

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
CAMPUS ESTADO DE MÉXICO**



**CODIFICACIÓN FUENTE - CANAL DE SEÑALES
DE VIDEO PARA LA TRANSMISIÓN TOLERANTE A
PÉRDIDAS EN REDES HETEROGÉNEAS**

TESIS QUE PARA OPTAR EL GRADO DE MAESTRO EN CIENCIAS
COMPUTACIONALES CON ESPECIALIDAD EN REDES
PRESENTA

ADRIANA VIVEROS DOMINGUEZ

Asesor: Dr. GUSTAVO A. SANTANA TORRELLAS

Comité de Tesis: Dr. EDUARDO GARCÍA GARCÍA
Dr. CARLOS RODRÍGUEZ LUCATERO
M. en C. MA. DE LOS ANGELES JUNCO REY

Jurado: Dr. EDUARDO GARCÍA GARCÍA
M. en C. MA. DE LOS ANGELES JUNCO REY
Dr. CARLOS RODRÍGUEZ LUCATERO
Dr. GUSTAVO A. SANTANA TORRELLAS

Presidente
Secretario
Vocal
Vocal

Atizapán de Zaragoza, Estado de México, Mayo de 1999

RESUMEN

El problema de la transmisión de datos multimedia sobre redes heterogéneas involucra varias consideraciones, entre las más relevantes se encuentran el diseño y la selección de esquemas de codificación, y las técnicas de transmisión. Recientemente, el interés en las redes móviles con capacidad para soportar tráfico multimedia han estimulado el campo de la investigación, habiendo logrado un progreso significativo en esta tecnología [1], [3], [17], [24], [10].

La información (video) se comprimir con la finalidad de adaptarse a los requerimientos del canal: velocidad, capacidad y ancho de banda. La codificación canal es un proceso en el cual después de la compresión se agrega redundancia con el fin de garantizar la confiabilidad en la transmisión de los datos.

La transmisión progresiva es una técnica muy útil que emplea elementos de codificación fuente-canal para transmitir video, y en general, datos de multimedia sobre canales con un ancho de banda limitado; sin embargo, presenta problemas sobre canales ruidosos debido a que utiliza codificación de longitud variable, es por ello se que requieren métodos de mayor calidad para evitar el ruido, situación propia de los canales inalámbricos; no obstante, en este trabajo no se considera en su totalidad. Un método es combinar la transmisión progresiva con el VQ 2-D-CELP de video sobre canales con desvanecimientos. El método 2-D-CELP extiende el concepto del CELP usado para la codificación de voz. La técnica VQ 2-D-CELP emplea un filtro de predicción de error inverso, un método iterativo de agrupación para diseñar los predictores a partir de un arreglo inicial y un algoritmo para determinar el tamaño del bloque.

Como resultado se obtiene un sistema de codificación fuente-canal mediante el cual se pretende explotar las características entrópicas de la señal y las características estadísticas de los canales de comunicación, mismo que puede ser empleado en una amplia gama de aplicaciones para transmisión de video sobre canales de ancho de banda reducido, canales inalámbricos y canales en sistemas de videotelefonía y videoconferencia sobre redes heterogéneas.

CONTENIDO

Lista de Figuras	8
Lista de Tablas	10
Abreviaturas	11
1. Motivación – Orígenes y Antecedentes	13
2. Fundamentos y Marco Teórico	16
2.1 Introducción al Video Digital	16
2.1.1 Transformadas	20
2.1.2 DCT	20
2.1.3 Cuantificación	21
2.1.4 Codificación intertrama	22
2.1.5 Predicción con compensación de movimiento	23
2.1.6 Interpolación con predicción de movimiento	24
2.1.7 JPEG	25
2.1.8 MPEG	26

2.2 Elementos de codificación Fuente-Canal	32
2.2.1 Teoría de codificación de canal	32
2.2.1.1 Códigos de bloque	36
2.2.1.2 Teorema de codificación inverso	37
2.2.1.2 Códigos Convolucionales	59
2.2.1.3 Decodificación Viterbi	63
2.3 Teoría de distorsión de velocidad	80
2.3.1 Códigos de Bloque	82
2.3.2 Códigos Trellis	91
2.3.3 Teorema de convergencia	99
2.3.4 Discusión	102
2.4 Elementos de la Transmisión en Redes	106
2.4.1 Conceptos Generales	106
3. Codificación fuente-canal para la transmisión tolerante a pérdidas en redes heterogéneas	109
3.1 Transmisión Progresiva	109
3.2 Sistema de codificación 2-D-CELP	111
3.3 Canales con desvanecimiento	114
3.4 2-D-CELP con tamaño de bloque variable	116

Conclusiones

119

Bibliografía

122

LISTA DE FIGURAS

Figura 1. Codificador Predictivo	18
Figura 2. Transmisor DPCM	19
Figura 3. Receptor DPCM	19
Figura 4. Patrón de zigzag	22
Figura 5. Predicción	23
Figura 6. Algoritmo básico de JPEG	26
Figura 7. Representación 4:2:0	27
Figura 8. Estructura en capas	28
Figura 9. Ejemplo GOP	31
Figura 10. BSC	33
Figura 11. Sistema de Codificación Canal Básico	34
Figura 12. Curva $E_0(\rho)$	40
Figura 13. Exponentes	59
Figura 14.a. Codificador	62
Figura 14.b. Diagrama de estados	62
Figura 14.c Diagrama Trellis	63
Figura 15. Caminos unidos	64
Figura 16. Ejemplo de decodificación de Viterbi	66
Figura 17. Evento de error	67
Figura 18. Comparación de exponentes aleatorios de codificación	71
Figura 19. Exponentes convolucionales de codificación	76
Figura 20. Ejemplo de función de transferencia	79
Figura 21. Modelo del sistema de comunicación	81
Figura 22. Modelo de codificación fuente	82
Figura 23. Canal de prueba hacia atrás	87
Figura 24. Codificador 2-D-CELP	112
Figura 25. Decodificador 2-D-CELP	112
Figura 26. Modelo de Gilbert-Elliott	114

Figura 27. Geometría de un predictor 2-D	115
Figura 28. Formas de bloque que aseguran el cálculo causal para $P_2 = 1, 2, 3$	115
Figura 29. Orientación de los predictores iniciales	116
Figura 30. Diagrama de flujo de la codificación VB 2-D-CELP	118

LISTA DE TABLAS

Tabla 1. Características del procesamiento de video	16
Tabla 2. Velocidad de datos sobre diversos medios de comunicación	16
Tabla 3. Matrices de ponderación	26
Tabla 4. Matriz de cuantificación	30

ABREVIATURAS

Abreviatura	Significado
2-D-CELP	2-D Code Excited Linear Prediction
ACELP	Adaptive Code Excited Linear Prediction
ARPA	Advanced Research Projects Agency
ARQ	Automatic Repeat reQuest
B	Bidireccional
BER	Bit-Error Rate
BSC	Binary Symmetric Channel
BSS	Binary Symmetric Source
CCIR	International Radio Consultative Committee
Cr	Señal de diferencia del color azul
Cb	Señal de diferencia del color rojo
DCT	Discrete Cosine Transform
DMC	Discrete Memoryless Channel
DPCM	Differential Pulse Code Modulation
ERT	Erasure Resilient Transmission
FEC	Forward Error Correction
Gbps	Giga bits por segundo
GOP	Group of Pictures
HVS	Human Visual System
I	Intertrama
ICSI	International Computer Science Institute
JPEG	Joint Photographic Experts Group
JTC1	Joint ISO/IEC Technical Committee on Information Technology
Kbps	Kilo bits por segundo
KLTL	Karhunen Loeve Transform
MHz	Mega Hertz
MPEG	Moving Picture Experts Groups

Mbps	Mega bits por segundo
P	Predecida
PET	Priority Encoding Transmission
RLE	Run Length Encoding
PSNR	Peak Signal-to-Noise Ratio
VQ	Vector Quantization
WWW	World Wide Web
Y	Señal de luminosidad

Observación: se utilizó un método consecutivo para la numeración de las ecuaciones.

1. MOTIVACIÓN – ORÍGENES Y ANTECEDENTES

El trabajo surge como una propuesta de tesis del Dr. Carlos Rodríguez Lucatero, en la cual, se busca un tipo de codificación para la transmisión de información que permita la “recuperación” adecuada de dicha información en redes con pérdidas, es decir, en aquellas redes en las cuales existen pérdidas de paquetes.

Hasta hace pocos años únicamente en los E.E.U.U. y en algunos países europeos, se contaba con la infraestructura adecuada para el transporte masivo de información proporcionado por las redes de computadoras. Sin embargo, actualmente, una gran cantidad de países en vías de desarrollo se incorporan a Internet, adoptando a las redes de datos como principal medio para el flujo de información, gracias a lo cual se obtiene el acceso a grandes cantidades de información de manera eficiente, instantánea y a muy bajo costo.

Hace 17 años, Internet comenzó como una red heterogénea de interconexiones (50 computadoras) construida por la ARPA (Advanced Research Projects Agency) [29]. En la actualidad es una red mundial que ofrece distintos servicios y con un número de usuarios que crece día con día.

Con la expansión de la comunicación proporcionada por la interfase gráfica WWW (World Wide Web) no solo se transmite texto a través de Internet, sino que también se transmiten imágenes, audio y, en algunos casos, video, lo que ha servido para plantear la opción de emplear Internet para la educación a distancia, utilizando herramientas de comunicación desarrolladas en ésta, como son las herramientas de video para la colaboración, obteniendo como principal ventaja el poder interactuar a través de Internet, característica que los usuarios estamos ansiosos de tener, con la finalidad generalizar aún más esta tecnología.

Pero debido al enorme crecimiento del tráfico de datos en las redes, provocado por la expansión de estos recursos, se producen eventos de congestión debido al tráfico en la red. Es por ello que existe un gran número de proyectos que pretenden aumentar la velocidad de transmisión y la capacidad de transporte de Internet. Sin embargo, como el desarrollo de Internet es

heterogéneo, se convierte en todo un desafío definir protocolos y algoritmos de transmisión que puedan operar sobre toda la red.

La mayoría de las redes de datos se basan en la transmisión de paquetes, los cuales son una unidad de información indivisible de longitud definida, que llegan intactos después de la transmisión o se pierden por completo, por lo que es necesario proteger dichos paquetes mientras se transportan sobre la red de datos, en particular, si las transmisiones necesitan ser en tiempo real, como lo son las videoconferencias interactivas.

Por definición, una videoconferencia interactiva es la combinación de audio, video y comunicación a través de la red con interacción en tiempo real [12], de aquí que se plantee un problema de difusión de información en una red, donde los paquetes pueden perderse, lo que produce un aumento en el tiempo real y en la tasa útil de recuperación, aspectos que disminuyen la eficiencia de la transmisión.

Lo anterior representa un gran problema debido a que en la actualidad el uso de sistemas que permitan un mejor acceso en la comunicación a larga distancia es muy cotizado y valorado por todos nosotros, no solo por la reducción de costos que esto implica sino también por las facilidades que proporciona, como son: menor pérdida de tiempo e interacción o comunicación a distancia.

La enseñanza a distancia, que emplea sistemas telemáticos, comenzó a finales de los 80's y principios de los 90's, como un sistema punto a punto, utilizando una línea de datos E1.

Los cursos también pueden transmitirse vía satélite, sin embargo, los servicios satelitales presentan diversas desventajas, como son: incremento en el costo de los recursos (ya que en ocasiones las tarifas exceden \$1000/hora [6]), capacidad y velocidad limitada, y la señal es más propensa a desvanecimientos, es por ello que la transmisión vía satélite no es una solución a bajo costo e inmune al ruido.

Sin embargo, gracias a los avances tecnológicos que se han logrado en la transmisión de datos a través de las redes (incremento en la velocidad y procesamiento, y mayor ancho de banda gracias a la fibra óptica) se pueden realizar transmisiones en tiempo real a menor costo y con menores

pérdidas, obteniendo una imagen con calidad, generalmente, buena, aunque en ocasiones el movimiento de los participantes es inapreciable, y la señal de audio puede sufrir retrasos con respecto al video (pérdidas de sincronía). Estos problemas surgen debido a la heterogeneidad en el crecimiento de las redes de datos, provocando la existencia de distintas capacidades de procesamiento y almacenamiento, anchos de banda, velocidades de transmisión y políticas de control, entre otras cosas. Es por ello que la transmisión de audio y video en tiempo real aún es problemática, ya que no existe un estándar universal para que el flujo de bits sea homogéneo sobre toda la red de datos. Una opción para resolver esto último es la presentación de iniciativas sobre estandarización en la transmisión de datos, audio, y video sobre la red. Algunos estándares para la transmisión de video sobre una red de datos son: H.320, H.323 y, en general, la familia de H.261, todos ellos utilizan algoritmos de compresión digital.

2. FUNDAMENTOS Y MARCO TEÓRICO

2.1 INTRODUCCIÓN AL VIDEO DIGITAL

La necesidad de contar con un método de compresión digital está justificada con el hecho: de lograr un algoritmo de compresión que utilice técnicas de las cuales se obtengan altos ratios de compresión con el menor sacrificio de fidelidad, de tal forma que los datos originales puedan reconstruirse después de ser transmitidos sobre la red de datos.

La necesidad de comprimir el video para su transmisión surge de las características necesarias para su procesamiento (velocidad de muestreo, cuantificación y velocidad de datos) y las características del medio de comunicación (velocidad), como se muestra en la Tabla 1 y Tabla 2.

Tabla 1. Características del procesamiento de video			
Estándar para la TV digital (CCIR 601)			
	Y	R-Y	B-Y
Velocidad de muestreo	13.5 MHz	6.75 MHz	6.75 MHz
Cuantificación	8 bits	8 bits	8 bits
Velocidad de datos	216 Mbps		
HDTV	1 Gbps		

Tabla 2. Velocidad de datos sobre diversos medios de comunicación	
Canal	Velocidad
Broadcast terrestre de TV	20 Mbps
Disco Duro	10 Mbps
CD-ROM	1.5 Mbps
Ethernet	< 10 Mbps
ISDN	64 Kbps

Analizando ambas tablas se puede observar que la velocidad requerida por el video es demasiado grande para los medios de comunicación, es por ello que se requiere la compresión del video para poder contar con sistemas de comunicación que transmitan información confiable desde la fuente hasta el usuario.

Por definición, la salida de la fuente de información es una secuencia de símbolos de un alfabeto discreto. Estos símbolos se procesan en un codificador fuente que remueve la dependencia entre símbolos y asigna palabras código a los símbolos, o a un grupo de símbolos, con el fin de minimizar el número de símbolos transmitidos hacia el usuario, considerando que no se debe exceder cierto nivel aceptable de distorsión. Para lograr ésto se utilizan dos principios:

1. Las propiedades de la fuente de información conocidas a priori representan información redundante que no es necesario transmitir (reducción de redundancia).
2. El ojo humano no percibe ciertas desviaciones entre la señal recibida y la original (reducción de irrelevancia).

La salida del codificador fuente es la entrada al codificador de canal. El codificador de canal mejora la eficiencia de la comunicación, convirtiendo la secuencia de salida del codificador fuente en una secuencia de símbolos de un alfabeto (posiblemente) diferente. La salida del codificador se transmite a través de un DMC (Discrete Memoryless Channel) hacia el decodificador de canal.

Definiendo una fuente sin memoria como el conjunto U_0 con el alfabeto $\{a_0, a_1, a_2, \dots, a_{K-1}\}$ y probabilidades $\{P(a_0), P(a_1), P(a_2), \dots, P(a_{K-1})\}$, donde

$$I(a_k) = -\log_2(P(a_k)) \quad (1)$$

es la medida de información del mensaje " a_k ", y

$$\begin{aligned}
 H(U_0) &= E\{I(a_k)\} = \sum_{k=0}^{K-1} P(a_k) \cdot \log_2(P(a_k)) \\
 &= -\sum_{u_0} (P(u_0)) \cdot \log_2(P(a_k))
 \end{aligned}
 \tag{2}$$

es la entropía de la fuente, es decir, el contenido de información promedio y el límite inferior para la longitud de la palabra código promedio.

El objetivo es explotar la memoria de la fuente, un método para lograr esto es utilizar un codificador predictivo (Figura 1), el cual se basa en los siguientes principios:

- S_0 es la muestra actual de la señal original S
- \hat{S} es una predicción de S_0 calculada a partir las muestras anteriores S_1, S_2, \dots, S_N
- e es la predicción de error
- El receptor puede reconstruir S sin pérdidas

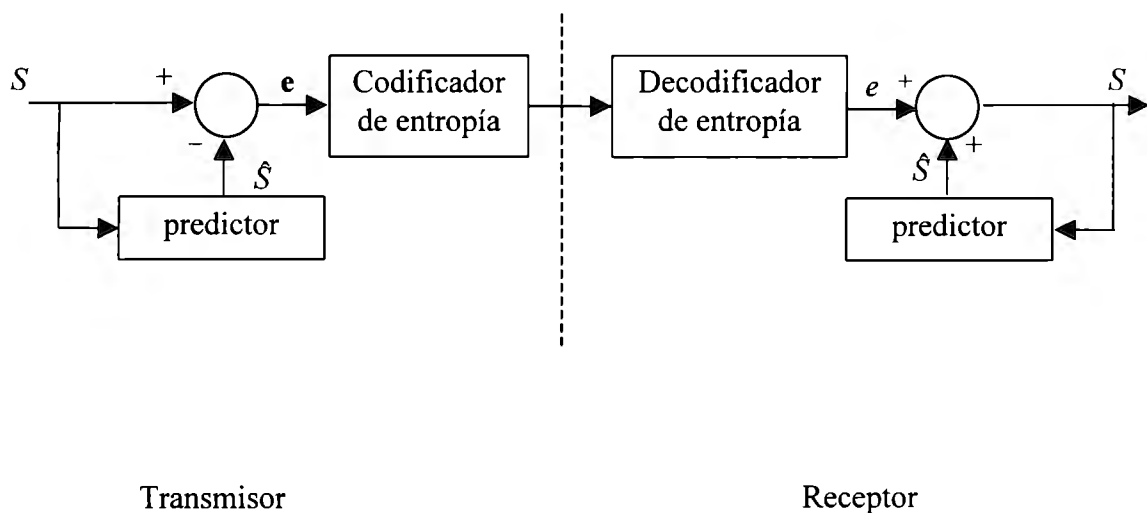


Figura 1. Codificador Predictivo

En las Figuras 2 y 3 se muestra el transmisor y receptor DPCM, con las siguientes características

$$e = s - \hat{s}$$

$$s' = e' - \hat{s}$$

$$s' - s = e' - e = q \quad (3)$$

donde q es el error de cuantificación, lo que significa que la reconstrucción del error es igual a la cuantificación del error.

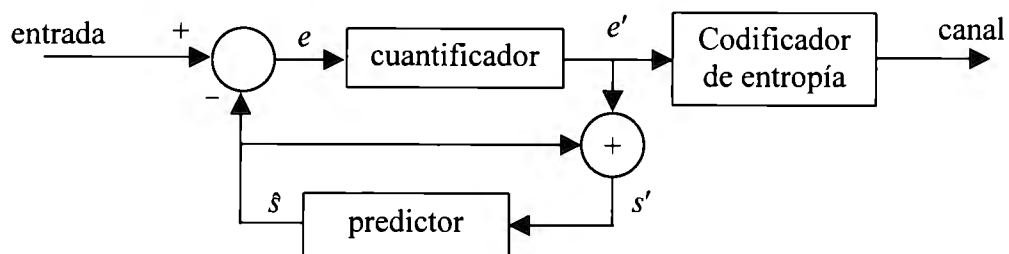


Figura 2. Transmisor DPCM

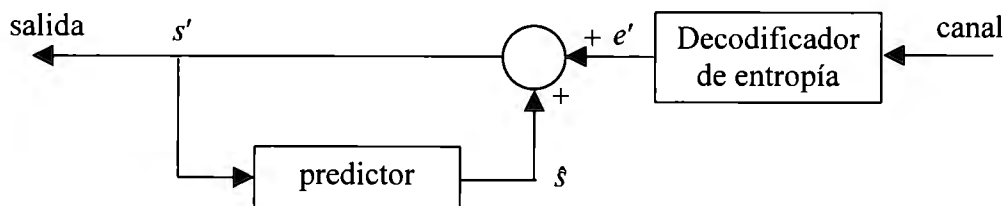


Figura 3. Receptor DPCM

Las distorsiones producidas por esta codificación son:

- Ruido granular: ruido aleatorio en áreas planas de la imagen
- Ocupación de la orilla: discontinuidad, apariencia inquieta de las orillas
- Inclinación de sobrecarga: trazos borrosos de orillas con alto contraste, patrones de moaré en estructuras periódicas

2.1.1 TRANSFORMADAS

Haciendo referencia al teorema de Parseval, una transformada es la rotación de un vector señal alrededor del origen en un espacio de $N \times N$ dimensiones.

Entre los métodos de transformadas propuestos para la concentración de energía se encuentran los siguientes:

- KLT (Karhunen Loeve Transform)
- Transformada de Haar (1910)
- Transformada de Walsh (Transformada de Hadamard, 1923)
- Transformada de Slant (Enomoto, Shibata, 1971)
- DCT (Discrete Cosine Transform) (Ahmed, Natarajan, Rao, 1974)

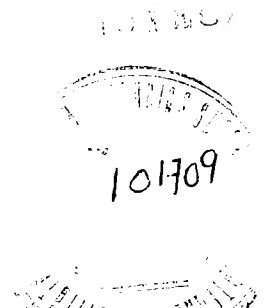
Para la selección de un método de transformada existen diversos criterios que deben tomarse en cuenta:

- Decorrelación, concentración de energía (KLT, DCT,...)
- Funciones bases visualmente placenteras (m secuencias de ruido pseudoaleatorio)
- Baja complejidad computacional

2.1.2 DCT

La DCT realiza un mapeo uno a uno de un arreglo de valores de pixeles a un arreglo de valores en el dominio de la frecuencia.

La DCT no provee compresión, sin embargo “acomoda” la información de la imagen de forma más adecuada para que otras técnicas de compresión reduzcan la precisión de los coeficientes de la DCT de acuerdo al modelo HVS (Human Visual System).



2.1.3 CUANTIFICACIÓN

La cuantificación es el medio más importante de la reducción de irrelevancia. Como resultado de la DCT, se obtiene cierto orden en los componentes de frecuencia: los coeficientes de menor frecuencia están más cerca a la esquina superior izquierda de la matriz de coeficientes y corresponden al contorno espacial; el elemento de la esquina superior izquierda corresponde al valor de luminosidad o color para el bloque; y los coeficientes de mayor frecuencia están más cerca a la esquina inferior derecha de la matriz de coeficientes y corresponden a patrones espaciales más finos o al ruido en la imagen.

Como el HVS es menos sensible a errores en los coeficientes de altas frecuencias, es deseable que se cuantifiquen más toscamente.

En el proceso de cuantificación cada coeficiente se divide entre el valor de la matriz de cuantificación. El objetivo de la cuantificación es forzar a que la mayoría de los coeficientes de la DCT sean cero o un valor cercano a éste, tanto como sea posible dentro de los límites de la velocidad de bits prescritos y los parámetros de calidad de video.

Como se mencionó anteriormente la DCT no pierde información, sin embargo, el proceso de cuantificación sí.

La posición de los coeficientes de la transformada distintos de cero se transmiten junto con sus valores de amplitud.

Una codificación eficiente para colocar de forma adecuada los coeficientes de la transformada distintos de cero para su transmisión es mediante el escaneo en zigzag y codificar la matriz mediante un método RLE (Run Length Encoding), el cual substituye las secuencias de valores consecutivos.

Analizando la matriz de coeficientes cuantificada se puede observar que bastante energía se concentra en la porción de baja frecuencia de la matriz y que varios de los coeficientes de alta frecuencia se cuantificaron como cero.

El propósito es representar varios ceros de forma más eficiente por lo que los coeficientes de la cuantificación se reordenan. Como muchos de los coeficientes distintos de cero se concentraron en la esquina superior izquierda de la matriz, siguiendo un patrón en zigzag (Figura 4) se logra tener un mayor número de ceros consecutivos, los cuales pueden codificarse mediante RLE.

Estos valores se buscan en una tabla predefinida de códigos de longitud variable, donde la ocurrencia más probable tiene un código relativamente pequeño y la ocurrencia menos probable uno más largo. Los valores poco frecuentes se codifican con secuencias de escape para prevenir que las tablas de códigos sean muy grandes.

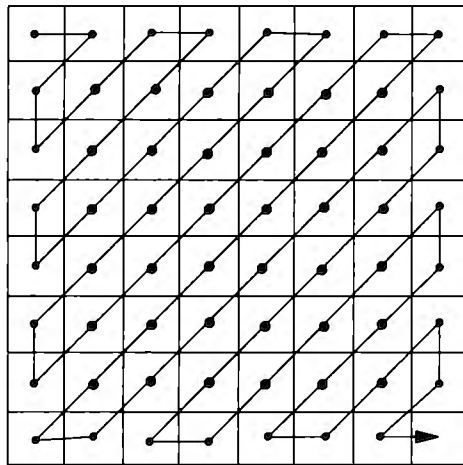


Figura 4. Patrón de zigzag

2.1.4 CODIFICACIÓN INTERTRAMA

La compresión intertrama asume que a pesar de que ocurra algún movimiento, el fondo de la imagen permanece igual en la mayor parte del video, por lo que no es necesario comprimir la trama completa, sino únicamente las diferencias entre tramas, usando una imagen de referencia, en donde los bloques de pixeles similares, comunes a dos o más tramas sucesivas, serán reemplazados con un apuntador de referencia hacia otro de los bloques.

Lo más complicado para lograr la codificación es definir el orden de las tramas, ya que en ocasiones la referencia se hace con un bloque de una trama anterior y en otras con un bloque de la trama posterior, en cuyo caso, el codificador reemplaza al bloque con un apuntador y revierte el

orden de la trama. La rutina de descompresión deberá colocar las tramas en el orden apropiado para desplegarlas correctamente.

Entre los métodos más importantes de codificación intertrama se encuentran: predicción con compensación de movimiento e interpolación con compensación de movimiento.

2.1.5 PREDICCIÓN CON COMPENSACIÓN DE MOVIMIENTO

La predicción con compensación de movimiento se basa en la estimación del movimiento, considerando que en la mayoría de los casos las tramas consecutivas de video son similares, excepto por los cambios provocados por los objetos en movimiento.

En la Figura 5 se muestra a grandes rasgos el proceso de la predicción.

