

Universidad Nacional de Mar del Plata

Facultad de Ingeniería

CDMA

Simulación del Canal Directo
en un sistema de Telefonía Celular

Autor: Marcelo Daniel Polvorín

Director: Jorge Castiñeira Moreira

Año 2000

Índice

Capítulo 1: Resumen	6
Capítulo 2: Introducción	7
Capítulo 3: Anteproyecto	10
Capítulo 4: Introducción a CDMA	13
4.1: ¿Qué es CDMA?	13
4.1.1: Acceso múltiple en comunicaciones inalámbricas	13
4.1.2: La revolución de CDMA	14
4.1.3: Aspectos de fondo	14
4.1.4: Reuso de frecuencia en los sistemas tradicionales	15
Sectorización de las antenas	17
4.1.5: Reuso en CDMA	17
4.1.6: Otros Standards celulares actuales	18
4.1.7: El Standard celular CDMA	19
4.1.8: Comparación entre técnicas de acceso múltiple	19
FDMA	19
TDMA	20
CDMA	20
4.1.9: La tecnología de CDMA	21
Espectro esparcido	21
Sincronización	22
El equilibrio de las características	22
4.2: Beneficios de CDMA	22
4.2.1: Aumento de capacidad	23
CDMA y reuso en las células	23
Eb/No y umbral de interferencia	24
4.2.2: Mejor calidad de la llamada	24
Avanzada detección y corrección de errores	25
Vocoders sofisticados	25
Múltiples formas de diversidad	26
Diversidad de frecuencia	26
Diversidad espacial	27
Diversidad de caminos	27
Diversidad de tiempo	28
Soft handoff	28
Control de potencia preciso	30
4.2.3: Planificación del sistema simplificada	30
4.2.4: Mayor privacidad	31
4.2.5: Cobertura mejorada	31
4.2.6: Aumento del tiempo de conversación portátil	31
4.2.7: Ancho de banda según la demanda	31

4.3: Capacidad en CDMA	32
4.3.1: El problema cerca-lejos	33
4.3.2: Control de potencia	33
4.3.3: Capacidad de la célula	35
4.3.4: La ventaja de usar CDMA	35
4.3.5: Codificación de voz	36
4.3.6: Propagación en múltiples caminos (Multipath)	37
¿Cuándo causa fading el multipath?	37
4.3.7: Problemas de Cobertura	38
4.3.8: Compromiso cobertura-capacidad en el vínculo inverso	38
4.3.9: Capacidad real	41
4.4: Ancho de banda elegido	41
4.5: Control de potencia	44
4.5.1: El ambiente de propagación en telefonía móvil	44
4.5.2: Control de potencia en el vínculo inverso	45
Control de lazo abierto	45
Control de lazo cerrado	46
Soft handoff	46
4.5.3: Control de potencia en el vínculo directo	46
4.6: Handoff	47
4.6.1: Frecuencia del handoff	48
4.6.2: Pasos en un handoff	48
4.6.3: Proceso de handoff tradicional	49
4.6.4: Proceso del handoff en CDMA	51
4.6.5: Detalles del handoff	51
4.6.6: Soft handoff	52
4.6.7: Canal directo	52
4.6.8: Control de potencia en el canal inverso	52
4.6.9: Canal inverso	52
4.6.10: Softer handoff	53
4.6.11: ¿Cómo se inicia un soft handoff?	53
Búsqueda del piloto	53
Umbrales de detección	53
4.6.12: ¿Usa CDMA siempre el soft handoff?	54
4.6.13: Handoffs a distintas frecuencias	54
4.6.14: Digital a analógico	54
Capítulo 5: Simulación del Canal Directo	55
5.1: Transmisión	55
5.1.1: Generación de los frames	56
5.1.2: Codificación CRC	57
5.1.3: Codificación convolucional	60
5.1.4: Repetición	64
5.1.5: Interleaving	64
5.1.6: Scrambling	65

5.1.7: Esparcido	69
5.1.8: Modulaci3n en cuadratura	70
5.1.9: Modulaci3n	74
5.1.10: Canal Piloto	78
5.1.11: Canal de Sincronismo	79
5.1.12: Canales de Paging	80
5.2: Características del canal de transmisi3n	81
5.3: Recepci3n	83
Capítulo 6: Funcionamiento del canal inverso	95
6.1: Generaci3n de los frames	96
6.2: Codificaci3n CRC	96
6.3: Codificaci3n Convolutiva	96
6.4: Interleaving	97
6.5: Modulaci3n Ortogonal	98
6.6: Data Burst Randomizer	98
6.7: Separaci3n de usuarios	98
6.8: Modulaci3n de RF	100
6.9: Generaci3n del Canal de Acceso	100
6.10: Recepci3n	101
Capítulo 7: Manual de operaci3n	102
7.1: Como realizar la simulaci3n	102
7.2: Como ver los resultados	104
7.2.1: Tabla "CRC"	105
7.2.2: Tabla "Convolutiva"	106
7.2.3: Tabla "Walsh"	107
7.2.4: Tabla "RF"	108
7.2.5: Tabla "Recibido"	109
7.2.6: Tabla "Recuperado"	109
7.2.7: Tabla "Decodificado"	110
Capítulo 8: Conclusiones	112
Capítulo 9: Bibliografía	120
<i>Apéndice I: Asignaci3n de frecuencias</i>	<i>121</i>
<i>Apéndice II: C3digos Convolutivos</i>	<i>122</i>
<i>Apéndice III: C3digos C3clicos</i>	<i>131</i>

Índice de figuras

Capítulo 4: Introducción a CDMA

4.1: Células hexagonales adyacentes	16
4.2: Distribución de las células cubriendo el área de servicio	16
4.3: Reuso de frecuencias en CDMA	18
4.4: FDMA	19
4.5: TDMA	20
4.6: DS-CDMA	21
4.7: Beneficios en Calidad por Diversidad de Frecuencia en CDMA	26
4.8: Beneficios en Calidad por Diversidad de Caminos	28
4.9: CDMA Soft Handoff mejora la Calidad de los Frames	29
4.10: CDMA Soft Handoff utiliza dos o más células	30
4.11: Multipath y retardos en el canal	42
4.12: Respuesta al impulso del canal	42
4.13: Efecto del ancho de banda	43
4.14: Límite de las coberturas de cada celda	47

Capítulo 5: Simulación de la comunicación en el Canal Directo

5.1: Generación de un Canal Directo	55
5.2: Composición de los Frames	56
5.3: Codificador convolucional usado en el canal directo	61
5.4: Bits transmitidos para un vocoder de tipo 2	61
5.5: Ganancia de codificación	62
5.6: Enmascaramiento del Código Largo	66
5.7: Long Code Mask (Máscara del Código Largo)	66
5.8: Chips utilizados para el Scrambling	67
5.9: Canales Directos en CDMA	69
5.10: Generación de las componentes I y Q	71
5.11: Autocorrelación de las secuencias	71
5.12: Modulador QPSK	75
5.13: Cambios de fase en la portadora	77
5.14: Características del filtro pasabajos especificado	77
5.15: Generación del Canal Piloto	78
5.16: Demodulación del Canal Piloto	79
5.17: Generación del Canal de Sincronismo	79
5.18: Generación de un Canal de Paging	80
5.19: Máscara del Código Largo de un Canal de Paging	80
5.20: Modelo del Canal de Transmisión	81
5.21: Diagrama en bloques del Receptor	84
5.22: Demodulador QPSK	84

Capítulo 6: Funcionamiento del Canal Inverso

6.1: Generación de un Canal Inverso	95
-------------------------------------	----

6.2: Frames generados por el Vocoder	96
6.3: Codificador convolucional de velocidad ½	97
6.4: Codificador convolucional de velocidad 1/3	97
6.5: Máscara del Código Largo para un Canal de Tráfico Inverso	99
6.6: Modulación del Canal Inverso	99
6.7: Modulación Offset QPSK	100
6.8: Máscara del Canal de Acceso	100
6.9: Diagrama en bloques del Receptor	101
Capítulo 7: Manual de operación	
7.1: Formulario Mensajes	102
7.2: Formulario características del canal. Ficha del Multipath	103
7.3: Formulario características del canal. Ficha del Ruido	104
7.4: Tablas de la Base de Datos, vistas con Access	105
7.5: Tabla CRC	105
7.6: Tabla Convolucional	106
7.7: Tabla Walsh	107
7.8: Tabla RF	108
7.9: Tabla Recibido	109
7.10: Tabla Recuperado	110
7.11: Tabla Decodificado	110
7.12: Tabla Decodificado. Frame inválido	111
Capítulo 8: Conclusiones	
8.1: Canal 0	114
8.2: Canal 1	114
8.3: Canal 2	115
8.4: Suma de los tres canales modulados	115
8.5: Efecto del Multipath en el canal de transmisión	116
8.6: Señal mas ruido en el receptor	116

Capítulo I: Resumen

El presente programa realiza la simulación del funcionamiento del canal directo¹ para un sistema de telefonía celular CDMA (Code División Múltiple Access). El resultado es una base de datos que contiene diferentes tablas, en donde puede analizarse paso a paso todo el proceso de transmisión y recepción que fue simulado.

El programa permite simular la transmisión de tres señales diferentes a la vez, para poder observar como se realiza la separación de usuarios mediante el uso de códigos ortogonales entre sí, ya que todos comparten la misma portadora la totalidad del tiempo. Los mensajes a transmitirse, así como el canal que ocupará cada uno, son elegidos por el usuario antes de simular la transmisión. Los mensajes que se ingresan son usados para conformar los tres frames (de 20 mseg. de duración) que se simulará transmitir.

En la base de datos que se genera puede analizarse como afectan a la señal original los procesos de codificación, Interleaving, encriptación y separación de usuarios, así como también la forma en que se diferencian entre sí las distintas celdas y la modulación QPSK usada para la transmisión. Al final del proceso, las señales tienen un ancho de banda de 1.2288 MHz, pudiendo analizarse sus formas de onda por separado y la suma de las tres.

El programa permite también simular las características del canal de transmisión. Para esto, el usuario puede seleccionar los retardos y atenuaciones en el ambiente (multipath), y la intensidad del ruido. Además, puede elegirse un intervalo de tiempo durante el cual se destruye la señal transmitida, para poder observar el funcionamiento de la corrección de errores en el receptor. En este punto, puede verse el resultado de agregar el ruido y el efecto del multipath a las tres señales transmitidas, que es lo que llega al receptor.

En el receptor puede seleccionarse la señal que se desee recibir, pudiendo observarse cómo se separa la señal deseada de los demás usuarios y el ruido. Comparando la señal transmitida con la recibida, puede analizarse el funcionamiento de la codificación convolucional aplicada, y comprobar la cantidad de errores que puede llegar a corregir el sistema.

