1q_2p_1$
1	1	P_iP_d	$q_1p_2p_1$
2	0	$P_i^2P_t$	$q_1q_2p_1^2$
2	1	$P_i^2P_d$	$q_1p_2p_1^2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table 1: Comparison of emission probabilities for the two different channel models as considered by Davey and Mackay and Zigangirov.

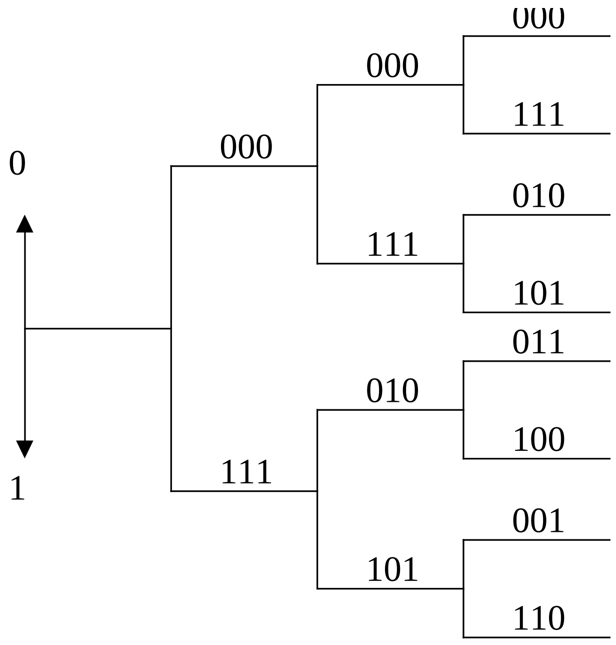

Figure 3: Example of the tree structure of a convolution code, m=3. Taken from [4]

At each node in the tree, the upper branch gives the symbols to be transmitted if a zero is to be sent, and the lower branch those corresponding to a one. Once this is done the encoder follows that branch to the next node and repeats the process. Each branch has the same number of bits and this number can be increased to give reliable communication over noisier channels. The encoder increases the length of the string to be sent by a factor of m, where m is the number of bits in each branch. The rate of a convolutional code is given by $R = 1/m$ bits per transmitted symbol. There are various ways in which the bits on each branch can be chosen, usually the transmitted bits are a linear function of previous source bits. However, it shall be assumed here that each bit is chosen randomly and is equally likely to be a zero or a one.

The job of the decoder is to deduce from the received string which path through the code tree was followed, enabling the original message to be reconstructed. Sequential decoding is a method of doing this. Each successive input bit is considered in turn. The decoder compares the received string to the code tree following whichever branch agrees best. As it goes it keeps a note of how much the path it is following differs to the received string. The measure of this difference is some predefined likelihood function. If the difference becomes greater than some specified threshold, the decoder rejects that path and returns to the previous node to explore a different path. Once this difference becomes smaller than some other specified value, the decoder accepts that its initial choice of which node to follow was correct and decodes the first digit accordingly. The process is then repeated for the next digit.

2.3 Watermark codes

Watermark codes work by identifying where insertions and deletions have taken place. Consider writing a message on a piece of paper with a complicated watermark on. The paper is then distorted during transmission. Provided the watermark is known to the receiver, they can reconstruct the message by aligning the warped watermark to its original state. It is important that the data not be too dense, otherwise it will cover too much of the watermark for it to be reconstructed.

Watermark codes use an outer and an inner code. First the message is encoded with the outer code which should be able to correct for substitution errors. The inner encoding procedure involves this data being sparsified and added, bitwise, modulo two, to a pseudo-random binary sequence, called the watermark, known to both the encoder and the decoder. The received data is a warped and distorted version of the original watermark. It is distorted by the addition of the data and by substitution errors and warped by the insertions and deletions. The decoder then infers where insertions and deletions have taken place by aligning the known watermark to the received string. The difference between these strings is the decoder's estimate as to what the transmitted data was. This is fed into the outer decoder which infers the original message.

In irregular watermark codes, the sparsification is non-uniform and it is expected that irregular watermark codes can outperform regular watermark codes [5].

3 Calculation of channel capacity

Having shown that the channel models considered by Zigangirov and Davey and Mackay are equivalent, we now look to calculate a bound on the channel capacity using tree codes and sequential decoding.