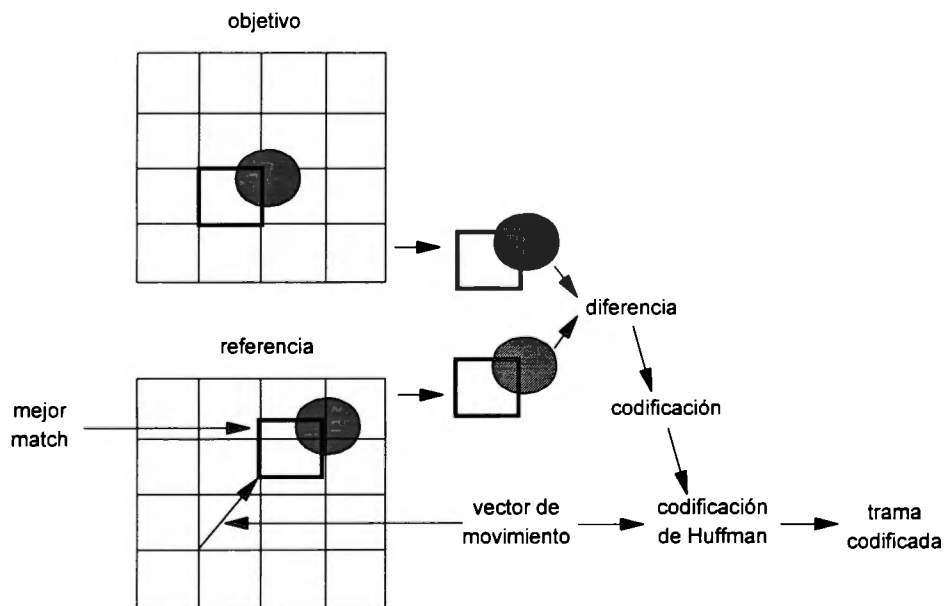


Figura 5. Predicción

En el caso trivial, es decir, cuando no existe movimiento, la trama actual se predice como un duplicado de la trama anterior, por lo que la única información necesaria para la transmisión es el encabezado requerido para reconstruir la imagen de la trama de referencia.

Cuando existe movimiento, la codificación se hace con una trama de referencia que precede a la trama actual. El vector de movimiento proporciona el desplazamiento del macrobloque de la trama actual con respecto a su trama de referencia, y se tiene un término de error que representa las diferencias de píxeles.

Para obtener la estimación del movimiento se cuenta con los siguiente métodos: métodos diferenciales, métodos de semejanza, métodos de dominio de frecuencia y métodos basados en características.

El método diferencial calcula el desplazamiento a partir de diferencias espaciales y temporales en la señal imagen y requiere de iteración.

El método de semejanza consiste de los siguientes pasos:

- Subdividir cada imagen en bloques cuadrados
- Encontrar un vector de desplazamiento para cada bloque
- Encontrar la semejanza que minimice/maximice un criterio de semejanza dentro de un rango de búsqueda.

2.1.6 INTERPOLACIÓN CON PREDICCIÓN DE MOVIMIENTO

Algunas tramas pueden codificarse usando dos tramas de referencia, una anterior y otra posterior, a este tipo de compresión se le llama interpolación bidireccional, y es de la cual resulta el mayor radio de compresión (500:1)¹. Sin embargo, no es conveniente emplearla en todas las aplicaciones, ya que para la codificación requiere la secuencia de transmisión, lo cual provoca latencia.

Para realizar la codificación la trama actual se procesa contra dos tramas de referencia, y se calculan dos vectores de movimiento, uno para cada trama de referencia.

2.1.7 JPEG

JPEG (Joint Photographic Experts Group) es un grupo de estandarización de la ITU-T e ISO, el estándar JPEG se emplea para la compresión de imágenes sin movimiento en escala de grises y a color, utilizando codificación intratrama, diseñado especialmente para imágenes sin movimiento. Las características que presenta son las siguientes:

- Las dimensiones de la imagen son flexibles
- Cada componente se codifica por separado
- Cualquier espacio de color es posible, sin embargo la mejor compresión es con Y/R-Y/B-Y
- Las relaciones de compresión son flexibles
- La compresión es de 24:1 para imágenes CCIR 601 sin pérdida de calidad

La codificación de los coeficientes DCT se realiza de la siguiente forma:

- codificación diferencial del coeficiente DC (0,0)
- escaneo en zigzag y RLC de los coeficientes AC,
- se cuenta con dos opciones para la codificación Huffman: con tablas código predefinidas (sistemas de un paso) con tablas código adaptables a la imagen (sistemas de dos pasos).

En la Figura 6 se muestra el algoritmo de codificación básico.

En la Tabla 3 se muestran las matrices de tabulación porcentual adaptadas a la función de transferencia del sistema visual humano que se definen en el codificador.

¹ SMITH, BRIAN C. A Survey of Compressed Domain Processing Techniques

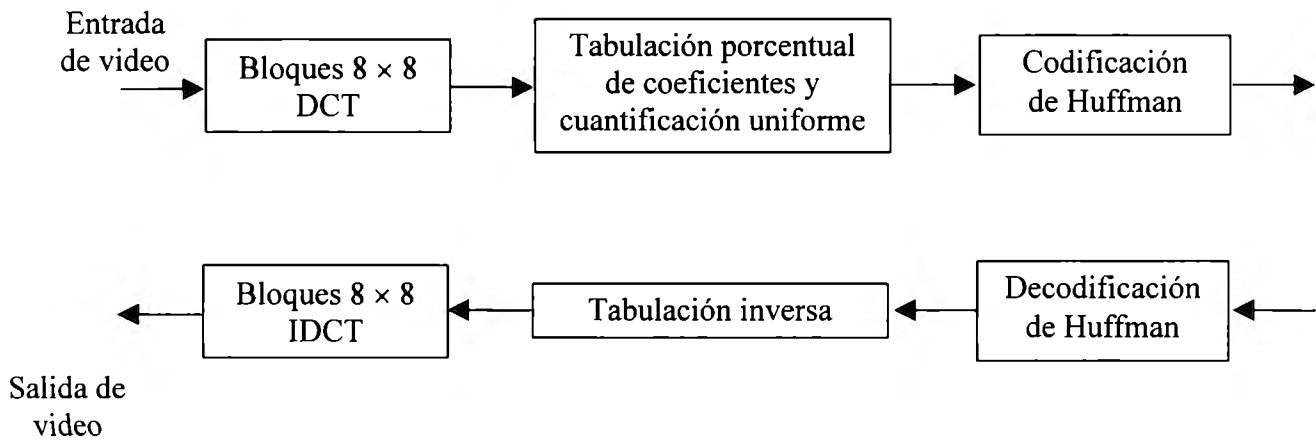


Figura 6. Algoritmo básico de JPEG

Tabla 3. Matrices de ponderación															
Luz								Color							
16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	58	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

2.1.8 MPEG

Uno de los estándares utilizados para la compresión de video es el propuesto por el MPEG (Moving Picture Experts Groups) [7], [17], grupo del comité de estándares internacionales utilizados para la codificación de información de audio y video en formato de compresión digital. El grupo de trabajo de MPEG se estableció en 1988 y pertenece a la JTC1 (Joint ISO/IEC Technical Committee on Information Technology).

El estándar MPEG se dividió en tres tipos, de MPEG-1, MPEG-2 y MPEG-4, los cuales no son actualizaciones de las versiones anteriores, sino, variantes independientes hechas con el fin de responder a distintos anchos de banda y restricciones en la calidad del video digital.

Las ventajas que el estándar MPEG ofrece ante otros esquemas de compresión/descompresión es que tiene compatibilidad mundial, ya que fue diseñado como un estándar internacional; ofrece mayores relaciones de compresión que cualquier otro formato, reduciendo el tamaño de entrada de video en relación 200:1 aproximadamente, pero lo más importante es que ofrece compatibilidad y compresión con el menor sacrificio de calidad.

La compresión de video de MPEG se basa en restricciones del HVS. Estudios recientes en el HVS muestran que el ojo es más sensible a los cambios en la luminosidad y menos sensible a variaciones en el color. Por ello, MPEG opera en el espacio de color, donde eficientemente toma ventaja de la sensibilidad del ojo a la información de luminosidad y color.

MPEG utiliza el espacio de color YCbCr, donde Y es la señal de luminosidad, Cb es la señal de diferencia del color azul y Cr es la señal de diferencia del color rojo, empleando una representación 4:2:0 (Figura 7), lo que permite una reducción de datos.

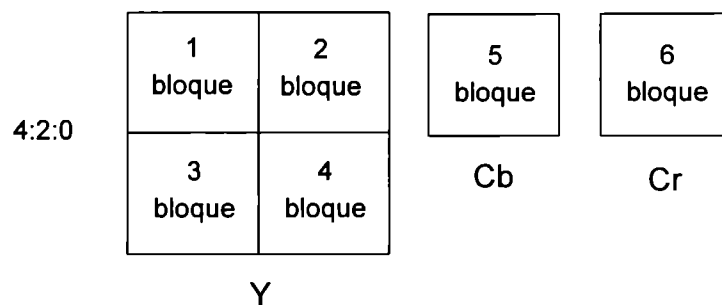


Figura 7. Representación 4:2:0

MPEG tiene una estructura en capas (Figura 8), organizada por las características del flujo de video y empleadas para satisfacer distintos requerimientos de compresión.

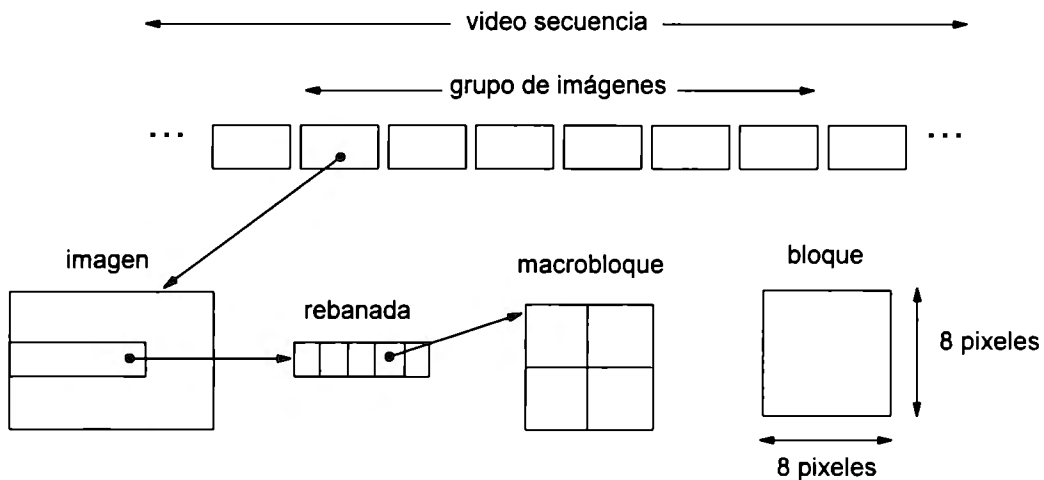


Figura 8. Estructura en capas

La capa superior es la secuencia de video. Esta capa contiene las imágenes que serán procesadas por el sistema, como una película o un anuncio.

La capa GOP agrupa las imágenes individuales de una secuencia de video. Está formada por imágenes de referencia e imágenes que pueden predecirse de la anteriores.

La capa imagen es la unidad primaria de la secuencia de video. Las imágenes son equivalentes a una trama de película. Una imagen consiste de tres matrices rectangulares: una de luminosidad Y (brillantez) y dos de color, Cr y Cb.

Las rebanadas proveen acceso tolerante a fallas en una imagen, es decir, son la unidad de recuperación de error y resincronización. El objetivo de una rebanada es que al ocurrir un error de decodificación, la rebanada puede desecharse y la decodificación comience en la siguiente rebanada, con lo que se evita desechar la trama completa.

Una rebanada puede ser una línea de la imagen o estar formada por cualquier número de macrobloques. El tamaño de la rebanada depende de la calidad del canal de comunicación. Si se esperan pocos errores, una rebanada puede ser una imagen completa.

El macrobloque combina la información de luminosidad y color. El tamaño de un macrobloque es de 16×16 componentes de luminosidad y los dos correspondientes componentes de 8×8 de color (Figura 7).

La capa de bloque es la unidad mínima en la compresión. El tamaño de un bloque es de 8×8 componentes de luminosidad o de color.

Las tramas que se emplean en el estándar MPEG son:

- Intertrama (I): codificada como un imagen inmóvil independiente.
- Predecida (P): codificada con una trama de referencia anterior P ó I.
- Interpolación bidireccional (B): codificada con una trama de referencia anterior P ó I, y una posterior P ó I.

El estándar MPEG emplea dos técnicas de compresión: intratrama e intertrama. La compresión intratrama solo se basa en la información actual de la imagen (tramas I), y la compresión intertrama en la información de imágenes que tienen relación (tramas P y B).

El proceso de compresión comienza tomando muestras de cada pixel, que será representado con tres muestras: una de luminosidad (Y) y dos de color (Cr, Cb), donde cada muestra, por separado, se organiza en matrices de 8×8 .

Las matrices obtenidas se transforman con DCT, y se cuantifican empleando la matriz de cuantificación mostrada en la Tabla 4.

Finalmente, se realiza el escaneo en zigzag, se aplica RLE y Huffman.

En la compresión intertrama, la unidad mínima para la compensación de movimiento es el macrobloque. Cada macrobloque es una unidad diferente y únicamente los macrobloques que presentan cambios serán enviados.

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

La estimación de movimiento se resuelve mediante el problema de búsqueda espacial bidimensional, que será realizada en la luminosidad de cada macrobloque.

En caso de existir movimiento, una trama se divide en sus macrobloques, cada uno de los cuales se codifica por separado. La codificación se hace con una trama de referencia (trama I ó P más cercana) que preceda la trama actual.

El estándar MPEG permite compresión de interpolación bidireccional. Sin embargo, como se mencionó anteriormente, no es conveniente utilizarla en todas las aplicaciones, ya que se requiere de la secuencia de transmisión.

Para decodificar una trama MPEG deben decodificarse todas las tramas de referencia relevantes, descomprimir los términos de error, y el resultado se agrega a la trama de referencia especificada por el vector de movimiento.

Para realizar la decodificación debe considerarse que las tramas B requieren mayor tiempo, por lo que deben enviarse después de sus tramas de referencia. El receptor almacena las imágenes intermedias y presenta el orden correcto para desplegar las imágenes (como se verá más adelante).

El diseño del GOP está restringido por las siguientes propiedades [ISO11172]:

I P B B P B I P (03125476)

orden del flujo de video

El decodificador recibe las tramas de referencia y las decodifica, pero no las despliega, sino que las mantiene en memoria hasta que sea su turno en el orden de desplegado, es decir, el orden en el que las tramas deben aparecer en la pantalla.

I B B P B P B I (01234567)

orden de desplegado

Sin embargo, cuando la información de video es paquetizada los datos se corrompen debido a retrasos en la red, fenómeno común en ciertas redes. Dicha pérdida de paquetes no se dispersa uniformemente por la red, ya que varía entre distintos sitios y fluctúa en el tiempo, por lo que es necesario realizar cierto tipo de codificación, codificación de canal, que permita la recuperación de la información, y que en conjunto con la codificación fuente logre obtener un esquema de codificación que permita la transmisión de información confiable.

2.2 ELEMENTOS DE CODIFICACIÓN FUENTE CANAL

2.2.1 TEORÍA DE CODIFICACIÓN DE CANAL

Considerando una fuente de información BSS (Binary Symmetric Source), es decir, que la información consiste de símbolos binarios igualmente probables e independientes, sea $\mathcal{X} = \{0,1\}$ el alfabeto de la fuente y un bit de información $u \in \mathcal{X}$, empleando para el arreglo de secuencias subíndices, es decir, los arreglos de secuencias de K bits de información se denotan como \mathcal{X}_K .

Asumiendo un canal BSC (Figura 10) con probabilidad de crossover ε , sean $\mathcal{X} = \{0,1\}$ y $\mathcal{Y} = \{0,1\}$ los alfabetos de entrada y salida del BSC, respectivamente, donde \mathcal{X}_N y \mathcal{Y}_N denotan los arreglos de secuencias de longitud N .

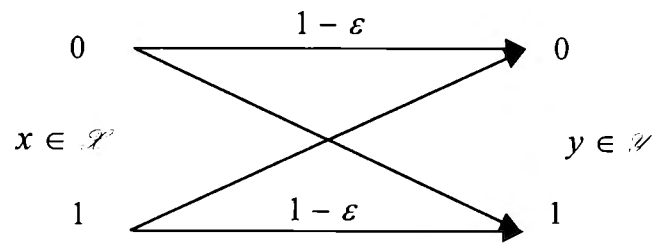


Figura 10. BSC

El símbolo de entrada del canal $x \in \mathcal{X}$ y el símbolo de salida del canal $y \in \mathcal{Y}$ tienen la probabilidad condicional de canal

$$P(y|x) = \begin{cases} \varepsilon & y \neq x \\ 1 - \varepsilon & y = x \end{cases} \quad (4)$$

Entonces para la secuencia de entrada del canal $\underline{x} = (x_1, x_2, \dots, x_N) \in \mathcal{X}_N$ y la secuencia de salida del canal $\underline{y} = (y_1, y_2, \dots, y_N) \in \mathcal{Y}_N$ se tiene la probabilidad condicional de canal

$$\begin{aligned} P_N(\underline{y} | \underline{x}) &= \prod_{n=1}^N P(y_n | x_n) \\ &= \varepsilon^{\omega(\underline{y}, \underline{x})} (1 - \varepsilon)^{N - \omega(\underline{y}, \underline{x})} \end{aligned} \quad (5)$$

donde $\omega = \left(\underline{y}, \underline{x} \right)$ es el número de componentes $y_n \neq x_n$; $n = 1, 2, \dots, N$. $\omega = \left(\underline{y}, \underline{x} \right)$ es la distancia de Hamming entre las secuencias binarias \underline{y} y \underline{x} [20].

Para enviar bits de información de forma confiable sobre el BSC, se debe agregar cierta redundancia a los bits de información con el fin de permitir la corrección de errores. Suponiendo que para K bits de información se envían N bits de entrada al canal como se muestra en la Figura 11, la cantidad de redundancia agregada se mide con la velocidad de información

$$R = \frac{K}{N} \quad \text{bits de información/bits de entrada al canal} \quad (6)$$

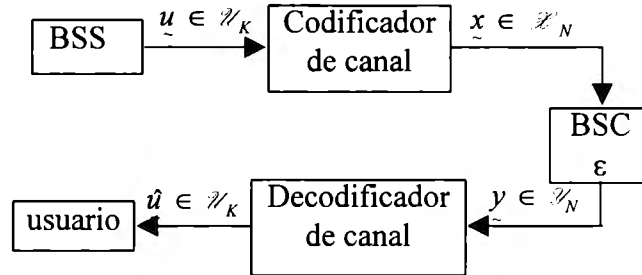


Figura 11. Sistema de Codificación Canal Básico

El codificador de canal únicamente asigna una palabra código a cada una de las 2^K secuencias de bits de información posibles. Sea $\{x_1, x_2, \dots, x_M\}$ donde $M = 2^K = 2^{RN}$ son las palabras código, por lo que solo existen $M = 2^K$ secuencias posibles de entrada de canal, sin embargo, debido a posibles errores de canal, pueden ocurrir cualquiera de las 2^N secuencias posibles de salida de canal. Entonces el decodificador canal toma $y \in \mathbb{Z}'_N$ y decide cual de las palabras código $\{x_1, x_2, \dots, x_M\}$ fue enviada. Al decidir la palabra código \hat{x} que fue transmitida, el decodificador puede obtener la secuencia de bits de información $\hat{u} \in \mathbb{Z}'_K$ correspondientes.

La regla de decisión de un decodificador determinístico puede expresarse mediante funciones de decisión:

$$\delta_m(y) = \begin{cases} 1 & \text{si el decodificador decide } x_m \text{ dado } y \in \mathbb{Z}'_N \\ 0 & \text{en otro caso} \end{cases} \quad (7)$$

$m = 1, 2, \dots, M$

Ya que únicamente se asume una decisión para cada secuencia $y \in \mathbb{Z}'_N$ de salida del canal, para cada secuencia se tiene

$$\sum_{m=1}^M \delta_m(\underline{y}) = 1 \quad (8)$$

Un criterio de decisión es la minimización de la probabilidad de error

$$P_e = \Pr\{\hat{\underline{u}} \neq \underline{u}\} \quad (9)$$

o, la minimización de la probabilidad de decisión correcta

$$\begin{aligned} P_c &= 1 - P_e \\ &= \Pr\{\hat{\underline{u}} = \underline{u}\} \end{aligned} \quad (10)$$

donde para cualquier arreglo de palabras código distintas $\{\underline{x}_1, \underline{x}_2, \dots, \underline{x}_M\}$ y cualquier regla de decisión se tiene

$$\begin{aligned} P_c &= \Pr\{\hat{\underline{u}} = \underline{u}\} \\ &= \sum_{m=1}^M \Pr\{\hat{\underline{u}} = \underline{u}_m\} P_N(\underline{u}_m) \\ &= \frac{1}{M} \sum_{m=1}^M \Pr\{\hat{\underline{x}} = \underline{x}_m \mid \underline{x}_m\} \\ &= \frac{1}{M} \sum_{m=1}^M \sum_{\underline{y}} \delta_m(\underline{y}) P_N(\underline{y} \mid \underline{x}_m) \\ &= \frac{1}{M} \sum_{\underline{y}} \left(\sum_{m=1}^M \delta_m(\underline{y}) P_N(\underline{y} \mid \underline{x}_m) \right) \end{aligned} \quad (11)$$

Como P_c se maximiza sobre la regla de si decisión para cada $\underline{y} \in \mathcal{Z}_N$, $\delta_m(\underline{y})$, $m = 1, 2, \dots, M$, se decide maximizar

$$\sum_{m=1}^M \delta_m(\underline{y}) P_N(\underline{y} \mid \underline{x}_m) \quad (12)$$

El decodificador óptimo es el decodificador de máxima semejanza, que tiene por regla de decisión

$$\delta_m(\underline{y}) = \begin{cases} 1 & \text{si } P_N(\underline{y} | \underline{x}_m) > P_N(\underline{y} | \underline{x}_{m'}) \text{ para toda } m' \neq m \\ 0 & \text{en otro caso} \end{cases} \quad (13)$$

Para el BSC, de la ecuación (5) se tiene

$$P_N(\underline{y} | \underline{x}) = \left(\begin{matrix} \varepsilon \\ 1 - \varepsilon \end{matrix} \right)^{\omega(\underline{y}, \underline{x})} (1 - \varepsilon)^N \quad (14)$$

si $\varepsilon < 1/2$ la regla de decisión de máxima semejanza se convierte en la regla de distancia mínima de Hamming

$$\delta_m(\underline{y}) = \begin{cases} 1 & \text{si } \omega(\underline{y}, \underline{x}_m) < \omega(\underline{y}, \underline{x}_{m'}) \text{ para toda } m' \neq m \\ 0 & \text{en otro caso} \end{cases} \quad (15)$$

2.2.1.1 Códigos de bloque

Considerando un código $\mathcal{C} = \{ \underline{x}_1, \underline{x}_2, \dots, \underline{x}_M \}$ en el sistema de la Figura 11, sin asumir restricciones de como se generan las palabras código de cada una de las M secuencias posibles de bits de información $\{ \underline{u}_1, \underline{u}_2, \dots, \underline{u}_M \}$ entonces cualquier componente de \underline{x}_m puede depender de cualquier componente u_m .

2.2.1.2 Teorema de codificación converso

De la ecuación (11) se observa que dado un código \mathcal{C} la probabilidad de decisión correcta del decodificador es

$$\begin{aligned}
 P_c &= \frac{1}{M} \sum_{\underline{y}} \left(\sum_{m=1}^M \delta_m(\underline{y}) P_N(\underline{y} | \underline{x}_m) \right) \\
 &= \frac{1}{M} \sum_{\underline{y}} \max_m P_N(\underline{y} | \underline{x}_m)
 \end{aligned}
 \tag{16}$$

asumiendo un decodificador óptimo de máxima semejanza. Para cualquier $-1 < \rho$ se tiene

$$\begin{aligned}
 \max_m P_N(\underline{y} | \underline{x}_m) &= \left(\max_m P_N(\underline{y} | \underline{x}_m)^{\frac{1}{1+\rho}} \right)^{1+\rho} \\
 &\leq \left(\sum_{m=1}^M P_N(\underline{y} | \underline{x}_m)^{\frac{1}{1+\rho}} \right)^{1+\rho}
 \end{aligned}
 \tag{17}$$

Definiendo una distribución de probabilidad basada en palabras código,

$$Q_N(\underline{x}) = \begin{cases} \frac{1}{M} & \underline{x} = \underline{x}_m \quad m = 1, 2, \dots, M \\ 0 & \text{en otro caso} \end{cases}
 \tag{18}$$

se obtiene el límite

$$\begin{aligned}
\max_m P_N(\underline{y} | \underline{x}_m) &\leq \left(M \sum_{\underline{x}} Q_N(\underline{x}) P_N(\underline{y} | \underline{x}_m)^{\frac{1}{1+\rho}} \right)^{1+\rho} \\
&= M^{1+\rho} \left(\sum_{\underline{x}} Q_N(\underline{x}) P_N(\underline{y} | \underline{x}_m)^{\frac{1}{1+\rho}} \right)^{1+\rho}
\end{aligned} \tag{19}$$

substituyendo en la ecuación (16), se tiene el límite en P_c

$$P_c \leq M^\rho \sum_{\underline{y}} \left(\sum_{\underline{x}} Q_N(\underline{x}) P_N(\underline{y} | \underline{x})^{\frac{1}{1+\rho}} \right)^{1+\rho} \tag{20}$$

El límite en P_c depende del código \mathcal{C} sobre la distribución de probabilidad $\{Q_N(x): x \in \mathcal{X}_N\}$, lo cual se limita aún más y se elimina esta dependencia del código maximizando sobre todas las distribuciones de probabilidad. Entonces

$$P_c \leq M^\rho \max_{Q_N} \sum_{\underline{y}} \left(\sum_{\underline{x}} Q_N(\underline{x}) P_N(\underline{y} | \underline{x})^{\frac{1}{1+\rho}} \right)^{1+\rho} \tag{21}$$

Cuando $-1 < \rho \leq 0$ la distribución de maximización es la distribución uniforme

$$Q_N(\underline{x}) = \frac{1}{2^N} \quad \text{para toda } \underline{x} \in \mathcal{X}_N \tag{22}$$

lo cual se muestra derivando después de agregar el término multiplicador de Lagrange para la restricción $\sum_x Q_N(x) = 1$. Entonces para $-1 < \rho \leq 0$ se tiene

$$\begin{aligned}
P_c &\leq M^\rho \sum_{\underline{y}} \left(\sum_{\underline{x}} \left(\frac{1}{2} \right)^N P_N(\underline{y} | \underline{x})^{\frac{1}{1+\rho}} \right)^{1+\rho} \\
&= 2^{\rho RN} 2^N \left[\frac{\varepsilon^{\frac{1}{1+\rho}} + (1-\varepsilon)^{\frac{1}{1+\rho}}}{2} \right]^{N(1+\rho)} \\
&= 2^{-N[E_0(\rho) - \rho R]}
\end{aligned} \tag{23}$$

donde

$$E_0 = \rho - (1 + \rho) \log \left[\varepsilon^{\frac{1}{1+\rho}} + (1 - \varepsilon)^{\frac{1}{1+\rho}} \right] \tag{24}$$

Diferenciado $E_0(\rho)$ con respecto a ρ se produce

$$E_0' = 1 - \mathcal{H}(\delta) \tag{25}$$

donde

$$\mathcal{H}(\delta) = -\delta \log \delta - (1 - \delta) \log(1 - \delta) \tag{26}$$

es la función binaria de entropía y δ se define como

$$\delta = \frac{\varepsilon^{\frac{1}{1+\rho}}}{\varepsilon^{\frac{1}{1+\rho}} + (1 - \varepsilon)^{\frac{1}{1+\rho}}} \tag{27}$$

Derivando dos veces se muestra que $E_0''(\rho) < 0$. Por lo que $E_0(\rho)$ tiene la forma mostrada en la Figura 12, donde C se define como

$$\begin{aligned}
 C &= E_0'(0) \\
 &= 1 - \mathcal{H}(\varepsilon)
 \end{aligned}
 \tag{28}$$

la capacidad del canal BSC.