Como esta tecnología es muy nueva y radicalmente distinta a las usadas anteriormente, se incluye en el informe un capítulo teórico donde se detalla el funcionamiento del sistema, sus principales características y los beneficios que se obtienen con respecto a las demás tecnologías existentes.

¹ El espectro para telefonía móvil inalámbrica normalmente se asigna en división de frecuencia doble (FDD), con dos bandas separadas para los dos sentidos de transmisión. En los sistemas celulares están separados por 45 MHz, y en las bandas de PCS por 80 MHz. Se llama canal directo al sentido de transmisión de la celda al teléfono móvil.

Capítulo II: Introducción

Code Division Multiple Access (CDMA) es un concepto sumamente nuevo en comunicaciones inalámbricas. Ha ganado una aceptación internacional cada vez más extendida entre los operadores de sistemas celulares, por ser una tecnología que aumenta dramáticamente la capacidad del sistema y la calidad de servicio. Ha sido igualmente escogida para utilizarse por la mayoría de los operadores en los Estados Unidos y desde 1999 se ha introducido en Argentina. Puede parecer, sin embargo, algo misterioso para aquéllos que no están familiarizado con ella.

CDMA es una forma de espectro esparcido, una familia de técnicas de comunicación digitales que se han usado en aplicaciones militares durante muchos años. La esencia del principio del espectro esparcido es el uso de señales pseudo-aleatorias como portadora, y, tal como el nombre implica, un ancho de banda mucho más grande que aquel requerido para una simple comunicación punto a punto a la misma velocidad. Existen dos motivos para su uso en sistemas militares: resistir esfuerzos enemigos para bloquear las comunicaciones (anti-jam, o AJ), y esconder el hecho de que la comunicación tiene lugar, algunas veces llamada Low Probability of Intercept (LPI). Tiene una historia que se remonta a los primeros días de la Segunda Guerra Mundial.

El uso de CDMA para aplicaciones de radio móviles civiles es nuevo. Se propuso teóricamente a finales de 1940's, pero la aplicación práctica en el mercado civil no tuvo lugar hasta 40 años después, instalándose los primeros sistemas durante 1995. Las aplicaciones comerciales se hicieron posibles debido a dos desarrollos evolutivos. Uno fue la disponibilidad, a un costo muy bajo, de circuitos integrados digitales de alta densidad que reducen el tamaño, peso, y costo de los teléfonos móviles a un nivel aceptablemente bajo. El otro fue la comprensión de que una comunicación de acceso múltiple óptima requiere que todas las estaciones (tanto las celdas como los móviles) ajusten las potencias de transmisión al nivel más bajo con el que se pueda lograr una calidad de señal adecuada, ya que lo que es señal para un teléfono es ruido para los demás.

Un sistema de radio móvil de acceso múltiple idealizado, consiste en una familia de estaciones de base, o "celdas", geográficamente distribuidas en el área de servicio. Cada celda abastece estaciones subscriptoras (móviles) dentro de una área geográfica limitada (célula). Cuando un subscriptor se mueve entre las células, el mensaje en el aire se usa para controlar el traspaso de la célula vieja a la nueva célula. Este traspaso es llamado handoff o handover.

Tradicionalmente los sistemas de comunicación de radio han separado a los usuarios por canales de frecuencia, slots de tiempo, o ambos. Los sistemas celulares modernos empezaron con el uso de canalización de FM analógica. Más recientemente se han desarrollado varios sistemas digitales FDM-TDM híbridos, que ostensiblemente reforzaron la calidad del servicio y la capacidad. En todos estos sistemas, a cada usuario se le asigna un recurso de tiempo y frecuencia particular.

El concepto de reuso de frecuencia es fundamental para el concepto de telefonía celular. Aunque hay centenares de canales disponibles, si cada frecuencia se asignara a sólo una célula, la capacidad del sistema total igualaría al número total de canales: sólo unos pocos miles de subscriptores por sistema. Reusando canales en células distintas, el sistema puede crecer sin límites geográficos.

La misma frecuencia no puede reusarse obviamente en cualquier par adyacente de células, esto lleva a que un canal utilizado por una célula no se pueda usar en ninguna de las 6 células adyacente, distribuyendo así 1/7 del espectro disponible en cada célula y repitiendo este arreglo en toda la superficie a cubrir.

CDMA ofrece una solución totalmente diferente. En lugar de dividir ya sea el espectro o el tiempo en distintas "hendeduras" a cada usuario se le asigna una diferente portadora pseudo-aleatoria, compartiendo el mismo espectro con los demás usuarios.

El mayor beneficio de las portadoras pseudo-aleatorias es que la sensibilidad del sistema a la interferencia se altera fundamentalmente, mejorando el desempeño y aumentando así la capacidad.

El reuso de frecuencia es universal, es decir, todas las células pueden utilizar cada frecuencia en CDMA.

Una llamada de CDMA comienza con una velocidad normal de 9600 bits por segundo (9.6 kilobits por segundo). Esto es después esparcido a una frecuencia transmitida de aproximadamente 1.23 Megabits por segundo. Este "esparcido" se hace aplicando códigos digitales a los bits del mensaje original. Estos bits se transmiten al mismo tiempo con las señales de todos los otros usuarios en esa célula. Cuando la señal se recibe, los códigos se remueven de la señal deseada, separando a los usuarios y devolviendo la llamada a su velocidad de 9600 bps.

En las fases finales de la codificación del vínculo de radio desde la estación base al móvil, CDMA agrega un "código pseudo-aleatorio" especial a la señal, que se repite después de una cantidad finita de tiempo. Las estaciones base en el sistema se distinguen entre sí transmitiendo versiones desplazadas en tiempo del mismo código pseudo-aleatorio. Para asegurar que los desplazamientos de tiempo usados permanezcan únicos entre sí, las estaciones de CDMA deben permanecer sincronizadas a una referencia de tiempo común, que es proporcionada por el Sistema de Posicionamiento Global (GPS).

CDMA cambia la naturaleza del teléfono móvil de un aparato predominantemente analógico a un aparato predominantemente digital. Los receptores de radio anticuados separan estaciones o canales filtrando en el dominio de frecuencia. Los receptores de CDMA no eliminan el proceso analógico completamente, sino que separan los canales de comunicación por medio de una modulación pseudo-aleatoria que se aplica y se quita en el dominio digital, no en frecuencia. Los distintos usuarios ocupan la misma banda de frecuencia. Este reuso universal de frecuencias no es fortuito. Al contrario, es crucial para la alta eficiencia espectral que es el sello de CDMA.

CDMA le está cambiando la cara a la comunicación celular y PCS porque:

1. Mejora considerablemente la capacidad del tráfico telefónico.
2. Mejora considerablemente la calidad de la voz y elimina los efectos audibles de desvanecimiento por multipath.
3. Reduce la incidencia de llamadas caídas debido a los fracasos del handoff
4. Proporciona un mecanismo de transporte fiable para comunicaciones de datos, como facsímil y tráfico de Internet.

5. Reduce el número de sitios necesario para dar servicio a una cantidad dada de tráfico.
6. Simplifica la selección de los sitios para ubicar las celdas.
7. Se reducen los costos de despliegue y operación porque se requieren menos celdas.
8. Se reduce la energía promedio transmitida.
9. Se reduce la interferencia a otros dispositivos electrónicos
10. Se reducen potenciales riesgos de salud.

En el presente proyecto se simula la transmisión en el canal directo (de la celda al teléfono móvil), pudiendo observarse paso a paso los resultados de aplicar a la señal original todos los procesos de codificación y modulación que se realizan en el transmisor, los efectos del canal sobre la señal transmitida y la demodulación y decodificación que se realiza en el receptor.

Para poder ver como se realiza la separación de canales, en el transmisor se simula transmitir tres mensajes distintos (como si fueran 3 conversaciones de diferentes teléfonos celulares), y luego puede elegirse cual de ellos decodificar en el receptor.

Tanto los mensajes a transmitirse como las características del canal de transmisión son elegidas por el usuario al realizar la simulación.

En el capítulo IV del presente informe, se explican teóricamente las principales características del sistema, las diferencias respecto a los demás standards existentes, los beneficios que se obtienen con CDMA y se realiza un cálculo del aumento de capacidad que se logra.

El capítulo V trata detalladamente el funcionamiento del canal directo, con una explicación teórica de cada proceso realizado a la señal en el transmisor, el modelo del canal de transmisión que se simula y el proceso de recepción y decodificación llevado a cabo por el receptor para recuperar la señal original. También se incluye en este capítulo el código mediante el cual se realiza la simulación de cada bloque.

En el capítulo VI se explica el funcionamiento del canal inverso, detallando sus principales diferencias con respecto a la transmisión en el otro sentido.

En el capítulo VII se explica el funcionamiento del programa, indicándose como se realiza la simulación y como pueden verse los resultados.

El capítulo VIII muestra las conclusiones que pueden sacarse de la simulación, basándose en varios ejemplos que se encuentran en el CD del proyecto.

Capítulo III: Anteproyecto

Para realizar el programa se eligió como herramienta de programación Visual Basic 6.0, debido a la facilidad con que puede realizarse una interfaz gráfica agradable y fácil de usar, con controles y menús habituales en Windows, con los que el usuario está ampliamente familiarizado.

Además, resultan muy útiles algunas herramientas que incluye, como el Help Workshop y el Asistente para Empaquetado y Distribución.

El Help Workshop es un compilador que permite crear un archivo de ayuda que podrá ser incorporado al proyecto, interactuando con el mismo y comportándose como cualquier archivo de ayuda de Windows. Para esto se debe escribir un archivo de texto, en formato RTF, que será luego compilado por el Help Workshop.

El Asistente para Empaquetado y Distribución permite diseñar una versión instalable del programa, ya sea para CD o en disquetes, que incluye todos los recursos necesarios para la ejecución del mismo.

El programa genera, como resultado de cada simulación, una base de datos que contiene todo el proceso. Se eligió esto debido a que resulta ser la mejor forma de almacenar la gran cantidad de datos que resultan de la simulación², y también una buena forma para analizar los resultados.

Si no se trabajara con bases de datos, deberían usarse matrices para contener los resultados, lo cual implica utilizar una gran cantidad de memoria, y de todos modos siempre se necesitaría guardar todos los datos en un archivo.

Al generar cada simulación una base de datos, no es necesario mostrar todos los resultados en el programa, ya que el objetivo es precisamente poder analizarlo desde aplicaciones específicas para ello, como Access o dBase.

