

01170
2ej. 5

DISEÑO DE UN DECODIFICADOR DE VITERBI

ROBERT HENRY MORELOS-ZARAGOZA ASCANIO

TESIS

Presentada a la División de Estudios de
Posgrado de la
FACULTAD DE INGENIERIA
de la
UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
como requisito para obtener
el grado de
MAESTRO EN INGENIERIA
(COMUNICACIONES)

CIUDAD UNIVERSITARIA 8 DE ENERO DE 1987.

TESIS CON
FALLA DE ORIGEN



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

"DISEÑO DE UN DECODIFICADOR DE VITERBI"

INDICE

AGRADECIMIENTOS

INDICE

RESUMEN

CAPITULO 1. INTRODUCCION

- 1.1. SISTEMAS DE COMUNICACION POR SATELITE.
- 1.2. CODIFICACION DE CANAL.
- 1.3. TECNICAS DE CODIFICACION.
- 1.4. CONSIDERACIONES PRACTICAS.

CAPITULO 2. CODIGOS CONVOLUCIONALES.

- 2.1. EL CODIFICADOR CONVOLUCIONAL.
- 2.2. ESTRUCTURA Y FORMAS DE REPRESENTACION.
- 2.3. MEDIDAS DE DISTANCIA.

CAPITULO 3. DECODIFICACION DE CODIGOS CONVOLUCIONALES.

- 3.1. INTRODUCCION.
- 3.2. DECODIFICACION SECUENCIAL.
 - 3.2.1. ALGORITMO DE FAND.
 - 3.2.2. ALGORITMO DE ZIGANGIROV-JELINEK.
 - 3.2.3. ALGORITMO DE PILA MULTIPLE.
- 3.3. DECODIFICACION DE VITERBI.
- 3.4. PROBABILIDAD DE ERROR Y GANANCIA DE CODIFICACION.
- 3.5. ESFUERZO DE CALCULO Y RETRASO DE DECODIFICACION.

CAPITULO 4. DISEÑO DE UN DECODIFICADOR DE VITERBI.

- 4.1. CONSIDERACIONES DE DISEÑO.
- 4.2. ALTERNATIVAS DE IMPLEMENTACION.
- 4.3. DESCRIPCION DEL DECODIFICADOR.
- 4.4. REQUERIMIENTOS DE FUNCIONAMIENTO.

CAPITULO 5. CONCLUSIONES Y TRABAJO A DESARROLLAR.

BIBLIOGRAFIA.

"DISEÑO DE UN DECODIFICADOR DE VITERBI"

RESUMEN

Esta tesis presenta el diseño de un decodificador de Viterbi, para trabajar con tasas de información de hasta 16000 bits por segundo. La idea de realizar este trabajo, nació con la necesidad de buscar un esquema idóneo de codificación de canal como parte del diseño de una estación terrena de comunicaciones vía satélite de voz, datos y video, en la banda Ku.

Como resultado de un estudio preliminar de diversas técnicas de codificación y decodificación de canal, se determinó que el uso de códigos convolucionales con longitudes cortas de restricción y decodificación de Viterbi, era lo más adecuado para esta aplicación. Específicamente, se encontró que el código convolucional de tasa $1/2$, longitud de restricción $K = 7$, con decodificador de Viterbi de decisión dura, es el que proporciona el mejor desempeño.

La tesis presenta, en su primera parte, una panorámica general de la teoría de los códigos convolucionales y de sus esquemas de decodificación más prácticos, atendiendo a aspectos importantes para evaluar su desempeño como la probabilidad de error, ganancia de codificación y retraso de decodificación. Se describe funcional y físicamente el decodificador diseñado, discutiéndose además las diversas alternativas de su implementación, y los requerimientos de funcionamiento.

CAPITULO 1. INTRODUCCION.

En este capítulo se exponen los principios de la codificación de canal, explicando brevemente conceptos básicos sobre las comunicaciones por satélite y las diversas técnicas de codificación de canal. Se presentan también algunas consideraciones prácticas sobre la aplicación de los códigos correctores y/o detectores de errores en los sistemas de comunicaciones.

1.1. SISTEMAS DE COMUNICACION POR SATELITE.

Los canales de satélite permiten el uso económico de la codificación con el objeto de mejorar la eficiencia de la comunicación. Los disturbios que se presentan en estos canales tienen características esencialmente aditivas y pueden modelarse mediante ruido Gaussiano, lo suficientemente "blanco" como para ser independiente entre los intervalos de duración de un bit. En muchas clases de canales de satélite existe suficiente ancho de banda como para permitir una expansión moderada del mismo.

La comunicación por satélite presenta problemas que no son tan críticos en los sistemas terrestres de comunicación. El más obvio de éstos es el alto costo de la potencia a bordo del satélite. Es deseable diseñar un sistema que logre que la potencia radiada por el satélite, y el peso del equipo asociado a ésta, sea mínima.

El diseño de un enlace de datos por vía satélite consiste en un balance entre ganancias y pérdidas. El enlace está compuesto de una cantidad de recursos de transmisión y recepción, fuentes de ruido, y

atenuadores de la señal, medidos desde el modulador y el transmisor, pasando por el canal, hasta llegar al receptor y demodulador. Ver figura 1.1.

El medio de propagación, o trayectoria electromagnética, que conecta al transmisor con el receptor se conoce como canal. El concepto de espacio libre supone una región de canal libre de cualquier objeto que pudiera afectar la propagación de energía por absorción, reflexión o refracción.

Las estaciones terrenas pequeñas que operan con satélites de servicio fijo ofrecen ventajas inherentes al problema de la comunicación de datos [17]. Resuelven el problema de enlaces largos de manera que se tiene una independencia de la distancia, y permiten tasas relativamente altas a un costo muy bajo. Las frecuencias de la banda Ku (12/14 GHz) están virtualmente dedicadas a servicios fijos de comunicación por satélite, y por lo tanto están libres de interferencia terrestre. En cualquier enlace de comunicaciones existen únicamente dos estaciones terrenas y un satélite, en comparación a los sistemas terrestres de larga distancia que generalmente incluyen varias decenas de repetidores entre los extremos del enlace.

La radiación electromagnética en la banda Ku sufre una gran atenuación debido a los gases atmosféricos, la lluvia y partículas sólidas, así como una temperatura de ruido en el receptor mayor a la de la banda C (4/6 GHz) debido a la mayor atenuación.

Como medida del funcionamiento del enlace en sus diversos puntos, se emplea la relación señal-a-ruido, SNR. Este parámetro se

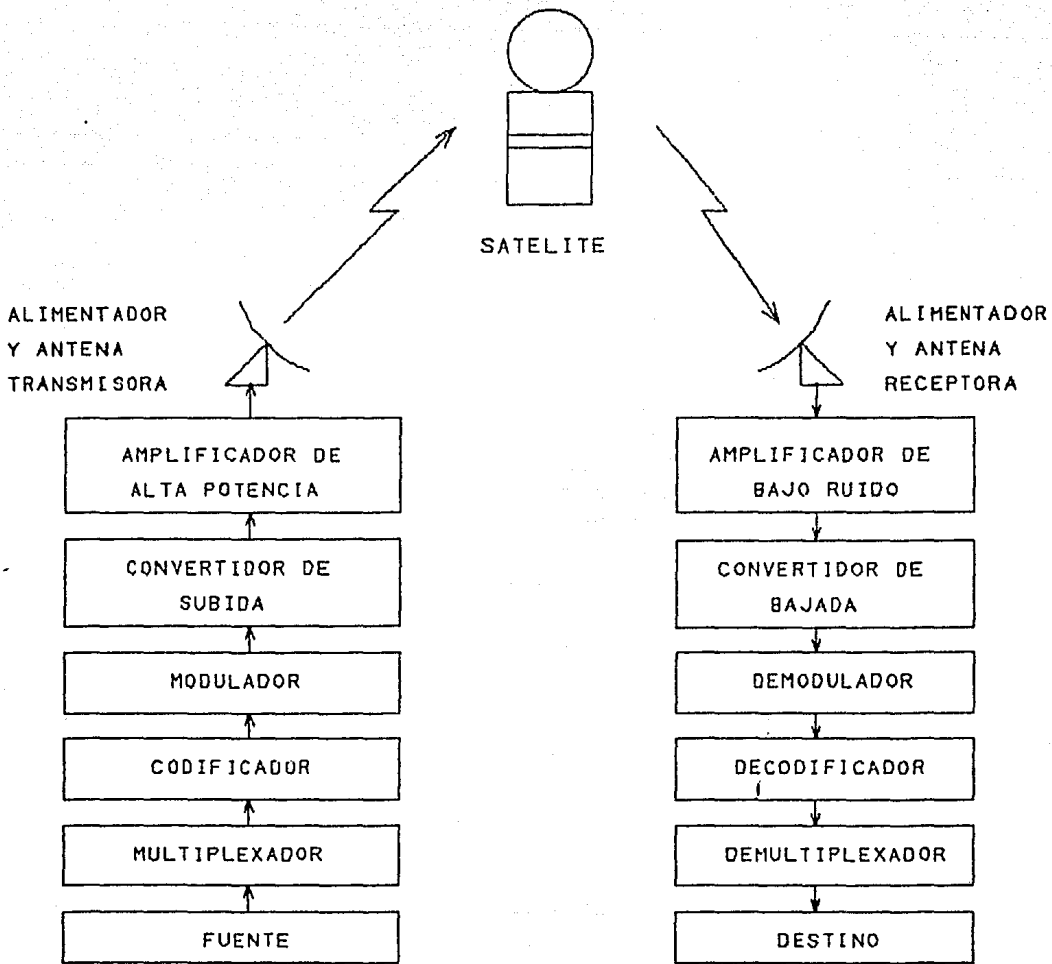


FIGURA 1.1. ESQUEMA DE UN ENLACE DE COMUNICACION DE DATOS POR SATELITE.

definió como la relación de la potencia de la señal que se desea transmitir a la potencia de la señal que interfiere. Existen dos tipos básicos de interferencia en una comunicación por satélite: por atenuación de la señal transmitida, y por un incremento de la señal que interfiere. Esto último es conocido como ruido. Existen dos tipos de ruido: el ruido térmico y el ruido de intermodulación, principalmente.

Una medida útil de la eficiencia de un sistema de comunicación de datos por satélite es la relación de energía por bit recibido-a-densidad de energía del ruido, $(E_b/N_0)_r$, necesaria para obtener una tasa específica de bits en error (BER) del sistema. Dicha tasa está determinada por una combinación de sistemas de modulación y codificación.

Los elementos básicos de una red de enlaces por satélite, empleando estaciones terrenas pequeñas, son tres: la estación terrena remota, el satélite geoestacionario (con un repetidor no regenerativo), y una estación terrena central.

En la dirección de llegada, de la estación terrena remota a la central, la baja potencia isotrópica radiada efectiva (PIRE) generada por la estación remota ilumina débilmente al repetidor del satélite, lo que resulta en un haz débil en el enlace de bajada a la estación central. Una figura de mérito (G/T) alta en la estación terrena receptora ayuda a resolver el problema de la recepción de una señal débil.

Las ecuaciones que gobiernan un enlace de comunicación por

satélite son:

$$(C/No)_u = (P_{et} + G_a) - P_u - P_{au} + (G/T)_s - k \quad [dB-Hz] \quad (1.1)$$

$$(C/No)_d = (E_s - B_{0s}) - P_d - P_{ad} + (G/T)_{et} - k \quad [dB-Hz] \quad (1.2)$$

$$(C/No)_t = (C/No)_u + (C/No)_d + (C/No)_{im} \quad [dB-Hz] \quad (1.3)$$

donde:

$(C/No)_u$ es la relación portadora-a-ruido de subida,

$(C/No)_d$ es la (C/No) de bajada,

P_{et} es la potencia de transmisión de la estación terrena (ET),

G_a es la ganancia de la antena de transmisión de la ET,

P_i son las pérdidas por propagación en la dirección i ,

P_{ai} es la pérdida atmosférica en la dirección i ,

$(G/T)_s$ es la figura de mérito del receptor en el satélite,

$(G/T)_{et}$ es la figura de mérito de la estación terrena receptora,

E_s es la PIRE del satélite,

B_{0s} es la diferencia entre los puntos de operación y saturación del repetidor del satélite, y

k es la constante de Boltzmann (-228.6dB).

El valor $(C/No)_{im}$ representa la relación portadora-a-ruido de intermodulación, provocada por el amplificador no lineal en el repetidor del satélite. Este valor depende de B_{0s} y del número de portadoras que utilizan el repetidor.

De acuerdo al tipo de estación terrena que se emplee, y al satélite, se obtendrá un valor total $(C/No)_t$ del enlace. Con este valor es posible calcular la relación $(E_b/No)_t$ total del enlace

$$(E_b/No)_t = (C/No)_t - 10 \log R_c \quad [dB-Hz] \quad (1.4)$$

donde R_c es la tasa de transmisión en bits/segundo.

El margen del enlace, M , se obtiene como la diferencia entre la (E_b/N_0) total o disponible y la (E_b/N_0) requerida para obtener la tasa de bits en error (BER) deseada, es decir

$$M = (E_b/N_0)_t - (E_b/N_0)_r \quad [\text{dB-Hz}] \quad (1.5)$$

Para mantener un margen positivo, debe hacerse un compromiso entre los parámetros del enlace; podría elegirse reducir la potencia de transmisión dando un margen excesivo, o podría también incrementarse la tasa de transmisión, logrando una reducción en la relación (E_b/N_0) requerida mediante la selección de un esquema mejor de modulación y codificación.

1.2. CODIFICACION DE CANAL.

Las técnicas de transmisión digital permiten obtener un sistema de comunicación cuya confiabilidad puede ser aumentada, gracias tanto a la inherente capacidad de la modulación digital para disminuir los efectos del ruido, como a la posibilidad de controlar los errores que eventualmente se presentan.

Considérese la figura 1.2, en la cual se representa un sistema canónico de comunicaciones digitales sin realimentación. La fuente de información puede ser voz, una señal de televisión, datos de una computadora, etc. En cualquier caso, el objetivo de la codificación

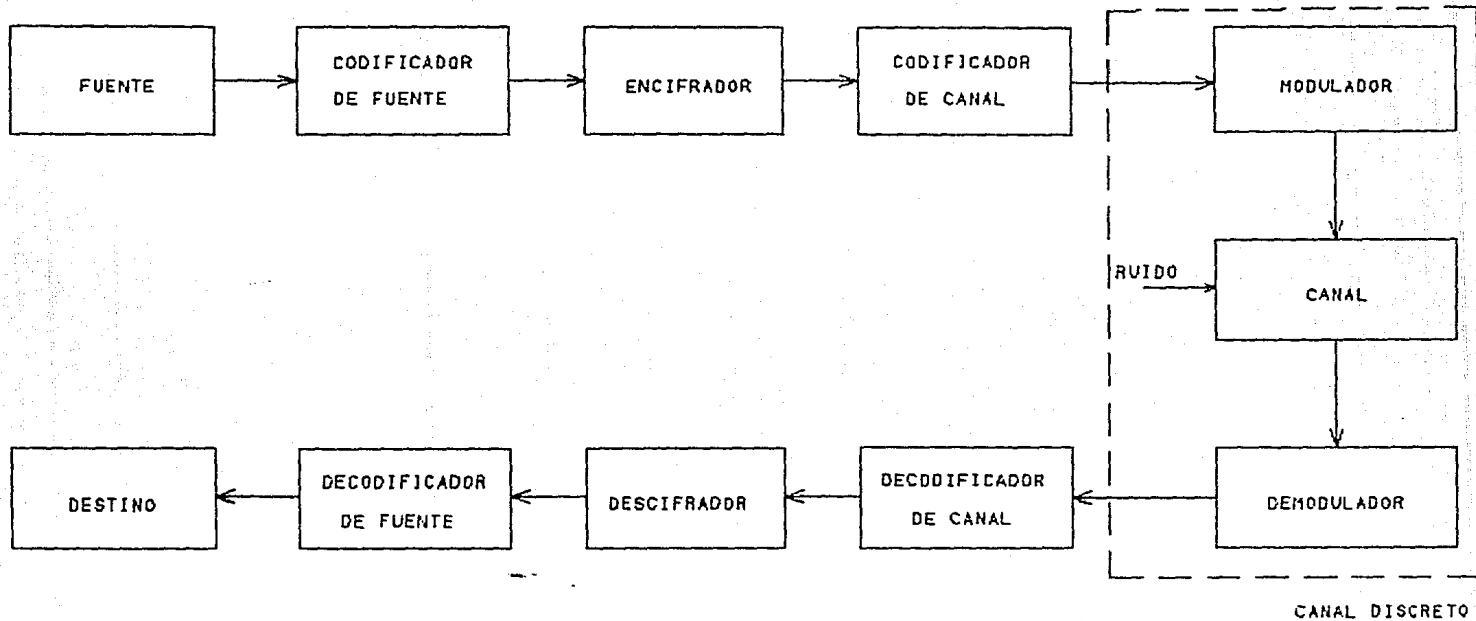


FIGURA 1.2. SISTEMA DE COMUNICACIONES DIGITALES.

de fuente es convertir la señal de la fuente en una secuencia de símbolos binarios (o "bits") a la menor tasa (en bits/segundo) posible. Por otra parte, el objetivo del cifrador es, bajo el control de una clave secreta K' , convertir la salida del codificador de fuente en otra secuencia de símbolos binarios con la misma tasa, o ligeramente mayor, de manera que el descifrador pueda, con la ayuda de la misma clave secreta, reconstruir la secuencia original.

Los sistemas de codificación y modulación deben entregar los símbolos de información, entre el cifrador y el descifrador, a través de un canal analógico, al nivel de seguridad especificado (como puede ser a la menor potencia de transmisión posible, o a la menor tasa de bits en error, etc.).

Los códigos para control de errores permiten que un sistema de comunicaciones alcance un alto grado de seguridad, sin que influya la presencia de ruido, tal y como lo estableció Shannon [20] en 1948. El demostró que si la tasa de símbolos de información (símbolos/segundo) de una fuente es menor que una cantidad llamada "Capacidad del canal", entonces es posible lograr una transmisión sobre un canal ruidoso, con una probabilidad de error tan pequeña como se desee, mediante un esquema adecuado de codificación y decodificación.

Escencialmente el trabajo de Shannon establece que la potencia de la señal a transmitir, el ruido del canal, y el ancho de banda disponible, imponen un límite sólo sobre la tasa de transmisión y no sobre la exactitud del enlace.

La medida más adecuada del desempeño de un sistema de comunica-

ciones digitales es la probabilidad de error establecida en términos del error en un mensaje, una palabra, o un bit.

A la salida del codificador de fuente (y del encifrador) se tiene una tasa de información de R_f bits/segundo. Estos bits de información son codificados con el propósito de protegerlos contra errores inducidos por el ruido del canal, y la salida del codificador proporciona otra secuencia binaria a una tasa de R_c símbolos/segundo.

La tasa de codificación R , en bits/símbolo, está dada por la relación $R = R_f/R_c$. Debido a que R_c es mayor que R_f , la tasa de transmisión en el canal es más alta que la tasa de información entregada por la fuente. Esto es, la introducción de la codificación para controlar errores requiere de una expansión del ancho de banda.

En años recientes, la mayor parte de la investigación se ha dirigido hacia la búsqueda de esquemas prácticos y eficientes de codificación para canales con diversos tipos de ruido. La mayoría del progreso en encontrar dichos esquemas se ha dado en los últimos veinticinco años, y en la actualidad es posible afirmar que la codificación de canal introduce mejoras significativas en el funcionamiento de muchos sistemas prácticos de comunicaciones. Existen numerosas aplicaciones en las que el equipo de codificación se ha construido y empleado con bastante éxito.

La creciente aplicación práctica de la codificación de canal se ha dado gracias a los nuevos avances en el campo de los códigos controladores de errores, y a la dramática reducción en el costo y tamaño de los dispositivos electrónicos de estado sólido que permiten su realización.

1.3. TECNICAS DE CODIFICACION.

La codificación de canal es una técnica de procesamiento de señales que se emplea para mejorar la confiabilidad de las comunicaciones digitales sobre canales con ruido. Esta técnica es conocida también como "Control de errores". Un canal digital puede ser un enlace de comunicaciones para transmitir información entre dos puntos en el espacio, o bien una memoria de computadora que transmite información entre dos puntos en el tiempo. En cualquier caso, el ruido provoca que los datos recibidos difieran de los originales. Con el uso de códigos para el control de errores, el sistema de comunicación puede alcanzar un alto grado de seguridad a pesar de la presencia de ruido.

El control de errores puede clasificarse en dos categorías básicas: (1) Detección de errores y retransmisión, lo que presupone la disponibilidad de un esquema de petición automática (ARQ). (2) Corrección de errores hacia adelante (FEC). Independientemente de si el control de errores es ARQ o FEC, la codificación puede ser de bloque o de árbol. La diferencia fundamental entre estos tipos de códigos es la presencia o ausencia de memoria.

De manera conceptual, el codificador de un código de bloque es un dispositivo sin memoria que transforma una secuencia de entrada de k símbolos a una secuencia de n símbolos. Sin memoria quiere decir que cada bloque de n símbolos depende únicamente de un bloque específico de k símbolos y de ningún otro más.

Los parámetros usuales que distinguen a un código de bloque son los valores de n , k , la tasa $R = k/n$ y la distancia mínima del código, d_{\min} . Donde la distancia mínima del código indica la capacidad de corrección y/o detección de errores del código. Los valores prácticos de k van desde 3 hasta varios cientos y los de R entre $1/4$ y $7/8$. Las secuencias de entrada y salida están formadas generalmente por símbolos binarios, pero pueden ser símbolos de alfabetos de orden mayor (Como los códigos de Reed-Solomon, por ejemplo).

El codificador de un código de árbol es un dispositivo con memoria que acepta vectores o secuencias de m símbolos y da como salida vectores de n símbolos. Cada vector de n símbolos de salida será determinado por el vector de entrada y por v de los símbolos de entrada que le precedieron. El tamaño de la memoria del codificador es, por lo tanto, $v = K - m$ símbolos de información.

El parámetro K se conoce como la longitud de restricción del código. Existe poca consistencia en la literatura a este respecto, algunos autores se refieren a ella como K , mientras que otros como v . En este trabajo, la longitud de restricción será denotada por K .

Para los códigos de árbol, además de la longitud de restricción K , otros parámetros que los identifican son la tasa $R = m/n$ y la distancia libre del código, d_f . La distancia libre de un código de árbol tiene un sentido más elaborado que la distancia mínima, como se explica en el siguiente capítulo, pero esencialmente proporciona la misma información. Los valores típicos de los parámetros m y n son entre 1 y 8, R entre $1/4$ y $7/8$, y K en el rango entre 3 y 61.

Los códigos de árbol de mayor utilidad práctica son los llama-

dos códigos convolucionales, los cuales reciben este nombre del hecho de que la operación de codificación puede ser visualizada como la convolución discreta entre la secuencia de entrada y la respuesta al impulso del codificador.

En cuanto a las técnicas de decodificación, para los códigos de bloque se tienen: La decodificación algebraica, empleada en los códigos de Bose-Chaudhuri-Hoquenghem (BCH) y Reed-Solomon (RS); la decodificación de lógica mayoritaria (Decodificador de Maggitt); los códigos de Fire o códigos entrelazados, para la corrección de paquetes de errores; y los códigos concatenados, para canales con diferentes patrones de error.

Las principales técnicas de decodificación de códigos de árbol son: umbral, secuencial y de máxima similitud (Viterbi). Siendo éstas últimas técnicas probabilísticas que se presentarán en este trabajo.

1.4. CONSIDERACIONES PRACTICAS.

Antes de comenzar la exposición de la teoría y práctica de los códigos convolucionales, es pertinente atender a las consideraciones que deben hacerse antes de elegir un esquema específico de codificación para una aplicación dada. Dependiendo de la aplicación particular, la selección del código y del esquema o algoritmo de decodificación puede ser una tarea complicada, debido a que dicho proceso es afectado por los parámetros de diseño del sistema de comunicaciones.

Con el objeto de determinar si un esquema de control de errores está justificado para una aplicación práctica, Wu [25] propone que

Los siguientes factores deben ser considerados:

- Características del canal de transmisión.
- Desempeño del decodificador.
- Naturaleza o forma del mensaje.
- Velocidad de operación del CODEC (CODificador-DECodificador).
- Tasa de codificación, R.
- Disponibilidad de un canal de realimentación.
- Retraso permisible en el sistema.
- Localización del CODEC en el enlace, y transparencia del código.
- Capacidad de decodificación multidestino.
- Complejidad y costo del CODEC.

Por supuesto, estos factores no son necesariamente independientes. Desde el punto de vista del diseño del sistema, el factor más importante es el desempeño del decodificador. Después de todo, este es el propósito fundamental del control de errores en sistemas de comunicaciones digitales.

Antes de seleccionar cualquier esquema de codificación, o un código específico, es altamente deseable conocer las características del canal de transmisión a emplear. Un código apareado al ruido del canal en cuestión proporcionará la máxima ganancia de codificación (concepto aclarado en el capítulo 3) con la mínima complejidad del CODEC.

La forma del mensaje puede afectar a la selección del CODEC, si el mensaje tiene estructura de bloque, entonces los códigos de bloque podrían seleccionarse. De otra manera, deberán ser considerados los

códigos de árbol. En muchas situaciones prácticas, los mensajes no tienen un tamaño fijo, ni una duración infinita en comparación con la longitud del código. En estos casos, la selección de un código, o conjunto de códigos, no es obvia.

La velocidad de los mensajes procesados por el decodificador es un factor fundamental. A tasas muy altas de transmisión (más de 100 Mbits/segundo con la tecnología actual), se eliminarán los esquemas de decodificación que de otra forma resultarían atractivos en términos de desempeño.

Por otra parte, la tasa de codificación y el retraso de funcionamiento afectan necesariamente a la eficiencia del sistema. A menudo se presenta un compromiso entre el desempeño y la eficiencia, ya que, en general, los códigos de tasas bajas con retrasos grandes proporcionan una mejor calidad que otros códigos de la misma complejidad.

Si el sistema no cuenta con un canal de realimentación, resulta imposible la detección de errores y la retransmisión de mensajes (ARQ).

La capacidad de decodificación multidestino se refiere a la habilidad de un decodificador para decodificar mensajes de codificadores correspondientes a transmisores múltiples.

De acuerdo al sistema de acceso múltiple empleado, se tendrá un problema específico de codificación. Por ejemplo, en un sistema TDMA con modalidad de paquetes, ni los códigos de bloque ni los de árbol son ideales, debido a que todos los paquetes transmitidos tienen longitud variable para poder absorber variaciones del tráfico, y a

que todos los paquetes tienen inicios y terminaciones abruptas que impiden el uso de almacenadores de mensajes.

Con todos estos factores en consideración, resulta claro que no existe un solo código, o clase de códigos, que sea óptimo en una situación real. Lo único que puede hacerse es una aproximación al código ideal requerido en un sistema particular de comunicaciones.

CAPITULO 2. CODIGOS CONVOLUCIONALES.

En este capítulo se expone la teoría básica de los códigos convolucionales, presentando su estructura, formas de representación gráfica y algebraica, y las propiedades de distancia de esta importante clase de códigos lineales.