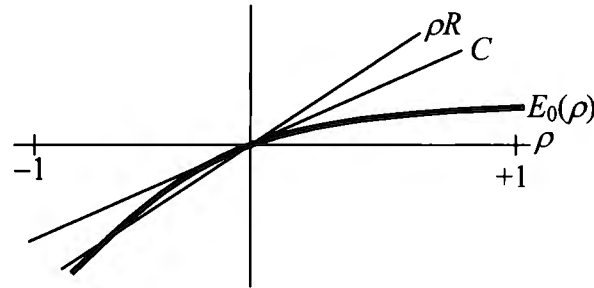


Figura 12. Curva $E_0(\rho)$

Como el límite de la ecuación 26 se aplica para todo $-1 < \rho \leq 0$, se define

$$E_c(R) = \sup_{-1 < \rho \leq 0} [E_0(\rho) - R]
 \tag{29}$$

Para $C < R < 1$, la ρ maximizante satisface

$$\begin{aligned}
 R &= E_0'(\rho) \\
 &= 1 - \mathcal{H}(\delta)
 \end{aligned}
 \tag{30}$$

Donde δ es un parámetro libre que puede variar de 0 a ε , de donde se obtiene la forma paramétrica de $E_c(R)$

$$E_c(R) = T_\varepsilon(\delta) - \mathcal{H}(\delta)
 \tag{31}$$

y

$$R = 1 - \mathcal{H}(\delta)$$

donde

$$T_\varepsilon(\delta) = -\delta \log \varepsilon - (1 - \delta) \log(1 - \varepsilon) \quad (32)$$

Entonces para cualquier código \mathcal{C} la probabilidad de correcta decodificación está limitada por

$$P_c \leq 2^{-NE_c(R)} \quad (33)$$

donde $E_c(R) > 0$ para $C < R \leq 1$.

$E_c(R) > 0$ se debe obtiene de la ecuación 31 y puede observarse gráficamente en la Figura 12, donde para ciertos $-1 < \rho < 0$, $E_0(\rho) - \rho R > 0$.

Sin importar el código empleado, la probabilidad de decisión correcta decrece exponencialmente en la longitud del bloque N , y la velocidad R es mayor que $C = 1 - \mathcal{H}(\varepsilon)$. Por lo que C depende del BSC y es la capacidad del canal.

Límite aleatorio de codificación

Este es el caso de un sistema de codificación de bloques con velocidad menor que la capacidad del canal, es decir, $R < C$.

Se asume un código $\mathcal{C} = \{x_1, x_2, \dots, x_M\}$ y el decodificador de máxima semejanza. La probabilidad de decodificación errónea de un bloque es

$$P_e = \frac{1}{M} \sum_{m=1}^M \sum_y \left(1 - \delta_m(\underline{y}) \right) P_N(\underline{y} | x_m) \quad (34)$$

Si $\rho > 0$, la regla de decodificación de máxima semejanza satisface

$$\begin{aligned}
1 - \delta_m(\underline{y}) &\leq \frac{\max_{m' \neq m} P_N(\underline{y} | \underline{x}_{m'})^{\frac{\rho}{1+\rho}}}{P_N(\underline{y} | \underline{x}_m)^{\frac{\rho}{1+\rho}}} \\
&\leq \left[\frac{\sum_{m' \neq m} P_N(\underline{y} | \underline{x}_{m'})^{\frac{\rho}{1+\rho}}}{P_N(\underline{y} | \underline{x}_m)^{\frac{\rho}{1+\rho}}} \right]^\rho
\end{aligned} \tag{35}$$

Entonces

$$P_e \leq \frac{1}{M} \sum_{m=1}^M \sum_{\underline{y}} P_N(\underline{y} | \underline{x}_m)^{\frac{1}{1+\rho}} \left[\sum_{m' \neq m} P_N(\underline{y} | \underline{x}_{m'})^{\frac{1}{1+\rho}} \right]^\rho \tag{36}$$

Seleccionando aleatoriamente un código $\mathcal{C} = \{x_1, x_2, \dots, x_M\}$ de acuerdo a la probabilidad

$$\begin{aligned}
Q(\mathcal{C}) &= Q(x_1, x_2, \dots, x_M) \\
&= \prod_{m=1}^M Q_N(x_m)
\end{aligned} \tag{37}$$

Cada palabra código se escoge independientemente, de acuerdo a la distribución $\{Q_N(x) : x \in \mathcal{X}_N\}$. Promediando sobre todos los códigos se tiene

$$P_e \leq \frac{1}{M} \sum_{m=1}^M \sum_{\underline{y}} P_N(\underline{y} | \underline{x}_m)^{\frac{1}{1+\rho}} \left[\sum_{m' \neq m} P_N(\underline{y} | \underline{x}_{m'})^{\frac{1}{1+\rho}} \right]^\rho \tag{38}$$

Para $0 \leq \rho \leq 1$, de la desigualdad de Jensen se obtiene $[f(x)]^\rho \leq [f(x)]^\rho$. Entonces

$$\begin{aligned}
P_e &\leq \sum_{\underline{y}} P_N(\underline{y} | \underline{x}_m)^{\frac{1}{1+\rho}} \left[(M-1) P_N(\underline{y} | \underline{x})^{\frac{1}{1+\rho}} \right]^\rho \\
&\leq M^\rho \sum_{\underline{y}} \left[P_N(\underline{y} | \underline{x})^{\frac{1}{1+\rho}} \right]^{1+\rho}
\end{aligned} \tag{39}$$

ó

$$P_e \leq M^\rho \sum_{\underline{y}} \left[\sum_{\underline{x}} Q_N(\underline{x}) P_N(\underline{y} | \underline{x}_m)^{\frac{1}{1+\rho}} \right]^{1+\rho} \quad 0 \leq \rho \leq 1 \tag{40}$$

Como puede observarse presenta la misma forma que el límite P_C de la ecuación 20.

Al seleccionar (para $0 \leq \rho \leq 1$, distribución que minimiza el límite sobre P_e)

$$Q_N(\underline{x}) = \frac{1}{2^N} \quad \text{para toda } \underline{x} \in \mathcal{X}_N \tag{41}$$

se obtiene

$$P_e \leq 2^{-NE_r(R)} \tag{42}$$

donde

$$E_r(R) = \max_{0 \leq \rho \leq 1} \{ E_0(\rho) - \rho R \} \tag{43}$$

Para $R < 1 - \frac{\sqrt{\epsilon}}{\sqrt{\epsilon} + \sqrt{1-\epsilon}}$, $\rho = 1$ maximiza el exponente. Para mayores velocidades, $E_r(R)$

puede expresarse paramétricamente como

$$E_r(R) = T_\varepsilon(\delta) - \mathcal{H}(\delta) \quad (44)$$

y

$$R = 1 - \mathcal{H}(\delta)$$

donde δ varía de ε a $\frac{\sqrt{\varepsilon}}{\sqrt{\varepsilon} + \sqrt{1-\varepsilon}}$. En la Figura 12 se puede observar que para $R = 1 - \mathcal{H}(\delta) < C = 1 - \mathcal{H}(\varepsilon)$, $E_r(R) > 0$.

Como P_e es la probabilidad de error promedio, debe existir al menos un código con probabilidad de error menor a la promedio, es decir, un código $\mathcal{C} = \{x_1, x_2, \dots, x_M\}$ con probabilidad de decodificación errónea de bloque que satisface

$$P_e \leq 2^{-NE_r(R)} \quad (45)$$

donde $E_r(R) > 0$ para $R < C$. Por lo tanto, para velocidades menores a la capacidad puede lograrse una probabilidad de error tan pequeña como se desee.

Límite de empaquetamiento de esfera

Los límites aleatorios de codificación muestran que para $R < C$ la probabilidad de decodificación errónea de bloque puede hacerse decrecer exponencialmente en la longitud del bloque con el exponente $E_r(R)$.

Se define el parámetro δ que satisface $R = 1 - \mathcal{H}(\delta)$, y para todas las constantes pequeñas $\gamma_1 > 0$ se considera un BSC “dummy” con parámetro $\tilde{\delta} = \delta + \gamma_1$. Este canal tienen capacidad $\tilde{C} = 1 - \mathcal{H}(\delta) < R = 1 - \mathcal{H}(\tilde{\delta})$. Para este canal “dummy” para cualquier código \mathcal{C} de velocidad R y longitud de bloque N , del límite de codificación converso, el límite inferior en la probabilidad de error es

$$\tilde{P}_e \geq 1 - 2^{-N\tilde{E}_c(R)} \quad (46)$$

donde $\tilde{E}_c(R) > 0$ para $R > \tilde{C}$.

Considerando un canal BSC con parámetro ε donde $R < C = 1 - \mathcal{H}(\varepsilon)$, con un código $\mathcal{C} = \{x_1, x_2, \dots, x_M\}$ de velocidad R , y función

$$\phi_m(\underline{y}) = \begin{cases} 0 & P_N(\underline{y} | \underline{x}_m) > \tilde{P}_N(\underline{y} | \underline{x}_m) 2^{-N[T_\varepsilon(\delta) - \mathcal{H}(\delta) + \gamma_2]} \\ 1 & \text{en otro caso} \end{cases} \quad (47)$$

donde $\tilde{P}_N(\underline{y} | \underline{x})$ y $P_N(\underline{y} | \underline{x})$ son las probabilidades de transición para el canal BSC “dummy” y el BSC actual, respectivamente, y γ_2 es una constante. Entonces, la probabilidad de error del BSC actual está limitada por

$$\begin{aligned} P_e &= \frac{1}{M} \sum_{m=1}^M \sum_{\underline{y}} (1 - \delta_m(\underline{y})) P_N(\underline{y} | \underline{x}_m) \\ &\geq \frac{1}{M} \sum_{m=1}^M \sum_{\underline{y}} (1 - \delta_m(\underline{y})) (1 - \phi_m(\underline{y})) P_N(\underline{y} | \underline{x}_m) \\ &\geq \frac{1}{M} \sum_{m=1}^M \sum_{\underline{y}} (1 - \delta_m(\underline{y})) (1 - \phi_m(\underline{y})) \times \tilde{P}_N(\underline{y} | \underline{x}_m) 2^{-N[T_\varepsilon(\delta) - \mathcal{H}(\delta) + \gamma_2]} \\ &= 2^{-N[T_\varepsilon(\delta) - \mathcal{H}(\delta) + \gamma_2]} \left\{ \frac{1}{M} \sum_{m=1}^M (1 - \delta_m(\underline{y})) \tilde{P}_N(\underline{y} | \underline{x}_m) - \frac{1}{M} \sum_{m=1}^M \sum_{\underline{y}} \phi(\underline{y}) \tilde{P}_N(\underline{y} | \underline{x}_m) \right\} \\ &= 2^{-N[T_\varepsilon(\delta) - \mathcal{H}(\delta) + \gamma_2]} \left\{ \tilde{P}_e - \tilde{\text{Pr}} \left[\frac{1}{N} \log \frac{P_N(\underline{y} | \underline{x}_m)}{P_N(\underline{y} | \underline{x}_m)} \geq T_\varepsilon(\delta) - \mathcal{H}(\delta) + \gamma_2 \right] \right\} \end{aligned} \quad (48)$$

donde

$$\begin{aligned}
\sum_{\underline{y}} \phi(\underline{y}) \tilde{P}_N(\underline{y} | \underline{x}) &= \tilde{\text{Pr}} \left\{ \frac{1}{N} \log \frac{\tilde{P}_N(\underline{y} | \underline{x})}{P_N(\underline{y} | \underline{x})} \geq T_\varepsilon(\delta) - \mathcal{H}(\delta) + \gamma_2 \right\} \\
&= \tilde{\text{Pr}} \left\{ \frac{1}{N} \sum_{n=1}^N \log \frac{\tilde{P}_N(\underline{y}_n | \underline{x}_n)}{P_N(\underline{y}_n | \underline{x}_n)} \geq T_\varepsilon(\delta) - \mathcal{H}(\delta) + \gamma_2 \right\}
\end{aligned} \tag{49}$$

para $\gamma_1 > 0$ suficientemente pequeñas se tiene

$$\begin{aligned}
\sum_{\underline{y}} \tilde{P}(\underline{y} | \underline{x}) \log \frac{\tilde{P}(\underline{y} | \underline{x})}{P(\underline{y} | \underline{x})} &= \delta \log \frac{\delta}{\varepsilon} + (1 - \delta) \log \frac{1 - \delta}{1 - \varepsilon} \\
&= T_\varepsilon(\delta) - \mathcal{H}(\delta) \\
&= T_\varepsilon(\delta + \gamma_1) - \mathcal{H}(\delta + \gamma_1) \\
&< T_\varepsilon(\delta) - \mathcal{H}(\delta) + \gamma_2
\end{aligned} \tag{50}$$

por la ley de los grandes números en forma débil se obtiene

$$\tilde{\text{Pr}} \left\{ \frac{1}{N} \sum_{n=1}^N \log \frac{\tilde{P}_N(\underline{y}_n | \underline{x}_n)}{P_N(\underline{y}_n | \underline{x}_n)} \geq T_\varepsilon(\delta) - \mathcal{H}(\delta) + \gamma_2 \right\} \xrightarrow{N \rightarrow \infty} 0 \tag{51}$$

Finalmente, para cualquier código \mathcal{C} de velocidad $R = 1 - \mathcal{H}(\delta) < C = 1 - \mathcal{H}(\varepsilon)$

$$P_e \geq \left[\left(1 - 2^{-NE_c(R)} - 0\left(\frac{1}{N}\right) \right) \right] 2^{-N[E_p(R) + \gamma_2]} \tag{52}$$

donde $\tilde{E}_c(R) > 0$ ya que $R > 1 - \mathcal{H}(\delta + \gamma_1)$, $0\left(\frac{1}{N}\right) \xrightarrow{N \rightarrow \infty} 0$ y

$$E_{sp}(R) = T_\varepsilon(\delta) - \mathcal{H}(\delta) \quad (53)$$

$$R = 1 - \mathcal{H}(\delta)$$

donde δ varía de ε a 1.

Como $\gamma_1 > 0$ y $\gamma_2 > 0$ puede hacerse pequeñas para códigos de velocidad R , el límite inferior decrece exponencialmente en N con exponente $E_{sp}(R)$. Donde

$$E_{sp}(R) = E_r(R) \quad (54)$$

para el rango de velocidades

$$1 - \mathcal{H}\left(\frac{\sqrt{\varepsilon}}{\sqrt{\varepsilon} + \sqrt{1-\varepsilon}}\right) \leq R \leq 1 - \mathcal{H}(\varepsilon) \quad (55)$$

Entonces, para $R < 1 - \mathcal{H}\left(\frac{\sqrt{\varepsilon}}{\sqrt{\varepsilon} + \sqrt{1-\varepsilon}}\right)$, $E_{sp}(R) > E_r(R)$.

Límite Expurgado

A velocidades menores que $1 - \mathcal{H}\left(\frac{\sqrt{\varepsilon}}{\sqrt{\varepsilon} + \sqrt{1-\varepsilon}}\right)$, existe la posibilidad de derivar un límite aún más estrecho que el límite de codificación aleatorio. Entonces, para un código lineal de la forma

$$\underline{x}_m = u_{m1} \underline{b}_{-1} \oplus u_{m2} \underline{b}_{-2} \oplus \dots \oplus u_{mK} \underline{b}_{-K} \quad m = 1, 2, \dots, M = 2^K \quad (56)$$

donde K bits de información u_m forman los coeficientes binarios básicos para los vectores de N dimensiones $\underline{b}_{-1}, \underline{b}_{-2}, \dots, \underline{b}_{-K}$. Si $u_{-1} = 0$ entonces $\underline{x}_{-1} = 0$.

El peso de las palabras código se define como

$$\omega_m = \omega(\underline{x}_m, \underline{x}_{-1}) \quad m = 2, 3, 4, \dots, M \quad (57)$$

Seleccionando aleatoriamente un código lineal escogiendo la base $\underline{b}_{-1}, \underline{b}_{-2}, \dots, \underline{b}_{-K}$ de acuerdo a

$$Q(\underline{b}_{-1}, \underline{b}_{-2}, \dots, \underline{b}_{-K}) = \prod_{k=1}^K Q_N(\underline{b}_{-k}) = \left(\frac{1}{2^N}\right)^K \quad (58)$$

se tiene

$$\Pr\{\omega_m = k\} = \binom{N}{k} \left(\frac{1}{2}\right)^N \quad m = 2, 3, \dots, M \quad (59)$$

Suponiendo el código lineal $\mathcal{C} = \{\underline{x}_{-1}, \underline{x}_{-2}, \dots, \underline{x}_{-M}\}$, como $\underline{x}_m \oplus \underline{x}_{m'}$ es una palabra código siempre que \underline{x}_m y $\underline{x}_{m'}$ sean palabras código, la distancia de Hamming de \underline{x}_m a otra palabra código es la misma que la de \underline{x}_1 a otra palabra código, por lo que la probabilidad condicional de error es la misma para todas las palabras código

$$\begin{aligned} P_e &= P_{e,m} \\ &= \sum_{\underline{y}} \left(1 - \delta_1(\underline{y})\right) P_N(\underline{y} | \underline{x}_{-1}) \quad m = 1, 2, \dots, M \end{aligned} \quad (60)$$

Por conveniencia,

$$P_e = \sum_{\underline{y}} \left(1 - \delta_1(\underline{y})\right) P_N(\underline{y} | \underline{x}_{-1}) \quad (61)$$

como $\sum_{m=1}^M \delta_m(\underline{y}) = 1$ se tiene

$$P_e = \sum_{m=2}^M \sum_{\underline{y}} \left(\delta_m(\underline{y}) \right) P_N(\underline{y} | \underline{x}_1) \quad (62)$$

Como la regla de decisión de máxima semejanza satisface

$$\delta_m(\underline{y}) \leq \sqrt{\frac{P_N(\underline{y} | \underline{x}_m)}{P_N(\underline{y} | \underline{x}_1)}} \quad (63)$$

entonces

$$\begin{aligned} P_e &\leq \sum_{m=2}^M \sum_{\underline{y}} \sqrt{P_N(\underline{y} | \underline{x}_m) P_N(\underline{y} | \underline{x}_1)} \\ &= \sum_{m=2}^M \prod_{n=1}^N \sum_{\underline{y}} \sqrt{P_N(\underline{y} | \underline{x}_{mn}) P_N(\underline{y} | \underline{x}_{1n})} \\ &= \sum_{m=2}^M \left[\sqrt{4\varepsilon(1-\varepsilon)} \right]^{\rho(\underline{x}_m, \underline{x}_1)} \\ &= \sum_{m=2}^M \left[\sqrt{4\varepsilon(1-\varepsilon)} \right]^{\omega_m} \end{aligned} \quad (64)$$

Para cualquier $0 \leq \alpha \leq 1$, $\left(\sum_m a_m \right)^\alpha \leq \sum_m a_m^\alpha$ para toda $a_m \geq 0$. Por lo que

$$P_e^\alpha \leq \sum_{m=2}^M \left(\sqrt{4\varepsilon(1-\varepsilon)} \right)^{\alpha \omega_m} \quad (65)$$

Promediando lo anterior sobre el conjunto de los códigos lineales se obtiene

$$\begin{aligned}
\overline{P_e^\alpha} &\leq \sum_{m=2}^M \left(\sqrt{4\varepsilon(1-\varepsilon)} \right)^{\alpha \omega_m} \\
&= (M-1) \sum_{k=0}^N \binom{N}{k} \left(\frac{1}{2} \right)^N \left(\sqrt{4\varepsilon(1-\varepsilon)} \right)^{\alpha k} \\
&\leq M \left[\frac{1 + \left(\sqrt{4\varepsilon(1-\varepsilon)} \right)^\alpha}{2} \right]^N \\
&= 2^{-N \left\{ (1-R) - \log \left[1 + \left(\sqrt{4\varepsilon(1-\varepsilon)} \right)^\alpha \right] \right\}}
\end{aligned} \tag{66}$$

Entonces existe un código lineal tal que $P_e^\alpha \leq \overline{P_e^\alpha}$ ó $P_e \leq \left(\overline{P_e^\alpha} \right)^{1/\alpha}$, y si $\rho = \frac{1}{\alpha} \geq 1$

$$P_e \leq 2^{-N \left\{ \rho(1-R) - \rho \log \left[1 + \left(\sqrt{4\varepsilon(1-\varepsilon)} \right)^\rho \right] \right\}} \tag{67}$$

Definiendo

$$D = \frac{\left(\sqrt{4\varepsilon(1-\varepsilon)} \right)^{1/\rho}}{1 + \left(\sqrt{4\varepsilon(1-\varepsilon)} \right)^{1/\rho}} \tag{68}$$

se obtiene que existe un código lineal \mathcal{C} tal que

$$P_e \leq 2^{-NE_{ex}(R)} \tag{69}$$

donde

$$E_{ex}(R) = \sup_{\rho \geq 1} \left\{ \rho(1-R) - \rho \log \left[1 + \left(\sqrt{4\varepsilon(1-\varepsilon)} \right)^\rho \right] \right\} \tag{70}$$

que se expresa paramétricamente como

$$E_{ex}(R) = -D \sqrt{4\varepsilon(1-\varepsilon)} \tag{71}$$

$$R = 1 - \mathcal{H}(D)$$

$$\text{y } D \text{ varía de } \frac{\sqrt{4\varepsilon(1-\varepsilon)}}{[1 + \sqrt{4\varepsilon(1-\varepsilon)}]} \text{ a } \frac{1}{2}.$$

Entonces, para bajas velocidades $0 \leq R \leq 1 - \mathcal{H}\left(\frac{\sqrt{4\varepsilon(1-\varepsilon)}}{[1 + \sqrt{4\varepsilon(1-\varepsilon)}]}\right)$, el límite expurgado produce el límite superior más estrecho conocido en la probabilidad de error.

Límites para códigos específicos

Para el código lineal $\mathcal{C} = \{x_1, x_2, \dots, x_M\}$, donde $\omega_m = \omega(x_m, x_1)$ para $m = 1, 2, \dots, M$ tiene un límite definido por la ecuación 64. Definiendo

$$A_k = \# \text{ de palabras código de peso } \omega_m = k \quad (72)$$

la ecuación 64 se convierte en

$$P_e \leq \sum_{k=d_{\min}}^N A_k (\sqrt{4\varepsilon(1-\varepsilon)})^k \quad (73)$$

donde $d_{\min} = \min_{m \neq 1} \{\omega_m\}$ es el peso mínimo del código o distancia mínima de Hamming.

Definiendo el polinomio

$$A(z) = \sum_{k=d_{\min}}^N A_k z^k \quad (74)$$

este límite puede escribirse como

$$P_e \leq A(z)_{z=\sqrt{4\epsilon(1-\epsilon)}} \quad (75)$$

Para evaluar este límite para un código lineal se requiere determinar la estructura de peso del código. Para valores pequeños de K esto es posible mediante enumeración exhaustiva.