La principal desventaja de las bases de datos es la lentitud de la simulación, por estar continuamente accediendo al disco rígido.

Visual Basic presenta una forma simple de crear y modificar las bases de datos, pudiendo definir las como variables y trabajar con ellas mediante funciones específicas y sentencias SQL.

Se eligió trabajar con bases de datos Microsoft Jet, ya que es el motor compartido por Visual Basic y Access. De esta manera se generan bases de datos que pueden ser leídas por uno de los gestores de bases de datos más ampliamente difundidos³ y se procura obtener la máxima velocidad posible durante la simulación, ya que Jet es el motor nativo de Visual Basic.

Los datos en formato binario se trataron como variables booleanas, lo que facilita las decisiones y las operaciones entre bits (como una or exclusiva). En la base de datos, estos datos se verán como -1 y 0, ya que es la forma en que Visual Basic trata al true y false respectivamente.

Para simular la transmisión se eligió enviar 3 mensajes diferentes para poder apreciar como se realiza la codificación de los canales, y cómo el receptor puede separar el que está recibiendo de los demás. El usuario puede elegir los canales (códigos de Walsh) que se utilizarán para cada mensaje.

² Cada base de datos generada por la simulación contiene 32 MB de información.

³ Igualmente pueden verse desde otros programas, realizando una conversión.

Como también es interesante ver que sucede con la señal de una celda vecina transmitiendo en el mismo canal (en este caso lo que diferencia a las señales es el código corto que identifica a cada celda), se permite simular esta posibilidad seleccionando el mismo código de Walsh para los dos últimos mensajes.

En realidad lo que se estará simulando es la interferencia de otro sector de la misma celda, ya que si fueran celdas distintas el canal de transmisión sería completamente diferente para cada celda, dificultando la simulación del mismo.

Por cada canal se transmitirá un frame (20 mseg), ya que es suficiente para poder apreciar todo el proceso. Simular una mayor cantidad de frames incrementaría tanto el tamaño de la base de datos como el tiempo requerido para realizar la simulación, sin agregar un gran beneficio que lo justifique.

Cada frame es introducido por el usuario, que puede elegir tanto la velocidad (full, $\frac{1}{2}$, $\frac{1}{4}$, o $\frac{1}{8}$) como el mensaje en sí. Para facilitar la entrada de estos datos, se ingresan en formato hexadecimal. El programa lo convierte a binario para comenzar la simulación y lo vuelve a transformar en hexadecimal cuando muestra el resultado de la recepción.

También el usuario puede ingresar, en formato binario, los 32 bits que corresponden al ESN⁴ de cada teléfono móvil, que se usarán para enmascarar el código largo.

Para simular el canal se utiliza un modelo de multipath, con tres componentes que presentan distintas atenuaciones y retardos, al que se suma ruido aleatorio y también se simula el efecto de una interferencia.

Se simulan 3 componentes porque son las que pueden ser resueltas por el receptor Rake, las demás componentes que no son resueltas por el receptor se comportan como la interferencia de los demás usuarios, y como son de menor magnitud aportan poco al ruido de fondo.

El usuario puede elegir los retardos y atenuaciones del canal para las dos últimas componentes en arribar, respecto de la primera. Es decir, se elige para ambos casos el retardo (en μseg) respecto de la primer componente y la magnitud (0.8; 0.5 o 0.3) comparada con la primer componente.

Para simular el ruido, se elige también la magnitud del mismo con respecto a la señal transmitida. El ruido se genera mediante la función random de Visual Basic.

Para poder apreciar el funcionamiento del proceso de Interleaving y el decodificador de Viterbi, es necesario introducir errores en la transmisión. Esto puede ocurrir ya sea por una interferencia muy fuerte o, más comúnmente, por un desvanecimiento temporario de la señal, como por ejemplo al pasar bajo un puente.

Para simular esto se decidió destruir la señal por un lapso de tiempo determinado. Este lapso de tiempo es elegido por el usuario, como así también el instante en que comienza, ya que esto también afecta el funcionamiento del decodificador.

Algunos datos necesarios para el funcionamiento del programa se encuentran en la base de datos "Auxiliar". La tabla "Walsh" contiene los 64 códigos de Walsh, que serán utilizados tanto en la transmisión como para separar los canales en el receptor. La tabla "Interleaver" contiene el orden en que se deben intercalar y volver a reordenar los bits en el transmisor y el receptor respectivamente. La tabla "Convolutiva" es utilizada para instrumentar el decodificador de Viterbi, ya que contiene, para cada uno de los 256 estados posibles, el próximo estado y las salidas según el valor del siguiente bit. Toda

⁴ Electronic Serial Number.

esta información puede verse abriendo la base de datos con Access, teniendo cuidado de no modificar nada para no alterar el normal funcionamiento del programa.

Todos los datos introducidos por el usuario son almacenados en la base de datos "Setup", que también puede observarse utilizando Access. Las tablas "Mensaje" y "Canal" contienen los mensajes a transmitirse y las características del canal que fueran elegidos por el usuario, y tanto en "LongCode" como en "ShortCode" aparecen las columnas "Conexiones" y "Reg"⁵, que contienen las conexiones de los registros que generan los códigos pseudo-aleatorios y el estado inicial de los registros. Lógicamente puede alterarse el estado inicial, pero no las conexiones.

Como esta base de datos contiene los valores con que se realizó la simulación, al guardar los resultados se deben guardar también estos datos. Esto se realiza creando un archivo con el mismo nombre que se elija para la simulación y extensión STP, donde se copia el contenido de "Setup".

Así, al abrir la base de datos con el programa se carga también este archivo (copiándolo sobre "Setup"), y se puede seguir trabajando con una simulación interrumpida o cambiar algún parámetro y volver a simular.

Es muy útil el hecho de poder realizar la simulación por partes, tanto por una cuestión de tiempo como por la posibilidad de realizar cambios en el canal y simular a partir de ahí, sin tener que empezar desde el transmisor.

Para realizar esto, se trabaja siempre sobre una base de datos temporal, que se crea al iniciar la simulación o al abrirse una ya existente, y que podrá guardarse en cualquier momento con el nombre que se elija (esto permite sobrescribir una simulación anterior o mantenerla y crear una nueva con los cambios realizados). Este archivo temporal se elimina al salir del programa.

⁵ En el caso de ShortCode hay dos juegos de columnas, correspondiendo a las componentes en fase y en cuadratura, y en LongCode existen también las columnas que contienen los ESN elegidos.

Capítulo IV: Introducción a CDMA

4.1: ¿Que es CDMA (Code Division Multiple Access)?

4.1.1: Acceso múltiple en Comunicaciones Inalámbricas

Uno de los conceptos más importantes en cualquier sistema de telefonía celular es el de "acceso múltiple", lo cual significa que pueden soportarse usuarios simultáneos. En otras palabras, un número grande de usuarios comparte un sistema común de canales de radio y cualquier usuario puede ganar acceso a cualquier canal (el usuario no siempre se asigna al mismo canal). Un canal puede pensarse como sólo una porción del recurso de radio, que se asigna temporalmente para una llamada telefónica. Un método de acceso múltiple es una definición de cómo el espectro de radio es dividido en canales y cómo estos se asignan a los muchos usuarios del sistema.

Las metas de los sistemas de comunicaciones de acceso múltiple, como celular y PCS, son:

- ◆ Un servicio con calidad de voz cercana a la de los sistemas fijos.
- ◆ Cobertura geográfica casi universal
- ◆ Bajo precio de los equipos, tanto los móviles de los suscriptores como las estaciones fijas.
- ◆ Número mínimo de sitios de radio fijos

Las agencias reguladoras han asignado un ancho de banda limitado a estos servicios, así que las soluciones deben lograr una alta eficiencia espectral. Los operadores celulares tienen 25 MHz de ancho de banda cada uno, dividido entre las dos direcciones de comunicaciones. El servicio de PCS en los Estados Unidos tiene tres asignaciones de 30 MHz y tres 10 de MHz, también divididas.

Las aplicaciones prácticas de sistemas celulares que tienen centenares de canales se hicieron posibles con la disponibilidad de sintetizadores de frecuencia compactos y económicos. El control por microprocesador permite que complejos diálogos entre el transmisor y el receptor se lleven a cabo mediante sofisticados protocolos.

La solución celular, originalmente diseñada por los laboratorios Bell en los años setenta, hace uso de múltiples estaciones fijas, o células (el término célula se refiere al área de servicio, y celda al equipo). Cada celda abastece estaciones suscriptoras (móviles) dentro de una área geográfica limitada (Célula). Cuando un suscriptor se mueve entre las células, el mensaje en el aire se usa para controlar el traspaso de la célula vieja a la nueva célula. Este traspaso es llamado handoff o handover.

El sistema original se llamó Advanced Mobile Phone System, o AMPS. Es el sistema que se usó a lo largo de América del Norte y el sistema analógico que se utiliza aquí en la Argentina. Los sistemas similares, con variaciones ligeras, son el Nordic Mobile

Telephone (NMT) usado en Escandinavia, y el Total Access Communications System (TACS) usado en el Reino Unido, China, y otros países. Las asignaciones espectrales están en la región de 800-900 MHz.

Varios cientos de canales están disponibles dentro de la asignación del espectro. Un canal de una estación base se usa para cada conversación. En el handoff, el móvil del subscriber es dirigido para discontinuar el uso del canal viejo y poner a punto al nuevo, en el que se comunica con la nueva celda.

4.1.2: La Revolución de CDMA

La gran atracción de la tecnología CDMA desde el principio ha sido la promesa de un extraordinario aumento de capacidad por encima de las demás tecnologías inalámbricas de acceso múltiple. Los modelos mas simples sugieren que la mejora de capacidad pueda ser más de 20 veces con respecto a las normas celulares de banda angosta existentes, como AMPS en América del Norte, NMT en Escandinavia y TACS en el Reino Unido. Históricamente, la capacidad se calculó usando argumentos simples. La realidad, por supuesto, es mucho más compleja que los modelos idealizados. Las áreas de cobertura de las células reales son muy irregulares, no los hexágonos ordenados encontrados en los libros de texto. La capacidad requerida no es espacialmente uniforme, cambia dramáticamente con los horarios, y está a menudo sujeta a otras influencias ingobernables.

4.1.3: Aspectos de fondo

Un sistema de radio móvil de acceso múltiple idealizado, consiste en una familia de estaciones de base, o "celdas," geográficamente distribuidas en el área de servicio, y las estaciones móviles. Usamos el término "móvil" genéricamente para significar cualquier teléfono subscriber, se mueva o no. También se espera que usos no tradicionales, como módem inalámbricos en laptops, crezcan dramáticamente en un futuro cercano.