Suppose that the output of the channel is some received string $b_1, b_2, ..., b_{n'}, ...$ and the decoder is attempting to match this sequence to a path through the code tree given by the string of digits $a_1, a_2, ..., a_n, ...$. If we now assume that $n-1$ symbols of the branch have been matched to $n'-1$ symbols of the received string then a matching is defined as an event whereby either the symbol a_n is put into correspondence with the symbol $b_{n'}$, or a_n is deleted, or $b_{n'}$ is deleted. The two symbols can only be put into correspondence if they are identical. Each matching involves a choice between these three options and there are an infinite number of ways of matching the received sequence to a path through the code tree, for instance it possible to choose a method of matching corresponding to every symbol sent being deleted and all the received symbols having been insertions.

The correct method of matching is the one which strictly follows the pattern of insertions and deletions which took place in transmission. This is done by taking the path through the code tree that was transmitted $a_1, a_2, ..., a_n, ...$ and deleting from it all the symbols that dropped out. All symbols inserted during transmission are then deleted from the received string $b_1, b_2, ..., b_{n'}, ...$. The remaining symbols are then put into 1-1 correspondence.

To distinguish between all the different possible methods of matching a probability function is introduced. Assume that the logarithm of the probability function having matched the symbols $a_1, a_2, ..., a_{n-1}$ with the symbols $b_1, b_2, ..., b_{n'-1}$ is z. The next step in the matching process is then the choice of which of the three options to follow. If the matching is such that a_n corresponds to $b_{n'}$ then the logarithm of the probability function is increased by $(\alpha_1 + \beta_1)$. If a_n is deleted then it is increased by $(\alpha_1 + \beta_2)$, and if $b_{n'}$ is deleted it increases by β_1. The initial value of z is set to zero, and this value increases in steps with each matching event.

The sequential decoder then searches through paths in the code tree and calculates the value of z for each method of matching. This process continues until the value for at least one path reaches a (> 0). The decoder then accepts the symbol that corresponds to the first branch of that path as the one which was transmitted. The subtree is then deleted and the logarithm of the probability function is reset to zero and the process repeated to decode the next digit. In addition, the decoder will reject a path and method of matching if the logarithm of its probability function drops below b (< 0).

3.1 Condition for Correct Decoding

We have correct decoding when a path in the correct subtree is accepted and we have an error when a path in the incorrect subset is accepted. There are two conditions which must be met for correct decoding. First, the matching of the correct path to the received string must be such that the logarithm of the probability function reaches the value a before it reaches the value b and second there must be no path in the incorrect subtree which reaches a, otherwise this path may be accepted ahead of the correct one.

In order to ensure that the second criteria is met, it is necessary consider the branches of the incorrect subtree. Consider a subset of these, $G_n(\epsilon)$, with the first n symbols fixed, and the set of methods of matching these symbols with the received sequence fixed so that the logarithm of the probability function, z for the common part of these branches is ϵ. We then calculate $L(\epsilon, n)$, the expectation value of the number of methods of matching belonging to $G_n(\epsilon)$ for which z reaches the threshold value a (giving an incorrect decoding of the symbol) before it reaches b and is rejected. Assuming that we transmit a large number of symbols so that the code tree becomes infinite, we have

$$L(\epsilon, i) = L(\epsilon, i+m) = L(\epsilon, i+2m) = ... \tag{3}$$

where as before m is the number of symbols in each branch. Having matched these n common symbols with $z = \epsilon$, the next operation in the matching process is to choose how to match the next symbol in the received string with the $(n+1)$th symbol in the code tree. As before there are three options and z can be increased by either $(\alpha_1 + \alpha_2)$, $(\alpha_1 + \beta_2)$ or β_1. In considering the incorrect subset, we assume that if the two symbols are identical they are put into correspondence with one another. With probability 1/2 the two symbols will be identical, increasing z by $(\alpha_1 + \alpha_2)$ and with probability 1/2 they will be different. $L(\epsilon, i)$ must be the same before and after this matching operation giving the continuity equations

$$L(\epsilon, i) = \tfrac{1}{2}L(\epsilon + \alpha_1 + \alpha_2, i+1) + \tfrac{1}{2}L(\epsilon + \alpha_1 + \beta_2, i+1) + \tfrac{1}{2}L(\epsilon + \beta_1, i) \tag{4}$$