2.1. EL CODIFICADOR CONVOLUCIONAL.

La entrada de un codificador convolucional consiste en m secuencias de símbolos (pertenecientes a un campo $CG(q)$)

$$X_i = (x_k, x_{k+1}, x_{k+2}, \dots), \quad i = 1, \dots, m \quad (2.1)$$

y la salida consta de n secuencias de símbolos q -arios (Siendo la tasa del código, por lo tanto, $R = m/n$)

$$Y_j = (y_k, y_{k+1}, y_{k+2}, \dots), \quad j = 1, \dots, n \quad (2.2)$$

El término "convolucional" proviene de la observación de que las secuencias de salida pueden referirse como una convolución de la secuencia de entrada con ciertas secuencias generadoras. A las secuencias de entrada y salida se les asocian polinomios en el operador D (llamados transformadas D):

$$X_i(D) = x_k + x_{k+1}D + \dots, \quad i = 1, \dots, m$$

$$Y_j(D) = y_k + y_{k+1}D + \dots, \quad j = 1, \dots, n \quad (2.3)$$

(el operador de retraso D corresponde a z^{-1} en la teoría del muestreo de datos, pero aquí se considera un indeterminante). Entonces, las

secuencias de entrada y salida se relacionan por la expresión

$$Y_j(D) = \sum_{i=1}^m X_i(D)g_{ij}(D), \quad j=1, \dots, n \quad (2.4)$$

donde los polinomios generadores $g_{ij}(D)$ especifican las conexiones entre los K elementos de memoria del codificador y los n grupos de sumadores en $CG(q)$.

En la figura 2.1 se muestra el esquema general de un codificador convolucional. Forney [8] da la siguiente definición de un codificador convolucional:

"Un codificador convolucional (n,m) sobre un campo finito F es un circuito secuencial constante, lineal, causal y de estado finito de m entradas y n salidas."

El codificador está caracterizado por la transformación G de vectores de secuencias de entrada a vectores de secuencias de salida, y se puede escribir en la forma funcional:

$$\bar{y} = G(\bar{x}) \quad (2.5)$$

Constante: Si todas las secuencias de entrada son recorridas en el tiempo, todas las secuencias de salida lo son también. En la notación del operador retraso se tiene

$$G(D^n \bar{x}) = D^n G(\bar{x}) \quad (2.6)$$

$$\begin{aligned} \text{Lineal: } G(\bar{x}_1 + \bar{x}_2) &= G(\bar{x}_1) + G(\bar{x}_2) \\ G(\alpha \bar{x}_1) &= \alpha G(\bar{x}_1) \end{aligned} \quad (2.7)$$

Causal: Si las entradas no nulas comienzan en el instante de tiempo t , entonces las salidas no nulas empiezan en el tiempo $t' > t$.

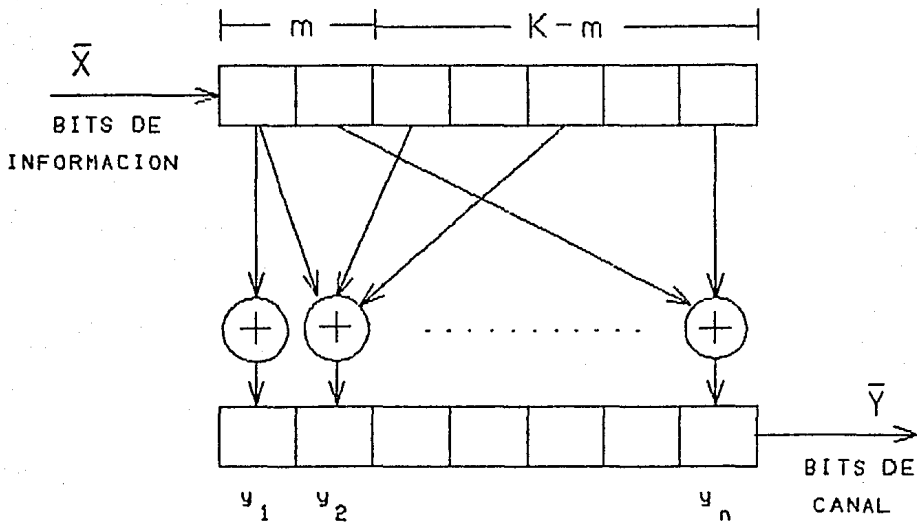


FIGURA 2.1. ESQUEMA GENERAL DE UN CODIFICADOR CONVOLUCIONAL.

Estado finito: El codificador debe tener un número finito de elementos de memoria, cada uno capaz de asumir un número finito de valores. El estado físico de un codificador en cualquier momento está dado por el contenido de sus elementos de memoria; por lo tanto, existe un número finito de estados físicos.

"El código generado por un codificador convolucional G es el conjunto de todas las palabras de código $\bar{y} = G(\bar{x})$, donde \bar{x} es cualquier secuencia de m símbolos."

Se observa que los n símbolos de salida, correspondientes a los m símbolos de entrada al codificador, dependen de los símbolos de entrada y de los $(K - m)$ símbolos que les precedieron. La longitud de restricción del código, expresada en símbolos de información, se define como el número de corrimientos en los que un símbolo de información influye en la salida del codificador [2].

Para los códigos convolucionales binarios, la longitud de restricción es igual a K , la longitud de restricción del registro de corrimiento. Sin embargo, existen otras definiciones de este parámetro, como se señaló en el primer capítulo. Algunas veces, la longitud de restricción del código se expresa en función de los símbolos codificados, como el número de símbolos de salida que son afectados por un símbolo de información, en este caso toma el valor Kn .

Ya que para generar los n símbolos de salida el codificador considera m símbolos de entrada y los $(K - m)$ precedentes, la memoria del codificador es igual a $(K - m)$ símbolos de información.

El codificador convolucional puede caracterizarse mediante n

polinomios de K coeficientes cada uno

$$G'_j = (g_{1j}, g_{2j}, \dots, g_{kj}), j = 1, \dots, n \quad (2.8)$$

o bien como K vectores de conexión de n elementos

$$\bar{G}_j = (g_{1j}, g_{2j}, \dots, g_{nj}), j = 1, \dots, K \quad (2.9)$$

Como el codificador convolucional es un circuito lineal de estado finito, puede describirse mediante su respuesta al impulso. La respuesta al impulso de un codificador convolucional es la salida correspondiente a la secuencia de "impulso" de entrada (1 0 0 0). A partir de lo expuesto anteriormente, la respuesta al impulso consiste en secuencias generadoras del código seguidas de una cadena infinita de ceros:

$$\bar{Y}_1 = (\bar{G}_1, \bar{G}_2, \dots, \bar{G}_K, 0, 0, \dots) \quad (2.10)$$

Debido a que cualquier secuencia de entrada puede ser expresada como una combinación lineal de la respuesta al impulso y de sus versiones retrasadas, usando la linealidad del codificador se llega a la conclusión de que una secuencia de salida puede expresarse mediante la misma combinación lineal de versiones retrasadas de la respuesta al impulso del codificador. Lo anterior implica una multiplicación matricial dada por:

$$\bar{Y} = \bar{X} [G] \quad (2.11)$$

donde $[G]$ es la llamada matriz generadora del código:

$$[G] = \begin{bmatrix} \bar{G}_1 & \bar{G}_2 & \dots & \bar{G}_k & 0 & 0 & 0 & \dots & 0 \\ 0 & \bar{G}_1 & \bar{G}_2 & \dots & \bar{G}_k & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \bar{G}_1 & \bar{G}_2 & \dots & \bar{G}_k & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 & \dots & \bar{G}_1 & \bar{G}_2 & \dots & \bar{G}_k \end{bmatrix}$$

(2.12)

Las características del codificador convolucional pueden resumirse en los siguientes puntos:

- Los símbolos codificados dependen de los valores presentes y pasados de los símbolos de entrada.
- La dependencia en el pasado no se extiende al infinito, sino que está limitada por la longitud del registro de corrimiento.
- El código obtenido es lineal, ya que los símbolos de salida son combinaciones lineales de símbolos anteriores.

2.2. ESTRUCTURA Y FORMAS DE REPRESENTACION.

Ya que la entrada puede tomar en general q^m valores distintos que generan q^m secuencias de salida, la salida de un codificador convolucional puede representarse mediante un árbol con q^{m1} secuencias posibles, correspondientes a $m1$ entradas.

El nodo origen del árbol no tiene ningún predecesor, mientras que cualquier otro nodo del árbol tiene un nodo que le precede. Existen q^m ramas que emanan de cada nodo, con excepción de aquellos

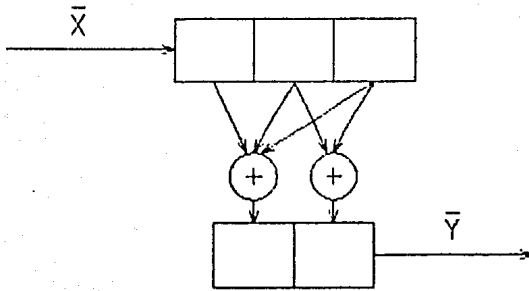
nodos correspondientes a la "cola" del mensaje. La cola del mensaje es empleada para regresar al codificador al estado cero (del cual partió), y consiste en una secuencia de $(K - m)$ ceros que son alimentados al codificador, hasta que el último símbolo de información abandona el registro de corrimiento.

En el árbol del código, cada rama lleva n símbolos codificados. Una trayectoria en el árbol se traza de derecha a izquierda de acuerdo a la secuencia de símbolos de entrada que la especifica. Para un árbol binario (Fig. 2.2), un "0" corresponde - por convención - a tomar la rama superior que sale del nodo, y un "1" significa tomar la rama inferior. Los bits codificados, sobre las ramas de la trayectoria, son las entradas al canal.

El número de niveles del árbol puede extenderse indefinidamente, y naturalmente existe una correspondencia de uno-a-uno entre el conjunto de todas las q^L secuencias posibles de información, de L símbolos de longitud, y el conjunto de trayectorias a lo largo de L niveles del árbol.

Por ser el codificador convolucional un circuito lineal de estado finito, puede representársele mediante un diagrama de estados. Ya que el circuito secuencial de estado finito cambia de un estado a otro de acuerdo a la entrada que recibe, el estado de un codificador convolucional es entonces los $(K - m)$ símbolos de entrada que preceden a los símbolos de entrada en ese instante.

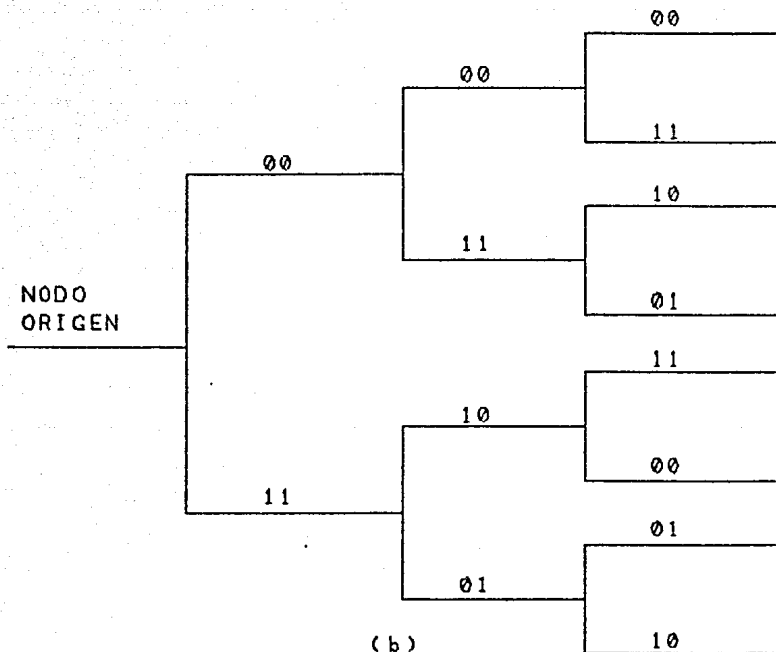
Existen $q^{(K-m)}$ estados para los codificadores de m entradas y, en la transición de un estado a otro, el circuito puede moverse a q^m



$$G'(1) = 101$$

$$G'(2) = 111$$

(a)



(b)

FIGURA 2.2. (a) CIRCUITO CODIFICADOR Y (b) REPRESENTACION DE ARBOL PARA EL CODIGO CONVOLUCIONAL BINARIO DE TASA 1/2 Y K = 3.

estados. Lógicamente, algunas entradas provocarán que el circuito permanezca en el mismo estado.

Lo anterior sugiere representar al codificador con un diagrama de estados con $q^{(K-m)}$ estados o nodos que tienen q^m ramas abandonando y llegando a cada estado. Estas ramas representan la transición del codificador convolucional de un estado a otro, y contienen a los n símbolos codificados que resultan. En la figura 2.3 se muestra el diagrama de estados correspondiente al código convolucional binario de tasa $1/2$ y $K = 3$. La tabla 2.1 muestra las transiciones entre estados para el código anterior.

Puede verse la simplificación lograda al representar al código no con un árbol de profundidad infinita, sino mediante un diagrama de estados consistente en $q^{(K-m)}$ estados. El diagrama de estados es una gráfica de flujo, y los códigos convolucionales pueden analizarse mediante la función de transferencia de dicho diagrama. Resulta evidente que el encontrar esta función de transferencia, para códigos de longitudes grandes de restricción, resulta impráctico, por lo que este análisis está limitado a códigos con longitudes muy cortas de restricción.

A cada nodo del árbol le corresponde un estado del codificador. Si una secuencia de L símbolos se codifica, existen q^L nodos terminales en el árbol de L niveles de profundidad, pero sólo existen $q^{(K-m)}$ estados diferentes. Entonces, si $L \gg (K-m)$, el número de nodos en el árbol excede el número de estados del codificador, y varios nodos tendrán el mismo estado.

Después de $(K - m)$ símbolos consecutivos de información, los

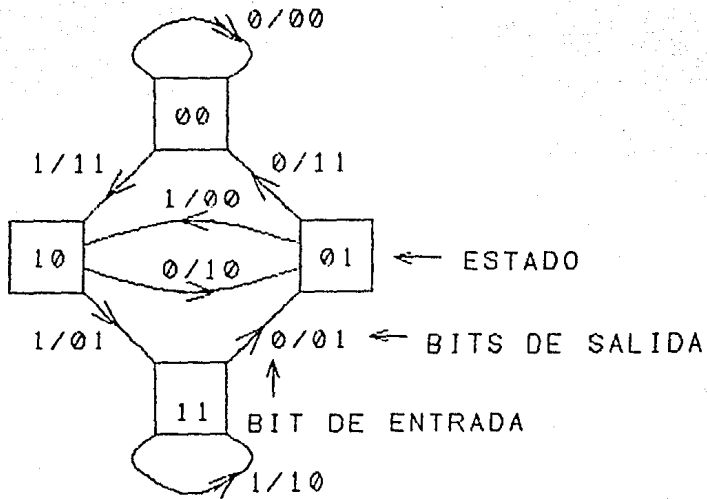


FIGURA 2.3. DIAGRAMA DE ESTADOS PARA EL CODIGO CONVOLUCIONAL BINARIO DE $R = 1/2$ Y $K = 3$.

ESTADO ANTERIOR	BIT DE ENTRADA	ESTADO ACTUAL	BITS DE SALIDA
00	0	00	00
00	1	10	11
01	0	00	11
01	1	10	00
10	0	01	10
10	1	11	01
11	0	01	01
11	1	11	10

TABLA 2.1. TRANSICIONES ENTRE ESTADOS PARA EL CODIGO DE LA FIGURA 2.2.

símbolos codificados son los mismos, es decir, las trayectorias reemergen. Por lo tanto, el árbol contiene información redundante que puede eliminarse si se juntan, en el mismo nivel, aquellos nodos que tienen el mismo estado.

El árbol así obtenido es conocido como enrejado. Cada estado del enrejado tiene q^m sucesores, q^m antecesores, y en cada nivel esta estructura se repite. Este trazado de las trayectorias seguidas por el codificador tiene las siguientes propiedades:

- Si dos secuencias de símbolos son iguales excepto en j símbolos consecutivos de información, entonces las palabras de código serán distintas en $(j + K - m)$ ramas consecutivas.

- Si los símbolos de información de dos trayectorias concuerdan en $(K - m)$ ramas, entonces los árboles secundarios que se extienden a partir de ese momento serán iguales. En la figura 2.4 se muestra el enrejado para el ejemplo de la figura 2.3.

2.3. MEDIDAS DE DISTANCIA.

Las propiedades de la distancia de las secuencias codificadas por codificadores convolucionales son un factor fundamental para determinar su capacidad de corrección de errores [2].

Varias medidas de distancia se han propuesto para esta clase de códigos, y resultan útiles para analizar diferentes técnicas de decodificación.

La medida de distancia más conocida es la distancia de Hamming y está definida como el número de posiciones (símbolos) en las que

símbolos codificados son los mismos, es decir, las trayectorias reemergen. Por lo tanto, el árbol contiene información redundante que puede eliminarse si se juntan, en el mismo nivel, aquellos nodos que tienen el mismo estado.

El árbol así obtenido es conocido como enrejado. Cada estado del enrejado tiene q^m sucesores, q^m antecesores, y en cada nivel esta estructura se repite. Este trazado de las trayectorias seguidas por el codificador tiene las siguientes propiedades:

- Si dos secuencias de símbolos son iguales excepto en j símbolos consecutivos de información, entonces las palabras de código serán distintas en $(j + K - m)$ ramas consecutivas.

- Si los símbolos de información de dos trayectorias concuerdan en $(K - m)$ ramas, entonces los árboles secundarios que se extienden a partir de ese momento serán iguales. En la figura 2.4 se muestra el enrejado para el ejemplo de la figura 2.3.

2.3. MEDIDAS DE DISTANCIA.

Las propiedades de la distancia de las secuencias codificadas por codificadores convolucionales son un factor fundamental para determinar su capacidad de corrección de errores [2].

Varias medidas de distancia se han propuesto para esta clase de códigos, y resultan útiles para analizar diferentes técnicas de decodificación.

La medida de distancia más conocida es la distancia de Hamming y está definida como el número de posiciones (símbolos) en las que

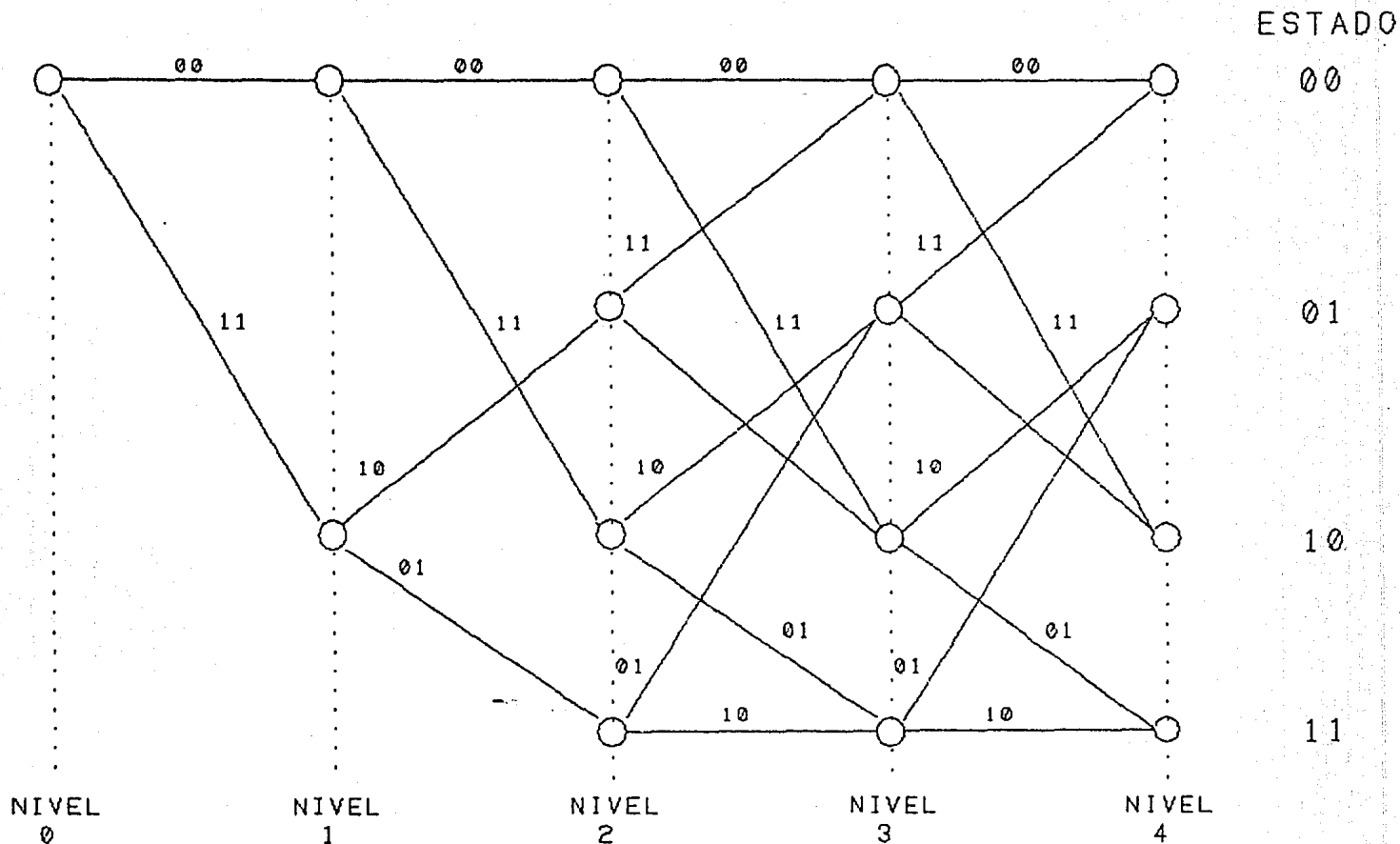


FIGURA 2.4. ENREJADO CORRESPONDIENTE AL CIRCUITO CODIFICADOR DEL CODIGO CONVOLUCIONAL DE TASA 1/2 Y K = 3.

dos palabras de código, de longitud n , difieren

$$d^{(\bar{Y}_1, \bar{Y}_2)}(n) = d_H(\bar{Y}_{1n}, \bar{Y}_{2n}) = W_H(\bar{Y}_{1n} \oplus \bar{Y}_{2n}) \quad (2.13)$$

donde $d_H(\bar{A}, \bar{B})$ es la distancia de Hamming entre las palabras \bar{A} y \bar{B} , y $W_H(\bar{A} \oplus \bar{B})$ es igual al peso de Hamming [2].

La función de distancia de columna de orden j , $dc(j)$, de un código convolucional, es la distancia mínima de Hamming entre parejas de palabras de código con una longitud de j ramas que difieren en la primera rama del árbol [2]. Está dada por

$$dc(j) = \min_{Y_{1j} \neq Y_{2j}} d(\bar{Y}_{1j}, \bar{Y}_{2j}) \quad (2.14)$$

Esta función es no decreciente con respecto a j , y toma dos valores de especial interés: d_{min} , la distancia mínima del código, cuando $j = K$; y df , la distancia libre del código, que se obtiene cuando el valor de j tiende a infinito,

$$d_{min} = dc(K) \quad (2.15)$$

y

$$df = \lim_{j \rightarrow \infty} dc(j) \quad (2.16)$$

Como se mencionó antes, la función de distancia de columna $dc(j)$ es no decreciente con respecto a j , y por tanto debe cumplirse que

$$d_{min} \leq df \quad (2.17)$$

Ya que la distancia libre es la distancia mínima entre dos secuencias codificadas lo suficientemente largas, df es el parámetro

más adecuado para determinar el desempeño del código con un algoritmo de decodificación que observe las secuencias codificadas en más de K símbolos de información. La decodificación secuencial y la de Viterbi son dos técnicas que cumplen esta característica.

Otra medida de distancia la constituye el perfil de distancia de un código convolucional [2], y consta de K funciones de distancia de columna

$$\bar{d} = (dc(1), dc(2), \dots, dc(K)) \quad (2.18)$$

e indica la tasa de crecimiento de las funciones de distancia de columna. Para las técnicas de decodificación secuencial, un buen perfil de distancia se traducirá generalmente en una rápida decodificación.

Dependiendo de la aplicación y del tipo de decodificador a emplear, debe seleccionarse un código convolucional de distancia mínima, distancia libre, o perfil de distancia óptimos. En todos los casos, la búsqueda de estos códigos no es tarea fácil, y se realiza por computadora, ya que no se han encontrado aún métodos analíticos para determinar los códigos óptimos.

Sin embargo, debido a que el número de secuencias a examinar crece exponencialmente con K, estos métodos son ineficientes por la gran cantidad de tiempo de cálculo que es necesario invertir y, aún para códigos de longitudes cortas de restricción, la búsqueda exhaustiva es imposible. Se han encontrado resultados, obtenidos por computadora, mediante métodos que rechazan lo más pronto posible aquellos códigos que no pueden ser óptimos con respecto a la distancia [16,

CAPITULO 3. DECODIFICACION DE CODIGOS CONVOLUCIONALES.

En este capítulo se exponen de manera breve y concisa las técnicas más importantes de decodificación de códigos convolucionales, presentando además un análisis de su desempeño en función del esfuerzo de cálculo, retraso de decodificación, probabilidad de error y ganancia de codificación.

3.1. INTRODUCCION.

En el capítulo anterior, se analizó el funcionamiento del codificador convolucional y las formas de representarlo. La operación de codificación puede representarse en términos de un árbol en el que los símbolos de información determinan, en cada nodo, la rama a seguir.

La trayectoria en el árbol que resulta de estas selecciones sucesivas constituye la salida del codificador. Lo anterior equivale a decir que cada bloque de m símbolos alimentados al codificador es representado, para su transmisión, mediante una secuencia de símbolos seleccionados entre un conjunto de q^m secuencias distintas, y el conjunto que en particular se selecciona depende de los $(K - m)$ símbolos precedentes.

Es claro entonces que la salida al canal, en el intervalo correspondiente a la duración de un bloque de m símbolos de información, proporciona datos no sólo acerca de estos símbolos, sino también de los $(K - m)$ símbolos que los anteceden.

La operación de la decodificación puede ser referida como el

proceso de determinar, a partir de la salida del canal, la trayectoria en el árbol seguida por el codificador convolucional.

Para canales completamente generales, se puede demostrar que si todas las secuencias de entrada al codificador son igualmente probables, entonces el decodificador que minimiza la probabilidad de error de decodificación es aquel que compara las probabilidades condicionales, llamadas también funciones de similitud, $P(Y|X^{(m)})$, donde Y es la secuencia total recibida y $X^{(m)}$ es una entre todas las secuencias transmitidas posibles, y decide a favor de la máxima. Este es un decodificador de máxima similitud.

En general, es más conveniente trabajar con las cantidades $\log P(Y|X^{(m)})$, llamadas funciones de similitud logarítmica o métricas, y el resultado de la decodificación es igual ya que el logaritmo es una función monotónica de su argumento.

Para el canal binario simétrico (CBS), cada símbolo transmitido es recibido erróneamente con una probabilidad $p < 1/2$. Si se transmite una secuencia $X^{(m)}$ de N bits y se recibe la secuencia Y , con una diferencia de d_m bits (donde d_m es la distancia de Hamming), entonces la probabilidad de recibir Y dado que se transmitió $X^{(m)}$ es

$$P(Y|X^{(m)}) = p^{d_m} (1-p)^{N-d_m} \quad (3.1)$$

y la función de similitud logarítmica es

$$\begin{aligned} \log P(Y|X^{(m)}) &= -d_m \log ((1-p)/p) + N \log (1-p) \\ &= -A d_m - B \end{aligned} \quad (3.2)$$

Por lo tanto, maximizar la función de similitud logarítmica en un CBS equivale a minimizar la distancia de Hamming d_m . Esto quiere decir que, para que la probabilidad de error de decodificación sea mínima, debe seleccionarse aquella secuencia de código a la menor distancia de la secuencia recibida.