El espectro para telefonía móvil inalámbrica normalmente se asigna en división de frecuencia doble (FDD), con dos bandas separadas para los dos sentidos de transmisión (de la celda al móvil y del móvil a la celda). En los sistemas celulares están separados por 45 MHz, y las bandas de PCS por 80 MHz. Aunque ha habido algunas propuestas para el uso de división de tiempo doble (TDD), tal funcionamiento limita el área de cobertura, y no ha logrado aceptación general.

La comunicación entre las estaciones de base y las estaciones móviles son establecidas por una negociación en el origen de la llamada. Una vez que se establece la comunicación entre la base y el móvil, cualquier desplazamiento del móvil se detecta y el servicio se pasa de una estación de base a otra (handoff). Una celda a la vez presta servicio a cada móvil en los servicios de banda angosta.

El concepto de handoff se extiende a un multi-canal simultáneo "soft" handoff en CDMA.

Cuando la telefonía celular fue propuesta por Laboratorios Bell, la subdivisión de las células se citó como el método por el que el sistema crecería en su capacidad de tráfico. Sin embargo el éxito del mercado celular ha superado las predicciones por semejante margen que la subdivisión de células para aumentar la capacidad, como una solución práctica, está perdiendo efectividad. Los móviles AMPS tienen una capacidad muy

limitada para reducir la potencia transmitida, y al achicarse demasiado las células comienzan las interferencias entre las células que usan los mismos canales. Con una densidad alta, las células pequeñas también aumentan la frecuencia del handover, por eso aumenta la probabilidad que cualquier llamada se caiga. El alto costo asociado con los grandes números de células, como bienes raíces, equipos, y mantenimiento, llegan a ser un fragmento muy grande de capital y de costos operativos. Los gobiernos se resisten cada vez más a solicitudes para nuevas torres de antenas, que hacen que sus jurisdicciones se parezcan a un campo petrolero. Por todo esto, era necesario un método alternativo para mejorar la capacidad.

4.1.4: Reuso de frecuencia en los sistemas tradicionales

Tradicionalmente los sistemas de comunicación de radio han separado a los usuarios por canales de frecuencia, slots de tiempo, o ambos. Estos conceptos datan de los primeros días de la radio. Los sistemas celulares modernos empezaron con el uso de canalización de FM analógica. Más recientemente se han desarrollado varios sistemas digitales FDM-TDM híbridos, que ostensiblemente reforzaron la calidad del servicio y la capacidad. En todos estos sistemas, a cada usuario se le asigna un recurso de tiempo y frecuencia particular.

El concepto de reuso de frecuencia es fundamental para el concepto de telefonía celular. Aunque hay centenares de canales disponibles, si cada frecuencia se asignara a sólo una célula, la capacidad del sistema total igualaría al número total de canales: sólo unos pocos miles de subscriptores por sistema. Reusando canales en células distintas, el sistema puede crecer sin límites geográficos.

En sistemas grandes las asignaciones de los recursos de tiempo y frecuencia no pueden ser únicas. Deben reusarse en múltiples células para cubrir grandes áreas de servicio. El desempeño satisfactorio en estos sistemas depende críticamente del control de la interferencia mutua que surge del reuso. El concepto de reuso incluso nos es familiar en transmisión de televisión, donde no se reusan canales en ciudades adyacentes.

El reuso es extremadamente dependiente del hecho que la atenuación del campo electromagnético en las bandas celulares tiende a ser más rápido con la distancia que en el espacio libre. Las medidas han mostrado repetidamente que típicamente la intensidad de campo cae como R^{-n} , con $3 < n < 5$. En el espacio libre $n = 2$.

El reuso típico en telefonía celular (pre-CDMA) es razonado fácilmente considerando un sistema idealizado. Si asumimos que esa propagación es uniformemente R^{-n} , y los límites de la célula están en los puntos de igual intensidad de señal, entonces una área de servicio plana es cubierta óptimamente por la serie hexagonal clásica de células que se puede ver en la Fig. 4.1, donde los distintos colores indican frecuencias distintas.

La misma frecuencia no puede reusarse obviamente en cualquier par adyacente de células porque un usuario en el límite entre esas células recibiría ambas señales con igual amplitud, conduciendo a un inaceptablemente alto nivel de interferencia.

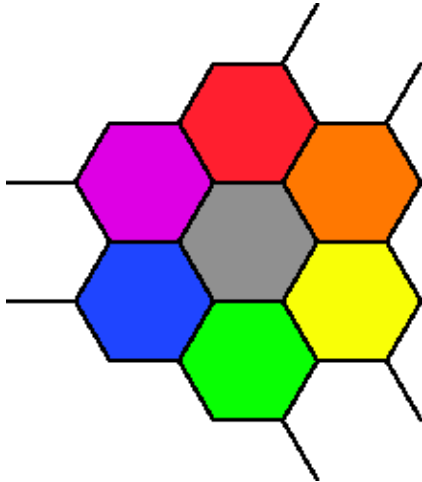


Fig.4.1: Células hexagonales adyacentes.

Un territorio puede “azulejarse” con células hexagonales, ordenadas de acuerdo con el modelo de siete mostrado en la figura 4.2. Así, si un único juego de canales se asigna a cada una de las siete células, entonces el modelo puede repetirse sin violar el requisito de adyacencia. Se usan siete juegos de canales, un juego en cada célula coloreada. Esta unidad de siete células se reproduce entonces sobre toda el área de servicio.

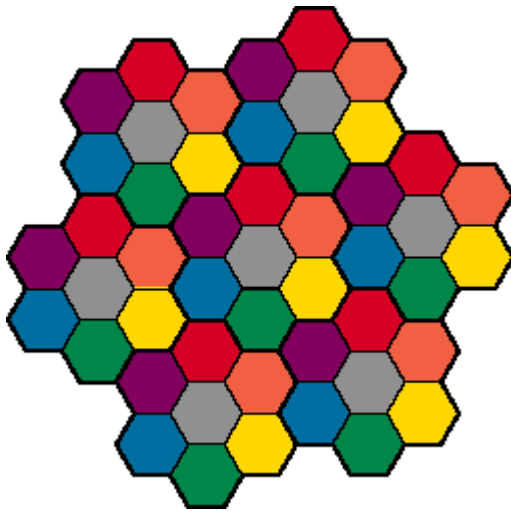


Fig. 4.2: Distribución de las células cubriendo el area de servicio.

Ninguna célula similarmente coloreada es adyacente, y por consiguiente ninguna célula adyacente usa el mismo canal . Mientras que los sistemas reales no se parecen a éstos en la forma hexagonal, el patrón de reuso de siete células es típico de lograr en la práctica. La capacidad de una célula simplemente es el número total de canales disponibles dividido por K . En Norteamérica se asignan aproximadamente 416 canales AMPS a cada operador (30 kHz de ancho cada uno, con una asignación total de 12.5 MHz en cada dirección - en la Argentina son 10 MHz). Con $K=7$ y 416 canales, hay aproximadamente 57 canales disponible por cada célula. A una carga típica de 0.05

Erlangs por subscriptor (se supone que sólo 1 de cada 20 está hablando en un momento dado), cada célula sirve a aproximadamente 1140 subscriptores.

La realidad es, por supuesto, mucho más complicada que este simple cuadro, pero los números están en lo correcto.

Sectorización de las Antenas

Las imágenes anteriores asumen que las células están usando antenas omnidireccionales. Podría esperarse que la capacidad del sistema pudiera ser aumentada mediante sectorización en las antenas. Las células son de hecho sectorizadas por los operadores, normalmente en tres partes. Es decir, cada sitio está provisto con tres juegos de antenas direccionales, con sus ejes separados por 120° . Desgraciadamente la sectorización no lleva en la práctica a un aumento en capacidad. La razón es que el aislamiento sector a sector, a menudo no más de unos dB, es insuficiente para garantizar una interferencia aceptablemente baja. Sólo en parte es debido a la pobre relación del frente a la parte de atrás de las antenas direccionales. Las extravagancias de la propagación electromagnética en el mundo real también conspiran para mezclar señales entre los sectores. El resultado práctico de la sectorización es sólo un aumento en la cobertura debido a la ganancia de la antena direccional. Nada se gana en reuso. Visto desde el punto de vista de los sectores, el reuso es $K = 7 * 3 = 21$, no 7.

4.1.5: El Reuso en CDMA

CDMA ofrece una respuesta al problema de la capacidad. La clave de su alta capacidad es el uso de señales portadoras pseudo-aleatorias, como fue sugerido hace décadas por Claude Shannon. En lugar de dividir ya sea el espectro o el tiempo en distintas "hendeduras" a cada usuario se le asigna una diferente portadora pseudo-aleatoria. Las aplicaciones prácticas de este principio siempre han usado pseudo-ruido digitalmente generado, en lugar de verdadero ruido térmico. Aun cuando las formas de onda no son rigurosamente ortogonales, son lo suficientemente descorrelacionadas. Los beneficios básicos se conservan, y se simplifican los transmisores y receptores porque pueden llevarse a cabo usando dispositivos digitales de alta densidad.

El mayor beneficio de las portadoras pseudo-aleatorias es que la sensibilidad del sistema a la interferencia se altera fundamentalmente. Los sistemas tradicionales con división en el tiempo o en frecuencia deben diseñarse con una relación de reuso que satisface la situación del peor caso de interferencia, pero sólo un pequeño fragmento de los usuarios experimenta aquel peor caso realmente. El uso de portadoras pseudo-aleatorias, con todos los usuarios ocupando el mismo espectro, hace que el ruido eficaz sea la suma de todas las señales de los otros usuarios. El receptor correlaciona su entrada con la portadora deseada, reforzando la relación señal a ruido en el detector. La mejora supera al ruido sumado lo suficiente para proporcionar una SNR adecuada al detector. Debido a que la interferencia se suma, el sistema no es más sensible al peor-caso, sino a la interferencia promedio. El reuso de frecuencia es universal, es decir, todas las células pueden utilizar cada frecuencia en CDMA. El patrón de reuso es ahora como el que se ve en la Fig. 4.3.

Las células de color arco iris indican que la totalidad de la señal pasabanda de 1.25 MHz es usada por cada usuario, y ese mismo espectro se reusa en cada célula.