Con el fin de derivar un límite inferior para este código lineal se toma una $x \in \mathcal{C}$ tal que

$$d_{min} = \omega(\hat{x}, x_1) \quad (76)$$

Suponiendo que las palabras código x_1 y \hat{x} son las únicas palabras código posible con igual probabilidad de transmisión sobre el BSC y con una función de decisión para dos palabras código definida por

$$\delta(y) = \begin{cases} 1 & \text{si } P_N(y|\hat{x}) > P_N(y|x_1) \\ 0 & \text{si } P_N(y|\hat{x}) \leq P_N(y|x_1) \end{cases} \quad (77)$$

Se tiene que las dos probabilidades de error son

$$P_e(x_1, \hat{x}) = \sum_y \delta(y) P_N(y|x_1) \quad (78)$$

y

$$P_e(x_1, \hat{x}) = \sum_y (1 - \delta(y)) P_N(y|\hat{x}) \quad (79)$$

Para códigos lineales la probabilidad de error promedio con dos palabras código es menor que la probabilidad de error promedio con M palabras código posibles, por lo que

$$P_e = P_{e,1} \geq \frac{1}{2} P_e(\underline{x}, \underline{\hat{x}}) + \frac{1}{2} P_e(\underline{x}_1, \underline{\hat{x}}) \quad (80)$$

ó

$$\begin{aligned} P_e &\geq \frac{1}{2} \sum_{\underline{y}} \left[\delta(\underline{y}) P_N(\underline{y} | \underline{x}_1) + (1 - \delta(\underline{y})) P_N(\underline{y} | \underline{x}) \right] \\ &= \frac{1}{2} \sum_{\underline{y}} \sqrt{P_N(\underline{y} | \underline{\hat{x}}) P_N(\underline{y} | \underline{x}_1)} \left[\delta(\underline{y}) \sqrt{\frac{P_N(\underline{y} | \underline{x}_1)}{P_N(\underline{y} | \underline{\hat{x}})}} + (1 - \delta(\underline{y})) \sqrt{\frac{P_N(\underline{y} | \underline{\hat{x}})}{P_N(\underline{y} | \underline{x})}} \right] \\ &= \frac{1}{2} 2^{d_{\min} \log \sqrt{4\varepsilon(1-\varepsilon)}} \times \sum_{\underline{y}} Q(\underline{y}) \left[\delta(\underline{y}) 2^{\frac{1}{2}D(\underline{y})} + (1 - \delta(\underline{y})) 2^{-\frac{1}{2}D(\underline{y})} \right] \end{aligned} \quad (81)$$

definiendo

$$\begin{aligned} Q(\underline{y}) &= \frac{\sqrt{P_N(\underline{y} | \underline{\hat{x}}) P_N(\underline{y} | \underline{x}_1)}}{\sum_{\underline{y}} \sqrt{P_N(\underline{y} | \underline{\hat{x}}) P_N(\underline{y} | \underline{x}_1)}} \\ &= \frac{\sqrt{P_N(\underline{y} | \underline{\hat{x}}) P_N(\underline{y} | \underline{x}_1)}}{2^{d_{\min} \log \sqrt{4\varepsilon(1-\varepsilon)}}} \end{aligned} \quad (82)$$

y

$$D(\underline{y}) = \log \frac{P_N(\underline{y} | \underline{x}_1)}{P_N(\underline{y} | \underline{x})} \quad (83)$$

Por simetría

$$\sum_{\underline{y}} D(\underline{y}) Q(\underline{y}) = 0 \quad (84)$$

obteniendo el límite de varianza

$$\sum_{\underline{y}} D^2(\underline{y}) Q(\underline{y}) \leq N\sigma^2 \quad (85)$$

donde

$$\sigma^2 = \left(2 \log \frac{1-\varepsilon}{\varepsilon} \right)^2 \quad (86)$$

Definiendo la región

$$Y(\sigma^2) = \left\{ \underline{y} : \left| D(\underline{y}) \right| \leq \sqrt{2N\sigma^2} \right\} \quad (87)$$

se obtiene la desigualdad de Chebyshev

$$\sum_{\underline{y} \in Y(\sigma^2)} Q(\underline{y}) \geq \frac{1}{2} \quad (88)$$

Lo que proporciona el límite

$$\begin{aligned}
P_e &\geq \frac{1}{2} 2^{d_{min} \log \sqrt{4\varepsilon(1-\varepsilon)}} \times \sum_{y \in \mathcal{Y}(\sigma^2)} \mathcal{Q}\left(\frac{y}{\sigma}\right) \left[\delta\left(\frac{y}{\sigma}\right) 2^{2^{d(y)}} + \left(1 - \delta\left(\frac{y}{\sigma}\right)\right) 2^{-2^{d(y)}} \right] \\
&\geq \frac{1}{2} 2^{d_{min} \log \sqrt{4\varepsilon(1-\varepsilon)}} 2^{-\frac{1}{2}\sqrt{2N\sigma^2}} \sum_{y \in \mathcal{Y}(\sigma^2)} \mathcal{Q}(y) \\
&\geq \frac{1}{4} 2^{d_{min} \log \sqrt{4\varepsilon(1-\varepsilon)}} 2^{-\frac{1}{2}\sqrt{2N\sigma^2}}
\end{aligned} \tag{89}$$

Entonces, para cualquier código \mathcal{C} con distancia mínima d_{min} puede evaluarse un límite inferior como el anterior. Sin embargo, por lo general se expresa en términos de la distancia mínima normalizada

$$\hat{d}_{min} = \frac{d_{min}}{N} \tag{90}$$

ya que generalmente d_{min} crece linealmente con N . Entonces se obtiene el límite inferior

$$P_e \geq \frac{1}{4} 2^{-\frac{1}{2}\sqrt{2N\sigma^2}} 2^{Nd_{min} \log \sqrt{4\varepsilon(1-\varepsilon)}} \tag{91}$$

Hasta ahora se mostró que para cualquier código \mathcal{C} de velocidad R y longitud de bloque N se tiene la probabilidad de decodificación correcta limitada por

$$P_c \geq 2^{-NE_c(R)} \quad R > C \tag{92}$$

y la probabilidad de error está limitada por

$$P_e \geq \left[1 - 0 \binom{1}{N} \right] 2^{-N[E_{sp}(R) + \gamma_2]} \quad R < C \tag{93}$$

donde γ_2 es un número positivo y $0\left(\frac{1}{N}\right) \xrightarrow{N \rightarrow \infty} 0$.

Se demostró que existen códigos tal que

$$P_e \leq 2^{-NE_r(R)} \quad R < C \quad (94)$$

y

$$P_e \leq 2^{-NE_{er}(R)} \quad R < C \quad (95)$$

Para código lineales (se generaliza fácilmente) con la estructura de peso $\{A_k\}_{k=d_{min}}^N$ se tiene

$$\frac{1}{4} 2^{-\frac{1}{2}\sqrt{2N\sigma^2}} z^{d_{min}} \leq P_e \leq \sum_{k=d_{min}}^N A_k z^k \quad (96)$$

donde $z = \sqrt{4\varepsilon(1-\varepsilon)}$. Para BSC con parámetro ε y capacidad $C = 1 - \mathcal{H}(\delta)$ los exponentes empleados son

$$E_C(R) = T_\varepsilon(\delta) - \mathcal{H}(\delta) > 0$$

para

$$C < R = 1 - \mathcal{H}(\delta) \leq 1$$

$$E_r(R) = \begin{cases} 1 - R - \log_2 [1 + \sqrt{4\varepsilon(1-\varepsilon)}] & 0 \leq R \leq \mathcal{H}\left(\frac{\sqrt{\varepsilon}}{\sqrt{\varepsilon} + \sqrt{1-\varepsilon}}\right) \\ E_{sp}(R) & 1 - \mathcal{H}\left(\frac{\sqrt{\varepsilon}}{\sqrt{\varepsilon} + \sqrt{1-\varepsilon}}\right) < R \leq C \end{cases}$$

(97)

y

$$E_{sp}(R) = T_\varepsilon(\delta) - \mathcal{H}(\delta) \quad (98)$$

para

$$0 \leq R = 1 - \mathcal{H}(\delta) < C$$

y

$$E_{ex}(R) = -D \log \sqrt{4\varepsilon(1-\varepsilon)} > 0$$

para

$$0 < R = 1 - \mathcal{H}(D) < 1 - \mathcal{H}\left(\frac{\sqrt{4\varepsilon(1-\varepsilon)}}{1 + \sqrt{4\varepsilon(1-\varepsilon)}}\right)$$

Al combinar la ecuación 95 con la ecuación 96, se obtiene que existe un código lineal de distancia mínima d_{min} tal que

$$\frac{1}{4} 2^{-\frac{1}{2}\sqrt{2N\sigma}} \left(\sqrt{4\varepsilon(1-\varepsilon)}\right)^{d_{min}} \leq P_e \leq \left(\sqrt{4\varepsilon(1-\varepsilon)}\right)^{DN} \quad (99)$$

Lo que muestra que existe una secuencia de códigos lineales de velocidad fija R tal que

$$\lim_{N \rightarrow \infty} \frac{d_{min}}{N} \geq D \quad (100)$$

donde

$$R = 1 - \mathcal{H}(D)$$

Conocido como límite de Gilbert en la distancia mínima normalizada del código, el cual es un límite muy cerrado, es decir, con longitud N de bloque incrementando, la distancia mínima máxima posible se aproxima al límite de Gilbert D donde $R = 1 - \varepsilon(D)$.

La generalización para canales discretos sin memoria con probabilidades de transición $\{P(y|x): x \in \mathcal{X}, y \in \mathcal{Y}\}$ es como se muestra a continuación

$$C = \max_Q \sum_y \sum_x P(y|x) Q(x) \log \frac{P(y|x)}{\sum_{x'} P(y|x') Q(x')}$$

$$E_c(R) = \max_{-1 \leq \rho \leq 0} \min_Q [E_0(\rho, Q) - \rho R]$$

$$E_r(R) = \max_{0 \leq \rho \leq 1} \min_Q [E_0(\rho, Q) - \rho R]$$

$$E_{sp}(R) = \sup_{\rho \geq 0} \max_Q [E_0(\rho, Q) - \rho R]$$

$$E_{ex}(R) = \sup_{\rho \geq 1} \max_Q [E_x(\rho, Q) - \rho R] \quad (101)$$

donde

$$E_0(\rho, Q) = -\log \sum_y \left[\sum_x Q(x) P(y|x) \right]^{1+\rho}$$

y

$$E_x(\rho, Q) = -\rho \log \left\{ \sum_x \sum_{x'} Q(x) Q(x') \left[\sum_y \sqrt{P(y|x) P(y|x')} \right]^\rho \right\}$$

Definiendo la distancia de Bhattacharyya entre $x, x' \in \mathcal{X}$ como

$$d(x, x') = -\log \sum_y \sqrt{P(y|x) P(y|x')} \quad (102)$$

se generalizan los límites para códigos específicos de ecuación 96, así como el límite de Gilbert de la ecuación 100. En la Figura 13 se muestran varios exponentes para un canal discreto sin memoria.

También existe un límite inferior de línea recta de P_e que produce un exponente que conecta a $E_{ex}(0)$ a un punto tangente en la curva $E_{sp}(R)$ con una línea recta. De la ecuación 91 y como siempre cumple con $d_{min} \leq 1/2$, se obtiene

$$P_e \geq \frac{1}{4} 2^{-\frac{1}{2} \sqrt{2N\sigma}} 2^{-NE_{ex}(0)} \quad (103)$$

Lo que muestra que el límite expurgado está muy cerrado en $R = 0$.

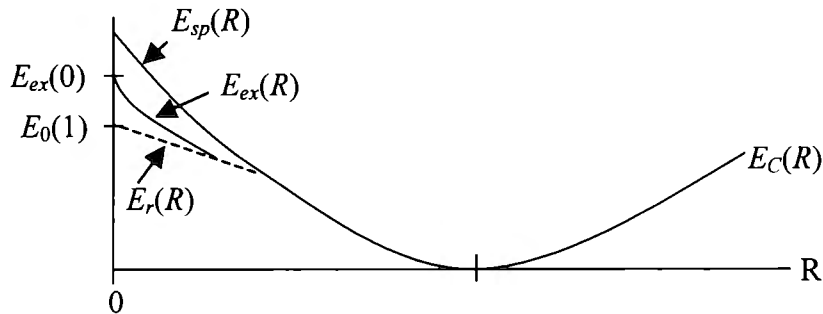


Figura 13. Exponentes

2.2.1.2 Códigos Convolucionales

Utilizando el sistema de codificación canal de la Figura 11, y suponiendo que la velocidad es

$$R = \frac{K}{N} = \frac{1}{n} \quad (104)$$

donde n es un entero, cualquier palabra código $x \in \mathcal{X}^n$ puede agruparse en subsecuencias de n bits como se muestra en la ecuación 105

$$\underline{x} = (\underline{x}^1, \underline{x}^2, \dots, \underline{x}^K) \quad (105)$$

donde $x^{k'} \in \mathcal{X}_n$ para $k = 1, 2, \dots, K$. En el análisis de códigos de bloque se asumió que cualquier componente de la palabra código puede depender de cualquiera de los K bits de información, en cambio, para los códigos convolucionales se impone la restricción de que \underline{x}^k depende de los bits de información $u_k, u_{k-1}, u_{k-2}, \dots, u_{k-\nu+1}$ donde ν es la restricción de longitud o restricción de memoria del código convolucional.

Formalmente se tiene

$$\underline{x}^k = f(u_k, u_{k-1}, \dots, u_{k-\nu+1}) \quad k = 1, 2, \dots, K \quad (106)$$

donde inicialmente $u_k = 0$ y $k \leq 0$, por lo que los códigos convolucionales son un caso especial de los códigos bloque.

Sin embargo, tanto el codificador óptimo como el análisis de error no dependen de la longitud del bloque N , el cual puede hacerse arbitrariamente larga. Entonces, los parámetros claves en los códigos convolucionales son la velocidad $R = 1/n$ y la longitud de restricción ν .

El codificador convolucional se modela como una máquina de estados finitos con entrada y salida binaria, n bits cada unidad de tiempo. Por ejemplo, definiendo un vector de estado en el tiempo k

$$\underline{s}^k = (u_{k-\nu+2}, u_{k-\nu+3}, \dots, u_{k-1}, u_k) \quad (107)$$

tenemos para el codificador en el estado \underline{s}^{k-1} con entrada u_k los n bits de salida

$$\underline{x}^K = f(\underline{u}_k, \underline{s}^{k-1}) \quad (108)$$

Todos los efectos de las entradas al codificador se incluyen en el estado del codificador, en el ejemplo hay $2^{\nu-1}$ estados.

Un diagrama de estados puede describir la generación de los bits de canal, codificados a partir de los bits de información, sin embargo, mediante un diagrama trellis se obtiene una descripción más completa de todos los estados y sus transiciones. En ambos diagramas los n componentes de palabras código se muestran en las ramas que llevan de un estado a otro.

Por ejemplo, si $R = 1/2$, $\nu = 3$, y se tiene el código convolucional cuyas palabras código están dadas por

$$\underline{x}^K = (u_k \oplus u_{k-1} \oplus u_{k-2}, u_k \oplus u_{k-2}) \quad (109)$$

Entonces, el vector de estado $\underline{s}^{k-1} = (u_{k-1}, u_k)$ tiene cuatro estados distintos. En la Figura 14 se muestra el codificador, el diagrama de estado, y el diagrama trellis para este código convolucional, con los cuatro estados representados por \underline{a} , \underline{b} , \underline{c} , y \underline{d} .

Como puede observarse, el diagrama trellis es una forma conveniente de representar todas las palabras código. Una palabra código consiste de una secuencia de bits de la rama de salida a través del camino en el diagrama trellis. El diagrama trellis también muestra como se relacionan las palabras código y como pueden decodificarse eficientemente dada una secuencia de salida de canal.

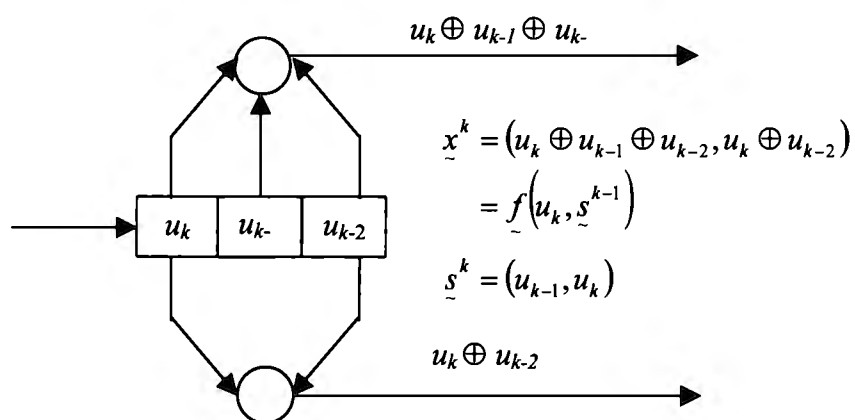


Figura 14.a. Codificador

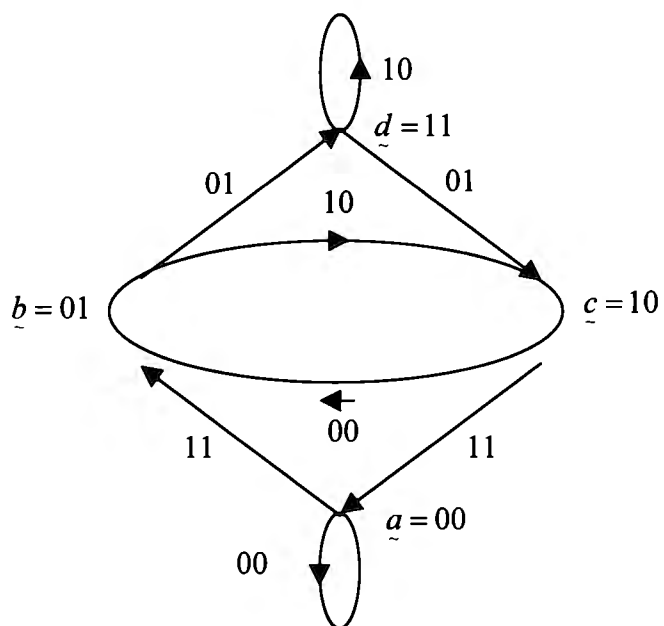


Figura 14.b. Diagrama de estados

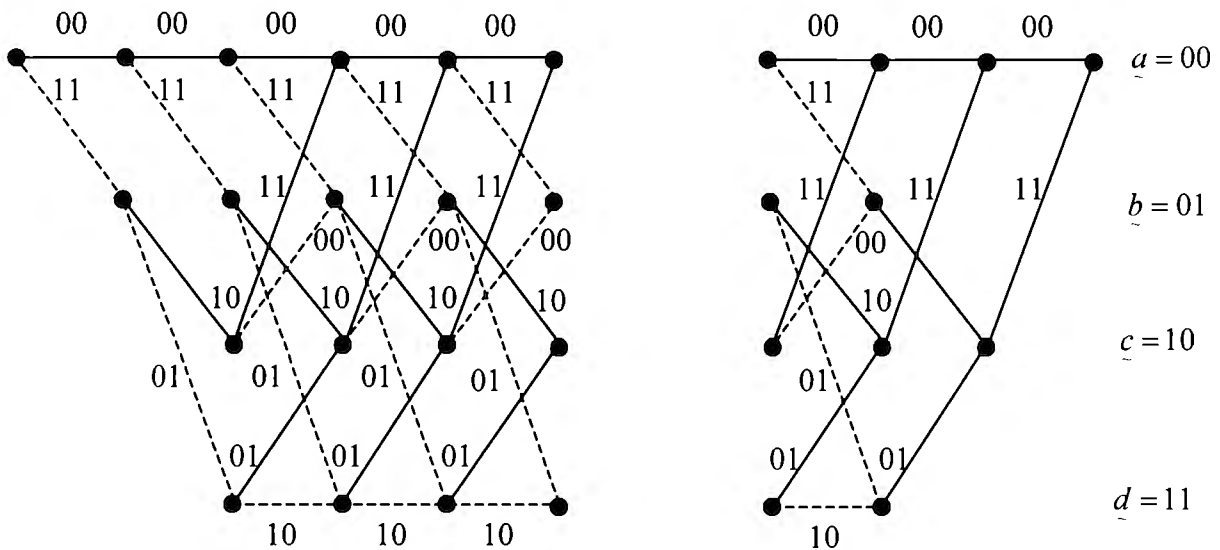


Figura 14.c Diagrama Trellis

2.2.1.3 Decodificación Viterbi

Suponiendo un código convolucional de velocidad $R = 1/n$ y longitud de restricción ν , sobre un BSC con probabilidad de crossover $\varepsilon < 1/2$, K bits de información $u \in \mathbb{Z}_k$ se codifican en $N = nK$ bits de canal (ecuación 108). Si $\underline{u}_1, \underline{u}_2, \dots, \underline{u}_M$ son las $M = 2^K = 2^{RN}$ secuencias de bits de información y $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_M$ son las palabras código correspondientes, entonces el decodificador de máxima semejanza toma la secuencia de salida del canal $y \in \mathbb{Z}_N$ y decide la palabra código que minimiza $\omega(\underline{x}_m, y)$, $m = 1, 2, \dots, M$.

En los códigos convolucionales todas las palabras código corresponden a caminos del diagrama trellis y por lo que el decodificador debe comparar todos los caminos del diagrama trellis con la secuencia de salida del canal $y \in \mathbb{Z}_N$ y escoger el camino cuya rama de secuencia de salida sea más cercana en distancia de Hamming.

La clave de la decodificación de Viterbi [11] consiste en considerar que algunos caminos (palabras código) pueden eliminarse, antes de examinar los N bits del camino.

En la Figura 15 se consideran dos caminos \underline{x} y \underline{x}' , los cuales se unen en cierto estado y permanecen unidos.

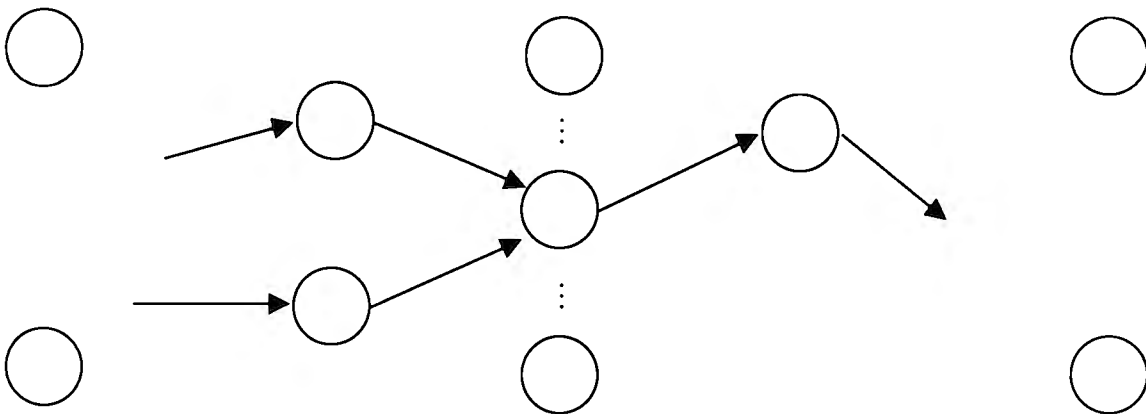


Figura 15. Caminos unidos

Cuando dos caminos se unen la distancia de Hamming de $y \in \mathcal{Y}_N$ es la misma para ambas palabras código por lo que el camino más cercano a $y \in \mathcal{Y}_N$ puede determinarse por las distancias a partir del punto de unión, entonces uno de los caminos puede eliminarse de consideraciones posteriores sin pérdida de óptimidad.

El decodificador de Viterbi mantiene la ruta de todos los caminos iniciales del estado inicial cero y elimina uno de los dos caminos que se unen en un estado. Como existen 2^{v-1} estados en el trellis, el decodificador Viterbi mantiene la ruta de cuando más 2^{v-1} caminos en cada paso del diagrama trellis.

Por ejemplo, tomando el código de la Figura 14, si ocurren $\underline{u} = \underline{0} \in \mathcal{U}_k$ secuencias de información y si se envía la palabra código $\underline{x} = \underline{0} \in \mathcal{X}_k$ sobre el BSC, teniendo que la secuencia de salida del canal es

$$\underline{y} = (00,01,10,00,00,\dots, \text{el resto son ceros}) \quad (110)$$

la operación del decodificador de Viterbi se muestra en la Figura 16, donde se denota la distancia de Hamming de las n salidas del codificador a partir las n salidas correspondientes de canal en cada rama. La flecha indica el camino escogido y el número en cada estado indica la distancia de Hamming total del “camino sobreviviente” hasta ese estado. Al final se selecciona el camino sobreviviente con distancia mínima.

Para cada bit de información el decodificador Viterbi realiza 2^ν cálculos de distancia de Hamming de n secuencias binarias, y $2^{\nu-1}$ comparaciones de pares. Por lo que la complejidad del decodificador Viterbi por bit de información generalmente es proporcional a 2^ν , lo cual es independiente de la longitud del bloque N , que puede hacerse larga.

Para la decodificación de bloque se realizan $m = 2^K$ cálculos de distancia de Hamming de N secuencias para cada K bits de información. Es por ello que en términos de complejidad de decodificadores se puede comparar ν del código convolucional con K del código bloque. Lo cual indica que la complejidad de ambos decodificadores depende exponencialmente en la “memoria” del codificador.

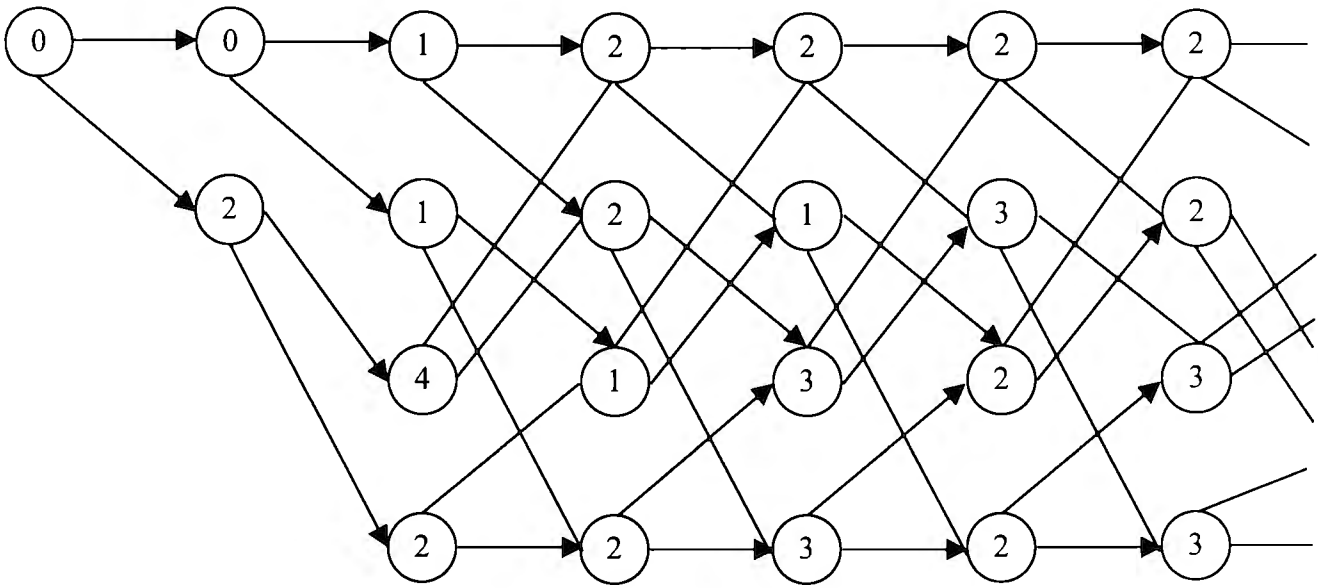


Figura 16. Ejemplo de decodificación de Viterbi

se aproxima a 1 cuando $K \rightarrow \infty$.

Una medida apropiada para el desempeño es la probabilidad de hacer un error de bit

$$P_B = \Pr\{\hat{u} \neq u\} \quad (112)$$

donde u es cualquier bit de información y \hat{u} es el correspondiente bit decodificado.

Para $K \rightarrow \infty$, puede ocurrir un error si y solo si un segmento del camino actualmente transmitido en el diagrama trellis es eliminado por el algoritmo de Viterbi. Por ejemplo, \underline{x} es un segmento de la palabra código transmitida y $\tilde{\underline{x}}$ es otro segmento de camino que diverge de \underline{x} en el tiempo α y se reúne con \underline{x} / ramas después, como se muestra en la Figura 17. \underline{y} será el segmento correspondiente de la secuencia de salida del canal, si

$$\omega(\tilde{\underline{x}}, \underline{y}) < \omega(\underline{x}, \underline{y}) \quad (113)$$

\underline{x} será eliminada cuando se unen los dos segmentos y ocurrirá un error, conocido como error de evento.