En el caso del canal con ruido Gaussiano blanco aditivo (RGBA) con modulación PSK, la métrica está dada por [23]

$$\ln P(Y_j | X_j^{(m)}) = C \sum_{k=1}^n y_{jk} x_{jk}^{(m)} - D \quad (3.3)$$

donde n es la longitud del código y $x_{jk}^{(m)}$ es el k -ésimo símbolo de la j -ésima rama del árbol del código.

Así, el decodificador de máxima similitud para el canal RGBA es aquel que forma el producto interno entre las secuencias de código y recibida, y selecciona a aquella con el máximo. Para este canal la métrica es el producto interno (un número real), mientras que la métrica del CBS es la distancia de Hamming (un número entero). Aunque de hecho maximizar el producto interno es equivalente a minimizar la distancia Euclidiana entre los vectores correspondientes [24].

El concepto de decodificación por máxima similitud puede generalizarse a cualquier tipo de canal sin memoria, usando la métrica apropiada $\log P(Y|X^{(m)})$, la cual es siempre determinada a partir de la descripción estadística del canal, tomando en cuenta además la distribución de probabilidad de los símbolos de entrada.

Existen dos tipos principales de decodificación de códigos convolucionales: Secuencial y de Viterbi. Ambas técnicas se han

desarrollado independientemente y parecen ser caminos distintos para determinar la secuencia de información mas probable, dada la secuencia recibida.

3.2. DECODIFICACION SECUENCIAL.

La idea central de la decodificación secuencial es decodificar la secuencia recibida un símbolo a la vez, sin buscar en todo el árbol del código. Comenzando por el origen del árbol, la trayectoria a explorar un paso más es aquella trayectoria cuya métrica es la mayor entre las trayectorias ya examinadas. En consecuencia, al extender sólo aquella trayectoria que parece la más prometedora, la mayor parte de los cálculos necesarios en una decodificación óptima (i.e., considerando todas las posibles trayectorias del árbol) se evita.

La trayectoria que arriva primero al último nivel del árbol con la mayor métrica es aceptada como la trayectoria decodificada. Conforme avanza la decodificación, el decodificador regresa ocasionalmente en el árbol y extiende trayectorias anteriores, posiblemente erróneas. Para minimizar este regreso y extensión de trayectorias no probables, la métrica es modificada de manera que, en promedio, se incremente a lo largo de la trayectoria correcta y decrezca a lo largo de todas las trayectorias incorrectas [11].

El propósito del decodificador es construir una trayectoria de L símbolos extendiéndose en un árbol truncado hasta llegar a uno de los q^L nodos terminales. Al determinar tal trayectoria, el primer símbolo de información es decodificado, los símbolos son recorridos a

la derecha en n símbolos, y el procedimiento es reiterado para determinar el segundo símbolo decodificado.

Para el CES, sea $d(l)$ la distancia acumulada de Hamming entre la trayectoria que se está siguiendo en la l -ésima rama del árbol, $y^*(l)$, y la l -ésima rama de la secuencia recibida, $r(l)$

$$d(l) = \sum_H [y^*(l) \oplus r(l)] \quad (3.4)$$

Conforme el decodificador secuencial penetra rama por rama en el árbol del código a lo largo de la trayectoria tentativa, mantiene una cuenta de $d(l)$. Después de cada penetración sucesiva, el decodificador compara $d(l)$ contra una función de criterio de selección, $k(l)$. Si $d(l)$ excede siempre a $k(l)$, la trayectoria es descartada como poco probable. El decodificador regresa a la rama sin explorar más cercana para la cual $d(l) < k(l)$ y nuevamente la extiende tanto como el criterio de selección lo permita. El decodificador lleva una cuenta de las ramas ya exploradas para evitar retrazos innecesarios de cualquier rama.

Una función conveniente de criterio de selección $k(l)$ es una línea recta (Fig. 3.1). Cuando $y^*(l)$ es correcta, $d(l)$ oscilará alrededor de una línea recta de pendiente pn . Por otro lado, cuando $y^*(l)$ se aleja del nodo origen ($l=0$) a lo largo de una trayectoria incorrecta, $d(l)$ oscilará alrededor de una recta de pendiente $n/2$ (Debido a que, para códigos bien conformados, la distancia de Hamming entre trayectorias correctas e incorrectas se aproxima a la mitad de la longitud del código). Por lo tanto, $k(l)$ es una línea recta de pendiente intermedia $p'n$, $p < p' < 1/2$ [24].

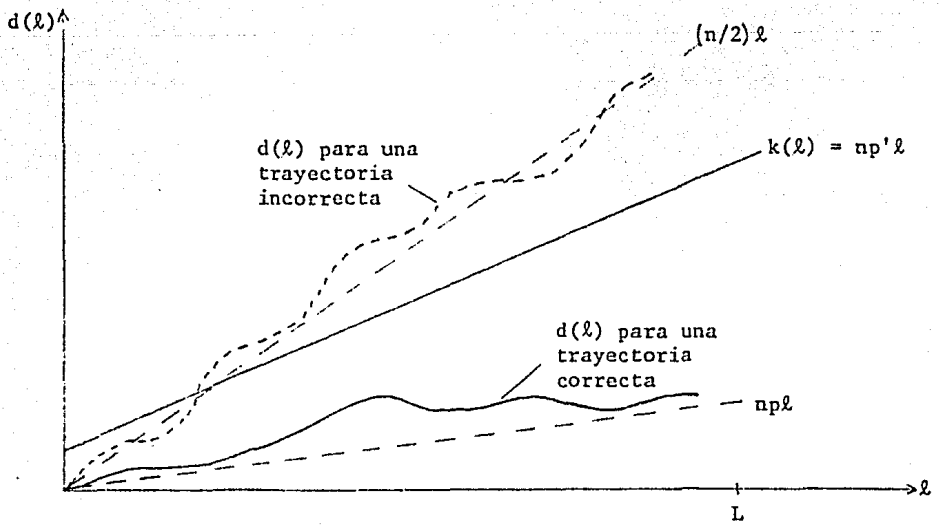


Figura 3.1. FUNCION DE CRITERIO DE SELECCION

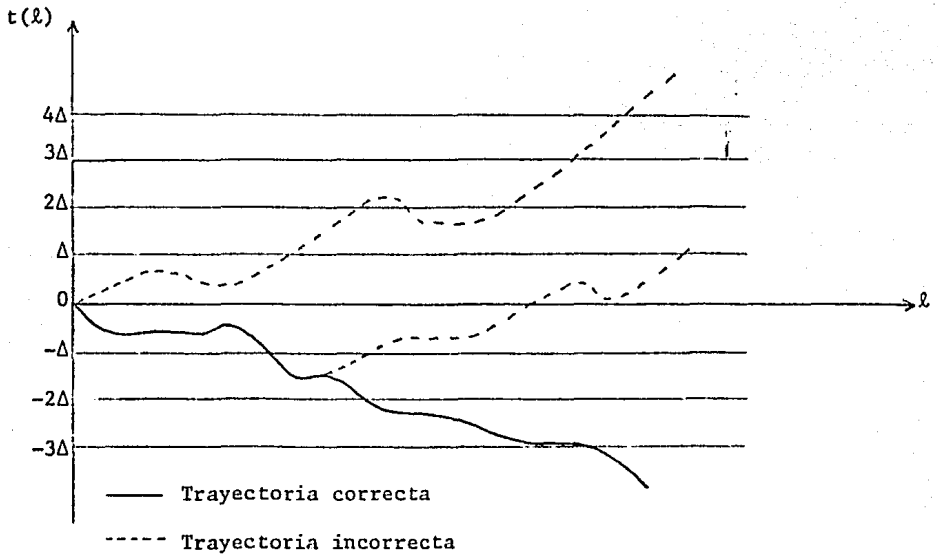


Figura 3.2. FUNCION DE CRITERIO DE SELECCION (UMBRALES) EMPLEADA EN EL ALGORITMO DE FANO.

3.2.1. ALGORITMO DE FANO.

Un algoritmo de decodificación secuencial debe ser capaz de detectar una métrica (o distancia) decreciente (o creciente) y rápidamente encontrar una trayectoria. Para ilustrar el algoritmo de Fano, se considerará el CBS. Para este canal, se define la "función de distancia modificada"

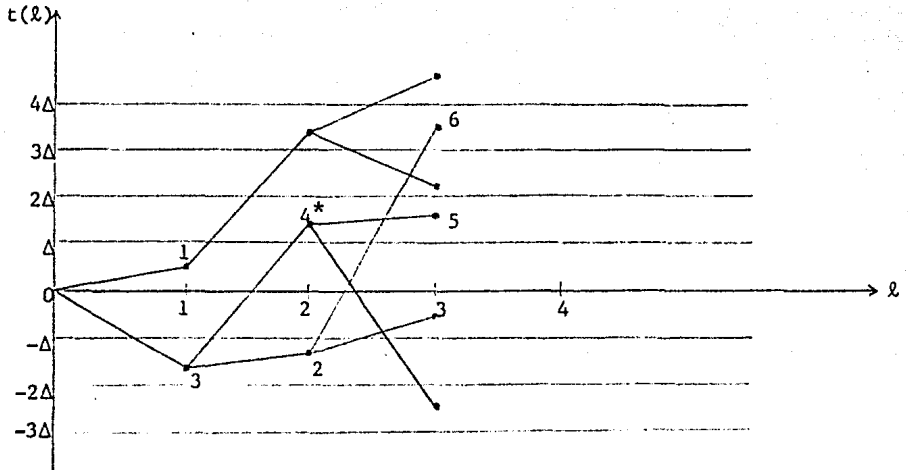
$$t(l) = d(l) - p'nl \quad (3.5)$$

y la función de criterio de selección se convierte en líneas horizontales, llamadas umbrales, con espaciamiento Δ , como se muestra en la figura 3.2 [24].

En este caso, cuando $y^*(l)$ es la trayectoria correcta, $t(l)$ tiende al valor negativo $(p - p')nl$ y decrece conforme l crece. En el caso en el que $y^*(l)$ es incorrecta, $t(l)$ se comporta como $(1/2 - p')nl$ y tiende a incrementarse conforme crece el valor de l .

Las convenciones empleadas en el algoritmo de Fano son: Un nodo del árbol satisface todos los umbrales que se encuentran en o sobre él, y viola todos los umbrales debajo de él. El umbral "más ajustado" satisfecho por un nodo es aquel que está más cercano. De todos los nodos que divergen de un nodo dado, aquel con el menor valor de $t(l)$ es llamado el "mejor", y aquel con valor mayor el "peor". Lo anterior está ilustrado en la figura 3.3.

El nodo considerado por el decodificador se designa mediante un apuntador de nodo de búsqueda. El decodificador de Fano conserva un umbral móvil $T = i \Delta$, donde i es una variable entera. El umbral



*Apuntador de nodo de búsqueda

Figura 3.3. ARBOL DE DISTANCIAS DEL ALGORITMO DE FANO.

móvil es ajustado cuando el valor de i es tal que T es el umbral más ajustado satisfecho por el nodo de búsqueda, es decir, por el nodo considerado en ese momento.

Dado un vector recibido \bar{r}' , el decodificador de Fano busca la trayectoria correcta moviendo su apuntador de nodo de búsqueda a lo largo del árbol. El apuntador puede moverse hacia adelante o hacia atrás, pero sólo a un nodo adyacente. El movimiento del apuntador es controlado por el diagrama de flujo mostrado en la figura 3.4 [7].

Una característica esencial del algoritmo es que el apuntador nunca se mueve si viola el umbral móvil, éste es incrementado sólo cuando es necesario forzar dicho movimiento.

Resulta claro de este diagrama que el algoritmo de Fano trazará eventualmente la trayectoria correcta siempre que $t(1)$ decrezca, y que $t(1)$ crezca para trayectorias incorrectas. En particular, el algoritmo no caerá nunca en un lazo infinito buscando siempre los mismos nodos con los mismos umbrales. La figura 3.5 presenta un ejemplo que sirve para ilustrar lo expuesto antes.

3.2.2. ALGORITMO DE ZIGANGIROV-JELINEK.

Una clase interesante de algoritmos de decodificación secuencial son los algoritmos de pila, propuestos de manera independiente por Zigangirov [26] y Jelinek [14]. El algoritmo de Zigangirov-Jelinek (Z-J), o de Pila, es una técnica simple, que ilustra claramente los conceptos de la decodificación secuencial.

El decodificador consiste de una lista o pila de las trayecto-

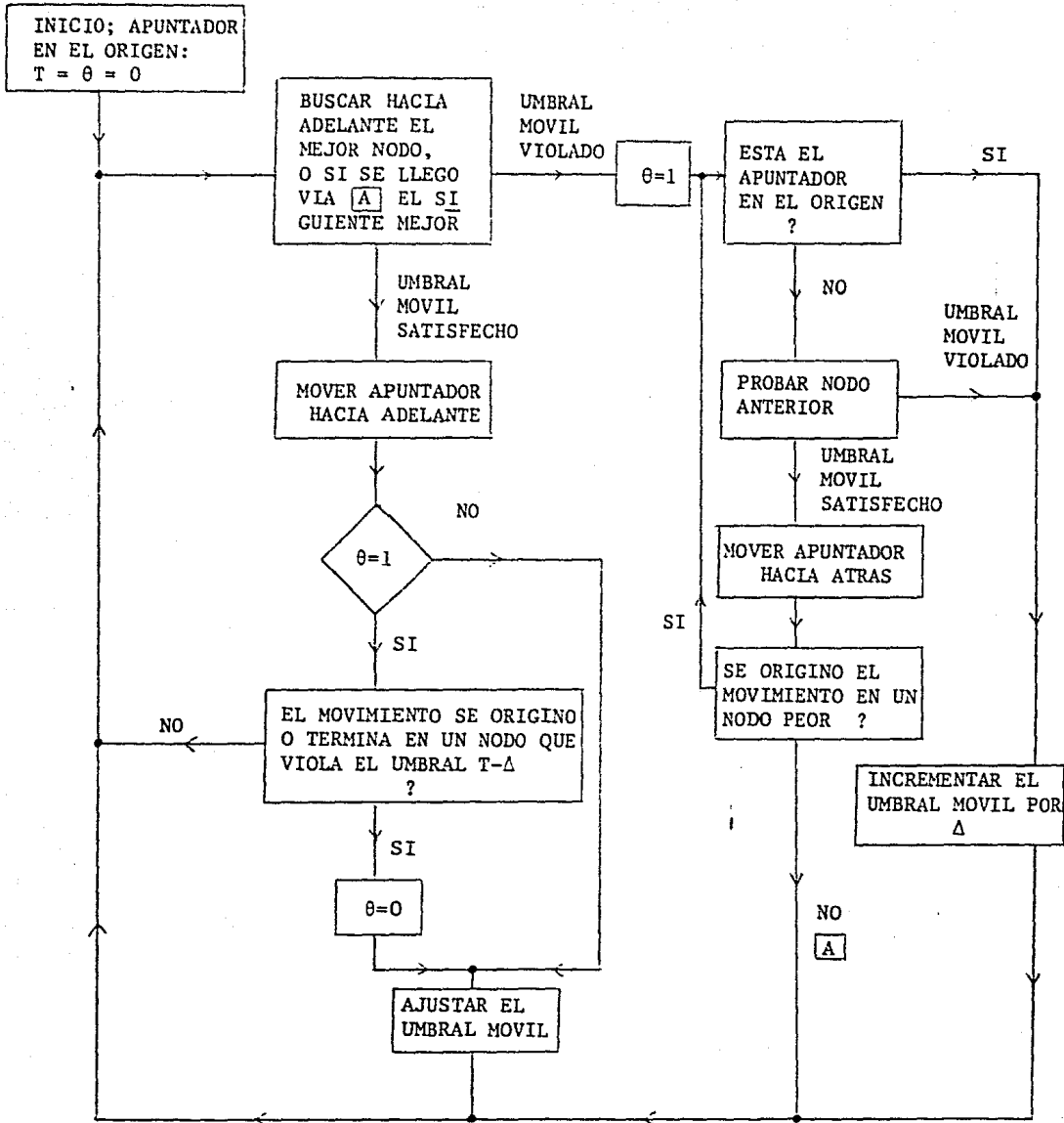
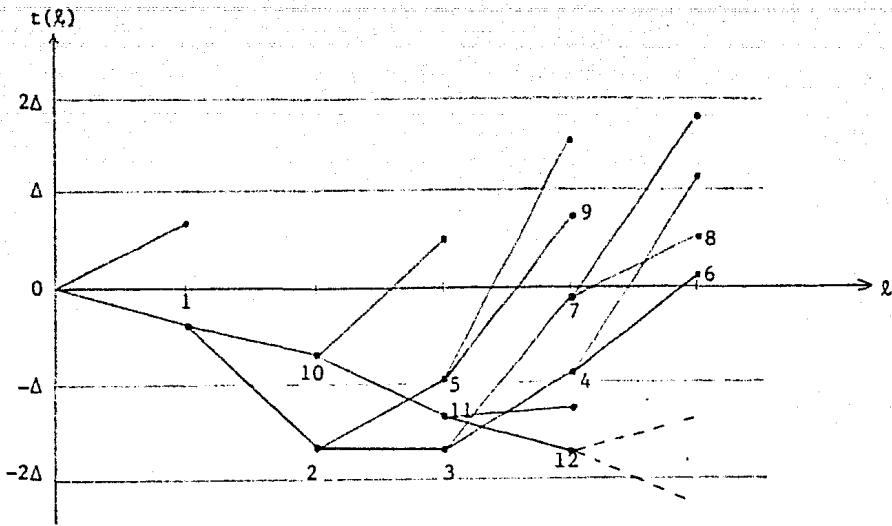


Figura 3.4. ALGORITMO DE FANO.



APUNTADOR EN EL NODO	UMBRAL MOVIL	θ	ACCION TOMADA	
0	0	0	Probar 1	Apuntar a 1
1	0	0	Probar 2	Apuntar a 2 $T=-\Delta$
2	$-\Delta$	0	Probar 3	Apuntar a 3
3	$-\Delta$	0	Probar 4	Apuntar a 2 $\theta=1$
2	$-\Delta$	1	Probar 5	Probar 1 $\theta=1$
2	0	1	Probar 3	Apuntar a 3
3	0	1	Probar 4	Apuntar a 4 $\theta=0$
4	0	0	Probar 6	Probar 3 $\theta=1$
3	0	1	Probar 7	Apuntar a 7 $\theta=0$
7	0	0	Probar 8	Probar 3 $\theta=1$
3	0	1	Probar 2	Apuntar a 2
2	0	1	Probar 5	Apuntar a 5 $\theta=0$
5	0	0	Probar 9	Probar 2 $\theta=1$
2	0	1	Probar 1	Apuntar a 1
1	0	1	Probar 10	Apuntar a 10 $\theta=0$
10	0	0	Probar 11	Apuntar a 11 $T=-\Delta$
11	$-\Delta$	0	Probar 12	Apuntar a 12

Figura 3.5 EJEMPLO DE LA BUSQUEDA EN EL ARBOL DEL CODIGO CON EL ALGORITMO DE FANO.

rias ya exploradas ordenadas en orden decreciente de sus valores de métrica (o en orden creciente de sus distancias de Hamming). La cabeza de la pila es la trayectoria con la mayor métrica acumulada entre las trayectorias en la pila, y será explorada (extendida en sus ramas un nivel adelante en el árbol), ya que es más probable que sea la correcta.

La pila es reordenada después de cada extensión, para que la trayectoria cuya métrica siempre crece (o cuya distancia se mantiene como la menor) sea explorada más adelante. Si su métrica decrece y cae de la posición superior, esta trayectoria es reordenada en la pila y el nuevo nodo superior extendido.

Si denotamos a cada trayectoria por el nodo en su extremo, la pila puede visualizarse como una lista de nodos ordenados de acuerdo a sus valores de métrica. Las funciones del decodificador son entonces: la búsqueda del nodo superior, la extensión del mismo en sus nodos descendientes, y la reordenación apropiada de la pila. El procedimiento a realizar para un código de tasa m/n y elementos de un campo finito $CG(q)$ es el siguiente [26]

Paso 1. Calcular los valores de la métrica para las q^m trayectorias que parten del nodo inicial, y colocarlas en la pila.

Paso 2. Tomar el nodo correspondiente a la mayor métrica. Calcular los valores de métrica para las ramas que parten de él y colocarlas en la pila, eliminando el nodo original.

Paso 3. Ordenar en orden decreciente de valor de métrica a los nodos del árbol.

Paso 4. Si la trayectoria superior es de profundidad L , ir al

paso 5. En caso contrario regresar al paso 2.

Paso 5. El proceso de decodificación termina cuando la trayectoria a extender es de profundidad L .

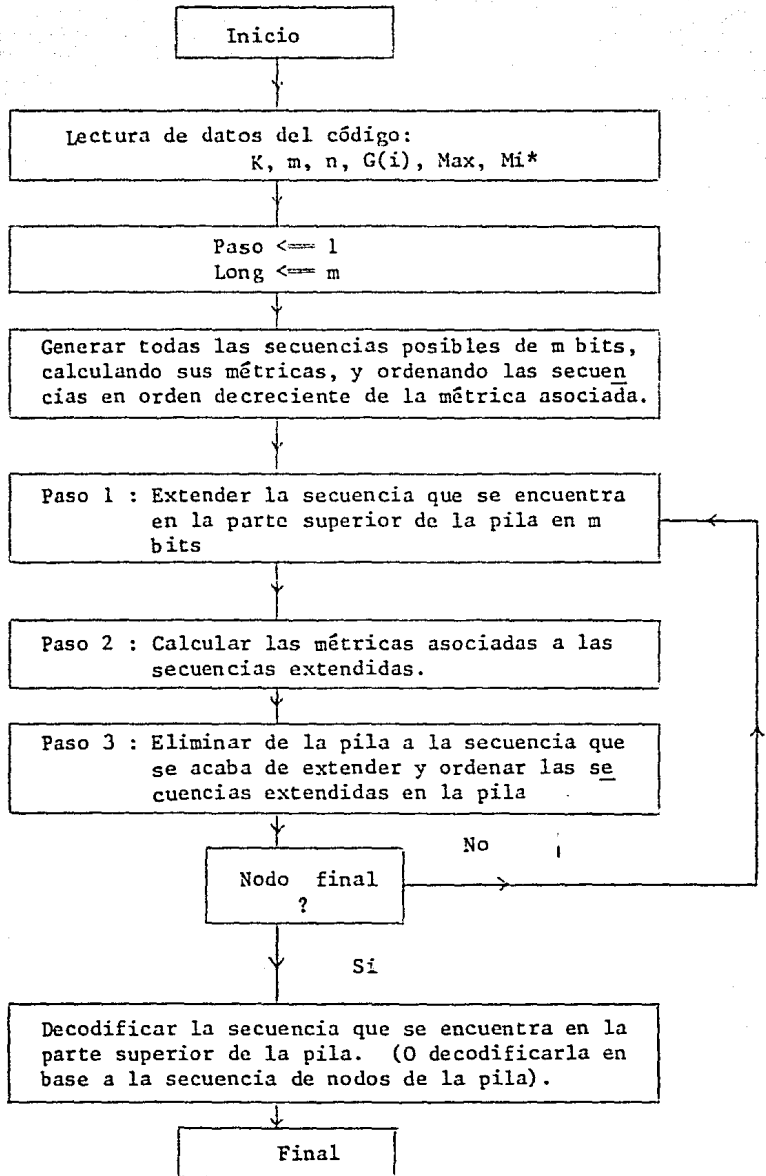
El algoritmo Z-J fué simulado por computadora, con el diagrama de flujo de la figura 3.6. Para ilustrar esta técnica de decodificación se presenta un ejemplo, para el código de tasa $1/2$, $K = 3$ y polinomios generadores $G_1' = 101$ y $G_2' = 111$, en la figura 3.7.

3.2.3. ALGORITMO DE PILA MULTIPLE.

Como se muestra en la figura 3.8, el algoritmo de pila múltiple (APM) [5] opera inicialmente de igual manera que el algoritmo de pila sencilla (APS) descrito en el inciso anterior. Comenzando por el nodo origen, el nodo superior de la pila es extendido. Después de su eliminación de la pila, los nodos sucesores son insertados y la pila es reordenada de acuerdo a los valores métricos.

Si se alcanza un nodo terminal del árbol antes de que la primera pila se llene, la decodificación es completa y la trayectoria que lleva del origen a este nodo terminal se toma como la palabra decodificada. Por lo tanto, si la decodificación puede completarse en la primera pila, el APM se comporta de manera igual que el APS, y obtiene la misma decisión de decodificación.

Si la secuencia recibida requiere de búsquedas extendidas (como en el caso de un borrado o de un paquete de errores), la primera pila se llenará. En este caso, los T nodos superiores de la primera pila - aquellos cuyas métricas son las "mejores" - se transfieren a una segunda pila, continuando la decodificación.



* M_0, M_1 : medidas de similitud. M_0 si los bits son iguales, M_1 si los bits difieren.

Figura 3.6 DIAGRAMA DE FLUJO DEL ALGORITMO Z-J.

<u>PASO DE DECODIFICACION</u>	<u>TRAYECTORIAS</u>	<u>METRICAS</u>
1	0	+2
	1	-18
2	00	+4
	10	-16
	1	-18
3	100	+6
	000	-14
	10	-16
	1	-18
4	1100	+8
	0100	-12
	000	-14
	10	-16
	1	-18
5	01100	+10
	11100	-10
	0100	-12
	000	-14
	10	-16
	1	-18

Secuencia transmitida: 01100...

Figura 3.7. EJEMPLO DEL CONTENIDO DE LA PILA, PARA EL CODIGO DE TASA $1/2$, $K=3$, Y POLINOMIOS GENERADORES $G'_1=101$ Y $G'_2=111$, ME DIANTE EL ALGORITMO Z-J. ($M_0=+1$, $M_1=-9$ SON LAS METRICAS EMPLEADAS CUANDO LOS BITS SON IGUALES O DIFIEREN, RESPEC TIVAMENTE, DE LA SECUENCIA RECIBIDA).