which is valid for $i = 1, 2, ..., m-1$, and

$$\begin{aligned} L(\epsilon, m) &= 2[\tfrac{1}{2}L(\epsilon + \alpha_1 + \beta_1, m+1) + \tfrac{1}{2}L(\epsilon + \alpha_1 + \beta_2, m+1)] + \tfrac{1}{2}L(\epsilon + \beta_1, m) \\ &= 2[\tfrac{1}{2}L(\epsilon + \alpha_1 + \beta_1, 1) + \tfrac{1}{2}L(\epsilon + \alpha_1 + \beta_2, 1)] + \tfrac{1}{2}L(\epsilon + \beta_1, m) \end{aligned} \tag{5}$$

where the extra factor of two in this equation arises from the fact that there are two branches to consider. The boundary conditions for these equations are

$$L(\epsilon, i) = 0 \quad \epsilon \leq b \tag{6}$$
$$L(\epsilon, i) = 1 \quad \epsilon \geq a \tag{7}$$

To solve this set of equations, look for solutions of the form $L(\epsilon, i) \leq A_i 2^{\lambda\epsilon}$,giving

$$\begin{aligned} A_1 &= \tfrac{1}{2}(2^{\lambda(\alpha 1+\alpha 2)} + 2^{\lambda(\alpha 1+\beta 2)})A_2 + \tfrac{1}{2}2^{\lambda\beta_1}A_1 \\ A_2 &= \tfrac{1}{2}(2^{\lambda(\alpha 1+\alpha 2)} + 2^{\lambda(\alpha 1+\beta 2)})A_3 + \tfrac{1}{2}2^{\lambda\beta_1}A_2 \\ &\vdots \\ A_{m-1} &= \tfrac{1}{2}(2^{\lambda(\alpha 1+\alpha 2)} + 2^{\lambda(\alpha 1+\beta 2)})A_m + \tfrac{1}{2}2^{\lambda\beta_1}A_{m-1} \\ A_m &= (2^{\lambda(\alpha 1+\alpha 2)} + 2^{\lambda(\alpha 1+\beta 2)})A_1 + \tfrac{1}{2}2^{\lambda\beta_1}A_m. \end{aligned} \tag{8}$$

This set of equations is solved by setting the determinant of the system equal to zero and rearranging to give

$$1 = 2^{R-1}2^{\lambda(\alpha_1+\alpha_2)} + 2^{R-1}2^{\lambda(\alpha_1+\beta_2)} + 2^{-1}2^{\lambda\beta_1} \tag{9}$$

Taking values of α and β which give rise to not more than two roots of this equation gives

$$L(\epsilon, i) \leq A_i 2^{\lambda_0 \epsilon} + B_i 2^{\lambda_1 \epsilon}, \tag{10}$$

which is an expression for the expected number of methods of matching a path in the incorrect subtree to the received sequence which can give rise to incorrect decoding. The constants A_i and B_i can be calculated from the boundary conditions.

From equations (4) and (5), and applying the boundary conditions (6) and (7), it is possible to obtain the following bound,

$$L(0,0) \leq C\ 2^{-\lambda_0 a} \tag{11}$$

where C is some constant not dependent on a. $L(0,0)$ is the maximum number of expected ways that can give rise to incorrect decoding so for the decoder to work it is required that this be zero. As a is positive and can be made as large as required, the condition that no branch in the incorrect subset of the code tree be accepted reduces to having λ_0 be real and positive. Therefore equation (9) must have positive real roots.

3.2 Condition for decoding to finish

The second criteria that must be met if the decoder is to be successful is that analysis of the correct subset of the tree be such that the correct branch is accepted, that is that z reach the value a (and the branch be accepted) before it reaches b (and the branch is rejected).

The logarithm of the probability function, z is initially set to zero and increases after each matching event in one of three ways. For the correct path, the probabilities of it taking these values is known. A correct transmission of a digit occurs with probability q_1q_2, the correct method of matching will therefore increase by $(\alpha_1 + \alpha_2)$ with this probability. At least one insertion occurs during transmission with probability p_1 which is the probability that z will increase by $(\alpha_1 + \beta_2)$. Subtracting these probabilities from one gives the probability that z increases by β_1, which is q_1p_2 the probability of a deletion with no insertions. The value of z along the transmitted branch using the correct method of matching undergoes a 3-way random walk, as shown in figure (4).