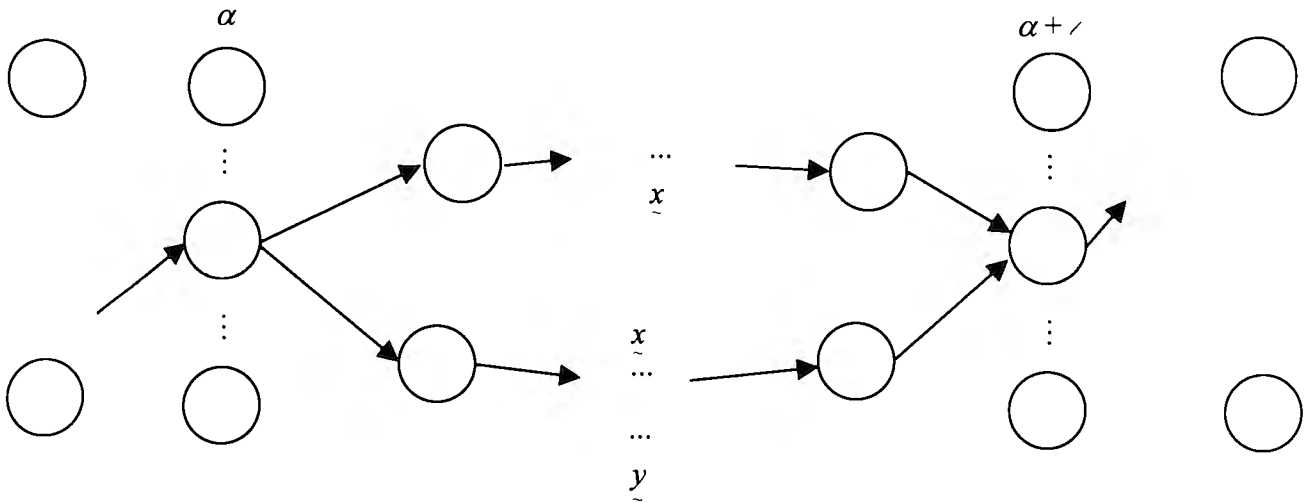


Figura 17. Evento de error

Se requiere que $l \geq \nu$ debido a que los estados de dos segmentos

$$\begin{aligned} \underline{s}^K &= (u_{k-\nu-2}, u_{k-\nu+3}, \dots, u_{k-1}, u_k) \\ \tilde{\underline{s}}^K &= (\tilde{u}_{k-\nu-2}, \tilde{u}_{k-\nu+3}, \dots, \tilde{u}_{k-1}, \tilde{u}_k) \end{aligned} \quad (114)$$

deben igualarse antes de que los caminos diverjan y cuando divergen el último bit en los dos estados es distinto. Para ambos estados toma al menos ν bits de información igualarse de nuevo y que se reúnan sus caminos.

Como los $\nu-1$ bits de información de cualquier segmento de camino incorrecto produce el mismo estado final que el camino correcto en el punto de reunión y como los bits de información iniciales son distintos de los del camino correcto, no más de $l-\nu$ bits de información pueden ser arbitrarios. Si $j = l-\nu$, hay cuando más 2^j caminos que divergen del camino correcto en el tiempo α y se reúnen $l = K + j$ ramas después.

Sea E_j un evento de un camino que diverge del camino correcto en el tiempo α y se reúne $\nu + j$ ramas después causando que el camino correcto sea eliminado. $P_j = \Pr\{E_j\}$, la probabilidad de un evento de error, es esencialmente la probabilidad de error de codificación de bloque con

$$M_j = 2^j \text{ palabras código}$$

$$N_j = n(\nu + j) \text{ longitud de bloque}$$

y velocidad

$$R_j = \frac{\log M_j}{N_j} = \frac{j}{n(\nu + j)} \quad (115)$$

Como 2^j es un límite superior en el número de segmentos de palabras código de longitud $K + j$, puede obtenerse un límite superior en P_j . Considerando códigos conjunto convolucionales donde todos los bits de las ramas de salida en el diagrama de trellis son independientes y equiprobables, puede aplicarse el límite de codificación aleatoria (ecuación 40) sobre $P_j = \Pr\{E_j\}$ y obtener,

$$\begin{aligned} P_j &\leq 2^{-n(\nu+j) \left[E_0(\rho) - \rho \frac{j}{n(\nu+j)} \right]} & 0 \leq \rho \leq 1 \\ &= 2^{-j[nE_0(\rho) - \rho]} 2^{-\nu n E_0(\rho)} \end{aligned} \quad (116)$$

y

$$E_0 = \rho - (1 + \rho) \log \left[\varepsilon^{\frac{1}{1+\rho}} + (1 - \varepsilon)^{\frac{1}{1+\rho}} \right] \quad (117)$$

Los códigos conjunto requeridos para este límite, desafortunadamente, incluyen códigos convolucionales que varían en el tiempo. Hasta ahora no se han encontrado límites para códigos conjunto convolucionales fijos.

Cuando ocurre un evento de error de longitud $\nu + j$, existen cuando más $j+1$ bits de información que pueden decodificarse incorrectamente debido la eliminación del camino correcto. Suponiendo que en cualquier tiempo, u son los bits de información codificador y \hat{u} son los bits decodificados correspondientes, el error de bits decodificados, $\hat{u} \neq u$, puede ser causado por un camino divergente de longitud $\nu + j$ que comenzó $i \leq j$ ramas antes. Sea

$$\begin{aligned} E_{ji} &= \text{evento de error de longitud } \nu + j \text{ comenzando } i \text{ ramas antes} \\ j &= 0, 1, 2, \dots \\ i &= 0, 1, \dots, j \end{aligned} \tag{118}$$

la unión del límite de eventos es

$$\begin{aligned} P_B &= \Pr\{\hat{u} \neq u\} \\ &\leq \sum_{j=0}^{\infty} \sum_{i=0}^j \Pr\{E_{ji}\} \end{aligned} \tag{119}$$

Promediando la ecuación 119 sobre el código conjunto y usando el límite de codificación aleatoria (ecuación 116) se obtiene

$$\begin{aligned} \bar{P}_B &\leq \sum_{j=0}^{\infty} (j+1) \bar{P}_j \\ &= 2^{-nE_0(\rho)} \sum_{j=0}^{\infty} (j+1) 2^{-j[nE_0(\rho)-\rho]} \\ &= \frac{2^{-nE_0(\rho)}}{\left(1 - 2^{-[nE_0(\rho)-\rho]}\right)^2} \end{aligned}$$

y

$$nE_0(\rho) - \rho > 0$$

$$R = \frac{1}{n}$$

$$E_0(\rho) > \rho R \quad (120)$$

En la Figura 12 se observa que mientras $R < C$, siempre puede encontrarse una $0 \leq \rho \leq 1$ que satisfaga la condición de la ecuación 120. Si $\delta > 0$ y definiendo

$$E_{cr}(R) = \sup_{0 \leq \rho \leq 1} E_0(\rho) \quad (121)$$

$$nE_0(\rho) - \rho \geq \delta$$

Entonces existe un código convolucional (posiblemente variante en el tiempo) de velocidad $R = 1/n$ y restricción de longitud ν con probabilidad de error limitada por

$$P_B \leq \frac{2^{-(\nu/R)E_{cr}(R)}}{(1 - 2^{-\delta})^2} \quad (122)$$

En el límite de una δ pequeña, de la Figura 12, se tiene

$$\lim_{\delta \rightarrow 0} E_{cr}(R) = \begin{cases} E_0(1) = 1 - \log[1 + \sqrt{4\varepsilon(1-\varepsilon)}] & R \leq E_0(1) \\ E_0(\rho) \quad \text{donde } E_0(\rho) = \rho R & E_0(1) \leq R < C \end{cases} \quad (123)$$

Para la codificación bloque, el límite de codificación aleatoria para la decodificación incorrecta de una palabra código está dada por la ecuación 42, donde

$$\begin{aligned} P_e &\leq 2^{-NE_c(R)} \\ &= 2^{-(K/R)E_c(R)} \end{aligned} \quad (124)$$

Como varios de los K bits pueden ser erróneos, cuando un bloque se decodifica incorrectamente, la probabilidad de error de bit para códigos bloque está limitada por

$$P_B \leq K 2^{-(K/R)E_r(R)} \quad (125)$$

Como los codificadores (bloque y convolucional) tienen la misma memoria $\nu = K$, los límites aleatorios dependen de $E_r(R)$ para códigos bloque y de $E_{cr}(R)$ para códigos convolucionales. En la Figura 18 se presentan los exponentes de error de codificación aleatoria, donde puede observarse que los códigos convolucionales pueden producir probabilidades de error menores que los códigos bloque, para la misma complejidad de codificación $\nu = K$ del decodificador óptimo.

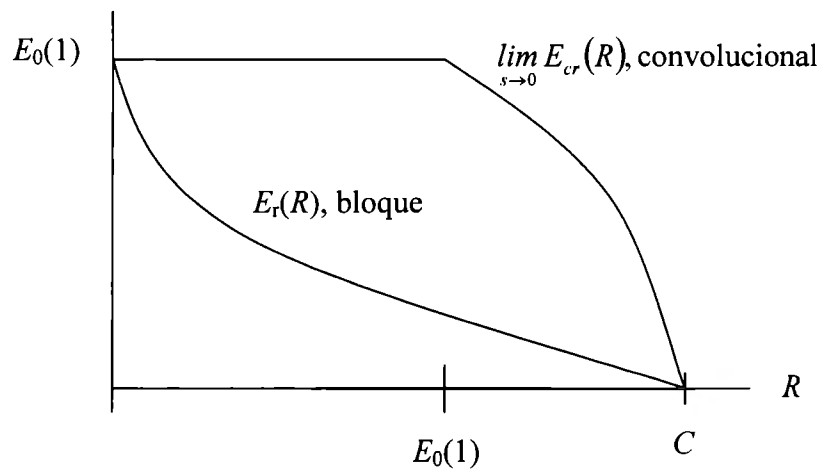


Figura 18. Comparación de exponentes aleatorios de codificación

Límite de empaquetamiento de esfera

La probabilidad de error de bit de decodificación es la probabilidad de la unión de todos los eventos de error que pueden causar un error en los bits de información en cuestión. Un límite inferior de P_B , es la probabilidad de cualquier evento de error E_{ji} ,

$$P_B \geq \Pr\{E_{ij}\} \quad i \leq j = 0,1,\dots \quad (126)$$

Pero $\Pr\{E_{ji}\}$ es esencialmente una probabilidad de error de codificación de bloque que puede limitarse inferiormente con el límite inferior de empaquetamiento de esfera (ecuación 52), por lo que

$$P_B \geq \left[1 - 0 \binom{1}{N_j} \right] 2^{-N_j [E_{sp}(R_j) + \gamma_2]} \quad (127)$$

donde

$$N = n(\nu + j)$$

$$R_j = \frac{j}{n(\nu + j)} \quad j = 0, 1, 2, \dots$$

Si $\lambda = \frac{j}{\nu}$ entonces

$$P_B \geq 2^{-\nu n \left[(1+\lambda) E_{sp} \left(R \frac{\lambda}{1+\lambda} \right) + 0 \left(\frac{1}{\nu} \right) \right]} \quad (128)$$

en $0(1/\nu)$ se incluye el término de desvanecimiento γ_2 y la constante multiplicativa. Además, incluyendo en $0(1/\nu)$ las diferencias provocadas por la aproximación de j/ν con un parámetro no negativo continuo $\lambda \geq 0$, para ν largas

$$\lambda = \frac{\rho E'_0(\rho)}{E_0(\rho) - \rho E'_0(\rho)} \quad (129)$$

donde $\rho \geq 0$ satisface las ecuaciones

$$E_{sp} \left(R \frac{\lambda}{1+\lambda} \right) = E_0(\rho) - \rho E'_0(\rho) \quad (130)$$

$$R \frac{\lambda}{1+\lambda} = E_0'(\rho) \quad (131)$$

Entonces para $R < C$ se tiene $E_{sp}\left(R \frac{\lambda}{1+\lambda}\right) > 0$ y $\lambda \geq 0$, obteniendo el exponente

$$\begin{aligned} E_{csp}(R) &= (1+\lambda)E_{sp}\left(R \frac{\lambda}{1+\lambda}\right) \\ &= E_0(\rho) \end{aligned} \quad (132)$$

donde ρ satisface

$$\begin{aligned} R &= \frac{1+\lambda}{\lambda} E_0'(\rho) \\ &= \frac{E_0(\rho)}{\rho} \end{aligned} \quad (133)$$

Al igual que los códigos bloque, el exponente de empaquetamiento de esfera es el mismo que el del exponente de codificación aleatoria para altas velocidades. Esto es,

$$\begin{aligned} E_{csp}(R) &= \lim_{\delta \rightarrow 0} E_\delta(R) \\ E_0(1) &\leq R \leq C \end{aligned} \quad (134)$$

Límite expurgado

A bajas velocidades puede derivarse un límite expurgado

$$P_B \leq \sum_{j=0}^{\infty} \sum_{i=0}^j \Pr\{E_{ji}\} \quad (135)$$

Para parámetros $0 \leq \alpha \leq 1$ se tiene

$$P_B^\alpha \leq \sum_{j=0}^{\infty} \sum_{i=0}^j \left[\Pr\{E_{ij}\} \right]^\alpha \quad (136)$$

Al promediar sobre el conjunto de los códigos lineales se obtiene

$$\overline{P_B^\alpha} \leq \sum_{j=0}^{\infty} \sum_{i=0}^j \overline{\left[\Pr\{E_{ij}\} \right]^\alpha} \quad (137)$$

de la ecuación 66

$$\Pr\{E_{ji}\}^\alpha \leq 2^{-N_j \left\{ (1-R_j) - \log \left[1 + (\sqrt{4\varepsilon(1-\varepsilon)})^\alpha \right] \right\}} \quad (138)$$

donde

$$N_j = n(\nu + j)$$

$$R_j = \frac{j}{n(\nu + j)}$$

Sea $\beta = 1 - \log \left[1 + (\sqrt{4\varepsilon(1-\varepsilon)})^\alpha \right]$ entonces

$$\begin{aligned} \overline{P_B^\alpha} &\leq \sum_{j=0}^{\infty} (j+1) 2^{-n(\nu+j) \left\{ \beta - \frac{1}{n(\nu+j)} \right\}} \\ &= 2^{-n\nu\beta} \sum_{j=0}^{\infty} (j+1) 2^{-j\{n\beta-1\}} \\ &= \frac{2^{-n\nu\beta}}{\left[1 - 2^{-\{n\beta-1\}} \right]^2} \end{aligned} \quad (139)$$

donde se requiere que $n\beta - 1 > 0$. Para $\rho = 1/\alpha \geq 1$ existe un código convolucional de velocidad $R = 1/n$ y restricción de longitud ν tal que

$$P_B \leq \left(\frac{1}{P_B^\rho} \right)^\rho \leq \frac{2^{-n\nu\rho\beta}}{[1 - 2^{-[n\beta-1]}]^{2\rho}} \quad (140)$$

Para $R < E_0(1) = 1 - \log[1 + \sqrt{4\varepsilon(1-\varepsilon)}]$ existe alguna $\rho \geq 1$ que satisfaga

$$R < 1 - \log \left[1 + \left(\sqrt{4\varepsilon(1-\varepsilon)} \right)^{\frac{1}{\rho}} \right] = \beta \quad (141)$$

y se cumple que $n\beta - 1 > 0$. Sea $\delta > 0$ y $\rho \geq 1$ para satisfacer

$$(1 + \delta)R = 1 - \log \left[1 + \left(\sqrt{4\varepsilon(1-\varepsilon)} \right)^{\frac{1}{\rho}} \right] \quad (142)$$

ó

$$\rho = \frac{\log \sqrt{4\varepsilon(1-\varepsilon)}}{\log(2^{1-(1+\delta)R} - 1)} \quad (143)$$

Entonces

$$P_B \leq \frac{2^{-n\nu E_{cer}(R)}}{(1 - 2^{-\delta})^{2\rho}} \quad (144)$$

donde

$$E_{cex}(R) = \frac{(1 + \delta)R \log \sqrt{4\varepsilon(1 - \varepsilon)}}{\log(2^{1-(1+\delta)R} - 1)} \quad 0 \leq R \leq \frac{E_0(1)}{1 + \delta} \quad (145)$$

En la Figura 19 se muestran los exponentes de la codificación aleatoria, empaquetamiento de esfera, y límite expurgado.

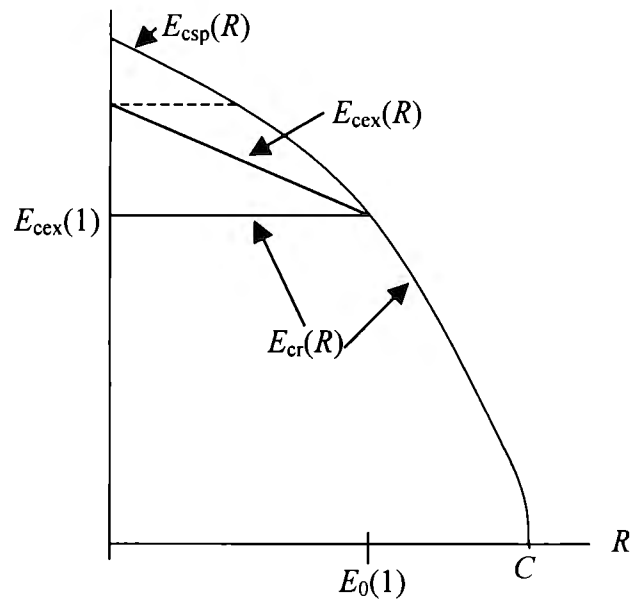


Figura 19. Exponentes convolucionales de codificación

Límites para códigos específicos

Retomando la Figura 17, sea \underline{x} un segmento del camino transmitido y $\underline{\tilde{x}}$ un segmento de otro camino, y suponga

$$\omega(\underline{x}, \underline{\tilde{x}}) = d \quad (146)$$

es decir, \underline{x} y $\underline{\tilde{x}}$ difieren en d bits. Entonces, la probabilidad de que el decodificador Viterbi rechace el segmento \underline{x} en favor del segmento $\underline{\tilde{x}}$ está limitada por

$$\begin{aligned}
P_d &= \Pr \left\{ \omega(\tilde{x}, y) \leq \omega(\underline{x}, y) \mid \underline{x} \text{ es enviada} \right\} \\
&\leq \left(\sqrt{4\varepsilon(1-\varepsilon)} \right)^d
\end{aligned}
\tag{147}$$

el cual es un caso especial de la ecuación 64 con $M=2$.

Suponiendo un código lineal, se define

$$\begin{aligned}
a(n, d) &= \text{número de caminos que inicialmente divergen del} \\
&\quad \text{camino de puros ceros } (\underline{u} = \underline{0}) \text{ que tienen } n \text{ bits "1"} \\
&\quad \text{de información y peso total } d \text{ mientras no están unidos}
\end{aligned}
\tag{148}$$

El código convolucional $a(n, d)$ es esencialmente una estructura de peso del código. Con el BSC se puede asumir que el camino de puros ceros fue la secuencia de camino transmitida.

Definiendo el evento

$$\begin{aligned}
E(n, d) &= \text{un camino de peso } d \text{ con } n \text{ bits "1" de} \\
&\quad \text{información causando un evento de error}
\end{aligned}
\tag{149}$$

Un error de bit puede ser causado por el evento $E(n, d)$ donde el camino divergente comience a divergir en cualquiera de las n ramas anteriores. Tomando la unión del límite de eventos tenemos

$$\begin{aligned}
P_B &\leq \sum_{d=d_f}^{\infty} \sum_{n=1}^{\infty} na(n, d) \Pr\{E(n, d)\} \\
&= \sum_{d=d_f}^{\infty} \sum_{n=1}^{\infty} na(n, d) \left(\sqrt{4\varepsilon(1-\varepsilon)} \right)^d
\end{aligned}
\tag{150}$$

donde d_f es el peso mínimo del camino de puros ceros y se conoce como la distancia libre del código. Este límite es análogo al de la ecuación 75 para códigos de bloque. El código específico determina los parámetros d_f y $\{a(n, d)\}$.

Al igual que para los código bloque, donde se definió un polinomio de pesos del código (ecuación 74), se puede definir una función o polinomio generador para el código convolucional como

$$T(D, I) = \sum_{d=d_1}^{\infty} \sum_{n=1}^{\infty} a(n, d) D^d I^n \quad (151)$$

Entonces

$$P_B \leq \frac{\partial T(D, I)}{\partial I} \Big|_{I=1, D=\sqrt{4\epsilon(1-\epsilon)}} \quad (152)$$

$T(D, I)$ puede encontrarse examinando el diagrama de estados y etiquetando cada rama con $D^{\omega(x)} I^u$, donde $\omega(x)$ es el peso de la n -tupla de la rama de salida $\underline{x} \in \mathcal{X}_N$, y u es el bit de información correspondientes a la rama. La función de transferencia $T(D, I)$ consiste de la suma de todos los pesos de los caminos que dejan el estado cero y regresan. Aquí el peso del camino es el producto de todas las ramas etiquetadas a lo largo del camino.

Retomando la Figura 14, para $R = 1/2$ y $\nu = 3$, se separa el estado $\underline{a} = [00]$ para trazar fácilmente los caminos que dejan y regresan a \underline{a} , tal y como se muestra en la Figura 20.

Sea $T_{\underline{b}}, T_{\underline{d}}, T_{\underline{c}}$ el arreglo de estados de todos los caminos que comienzan en \underline{a} y terminan en sus estados respectivos. De la Figura 20 se tiene

$$\begin{aligned} T_{\underline{b}} &= D^2 I + I T_{\underline{c}} \\ T_{\underline{c}} &= D T_{\underline{b}} + D T_{\underline{d}} \\ T_{\underline{d}} &= D I T_{\underline{b}} + D I T_{\underline{d}} \end{aligned} \quad (153)$$

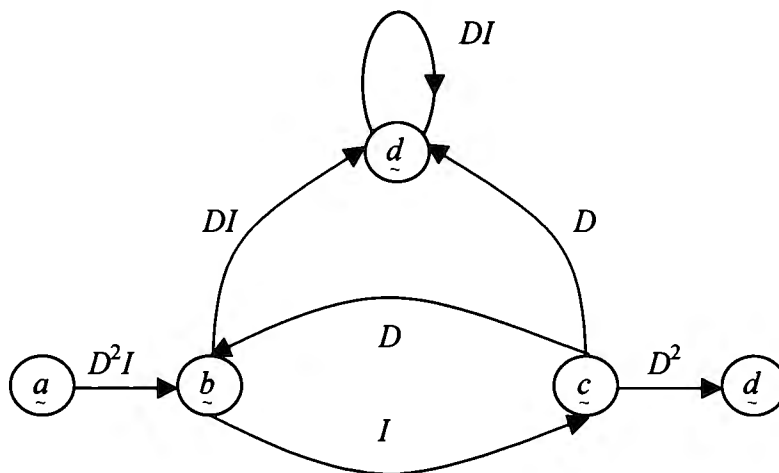


Figura 20. Ejemplo de función de transferencia

Resolviendo para T_ε

$$T_\varepsilon = \frac{D^3 I}{1 - 2DI} \quad (154)$$

y la función de transferencia

$$\begin{aligned} T(D, I) &= D^2 T_\varepsilon \\ &= \frac{D^5 I}{1 - 2DI} \end{aligned} \quad (155)$$

Como

$$\frac{\partial T(D, I)}{\partial I} = \frac{D^5}{(1 - 2DI)^2} \quad (156)$$

para éste código se obtiene

$$P_B \leq \frac{D^5}{(1 - 2DI)^2} \Big|_{D=\sqrt{4\varepsilon(1-\varepsilon)}} \quad (157)$$

y $d_f = 5$.

Un límite inferior de P_B para un código específico puede obtenerse considerando que P_B es la probabilidad de la unión de los eventos que la forman $\{E_{ij}\}$

$$P_B \geq \Pr\{E_{ij}\} \quad (158)$$

De los resultados de codificación de bloque (ecuación 89) se tiene

$$P_B \geq \frac{1}{4} 2^{-\frac{1}{2}\sqrt{2n(\nu+j)\sigma^2}} 2^{d_j \log \sqrt{4\varepsilon(1-\varepsilon)}} \quad (159)$$

donde d_j es el camino de peso mínimo que diverge en $\nu + j$ ramas. En general $d_j \leq \frac{1}{2}n(\nu + j)$ y si $j = 0$ se obtiene

$$P_B \geq \frac{1}{4} 2^{-\frac{1}{2}\sqrt{2n\nu\sigma^2}} 2^{-n\nu E_{\text{cex}}(0)} \quad (160)$$

donde $E_{\text{cex}}(0) = -\frac{1}{2} \log \sqrt{4\varepsilon(1-\varepsilon)}$ es el exponente expurgado de codificación convolucional en $R = 0$. La línea punteada de la Figura 19 representa un límite inferior más estrecho que el límite de empaquetamiento a bajas velocidades.

2.3 TEORÍA DE DISTORSIÓN DE VELOCIDAD

La presentación de la teoría de distorsión de velocidad se limita a fuentes de información sin memoria y discretas con alfabeto $\mathcal{A} = \{a_1, a_2, \dots, a_A\}$ y probabilidades de letras $Q(a_1), Q(a_2), \dots, Q(a_A)$. Cada salida de la fuente es independiente de las otras salidas de la fuente.

El alfabeto usado se denota por $\mathcal{V} = \{b_1, b_2, \dots, b_A\}$ (por lo general $\mathcal{V} = \mathcal{Z}$) y se asume una medida de distorsión no negativa $d(u, v)$ que satisface

$$0 \leq d(u, v) \leq d_0 \quad (161)$$

para toda $(u, v) \in \mathcal{Z} \times \mathcal{V}$.

Generalmente la fuente de información está separada del usuario por un canal de comunicación ruidoso por lo que es necesario codificar las salidas de la fuente y enviar símbolos codificados a través del canal.

Los símbolos de salida del canal se decodifican y cierta representación de las salidas de la fuente se presentan al usuario. En la Figura 21 se muestran los procesos de codificación y decodificación fuente y canal. Si la velocidad de bit fuera del codificador fuente es menor que la capacidad del canal se puede asumir que el codificador canal, el canal, y el decodificador canal son libres de errores y representables como un canal sin ruido, como se muestra en la Figura 22, limitando que la velocidad de bit de la salida del decodificador fuente sea menor a la capacidad del canal.