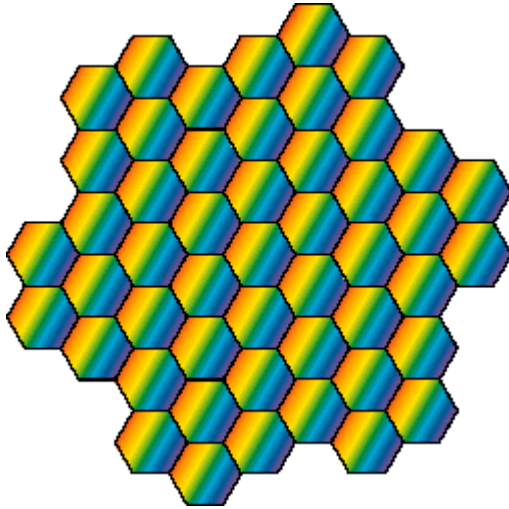


Fig. 4.3: Reuso de frecuencias en CDMA.

4.1.6: Otros Standards Celulares actuales

Los diferentes tipos de sistemas celulares emplean métodos diversos de acceso múltiple. Los sistemas celulares analógicos tradicionales, como aquéllos basados en el Advanced Mobile Phone Service (AMPS) y Total Access Communications System (TACS) standards, usan Frequency Division Multiple Access (FDMA).

Por ejemplo, los sistemas AMPS usan 30 kHz de espectro para cada canal. AMPS de banda angosta (NAMPS) requiere sólo 10 kHz por canal. Los canales de TACS son de 25 kHz de ancho. Con FDMA, sólo un subscriptor por vez se asigna a un canal.

Ninguna otra conversación puede tener acceso a este canal hasta que la llamada del subscriptor se termine, o hasta que esa llamada original sea pasada a un canal diferente por el sistema (Handoff).

Un método de acceso múltiple comúnmente empleado en los nuevos sistemas celulares digitales es el Time Division Multiple Access (TDMA). Los standards TDMA digitales incluyen al North American Digital Cellular (conocido por su número de norma IS-54), el europeo Global System for Mobile Communications (GSM), y el Personal Digital Cellular (PDC).

Los sistemas de TDMA normalmente se dividen en porciones de espectro llamadas cada una una "portadora". Cada portador es entonces dividido en slots de tiempo.

Sólo un subscriptor por vez se asigna a cada slot de tiempo, o canal. Ninguna otra conversación puede acceder a este canal hasta que la llamada del subscriptor se termina, o hasta que esa llamada original pase a un canal diferente por handoff.

Por ejemplo, en los sistemas IS-54, diseñados para coexistir con sistemas AMPS, se dividen cada 30 kHz de espectro en tres canales. PDC divide cada 25 kHz de espectro en tres canales. Los sistemas de GSM crean 8 canales en cada portadora de 200 kHz de ancho de banda.

4.1.7: El Standard Celular CDMA

Con CDMA, se usan códigos digitales distintos, en lugar de frecuencias de RF separadas o slots de tiempo, para diferenciar a los diferentes subscriptores. Los códigos son compartidos por la estación móvil (teléfono celular) y la estación base (celda), y son secuencias pseudo-aleatorias. Todos los usuarios comparten el mismo rango de espectro de radio.

Para la telefonía celular, CDMA es una técnica de acceso múltiple digital especificada por la Telecommunications Industry Association (TIA) como "IS-95."

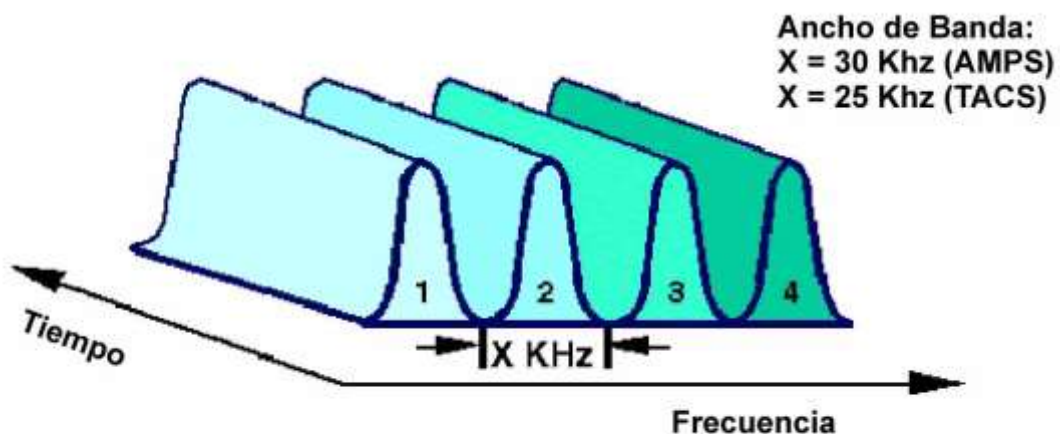
En marzo de 1992, el TIA estableció el TR-45.5, un sub-comité con la misión de desarrollar un standard celular digital de espectro esparcido, que fue aprobado en julio de 1993 como el CDMA IS-95 standard.

Los sistemas IS-95 dividen el espectro de radio en portadores de 1,250 kHz (1.25 MHz) de ancho de banda. Uno de los aspectos extraordinarios de CDMA es que mientras hay límites ciertamente al número de llamadas telefónicas que pueden ser manejadas por un portador, éste no es un número fijo. Más bien, la capacidad del sistema dependerá de varios factores diferentes. Esto se discutirá posteriormente.

4.1.8: Comparación entre técnicas de Acceso múltiple

Es más fácil entender CDMA si se compara con otras tecnologías de acceso múltiple. Las secciones siguientes describen las diferencias fundamentales entre una tecnología analógica dividida en frecuencia (FDMA), una tecnología digital dividida en el tiempo (TDMA) y una tecnología digital dividida en códigos (CDMA).

FDMA- Frequency División Múltiple Access



1 Usuario por cada canal de banda angosta

Figura 4.4: FDMA

FDMA se usa en los sistemas celulares analógicos. Cada usuario se asigna una porción discreta del espectro de RF. FDMA permite a sólo un usuario por canal, ya que le permite al usuario usar el canal 100% del tiempo. Por consiguiente, sólo la frecuencia se usa para definir canales.

TDMA - Time División Múltiple Access

El punto importante para entender sobre TDMA es que todavía se asignan usuarios a una porción discreta del espectro de RF, pero los múltiples usuarios comparten ahora la portadora de RF, en distintos slots de tiempo. Cada uno de los usuarios alterna el uso del canal de RF. La división de frecuencia todavía es empleada, pero estas portadoras son subdivididas ahora además en algún número de slots de tiempo.

Un usuario se asigna a un slot de tiempo particular en una portadora y sólo puede enviar o recibir información en esos momentos. Esto es así estén usándose o no los otros slots de tiempo. El flujo de datos no es continuo para cualquier usuario, sino que se envía y se recibe en "bursts." Los burst se reagrupan en el receptor, y parecen proporcionar sonido continuo porque el proceso es muy rápido.

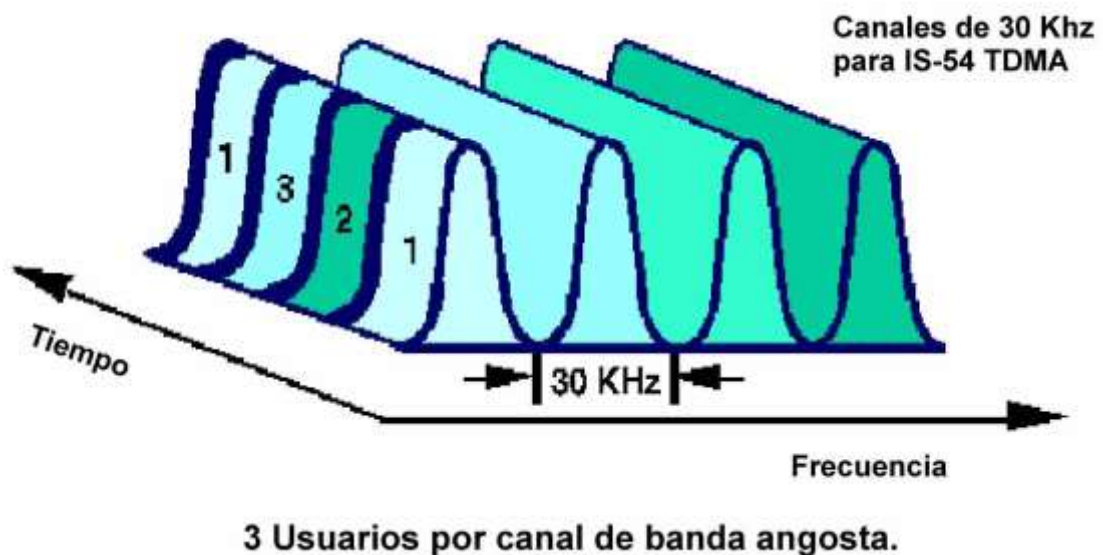


Figura 4.5: TDMA

CDMA - Code División Múltiple Access

IS-95 emplea una técnica de acceso múltiple de espectro esparcido llamada Secuencia Directa (DS) CDMA. Durante una llamada, un código binario se asigna a cada usuario. El código es una señal generada por modulación lineal con secuencias de ruido pseudoaleatorias (PN) de banda ancha. Como resultado, DS CDMA usa señales mucho más anchas que aquellas usadas en otras tecnologías. Las señales de banda ancha reducen la interferencia y permiten reuso de frecuencia en cada célula. No hay ninguna división de tiempo, y todos los usuarios usan la portadora íntegra, todo el tiempo.

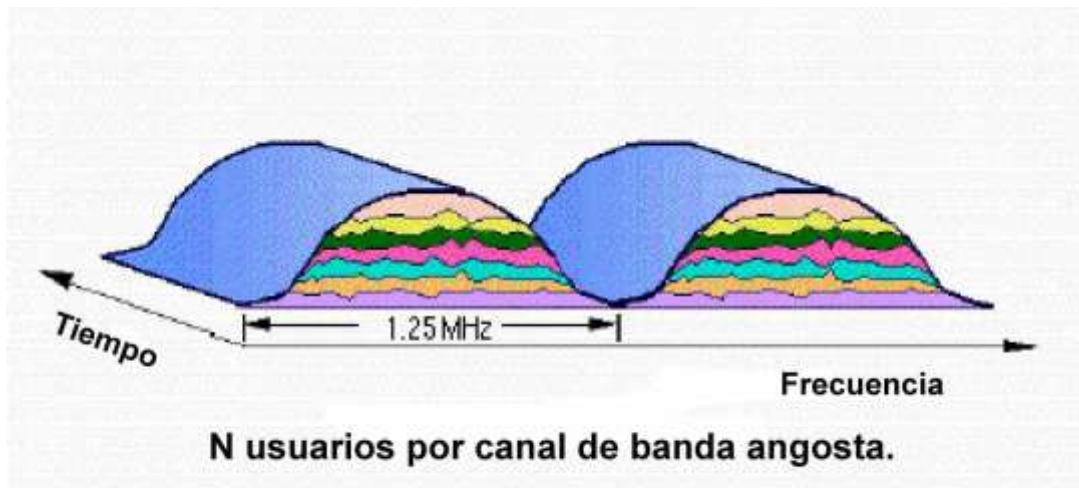


Figura 4.6: DS-SS

4.1.9: La Tecnología de CDMA

Aunque la aplicación de CDMA en telefonía celular es relativamente nueva, no es una nueva tecnología. CDMA se ha usado en muchas aplicaciones militares por presentar varias ventajas, como anti-jamming (debido a que una señal de espectro esparcido es difícil de interferir o bloquear), y comunicaciones seguras (una señal del espectro esparcido es muy difícil de descubrir, y más aún de descifrar).