Here the decoding procedure has been defined such that if both an insertion and a deletion have taken place, the insertion will be matched first.

Now consider the probability that z is bigger that b, it can be shown [6], that

$$P(b < z) \leq \begin{cases} 1, & z \geq 0 \\ 2^{-h_1 z}, & z < 0 \end{cases} \tag{12}$$

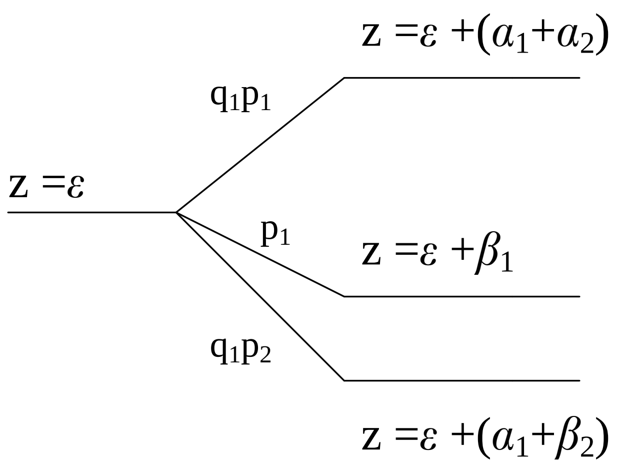

Figure 4: Illustrative diagram showing the three possible matching operations and there probabilities

Considering this probability at each of the nodes in figure(4) gives the continuity equation

$$1 = q_1 q_2 2^{h(\alpha_1+\alpha_2)} + q_1 p_2 2^{h(\alpha_1+\beta_2)} + p_1 2^{h\beta_1} \tag{13}$$

A comparison of this to equation (9), shows that the two equations are related by the condition $\lambda = 1 + h$. By then choosing the α and β values such that the two equations become identical, it is insured that both conditions for successful decoding can be met. These values are set as

$$\begin{aligned} \alpha_1 &= \log_2 2q_1 \\ \beta_1 &= \log_2 2p_1 \\ \alpha_2 &= \log_2 q_2 - R \\ \beta_2 &= \log_2 p_2 - R \end{aligned} \tag{14}$$

For the decoding process to succeed, it is necessary that the random walk behavior of z be such that the expectation value of its increase at each step be positive. Hence it is required that

$$q_1 q_2(\alpha_1 + \alpha_2) + q_1 p_2(\alpha_1 + \beta_2) + p_1 \beta_1 > 0. \tag{15}$$

Substituting in the values for α and β and rearranging gives

$$R_0 = q_1^{-1}(q_1 q_2 \log_2 2q_1 q_2 + q_1 p_2 \log_2 2q_1 p_2 + p_1 \log_2 2p_1) \tag{16}$$

which is an estimate of a lower bound for the channel capacity. This bound can be improved by considering different methods of matching. The correct method, as used here, strictly follows the pattern of insertions and deletions. However, a real decoder need not do and can improve upon the limit. For instance a deletion followed by an insertion of the same symbol will be treated as a successful transmission. The best method of matching is defined as the one which maximises the logarithm of the probability function. We now attempt to alter the random walk process to represent the best method of matching.

A first step is to take into account the effect of an insertion and deletion of the same symbol. This occurs with probability $\frac{1}{2}q_1p_1p_2$ and to match the received string to the code tree the correct method uses two matching operations, the first increasing z by β_1, and the second by $(\alpha_1 + \beta_2)$. The best method uses only one matching operation and increases z by $(\alpha_1 + \alpha_2)$. It is therefore necessary to change the probabilities associated with the increments in the random walk of z, considering the first matching operation gives the continuity equation,

$$1 = (q_1q_2 + \tfrac{1}{2}q_1p_1p_2)2^{h(\alpha_1+\alpha_2)} + q_1p_2 2^{h(\alpha_1+\beta_2)} + (p_1 - \tfrac{1}{2}q_1p_1p_2)2^{h\beta_1} \qquad (17)$$

in the same way as we arrived at equation(13).

