

7. J. J. Linger, "a Theorem in Finite Projective Geometry and some applications to Number Theory," Trans. American Math Soc., vol 43, pp 377-385, May 1938.

C. Sequential Decoding

Ref: Gallager (3), Wozencraft + Jacobs (10)

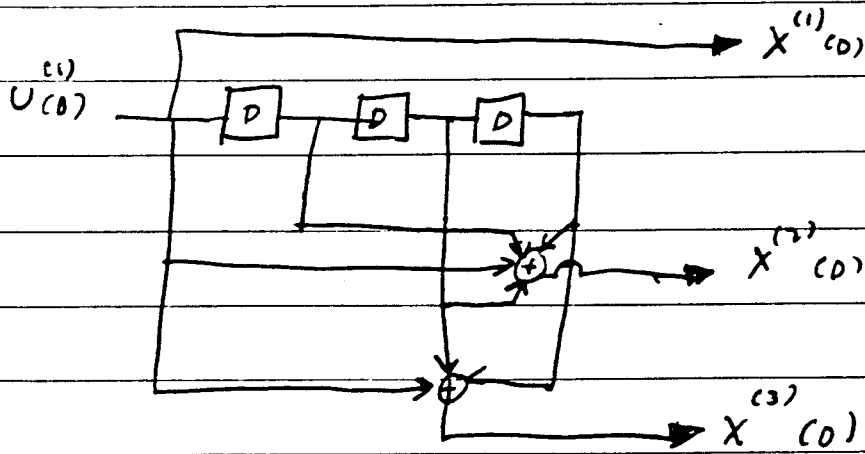
For equally likely messages the minimum probability of error decoder is a maximum likelihood decoder. This involves calculating $P(\bar{Y} / \bar{X})$ for all q^K code words. In practice this is computationally prohibitive for even moderate values of K . Sequential decoding is a method that approximates maximum likelihood decoding but requires far less computation. It differs from previously discussed decoding techniques in that the amount of computation to decode a specific bit is variable.

1. Tree structure of Convolutional codes

as before, the contents of all storage elements will be assumed to be 0 initially and all information, code, and received sequences will be taken as 0 for negative times.

The code words of a convolutional code can be displayed on a code word tree. As an example