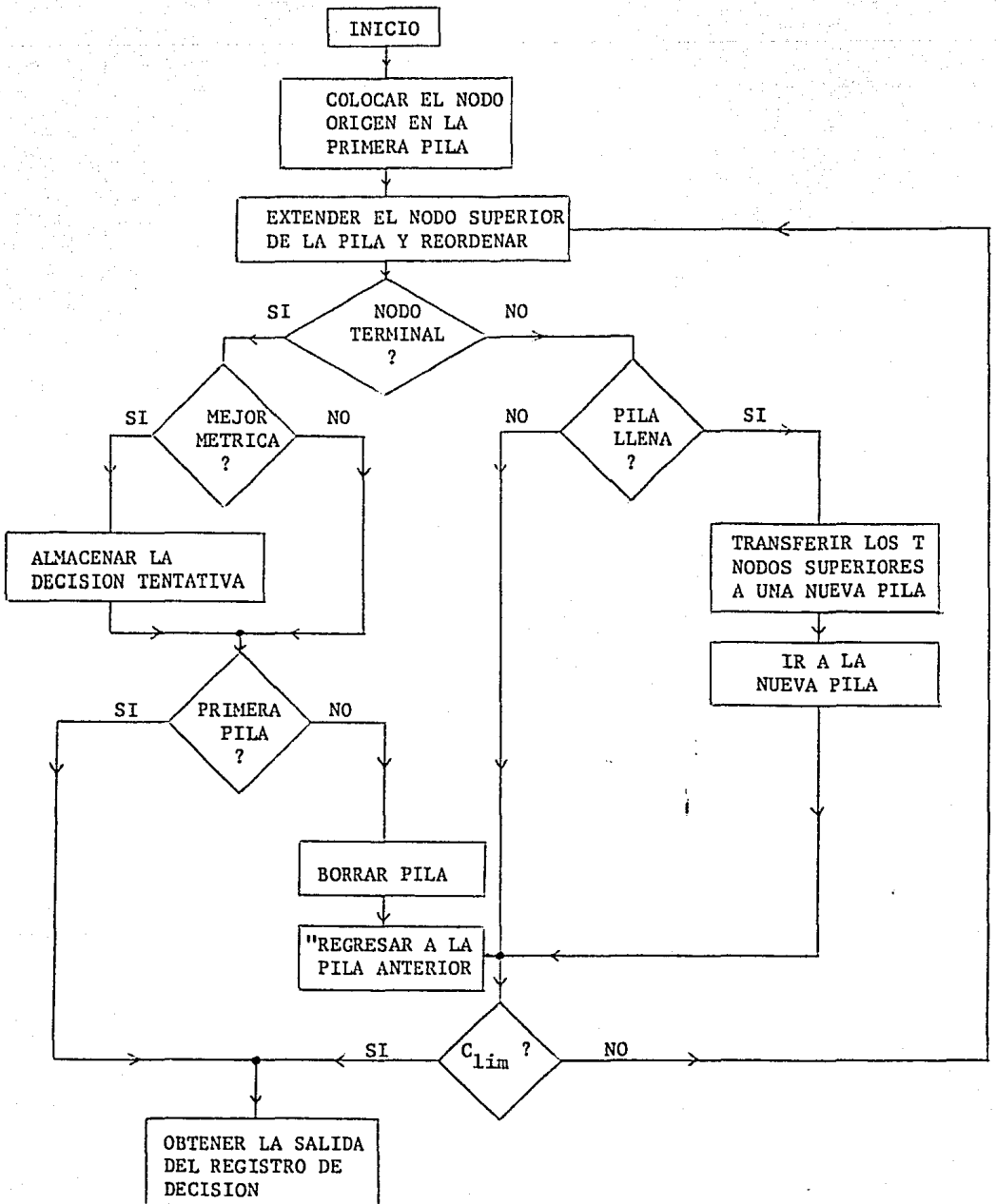


Figura 3.8. ALGORITMO DE PILA MULTIPLE [15] .

En el caso de que el nodo superior de la segunda pila llegue al final del árbol antes de llenar la pila, el nodo terminal es almacenado como una decisión tentativa en un registro especial. El decodificador elimina entonces la segunda pila y regresa a la primera para continuar la decodificación (usando los T lugares libres). Si se llega al final del árbol antes de llenar la primera pila, la métrica del nuevo nodo superior es comparada con la decisión tentativa, y el nodo con la mayor métrica se convierte en la decisión final de decodificación.

Sin embargo, en caso de que la primera pila se llene de nuevo antes del final del árbol, se crea una nueva segunda pila transfiriendo los T nodos superiores de la primera pila. Si la segunda pila se llena también, una tercera pila es creada transfiriendo los T nodos superiores de la segunda pila. Las pilas adicionales que se requieran se crean de manera similar, hasta obtener una decisión tentativa.

El decodificador siempre compara al nuevo nodo terminal con la decisión tentativa y conserva al nodo con mejor métrica. La pila correspondiente es eliminada y la decodificación continúa en la pila previa. La decodificación termina si se llega al final del árbol en la primera pila. La única alternativa de terminación es que el número de cálculos exceda el límite de cálculo, $Clim$, en cuyo caso la mejor decisión tentativa hasta entonces se convierte en la decisión final de decodificación.

Los parámetros de diseño de esta técnica son Z_1 , el tamaño de

la primera pila; k la longitud de decodificación; Z_i el tamaño de las pilas secundarias, $i = 2, \dots, h-1$; T el número de nodos transferidos y $Clim$ el límite de cálculo después del cual el algoritmo debe terminar.

3.3. ALGORITMO DE VITERBI.

El algoritmo de decodificación por máxima similitud de Viterbi, es una técnica poderosa y práctica para decodificar códigos convolucionales, y es particularmente adecuada en canales de satélite [12]. El algoritmo aprovecha la estructura de enrejado del código (ver figura 2.4, capítulo 2) y consta, para un código convolucional de tasa m/n , longitud de restricción K y símbolos en $CG(q)$, de los siguientes pasos [21]

En el paso 1, el decodificador considera todas las q^K trayectorias para las primeras K ramas y calcula las q^K funciones de similitud, o métricas, acumuladas. El decodificador compara entonces la métrica de las q^m trayectorias

$$\begin{aligned}
 & (0, a_{m+1}, a_{m+2}, \dots, a_K), \\
 & (1, a_{m+1}, a_{m+2}, \dots, a_K), \\
 & \dots \dots \dots \\
 & (q^m - 1, a_{m+1}, a_{m+2}, \dots, a_K)
 \end{aligned} \tag{3.6}$$

para cada uno de los $q^{(K-m)}$ vectores (estados) posibles $(a_{m+1}, a_{m+2}, \dots, a_K)$. Realiza por lo tanto $q^{(K-m)}$ comparaciones, cada una entre q^m métricas. A la trayectoria con mayor métrica, en cada comparación, se le llama sobreviviente. Solo las $q^{(K-m)}$ sobrevivientes son conservadas

para futura consideración; el resto de las trayectorias es descartado.

El paso 2 inicia con el cálculo, para cada sobreviviente del paso 1, de las métricas de las q^m ramas que emanan del $(K + 1)$ -ésimo nodo y las acumula a las métricas de las K ramas previas de la trayectoria particular. De nuevo las q^K métricas son comparadas en grupos de q^m , y cada comparación se realiza entre el conjunto de trayectorias:

$$\begin{aligned}
 & (\alpha_{11}^{(1)}, 0, a_{m+1}, a_{m+2}, \dots, a_K), \\
 & (\alpha_{21}^{(1)}, 1, a_{m+1}, a_{m+2}, \dots, a_K), \\
 & \dots \dots \dots \\
 & (\alpha_{q^m}^{(1)}, q^m - 1, a_{m+1}, a_{m+2}, \dots, a_K) \quad (3.7)
 \end{aligned}$$

donde $\alpha_{k1}^{(1)}$ corresponde a la primera rama de la sobreviviente de una comparación realizada en el primer paso. Nuevamente, sólo se conservan las sobrevivientes del conjunto de $q^{(K-m)}$ comparaciones, y las demás son descartadas.

En particular, en el paso $(i + 1)$, el decodificador realiza $q^{(K-m)}$ conjuntos de comparaciones entre grupos de q^m trayectorias:

$$\begin{aligned}
 & (\alpha_{11}^{(i)}, \alpha_{12}^{(i)}, \dots, \alpha_{1j}^{(i)}, 0, a_{j+m+1}, a_{j+m+2}, \dots, a_{j+K}), \\
 & (\alpha_{21}^{(i)}, \alpha_{22}^{(i)}, \dots, \alpha_{2j}^{(i)}, 1, a_{j+m+1}, a_{j+m+2}, \dots, a_{j+K}), \\
 & \dots \dots \dots \\
 & (\alpha_{q^m}^{(i)}, \alpha_{q^m}^{(i)}, \dots, \alpha_{q^m}^{(i)}, q^m - 1, a_{j+m+1}, a_{j+m+2}, \dots, a_{j+K}) \quad (3.8)
 \end{aligned}$$

donde los vectores $(\alpha_{k1}^{(i)}, \alpha_{k2}^{(i)}, \dots, \alpha_{kj}^{(i)})$ dependen de la salida de conjuntos previos de comparaciones.

Por la naturaleza de las comparaciones, no hay sobrevivientes que concuerden en todas las $(K - m)$ ramas, ya que éstas corresponden al estado del codificador, y existe una correspondencia de uno-a-uno entre cada una de las $q^{(K-m)}$ sobrevivientes y las subsecuencias $(a_{j,m+1}, \dots, a_{j,K})$.

El algoritmo procede de esta forma, incrementando en cada paso la población por un factor de q^m , cuando considera al conjunto de q^m ramas que emanan de cada trayectoria sobreviviente, y reduciéndola por este factor al realizar un nuevo conjunto de comparaciones y descartando a todas las trayectorias con excepción de las sobrevivientes.

Mediante simulaciones exhaustivas [22], se ha observado que no todas las trayectorias sobrevivientes difieren a lo largo de la secuencia recibida, sino que tienden a ser iguales a partir de cierto nodo a varias longitudes de restricción del nivel actual (para códigos convolucionales de tasa $1/2$, esta distancia es de 4 a 5 veces la longitud de restricción del código). Por lo tanto, no es necesario esperar a que toda la trayectoria decodificada sea obtenida para empezar a enviar la información al destino, el decodificador entrega los símbolos decodificados con un retraso pequeño conforme avanza por el enrejado. Con este procedimiento, conocido como decodificador truncado de Viterbi, se elimina la necesidad de añadir una "cola" al mensaje (alimentando $(K - m)$ ceros al codificador) y se logra una operación continua del decodificador.

Como parte de este trabajo, se elaboró un programa de computadora que permitiera simular esta importante técnica de decodificación

de códigos convolucionales. El diagrama de flujo de dicho programa se muestra en la figura 3.9, y un ejemplo de este algoritmo, para el código de tasa 1/2, $K = 3$, y polinomios generadores $G_1' = 101$ y $G_2' = 111$, se presenta en la figura 3.10.

Los parámetros de diseño de este algoritmo son el código a emplear (tasa, longitud de restricción y polinomios generadores) y la longitud de decodificación.

3.4. ESFUERZO DE CALCULO Y RETRASO DE DECODIFICACION.

Para la decodificación de Viterbi, tanto el esfuerzo de cálculo como el retraso de decodificación son proporcionales a q^K (donde K es la longitud de restricción del código), debido a que el número de estados del codificador es $q^{(K-m)}$. Por lo tanto, la limitación práctica la impone el número de dispositivos electrónicos del decodificador y solo pueden implementarse códigos cortos, usualmente de $K \leq 8$.

Para la decodificación secuencial, tanto el esfuerzo de cálculo como el retraso de decodificación son casi independientes de K , pero la desventaja es que el número de cálculos (definiendo un cálculo como la extensión de un nodo en sus q^m ramas) que el decodificador debe realizar para decodificar completamente la secuencia recibida es una variable aleatoria con distribución de Pareto [13]:

$$P(C > L) = L^{-\alpha}, \quad L \gg 1 \quad (3.9)$$

donde el exponente de Pareto α , $\alpha > 0$, depende sólo de la tasa del código y del canal [24].

DIAGRAMA DE FLUJO DEL ALGORITMO DE DECODIFICACION DE VITERBI

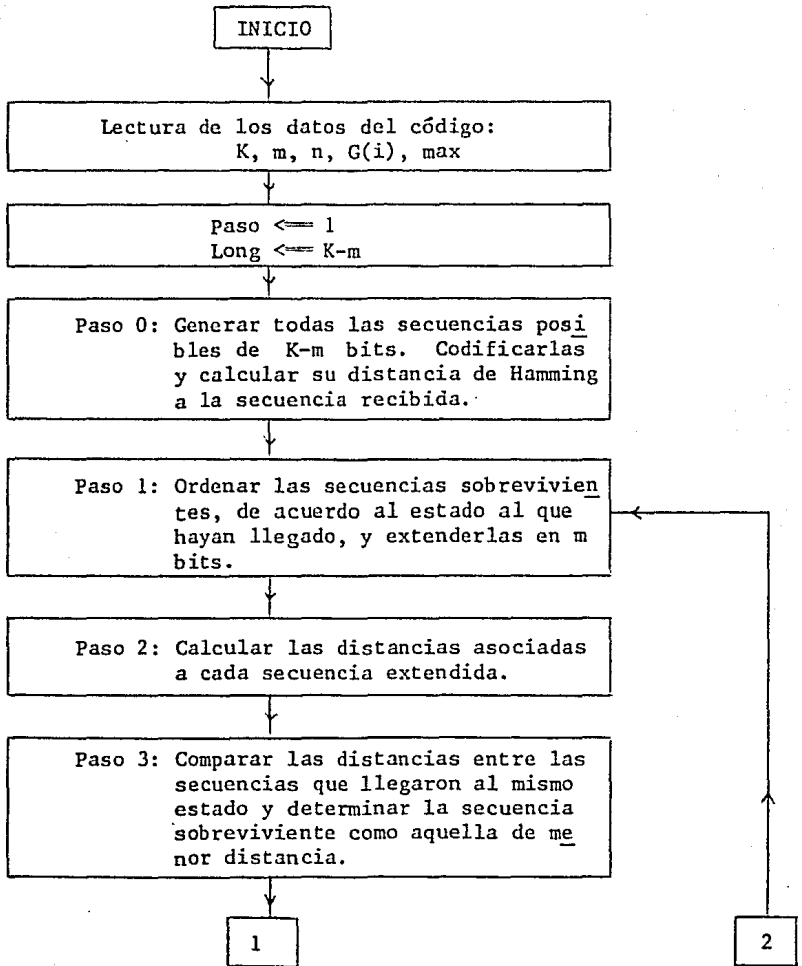
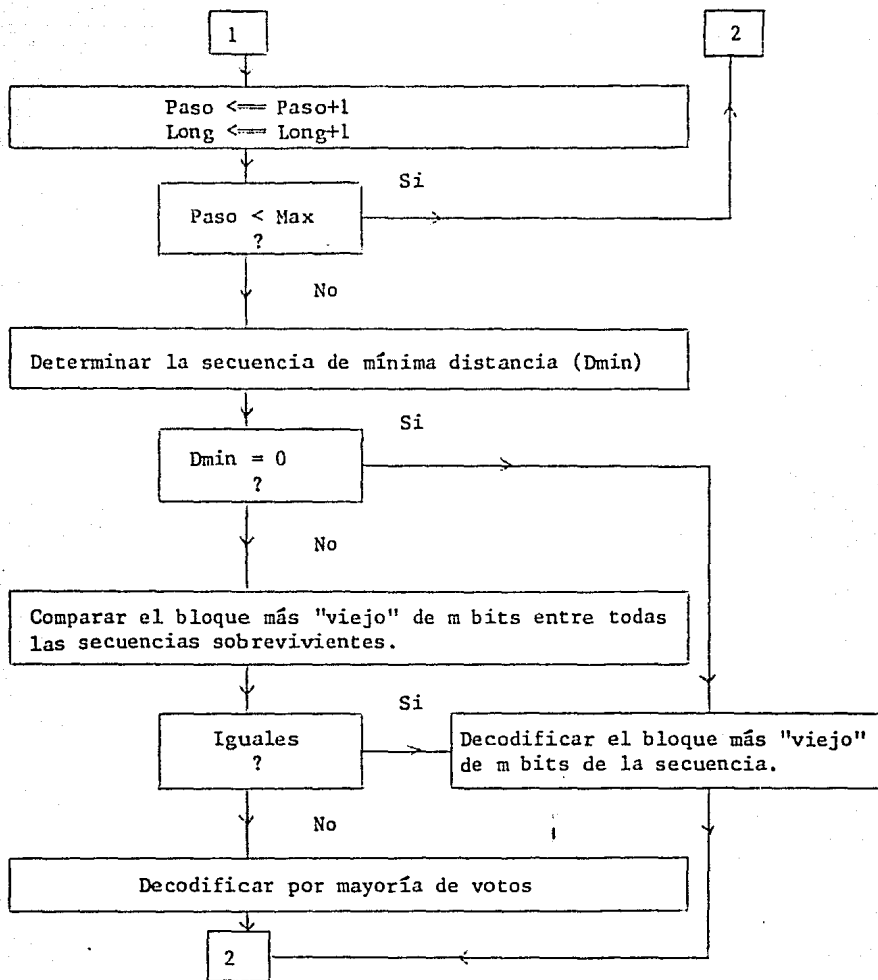
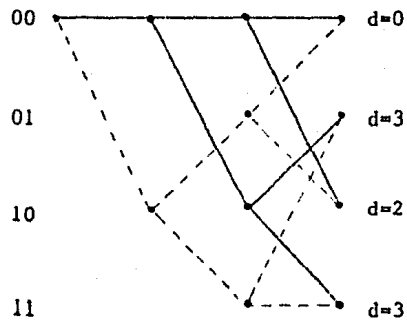


Figura 3.9 ALGORITMO DE VITERBI.

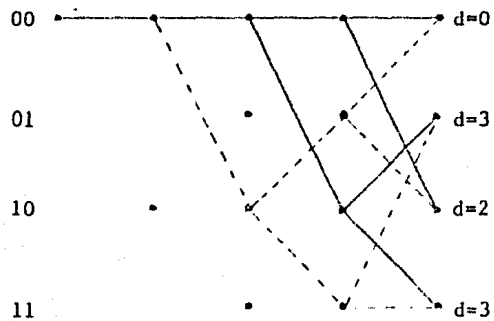


Max = Longitud de decodificación.

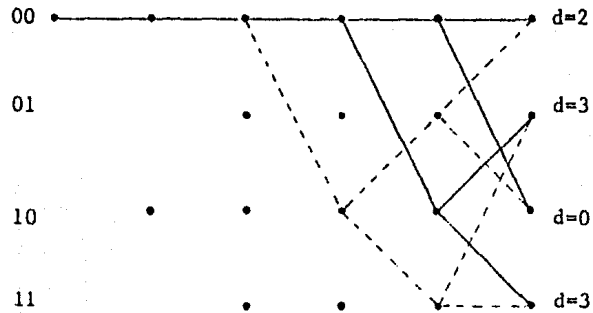
Figura 3.9 ALGORITMO DE VITERBI (Cont.)



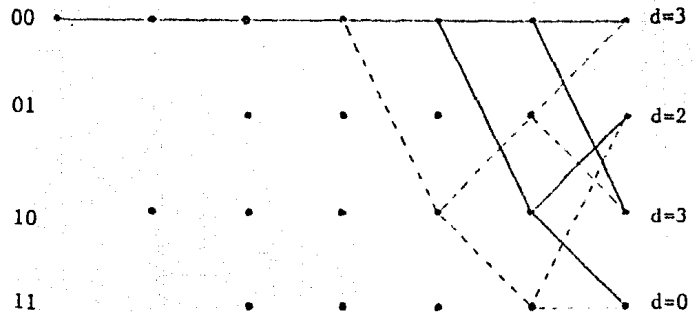
PASO 1



PASO 2



PASO 3



PASO 4

Secuencia transmitida: 000011...

Figura 3.10. EJEMPLO DEL PROGRESO DEL ALGORITMO DE VITERBI, PARA EL CODIGO DE TASA 1/2, $K=3$ Y POLINOMIOS GENERADORES $G_1^1=101$ Y $G_2^1=111$. LAS LINEAS SOLIDAS INDICAN LAS TRAYECTORIAS SOBREVIVIENTES.

El número promedio de cálculos para la decodificación secuencial C_{prom} es, en general, muy pequeño siempre que $\alpha > 1$. Cuando $\alpha < 1$, entonces C_{prom} es no acotada, lo que en la práctica se traduce en sobreflujo del almacenador ("buffer") y en interrupciones en el enlace. La tasa de codificación R_{comp} , correspondiente a $\alpha = 1$, se conoce como la tasa de corte de cálculo (a partir de la cual se puede deducir la expresión para C_{lim} , mencionada en el algoritmo de pila múltiple), y constituye la tasa límite de operación de los decodificadores secuenciales. Para el CBS, dicha tasa está dada por

$$R_{comp} = 1 - \log \left(1 + 2 \left[p(1-p) \right]^{-1/2} \right) \quad (3.10)$$

mientras que para el canal RGBA es

$$R_{comp} = 1 - \log \left(1 + e^{-rE_b/N_0} \right) \quad (3.11)$$

donde rE_b/N_0 es la relación energía-a-ruido para cada símbolo transmitido.

Tanto para los decodificadores de Viterbi como para los secuenciales, el retraso de decodificación es ineludible. Para los decodificadores secuenciales el retraso es una consecuencia directa de la variación en el esfuerzo de cálculo, mientras que para los decodificadores de Viterbi corresponde al almacenamiento de trayectorias en cada estado. La figura 3.11 muestra una comparación de los requerimientos de almacenamiento entre el algoritmo de pila múltiple y el algoritmo de Viterbi.

3.5. PROBABILIDAD DE ERROR Y GANANCIA DE CODIFICACION.

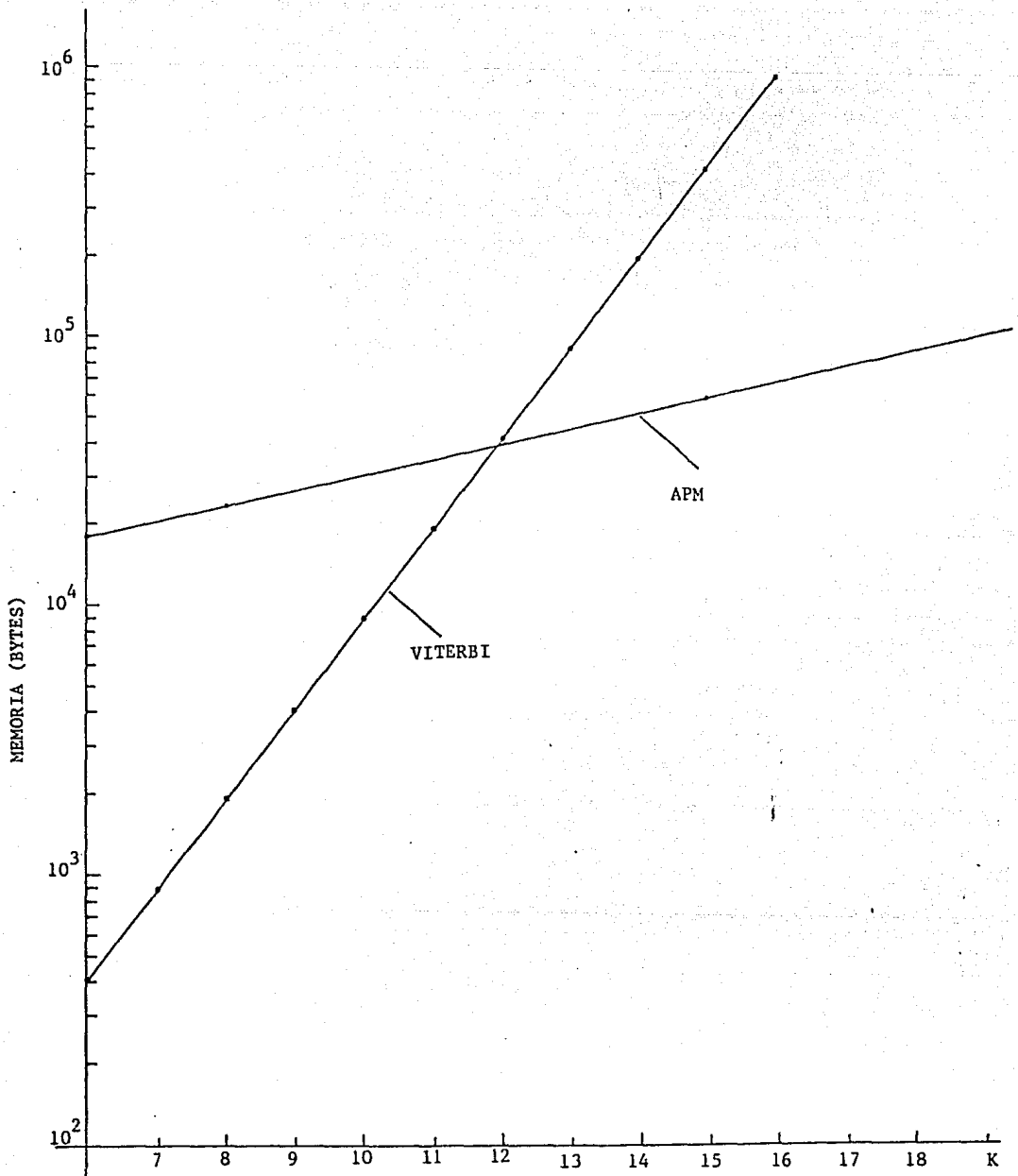


Figura 3.11 COMPARACION DE LOS REQUERIMIENTOS DE MEMORIA ENTRE EL ALGORITMO DE VITERBI Y EL ALGORITMO DE PILA MULTIPLE [26] .

En la decodificación secuencial los errores provienen de dos fuentes: La selección del código (tasa, longitud de restricción y polinomios generadores) determina la probabilidad de no detección, P_{ud} . Esta es la tasa de símbolos en error que se presentaría si el decodificador no tuviera límite en el número de cálculos necesarios para decodificar cada símbolo.

La segunda causa de errores es el tamaño limitado del almacenador en el decodificador, y su probabilidad de sobreflujo asociada, P_a . La probabilidad total de un símbolo en error es entonces

$$P_b = P_{ud} + P_a \quad (3.12)$$

donde P_{ud} depende de la longitud de restricción del código K , y P_a está dada por

$$P_a = AL(\mu T)^{-\gamma} \quad (3.13)$$

donde

μ .- Es el factor de velocidad del decodificador, en número de cálculos por símbolo recibido,

A .- Constante que depende del algoritmo empleado.

L .- Longitud de decodificación en símbolos de información.

T .- Tamaño del almacenador.

γ .- El exponente de Pareto.

En las figuras 3.12 y 3.13 se presentan curvas de probabilidad de error para códigos con diferentes métricas. (La relación de métricas MR es igual a m_0/m_1 para un CBS donde

$$m_0 = \log (2(1 - p)) - B$$

es la métrica cuando el símbolo recibido concuerda con la rama considerada y

$$m_1 = \log (2p) - B$$

cuando los símbolos son distintos. Se elige $B = R$ (la tasa del código) para producir una búsqueda eficiente.)

Para la decodificación de Viterbi, la probabilidad de error se calcula suponiendo que la secuencia nula se transmitió (suposición válida ya que se trata con códigos lineales). Un error de primer evento se comete si la trayectoria nula es descartada por otra que llega al mismo estado. La probabilidad de que la trayectoria nula sea eliminada por un trayectoria de peso j depende sólo del peso de dicha trayectoria. El número de posibles trayectorias de peso j que llegan al mismo estado que la trayectoria nula es n_j , y el peso total de estas trayectorias es w_j .