As before, comparing with equation (9) gives the α and β values which allow both decoding conditions to be met, and in this case they are

$$\begin{aligned}\alpha_1 &= \log_2 2q_1 - R \\ \alpha_2 &= \log_2(q_2 + \tfrac{1}{2}q_1p_1p_2) \\ \beta_2 &= \log_2 2(p_2) - R \\ \beta_1 &= log_2 2(p_1 - \tfrac{1}{2}q_1p_1p_2).\end{aligned} \qquad (18)$$

We now apply the condition that the expected increase of z be positive. To do this equation (15) is altered to represent the decoder which treats the insertion and deletion as a correct transmission (increases z by $(\alpha_1+\alpha_2)$) as opposed to the correct decoder which will increase z by $(\alpha_1 + \beta_2) + \beta_1$.

$$q_1q_2(\alpha_1 + \alpha_2) + q_1p_2(\alpha_1 + \beta_2) + p_1\beta_1 + \tfrac{1}{2}q_1p_1p_2[(\alpha_1 + \alpha_2) - (\alpha_1 + \beta_2) - \beta_1] > 0$$

collecting like terms,

$$q_1(q_2 + \tfrac{1}{2}p_1p_2)(\alpha_1 + \alpha_2) + q_1p_2(1 - \tfrac{1}{2}p_1)(\alpha_1 + \beta_2) + p_1(1 - \tfrac{1}{2}q_1p_2)\beta_1 > 0 \qquad (19)$$

.

Substituting in for α and β from equation (18) then gives a revised estimate of a lower bound for the channel capacity of

$$R' = \frac{1}{q_1}[(q_1(q_2 + \tfrac{1}{2}p_1p_2)\log_2 2q_1(q_2 + \tfrac{1}{2}p_1p_2) + q_1p_2(1 - \tfrac{1}{2}p_1)\log 2q_1p_2$$
$$+p_1(1 - \tfrac{1}{2}q_1p_2)\log_2 2p_1(1 - \tfrac{1}{2}q_1p_2)]. \qquad (20)$$

It is expected that this limit can be improved further by adjusting for the probability of two deletions followed by two identical insertions and so on. However it was found that adjusting for these in the same way gave only small increases to the limit as the probability of the occurrences scales as $(p_2p_1)^n$ where n is the number of

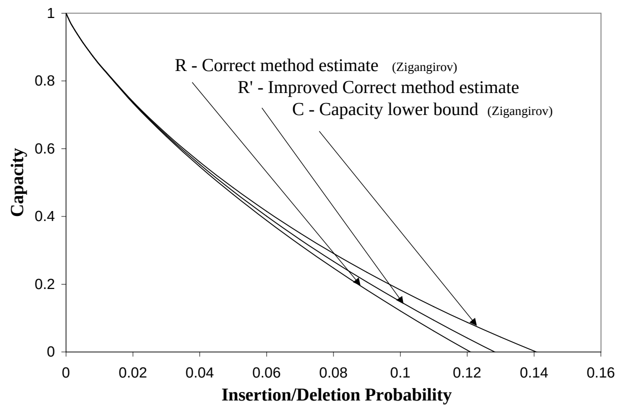

Figure 5: Comparison of the three different estimates for the channel capacity.

insertions and did not reach the bound for capacity obtained by Zigangirov, [2] given by

$$C \geq [q_1(1+p_1p_2)]^{-1}[1 + q_1^2p_2 \log_2 q_1^2p_2 + p_1(1-q_1p_2)\log_2 p_1(1-q_1p_2) \\ +q_1(1-q_1p_2+p_1p_2)\log_2 q_1(1-q_1p_2+p_1p_2)]. \qquad (21)$$

These three estimates for the channel capacity are compared in figure (5) for the case when $p_1 = p_2$.

All three of these estimates for lower bounds to the capacity of the channel, fall short of the rates that the watermark codes can achieve [3]. We now investigate the procedure used to achieve these limits in more depth.

3.3 Random walk behaviour

Consider the possible outputs from the channel that can occur with the transmission of a single digit. These outputs, along with their probabilities and the resulting increase in logarithm of the probability function using the correct method of matching are shown in table 2.