consider the following binary encoder $R = 1/3, N_0 = 3$
 $m = 3, N_A = 12$

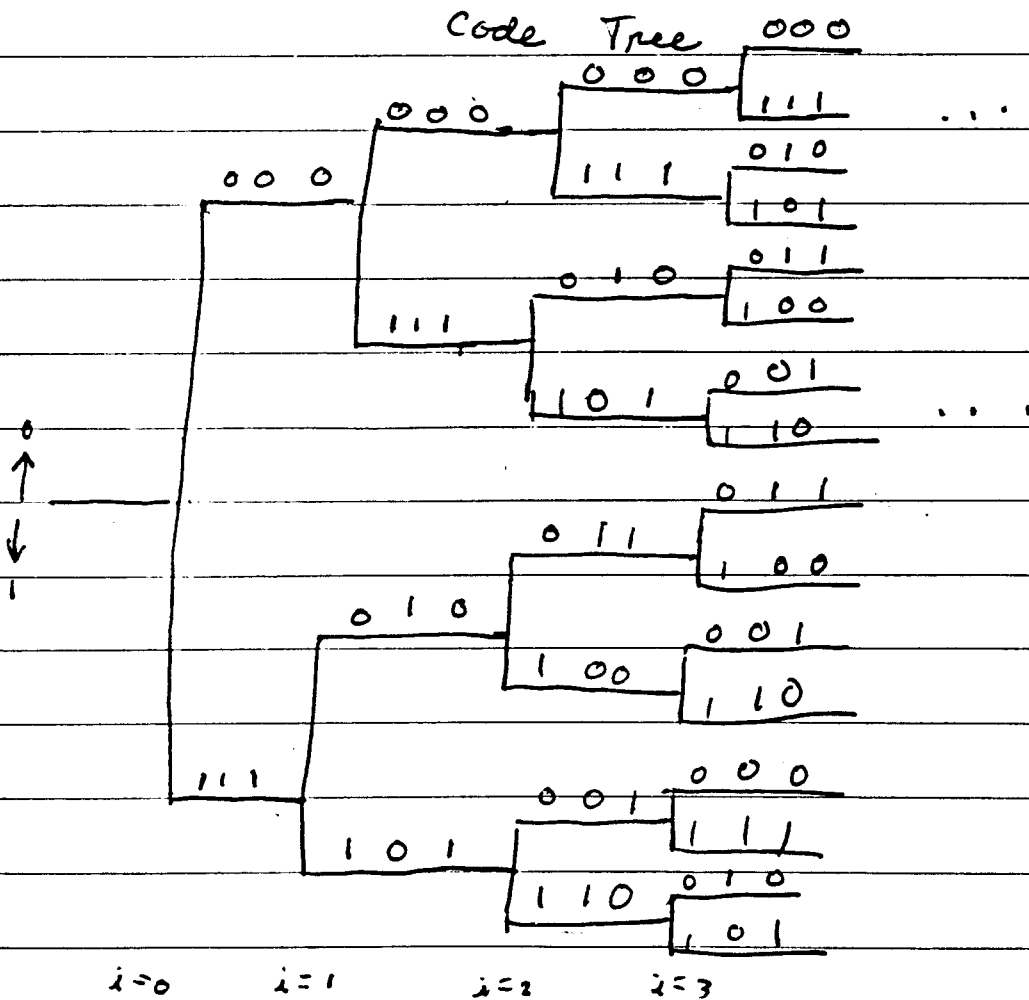


$$x_{\lambda}^{(1)} = u_{\lambda}^{(1)}$$

$$x_{\lambda}^{(2)} = u_{\lambda}^{(1)} + u_{\lambda-1}^{(1)} + u_{\lambda-2}^{(1)} + u_{\lambda-3}^{(1)}$$

$$x_{\lambda}^{(3)} = u_{\lambda}^{(1)} + u_{\lambda-2}^{(1)} + u_{\lambda-3}^{(1)}$$

Code Tree



The information sequence directs the encoder on a path through the tree. Each node represents a point in time. If the information bit is 0 the encoder takes the upper branch. If it is 1 the encoder takes the lower branch.

If the channel is a BSC, the maximum likelihood decoder would compute the Hamming distance to each of the 2^k code words and decide on the one with the smallest distance. A sequential decoder decides on each information bit u_i when it is received. If it appears that a decoding mistake has been made the decoder can go back and change some of its hypotheses. As an example suppose the first constraint length of received data is 101 101 001 000. The first block 101 is closest to 111 so the decoder hypothesizes that $u_0 = 1$. The second block is identical to $\bar{x}_1 = 101$ so hypothesize $u_1 = 1$ and similarly $u_2 = 0$, $u_3 = 0$. The agreement between $\bar{x}$ and $\bar{y}$ beyond $\bar{x}_0$ indicates that u_0 was hypothesized correctly. On the other hand assume that $\bar{x} = 111 101 001 000$ is again transmitted and 010 101 001 000 is received. Then the decoder would hypothesize $u_0 = 0$ with $d(\bar{x}_0, \bar{y}_0) = 1$, $u_1 = 1$ with $d(\bar{x}_1, \bar{y}_1) = 1$, $u_2 = 1$ with $d(\bar{x}_2, \bar{y}_2) = 1$, and $u_3 = 0$ with $d(\bar{x}_3, \bar{y}_3) = 1$. Once the decoder makes an error, then it is forced to make subsequent