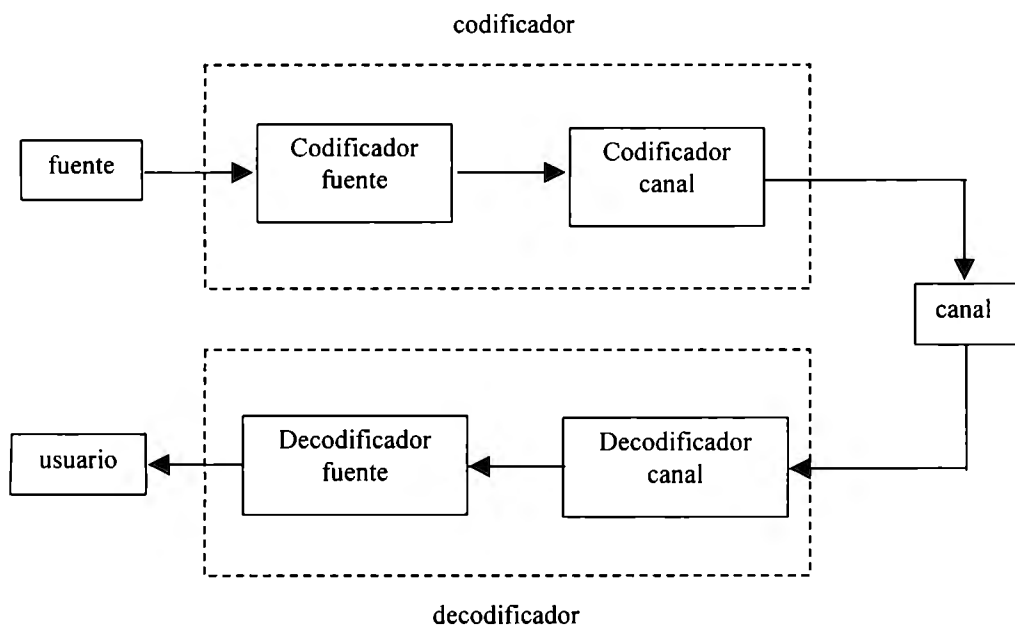


Figura 21. Modelo del sistema de comunicación

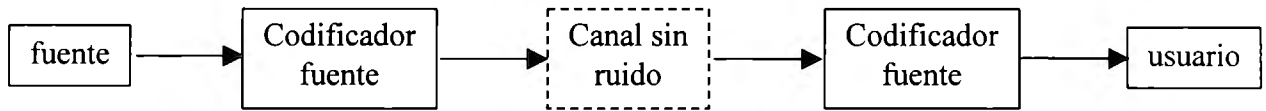


Figura 22. Modelo de codificación fuente

Suponiendo que la velocidad del canal es fija, R bits por símbolo de salida fuente. Para N salidas fuente habrá $K = N/R$ bits que cruzan el canal sin ruido del codificador fuente al decodificador fuente, lo que significa que el decodificador fuente puede seleccionar cualquiera de las $M = 2^K$ secuencias de usuario posibles de longitud N para representar la secuencia de salida fuente de longitud N . La cantidad promedio de distorsión entre N símbolos de salida fuente $\underline{u} = (u_1, u_2, \dots, u_N)$ y N símbolos de representación $\underline{v} = (v_1, v_2, \dots, v_N)$ está dada por

$$d_N(\underline{u}, \underline{v}) = \frac{1}{N} \sum_{n=1}^N d(u_n, v_n) \quad (162)$$

Sea $(\underline{v}_1, \underline{v}_2, \dots, \underline{v}_M)$ el arreglo de M secuencias de representación de N símbolos usuarios, a partir de los cuales el decodificador fuente hace la selección cuando se reciben K bits sobre el canal. Dada la secuencia de salida $\underline{u} \in \mathcal{U}$, el codificador fuente puede seleccionar la secuencia que minimice $d_N(\underline{u}, \underline{v})$ $m = 1, 2, \dots, M$ enviando al decodificador fuente los K bits correspondientes.

2.3.1 CÓDIGOS BLOQUE

Con N fija y asumiendo que no existen restricciones en como el decodificador fuente emplea los K bits de la salida del canal para seleccionar una secuencia de representación de N símbolos, sea $\mathcal{B} = (\underline{v}_1, \underline{v}_2, \dots, \underline{v}_M)$ un arreglo de M secuencias de representaciones y longitud de bloque N y cada secuencia en \mathcal{B} se llama palabra código. El código \mathcal{B} se usa para codificar una secuencia

fuelle $\underline{u} \in \mathcal{U}_N$ seleccionando la palabra código $\underline{v} \in \mathcal{B}$ que minimice $d_N(\underline{u}, \underline{v})$. Este mínimo se denota como

$$d(\underline{u} | \mathcal{B}) = \min_{\underline{v} \in \mathcal{B}} d_N(\underline{u}, \underline{v}) \quad (163)$$

definiendo la distorsión promedio lograda con el código \mathcal{B} como

$$d(\mathcal{B}) = \sum_{\underline{u}} Q_N(\underline{u}) d(\underline{u} | \mathcal{B}) \quad (164)$$

donde

$$Q_N(\underline{u}) = \prod_{n=1}^N Q(u_n) \quad (165)$$

Introduciendo una distribución de probabilidad condicional $\{P(\underline{u} | \underline{v}): \underline{v} \in \mathcal{V}, \underline{u} \in \mathcal{U}\}$, para secuencias $\underline{u} \in \mathcal{U}_N$ y $\underline{v} \in \mathcal{V}_N$ se asume la independencia condicional con esta distribución por lo que

$$P_N(\underline{v} | \underline{u}) = \prod_{n=1}^N P(v_n | u_n) \quad (166)$$

Las probabilidades incondicionales correspondientes son

$$\begin{aligned} P_N(\underline{v}) &= \sum_{\underline{u}} P_N(\underline{v} | \underline{u}) Q_N(\underline{u}) \\ &= \prod_{n=1}^N P(v_n) \end{aligned} \quad (167)$$

donde

$$P(v) = \sum_u P(v|u)Q(u) \quad (168)$$

Aplicando la regla de Bayes

$$\begin{aligned} Q_N(\underline{u}|\underline{v}) &= \frac{P_N(\underline{v}|\underline{u})Q_N(\underline{u})}{P_N(\underline{v})} \\ &= \prod_{n=1}^N Q(u_n | v_n) \end{aligned} \quad (169)$$

donde

$$Q(u|v) = \frac{P(v|u)Q(u)}{P(v)} \quad (170)$$

A la probabilidad condicional $\{P(v|u): v \in \mathcal{V}, u \in \mathcal{U}\}$ no se le asigna un significado físico, solo se emplea como una herramienta para derivar límites en la distorsión promedio para un código \mathcal{B} de tamaño M y longitud de bloque N .

De la ecuación 164 la distorsión promedio usando el código \mathcal{B} es

$$d(\mathcal{B}) = \sum_{\underline{u}} Q_N(\underline{u}) d(\underline{u} | \mathcal{B}) \quad (171)$$

Como $\sum_{\underline{v}} P_N(\underline{v}, \underline{u}) = 1$ puede escribirse

$$d(\mathcal{B}) = \sum_{\underline{u}} \sum_{\underline{v}_0} Q_N(\underline{u}) P_N(\underline{v}_0 | \underline{u}) d(\underline{u} | \mathcal{B}) \quad (172)$$

$\underline{v}_0 \in \mathcal{V}_N^+$ no es una palabra código, solo una variable “dummy” de suma. Dividiendo la suma sobre \underline{u} y \underline{v}_0 en dos regiones disjuntas mediante la definición de la función indicador

$$\phi(\underline{u}, \underline{v}_0; \mathcal{B}) = \begin{cases} 1 & d_N(\underline{u}, \underline{v}_0) < d_N(\underline{u} | \mathcal{B}) \\ 0 & d_N(\underline{u}, \underline{v}_0) \geq d_N(\underline{u} | \mathcal{B}) \end{cases} \quad (173)$$

como $[1 - \phi] + \phi = 1$ se obtiene

$$\begin{aligned} d(\mathcal{B}) &= \sum_{\underline{u}} \sum_{\underline{v}_0} Q_N(\underline{u}) P_N(\underline{v}_0 | \underline{u}) d(\underline{u} | \mathcal{B}) [1 - \phi(\underline{u}, \underline{v}_0; \mathcal{B})] + \\ &\quad \sum_{\underline{u}} \sum_{\underline{v}_0} Q_N(\underline{u}) P_N(\underline{v}_0 | \underline{u}) d(\underline{u} | \mathcal{B}) \phi(\underline{u}, \underline{v}_0; \mathcal{B}) \end{aligned} \quad (174)$$

usando desigualdades

$$d(\underline{u} | \mathcal{B}) [1 - \phi(\underline{u}, \underline{v}_0; \mathcal{B})] \leq d_N(\underline{u} | \underline{v}_0) \quad (175)$$

en la primer suma y desigualdad

$$\begin{aligned} d(\underline{u} | \mathcal{B}) &= \min_{\underline{v} \in \mathcal{B}} d_N(\underline{u} | \underline{v}_0) \\ &\leq d_0 \end{aligned} \quad (176)$$

en la segunda suma en la ecuación 174 produce el límite

$$\begin{aligned} d(\mathcal{B}) &\leq \sum_{\underline{u}} \sum_{\underline{v}_0} Q_N(\underline{u}) P_N(\underline{v}_0 | \underline{u}) d_N(\underline{u} | \underline{v}_0) + \\ &\quad d_0 \sum_{\underline{u}} \sum_{\underline{v}_0} Q_N(\underline{u}) P_N(\underline{v}_0 | \underline{u}) \phi(\underline{u}, \underline{v}_0; \mathcal{B}) \end{aligned} \quad (177)$$

El primer término en este límite se simplifica a

$$\sum_{\underline{u}} \sum_{\underline{v}_0} Q_N(\underline{u}) P_N(\underline{v}_0 | \underline{u}) d_N(\underline{u} | \underline{v}_0) = \sum_{\underline{u}} \sum_{\underline{v}} Q(\underline{u}) P(\underline{v} | \underline{u}) d(\underline{u}, \underline{v})$$

$$\cong D(\underline{P})$$

(178)

Para limitar el segundo término se aplica el argumento de codificación aleatoria, considerando un conjunto de códigos bloque de tamaño M y longitud de bloque N donde la probabilidad de escoger $\mathcal{B} = \{v_{-1}, v_{-2}, \dots, v_{-M}\}$ está dada por

$$P(\mathcal{B}) = \prod_{m=1}^M P_N(v_{-m}) \quad (179)$$

$P_N(v)$ está dada por la ecuación 167, la distribución incondicional correspondiente a la distribución de probabilidad condicional inicial $\{P(v|u): v \in \mathcal{V}, u \in \mathcal{U}\}$. Promediando sobre este conjunto código se obtiene

$$\overline{d(\mathcal{B})} \leq \overline{D(\underline{P})} + d_0 \Pr\{d_N(\underline{u}, \underline{v}_0) < d(\underline{u}, \mathcal{B})\} \quad (180)$$

donde

$$\overline{\Pr\{d_N(\underline{u}, \underline{v}_0) < d(\underline{u}, \mathcal{B})\}} = \sum_{\underline{u}} \sum_{\underline{v}_0} Q_N(\underline{u}) P_N(\underline{v}_0 | \underline{u}) \phi(\underline{u}, \underline{v}_0; \mathcal{B}) \quad (181)$$

Para limitar la ecuación 181 también se considera la situación de codificación de canal mostrada en la Figura 23, con un DMC con alfabeto de entrada \mathcal{U} , alfabeto de salida \mathcal{V} , y probabilidad de transición $\{Q(u|v): u \in \mathcal{U}, v \in \mathcal{V}\}$. Considerando un código bloque canal que consiste de $M + 1$ palabras código $v_{-0}, v_{-1}, v_{-2}, \dots, v_{-M}$, y una regla de decodificación subóptima basada en las distorsiones $d(\underline{u}, \underline{v}_{-0}), d_N(\underline{u}, \underline{v}_{-1}), d_N(\underline{u}, \underline{v}_{-2}), \dots, d_N(\underline{u}, \underline{v}_{-M})$.

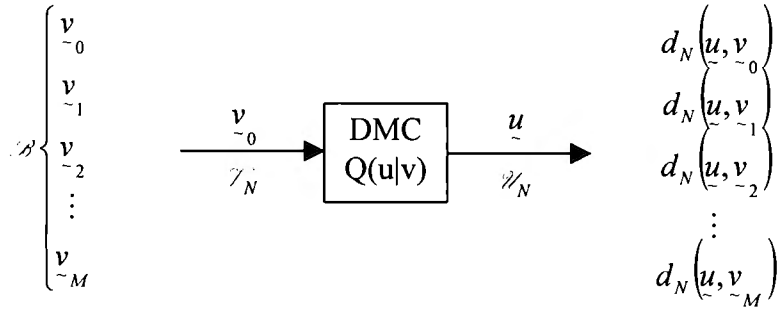


Figura 23. Canal de prueba hacia atrás

Cuando todas las palabras código son igualmente probables la probabilidad máxima de decisión correcta está dada por (ecuación 16)

$$P_C = \frac{1}{M+1} \sum_u \max_m Q_N(u, v_{-m}) \quad (182)$$

y $\Pr\{d_N(u, v_{-0}) < d(u | \mathcal{B})\}$, donde $d(u | \mathcal{B}) = \min_{m \geq 1} d_N(u, v_{-m})$ es la probabilidad de decisión correcta usando la regla de distorsión mínima, y v_{-0} es la palabra código transmitida. Entonces, promediar sobre el conjunto código es equivalente a seleccionar las palabras código de la Figura 23 de acuerdo a

$$\Pr\{v_{-0}, v_{-1}, \dots, v_{-M}\} = \prod_{m=0}^M P_N(v_{-m}) \quad (183)$$

como existe simetría con respecto a todas las palabras código,

$$\Pr\{d_N(u, v_{-0}) < \min_{m \geq 1} d_N(u, v_{-m})\} \leq P_C \quad (184)$$

Para cualquier $-1 < \rho$ se usa el límite

$$\begin{aligned}
\max_m Q_N(u|v_M) &\leq \left(\max_m Q_N(u|v_M)^{\frac{1}{1+\rho}} \right) \\
&\leq \left(\sum_{m=0}^M Q_N(u|v_M)^{\frac{1}{1+\rho}} \right)^{\frac{1}{1+\rho}}
\end{aligned}
\tag{185}$$

por lo que

$$P_C \leq \frac{1}{M+1} \sum_u \left(\sum_{m=0}^M Q_N(u|v_M)^{\frac{1}{1+\rho}} \right)^{\frac{1}{1+\rho}}
\tag{186}$$

Requiriendo $-1 < \rho \leq 0$ y al aplicar la desigualdad de Jensen se obtiene

$$\begin{aligned}
\overline{\Pr\{d(u|v_0) < d(u|\mathcal{S})\}} &\leq \frac{1}{M+1} \sum_u \left(\sum_{m=0}^M Q_N(u|v_M)^{\frac{1}{1+\rho}} \right)^{\frac{1}{1+\rho}} \\
&\leq \frac{1}{M+1} \sum_u \left(\sum_{m=0}^M Q_N(u|v_M)^{\frac{1}{1+\rho}} \right)^{\frac{1}{1+\rho}} \\
&\leq (M+1)^\rho \sum_u \left(\sum_v P_N(v) Q_N(u|v_M)^{\frac{1}{1+\rho}} \right)^{\frac{1}{1+\rho}} \\
&\leq 2^{\rho RN} \left[\sum_u \left(\sum_v P(v) Q(u|v)^{\frac{1}{1+\rho}} \right)^{\frac{1}{1+\rho}} \right]^N \\
&= 2^{-N[E_0(\rho, P) - \rho R]}
\end{aligned}
\tag{187}$$

donde

$$E_0(\rho, P) = -\log \sum_u \left(\sum_v P(v) Q(u|v)^{\frac{1}{1+\rho}} \right)^{\frac{1}{1+\rho}}
\tag{188}$$

tiene la forma de la Figura 12 con

$$\begin{aligned}
 E_0(\rho, \underline{P})_{\rho=0} &= \sum_u \sum_v P(v) Q(u|v) \log \frac{Q(u|v)}{Q(u)} \\
 &= \sum_u \sum_v P(v|u) Q(u) \log \frac{P(v|u)}{P(v)} \\
 &\triangleq I(\underline{P})
 \end{aligned}
 \tag{189}$$

Combinando estos resultados en el límite de la distorsión promedio usando códigos bloque de longitud N y velocidad R , resulta

$$\overline{d(\mathcal{B})} \leq D(\underline{P}) + d_0 2^{-NE(R; \rho, \underline{P})}
 \tag{190}$$

para cualquier $-1 < \rho < 0$. Minimizando este límite con respecto a ρ se produce

$$\overline{d(\mathcal{B})} \leq D(\underline{P}) + d_0 2^{-N \left\{ \max_{-1 \leq \rho \leq 0} E(R; \rho, \underline{P}) \right\}}
 \tag{191}$$

de la Figura 12 se observa que

$$\max_{-1 \leq \rho \leq 0} E(R; \rho, \underline{P}) > 0 \text{ para } R > I(\underline{P})
 \tag{192}$$

Entonces se escoge la probabilidad condicional $\{P(v|u)\}$ con el fin de minimizar aún más el límite $\overline{d(\mathcal{B})}$. Asumiendo un criterio de fidelidad D que deberá satisfacer el sistema codificador y decodificador bloque fuente, y definiendo un arreglo de probabilidades condicionales que satisfagan la condición $D(\underline{P}) \leq D$, se obtiene

$$\mathcal{A}_D = \{P(v|u) : D(\underline{P}) \leq D\}
 \tag{193}$$

Definiendo ahora una función de confiabilidad de la fuente

$$E(R, D) = \max_{\underline{P} \in \mathcal{A}_D} \max_{-1 \leq \rho \leq 0} E(R; \rho, \underline{P}) \quad (194)$$

y la función

$$R(D) = \min_{\underline{P} \in \mathcal{A}_D} I(\underline{P}) \quad (195)$$

conocida como función de distorsión de velocidad de la fuente. Finalmente, se obtiene que para cualquier bloque de longitud N y velocidad R existe un código bloque \mathcal{B} con distorsión promedio $d(\mathcal{B})$ que satisface

$$d(\mathcal{B}) \leq D + d_0 2^{-NE(R, D)} \quad (196)$$

donde

$$E(R, D) > 0 \text{ para } R > R(D) \quad (197)$$

Por ejemplo, sea $\mathcal{U} = \mathcal{V} = \{0, 1\}$, $d(u, v) = 1 - \delta_{uv}$, y $Q(0) = Q(1) = 1/2$. Por simetría, la distribución $\underline{P} \in \mathcal{A}_D$ que da $E(R, D)$ y $R(D)$ está dada por

$$P(v|u) = \begin{cases} D & v \neq u \\ 1 - D & v = u \end{cases} \quad (198)$$

y $E(R, D)$ se define paramétricamente como

$$E(R, D) = T_D(\delta) - \mathcal{H}(\delta) \quad (199)$$

$$R = 1 - \mathcal{H}(\delta)$$

donde

$$0 \leq \delta \leq D$$

y la función de distorsión de velocidad es

$$R(D) = 1 - \delta(D) \quad (200)$$

De las ecuaciones 196 y 197 se observa que al incrementar la longitud N del bloque puede obtenerse una distorsión promedio cercana a D mientras que la velocidad del código sea mayor que $R(D)$, la función de distorsión de velocidad.

2.3.2 CÓDIGOS TRELIS

Suponiendo que la velocidad del canal en la Figura 22 es

$$R = \frac{K}{N} = \frac{1}{n} \text{ bits / simbolo fuente} \quad (201)$$

donde n es un entero, y se puede agrupar cualquier secuencia de salida de la fuente $\underline{u} \in \mathcal{Z}_N$ en subsecuencias de n símbolos fuente,

$$\underline{u} = (\underline{u}^1, \underline{u}^2, \dots, \underline{u}^K) \quad (202)$$

donde $\underline{u}^k \in \mathcal{Z}_n$, para $k = 1, 2, \dots, K$. De forma similar, cualquier secuencia de representación $\underline{v} \in \mathcal{Z}_N$ puede agruparse en subsecuencias de longitud n ,

$$\underline{v} = \left(\underline{v}^1, \underline{v}^2, \dots, \underline{v}^K \right) \underline{u} \in \mathcal{N}_N \quad (203)$$

donde $\underline{v}^k \in \mathcal{N}_n$, para $k = 1, 2, \dots, K$.

Como se mencionó anteriormente, para los códigos bloque no existe restricción para seleccionar la secuencia de representación de los N símbolos en K bits de canal, pero en los códigos trellis se tiene la restricción de que \underline{v}^k depende de los bits de canal $b_k, b_{k-1}, \dots, b_{k-\nu+1}$, donde ν es la restricción de longitud o memoria del código trellis y (b_1, b_2, \dots, b_k) son los K bits de canal transmitidos cuando ocurren $N = nK$ símbolos fuente, representado como

$$\begin{aligned} \underline{v}^k &= \underline{g}(b_k, b_{k-1}, \dots, b_{k-\nu+1}) \\ k &= 1, 2, \dots, K \end{aligned} \quad (204)$$

donde inicialmente $b_k = 0$ para $k \leq 0$. Nuevamente los códigos trellis se ven como un caso especial de los códigos bloque y el análisis solo depende de $R = 1/n$ y de la longitud de restricción ν pudiendo seleccionar una longitud N de bloque larga.

El vector de estados, en el tiempo k , se define como

$$\underline{s}^k = \underline{g}(b_{k-\nu+2}, b_{k-\nu+3}, \dots, b_{k-1}, b_k) \quad (205)$$

con representación

$$\underline{v}^k = \underline{g}(b_k, \underline{s}^{k-1}) \quad (206)$$

El código trellis definido por la ecuación 206 puede representarse adecuadamente con un diagrama de estados o diagrama trellis, para el análisis el diagrama trellis es mejor. Por ejemplo, el código convolucional canal de la Figura 14, puede ser un código trellis fuente para una fuente con alfabeto binario $\mathcal{N} = \mathcal{V} = \{0,1\}$, velocidad $R = 1/2$, y $\nu = 3$.

Los K bits de canal especifican una de las $M = 2^K$ posibles secuencias de representación que se genera de acuerdo a la ecuación 206. Cada una de las M secuencias corresponden a los valores de las ramas a lo largo de algún camino en el diagrama trellis. En el diagrama trellis existe una rama para cada uno de los K bits de canal y n símbolos de representación en cada rama resultante en una secuencia de representación de longitud $N = nK$. Dada la secuencia de salida de la fuente $\underline{u} \in \mathcal{Z}_N$ el codificador fuente selecciona el camino en el diagrama trellis con secuencia de representación $\underline{v} \in \mathcal{Z}_N$ que minimiza $d_N(\underline{u}, \underline{v})$

$$d_N(\underline{u}, \underline{v}) = \frac{1}{K} \sum_{k=1}^K d_n(\underline{u}^k, \underline{v}^k) \quad (207)$$

donde

$$d_n(\underline{u}^k, \underline{v}^k) = \frac{1}{n} \sum_{i=1}^n d(\underline{u}_i^k, \underline{v}_i^k) \quad (208)$$

$$k = 1, 2, \dots, K$$

la distorsión de cualquier rama es independiente de distorsiones debidas a otras ramas en el mismo camino. Para encontrar el camino de distorsión mínima en el trellis se usa el hecho de que cuando dos caminos se unen en un estado, uno de los caminos puede eliminarse de consideraciones posteriores, entonces el codificador fuente puede encontrar el camino de distorsión mínima en el diagrama trellis usando el algoritmo de Viterbi. Encontrado el camino de distorsión mínima, el codificador fuente saca la secuencia binaria que hace que el decodificador fuente, definido por la ecuación 206, obtenga la secuencia del camino de distorsión mínima.

Para examinar el desempeño de los códigos trellis se utiliza el argumento de codificación aleatoria, donde se considera un conjunto de códigos trellis fuente binarios con restricción de longitud ν y velocidad $R = 1/n$. El conjunto se escoge de tal forma que sobre el conjunto el diagrama trellis se distribuyan de forma aleatoria variable sobre todas las ramas con distribución de probabilidad $\{P(\nu): \nu \in \mathcal{Z}^+\}$. Para cualquier secuencia fuente \underline{u} y cualquier código trellis, la

secuencia de camino de distorsión mínima se denota como $\underline{v}(u)$. Por definición se tiene el límite $d_N(u, \underline{v}(u)) \leq d_N(u, v)$ para cualquier secuencia de camino v que pertenezca al código trellis. \underline{v} puede ser cualquier camino en el diagrama trellis mientras $\underline{v}(u)$ sea el camino de distorsión mínima. Escogiendo $\underline{v} = \underline{v}^*$ de la siguiente forma:

- a) Para un código trellis se reemplaza la secuencia de representación del camino de estados de puros ceros por la secuencia \underline{v}_0 , seleccionada aleatoriamente de acuerdo a la probabilidad condicional

$$P_N(\underline{v}_0, u) = \prod_{t=1}^N P(v_{0t} | u_t) \quad (209)$$

De lo que resulta en un código trellis nuevo, conocido como código trellis prohibido, que difiere del original solo en los valores del camino de estados de puros ceros, Se llama código prohibido porque, en general, no se puede seleccionar alguna parte del código trellis después de la secuencia de salida de la fuente u .

- b) Dada una secuencia fuente u , para el código prohibido sea \underline{v}^{**} la secuencia del camino de distorsión mínima, es decir, \underline{v}^{**} corresponde a la secuencia de salida del código trellis prohibido que representa u con distorsión mínima.
- c) \underline{v}^{**} define el camino a través del diagrama trellis prohibido, y \underline{v}^* es la secuencia del camino del diagrama trellis original. Entonces, \underline{v}^{**} y \underline{v}^* son iguales excepto por las subsecuencias en las ramas del camino del estado de puros ceros.

\underline{v}^* es la secuencia del código trellis en el código trellis original y se introdujo el código trellis prohibido como un medio para seleccionar la secuencia del código trellis. Nunca se usa un código

trellis prohibido en la codificación las secuencias fuente. Como \underline{v}^* es una secuencia de camino en el código trellis de la definición $\underline{v}(\underline{u})$, se tiene

$$d_{N_L}(\underline{u}, \underline{v}(\underline{u})) \leq d_{N_L}(\underline{u}, \underline{v}^*) \quad (210)$$

Para derivar un límite en $d_{N_L}(\underline{u}, \underline{v}(\underline{u}))$, promediando sobre todas las secuencias fuente y códigos fuente en el conjunto, se limita $d_{N_L}(\underline{u}, \underline{v}^*)$.