No. Insertions	No.Deletions	Prob	Increase in z
0	0	q_1q_2	$(\alpha_1+\alpha_2)$
0	1	q_1p_2	$(\alpha_1+\beta_2)$
1	0	$q_1q_2p_1$	$(\alpha_1+\alpha_2)+\beta_1$
1	1	$q_1p_2p_1$	$(\alpha_1+\beta_2)+\beta_1$
2	0	$q_1q_2p_1^2$	$(\alpha_1+\alpha_2)+2\beta_1$
2	1	$q_1p_2p_1^2$	$(\alpha_1+\beta_2)+2\beta_1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table 2: Emission probabilities and increase in logarithm of the probability function for the correct method of matching.

At each step in the random walk z is increased in one of three ways with probabilities q_1q_2, p_1q_2 and p_1. These probabilities are the same for each step. To see that this is correct, assume that during the transmission of our single digit, we had at least one insertion. The correct decoder will first follow the branch that increases z by β_1. Now consider what the next step of the decoder will be. With probability q_1q_2 it will place the next two symbols into correspondence. The probability of one insertion and one deletion is $q_1p_1p_2$, so once we have matched the insertion, the chance that the next matching event will correspond to a deletion is q_1p_2. The remaining probability is then $1 - q_1p_2 - q_1p_1p_2 = p_1$, the probability of a further insertion.

However when attempting to model the behaviour of a real decoder the probabilities for the increase in each step of the random walk must be changed. As well as leading to different α and β values, the probabilities of following each branch are different for different steps. As an example suppose we are using a decoder which increases z by $(\alpha_1+\alpha_2)$ for a deletion followed by an identical insertion, but otherwise acts identically to the correct decoder. Now consider the matching process for a single symbol. The first step in the walk will go to $(\alpha_1+\alpha_2)$ with probability $q_1q_2+\frac{1}{2}q_1p_1p_2$. Now again suppose in transmission there was at least one insertion and the decoder follows the branch which increases z by β_1. The second step in the walk will now increase z by $(\alpha_1+\alpha_2)$ with probability q_1q_2. So making adjustments in this way for alternative decoders results in the different steps in the random walk having different probabilities, making analysis complicated.

3.4 Different matching criteria

An alternative approach is to reconsider the matching process, and redefine a matching operation to be a matching of a symbol in the code tree to a number of symbols of the received string. We can then find a capacity estimate using this method. It is expected that for the correct method of matching these two procedures will give identical estimates. A calculation as before shows that both conditions for correct decoding can be met by comparing the following two equations.

$$1 = 2^{R-1}[2^{\lambda(\alpha_1+\alpha_2)}(1+2^{\lambda\beta_1}+2^{2\lambda\beta_1}+...)+2^{\lambda(\alpha_1+\beta_2)}(1+2^{\lambda\beta_1}+2^{2\lambda\beta_1}+...)] \quad (22)$$

which comes from the calculation on the incorrect subset and

$$1 = q_1q_2 2^{h(\alpha_1+\alpha_2)}(1+p_1 2^{h\beta_1}+p_1^2 2^{2h\beta_1}+...)+q_1p_2 2^{h(\alpha_1+\beta_2)}(1+p_1 2^{h\beta_1}+...) \quad (23)$$

which is the result of the calculation on the correct subset. From here we can calculate the α and β values and combine with the random walk criteria

$$q_2(\alpha_1+\alpha_2)+p_2(\alpha_1+\beta_2)+p_1q_1^{-1}\beta_1 > 0 \quad (24)$$

to give an estimation of

$$\begin{aligned} R &= q_2\log_2 2q_1q_2 + p_2\log_2 2q_1p_2 + p_1q_1^{-1}\log_2 p_1 \\ &= q_1^{-1}(q_1q_2\log_2 2q_1q_2 + q_1p_2\log_2 2q_1p_2 + p_1\log_2 p_1). \end{aligned} \quad (25)$$

This is an identical expression to that of equation (16), which shows that for the correct method of matching, these two methods are equivalent.