errors. By monitoring the number of errors between the received and hypothesized sequences the decoder will most likely recognize that it is on the wrong path. It can then back up, change a hypothesis and look for a better path. For a BSC with crossover probability ϵ one would expect $\epsilon(i+1)N_0$ channel errors in the received data $\bar{y}_0, \dots, \bar{y}_i$. Let the received data up to time i be $\bar{Y}_{1:i} = [\bar{y}_0, \dots, \bar{y}_i]$. If the decoder is on the right path one would expect $w(\hat{X}_{1:i}, \bar{Y}_{1:i}) \cong \epsilon(i+1)N_0$ where $\hat{X}$ is the hypothesized transmitted code sequence. On the wrong path $w(\hat{X}_{1:i}, \bar{Y}_{1:i}) \cong \frac{1}{2}(i+1)N_0$. The decoder can compare the actual Hamming distances with the expected distances to determine whether it is on the right path or not.

In general the encoder accepts blocks of K_0 information symbols and generates a tree with q^{K_0} branches at each node.

1. Fano algorithm

The first sequential decoding algorithm was devised by Wozencraft (1957). An improved algorithm due to Fano (1963) and modified by Gallager will be discussed here. The Hamming distance is matched to the BSC and special symmetric channel discussed previously. A more general measure of the difference between the received

and hypothesized transmitted sequences is

$$\Gamma(\bar{X}_{(L)}, \bar{Y}_{(L)}) = \sum_{n=0}^L \sum_{i=1}^{N_0} \left\{ \ln \left[\frac{P(y_n^{(i)} / x_n^{(i)})}{P(y_n^{(i)})} \right] - B \right\}$$

where B is an arbitrary bias term to be chosen later. Notice that $I(y_n^{(i)}; x_n^{(i)}) = \ln \left[\frac{P(y_n^{(i)} / x_n^{(i)})}{P(y_n^{(i)})} \right]$

is the mutual information between $y_n^{(i)}$ and $x_n^{(i)}$. For reasonable channels $P(y_n^{(i)} = x_n^{(i)} / x_n^{(i)})$ is the largest transition probability. Therefore Γ is large when $\hat{X}_{(L)}$ and $\bar{Y}_{(L)}$ agree and small if they disagree significantly. The decoder searches the tree for the path of best agreement. Γ is called the received value.

EXAMPLE: BSC, crossover ϵ , equally likely inputs

$$\begin{aligned} P(y_n^{(i)} = 1) &= P(x_n^{(i)} = 1) P(y_n^{(i)} = 1 / x_n^{(i)} = 1) \\ &\quad + P(x_n^{(i)} = 0) P(y_n^{(i)} = 1 / x_n^{(i)} = 0) \\ &= \frac{1}{2} [1 - \epsilon + \epsilon] = \frac{1}{2} \end{aligned}$$

$$\begin{aligned} \Gamma(\bar{X}_{(L)}, \bar{Y}_{(L)}) &= d(\bar{X}_{(L)}, \bar{Y}_{(L)}) \ln 2\epsilon + [(L+1)N_0 - d(\bar{X}_{(L)}, \bar{Y}_{(L)})] \\ &\quad \ln 2(1-\epsilon) - (L+1)N_0 B \\ &= d(\bar{X}_{(L)}, \bar{Y}_{(L)}) \ln \frac{\epsilon}{1-\epsilon} + (L+1)N_0 [\ln 2(1-\epsilon) - B] \end{aligned}$$

The decoder contains a replica of the encoder that it uses to generate the code tree. Thus the encoder complexity tends to increase linearly with m . The decoder makes a hypothesis on a

received block, shifts the hypothesized information into its replica encoder to calculate the corresponding code block hypothesis and then calculates Γ associated with the tree node specified by the hypothesized information sequence. Since the decoder will be allowed to look up, more than m past hypotheses must be stored.