Espectro Esparcido

CDMA es una tecnología de "espectro esparcido", que significa que extiende la información contenida en una señal particular sobre un ancho de banda mucho mayor que la señal original. Esto está en contraposición con las tecnologías tradicionales, que usan el menor ancho de banda posible para transmitir.

Una llamada de CDMA comienza con una velocidad normal de 9600 bits por segundo (9.6 kilobits por segundo). Esto es después esparcido a una frecuencia transmitida de aproximadamente 1.23 Megabits por segundo. Este "esparcido" se hace aplicando códigos digitales a los bits del mensaje original. Estos bits se transmiten al mismo tiempo con las señales de todos los otros usuarios en esa célula. Cuando la señal se recibe, los códigos se remueven de la señal deseada, separando a los usuarios y devolviendo la llamada a su velocidad de 9600 bps.

Los usos tradicionales del espectro esparcido están en las operaciones militares.

Debido al amplio ancho de banda de una señal de espectro esparcido, es muy difícil bloquearla, interferirla, e identificarla. La señal es muy difícil de descubrir porque aparece como nada más de un levantamiento ligero en el "piso del ruido" o nivel de la interferencia. Con otras tecnologías, se concentra la energía de la señal en una banda más angosta, que la hace más fácil de descubrir.

El aumento de la privacidad es innato en la tecnología de CDMA. Las llamadas telefónicas de CDMA estarán seguras ya que, a diferencia de una conversación

analógica, un receptor de radio simple no podrá discriminar conversaciones digitales individuales de entre la señal de RF total en una banda de frecuencias usada con CDMA.

Sincronización

En las fases finales de la codificación del vínculo de radio desde la estación base al móvil, CDMA agrega un "código pseudo-aleatorio" especial a la señal, que se repite después de una cantidad finita de tiempo. Las estaciones base en el sistema se distinguen entre sí transmitiendo porciones diferentes del código en un momento dado. En otras palabras, las estaciones base transmiten versiones desplazadas en tiempo del mismo código pseudo-aleatorio. Para asegurar que los desplazamientos de tiempo usados permanezcan únicos entre sí, las estaciones de CDMA deben permanecer sincronizadas a una referencia de tiempo común.

El Sistema del Posicionamiento Global (GPS) proporciona esta referencia de tiempo común y precisa. GPS es un sistema basado en satélites, capaz de proporcionar un medio práctico y económico de determinar posición continua, velocidad y tiempo a un número ilimitado de usuarios.

El equilibrio de las características

La cobertura de cada célula depende de la manera que el sistema se diseña. De hecho, tres características básicas del sistema -Cobertura, Calidad y Capacidad- deben equilibrarse entre sí para lograr el nivel deseado en el desempeño del sistema.

En un sistema CDMA estas tres características están firmemente interrelacionadas. Podría lograrse una capacidad aun más alta a través de algún nivel de degradación en la cobertura y/o calidad. Puesto que estos parámetros están todos entrelazados, los operadores no pueden lograr lo mejor de cada uno: tres veces la más amplia cobertura, 40 veces la capacidad, y un sonido de calidad "CD" no pueden lograrse en forma conjunta. Por ejemplo, el vocoder de 13 kbps proporciona calidad de sonido mejor, pero reduce la capacidad del sistema comparado con un vocoder de 8 kbps.

4.2: Beneficios de CDMA

Cuando se implementa en un sistema de telefonía celular, la tecnología de CDMA ofrece numerosos beneficios a los operadores celulares y a sus suscriptores. Lo siguiente es una visión global de los beneficios de CDMA.

1. La Capacidad aumenta de 8 a 10 veces comparada con el sistema analógico AMPS y 4 a 5 veces comparada con el sistema digital europeo GSM
2. La Calidad de la llamada mejora, con sonido mejor y más estable comparado a los sistemas AMPS.
3. La Planificación del sistema se simplifica, debido al uso de la misma frecuencia en cada sector de cada célula
4. La Privacidad aumenta.

5. Características de cobertura mejoradas, permitiendo la posibilidad de menos sitios de celdas.
6. Se aumenta el tiempo de conversación para los teléfonos móviles (mayor duración de las baterías).
7. El Ancho de Banda puede ajustarse a la Demanda.

4.2.1: Aumento de capacidad

Pueden lograrse aumentos de capacidad en sistemas celulares mediante dos maneras:

- Consiguiendo más canales por MHz de espectro.
- Consiguiendo mejor reuso de canales por unidad de área geográfica.

NAMPS es un ejemplo de una tecnología que logra capacidad mayor a través del método #1 (más canales por MHz de espectro). En lugar de un canal en 30kHz como en AMPS, NAMPS tiene tres canales en 30kHz, proporcionando así tres veces la capacidad de AMPS.

GSM es un ejemplo de un sistema que usa el método #2 (más reuso de canales por unidad de área geográfica). GSM permite 9dB C/I (relación portadora-interferencia) en lugar de los 17dB tradicionales usados en TACS (la tecnología de FDMA analógica en la banda de 900 MHz). Esto le permite a GSM poner más cerca los sitios de la células y se traduce en aproximadamente dos veces la capacidad de TACS. Un sistema de GSM de media-velocidad, usando los métodos #1 y #2, produciría aproximadamente un aumento de capacidad 4 a 5 veces por encima de TACS analógico.

CDMA ofrece una mayor capacidad del sistema que el ofrecido por sistemas celulares analógicos tradicionales usando el método #2. Permite el reuso de la misma frecuencia en cada sector de cada célula. Dependiendo del diseño del sistema específico, debe poder lograrse al menos una capacidad de 8 a 10 veces por encima de AMPS.

La capacidad real variará de célula a célula y sector a sector, dependiendo del terreno, niveles de interferencia, características de propagación y varios otros factores.

CDMA y reuso en las Células

Uno de los principales claves en el diseño de telecomunicaciones celulares es el uso de las mismas frecuencias, una y otra vez, en una región geográfica particular. Esto incrementa la capacidad del espectro usado en sistemas celulares cuando se compara a la práctica tradicional de cubrir el área más ancha posible con un solo sitio de transmisión. Sin embargo, en muchos tipos de sistemas celulares, no es posible usar cada frecuencia en cada sitio de la célula debido a la interferencia que resultaría.

Por consiguiente, con estas otras tecnologías celulares, es necesario planear qué frecuencias se usan en cada sitio de la célula para minimizar la interferencia entre células en la misma frecuencia. Este requisito ha llevado a "patrones de reuso de frecuencia". Los sistemas AMPS (y aquellos diseñados para ser compatibles con AMPS) a menudo usan una configuración de tres sectores y se diseñan con un modelo de reuso cada 7 células llamado "N = 7". En otras palabras, típicamente puede usarse en

cualquier célula un séptimo de todo el espectro de frecuencias celulares asignado a un operador.

Con CDMA, pueden recibirse señales en la presencia de niveles altos de interferencia, y todavía resultar en una misma, o mejor, calidad de la llamada. Todos los usuarios en una portadora comparten el mismo espectro de RF. La misma portadora de RF se usa en cada sector de cada célula, y en cada célula del sistema. Esto equivale a un modelo de reuso $N=1/S$, donde S es el número de sectores por la célula. Este reuso de frecuencias $N=1/S$ es lo que le da su capacidad mayor a CDMA por encima de AMPS y otras tecnologías.

Eb/No y Umbral de Interferencia

Eb/No es la relación en dB entre la energía de cada dígito binario de información y la densidad espectral del ruido. Representa la relación señal a ruido durante un solo bit en el vínculo inverso y proporciona una medida del desempeño del vínculo de CDMA entre el móvil y la célula. El ruido es una combinación de la interferencia de fondo y la interferencia creada por otros usuarios en el sistema. Una disminución en la relación de Eb/No indica que el nivel relativo de interferencia, comparado al nivel de la información, está aumentando. Esto bajará la calidad de la voz en la conversación. Mientras todos los sistemas celulares digitales usan codificación para corrección de errores, los sistemas basados en modulación digital de banda angosta generalmente usan esquemas menos sofisticados. Por consiguiente, para preservar una alta calidad de voz, los operadores de sistemas de banda angosta necesitan Eb/No más altos. Esto lleva a una necesidad de limitar el número de usuarios en el sistema, reduciendo su capacidad.

CDMA, por otro lado, usa una corrección de errores avanzada así como un demodulador digital, reduciendo la relación de Eb/No requerida. Usando un Eb/No más bajo para alcanzar los estándares de calidad de voz, CDMA logra más capacidad y usa menos potencia del transmisor que los sistemas de banda angosta.

CDMA describe Eb/No en términos de la Frame Erasure Rate (FER). Usando un umbral de interferencia, el sistema de CDMA borra los Frames de información recibidos que contienen demasiados errores como para ser decodificados correctamente. El FER, entonces, describe el número de Frames que se borraron debido a su pobre calidad. Por consiguiente, cuando el nivel de Eb/No aumenta, el FER disminuye, y la calidad de voz del sistema mejora. Recíprocamente, cuanto más alto sea el FER aceptable, más alta será la capacidad total de las células.

4.2.2: Mejor Calidad de la Llamada

Los sistemas de telefonía celulares que usan CDMA pueden proporcionar sonido de calidad más alto y menos llamadas dejadas caer que los sistemas basados en otras tecnologías. Varias características inherentes al sistema producen esta alta calidad.

Los esquemas de detección y corrección de error avanzados aumentan la probabilidad que se decodifiquen correctamente los frames. Los vocoders sofisticados ofrecen una alta velocidad de codificación y reducen el ruido del fondo.