Una frontera útil [4] en la probabilidad de error se obtiene sumando las probabilidades todas las posibles trayectorias que llegan al mismo estado que la secuencia nula

$$P_b < \frac{1}{m} \sum_{j=0}^{\infty} w_j P_j \quad (3.14)$$

donde w_j es la distribución de pesos del código obtenida a partir de la función de transferencia del diagrama de estados del código [4], y P_j para el CBS está dada por

$$P_j = \begin{cases} \sum_{e=(j,0)/2}^j \binom{j}{e} P^e (1-P)^{j-e} & , j \text{ impar} \\ \frac{1}{2} \binom{j}{j/2} P (1-P) + \dots & \end{cases}$$

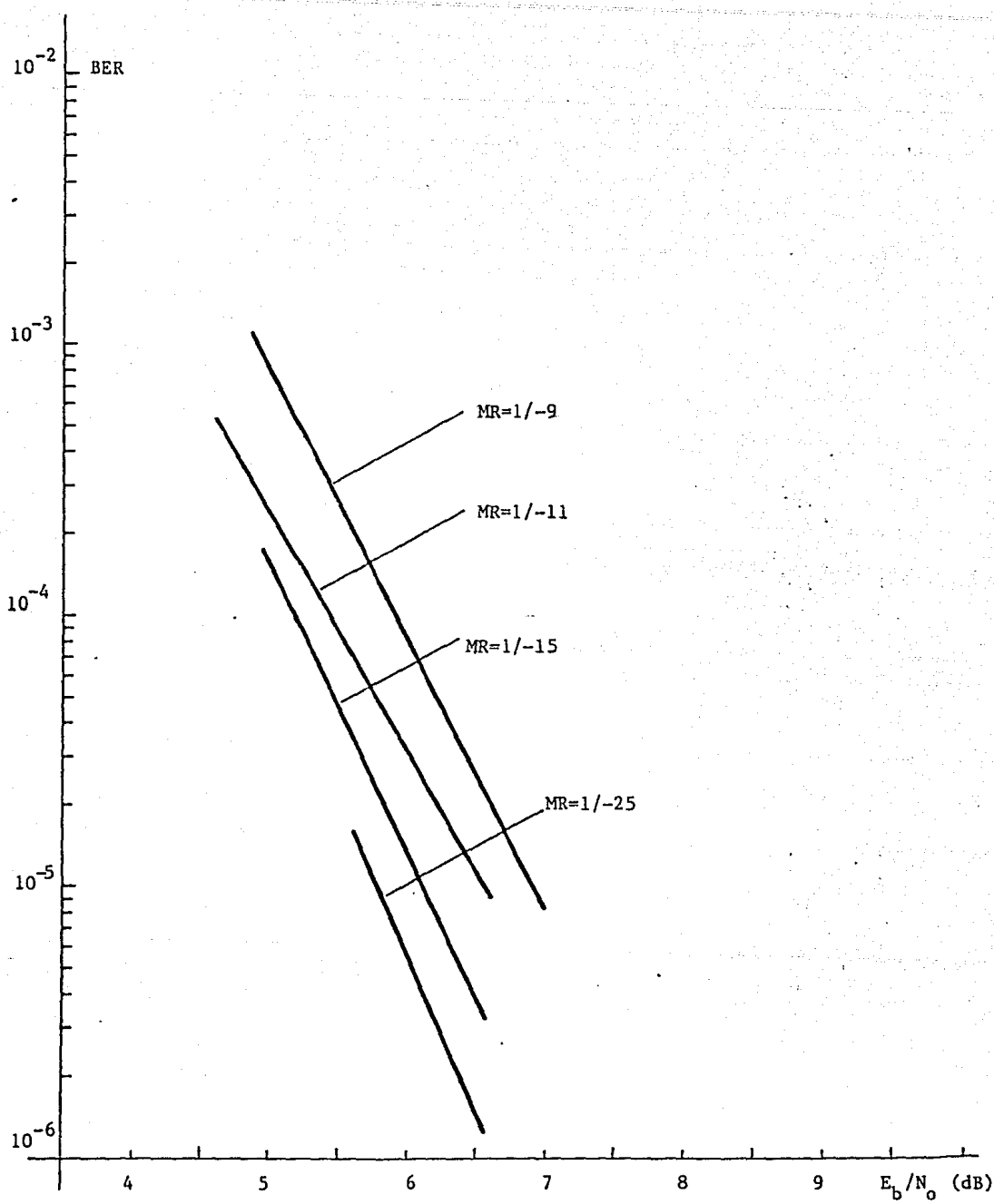


Figura 3.12. CURVAS DE PROBABILIDAD DE ERROR PARA EL CODIGO DE TASA 1/2 Y $K=16$, EMPLEANDO DECODIFICACION SECUENCIAL CON DECISIONES DURAS.

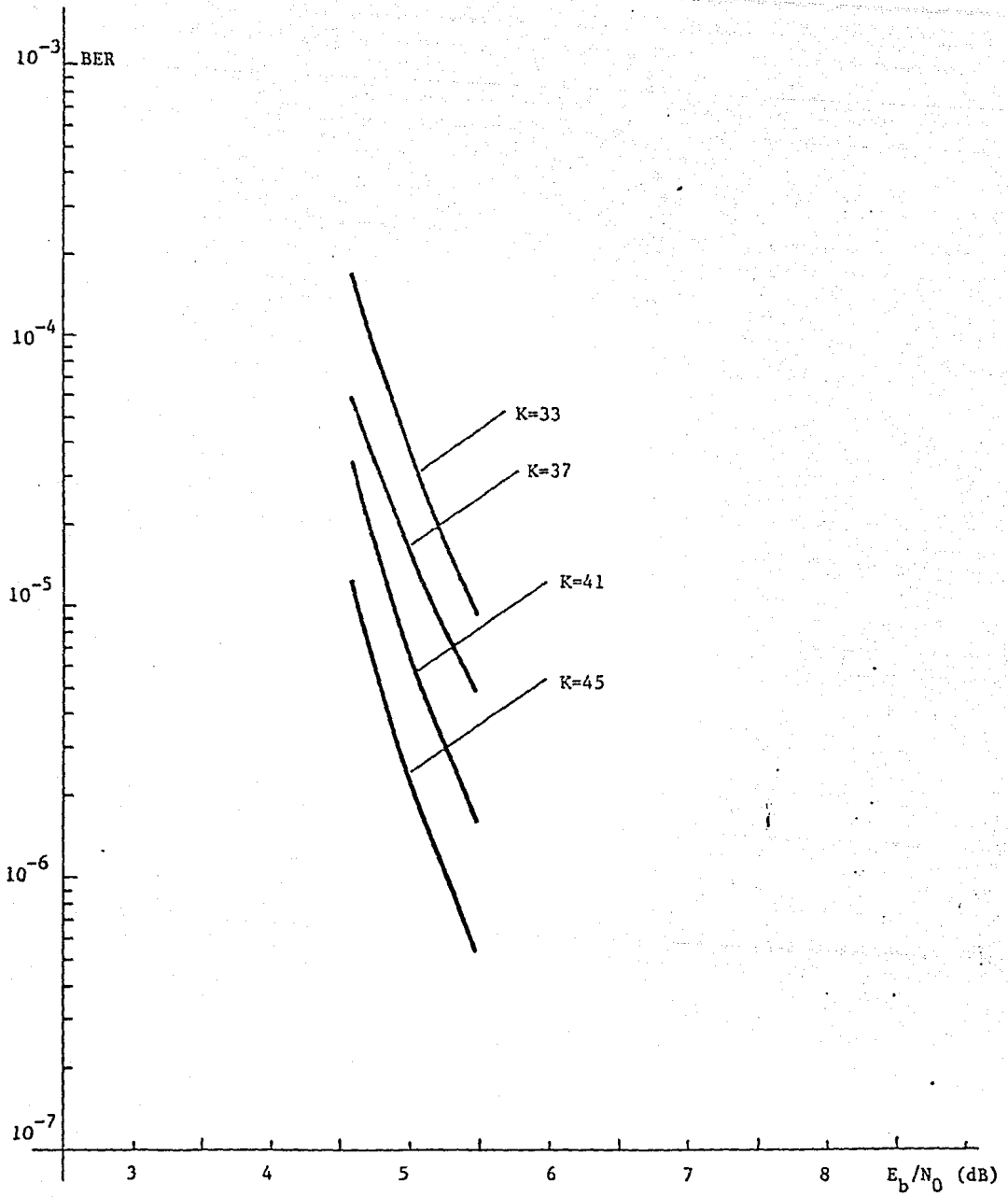


Figura 3.13 CURVAS DE PROBABILIDAD PARA CODIGOS DE TASA 1/2 CON DECODIFICACION SECUENCIAL, DECISIONES DURAS Y RELACION DE METRICAS $M_R=1/-9$.

$$+ \sum_{e=j/2+1}^j \binom{j}{e} p^e (1-p)^{j-e}, \quad j \text{ par} \quad (3.15)$$

mientras que para el canal RGEA con modulación PSK,

$$P_j = \text{erfc} \left(\left[(2j E_b R) / N_0 \right]^{-1/2} \right) \quad (3.16)$$

Las figuras 3.14 y 3.15 presentan las curvas de P_b contra la relación E_b/N_0 para diversos tipos de códigos.

La ganancia de codificación G_c es una medida del beneficio potencial de la codificación, y se define como la relación entre la E_b/N_0 requerida sin codificación a aquella requerida con codificación para una probabilidad de error, o tasa de bits en error, dada

$$G_c = 10 \log (R) + (E_b/N_0)_{sc} - (E_b/N_0)_{cc} \quad [\text{dB}] \quad (3.16)$$

donde R es la tasa del código, $(E_b/N_0)_{sc}$ es la relación E_b/N_0 sin codificación y $(E_b/N_0)_{cc}$ es la relación E_b/N_0 con codificación. En la figura 3.16 se ilustra gráficamente la obtención de G_c para un canal RGEA con modulación PSK.

La evaluación cuantitativa de esta cantidad se ha obtenido para canales RGEA (como el canal de satélite), pero puede obtenerse para otros canales [15]. En la tabla 3.1 se muestra la ganancia de codificación para diversas técnicas de codificación de canal [1].

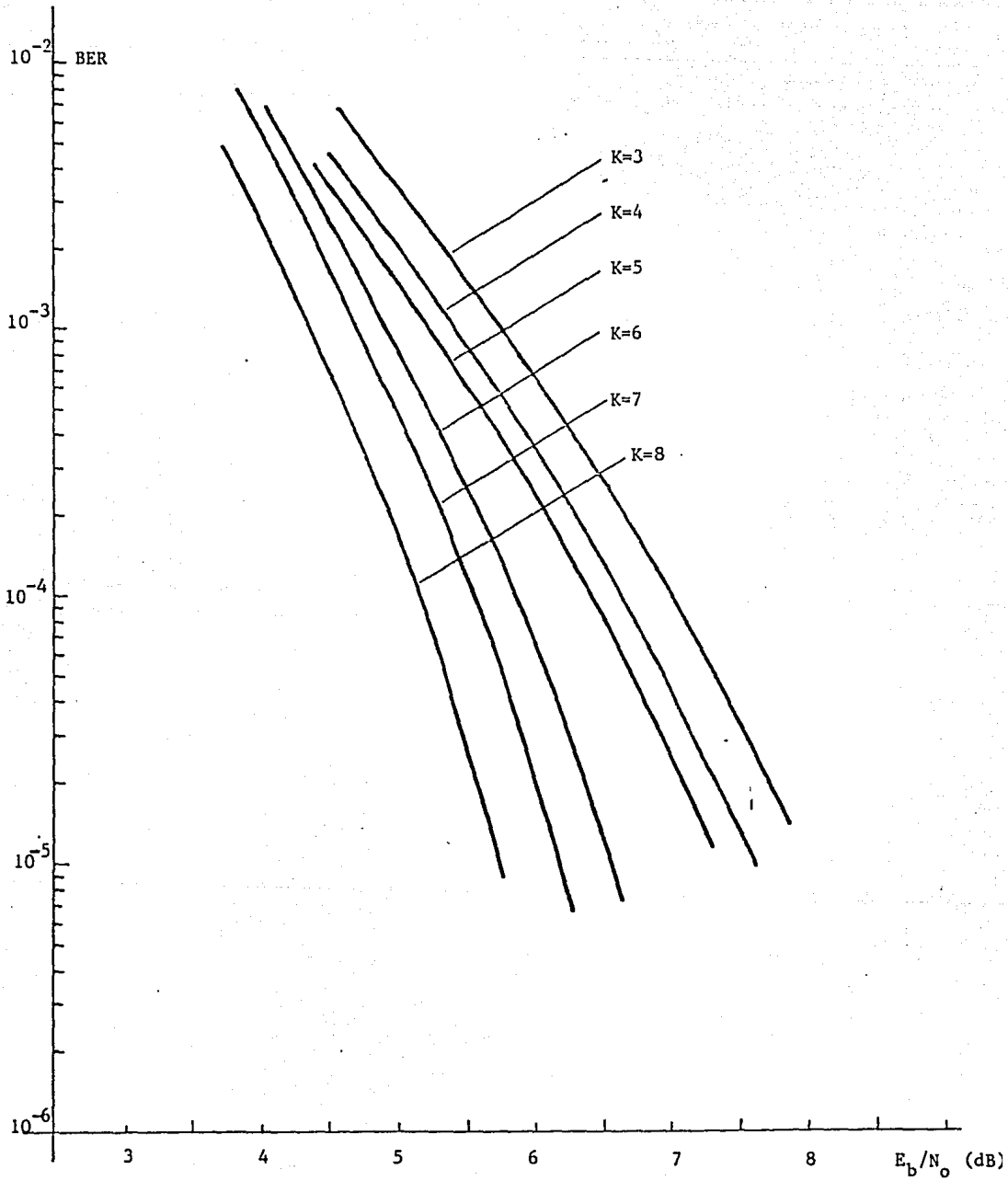


Figura 3.14. PROBABILIDAD DE ERROR PARA CODIGOS DE TASA 1/2 Y DECODIFICACION DE VITERBI SOBRE EL CBS CON UNA LONGITUD DE DECODIFICACION DE 32 BITS.

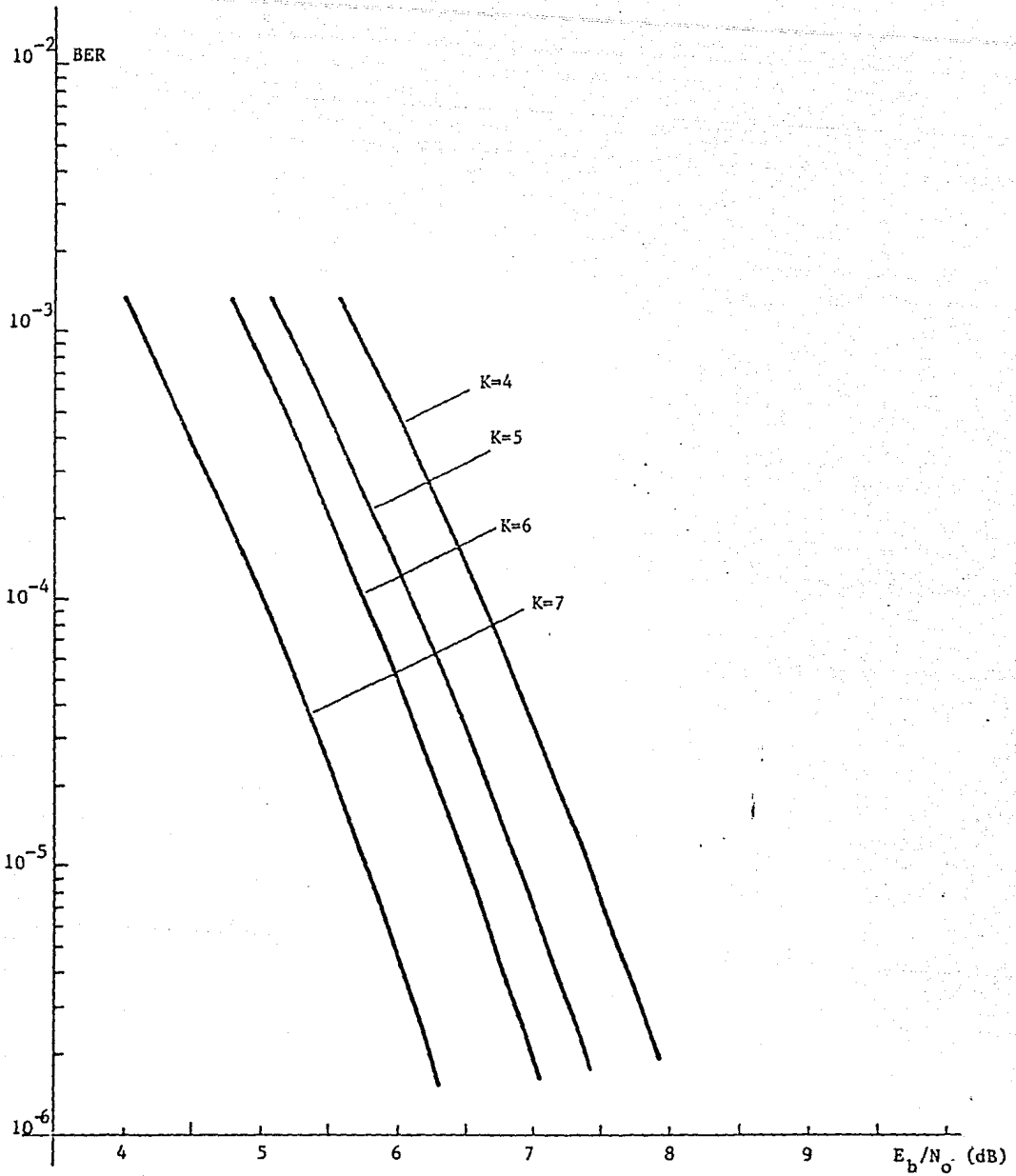


Figura 3.15. PROBABILIDAD DE ERROR PARA CODIGOS DE TASA 1/3, Y DECODIFICACION DE VITERBI SOBRE EL CBS, CON UNA LONGITUD DE DECODIFICACION INFINITA.

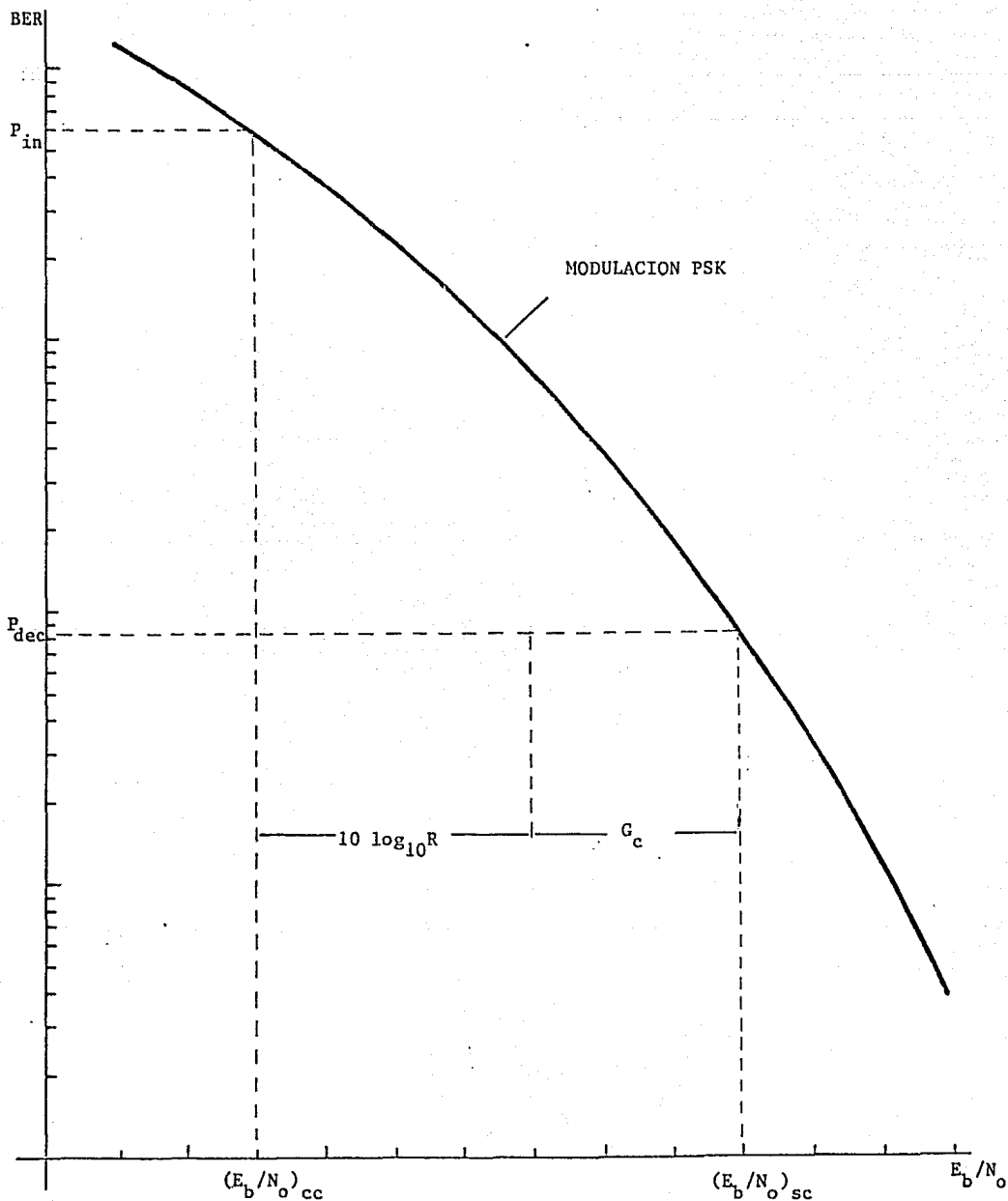


Figura 3.16. GANANCIA DE CODIFICACION PARA UN CANAL RGBA CON MODULACION PSK.

Técnica	G_c (dB)
Viterbi, Dec. Dura R=1/2, K=7	3.6
Viterbi, Dec. Dura R=1/2, K=6	3.1
Viterbi, Dec. Dura R=1/2, K=5	2.8
Viterbi. Dec. Suave (3 bits) R=1/2, K=7	5.8
Viterbi. Dec. Dura R=1/3, K=7	4.0
Secuencial. Dec. Dura R=1/2, K=41	6.3
Golay (24,12)	2.3
BCH (128,112)	2.1
BCH (15,7)	0.9

Tabla 3.1 GANANCIA DE CODIFICACION PARA DIVERSAS TECNICAS DE DECODIFICACION. $P_B = 10^{-7}$, $(E_b/N_0)_{sc} = 11.3$ dB.

Tabla 3.2 Comparación entre las técnicas de decodificación de códigos convolucionales.

Técnica de decodificación	Ventajas	Desventajas
1. Decodificación secuencial	<ul style="list-style-type: none"> - El número de cálculos y la probabilidad de sobre flujo (borrados) son independientes de la longitud de restricción K. - Pueden emplearse códigos con longitudes grandes de restricción, obteniéndose distancias libres considerables. 	<ul style="list-style-type: none"> - El número de cálculos es una variable aleatoria. - El decodificador posee un almacenador de un número finito de símbolos de entrada, dando por resultado una probabilidad de sobre flujo (borrados). - Es una técnica subóptima de decodificación.
1.1 Algoritmo de pila (Z-J)	<ul style="list-style-type: none"> - Es rápido para códigos de tasas altas. 	<ul style="list-style-type: none"> - Es lento con códigos de tasas bajas. - Los cálculos involucran reordenamiento de pila. - El tamaño de la pila afecta el desempeño.
1.2 Algoritmo de pila múltiple	<ul style="list-style-type: none"> - Disminuye efectivamente la probabilidad de sobre flujo (borrados). 	<ul style="list-style-type: none"> - Son necesarias pilas adicionales, incrementando la complejidad del algoritmo.
1.3 Algoritmo de Fano	<ul style="list-style-type: none"> - Practicamente no requiere de almacenamiento. - Los cálculos no requieren de reordenamiento de pilas. - Es rápido para códigos de tasas moderadas. 	<ul style="list-style-type: none"> - Es lento para códigos de tasas altas - Extiende más ramas que el algoritmo de pila.

Tabla 3.2 Comparación entre las técnicas de decodificación de códigos convolucionales. (Cont.)

Técnica de decodificación	Ventajas	Desventajas
2. Algoritmo de Viterbi	<ul style="list-style-type: none"> - Es una técnica óptima de decodificación (máxima similitud). - Los cálculos son más sencillos que en la decodificación secuencial. - La complejidad del decodificador es menor para códigos con tasas bajas. - La ganancia del código se incrementa notablemente al introducir decisión suave. 	<ul style="list-style-type: none"> - Tiene la limitación práctica del tamaño de la memoria en el decodificador (longitudes de restricción de hasta 8 bits) - El número de cálculos y el tamaño de la memoria crecen exponencialmente con la longitud de restricción K. - Sólo pueden emplearse códigos con distancias libres moderadas.

CAPITULO 4. DISEÑO DE UN DECODIFICADOR DE VITERBI.

En este capítulo se exponen los principios básicos del diseño de un decodificador de Viterbi, las alternativas de implementación del mismo, y una descripción detallada del diseño junto con los requerimientos de operación. Cabe señalar que este trabajo constituye un paso más hacia la construcción de un esquema de codificación de canal para sistemas de transmisión de datos por satélite, utilizando estaciones terrenas en la banda de los 12/14 GHz.

4.1. CONSIDERACIONES DE DISEÑO.

Antes de diseñar un esquema de codificación-decodificación de canal, conocido como CODEC, de acuerdo a Freeman [9] los siguientes factores deben tomarse en cuenta:

1. Desempeño adecuado para el tipo de modulación, canal y técnica de codificación. Por ejemplo, la probabilidad de error contra varios valores de relación energía por bit-a-ruido, E_b/N_0 .
2. Requerimientos de interfaz con el MODEM, como puede ser el grado de confiabilidad de la decisión del demodulador (es decir, decisión "dura" o "suave").
3. Requerimientos de sincronización, esto es, el método para determinar el inicio de un bloque u otra forma de agrupamiento del código.
4. Requerimientos en la tasa de datos.

5. Requerimientos de ambigüedad de fase del MODEM. Algunos decodificadores pueden compensar los efectos de una ambigüedad de fase de 90 o 180 grados presente en los MODEMS con BPSK o QPSK, los cuales obtienen una referencia de fase a partir de la señal de datos.

6. Requerimientos de retraso codificador-decodificador, es decir, el retraso en bits desde el instante en que un bit de información entra al codificador hasta el momento en el que sale del decodificador.

7. Retraso de arranque del decodificador.

8. Requerimientos de prueba interconstruida, esto es, que el CODEC sea capaz de realizar un autodiagnóstico del funcionamiento de sus circuitos.

9. Requerimientos de empaquetado. El decodificador podría estar en una tarjeta para insertarse en un MODEM ya existente, o bien podría ofrecerse en un gabinete por separado. Los requerimientos térmicos y de potencia deben también ser especificados.

10. Complejidad y costo de implementación del esquema contra la ganancia de codificación alcanzada. Puede obtenerse la misma ganancia mediante otros métodos que sean más económicos ?

De lo expuesto en el capítulo 3, resulta evidente que el algoritmo de decodificación de Viterbi es una técnica atractiva para la decodificación de códigos convolucionales, en el marco de las comunicaciones de datos por satélite, como ha sido demostrado a lo largo de los años [12].

Observando las características, y técnicas de decodificación empleadas, de diversas estaciones terrenas que ofrecen servicios de transmisión de datos en la banda Ku (Tabla 4.1), resulta claro que la decodificación de Viterbi es una técnica ampliamente utilizada en esta aplicación.

Lo anterior coincide con la ganancia de esta técnica comparada con otras (Tabla 3.1), se tiene una gran ganancia de codificación con una técnica relativamente fácil de implementar en comparación con otras técnicas de complejidad semejante (como el caso de los códigos de bloque). Además, desde el punto de vista del sistema, la misma ganancia (3.6 dB) podría obtenerse, por ejemplo, si se utilizara una antena con un diámetro de 7 m, en lugar de una de 5 m.

Para el caso de la estación terrena a diseñar, fue especificada una probabilidad de error (o BER) de 10^{-7} para una relación Eb/No menor que 8.3 dB, y una tasa de información (antes del codificador) de 16000 bits/segundo. Para este punto de operación, un código de tasa 1/2, longitud de restricción 7, y decodificación de Viterbi con decisión dura, fue seleccionado como el que proporciona un desempeño adecuado para este caso.

Debe notarse que con esta selección, no hay necesidad de calcular las funciones de similitud más complicadas que se necesitarían con decisiones suaves. La elección de la tasa 1/2 permite el uso de la disponibilidad de una expansión moderada del ancho de banda, debida a la limitación en potencia del satélite. Esta tasa permite también reducir la complejidad del decodificador, y podría aprovecharse el hecho de que, a partir de este esquema, se pueden obtener

TABLA 4.1 ESTACIONES TERRENAS QUE OFRECEN SERVICIOS DE TRANSMISION DE DATOS POR SATELITE, EN LA BANDA Ku.