Search algorithm

Rule	Conditions on Node		Action to Take	
	Previous Move	Γ_{l-1} vs Γ_l	Final Threshold	Move
1	F or L	$\Gamma_{l-1} < T + \Delta, \Gamma_l \geq T$	Raise	F
2	F or L	$\Gamma_{l-1} \geq T + \Delta, \Gamma_l \geq T$	No change	F
3	F or L	any $\Gamma_{l-1}, \Gamma_l < T$	No change	L or B
4	B	$\Gamma_{l-1} < T, \text{any } \Gamma_l$	Lower by Δ	F
5	B	$\Gamma_{l-1} \geq T, \text{any } \Gamma_l$	No change	L or B

$\Gamma_{-1} = 0, \Gamma_{-2} = -\infty$, for $l = -1$ use rule 1, start at base of tree
The decoder can make three types of moves.

1. Forward move (F)

The decoder moves one node ahead in time and shifts the corresponding k_0 hypothesized information digits into its encoder.

$$\Gamma(\hat{\bar{X}}_{l+1}, Y_{l+1}) = \Gamma(\hat{\bar{X}}_l, \bar{Y}_l) + \sum_{i=1}^{N_0} \left\{ \ln \left[\frac{P(y_{l+1}^{(i)} / \hat{x}_{l+1}^{(i)})}{P(y_{l+1}^{(i)})} \right] \right\}$$

$$\text{or } \Gamma_{l+1} = \Gamma_l + \sum_{i=1}^{N_0} \left\{ \ln \left[\frac{P(y_{l+1}^{(i)} / \hat{x}_{l+1}^{(i)})}{P(y_{l+1}^{(i)})} \right] - B \right\}$$

thus the decoder only needs to know Γ_l and $\hat{x}_{l+1}$ to calculate Γ_{l+1} . It will be assumed that the branches stemming from each node have a predetermined order, an 1 move means move forward to the 1st node stemming from present node.

2. Lateral Move (L)

at time l the decoder shifts to another branch attached to the same node. Once again this affects only the last term in the sum for Γ_l . an L move means move laterally to the next higher ordered branch. If already at last branch then move backwards

3. Backward move (B)

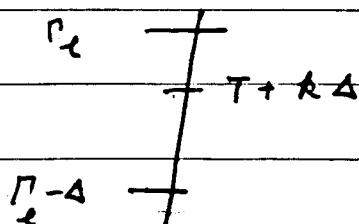
The decoder shifts one branch backward in time. The replica encoder register is shifted to the left with past hypotheses reinserted at the right. Γ is changed by subtracting of the quantity corresponding to the hypothesis shifted out of the register.

For each type of move the change in Γ is only determined by the last K hypothesized information digits.

Threshold (T)

The threshold T is constrained to change in increments of Δ . A "raise" in T means

add $k\Delta$ to T so that $\Gamma_l - \Delta < T + k\Delta \leq \Gamma_l$



In the initial Fano algorithm the branches at each node were ordered in decreasing value of Γ . The present algorithm eliminates the ordering computations but examines more nodes.

We will assume that the nodes are ordered in the natural order of $\vec{u}_e$.

Example:

$R = 1/3$ code of p. 86 with BSC

$$\Gamma_e = d(\hat{X}_{(e)}, \bar{Y}_{(e)})\beta + (l+1)\alpha$$

where $\alpha = N_0 [\ln(2(1-\epsilon)) - B]$

assume $0 < \alpha < \Delta$ and $\alpha \geq \Delta$

and $\beta = \ln \frac{\epsilon}{1-\epsilon} < 0$

assume $\Delta < |\beta| < 2\Delta$

assume $\vec{u} = 1 \quad 1 \quad 0 \quad 0$

$\vec{X} = \vec{Y} = 111 \quad 101 \quad 001 \quad 000$

l	$\hat{u}_{(e)}$	T_{signal}	Γ	Rule	Move
-2	-	?	$-\infty$		
-1	-	0	0	1	F
0	0	0	$3\beta + \alpha$	3	L
0	1	0	α	1	F
1	10	0	$3\beta + 2\alpha$	3	L
1	11	Δ	2α	1	F
2	110	2Δ	3α	1	F
3	1100	3Δ	4α	1	F

Rule 1 is used when the channel noise is small and the decoder makes the correct decisions. With no noise rules 1 and 3 only are used, i.e. only F and L moves.