Para una secuencia fuente $\underline{u} \in \mathcal{U}_N$ y $\underline{v}^* \in \mathcal{V}_N$ seleccionada arriba sea

$$\mathcal{X} = \{k : \underline{v}^{*k} \text{ es una rama vector del camino de estados de ceros}\} \quad (211)$$

$$\begin{aligned} Kd_N(\underline{u}, \underline{v}^*) &= \sum_{k=1}^K d_N(\underline{u}^k, \underline{v}^{*k}) \\ &= \sum_{k \in \mathcal{X}} d_N(\underline{u}^k, \underline{v}^{*k}) + \sum_{k \notin \mathcal{X}} d_N(\underline{u}^k, \underline{v}^{*k}) \end{aligned} \quad (212)$$

Para $k \in \mathcal{X}$ se usa el límite $d_n(\underline{u}^k, \underline{v}^{*k}) \leq d_0$ mientras que para $k \notin \mathcal{X}$ se tiene $d_n(\underline{u}^k, \underline{v}^{*k}) = d_n(\underline{u}^k, \underline{v}^{**k})$. Entonces

$$\begin{aligned} Kd_N(\underline{u}, \underline{v}^*) &\leq \sum_{k \in \mathcal{X}} d_n(\underline{u}^k, \underline{v}^{**k}) + \sum_{k \notin \mathcal{X}} d_0 \\ &\leq \sum_{k=1}^K d_n(\underline{u}^k, \underline{v}^{**k}) + \sum_{k \in \mathcal{X}} d_0 \\ &\leq \sum_{k=1}^K d_n(\underline{u}^k, \underline{v}_0^k) + \sum_{k \in \mathcal{X}} d_0 \end{aligned} \quad (213)$$

donde \underline{v}_0^k es la k -ésima rama de la secuencia de salida del camino de puros ceros del código trellis prohibido. La última desigualdad se obtiene del hecho de que por definición de \underline{v}^{**} se tiene

$$d_N(u, \underline{v}^{**}) \leq d_N(u, \underline{v}_0) \quad (214)$$

en el código trellis prohibido donde \underline{v}_0 es la secuencia de salida del estado de puros ceros.

Entonces de la ecuación 213 se tiene

$$d_N(u, \underline{v}(u)) \leq d_N(u, \underline{v}_0) + \sum_{k \in \mathcal{X}} \frac{d_0}{K} \quad (215)$$

Definiendo

$$P_{jk} = \Pr \left\{ \underline{v}^* \text{ se une con el estado de ceros en el nodo } j \text{ y permanece unido exactamente } k \text{ ramas} \right\} \quad (216)$$

y

$$D(\underline{P}) = \sum_u \sum_v Q(u) P(v|k) d(u, v) \quad (217)$$

se puede promediar la ecuación 215 sobre todas las secuencias fuente y el conjunto de código trellis. El primer término se convierte en $D(\underline{P})$, empleando la unión de límite de eventos en el segundo término se obtiene

$$d_N(u, \underline{v}(u)) \leq D(\underline{P}) + \frac{d_0}{K} \sum_{j=1}^K \sum_{k=1}^{K-j} k \overline{P_{jk}} \quad (218)$$

P_{jk} también es la probabilidad de que \underline{v}^{**} se una con el camino de puros ceros \underline{v}_0 en el nodo j y permanezca unido k ramas. Como \underline{v}_0 es el único camino dependiente de la secuencia fuente

$\underline{u} \in \mathbb{Z}_N$, en el código trellis prohibido se tiene una situación similar a la del código canal. P_{jk} puede limitarse usando los resultados de codificación bloque de la ecuación 187 donde

$$M_{jk} = 2^{k+\nu-1} \quad (219)$$

es el número de palabras código y

$$N_{jk} = nk \quad (220)$$

la longitud del bloque.

Entonces

$$\begin{aligned} \overline{P}_{jk} &= 2^{(k+\nu-1)\rho} 2^{-nkE_0(\rho, \underline{P})} \\ &= 2^{(\nu-1)\rho} 2^{-k\left\{\left[E_0(\rho, \underline{P})/R\right] - \rho\right\}} \end{aligned} \quad (221)$$

Usando esta ecuación en la ecuación 218 se obtiene

$$\begin{aligned} d_N(\underline{u}, \underline{v}(\underline{u})) &\leq D(\underline{P}) + \frac{d_0}{K} \sum_{j=1}^K \sum_{k=1}^{K-j} k 2^{(\nu-1)\rho} 2^{-k\left\{\left[E_0(\rho, \underline{P})/R\right] - \rho\right\}} \\ &= D(\underline{P}) + d_0 \sum_{k=1}^{\infty} k 2^{(\nu-1)\rho} 2^{-k\left\{\left[E_0(\rho, \underline{P})/R\right] - \rho\right\}} \\ &= D(\underline{P}) + \frac{d_0 2^{(\nu-1)\rho}}{\left[1 - 2^{-\left\{\left[E_0(\rho, \underline{P})/R\right] - \rho\right\}}\right]^2} \end{aligned} \quad (222)$$

haciendo $\frac{E_0(\rho, R)}{R - \rho} > 0$ ó $E_0(\rho, R) > \rho R$. De la Figura 12 se puede observar que mientras $R > I(\underline{P})$ puede encontrarse una $-1 < \rho \leq 0$ tal que satisfaga $E_0(\rho, R) > \rho R$.

Para $\delta > 0$ pequeña y fidelidad D , se escoge $\underline{P} \in \mathcal{P}_D$ tal que $I(\underline{P}) = R(D)$ y ρ sea el valor más pequeño en $-1 \leq \rho \leq 0$ que cumpla $\frac{E_0(\rho, R)}{R - \rho} > \delta$.

Entonces

$$d(u, v(\underline{u})) \leq D + \frac{d_0 2^{(v-1)E_T(R, D)}}{[1 - 2^{-\delta}]^2} \quad (223)$$

la Figura 12 muestra

$$\lim_{\delta \rightarrow 0} E_T(R, D) = \begin{cases} +1 & R \geq -E_0(-1, \underline{P}) \\ -\rho & \text{donde } E_0(\rho, \underline{P}) = \rho R; R(D) < R < -E_0(-1, \underline{P}) \end{cases} \quad (224)$$

donde $\lim_{\delta \rightarrow 0} E_T(R, D) > 0$ para $R > R(D)$.

Por ejemplo, sea $\mathcal{X} = \mathcal{Y} = \{0, 1\}$, $d(u, v) = 1 - \delta_{uv}$, y $Q(0) = Q(1) = 1/2$ que produce

$$P(v|u) = \begin{cases} D & v \neq u \\ 1 - D & v = u \end{cases} \quad (225)$$

y

$$E_0(\rho, \underline{P}) = -(1 + \rho) \log \left(D^{\frac{1}{1+\rho}} + (1-D)^{\frac{1}{1+\rho}} \right) \quad (226)$$

$$E_0(-1, \underline{P}) = -\log 2(1-D)$$

Por la ecuación 223 se sabe que existe un código trellis \mathcal{C} de velocidad $R = 1/n$ y restricción de longitud n tal que la distorsión promedio $d(\mathcal{C})$ satisface

$$d(\mathcal{C}) \leq D + d_0 \frac{2^{-(\nu-1)E_T(R,D)}}{[1-2^{-\delta}]^{\nu}} \quad (227)$$

donde $E_T(R, D) > 0$ para $R > R(D)$. Para D y R fijas cercanas a $R(D)$, en general, $E_T(R, D)$ es mayor que el exponente del código bloque correspondiente $E(R, D)$. Nuevamente se observa que para la misma memoria del código fuente, en este caso la complejidad del codificador fuente, los códigos trellis se desempeñan mejor que los códigos bloques. Estos resultados de los códigos trellis puede generalizarse para velocidades $R = b/n$, donde b es un entero.

2.3.3 TEOREMA DE CONVERGENCIA

Para mostrar que $R(D)$ es la función de distorsión de velocidad se muestra que es imposible lograr una distorsión promedio de D o menor con cualquier par codificador-decodificador fuente con velocidad $R < R(D)$.

Para mostrar esto se necesitan dos propiedades de $I(\underline{P})$: sea $\{P_N(\underline{v} | \underline{u}) : \underline{v} \in \mathcal{V}_N, \underline{u} \in \mathcal{U}_N\}$ distribución condicional arbitraria en las secuencias de longitud N , y $P^{(t)}(\underline{v}_t | \underline{u}_t)$ la distribución condicional marginal para el t -ésimo par (v_t, u_t) derivado de esta distribución. Definiendo

$$I(\underline{P}_N) = \sum_{\underline{u}} \sum_{\underline{v}} Q_N(\underline{u}) P_N(\underline{v} | \underline{u}) \log \frac{P_N(\underline{v} | \underline{u})}{P_N(\underline{v})} \quad (228)$$

y

$$I(\underline{P}^{(t)}) = \sum_u \sum_v Q(u) P^{(t)}(v|u) \log \frac{P^{(t)}(v|u)}{P^{(t)}(v)} \quad (229)$$

donde

$$Q_N(\underline{u}) = \prod_{i=1}^N Q(u_i) \quad (230)$$

y

$$P^{(t)}(v) = \sum_u Q(u) P^{(t)}(v|u) \quad (231)$$

obteniendo las siguientes desigualdades

$$I\left(\frac{1}{N} \sum_{i=1}^N \underline{P}^{(i)}\right) \leq \frac{1}{N} \sum_{i=1}^N I(\underline{P}^{(i)}) \quad (232)$$

y

$$\frac{1}{N} I\left(\underline{P}_N\right) \geq \frac{1}{N} \sum_{i=1}^N I(\underline{P}^{(i)}) \quad (233)$$

Cualquier par codificador-decodificador define un mapeo de las secuencias fuente a las secuencias usuario. Para cualquier longitud N se mapea de \mathcal{X}_N a \mathcal{Y}_N donde M puede ser un número distinto de secuencias en \mathcal{Y}_N en las que las secuencias \mathcal{X}_N son mapeadas. Definiendo la distribución condicional

$$P_N(\underline{v} | \underline{u}) \triangleq \begin{cases} 1 & \text{si } \underline{u} \text{ es mapeada en } \underline{v} \\ 0 & \text{en otro caso} \end{cases} \quad (234)$$

Sea $P^{(i)}$ el resultado de la distribución marginal en el N -ésimo término de la secuencia, y definiendo la distribución condicional

$$P_N(\underline{v} | \underline{u}) \triangleq \frac{1}{N} \sum_{i=1}^N P^{(i)}(v | u) \quad (235)$$

y asumiendo que del mapeo resulta en una distorsión promedio D o menor, entonces

$$\begin{aligned} D(P_N) &= \sum_{\underline{u}} \sum_{\underline{v}} Q_N(\underline{u}) P_N(\underline{v} | \underline{u}) d_N(\underline{u} | \underline{v}) \\ &= \sum_{\underline{u}} Q_N(\underline{u}) d_N(\underline{u}, \underline{v}(\underline{u})) \\ &\leq D \end{aligned} \quad (236)$$

Como $P(v | u)$ dada por la ecuación 235 pertenece a \mathcal{P}_d , entonces

$$\begin{aligned} R(D) &\leq I(\underline{P}) \\ &= I\left(\frac{1}{N} \sum_{i=1}^N P^{(i)}\right) \\ &\leq \frac{1}{N} \sum_{i=1}^N I(\underline{P}^{(i)}) \\ &\leq \frac{1}{N} I(\underline{P}^{(i)}) \\ &\leq \frac{1}{N} \log M \\ &= R \end{aligned}$$

$$R(D) < R \quad (237)$$

Usando las desigualdades 232, 233 e $I(\tilde{P}_N) \leq \log M$.

Lo cual aplica a todos los pares codificador-decodificador canal, y junto con los resultados de codificación bloque y trellis se muestra que $R(D)$ representa la velocidad mínima posible requerida para lograr la distorsión promedio para fuente discreta sin memoria con una medida de distorsión de una letra.

2.3.4 DISCUSIÓN

Para concluir se examinarán algunas relaciones entre la codificación fuente y canal. El codificador fuente remueve redundancia de las secuencias fuente mientras que la codificación canal agrega redundancia. El codificador fuente toma ventaja de la regularidad estadística de secuencias largas de la salida de la fuente para representarlas con una velocidad limitada $R(D)$. Este proceso se realiza de tal forma que cuando la n -ésima letra en $\underline{u} \in \mathcal{U}_N$ es u_n , la probabilidad condicional de la n -ésima letra en la palabra código seleccionada $\underline{v} \in \mathcal{V}_N$, v_n , es muy cercana a $P(v_n|u_n)$ donde $\{P(v|v): \underline{u} \in \mathcal{U}_N, \underline{v} \in \mathcal{V}_N\}$ cumple con la función de distorsión de velocidad. Mientras que el codificador canal agrega redundancia con el fin de lograr inmunidad a los errores de canal.

Suponiendo un DMC con alfabeto de entrada \mathcal{X} , alfabeto de salida \mathcal{Y} , y probabilidades condicionales de transición $\{P(y|x): y \in \mathcal{Y}, x \in \mathcal{X}\}$, para dos letras cualquiera de entrada al canal x y $x' \in \mathcal{X}$, se tiene que la distancia de Bhattacharyya se define como

$$d(x, x') = -\log \sum_y \sqrt{P(y|x)P(y|x')} \quad (238)$$

Suponiendo que las letras de entrada al canal tienen una distribución de probabilidad $\{Q(x): x \in \mathcal{X}\}$, para una fuente con alfabeto $\mathcal{X} = \mathcal{X}$, distribución de probabilidad $\{Q(x): x \in \mathcal{X}\}$, alfabeto de representación $\mathcal{Y} = \mathcal{X}$, y distancia de Bhattacharyya definida por la ecuación 238 como medida de distorsión, se tiene la función de distorsión denotada por $R(D; \underline{Q})$. La distorsión de velocidad para este DMC se define como

$$R(D) = \max_{\underline{Q}} R(D; \underline{Q}) \quad (239)$$

Para mostrar la relación entre $K(D)$ y el exponente expurgado para el DMC, se toma el BSC con una probabilidad de crossover ε , $\mathcal{X} = \mathcal{Y} = \{0,1\}$ y la medida de distorsión es

$$d(x, x') = \begin{cases} 0 & x = x' \\ -\log \sqrt{4\varepsilon(1-\varepsilon)} & x \neq x' \end{cases} \quad (240)$$

Si $\alpha = -\log \sqrt{4\varepsilon(1-\varepsilon)}$, la distancia de Bhattacharyya es proporcional a la distancia de Hamming, y

$$R(D) = 1 - \mathcal{H}\left(\frac{D}{\alpha}\right) \quad (241)$$

De las ecuaciones 69 a la 71 se obtiene que existe un código bloque \mathcal{C} de longitud N y velocidad R tal que

$$P_E \leq 2^{-NE_{ex}(R)} \quad (242)$$

donde $D = E_{ex}(R)$ satisface $R = R(D)$ y $R(D)$ está dada por 241.

Entonces, la función de distorsión de velocidad natural para el BSC produce el exponente expurgado como un nivel de distorsión.

El límite de Gilbert puede probarse usando la relación anterior y la teoría de distorsión de velocidad. Sea

$$d(N, R) = \max_C d_{\min}(C) \quad (243)$$

donde

$$d_{\min}(C) = \min_{\substack{x, x' \in \mathcal{C} \\ x \neq x'}} d_N(x, x') \quad (244)$$

la maximización es sobre todos los códigos de longitud de bloque N y velocidad R .

Sea \mathcal{C}^* el código de longitud de bloque N y velocidad R que logra la distancia mínima máxima con el menor de par de códigos que tienen la distancia mínima $d(N, R)$.

Entonces

$$\begin{aligned} d(N, R) &= d_{\min}(\mathcal{C}^*) \\ &\geq d(x | \mathcal{C}^*) \text{ para toda } x \in \mathcal{X}_N \end{aligned} \quad (245)$$

Esta desigualdad surge del hecho de que si existe un $\underline{x} \in \mathcal{X}_N$ tal que $d(\underline{x} | \mathcal{C}^*) \geq d(\mathcal{C}^*)$, intercambiando \underline{x} con una palabra código en \mathcal{C}^* que logre la distancia mínima cuando se iguale con otra palabra código resulta un nuevo código con menores pares de palabras código que logran la distancia mínima, lo que contradice la definición de \mathcal{C}^* . Probando con la ecuación 245 el límite de Gilbert.

El c^* definida anteriormente es una palabra código de velocidad R y distorsión promedio $d(c^*)$ que satisface

$$\begin{aligned} d(c^*) &\hat{=} \sum_{\underline{x}} Q_N(\underline{x}) d(\underline{x}|c^*) \\ &\leq d_{\min}(c^*) \\ &= d(N, R) \end{aligned} \tag{246}$$

en esta desigualdad se usa la ecuación 245. c^* se consideró como el código bloque fuente. El teorema de codificación fuente converso dice que se debe tener

$$R \geq R(d(N, R)) \tag{247}$$

Como D está dado por $R = R(D)$ y $R(D)$ es una función decreciente de $0 < D < 1/2$ se tiene

$$d(N, R) \geq D \tag{248}$$

Al dividir entre α se obtiene el límite de Gilbert para distancias de Hamming.

Los resultados para el BSC se pueden generalizar para todos los DMC al usar la distancia de Bhattacharyya si para el parámetro s , tal que $D = D_s$, la matriz $\{e^{s d(r, r')}\}$ es positiva. Esta condición positiva es para toda $s \leq 0$ en muchos de los canales de interés.

Esto muestra que para un DMC la distancia de Bhattacharyya es una generalización de la distancia de Hamming para códigos binarios usados sobre el BSC y puede encontrarse un límite de Gilbert generalizado.

2.4 ELEMENTOS DE LA TRANSMISIÓN EN REDES

2.4.1 CONCEPTOS GENERALES

Los protocolos de las redes actuales utilizan técnicas para que el transmisor (ARQ Automatic Repeat reQuest), o alguno de los participantes cercanos, retransmitan la información que no fue recibida. Sin embargo, cuando después de cierto tiempo el receptor no puede esperar más tiempo a los paquetes faltantes, debe recobrar la mayor parte del mensaje original de los paquetes recibidos.

Este problema se agrava aún más, en sesiones multicast, donde varios receptores deben recibir el mensaje, por lo que la tasa de información está dada por el peor receptor, además como la pérdida de paquetes puede ser distinta para cada receptor, los sistemas de retransmisión resultan poco prácticos. Por lo que es necesario hacer más segura la transmisión, para ello existen códigos de protección y corrección de error [9], [22], [28], [31].

Mediante los códigos FEC (Forward Error Correction) [18], [23], se componen paquetes de tal forma que se garantiza la reconstrucción completa del mensaje original, logrando recobrar por completo el mensaje a partir de cualquier grupo de paquetes que tengan una longitud igual a la del mensaje original.

El código FEC más popular para video es el Reed-Solomon [8]. Este sistema de codificación se basa en grupos de bits, en lugar de utilizar un solo bit (0 ó 1). Matemáticamente, estos códigos se basan en la aritmética de campos finitos, que por definición mapean de un vector de espacio de dimensión m a un vector de espacio de dimensión mayor [32]. Prácticamente, este código determina un polinomio mediante la graficación de un gran número de puntos y en caso de colocar valores incorrectos aún puede recuperarse el mensaje original.

La cantidad de errores aceptables s solo es posible sí y solo sí m , la longitud del mensaje, es estrictamente menor a $N - 2s$, donde N es el número de elementos del campo finito K .

Los algoritmos de codificación más eficientes operan sobre campos finitos, lo cual provoca que los algoritmos de codificación y decodificación sean complicados, por lo que se requiere gran especialización para su diseño. Además, no son suficientemente rápidos en aplicaciones de tiempo real. El tiempo de corrida en aplicaciones de tiempo real solo es razonable cuando el número de paquetes redundantes es pequeño (cuando mucho 100 paquetes), lo cual se debe a que el tiempo de codificación y decodificación es proporcional a la longitud de la codificación por el número de paquetes.

Para solucionar el tiempo de corrida se desarrollaron los códigos tornado (TC) [32], los cuales deben recibir más paquetes codificados que el número ideal (número de paquetes del mensaje) para decodificar el mensaje.

Los TC puede realizar la codificación y decodificación en un tiempo proporcional a la longitud de la codificación por una pequeña constante independiente del número de paquetes redundantes. Dicha constante es el tiempo de encabezado, cuyo valor se encuentra alrededor de seis.

Los TC tienen una longitud de encabezado diminuto y un tiempo de codificación y decodificación muy pequeños.

Sin embargo, tanto los FEC como los TC solo manejan un nivel de prioridad para todo el mensaje que será transmitido, lo cual solo los hace efectivos cuando las pérdidas se dispersan levemente.

En un ambiente de transmisión de video sobre una red homogénea existen diversos métodos que utilizan diversos niveles de prioridad para la codificación de los paquetes, entre los cuales están los siguientes:

Shacham [25] sugiere un método para enviar mensajes priorizados sobre las redes, el objetivo es dividir el mensaje en distintos niveles de prioridad y usar un canal diferente para enviar cada nivel. Cada receptor deberá unir, de acuerdo a su prioridad, tantos de estos canales como le sea posible. Este método requiere el cómputo de la capacidad de los canales entre el emisor y el receptor, lo cual es poco práctico en largas redes con capacidades que varían rápidamente debido al congestionamiento. Además, no maneja la pérdida de paquetes.

Otro método con priorización es el sistema PET (Priority Encoding Transmission) [2], [3], [27] desarrollado por el ICSI (International Computer Science Institute).

PET asigna la prioridad para cada parte del mensaje, con base en las prioridades, el sistema codifica el mensaje en conjuntos de paquetes para su transmisión y los envía a los receptores, asegurando la recuperación de los paquetes de acuerdo a la prioridad [5], [30].

La prioridad asignada a cada parte del mensaje determina la cantidad de paquetes suficientes para recobrar esa parte. Por lo que aún si algunos de los paquetes codificados se pierden, el receptor es capaz de recobrar el mensaje si recibe suficientes paquetes codificados. El sistema garantiza que un segmento puede decodificarse de cualquier subgrupo de paquetes siempre y cuando la fracción de paquetes del subgrupo sea al menos igual al valor de la prioridad del segmento.

Para proteger MPEG [4] usando un sistema PET la priorización se hace sobre los distintos tipos de tramas (I, P, y B), lo cual define una prioridad natural: las tramas I son las más importantes, seguidos de las tramas P y finalmente las tramas B.

PET presenta las siguientes ventajas:

- puede utilizarse con cualquier hardware o protocolo de switcheo de paquetes
- mejora la calidad de la transmisión de paquetes
- como no utiliza mucho encabezado se reduce el costo de transmisión por el canal
- tiene aplicaciones en multimedia y redes de alta velocidad

Sin embargo, a pesar de todas las ventajas que presenta, no provee corrección de errores y además también utiliza campos finitos para la codificación, lo que complica su diseño [16].

Sin embargo, todas estas técnicas incrementan el tráfico en la red, de mayor o menor forma, causado por la información redundante que se agrega al mensaje original, lo cual implica mayor trabajo tanto para el receptor como para el transmisor.

3. CODIFICACIÓN FUENTE-CANAL PARA LA TRANSMISIÓN TOLERANTE A PÉRDIDAS EN REDES HETEROGÉNEAS

De lo anterior podemos observar que las técnicas que trabajan con base en la tolerancia a pérdidas se basan fundamentalmente en consideraciones de codificación canal. El propósito de la codificación fuente-canal es explotar las características entrópicas de la señal y las características estadísticas de los canales de comunicación.

3.1 TRANSMISIÓN PROGRESIVA

La transmisión progresiva [15], [19] es la transmisión en baja resolución de una imagen, que se refina sucesivamente para aumentar la resolución de la imagen, con lo cual se reduce la cantidad de datos enviados y entonces el ancho de banda requerido, es por esto que es muy útil para la comunicación de imágenes, en particular de aquellas que requieren examen rápida y que son transmitidas en redes no homogéneas.

Sin embargo, la transmisión progresiva es problemática en canales ruidosos. Esto se debe a que los codificadores progresivos emplean codificación adaptable de longitud variable, donde el significado de los “últimos” bits depende de la decodificación de los bits “anteriores”.

La información se guarda en varios estados del modelo del canal (el cual se explicará más adelante). Entonces, un bit de error modifica, con alta probabilidad, el valor de estos estados, por lo que la imagen decodificada divergirá de la imagen original.

Debido a lo anterior, se requieren canales de alta calidad para la transmisión de los datos, sin embargo, ésta es una característica que no se poseen las redes y que implementarla produciría un alto costo. Es por ello que han surgido varios métodos para combatir el ruido del canal a través del control del error de codificación para los sistemas de transmisión de imágenes.

Uno de estos métodos son las técnicas de codificación espacial que se requieren tanto para la transmisión de imágenes sin movimiento como para la transmisión de imágenes con codificación periódica intratrama con variación en el tiempo, es decir, el video. Los métodos de codificación que utilizan estas técnicas son los métodos de transformadas y los métodos de codificación predictiva.

Los métodos de transformada, como ya fue definido anteriormente, rotan un vector señal alrededor del origen de un espacio de $N \times N$ dimensiones. La desventaja de éstos métodos es que son propensos a efectos de bloque y “ruido mosquito” en bloques que contienen bordes de alto contraste.

El método de codificación predictiva elimina las redundancias que existen entre los pixeles, codificando únicamente la información nueva entre el bloque actual y el anterior. Sin embargo, la codificación predictiva DPCM no es muy eficiente a bajas velocidades debido al error de cuantificación de retroalimentación. Esto puede sobrellevarse combinando el VQ (Vector Quantization) [21] con la codificación predictiva. El VQ consiste en descomponer la imagen en vectores, que contienen información sobre los pixeles, coeficientes de transformación y otros elementos propios de la transmisión de la imagen; estos vectores imagen se comparan contra vectores código y se codifican con el más adecuado.

Un método de codificación predictiva con VQ es el análisis por síntesis, que consiste en procesar cada vector código del código a través del filtro de síntesis con el fin de producir un vector imagen reconstruido, el vector código seleccionado es aquél que tenga mínima distorsión. La dirección de éste vector código en el código se transmite al decodificador, y no todo el código.

Otro método de codificación predictiva con VQ es el 2-D-CELP (2-D Code Excited Linear Prediction), extensión del método para la codificación de lenguaje, y emplea una mejora del método de análisis por síntesis. El filtro de síntesis del sistema 2-D-CELP es un filtro con

predicción de error inverso que emplea un predictor adaptable de interrupciones. En el codificador se realiza la adaptación del predictor escalar con cada bloque y los coeficientes de la predicción se cuantifican antes de transmitirse hacia el decodificador.

3.2 SISTEMA DE CODIFICACIÓN 2-D-CELP

Este esquema de codificación 2-D-CELP, emplea un arreglo finito de K predictores lineales H_k , con región 2-D de soporte F . En la predicción adaptable de bloques, se selecciona un predictor en el codificador para un bloque de muestras de la imagen, la información requerida para especificar este predictor se envía como información secundaria.

La imagen de entrada $x(m,n)$ es una matriz de dimensión $M_1 \times M_2$, donde m y $n \in Z^+$. Esta matriz se divide en bloques disjuntos $B_i = \{(o_{i1}, o_{i2}) + (b_1, b_2) : (b_1, b_2) \in B\}$ donde (o_{i1}, o_{i2}) es el origen del i -ésimo bloque y B es un arreglo de tamaño L que determina la geometría del bloque. El i -ésimo bloque de pixeles se denota como $\mathbf{X}_i = \{x(m,n) : (m,n) \in B_i\}$ (intensidad).