EQUIPO	MODULACION	ENTREFAZ	TASA DE DATOS	CODIGO EMPLEADO	BER	Eb/No	GANANCIA
HARAIS, 8003 LOW RATE MODEM	QPSK (RPSK)	CCITT REC. V.35 RS-232C	32, 46, 64 128, 256 KBPS	VITERBI 1/2	10^{-6}	—	5.1
				VITERBI 3/4	10^{-6}	—	4
				UMERAL 3/4	10^{-6}	—	2.6
				UMERAL 7/8	10^{-6}	—	2.6
MARCONI, P7001, 11/14 GHz SATELLITE DATA TERMINAL	BPSK	—	512 KBPS- 2 MBPS	VITERBI 1/2 (SOFT DEC. 3-BIT)	10^{-9} 10^{-5}	— 11.9	—
AVANTEK, ASAT-1214 Ku-BAND SMALL APERTURE TRANSMIT/RECEIVE TERMINAL	BPSK (QPSK)	—	9.6 KBPS- 1,544 MBPS	VITERBI 1/2	10^{-7}	6.0	—
MICROTEL, 83301 SINGLE CHANNEL REMOTE TERMINAL (SCRT)	QPSK	RS-232C TIPO D	ASINC.: 110,134.5,150, 300,1200,2400,4800, 9600 BPS SIN.: 1200,2400,4800, 9600 BPS	VITERBI 1/2 (SOFT DEC. 3-BIT)	10^{-8} 10^{-6}	— (UMERAL REC.)	— —
SCIENTIFIC ATLANTA, IBT-1200 INTEGRATED BUSINESS TERMINAL	BPSK (QPSK)	—	56 KBPS- 2,048 MBPS	SECUENCIAL 1/2	10^{-7}	7.0	—
				SECUENCIAL 3/4	10^{-7}	7.0	—
				SECUENCIAL 7/8	10^{-7}	7.0	—
				(TODOS SOFT DEC. 3-BIT)			
CALIFORNIA MICROWAVE, SERIE KB	BPSK	RS-422	56 KBPS- 256 KBPS	VITERBI 1/2 (SOFT. DEC. 3-BIT)	10^{-7}	7.8	—
VITALINK, 1800	QPSK	CCITT REC. V.35 RS-422 RS-232C DSI-IEEE 802.3	4.8 KBPS- 224 KBPS	— 1/2, 3/4, 7/8	10^{-7}	—	—

tasas de código mayores, mediante el empleo de códigos truncados [3].

4.2. ALTERNATIVAS DE IMPLEMENTACION.

Una vez elegidos todos los parámetros del código, y la técnica de decodificación a utilizar, debieron estudiarse las diversas alternativas de implementación del algoritmo de Viterbi. En general, se tienen dos formas de realizar la implementación de cualquier decodificador: serie y paralelo.

En el caso de una implementación en paralelo del algoritmo de Viterbi, se requieren $2^{(k-m)}$ unidades de procesamiento (que realicen las operaciones de cálculo y actualización de métricas, comparación, selección y actualización de trayectorias sobrevivientes) para cada uno de los estados del codificador. Con este esquema pueden obtenerse muy altas tasas de transmisión (hasta de 10 Mbps), con el costo de un gran número de elementos lógicos y de memoria. Esta implementación es adecuada para diseños con tecnología de muy alta escala de integración (VLSI), en los que puede obtenerse un decodificador completo en un sólo circuito integrado, mientras que los diseños tradicionales requerirían de espacio excesivo y resultarían imprácticos.

Por otro lado, la implementación serie permite un ahorro significativo en el número de dispositivos requeridos, pero con el costo de reducir la tasa de transmisión (hasta cientos de Kbps). En este tipo de esquema solo una unidad central de procesamiento es necesaria para todas las $2^{(k-m)}$ trayectorias sobrevivientes. Este tipo de implementación, o arquitectura, del decodificador presenta además la ven-

taja de ser capaz de incrementar la tasa de transmisión, mediante una configuración híbrida - serie/paralelo, en la que varias unidades de procesamiento comparten la tarea de decodificación.

Como primera aproximación, se realizó la implementación serie del algoritmo de Viterbi, para el código de tasa 1/2 y longitud de restricción 7, usando el microprocesador de 16 bits MC68010 de Motorola [10]. Con esta implementación, mediante un programa escrito en lenguaje ensamblador, se obtuvo sin embargo una tasa de transmisión de 495 bits/segundo. Debe notarse que en un trabajo anterior [18] se reporta una tasa de transmisión de 50 bits/segundo, utilizando el microprocesador de 8 bits Z-80 de Zilog.

En la implementación se empleó una tabla de distancias de Hamming y una tabla de bits codificados, almacenadas en memoria, y dos secciones de memoria para el almacenamiento y actualización de las trayectorias y sus distancias. Cuando las distancias de las trayectorias sobrevivientes sobrepasan del valor 32, son normalizadas, debido a que se tiene un número finito de bits para su representación.

Se tomó una longitud de decodificación de 31 bits, anadiendo al inicio de cada paso de decodificación un bit, recorriendo las trayectorias sobrevivientes en un bit a la izquierda, y tomando el 32o. bit de la primera trayectoria como el decodificado. La longitud es adecuada debido a que, como se recordará, las trayectorias sobrevivientes tienden a ser iguales después de 4 a 5 longitudes de restricción del código (ver inciso 3.3). En la figura 4.1 se muestra el listado del programa, subrutinas, tablas y parámetros empleados.

001000	4FF87FFE	LEA.L	\$00007FFE,A7
001004	41F80920	LEA.L	\$00000920,A0
001008	47F80C00	LEA.L	\$00000C00,A3
00100C	43F80902	LEA.L	\$00000902,A1
001010	45F80D00	LEA.L	\$00000D00,A2
001014	4283	CLR.L	D3
001016	3238090A	MOVE.W	\$0000090A,D1
00101A	2083	MOVE.L	D3,(A0)
00101C	6100020A	BSR.L	\$001228
001020	14C2	MOVE.B	D2,(A2)+
001022	06430001	ADD.W	#1,D3
001026	D0FC0004	ADD.W	#4,A0
00102A	51C9FFEE	DBF.L	D1,\$00101A
00102E	41F80920	LEA.L	\$00000920,A0
001032	47F80A20	LEA.L	\$00000A20,A3
001036	43F80C00	LEA.L	\$00000C00,A1
00103A	45F80D00	LEA.L	\$00000D00,A2
00103E	49F80D40	LEA.L	\$00000D40,A4
001042	4BF80FF0	LEA.L	\$00000FF0,A5
001046	4202	CLR.B	D2
001048	2010	MOVE.L	(A0),D0
00104A	E388	LSL.L	#1,D0
00104C	6506	BCS.S	\$001054
00104E	610001B0	BSR.L	\$001200
001052	6004	BRA.S	\$001058
001054	610001BE	BSR.L	\$001214
001058	20C0	MOVE.L	D0,(A0)+
00105A	323C003E	MOVE.W	#62,D1
00105E	2010	MOVE.L	(A0),D0
001060	E388	LSL.L	#1,D0
001062	20C0	MOVE.L	D0,(A0)+
001064	51C9FFF8	DBF.L	D1,\$00105E
001068	90FC0100	SUB.W	#256,A0
00106C	3A3C001F	MOVE.W	#31,D5
001070	2C3C00000001	MOVE.L	#1,D6
001076	3805	MOVE.W	D5,D4
001078	E34C	LSL.W	#1,D4
00107A	D846	ADD.W	D6,D4
00107C	3605	MOVE.W	D5,D3
00107E	06430020	ADD.W	#32,D3
001082	3E04	MOVE.W	D4,D7
001084	E54C	LSL.W	#2,D4
001086	3C45	MOVE.W	D5,A6
001088	E54D	LSL.W	#2,D5
00108A	20305000	MOVE.L	0(A0,D5.W),D0
00108E	3A0E	MOVE.W	A6,D5
001090	BD40	EOR.W	D6,D0
001092	3C43	MOVE.W	D3,A6
001094	E548	LSL.W	#2,D3
001096	22303000	MOVE.L	0(A0,D3.W),D1
00109A	360E	MOVE.W	A6,D3
00109C	BD41	EOR.W	D6,D1
00109E	0240007F	AND.W	#127,D0
0010A2	0241007F	AND.W	#127,D1

Figura 4.1. (a) LISTADO DEL PROGRAMA QUE REALIZA EL ALGORITMO DE VITERBI MEDIANTE EL MICROPROCESADOR MC68000.

0010A6	10310000	MOVE.B	0(A1,D0.W),D0
0010AA	12311000	MOVE.B	0(A1,D1.W),D1
0010AE	E548	LSL.W	#2,D0
0010B0	E549	LSL.W	#2,D1
0010B2	D0780FEE	ADD.W	#00000FEE,D0
0010B6	D2780FEE	ADD.W	#00000FEE,D1
0010BA	10350000	MOVE.B	0(A5,D0.W),D0
0010BE	12351000	MOVE.B	0(A5,D1.W),D1
0010C2	D0325000	ADD.B	0(A2,D5.W),D0
0010C6	D2323000	ADD.B	0(A2,D3.W),D1
0010CA	B200	CMF.B	D0,D1
0010CC	6E0E	BGT.S	#0010DC
0010CE	E548	LSL.W	#2,D3
0010D0	27B030004000	MOVE.L	0(A0,D3.W),0(A3,D4.W)
0010D6	19817000	MOVE.B	D1,0(A4,D7.W)
0010DA	6010	BRA.S	#0010EC
0010DC	3C45	MOVE.W	D5,A6
0010DE	E54D	LSL.W	#2,D5
0010E0	27B050004000	MOVE.L	0(A0,D5.W),0(A3,D4.W)
0010E6	3A0E	MOVE.W	A6,D5
0010E8	19807000	MOVE.B	D0,0(A4,D7.W)
0010EC	BDB34000	EOR.L	D6,0(A3,D4.W)
0010F0	0C3400007000	CMF.B	#0,0(A4,D7.W)
0010F6	6D04	BLT.S	#0010FC
0010FB	08C20000	BSET	#0,D2
0010FC	51CEFF78	DBF.L	D6,#001076
001100	51C9FF6E	DBF.L	D5,#001070
001104	C14B	EXG	A0,A3
001106	C54C	EXG	A2,A4
001108	4A02	TST.B	D2
00110A	6600FF3A	BNE.L	#001046
00110E	323C003F	MOVE.W	#63,D1
001112	063200801000	ADD.B	#128,0(A2,D1.W)
001118	063400801000	ADD.B	#128,0(A4,D1.W)
00111E	51C9FFF2	DBF.L	D1,#001112
001122	6000FF22	BRA.L	#001046

Figura 4.1. (a) LISTADO DEL PROGRAMA QUE REALIZA EL ALGORITMO DE VITERBI MEDIANTE EL MICROPROCESADOR MC68000.
(Continuación).

001200	48E7C180	MOVEM.L DO-D1/D7-A0, -(A7)
001204	1E3C00FB	MOVE.B #248,D7
001208	103C0030	MOVE.B #48,D0
00120C	4E4E	TRAP #14
00120E	4CDF0183	MOVEM.L (A7)+,DO-D1/D7-A0
001212	4E75	RTS
001214	48E7C180	MOVEM.L DO-D1/D7-A0, -(A7)
001218	1E3C00FB	MOVE.B #248,D7
00121C	103C0031	MOVE.B #49,D0
001220	4E4E	TRAP #14
001222	4CDF0183	MOVEM.L (A7)+,DO-D1/D7-A0
001226	4E75	RTS
001228	48E7CC40	MOVEM.L DO-D1/D4-D5/A1, -(A7)
00122C	4240	CLR.W DO
00122E	3A380908	MOVE.W #00000908,D5
001232	E348	LSL.W #1,D0
001234	2210	MOVE.L (A0),D1
001236	C259	AND.W (A1)+,D1
001238	38380906	MOVE.W #00000906,D4
00123C	E249	LSR.W #1,D1
00123E	6404	BCC.S #001244
001240	0A400001	EOR.W #1,D0
001244	51CCFFF6	DBF.L D4,#00123C
001248	51CDFFE8	DBF.L D5,#001232
00124C	16C0	MOVE.B DO,(A3)+
00124E	43FB0FF0	LEA.L #00000FF0,A1
001252	32380FEE	MOVE.W #00000FEE,D1
001256	E548	LSL.W #2,D0
001258	B340	EOR.W D1,D0
00125A	14310000	MOVE.B 0(A1,DO.W),D2
00125E	4CDF0233	MOVEM.L (A7)+,DO-D1/D4-D5/A1
001262	4E75	RTS

000FF0 00 01 01 02 01 00 02 01 01 02 00 01 02 01 01 00

000900 00 3F 00 6D 00 4F 00 06 00 01 00 7F FF FF FF FF .?.m.D.....

Figura 4.1. (b) LISTADO DE LAS SUBROUTINAS Y TABLAS EMPLEADAS EN LA REALIZACION DEL ALGORITMO DE VITERMI MEDIANTE EL MICROPROCESADOR MC68000.

Ya que el esquema anterior no cumple con los requerimientos de funcionamiento establecidos (esto es, una tasa de transmisión de 32000 bits/segundo), se decidió diseñar un decodificador de Viterbi con arquitectura serie y basado en circuitos integrados TTL, que permitiera alcanzar la tasa requerida. En el siguiente inciso se presenta la descripción de las diferentes partes y circuitos empleados en el decodificador propuesto.

4.3. DESCRIPCION DEL DECODIFICADOR.

El decodificador diseñado consta de los siguientes bloques:

1. Circuito de adquisición.
2. Circuito ACS (Añadir-Comparar-Seleccionar).
3. Memoria de trayectorias y métricas.
4. Circuito controlador de memoria.
5. Circuito normalizador de métricas.
6. Circuito de sincronización.
7. Circuito de salida.

En la figura 4.2 se muestra un diagrama de bloques del decodificador. A continuación se describirán las funciones y estructura de cada una de las partes del decodificador, y su interrelación empleando diagramas de tiempo.

1. Circuito de adquisición.

La función del circuito de adquisición es la de recibir los bits de canal entregados por el demodulador, y consta básicamente de

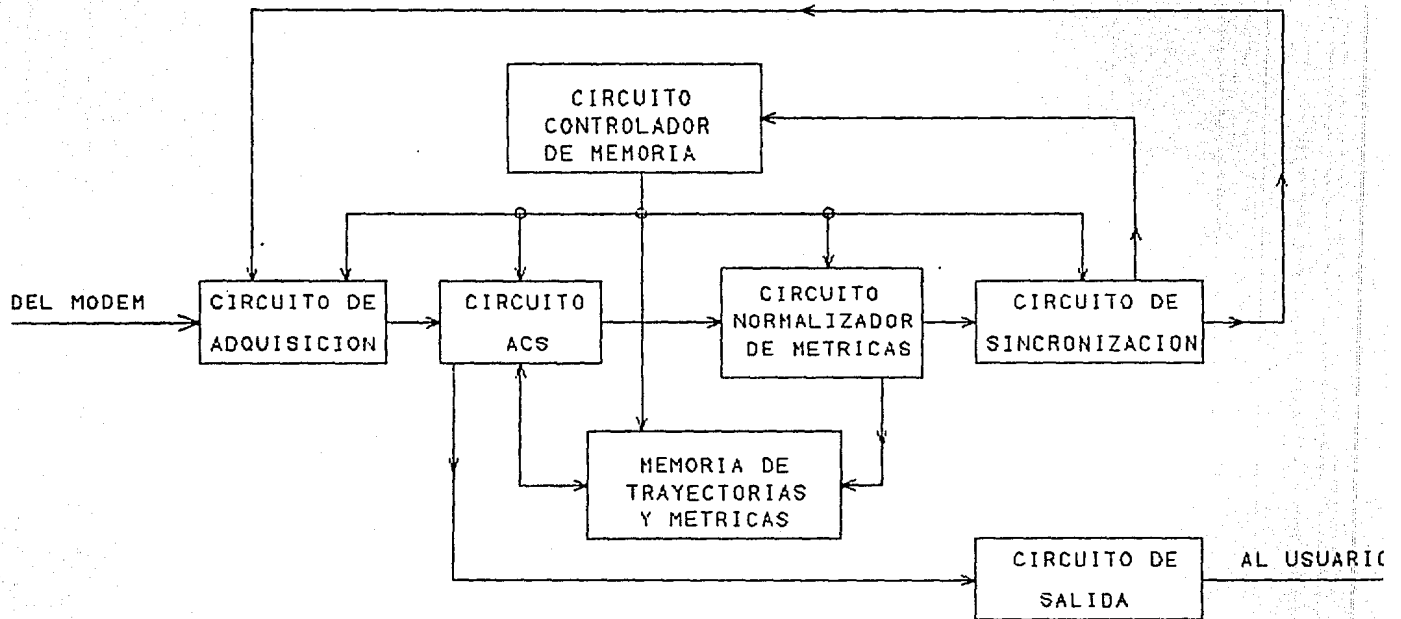


FIGURA 4.2. DIAGRAMA DE BLOQUES DEL DECODIFICADOR DE VITERBI.

un almacenador ("buffer") de 8 bits, del cual se extraen los bits correspondientes a la rama recibida. Es necesario obtener un reloj de referencia a la tasa de transmisión, y el propio circuito deriva esta señal a partir del reloj (de 2.048 MHz) del decodificador. La figura 4.3 muestra este circuito.

2. Circuito ACS.

La figura 4.4 presenta el diagrama de bloques del circuito ACS. Este es el corazón del decodificador, donde se realizan las funciones de cálculo de la distancia entre ramas, actualización de la distancia acumulada (sumarla a la distancia anterior de la trayectoria correspondiente) y selección de la trayectoria sobreviviente.

Se aprovecha la estructura de enrejado para localizar en la memoria de trayectorias y métricas aquellas trayectorias que llegan al mismo estado, y para almacenar la trayectoria sobreviviente en la localidad correspondiente a dicho estado. Es decir, existe una correspondencia de uno-a-uno entre las direcciones de las localidades de memoria y los estados del codificador. Lo anterior es ilustrado en la figura 4.5.

El funcionamiento de este circuito es como sigue: las trayectorias a extender son almacenadas en dos registros y extendidas empleando la señal D0, se realiza la codificación de los bits menos significativos de las trayectorias mediante dos réplicas gemelas del codificador (Fig. 4.6) y se calcula su distancia a la rama recibida mediante un circuito evaluador de distancia (Fig. 4.7).

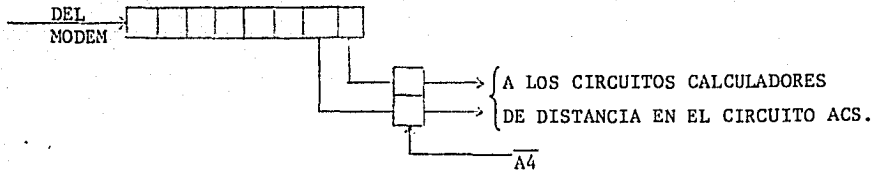


Figura 4.3. CIRCUITO DE ADQUISICION.

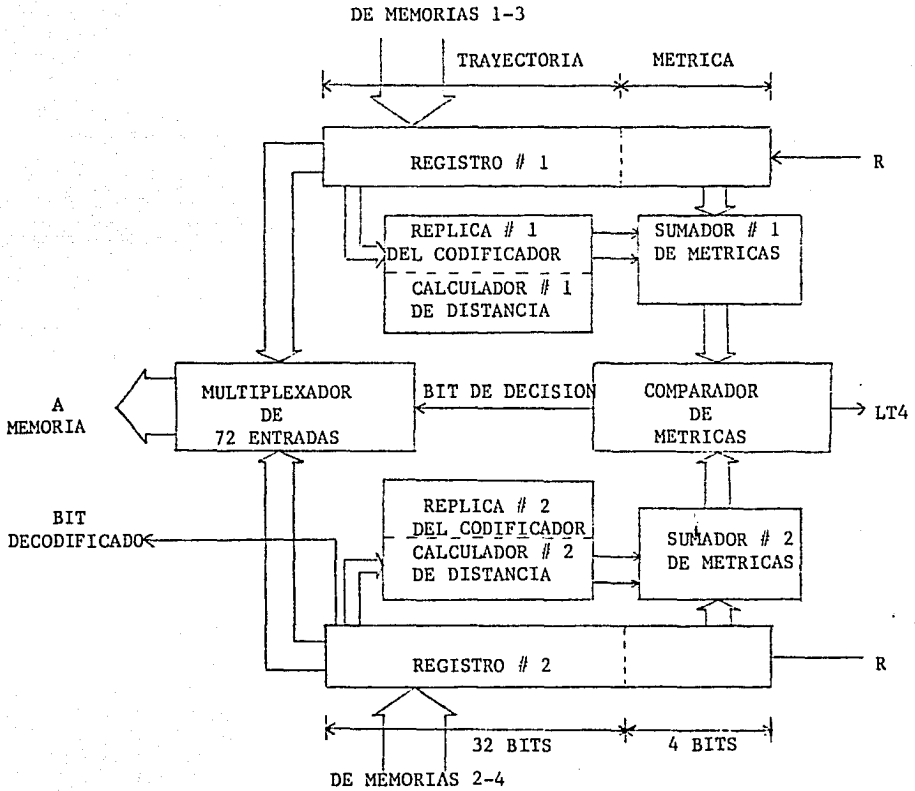
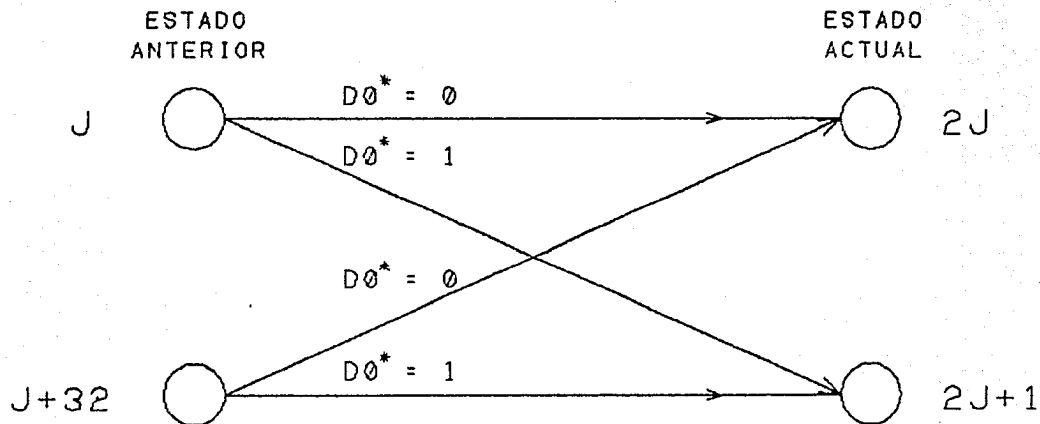


Figura 4.4. DIAGRAMA DE BLOQUES DEL CIRCUITO ACS.



CICLO PAR ($A5=0$):

MEMORIA 1 \longrightarrow J
 MEMORIA 2 \longrightarrow J+32

MEMORIAS 3-4 \longrightarrow 2J, 2J+1

CICLO IMPAR ($A5=1$):

MEMORIA 3 \longrightarrow J
 MEMORIA 4 \longrightarrow J+32

MEMORIAS 1-2 \longrightarrow 2J, 2J+1

FIGURA 4.5. RELACION ENTRE EL ENREJADO Y LAS DIRECCIONES DE LA MEMORIA DE TRAYECTORIAS Y METRICAS.

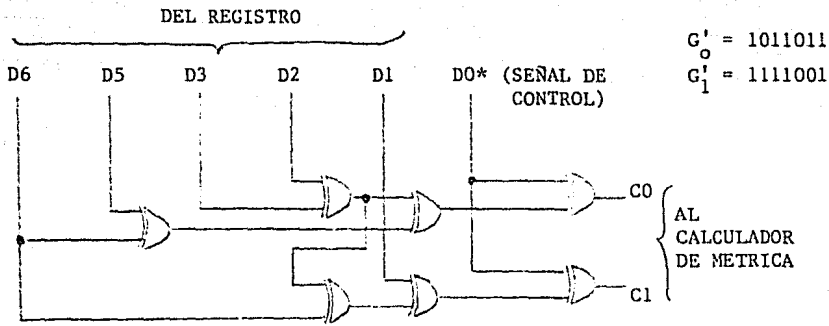


Figura 4.6 CIRCUITO REPLICA DEL CODIFICADOR

CALCULADOR DE METRICA:

RAMA CODIFICADA		RAMA RECIBIDA		METRICA	
A	B	C	D	M0	M1
0	0	0	0	0	0
		1	0	0	1
		0	1	0	1
		1	1	1	0
0	1	0	0	0	1
		1	0	1	0
		0	1	0	0
		1	1	0	1
1	0	0	0	0	1
		1	0	0	0
		0	1	1	0
		1	1	C	1
1	1	0	0	1	0
		1	0	0	1
		0	1	0	1
		1	1	0	0

$$M1 = A \oplus B \oplus C \oplus D$$

$$M0 = (A \oplus C) \cdot (B \oplus D)$$

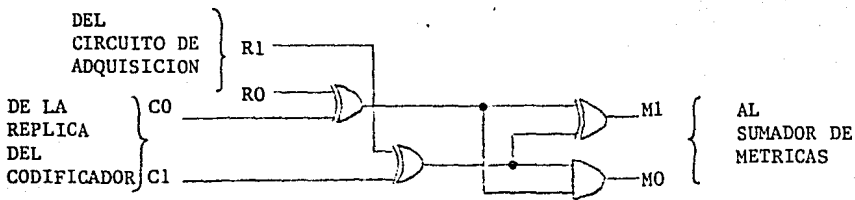


Figura 4.7 CIRCUITO CALCULADOR DE DISTANCIA

Posteriormente se suman las distancias obtenidas con aquellas de las trayectorias en cuestión (disponibles en los registros), mediante un sumador binario completo (Fig. 4.8), y se comparan las dos distancias mediante un comparador de magnitud de 4 bits (Fig. 4.9). Finalmente, el resultado de la comparación sirve como señal de selección para un multiplexador dos-a-uno de 72 entradas (Fig. 4.10).

3. Memoria de trayectorias y métricas.

Cada localidad de memoria es de 36 bits de longitud, dividida en dos partes correspondientes a la trayectoria del árbol (32 bits) y a su métrica asociada (4 bits). La memoria se divide en 4 secciones que corresponden a los grupos de nodos que coinciden en el mismo estados y que contienen 32 trayectorias cada una (Fig. 4.11). Esta estructura facilita el manejo de memoria, que de otra manera sería más lento (ya que se necesitarían emplear dos ciclos de lectura y dos ciclos de escritura).

4. Circuito controlador de memoria.

La función del circuito controlador de memoria es la de obtener la dirección tanto de las localidades de memoria en las que se encuentran las trayectorias a extender y comparar, como la de donde se escribirá la trayectoria sobreviviente en cada comparación. Se emplea un contador binario de 8 bits para generar las señales necesarias de direccionamiento y control.