Suppose there are some channel errors and

$$\bar{y} = 010 \ 101 \ 001 \ 000$$

$$\bar{x} = 111 \ 101 \ 001 \ 000$$

$$\bar{u} = 1 \quad 1 \quad 0 \quad 0$$

(see p. 270 Gallager)

l	$\bar{u}_{(2)}$	T_{final}	Γ	Rule	Move
-2	-	?	$-\infty$		
-1	-	0	0	1	F
0	0	0	$\beta + \alpha$	3	L
0	1	0	$2\beta + \alpha$	3	B
-1	-	$-\Delta$	0	4	F
0	0	$-\Delta$	$\beta + \alpha$	2	F
1	00	$-\Delta$	$3\beta + 2\alpha$	3	L
1	01	$-\Delta$	$2\beta + 2\alpha$	5	B
0	0	$-\Delta$	$\beta + \alpha$	5	L
0	1	$-\Delta$			B
-1	-	-2Δ			F
0	0	-2Δ			F
1	00	-2Δ			L
1	01	-2Δ			F
2	010	-2Δ			L
2	011	-2Δ			F

3	0110	-2Δ	L
3	0111	-2Δ	B
2	011	-2Δ	B
1	01	-2Δ	B
0	0	-2Δ	L
0	1	-2Δ	F
1	10	-2Δ	L
1	11	-2Δ	F
2	110	-Δ	F
3	1100	0	F

⋮

Roughly the decoder attempts to find a path through the tree such that Γ_k remains above the current threshold. If no such path exists it is forced back to the node where this T was set and lowers T by Δ . It then tries to move forward again.

Terminology

1. an F hypothesis is a hypothesis to which rules 1, 2, 4 apply
2. Then antecedents of a node $\tilde{u}_{k+1}$ are the nodes connecting $\tilde{u}_{k+1}$ with the tree origin
3. The path values $\Gamma_0, \dots, \Gamma_k$ associated with node $\tilde{u}_{k+1}$ are the values of $\tilde{u}_{k+1}$ and its antecedents.

4. The path thresholds $T_0, \dots, T_l$ for hypothesis $u_{[l]}$ are the final thresholds on most recent hypotheses of the antecedents of $u_{[l]}$.
5. The descendants of $u_{[l]}$ are the nodes branching to the right of $u_{[l]}$.
6. The immediate descendants of $u_{[l]}$ are the q^{k_0} descendants one branch away.
7. The path from a ~~the~~ node $u_{[i]}$ to a descendant $u_{[l]}$ is above the threshold T if $\Gamma_j \geq T$ for $i \leq j \leq l$.

The following theorem gives some of the properties of the search algorithm.

Theorem 6.9.1 (Gallager - Info. Th.) p. 271

(a) The path thresholds $T_0, \dots, T_l$ and path values $\Gamma_0, \dots, \Gamma_l$ associated with hypothesis of node $u_{[l]}$ satisfy for $0 \leq i \leq l-1$