4 Capacity of the Gallager channel model

We now look to find a lower bound for the channel model considered by Gallager using a similar method. Here there are four possible outputs of the channel for transmission of a single digit. We define the matching operations as follows. Putting a symbol of the received string into 1-1 correspondence with a symbol in the code tree increases z by α. Deleting a symbol from the received sequence (ie identifying it as a symbol inserted during transmission) increases z by β. Matching involving the deletion of a symbol form the code tree increases z by γ. As well as these three operations, this model of the channel allows substitutions. The matching operation which puts two symbols into correspondence when a substitution has occurred is set to increase z by δ. It is assumed that

$$\alpha > 0, \quad \beta < 0, \quad \gamma < 0, \quad \delta < 0. \tag{26}$$

Now consider the same two conditions which must be met for correct decoding. By considering the set of tree paths and methods of matching in the incorrect subset which could give rise to a decoding error, we arrive at the continuity equations

$$\begin{aligned} L(\epsilon, i) = \tfrac{1}{2}L(\epsilon+\alpha, i+1) + \tfrac{1}{2}L(\epsilon+\beta, i+1) + \tfrac{1}{2}L(\epsilon+\gamma, i+1) \\ +\tfrac{1}{2}L(\epsilon+\delta, i+1), \end{aligned} \tag{27}$$

$$L(\epsilon, m) = L(\epsilon+\alpha, 1) + L(\epsilon+\beta, 1) + L(\epsilon+\gamma, 1) + L(\epsilon+\delta, 1). \tag{28}$$

Note that a matching event corresponding to an insertion during transmission now increases i to $i+1$. This is because each insertion relates to a transmitted symbol so identifying an insertion identifies a symbol in the code tree. In the model considered above, the insertions come about in the gaps between transmission so this is not the case.

We again look for solutions of the form $L(\epsilon, i) \leq A_i 2^{\lambda\epsilon}$ which gives the set of equations

$$\begin{aligned} A_1 &= \tfrac{1}{2}(2^{\lambda\alpha} + 2^{\lambda\beta} + 2^{\lambda\gamma} + 2^{\lambda\delta})A_2 \\ A_2 &= \tfrac{1}{2}(2^{\lambda\alpha} + 2^{\lambda\beta} + 2^{\lambda\gamma} + 2^{\lambda\delta})A_3 \\ &\vdots \\ A_{m-1} &= \tfrac{1}{2}(2^{\lambda\alpha} + 2^{\lambda\beta} + 2^{\lambda\gamma} + 2^{\lambda\delta})A_m \\ A_m &= (2^{\lambda\alpha} + 2^{\lambda\beta} + 2^{\lambda\gamma} + 2^{\lambda\delta})A_1 \end{aligned} \tag{29}$$

Solving this system of equations gives the solution

$$L(\epsilon, i) \leq A_i 2^{\lambda_0\epsilon} + B_i 2^{\lambda_1\epsilon} \tag{30}$$

where λ is the root of the equation

$$1 = 2^{R-1}(2^{\lambda\alpha} + 2^{\lambda\beta} + 2^{\lambda\gamma} + 2^{\lambda\delta}) \tag{31}$$

and the constants A_i and B_i can be found from the boundary conditions

$$\begin{aligned} L(\epsilon, i) &= 0 \quad \epsilon \le b & (32) \\ L(\epsilon, i) &= 1 \quad \epsilon \ge a & (33) \end{aligned}$$

As before it is possible to make the probability that a path in the incorrect subset is accepted go to zero with a suitable choice for the value of a, provided that equation (31) has real, positive roots.

The condition that the correct path be accepted gives rise to the equation

$$1 = p_t 2^{h\alpha} + p_d 2^{h\beta} + p_i 2^{h\gamma} + p_s 2^{h\delta} \qquad (34)$$

Now set $\lambda = h + 1$ in order to ensure both conditions are met. Then we have

$$\begin{aligned} \alpha &= \log_2 2p_i - R \\ \beta &= \log_2 2p_d - R \\ \gamma &= \log_2 2p_i - R \\ \delta &= \log_2 2p_s - R \qquad (35) \end{aligned}$$

Again we consider the random walk process of z if the correct method of matching is followed. It is required that the expectation value of the increase of z with each matching event is positive, therefore

$$p_t\alpha + p_d\beta + p_i\gamma + p_s\delta > 0. \qquad (36)$$

Then substituting in for the α, β, γ and δ values from equations (29) and rearranging the following bound on the channel capacity is obtained.

$$C \ge 1 + p_t \log p_t + p_i \log p_i + p_d \log p_d + p_s \log p_s \qquad (37)$$

It is interesting that this estimate is identical to one which can be obtained by considering assuming a discrete memoryless channel model.