Durante un ciclo par de decodificación ($A5=0$) las memorias 1 y 2 contienen las trayectorias sobrevivientes a extender y comparar, y

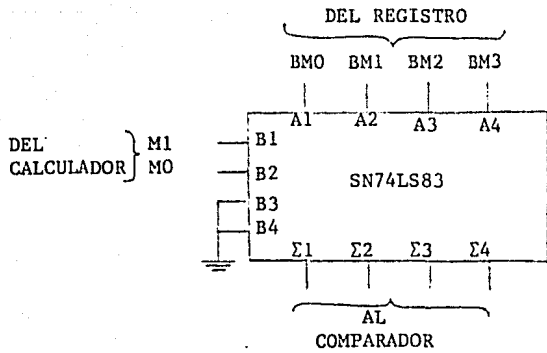


Figura 4.8. SUMADOR DE METRICAS

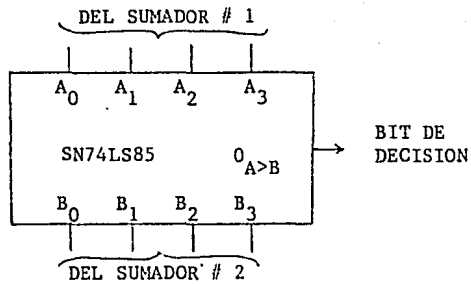


Figura 4.9. COMPARADOR DE METRICAS

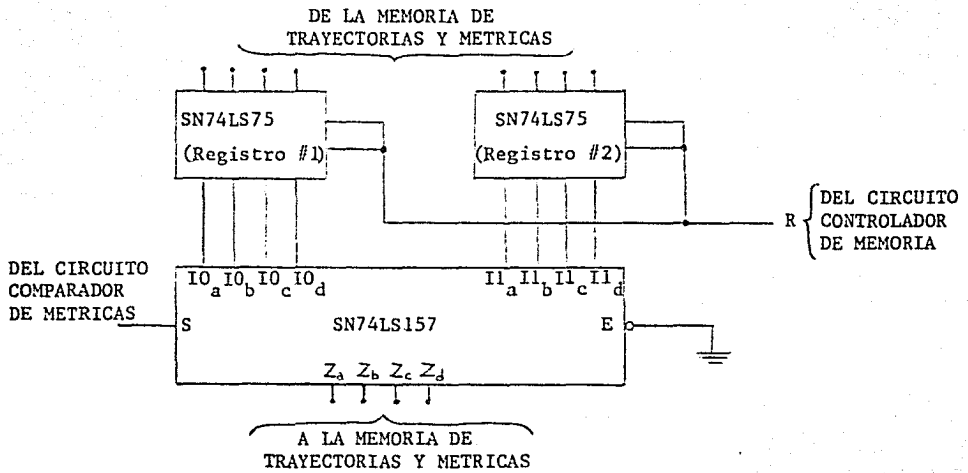


Figura 4.10 MULTIPLEXADOR 2-a-1 DE 72 ENTRADAS (SE MUESTRA UNA DE LAS 9 SECCIONES IDENTICAS QUE LO COMPONEN).

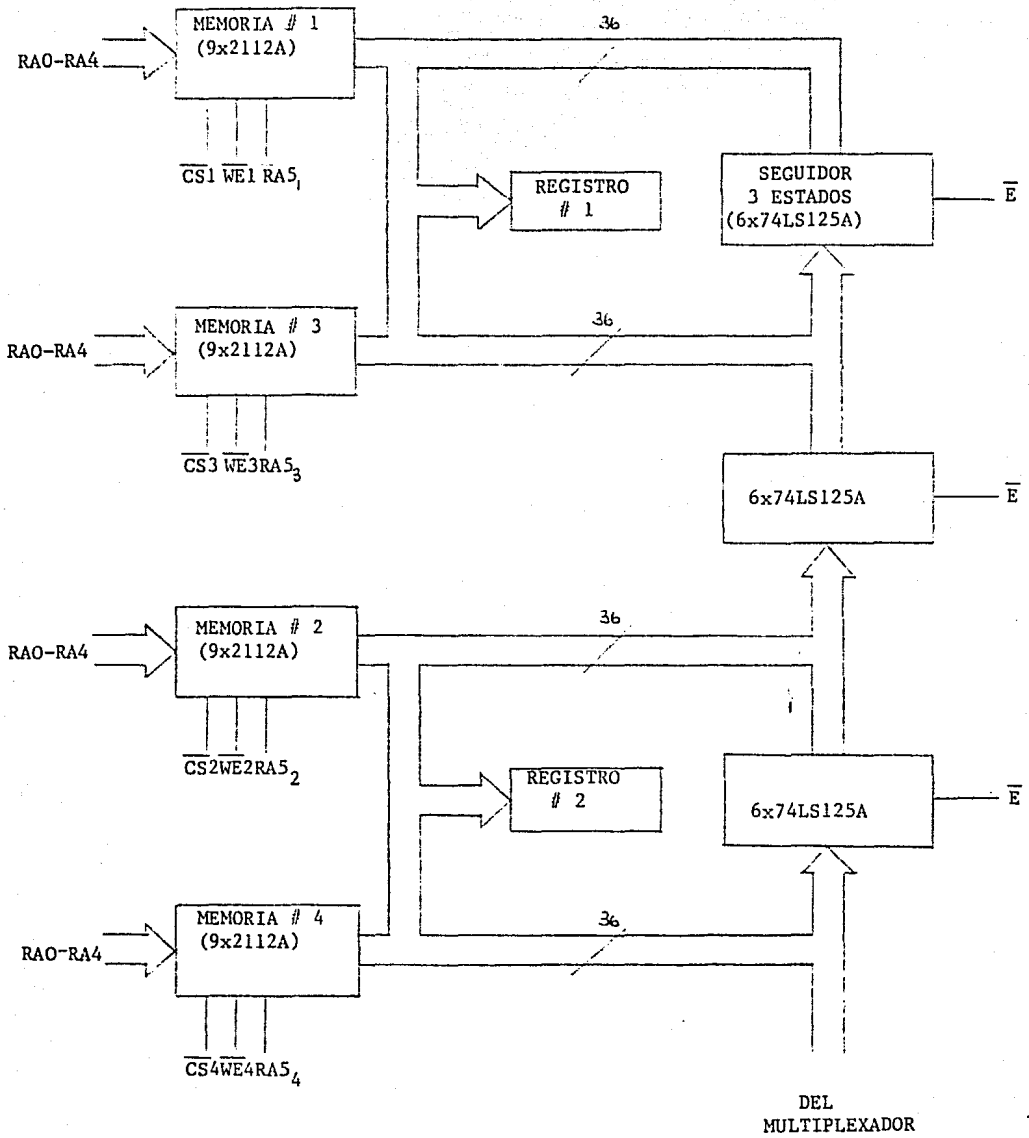


Figura 4.11. MEMORIA DE TRAYECTORIAS Y METRICAS

la ganadora (sobreviviente) es almacenada secuencialmente en las memorias 3 y 4 (Ver figura 4.5). En un ciclo impar, los papeles de las memorias 1-2 se cambian con las memorias 3-4. En la figura 4.12 se muestra el direccionamiento en la conmutación de memorias y las figuras 4.13 y 4.14 presentan los circuitos de control para las memorias 1-2 y 3-4.

5. Circuito normalizador de métricas.

La función de este circuito es la de evitar que las métricas (distancias de Hamming) crezcan indefinidamente y provoquen errores de saturación, ya que se tienen 4 bits para representar la métrica acumulada de cada trayectoria. Para ello, debe contarse con una señal que indique el momento en el que todas las métricas sobrepasan del valor 4 (determinado mediante simulación por computadora, y necesario para el circuito de sincronización), para entonces "normalizar" las métricas de las trayectorias sobrevivientes, regresándolas a un valor tal que la menor métrica tenga el valor cero.

En la figura 4.15 se muestra el diagrama de tiempos de las señales requeridas para efectuar la normalización, y en la figura 4.16 se presentan el circuito normalizador de métricas y el circuito que genera la señal que indica la necesidad de normalización (FMT).

6. Circuito de sincronización.

Mediante simulaciones realizadas empleando el simulador del algoritmo de Viterbi en computadora (cuyo diagrama de flujo se presenta en la figura 3.9, del capítulo 3), se determinó que la tasa de

DIRECCIONES DE MEMORIA

(a) Ciclo par de decodificación
A5=0:

	<u>RA0</u>	<u>RA1</u>	<u>RA2</u>	<u>RA3</u>	<u>RA4</u>	<u>RA5</u>	<u>WE</u>	<u>Estados del enrejado</u>
Memorias 1 y 2	A0	A1	A2	A3	A4	S*	1	(J y J + 32)
Memorias 3 y 4	D0*	A0	A1	A2	A3	A4	0	(2J y 2J+1)

(b) Ciclo impar de decodificación
A5=1:

	<u>RA0</u>	<u>RA1</u>	<u>RA2</u>	<u>RA3</u>	<u>RA4</u>	<u>RA5</u>	<u>WE</u>	<u>Estados del enrejado</u>
Memorias 1 y 2	D0*	A0	A1	A2	A3	A4	0	(J y J + 32)
Memorias 3 y 4	A0	A1	A2	A3	A4	S*	1	(2J y 2J+1)

donde: S*=0 Memorias 1 y 3
1 Memorias 2 y 4

RA0-RA5: Señales de dirección de la memoria de trayectorias y métricas

WE=0 Escribir
1 Leer

CS1-4=0 Memoria habilitada
1 Memoria deshabilitada

SEÑALES DE CONTROL

Memorias 1 y 2:

$$RA0 = A0\overline{A5} + D0*A5$$

$$RA1 = A1\overline{A5} + A0A5$$

$$RA2 = A2\overline{A5} + A1A5$$

$$RA3 = A3\overline{A5} + A2A5$$

$$RA4 = A4\overline{A5} + A3A5$$

$$RA5 = S*\overline{A5} + A4A5$$

$$\overline{WE} = \overline{A5}$$

$$\overline{CS1} = (\overline{\phi} + D0*)\overline{A5} + (\phi + A4)A5$$

$$\overline{CS2} = (\overline{\phi} + D0*)\overline{A5} + (\phi + A4)A5$$

Memorias 3 y 4:

$$RA0 = D0*\overline{A5} + A0A5$$

$$RA1 = A0\overline{A5} + A1A5$$

$$RA2 = A1\overline{A5} + A2A5$$

$$RA3 = A2\overline{A5} + A3A5$$

$$RA4 = A4\overline{A5} + A4A5$$

$$RA5 = A4\overline{A5} + S*A5$$

$$\overline{WE} = \overline{A5}$$

$$\overline{CS3} = (\phi + A4)\overline{A5} + (\overline{\phi} + D0*)A5$$

$$\overline{CS4} = (\phi + A4)\overline{A5} + (\overline{\phi} + D0*)A5$$

Figura 4.12(a). DIRECCIONAMIENTO PARA LA CONMUTACION DE MEMORIAS DE TRAYECTORIAS Y METRICAS

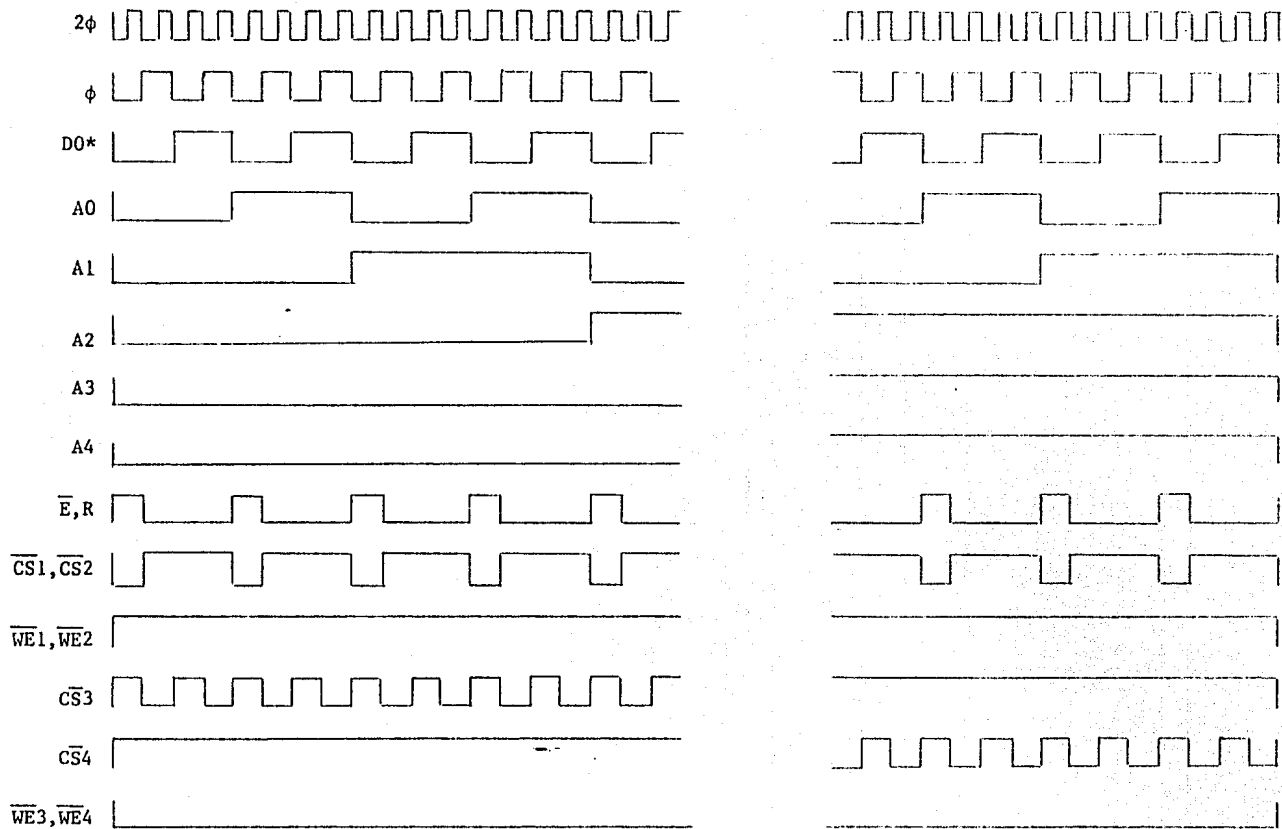
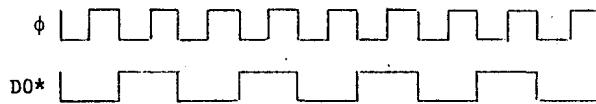


Figura 4.12 (b) SEÑALES DE CONTROL DURANTE UN CICLO PAR DE DECODIFICACION ($A_5=0$)



\longleftrightarrow \longleftarrow
 Tiempo de
 una comparación

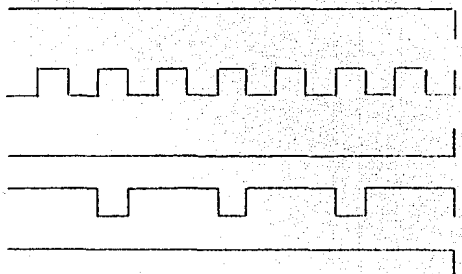
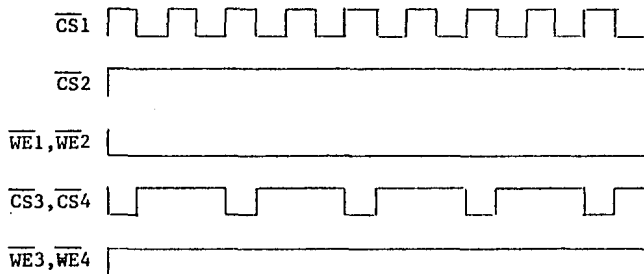


Figura 4.12 (c) SEÑALES DE CONTROL DURANTE UN CICLO IMPAR DE DECODIFICACION (A5=1)

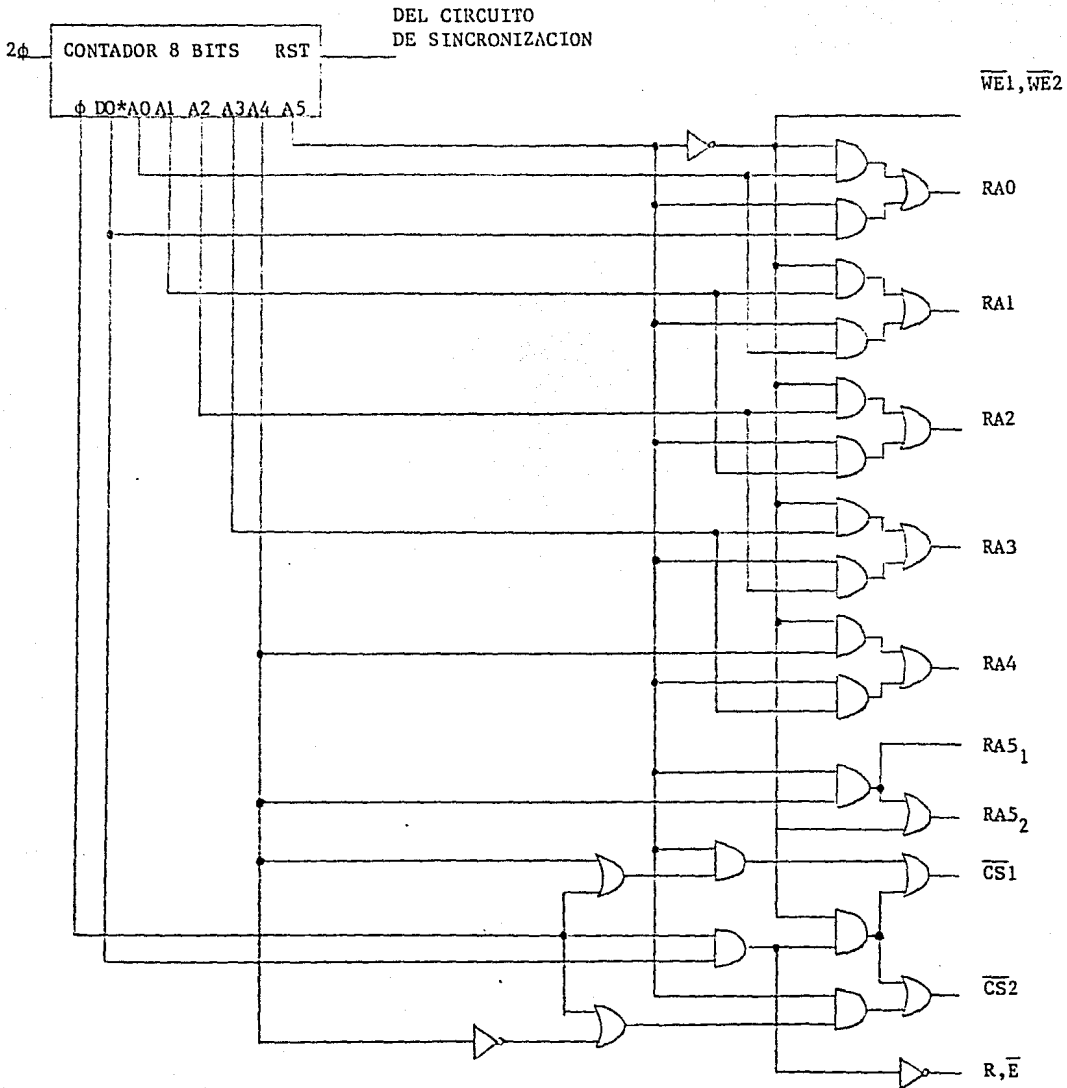


Figura 4.13. CIRCUITO DE CONTROL PARA LAS MEMORIAS 1 Y 2.

DEL CIRCUITO
DE SINCRONIZACION

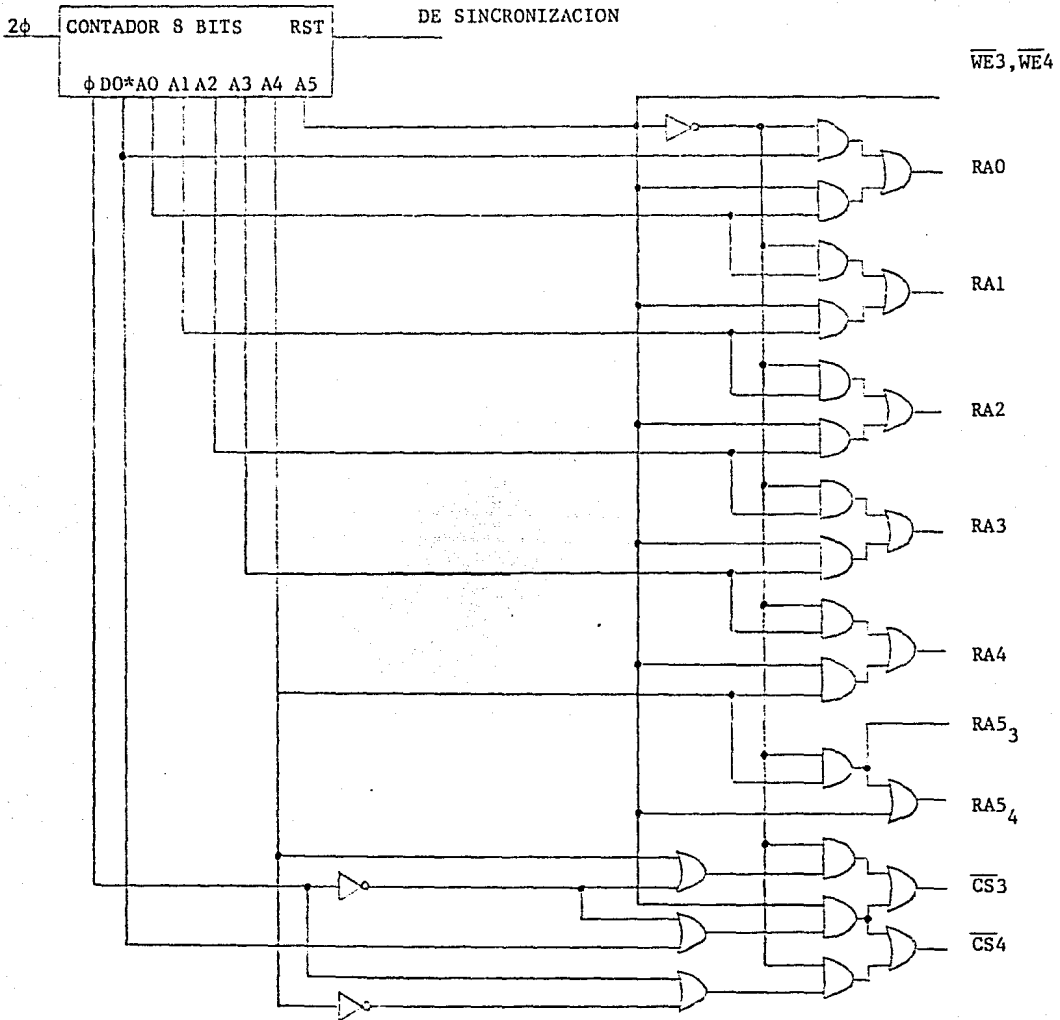
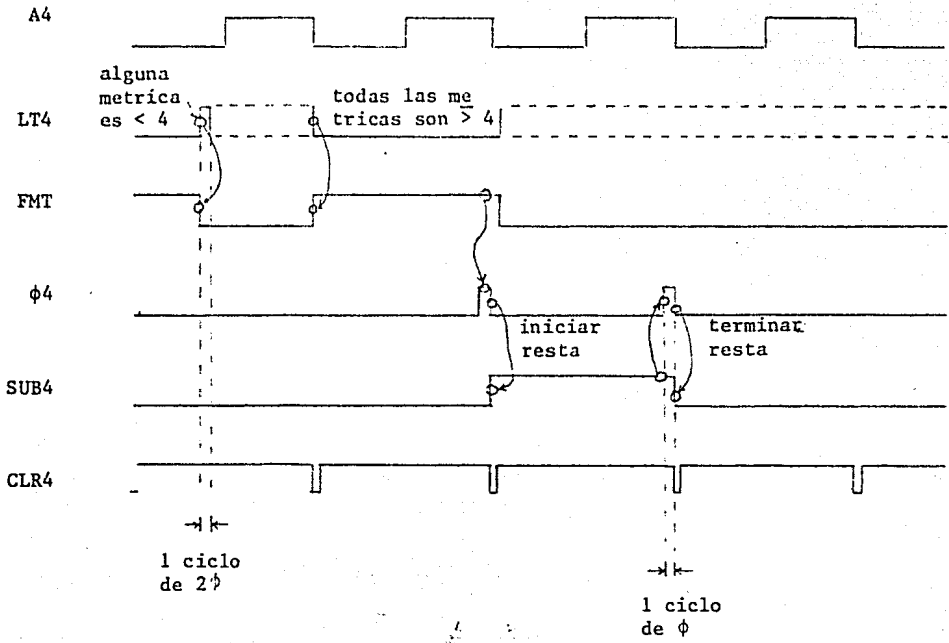


Figura 4.14 CIRCUITO DE CONTROL PARA LAS MEMORIAS 3 Y 4.



SEÑAL LT4:

METRICA	LT4
0000	1
0001	1
0010	1
0011	1
0100	0
⋮	⋮
⋮	⋮
1111	0
ABCD	

$$LT4 = \overline{AB} = \overline{(A+B)}$$

Figura 4.15. DIAGRAMA DE TIEMPOS PARA LA NORMALIZACION DE METRICAS

CIRCUITO NORMALIZADOR

$$\begin{aligned} \text{SUB}_4 = 0 & \quad \text{BM}^{(t+1)} = \text{BM}^{(t)} \\ & \quad \text{BM}^{(t+1)} = \text{BM}^{(t)} - 4 \end{aligned}$$

CIRCUITO PARA RESTAR 4:

BM	BM-4
0000	XXXX
1000	XXXX
0100	XXXX
1100	XXXX
0010	0000
1010	1000
0110	0100
1110	1100
0001	0010
0101	0110
1101	1110
0011	0001
1011	1001
0111	0101
1111	1101
ABCD	EFGH

$$\begin{aligned} E &= A & F &= B \\ G &= \bar{C} & H &= CD \end{aligned}$$

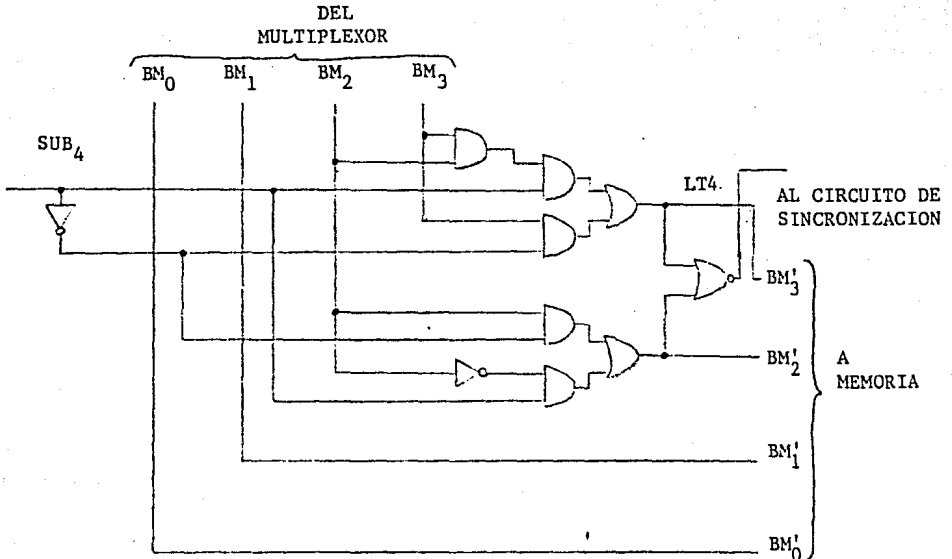


Figura 4.16 (a) CIRCUITO NORMALIZADOR DE DISTANCIAS

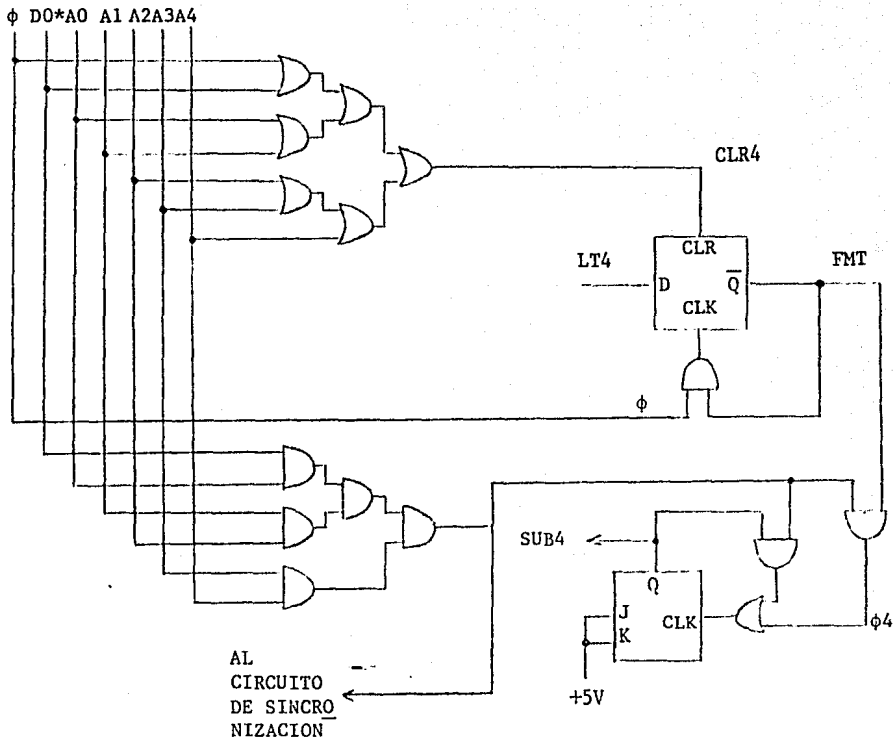


Figura 4.16 (b) CIRCUITO GENERADOR DE LA SEÑAL DE NORMALIZACION

crecimiento en condiciones de no sincronía es mucho mayor que aquella que se observa cuando el decodificador procesa ramas sincronizadas.