$$T_i \leq \Gamma_i$$

$$T_{i+1} \geq T_i$$

$$T_{i+1} \geq T_i + \Delta \Rightarrow T_i + \Delta > \Gamma_i$$

$$T_{i+1} \geq T_i + \Delta \Rightarrow T_{i+1} > \Gamma_i$$

(b) For each node that is ever $\mathbb{F}$ hypothesized,

$$T \leq \Gamma < T + \Delta$$

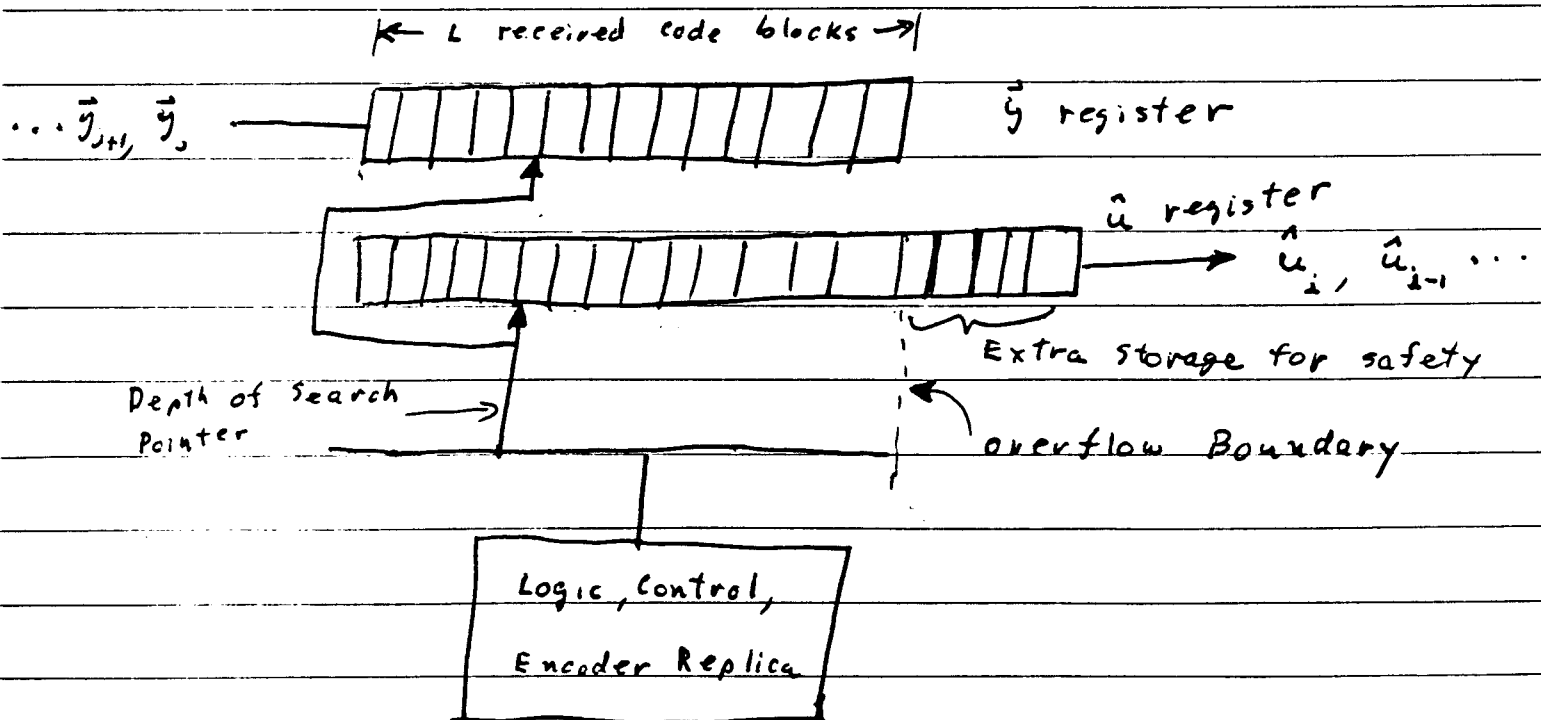
for T the final threshold on the first hypothesis of the node. On each subsequent hypothesis

of the node, the final threshold is Δ below what it was on the previous F hypothesis.