CDMA se aprovecha de varios tipos de diversidad para mejorar la calidad de la conversación:

- diversidad de frecuencia (protección contra el desvanecimiento selectivo de frecuencia)
- diversidad espacial (dos antenas receptoras)
- diversidad de caminos (el receptor "Rake" mejora la recepción de una señal que experimente multipath, y realmente mejora la calidad del sonido)
- diversidad de tiempo (entrelazando y codificando)

La técnica de diversidad se basa en el hecho de que, al llegar múltiples copias de la información, es más probable que un receptor especialmente diseñado pueda aprovechar ésto y decodificar el mensaje original sin errores que si se recibiera solo una copia.

El llamado "Soft" Handoff contribuye a una alta calidad de voz proporcionando una conexión "hecha antes de la interrupción". "Softer" Handoffs entre los sectores de la misma célula proporcionan beneficios similares.

El control preciso de potencia asegura que todos los móviles estén muy cerca del nivel de potencia óptimo para proporcionar la calidad de voz más alta posible.

Se ha considerado como muy alta la calidad de voz para CDMA en pruebas que la comparan a otras tecnologías.

Avanzada Detección y Corrección de Errores

El estándar de interface aérea IS-95 CDMA especifica poderosos algoritmos de detección y corrección de errores. Pueden hallarse datos de voz inválidos y pueden corregirse o manipularse para minimizar su impacto en la calidad de la conversación.

Vocoders sofisticados

PCM es la norma de codificación de voz usada en sistemas terrestres. Es simple como era necesario en los años sesenta, pero no muy eficiente. Tiene la calidad de sonido que se desea igualar en un sistema inalámbrico. Las comunicaciones mediante cables todavía usan PCM, ya que el ancho de banda se ha vuelto bastante barato vía fibra óptica y/o enlaces de microondas.

Los Vocoders inalámbricos, por el contrario, están restringidos en ancho de banda. Existen actualmente varios tipos de vocoders, ofreciéndoles a los operadores la alternativa entre la capacidad más alta y la calidad de voz mejor. Los sistemas de CDMA inicialmente usan un vocoder de 8 kbps de velocidad variable, revisión IS-96A. El vocoder transmite 8 kbps de información de voz a 9.6 kbps, cuando se le agregan los bits de corrección de error. Como regla general, las velocidades de vocoder más altas proporcionan una representación más precisa de una señal de voz. Sin embargo, los vocoder más viejos, menos sofisticados, pueden ser incapaces de igualar la calidad de voz de los más nuevos diseños, a pesar de una velocidad más alta.

El vocoder de CDMA también aumenta la calidad de la llamada suprimiendo el ruido del fondo. Cualquier ruido que sea de naturaleza constante se elimina. El sonido de fondo constante es visto por el vocoder como ruido que no lleva información inteligente, y se remueve tanto como sea posible. Esto refuerza la claridad de la voz en ambientes ruidosos, como dentro de los automóviles, o en lugares públicos ruidosos.

Múltiples formas de Diversidad

CDMA se aprovecha de varios tipos de diversidad, todos los cuales lleva a mejorar la calidad de la conversación. Los cuatro tipos son diversidad de frecuencia, diversidad espacial, diversidad de caminos y diversidad de tiempo.

Diversidad de frecuencia: En las comunicaciones de radio, se producen decaimientos o "agujeros" en ciertas frecuencias. Debilitamiento que ocurre en un ambiente de múltiples caminos, cuando dos o más señales se combinan y cancelan entre sí. Las transmisiones de banda angosta son especialmente propensas a este fenómeno. Para las señales de banda ancha como CDMA, esto es mucho menor problema. La señal de banda ancha está, por supuesto, también sujeta al desvanecimiento selectivo en frecuencia, pero la mayoría de la señal no es afectada y el efecto global es mínimo.

Por ejemplo, consideremos lo que pasa cuando hay un desvanecimiento selectivo en frecuencia de 12 dB de profundidad y 400kHz ancho. Para una señal de CDMA, que abarca 1.25 MHz, esto sólo afecta 1/3 del ancho de banda total. Ya que se esparce la energía de una llamada telefónica en el espectro total, el efecto del desvanecimiento se promedia, y representa una caída en señal de aproximadamente 2 dB.

Si este mismo desvanecimiento de 12 dB y 400kHz cae sobre una señal de banda angosta de 30kHz, como en sistemas AMPS o IS-54, los resultados son bastante diferentes. La señal entera es entonces afectada por esto debilitamiento. El resultado será una caída de señal de 12 dB completos. Éste es un golpe mucho más serio a la señal, y podría llevar a una grave degradación en la calidad de voz, o incluso dejar caer una llamada. Esto puede apreciarse en la Fig. 4.7.

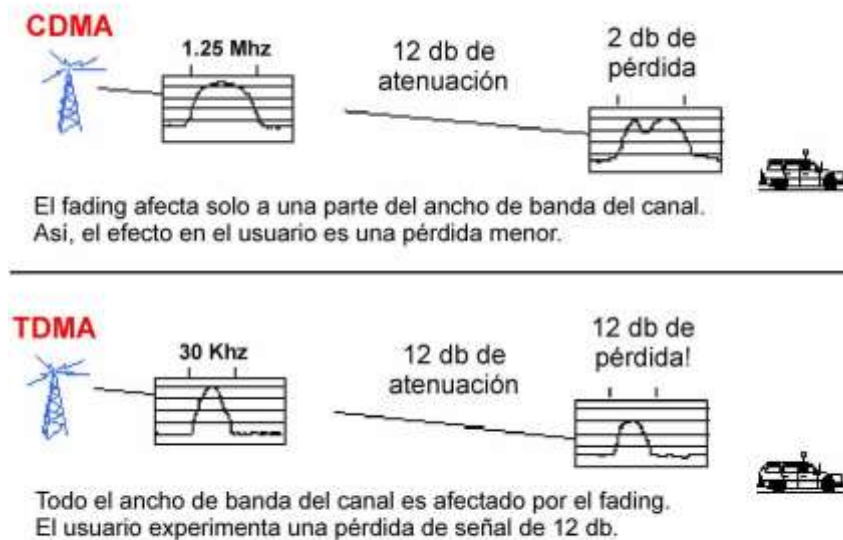


Figura 4.7: Beneficios en Calidad por Diversidad de Frecuencia en CDMA

De manera semejante, CDMA es más resistente a la interferencia o "jamming". En una tecnología de banda angosta típica como AMPS o TDMA, si esta interferencia de banda angosta estuviera en la misma frecuencia que la señal de interés, y tuviera la magnitud suficiente, habría tapado totalmente la señal.

Sin embargo, una interferencia de banda angosta tiene poco efecto en una señal de CDMA. En el proceso de "recolección", cuando la señal recibida se combina con el código que la esparció originalmente, solo la señal de interés tiene correlación con el código, y la señal deseada "salta" fuera del ruido. Una interferencia de banda angosta es una señal aleatoria, así que no tendrá correlación con ningún código usado. Por consiguiente, en el proceso se esparce la energía de la interferencia de banda angosta por el espectro y no interfiere con la señal de interés. Esta inmunidad fundamental a la interferencia es uno de los beneficios más atractivos de CDMA.

Diversidad espacial: La Diversidad espacial se refiere al uso de dos antenas separadas por alguna distancia física. El principio de diversidad espacial reconoce que cuando un móvil se está moviendo, crea un modelo de crestas y valles de señal. Cuando uno de estos valles cae en una antena causará que la energía de la señal recibida decaiga. Sin embargo, si una segunda antena se pone a una cierta distancia física, estará fuera del área de señal nula y así recibirá la señal a un nivel aceptable.

Diversidad de caminos: En comunicaciones de radio, hay normalmente más de un camino de RF del transmisor al receptor. Por consiguiente, múltiples versiones de la misma señal están normalmente presentes en el receptor. Sin embargo, estas señales que han llegado a lo largo de los diferentes caminos, están todo el tiempo cambiando con respecto a las otras debido a las diferencias en la distancia que cada señal ha viajado. Este efecto "multipath" se crea cuando una señal transmitida se refleja en objetos del ambiente (edificios, montañas, aviones, camiones, etc.). Estas reflexiones, combinadas con la señal transmitida, crean un patrón móvil de crestas y valles de la señal.

Cuando un receptor de banda angosta se mueve a través de estos valles hay una caída súbita en la intensidad de la señal. Este desvanecimiento causará ya sea una calidad de llamada inferior, más ruidosa o, si el desvanecimiento es suficientemente severo, la pérdida de señal y una llamada caída.

Aunque el multipath usualmente es perjudicial para una señal analógica o de TDMA, realmente es una ventaja para CDMA, debido a que los receptores "rake" de CDMA pueden emplear el multipath para mejorar una señal. El receptor de CDMA tiene varios "fingers" que son capaces de recibir las varias señales presentes. El receptor busca las tres señales más fuertes del multipath recibidas, compensa los distintos retardos, y luego los suma para producir una señal que es mejor que cualquiera de los componentes individuales de la señal. Sumar las señales del multipath refuerza la señal en lugar de degradarla.

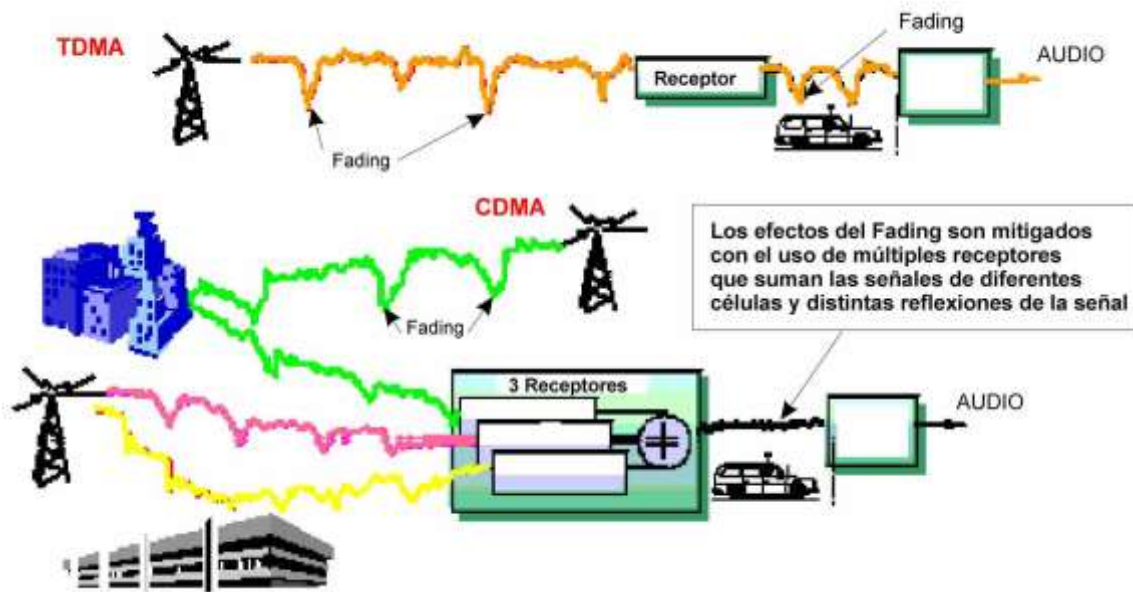


Figura 4.8: Beneficios de Calidad por Diversidad de Caminos

Diversidad de Tiempo: Los sistemas de CDMA usan códigos para corrección de errores, seguido de un entrelazado (interleaving).