El diagrama de bloque del codificador se muestra en la Figura 24. El codificador debe seleccionar el índice del predictor k_i y el vector residual \mathbf{e}_i con el objetivo de minimizar el error de codificación $d(X_i, \hat{X}_i)$, donde d es una medida apta de distancia. El predictor se selecciona con base en la señal de entrada únicamente, para minimizar el error cuadrado de predicción para el bloque.

El decodificador 2-D-CELP se muestra en la Figura 25. Cada bloque imagen B_i tiene asociada una palabra código transmitida \mathbf{u}_i , de la que se deriva el índice del predictor k_i y el vector de residuos $\mathbf{e}_i = \{e_i(b_1, b_2) : (b_1, b_2) \in B\}$, seleccionados del código, para manejar al decodificador.

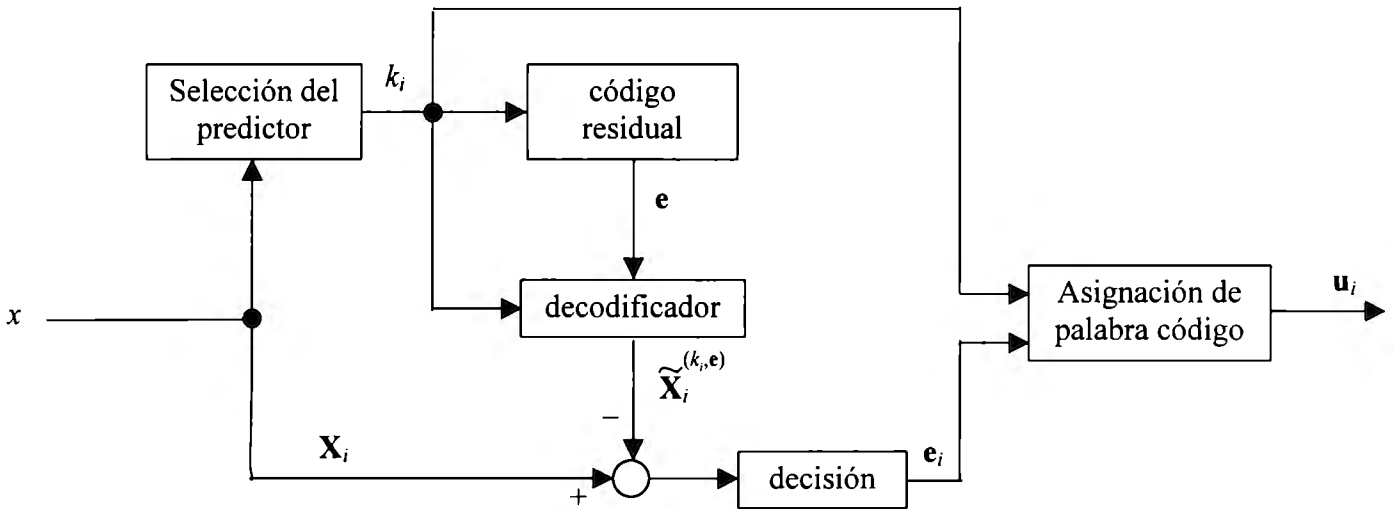


Figura 24. Codificador 2-D-CELP

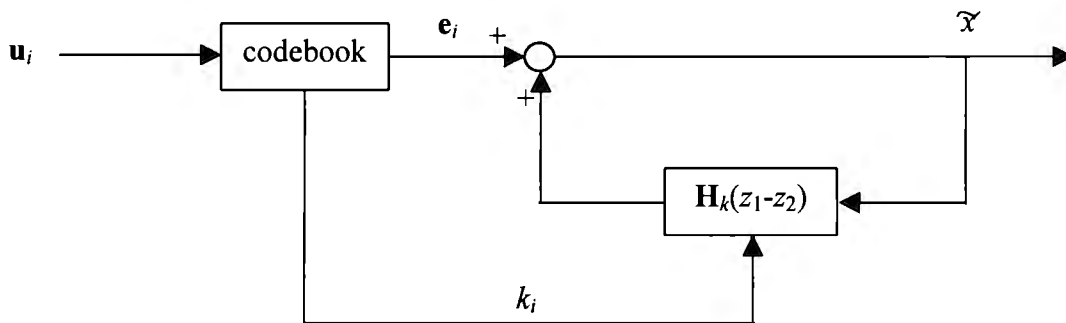


Figura 25. Decodificador 2-D-CELP.

El residuo cuadrado total para el bloque B_i se define como $r_i(H)$, usando el predictor H (ecuación 249). Por lo que $k_i = \arg \min_k r_i(H)$.

$$r_i(H) = \sum_{(m,n \in B_i)} \left[x(m,n) - \sum_{(p,q \in F)} h(p,q)x(m-p,n-q) \right]^2 \quad (249)$$

Enseguida se escoge el vector residual \mathbf{e}_i que minimice el error de codificación empleando el predictor k_i :

$$\mathbf{e}_i = \arg \min_{\mathbf{e} \in C_k} d(X_i, \hat{X}_i^{(k, \mathbf{e})}) \quad (250)$$

donde

$$\hat{x}^{(k, \mathbf{e})}(m, n) = \sum_{(p, q) \in F} h_k(p, q) \hat{x}^{(k, \mathbf{e})}(m - p, n - q) + e(m - o_{i1}, n - o_{i2}), (m, n) \in B \quad (251)$$

El diseño del código residual del sistema 2-D-CELP se basa en el método de agrupación sucesiva, en el cual la agrupación central se actualiza cada vez que los datos se toman de la secuencia de entrenamiento.

En el 2-D-CELP con vector código de tamaño fijo y K predictores, el código consiste de K subcódigos C_k , $k = 1, \dots, K$ de tamaño igual o diferente, y el algoritmo consiste en actualizar un vector código del código cada vez que un bloque de la secuencia de entrenamiento sea codificada.

Velocidad de encabezado

Para la transmisión de los datos es necesario cierto encabezado, como la protección del canal varía en el tiempo, el decodificador de canal necesita saber como decodificar el flujo en cada punto del tiempo. Para limitar el impacto de la velocidad del encabezado, se toman en consideración los siguientes hechos:

- a) en el sistema solo existe un número finito de códigos canal disponibles
- b) el código canal es constante sobre cada bloque
- c) la protección de error de canal no se incrementa uniformemente

El header de la cadena comienza con el número de bloques, seguido del índice de la primera velocidad código, y de la posición de las transiciones.

Las transiciones T_i 's se representan con la distancia a partir del final del flujo, debido a que éstas son más densas en esa parte. Como $T_i \geq T_{i-1}$, el número de bits necesarios para especificar T_i no es mayor a $\lceil \log_2 (T_{end} - T_i) \rceil$.

3.3 CANALES CON DESVANECIMIENTO

El canal inalámbrico con desvanecimiento puede modelarse como un proceso de estado finito de Markov. Un modelo común es el modelo de dos estados de Gilbert-Elliott, como se muestra en la Figura 26. En cada estado, el canal se describe como un BSC con un BER (Bit-Error Rate), e_b denotando el estado malo y e_g al estado bueno. El modelo de Markov se especifica mediante las probabilidades de transición P_{gb} y P_{bg} .

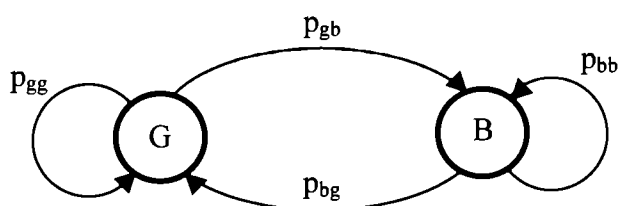


Figura 26. Modelo de Gilbert-Elliott

Como no se puede conocer el estado instantáneo del canal y el BER está variando, debe diseñarse un BER ϵ_d , bajo las siguientes consideraciones: un ϵ_d alto produce mayor protección de error de canal, pero menos bits disponibles para la codificación fuente, y viceversa. Pero si la protección de canal es inadecuada el sistema fracasa, y el decodificador no puede recobrar de los errores, aunque puede detectarlos. Lo cual motiva un método conservador para la protección contra errores en los canales con variación en el tiempo.

Algoritmo diseñador de predictores

El arreglo de los K predictores se seleccionan con base en el análisis del arreglo de entrenamiento. Se utiliza un método iterativo de agrupación [25] para el diseño de los predictores, comenzando a partir de un arreglo inicial de predictores. Para cierto arreglo de predictores, cada bloque del arreglo de entrenamiento se asigna a la clase del predictor que produzca el menor error cuadrado total de predicción para ese bloque. Para cierta clasificación, se encuentra un arreglo de predictores que minimice el error cuadrado total de predicción para todos los bloques dentro de cada clase. Este proceso se continúa iterativamente hasta lograr la convergencia. En cada iteración, la minimización es un problema de mínimos cuadrados.

Elementos de diseño

- La máscara de predicción se escoge de tal forma que $\hat{x}(m,n)$ sea causalmente calculable a partir de bloques reconstruidos previamente. En la Figura 27 se muestra una máscara de predicción apropiada, asumiendo que el cálculo procede repasando línea por línea. P_1 , P_2 , y Q deben escogerse de tal forma que se obtenga una buena predicción en una amplia variedad de imágenes.

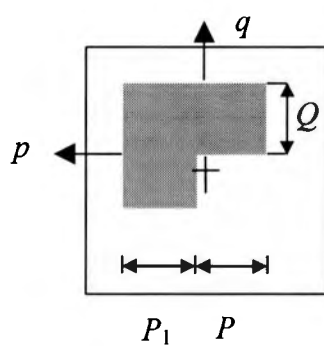


Figura 27. Geometría de un predictor 2-D

- La condición anterior impone la relación entre la geometría del bloque B y la máscara de predicción. La Figura 28 muestra formas adecuadas de bloques de 16 elementos para $P_2 = 1$, 2, y 3 que aseguran el cálculo causal.

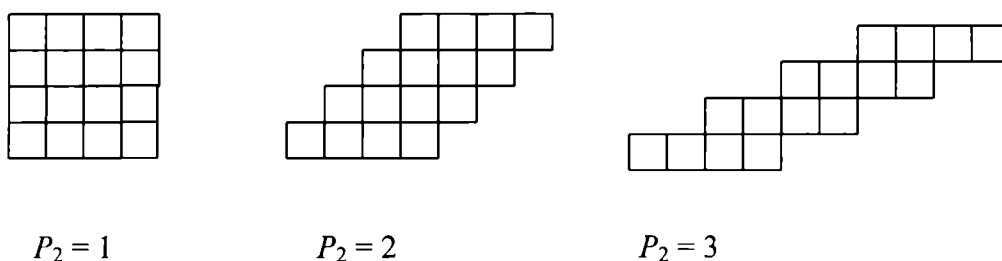


Figura 28. Formas de bloque que aseguran el cálculo causal para $P_2 = 1, 2, 3$

- Como el índice del predictor debe transmitirse para cada bloque, un bloque de tamaño grande minimiza el encabezado. Como la razón para usar predicción adaptable es responder a los cambios en la estructura local, el tamaño del bloque debe ser suficientemente pequeño para lograr esto. Es por ello que el tamaño del bloque se escoge experimentalmente con base en el desempeño del código.
- El número de predictores también afecta el encabezado requerido para transmitir el índice de predicción. Para adaptar diferentes tipos de estructuras locales se requiere un mayor número de predictores. La selección del número de predictores también se basa con el desempeño del código.
- La elección de los predictores iniciales es importante, ya que una elección pobre puede resultar en la convergencia de un óptimo local. Los predictores iniciales se escogen con base en el supuesto de que cada bloque contiene ninguna direccionalidad específica o una orientación dominante. Por lo que cada uno de los predictores iniciales favorece una de las orientaciones mostradas en la Figura 29.

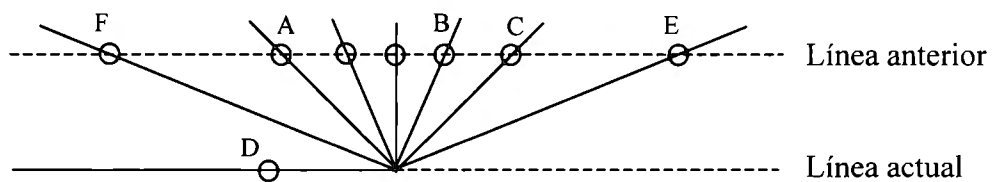


Figura 29. Orientación de los predictores iniciales

3.4 2-D-CELP CON TAMAÑO DE BLOQUE VARIABLE

Las imágenes más naturales contienen regiones con muchos detalles, así como regiones con pocos detalles. Las regiones con pocos detalles son casi homogéneas. Mientras que los segmentos con muchos detalles se caracterizan por bordes de orientaciones diferentes y largas transiciones en los niveles de gris. Por lo que el decodificador 2-D-CELP necesita más información para los

bloques que pertenecen a estas regiones para poder reconstruirlas adecuadamente. Lo que sugiere el uso de bloques más grandes para el cuantificador de vectores en áreas homogéneas y bloques más pequeños en áreas más detalladas.

Con base en esta observación, se propone un sistema 2-D-CELP con tamaño de bloque variable VQ. El tamaño del bloque del predictor de interrupciones es igual al tamaño del bloque más grande VQ. La decisión de la subdivisión del bloque cuantificado se basa en el análisis de la predicción de error. Usar n diferentes tamaños de bloque implica el diseño de más de un código. A cada código C_i le corresponde un tamaño de bloque específico y se divide en K subcódigos, donde K es el número de predictores usados: $C_i = \bigcup_k C_{ik}, i = 1, \dots, n$.

En la Figura 30 se muestra el diagrama de flujo del algoritmo. La operación del método se ilustra con un sistema que utiliza bloques de tres tamaños. Primero se divide la imagen en bloques disjuntos $B1_i$ que consisten de $L1$ pixeles. Una vez que se selecciona el predictor se obtiene el residual cuadrado total mínimo, $r1_i(H_k)$, y se compara con el umbral Δ_1 . Si $r1_i(H_k)$ es menor que Δ_1 , $B1_i$ se codifica con el 2-D-CELP con el subcódigo $C1_{k_i}$. Si $r1_i(H_k)$ es mayor que Δ_1 , $B1_i$ se subdivide en cuatro bloques iguales $\{B2_{ij}\}_{j=1}^4$ de tamaño $L2$. Para cada bloque $B2_{ij}$ se calcula el residual cuadrado total $r2_{ij}(H_k)$ y se compara con el umbral Δ_2 . Cada subbloque $B2_{ij}$ se codifica con 2-D-CELP empleando el subcódigo $C2_{k_i}$ o se subdivide en cuatro bloques $\{B3_{ijl}\}_{l=1}^4$ de tamaño $L3$, usando en cada codificación el subcódigo $C3_{k_i}$, de acuerdo al resultado de la comparación con el umbral Δ_2 .

La información de como un bloque $B1_i$ fue subdividido se transmite como información secundaria. Un bloque debe subdividirse solo si este no puede codificarse con suficiente precisión en el tamaño dado. Debido al costo de codificación, la decisión de subdivisión se basa en la predicción de error. Los valores de los umbrales se basan en la determinación empírica de la relación entre el residual de error de predicción cuadrado y el error de codificación; seleccionando los umbrales que minimizan el valor PSNR (Peak Signal-to-Noise Ratio). Lo cual se logra utilizando ACELP (Adaptive CELP) [13], el cual realiza la tabulación porcentual a nivel del código considerando que el filtro de síntesis no requiere procesar todas las secuencias, con lo cual se logra la codificación adaptable y reducir los cálculos que el algoritmo debe realizar.

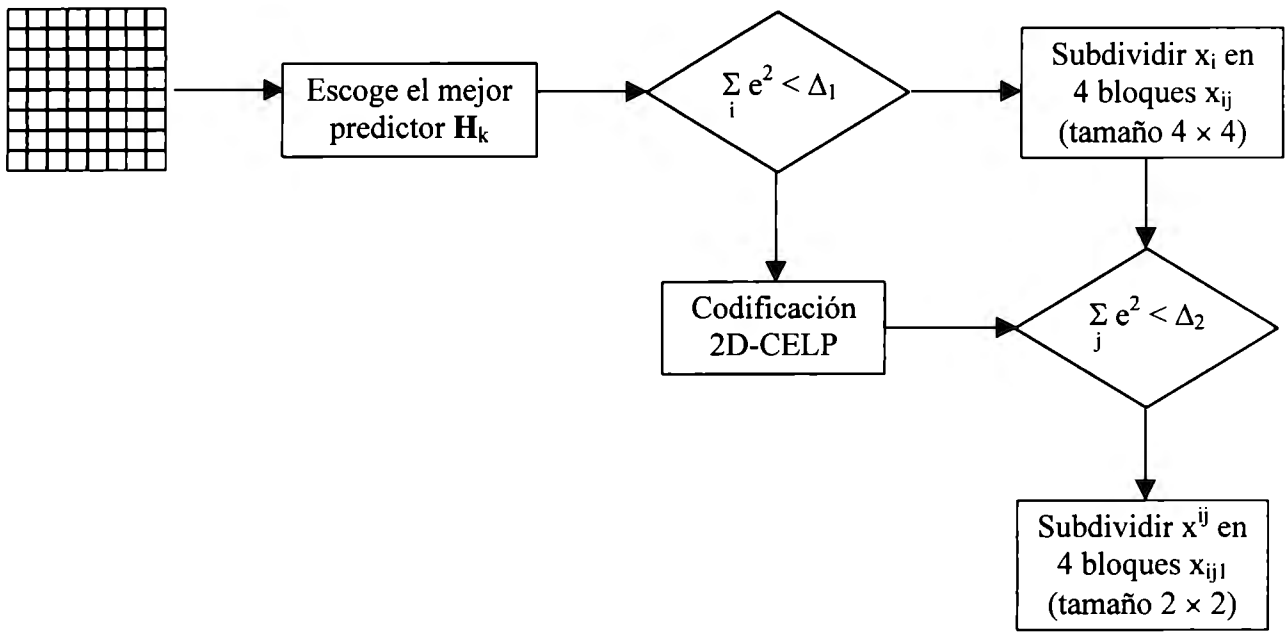


Figura 30. Diagrama de Flujo de la codificación VB 2-D-CELP

CONCLUSIONES

El objetivo de codificar el video es cumplir con los requerimientos de los medios de comunicación: velocidad de muestreo, cuantificación y velocidad de datos (Tabla 1 y 2). Para cumplir con estos requerimientos se han desarrollado métodos de transformadas, cuantificación, codificación intertrama, y predicción e interpolación con predicción de movimiento.

Como resultado de estas investigaciones se logró producir el estándar JPEG, para imágenes sin movimiento; y el estándar MPEG, para imágenes que varían en el tiempo, es decir, video.

En el Capítulo 2 se muestran varias acotaciones teóricas para códigos bloque y convolucionales en codificación fuente-canal. Con base en estos límites o acotaciones se desarrollan diversas técnicas para la transmisión de información, con la finalidad de obtener un umbral de codificación que permita reconstruir apropiadamente la imagen, sin pérdida de información sustantiva.

Para finalizar este capítulo, se presentaron diversos tipos de transmisión que trabajan con consideraciones de codificación canal, a partir de lo cual surge el planteamiento de emplear un método de codificación que utilice las ventajas de la codificación fuente-canal considerando las acotaciones desarrolladas, y que sea aplicable a la transmisión de video.

El método propuesto es un sistema de transmisión tolerante a pérdidas en redes heterogéneas, éste último representa una gran ventaja debido a que actualmente las redes presentan distintos anchos de banda, velocidad y capacidad.

Este sistema de codificación fuente-canal se obtiene utilizando la codificación progresiva (codificación canal) junto con el método 2-D-CELP con tamaño de bloque variable (codificación fuente) que es una extensión del CELP usado para la codificación del lenguaje.

La codificación progresiva emplea codificación adaptable de longitud variable, es por ello que cuando ocurre un error en alguno de los estados del modelo de Gilbert-Elliott que modela el canal como un proceso de estado finito de Markov, la imagen transmitida es distinta de la original.

Entonces surge la problemática de elegir algún tipo de codificación mediante la cual puede reconstruirse la señal apropiadamente, ya que los canales de transmisión que se emplean no son libres de errores.

La técnica propuesta, como ya se mencionó, es el método 2-D-CELP, el cual es un método de interpolación lineal que emplea un filtro de síntesis de predicción de error inverso que cuantifica a los coeficientes de predicción antes de transmitirlos, junto con un predictor adaptable de interrupciones, que adapta el predictor escalar a cada bloque.

Gracias a estas características de procesamiento se logra un sistemas adaptable a los requerimientos de la imagen, es decir, logra codificar adecuadamente regiones muy detalladas y regiones con pocos detalles.

Para lograr lo anterior se requieren ciertos elementos específicos de diseño: máscara de predicción, geometría del bloque, tamaño del bloque, número de predictores y predictores iniciales, que además de cumplir con los requerimientos necesarios para responder adecuadamente a cambios en la estructura local de la imagen, debe considerar la información secundaria requerida para especificar estos elementos (tamaño del bloque y número de predictores).

El proceso de asignación del predictor a un bloque de la imagen se basa en el criterio de mínimos cuadrados con el fin de minimizar el error cuadrado de predicción para el bloque en proceso, asignando la palabra código, que contiene el índice del predictor y el vector de residuos asignados al bloque.

Como característica adicional, se propone el uso de CELP adaptable (ACELP), con el fin de no procesar todas las secuencias candidatas a través del filtro de síntesis, con lo que se logra reducir la complejidad del algoritmo.

De esta manera, este trabajo representa un conjunto de consideraciones teóricas, sobre los aspectos de codificación fuente-canal de señales de video para la transmisión tolerante a pérdidas en redes heterogéneas. Los aspectos de implementación y experimentación pueden ser realizados en trabajos posteriores.

BIBLIOGRAFÍA

- [1] AÍSA SONIA, DUBOIS ERIC. 2-D-CELP Image Coding with Block-Adaptive Prediction and Variable Code-Vector Size. *IEEE Transactions on Image Processing*, Feb. 1996, pp. 369 – 374
- [2] ALBANESE, ANDRES; BLÖMER, JOHANNES; EDMONDS, JEFF; LUBY MICHAEL. Priority Encoding Transmission. *Proc. 35th Symposium on Foundations of Computer Science*, 1994
- [3] ALBANESE, ANDRES; BLÖMER, JOHANNES; EDMONDS, JEFF; LUBY MICHAEL. Priority Encoding Transmission. *IEEE Transactions on Information Theory*, 1996, Vol. 42, No. 6
- [4] ANDRONICO; LOMBARDO; et al. Performance Analysis of Priority Encoding Transmission of MPEG Video Streams. *Instituto di Informatica e Telecomunicazioni*, Facolta' di Ingegneria, Università di Catania
- [5] ASCIA; CATANIA; FICILI. Design of a VLSH Hardware PET Decoder. *Instituto di Informatica e Telecomunicazioni*, Università di Catania
- [6] AUTOR STEVE. Video Conference and Distance Learning. The Need for Convergence, October, 1997
- [7] CHIARIGLIONE LEONARDO. MPEG and Multimedia Communications. *CSELT*
- [8] CIPRA, BARRY. The Ubiquitous Reed-Solomon Codes. *SIAM News*, January 1993, Vol. 26, No. 1

- [9] DE CANNE, BART. Error-Recovery Codes. *Dr. Dobb's Journal*, December 1994, pp. 44 - 50
- [10] FISCHER, THOMAS; MARCELLIN MICHAEL. Joint Trellis Coded Quantization/Modulation, *IEEE Transactions on Communications*, February 1991, Vol. 39, No. 2
- [11] FORNEY, DAVID Jr. The Viterbi Algorithm. *Proceedings of the IEEE*, March 1973, Vol. 61, No. 3
- [12] FOURCHER ADRIANA. What is videoconferencing?. *Conference Club, Inc*, 1997
- [13] GALAND, CLAUDE; MENEZ, JEAN; ROSSO, MICHELE. Adaptive Code Excited Predictive Coding, *IEEE Transactions on Signal Processing*, June 1992, pp. 1317 – 1326.
- [14] GERSHO, ALLEN. Advances in Speech and Audio Compression. *Proceedings of the IEEE*, June 1994, Vol. 82, No. 6
- [15] GONZALEZ, RAFAEL; WOODS, RICHARD. *Digital Image Processing*. Addison-Wesley Publishing Company, June 1992.
- [16] GUENZEL CHRISTOPH. Erasure Resilient Transmission and Priority Encoding. *Institute for Computer Science, Rheinische Friedrich-Wilhelms-Universitaet Bonn*
- [17] LE GALL, D. MPEG: A video compression standard for multimedia applications, 1991
- [18] LUBY, MICHAEL; MITZENMACHER, MICHAEL; et al. Practical Loss-Resilient Codes. *29th ACM Symposium on Theory of Computing*, 1997
- [19] LU, JIN; NOSRATINIA, ARIA; AAZHANG, BEHNAAM. Progressive Source-Channel Coding of Images over Bursty Error Channels
- [20] OMURA, JIM K. Source and Channel Coding with Block and Convolutional Codes, *System Science Department, University of California*

- [21] POPESCU, ANDREI; MOREAU, NICOLAS. CELP Coding Using Trellis-Coded Vector Quantization of the Excitation. *IEEE Transactions on Speech and Audio Processing*, November 1995, Vol. 3, No. 6
- [22] RIZZO, LUIGI. Effective Erasure Codes for Reliable Computer Communication Protocols. *Dip. Di Ingegneria dell'Informazione*, Università di Pisa
- [23] RIZZO, LUIGI. Software FEC in computer communications. *Dip. Di Ingegneria dell'Informazione*, Università di Pisa
- [24] SANTANA, GUSTAVO. Intranets para el desarrollo de la educación a distancia. *Memorias del XIV Simposio Internacional de Computación en la Educación*. Noviembre, 1998. Morelos, México. p. 269 – 272
- [25] SHACHAM, N. Multicast Routing of Hierarchical Data. *Proceedings of ICC '92*, 1992
- [26] SHINOHARA, K.; MINAMI, T. Encoding of still pictures with successive clustering and predictive vector quantization, *Electron. Commun. Japan*, 1988, vol. 71, no. 6, pp. 98-110.
- [27] _____, Description of PET. *International Computer Science Institute*, Berkeley,
- [28] _____, Error Control Schemes. *International Computer Science Institute*, Berkeley, California
- [29] _____, Growth and Change of the Internet. *International Computer Science Institute*, Berkeley, California
- [30] _____, More about PET. *International Computer Science Institute*, Berkeley, California
- [31] _____, Networks Group Semianual Report. *International Computer Science Institute*, 1996

[32] _____, Standard FEC Codes: A Technical Overview. *MagiCast*, California

[33] ISO11172