(c) Let $\tilde{u}$ be F hypothesized with final threshold T . Then, before $\tilde{u}$ can be re-hypothesized, every descendent of $\tilde{u}$ for which the path of $\tilde{u}$ is above T must be F hypothesized with final threshold T . Furthermore, between the hypothesis of $\tilde{u}$ and its first re-hypothesis, the threshold cannot be lowered below T .

3. System Performance

a. Decoder Mechanization



In practice, a sequential decoder would have the form shown above. L received code blocks are stored in a buffer called the

$\hat{y}$ register here. The hypothesized information blocks are stored in the $\hat{u}$ register.

The decoder must make one or more hypothesis in decoding each block. Therefore, for real time decoding, the decoder must be able to make hypotheses several times faster than the rate at which blocks arrive. For low noise conditions the data will be decoded with little effort. The depth of search pointer indicates the information block that is presently being hypothesized. For low noise the pointer will hover near the left end of the $\hat{u}$ register. During noisy periods more computation will be required and the pointer will drift towards the right end of the register. When the pointer reaches the right end of the $\hat{y}$ register a buffer overflow occurs. When this happens it is likely that several neighboring blocks have been hypothesized incorrectly. The extra storage on the $\hat{u}$ register provides a safety margin so that incorrectly decoded data is not released before a buffer overflow is announced.

b. Decoding Computation and Buffer Overflow

To analyze waiting line behavior of the decoder it is necessary to know the statistical properties of the decoding computation

Let w_n be the number of F hypotheses made on the n^{th} node in the correct path and on all nodes in the tree of incorrect paths stemming from this n^{th} node. $\sum w_n$ is the number of F hypotheses required to decode the sequence and w_n can be considered to be the number of F hypotheses needed to decode the n^{th} block.

At any length l , the total number of LTB moves is $\leq q^{K_0} \times w_{l-1}$ since each node has q^{K_0} immediate descendants and only one LTB move can occur per F hypothesis of a given node.

Thus at any time the total number of hypotheses is $\leq (q^{K_0} + 1) \times (\text{number } F \text{ hypotheses})$. Therefore bounds on w_l also give bounds on the total number of hypotheses.

The performance of individual encoders is nearly impossible to analyze exactly. However, the ensemble of encoders with randomly selected $G_{i,j}^{(j)}$ coefficients can be analyzed by arguments similar to those used in proving the noisy channel coding theorem. Gallager (ch 6) analyzes the following system: The binary input data is grouped in blocks of K_0 digits. For each K_0 input digits the convolutional encoder has a N_0 binary output digits. Each block of " a " output digits is mapped to one

of 2^a channel symbols. Thus there are N_0 channel symbols generated for each K_0 binary encoder inputs and $R = K_0 \ln 2 / N_0$ nats/channel symbol

Thm 6.9.1 (Gallager)

For the system described above used over a discrete memoryless channel

$$E\{W_n\} \leq 4 \{1 - \exp[-N_0 (E_0(1, Q) - R)]\}^{-2}$$

for $R = K_0 \ln 2 / N_0 < E_0(1, Q) = R_{\text{comp}}$

where $B = E_0(1, Q)$, $\Delta = 2 \ln 2$

and

$$E_0(\rho, Q) = -\ln \sum_{j=0}^{J-1} \left[\sum_{k=0}^{K-1} Q(k) P(j/k) \right]^{\frac{1}{1+\rho}} \right)^{1+\rho}$$

is Gallager's zero rate exponent with

$P(j/k)$ = channel transition probabilities

$Q(k)$ are the channel input probabilities

Furthermore

$E\{W_n^\rho\}$ is finite for $1 \leq \rho$

and $R < E_0(\rho, Q) / \rho$

Simulations have shown that the computation has a Pareto distribution, i.e.,

$$P\{W \geq L\} \approx K L^{-\rho}$$

where $K \approx 1$ where ρ is the solution of

$$R = E_0(\rho, Q) / \rho$$

γ is called the Pareto exponent. The simulations also show that for $R > E_0(1, Q)$
 $E\{W\} \rightarrow \infty$.