Again it is expected that this limit can be improved upon by altering the method of matching used along the correct path. For example an insertion followed by a deletion will be decoded as error free transmission when the inserted symbols are identical to those deleted. The probability of this happening in the transmission of two symbols is $\frac{1}{2}p_i p_d$, which is small compared to p_t, so the improvement offered by using the best method of matching will be small.

Setting $p_s = 0$ enables a comparison of this capacity estimate to that for the Zigangirov channel. With $p_i = p_d = p_1 = p_2$, the probability of correct transmission of a symbol for the Gallagher channel is $p_t = 1 - 2p_i$, whilst for the Zigangirov channel, $p_t = (1 - p_i)^2 = 1 - 2p_i + p_i^2$, so it is expected that the capacity of the zigangirov channel is the larger of the two as is observed in figure(6).

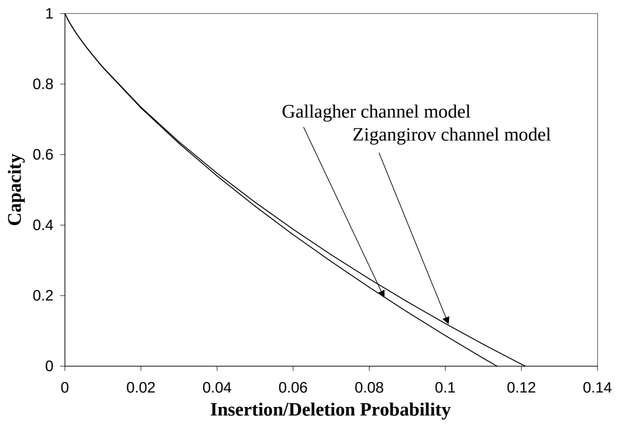

Figure 6: Comparison of the lower bounds for capacity of the two channels obtained using the correct method of matching.

5 Discussion

It has been shown that an analysis of insertion and deletion channels using convolutional coding to transmit the data and sequential decoding to recover the message can lead to estimates for the channel capacity. These estimates are expected to be lower than the actual capacity of the channel as they correspond to the use of a correct decoder which is not optimal for the reasons discussed. It is difficult to make similar calculations for more realistic decoders because of the effect on the random walk process of the logarithm of the probability function. The probabilities at each step are dependent on the direction taken at the previous node, suggesting that some form of conditional probability distribution should be built up.

It is possible to use the methods used here to produce estimates for the capacity of different channels, including ones which allow substitution errors, the Gallager channel being one example.

The improvements that were made to the correct decoder were unable to reach the bound on capacity obtained by Zigangirov [2]. It is known that this bound aims to take account of the phase of the received sequence to the code tree but beyond this, it is unclear how the estimate was made.

For real insertion and deletion channels, synchronisation problems can be worse when several consecutive 1s or 0s are transmitted, whereas in the limits calculated here the insertion and deletion probabilities are not dependent on the previous transmitted symbols.

Watermark codes, which allow communication at rates above these theoretical estimates, use information about the phases of the two sequences which suggests that a sequential decoding procedure that was able to do this could lead to a higher lower bound estimate than calculated here.

References

[1] Gallager, R. G., “Sequential Decoding for Binary Channels with Noise and Synchronization Errors”, Unpublished Lincoln Lab report 25 G-2, 1961.

[2] K. Sh. Zigangirov, “Sequential decoding for a binary channel with drop outs and insertions”, *Problemy Peredachi Informatsii*, vol. 5, no.2, pp.23-30, 1969.

[3]M. C. Davey and D. J. C. Mackay, “Reliable communication over channels with insertions, deletions and substitutions”, Submitted to *IEEE Transactions on Information Theory*, December 1999.

[4] J. M. Wozencraft and I. M. Jacobs, “Principles of Communication Engineering”, Wiley, New York, 1965.

[5] Edward A. Ratzer and David J.C. MacKay, “Codes for Channels with Insertions, Deletions and Substitutions”, published in the *proceedings of the 2nd International Symposium on Turbo Codes and Related Topics*, Brest, France, 4-7 September 2000.

[6] K. Sh. Zigangirov, “Some sequential decoding procedures”, *Problemy Peredachi Informatsii*, vol. 2, no.4, pp.1-10, 1966.