Específicamente, cuando el decodificador está sincronizado con las ramas recibidas, la tasa de crecimiento de la distancia crece conforme a una recta de pendiente np (De acuerdo a la ley de los grandes números), donde p es la probabilidad de error del canal. Cuando no existe la sincronía, dicha tasa es de aproximadamente $1/4$, como se muestra en las figuras 4.17 y 4.18 para el código convolucional de tasa $1/2$ y longitudes de restricción 3 y 7, respectivamente.

Por lo tanto, se diseñó un circuito que detectara un número excesivo de normalizaciones de métrica, usando la señal SUB4 del circuito normalizador de métricas. Se determinó, mediante simulaciones por computadora, que cuando el número de normalizaciones en 128 ciclos de decodificación (es decir, en 128 bits decodificados) sobrepasa el valor de umbral 6, esto significa que existe falta de sincronía.

En este caso, el circuito envía una señal de suspensión a los contadores del circuito controlador de memoria ("RST") durante un ciclo de decodificación, con objeto de dejar pasar un bit de la rama recibida, y reinicializa la operación del decodificador al término de este lapso. El circuito de sincronización y el diagrama de tiempos se muestran en la figura 4.19.

7. Circuito de salida.

La última etapa del decodificador consiste en un circuito de salida de bits decodificados, el cual esencialmente es un almacena-

36

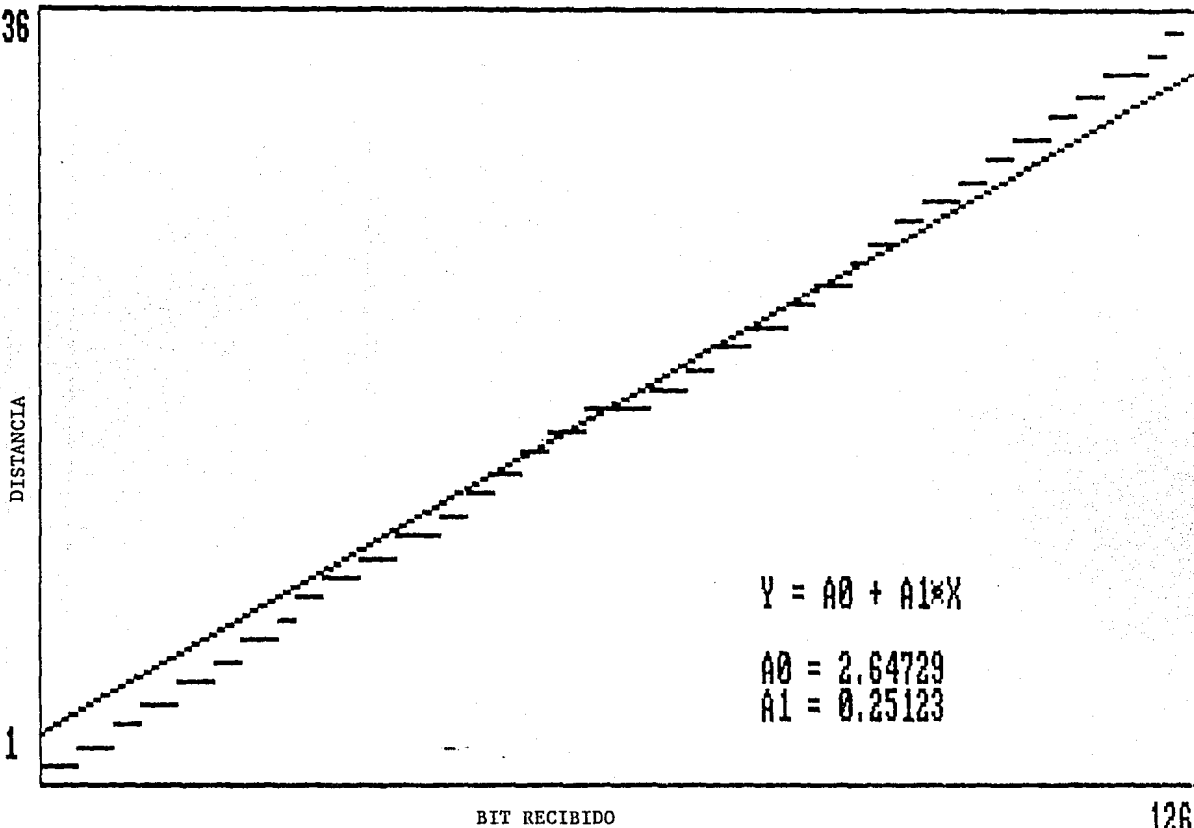


Figura 4.17. TASA DE CRECIMIENTO DE LA DISTANCIA CUANDO EL ALGORITMO DE VITERBI NO ESTA SINCRONIZADO, PARA EL CODIGO DE TASA 1/2, K=3 Y LONGITUD DE DECODIFICACION L=16 BITS.

ESTA
TESIS
NO PUEDE
SALIR
DE LA
BIBLIOTECA

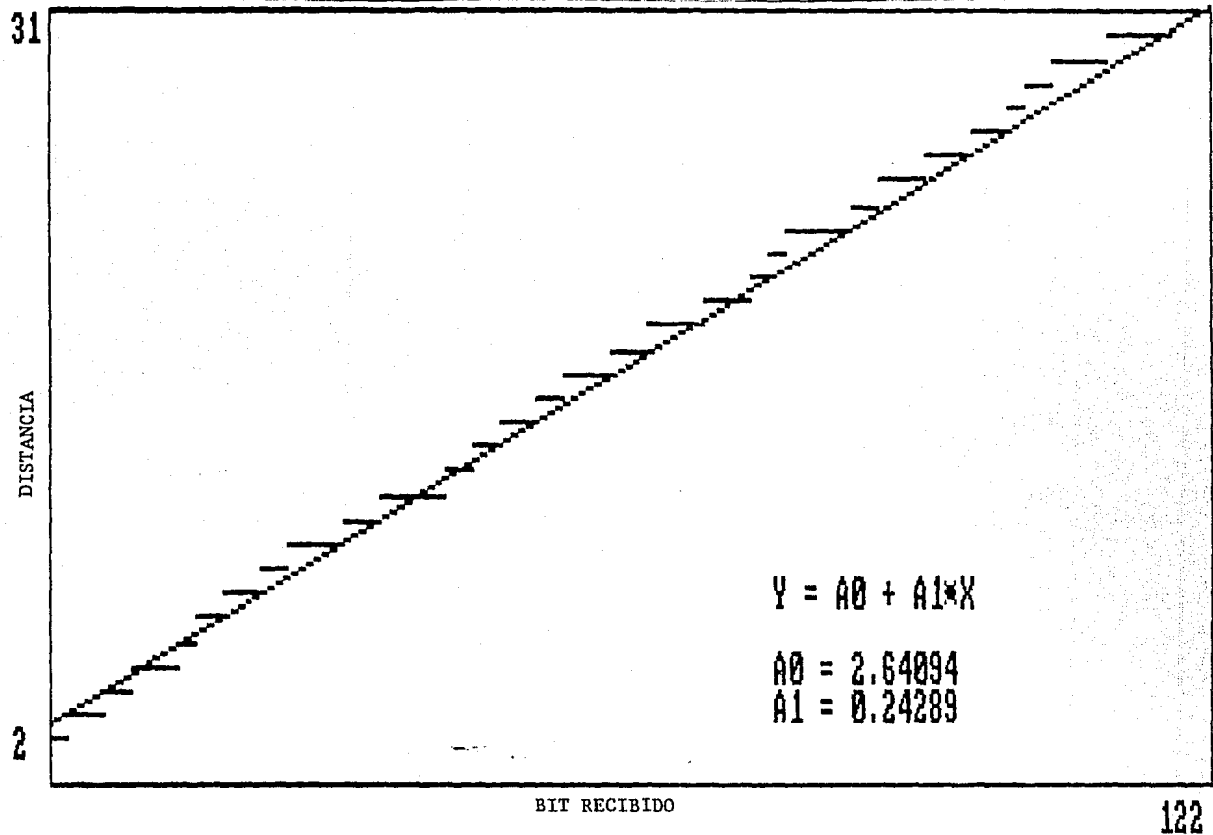


Figura 4.18. TASA DE CRECIMIENTO DE LA DISTANCIA CUANDO EL ALGORITMO DE VITERBI NO ESTA SINCRONIZADO, PARA EL CODIGO DE TASA 1/2, K=7 Y LONGITUD DE DECODIFICACION L=32 BITS.

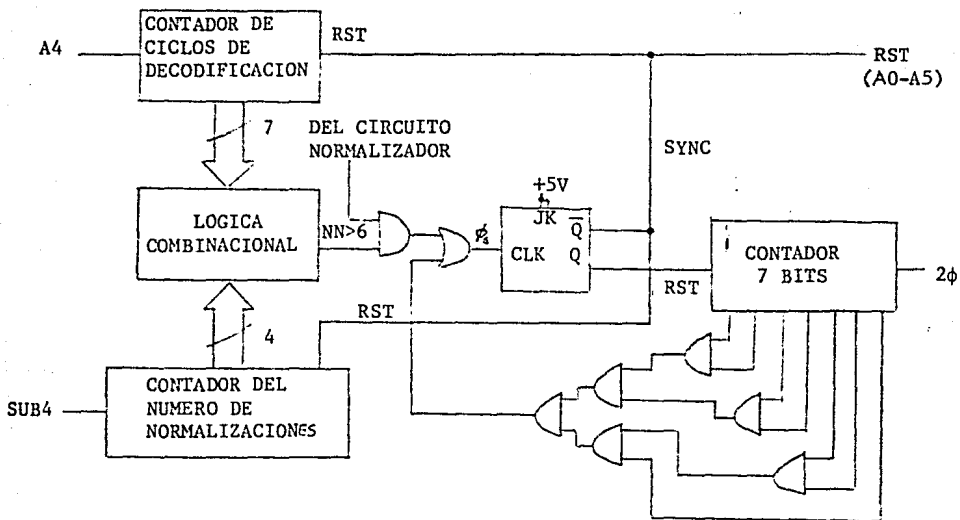
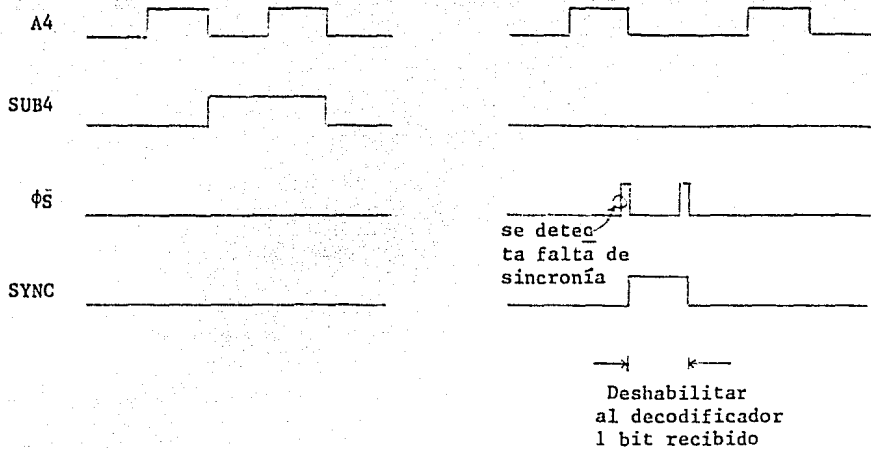


Figura 4.19(a) DIAGRAMA DE TIEMPOS Y CIRCUITO DE SINCRONIZACION.

LOGICA COMBINACIONAL:

NUMERO DE
NORMALIZACIONES

NN_3	NN_2	NN_1	NN_0	$NN>6$
0	0	0	0	0
		0	1	0
		1	0	0
		1	1	0
0	1	0	0	0
		0	1	0
		1	0	0
		1	1	1
1	0	0	0	1
		0	1	1
		1	0	1
		1	1	1
1	1	0	0	1
		0	1	1
		1	0	1
		1	1	1

$$NN>6 = NN_2 \cdot NN_1 \cdot NN_0 + NN_3$$

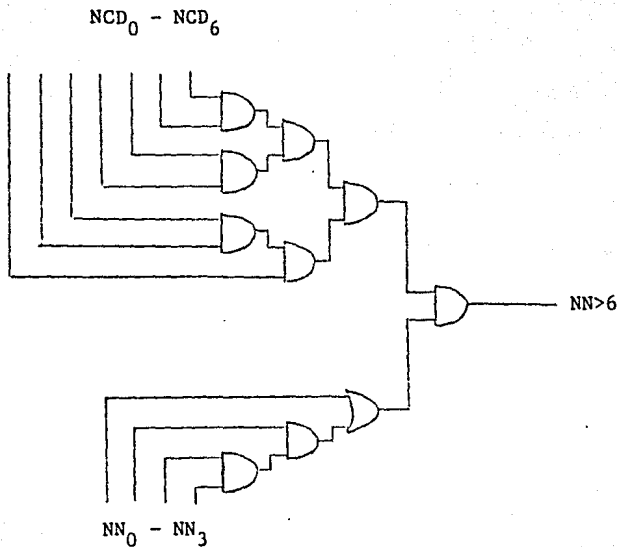


Figura 4.19 (b) LOGICA COMBINACIONAL EMPLEADA EN EL CIRCUITO DE SINCRONIZACION.

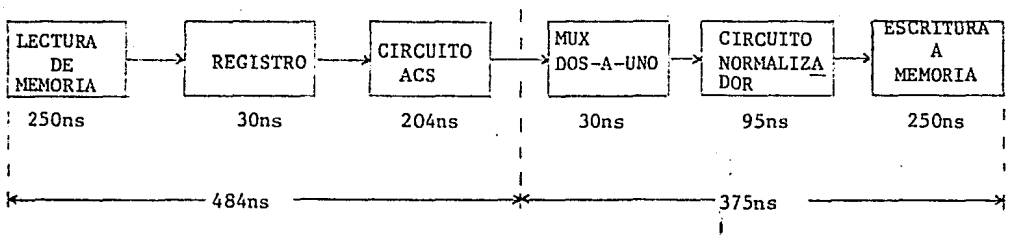
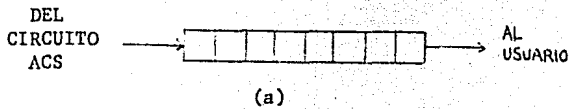
dor ("buffer") de 8 bits, con una señal de reloj de 16 KHz, y cuya estructura es presentada en la figura 4.20.

4.4. REQUERIMIENTOS DE FUNCIONAMIENTO.

En cuanto a la complejidad del decodificador, son necesarios 104 circuitos integrados de la familia TTL y 36 circuitos para la memoria RAM estática (Fig.4.21). La potencia necesaria de la fuente es de 17 Watts. Sin embargo, el uso de circuitos y memorias de tecnologías más avanzadas podrían permitir disminuir tanto la complejidad como los requerimientos de potencia del decodificador .

En cuanto al retraso de arranque del decodificador, se verificó, mediante simulaciones por computadora, que al cabo de un periodo corto de tiempo, 64 ciclos de decodificación, el decodificador opera normalmente sin importar el contenido de la memoria de trayectorias y métricas al iniciar su funcionamiento. Esto quiere decir que no es necesario determinar las q^K métricas de las trayectorias correspondientes al primer paso del algoritmo de Viterbi, reduciendo de manera importante la complejidad del decodificador.

El esquema funciona en tiempo real, es decir, los bits recibidos son procesados en un tiempo tal que el decodificador inicia un nuevo ciclo de decodificación al inicio del siguiente grupo de n bits. El retraso codificador-decodificador es de 44 bits, correspondientes al almacenador de 8 bits de canal (4 bits de información) a la entrada del decodificador, a la longitud de decodificación de 32 bits, y al almacenador de 8 bits de información a la salida.



.Frecuencia máxima de operación:

$$f_{\max} = \frac{1}{484\text{ns}} = 2.066 \text{ MHz}$$

.Tasa máxima de información:

$$R_{\max} = \frac{f_{\max}}{128 \text{ ciclos/bit}} = 16.141 \text{ Kbps}$$

(b)

Figura 4.20 (a) CIRCUITO DE SALIDA. EL CIRCUITO EMPLEA UN REGISTRO DE CORRIMIENTO DE 8 BITS. (b). ANALISIS DE LOS TIEMPOS DE PROPAGACION Y TASA MAXIMA DE INFORMACION DEL CIRCUITO.

1.	CIRCUITO DE ADQUISICION	2
2.	CIRCUITO ACS	36
3.	MEMORIA DE TRAYECTORIAS Y METRICAS	63
4.	CIRCUITO CONTROLADOR DE MEMORIA	18
5.	CIRCUITO NORMALIZADOR DE METRICAS	9
6.	CIRCUITO DE SINCRONIZACION	11
7.	CIRCUITO DE SALIDA	<u>1</u>

TOTAL 140

CLAVE	FUNCION	I _{cc} (Max) mA	No.	P _{TOT} mW	ζ max nS	PATILLAS
SN74LS04	HEX INVERTER	6.6	3	99.0	15.0	14
08	QUAD 2-INPUT AND GATE	8.8	17	748.0	20.0	14
32	QUAD 2-INPUT OR GATE	9.8	10	490.0	22.0	14
75	4-BIT D LATCH	12.0	19	1140.0	30.0	16
83A	4-BIT BINARY FULL ADDER	39.0	2	390.0	24.0	16
85	4-BIT MAGNITUDE COMPARATOR	20.0	1	100.0	30.0	16
86	QUAD 2-INPUT X-OR GATE	10.0	5	250.0	30.0	14
91	8-BIT-SHIFT REGISTER	20.0	2	200.0	40.0	14
93	4-BIT BINARY COUNTER	15.0	7	525.0	70.0	14
112A	DUAL JK NEGATIVE EDGE-TRIGERED FLIP-FLOP	6.0	2	60.0	20.0	16
125A	QUAD 3-STATE BUFFERS, LOW ENABLE	20.0	27	2700.0	25.0	14
157	QUAD 2-INPUT MULTIPLEXER NONINVERTING	16.0	9	720.0	27.0	16
2112A-2	256X4 BIT STATIC RAM	55.0	36	9900.0	250.0	16
	TOTAL		140	17322.0		

Figura 4.21 (a) CIRCUITOS INTEGRADOS EMPLEADOS EN EL DECODIFICADOR DE VITERBI Y (b) SUS CARACTERISTICAS.

Finalmente, debe hacerse notar que este diseño tiene el objetivo de proporcionar directrices para la implementación física de un decodificador de Viterbi. No deberá sorprender, por lo tanto, que en la etapa de construcción del prototipo sea necesaria la modificación de algunos circuitos para operar eficientemente.

CAPITULO 5. CONCLUSIONES Y TRABAJO A DESARROLLAR.

Lo realizado durante este trabajo ha permitido el diseño de un decodificador de Viterbi para trabajar a tasas de transmisión de 32000 bits/segundo, en el marco del diseño de una estación terrena para la transmisión de datos por satélite en la banda Ku. Se determinó que la decodificación de Viterbi, empleando un código de tasa 1/2 con una longitud de restricción de 7, constituye la alternativa más adecuada para esta aplicación.

Además del diseño en sí, se obtuvieron conclusiones interesantes acerca del desempeño de este tipo de decodificador, como lo es el no tener que iniciar la operación calculando las métricas de todas las q^k trayectorias sobrevivientes iniciales, requerido en teoría por el algoritmo de Viterbi, sino que el decodificador, por la estructura del código convolucional empleado, se recupera después de aproximadamente 8 a 9 longitudes de restricción.

También se determinó la tasa de crecimiento de la distancia mediante simulaciones por computadora, cuando no existe sincronía entre las ramas recibidas y las ramas consideradas por el decodificador, y cuyos resultados no habían sido publicados antes. Sería interesante realizar simulaciones adicionales para determinar la relación que existe entre la tasa de crecimiento de la distancia y los parámetros del código.

Al investigar las diversas alternativas de implementación, se logró la realización de un decodificador serie de Viterbi basado en un microprocesador de 16 bits. Con este esquema, se logró simular

esta técnica de decodificación, y constituye un resultado importante de este trabajo, aún cuando no se haya obtenido la tasa de transmisión especificada.

El diseño proporciona bases firmes para la construcción de un decodificador que pueda integrarse a una estación terrena que proporcione servicios de transmisión de datos por satélite en México. El trabajo a desarrollar consistirá entonces de la construcción de un prototipo del decodificador diseñado, con las modificaciones pertinentes, y de pruebas de funcionamiento en laboratorio antes de su transferencia a la industria.

BIBLIOGRAFIA

- [1] E.R. Berlekamp, "The Technology of Error-Correcting Codes," *Proc. IEEE*, Vol.68, May 1980, pp.564-593.
- [2] V.K. Bhargava, D. Haccoun, R. Matyas and P. Nuspi, *Digital Communications by Satellite*, Wiley: New York, 1981.
- [3] J.B. Cain, G.C. Clark and J.M. Gaist, "Punctured Convolutional Codes of Rate $(n-1)/n$ and Simplified Maximum Likelihood Decoding," *IEEE Trans. Info. Theory*, Vol.IT-25, No.1, Jan.1979, pp.97-100.
- [4] M. Cedervall, R. Johannesson and K.S. Zigangirov, "A New Upper Bound to the First-Event Error Probability for Maximum Likelihood Decoding of Fixed Binary Convolutional Codes," *IEEE Trans. Info. Theory*, Vol.IT-30, No.5, Sep.1984, pp.762-766.
- [5] P.R. Chevillat and D.J. Costello, Jr., "A Multiple Stack Algorithm for Erasurefree Decoding of Convolutional Codes," *IEEE Trans. Comm.*, Vol.COM-25, No.12, Dec.1977, pp.1460-1470.
- [6] G.C. Clark and J.B. Cain, *Error-Correction Coding for Digital Communications*, Plenum Press, 1981.
- [7] R.M. Fano, "A Heuristic Discussion of Probabilistic Decoding," *IEEE Trans. Info. Theory*, Vol.IT-9, Apr.1963, pp.64-74.
- [8] G.D. Forney, "Convolutional Codes I: Algebraic Structure," *IEEE Trans. Info. Theory*, Vol.IT-16, No.6, Nov.1970, pp.720-738.
- [9] R.L. Freeman, *Telecommunication Transmission Handbook*, Wiley: New York, 1975.

[10] F.J. García Ugalde and R.H. Morelos-Zaragoza, "Design of a Viterbi Decoder with Microprocessor-Based Serial Implementation," AAEQQ-4, Karlsruhe, Federal Republic of Germany, Sep.23-26 1986.

[11] D. Haccoun and M.J. Ferguson, "Generalized Stack Algorithms for Decoding Convolutional Codes," IEEE Trans. Info. Theory, Vol.IT-21, No.6, Nov.1975, pp.638-656.

[12] J.A. Heller and I.M. Jacobs, "Viterbi Decoding for Satellite and Space Communications," IEEE Trans. Comm. Tech., Vol.COM-19, No.5, Oct.1971, pp.835-848.

[13] I.M. Jacobs and E.R. Berlekamp, "A Lower Bound to the Distribution of Computations for Sequential Decoding," IEEE Trans. Info. Theory, Vol.IT-13, No.2, Apr.1967, pp.167-174.

[14] F. Jelinek, "Fast Sequential Decoding Algorithm using a Stack," IBM Journal on R & D, Nov.1969, pp.675-695.

[15] M.J. Jeruchim, "On the Coding Gain for Degraded Channels," IEEE Trans. Comm., Vol.COM-34, No.5, May 1986, pp.492-496.

[16] K.J. Larsen, "Short Convolutional Codes with Maximal Free Distance for Rates 1/2, 1/3 and 1/4," IEEE Trans. Info. Theory, Vol.IT-19, No.3, May 1973, pp.371-372.

[17] D.L. Lyon, "Personal Computer Communications Via Ku-Band Small-Earth Stations," IEEE Journal on Selected Areas of Communications, Vol.SAC-3, May 1985, pp.440-448.

[18] H.H. Ma, "The Multiple Stack Algorithm Implemented on a

Zilog Z-80 Microcomputer," IEEE Trans.Compl., Vol.COM-28, No.11, Nov.1980, pp.1876-1882.

[19] E. Paaske, "Short Binary Convolutional Codes with Maximal Free Distance for Rates 2/3 and 3/4," IEEE Trans.Info.Theory, Vol.IT-20, No.5, Sept.1974, pp.683-688.

[20] C. Shannon, "A Mathematical Theory of Communication," Bell System Technical Journal, Vol.27, 1948, pp.379-423,623-656.

[21] A.J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Trans.Info.Theory, Vol.IT-13, No.3, Apr.1967.

[22] A.J. Viterbi, "Convolutional Codes and Their Performance in Communications Systems," IEEE Trans.Compl., Vol. COM-19, No.5, Oct.1971, pp.751-772.

[23] A.J. Viterbi and J.K. Omura, Principles of Digital Communications, McGraw-Hill: New York, 1979.

[24] J.M. Wozencraft and I.M. Jacobs, Principles of Communication Engineering, Wiley: New York, 1965.

[25] W. Wu, Elements of Digital Satellite Communications, Vol II, Computer Science Press, 1985.

[26] K.S. Zigangirov, "On the Error Probability of Sequential Decoding on the BSC," IEEE Trans.Info.Theory, Vol.IT-18, No.1, Jan.1972, pp.199-202.