In designing a decoder the $\bar{y}$ register length L must be long enough to make the probability of overflow, P_{ov} , small. Let the decoder be able to search γ branches in the time 1 branch is received, assuming that the depth of search pointer is initially at the left of the $\bar{y}$ register, the decoder can search γL branches before overflow. Therefore $P_{ov} \approx P\{W \geq \gamma L\} = K(\gamma L)^{-\gamma}$

or

$$L \approx \frac{1}{\gamma} \left[\frac{K}{P_{ov}} \right]^{1/\gamma}$$

EX: $R = E_0(1, Q) \Rightarrow \gamma = 1$

$\gamma = 10$, $K = 1$, $P_{ov} = 10^{-6}$

Then $L \approx \frac{1}{10} \times 10^6 = 10^5$ branches

From this example it is apparent that a significant amount of storage is needed for small buffer overflow probability!

c. Decoding Error Probability

Let $P_{e,n}$ = probability that the n^{th} block is decoded incorrectly. Then Gallager (p. 284) shows that

$$P_{e,n} \leq A \exp [-N_A E_0(1, Q)]$$

$$\text{where } A = \left[\frac{m}{(1-z)^2} + \frac{1+z}{(1-z)^3} \right] \exp [N_0 R + \frac{\Delta}{2}]$$

$$z = \exp \{ -N_0 [E_0(1, Q) - R] \}$$

$$\text{for } R < E_0(1, Q)$$

Thus for convolutional codes with sequential decoding, $P_{e,n}$ can be made to decrease exponentially with R fixed by increasing N_A . No practical block codes are known with this property. Justesen has constructed a class of block codes that are good for large N but a well developed collection of decoding algorithms does not exist. (Blaum, Theory and Practice of Error Control Codes, pp. 201-204)

For very noisy channels it can be shown (Gallager - p 148)

$$E_0(r) = \max_Q E_0(r, Q) \approx \frac{r}{1+r} C$$

where C is the channel capacity.

If data is transmitted over an analog channel, when the rate is increased the channel symbols must have shorter duration so that the combined channel consisting of the discrete input, channel encoder, analog channel, and detector will look noisier as R increases. From the results above, the code rate R must satisfy

$$R < E_0(1) \approx \frac{C}{2}$$

for successful decoding. Thus convolutional encoding with sequential decoding can only reach rates = half channel capacity. This is /

still far better than other known practical coding schemes!

In most practical cases $P_{e,n}$ can be made far smaller than necessary with reasonable values of m . The decoder performance is determined primarily by buffer overflow probability.

→ Viterbi Algorithm: com Tech Oct Sept. '91 Excellent analysis of conv. code.
Jelinek Stack Algorithm

IV. Algebraic Cyclic Block Codes [Ref: Gallager, Ch 6]

A. Review of Algebra

Def: Group

a group $\mathcal{G}$ is a set of elements $a, b, c, \dots$ and an operation, $\cdot$, for which

$$(1) a \in \mathcal{G}, b \in \mathcal{G} \Rightarrow a \cdot b \in \mathcal{G} \text{ (closure)}$$

$$(2) \text{ associative law } a, b, c \in \mathcal{G}$$

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

$$(3) \text{ Identity element } e \in \mathcal{G}$$

$$a \cdot e = e \cdot a = a \text{ for all } a \in \mathcal{G}$$

$$(4) \text{ Inverses: for each } a \in \mathcal{G}, \text{ have inverse } a^{-1}$$

$$a \cdot a^{-1} = a^{-1} \cdot a = e$$

Def: Abelian or Commutative Group

$$a \cdot b = b \cdot a$$

Def: Subgroup

a subset $\mathcal{H}$ of $\mathcal{G}$ is a subgroup of $\mathcal{G}$ if the elements of $\mathcal{H}$ satisfy axioms for $\mathcal{G}$.