Los esquemas de corrección de error son más eficaces cuando los bits erróneos en el flujo de datos están esparcidos más uniformemente con el tiempo. Separando los fragmentos de información en el tiempo, una súbita ruptura en los datos no causará la ruptura correspondiente en la señal de voz decodificada. Cuando los Frames son reunidos y reorganizados en el decodificador, cualquier "rotura" de los bits del Frame estará dispersa en una extensión relativamente más larga de conversación real, reduciendo o eliminando el impacto en la calidad de voz de la llamada.

El proceso de Interleaving, que es común a la mayoría de los sistemas de comunicación digitales, garantiza que no se transmitan bits contiguos de datos consecutivamente. Con esto se busca que los errores se dispersen, para mejorar el desempeño del decodificador.

Soft Handoff

Con el tradicional "hard" handoff que se usa en todos los otros tipos de sistemas celulares, el móvil deja caer un canal antes de conectarse al próximo. Cuando una llamada está en una condición de "soft" handoff, un usuario móvil es supervisado por dos o más células, y la circuitería compara la calidad de los Frames recibidos de cada célula. El sistema puede aprovecharse de los cambios momento a momento en la potencia de la señal de cada célula y escoger la mejor señal.

Esto asegura que el mejor Frame posible se use en el proceso de decodificación de CDMA. Los transcodificadores pueden literalmente oscilar de un lado a otro entre las celdas involucradas en un "Soft" handoff Frame a Frame, si esto es necesario para seleccionar el mejor frame posible.

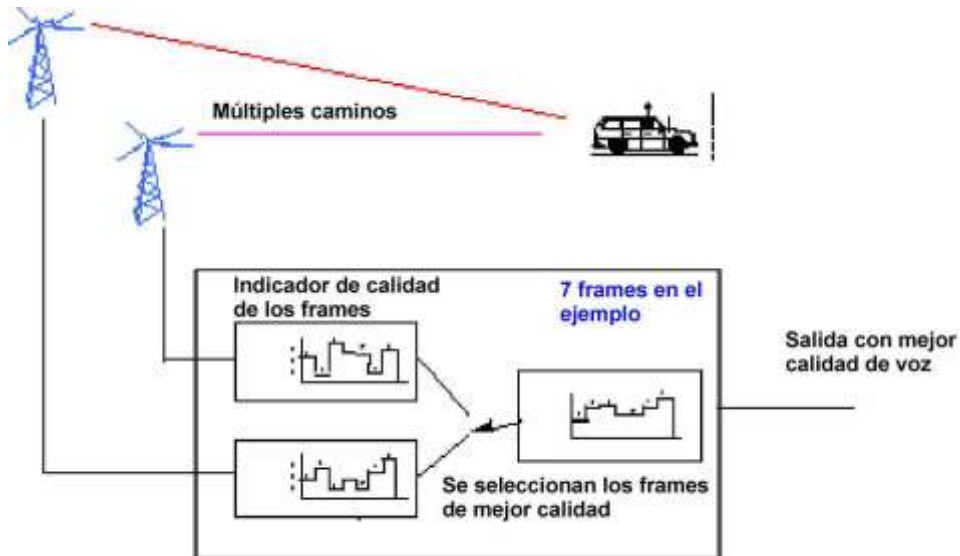


Figura 4.9: CDMA Soft Handoff Mejora la Calidad de los Frames

Los soft handoffs también contribuyen a una calidad superior de la llamada proporcionando una nueva conexión antes de interrumpir la anterior. Esto elimina el corte de conversación que uno oye con otras tecnologías cuando la conexión de RF se interrumpe con una célula para establecer la llamada en la célula de destino durante el handoff. En tecnologías de banda angosta hay una "competencia" por la señal, y cuando la Célula B "gana" sobre la Célula A, el usuario es desconectado por célula A (Hard handoff). En CDMA las células "trabajan en equipo" para obtener el mejor flujo de información posible. Eventualmente, la célula A ya no recibirá una señal lo suficientemente fuerte del móvil, y los transcoders estarán obteniendo sólo Frames de la Célula B. El handoff se habrá completado, sin ser detectado por el usuario. Los handoffs de CDMA no crean el "agujero" en la conversación que se oye en otras tecnologías.

Algunos sistemas celulares padecen el "efecto ping pong" de una llamada que se cambia repetidamente de una a otra célula cuando la unidad del suscriptor está cerca del límite entre ambas. En el peor de los casos, semejante situación aumenta la posibilidad de que una llamada se deje caer durante uno de los handoffs, y como mínimo, causa handoffs más ruidosos. El soft handoff de CDMA evita este problema completamente.

Y finalmente, debido a que una llamada de CDMA puede depender al mismo tiempo en una condición de soft handoff de tres células, la posibilidad de una pérdida de conexión de RF (una llamada dejada caer) es muy reducida.

CDMA también proporciona "Softer" handoffs. Un "Softer" handoff ocurre cuando un suscriptor está comunicando simultáneamente con más de un sector de la misma célula.

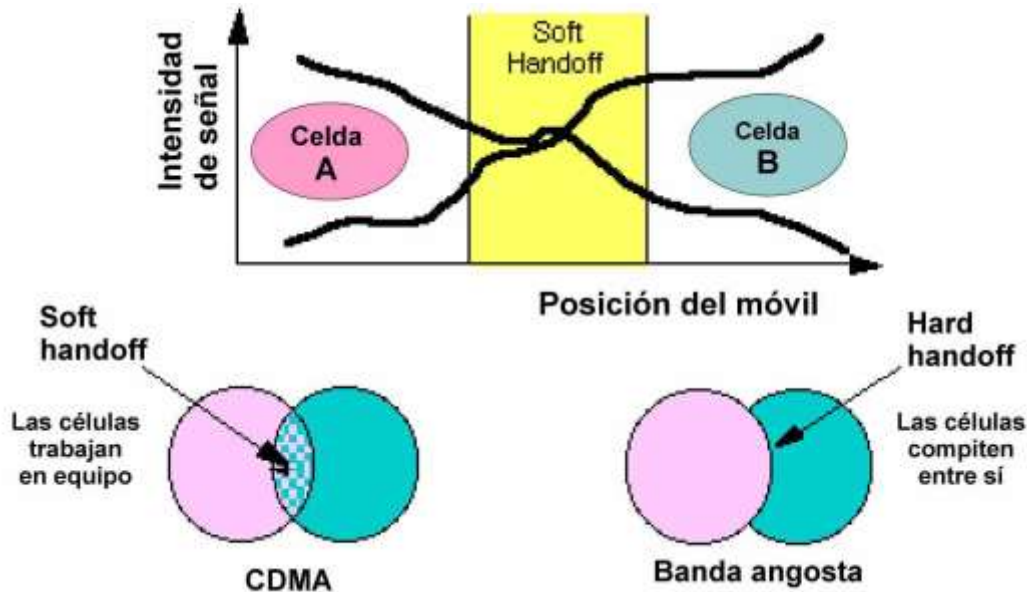


Figura 4.10: CDMA Soft Handoff Utiliza dos o más células

Control de Potencia preciso

El control de potencia de CDMA no sólo incrementa la capacidad (como se describió antes) sino que también aumenta la calidad de la conversación minimizando y venciendo a la interferencia. Los algoritmos de control de potencia de CDMA están diseñados para reducir el nivel de potencia de la señal al mínimo requerido para mantener una llamada de calidad.

4.2.3: Planificación del Sistema simplificada

Todos los usuarios en una portadora de CDMA comparten el mismo espectro de RF. Este $N=1/S$ reuso de frecuencias (donde S = número de sectores por célula) es un factor que le da su gran capacidad a CDMA, por encima de AMPS y otras tecnologías, pero también hace más simples ciertos aspectos de la planificación del sistema. Los Ingenieros ya no tendrán que realizar la planificación de frecuencia detallada que es necesaria en sistemas analógicos y de TDMA. Debido a que la planificación de frecuencia es innecesaria, esa parte del trabajo de ingeniería anteriormente requerido como parte de un plan inicial del sistema se elimina. Más significativamente aun, también se eliminan re-sintonizaciones de frecuencia para la expansión de un sistema. Si un cliente quiere agregar sitios de células o canales, ya no requerirá un nuevo plan de frecuencias para hacerlo.

Note que cuando CDMA se agrega como una extensión en un sistema analógico existente, la planificación de frecuencias es necesaria para despejar el espectro que será utilizado por CDMA.

4.2.4: Mayor Privacidad

La mayor privacidad por encima de otros sistemas celulares, analógicos y digitales, es inherente en la tecnología de CDMA. Es sumamente difícil para alguien bloquear la señal de CDMA. Además, puesto que los frames digitalizados de información se esparcen en una porción ancha del espectro, es improbable que alguien pueda ser capaz de escuchar una conversación. Además, hay una "encriptación" digital que proporciona niveles aun mayores de seguridad y privacidad.

4.2.5: Cobertura mejorada

Al comienzo de un nuevo sistema, hay menos subscriptores, así que se necesitan menos células para manejar el tráfico. Sin embargo, hay todavía necesidad de proporcionar una amplia cobertura geográfica inicial.

Una célula de CDMA tiene un alcance mayor que una célula analógica o digital típica. Por consiguiente, se necesitan menos células para cubrir la misma área. Dependiendo de la carga y la interferencia, la reducción en la cantidad de células podría ser tanta como 50% comparada con GSM.

El alcance mayor es debido al hecho que CDMA usa un receptor más sensible que otras tecnologías.

Por favor note que esa reducción del número de células necesarias es cierta para prestadores que EMPIEZAN con CDMA. Si se comparte el espectro con AMPS (como ocurre generalmente) se utilizan las mismas celdas ya instaladas.

4.2.6: Aumento del Tiempo de Conversación Portátil

Debido al preciso control de potencia y otras características del sistema, las unidades de CDMA transmiten normalmente sólo un fragmento de la potencia de los teléfonos analógicos y de TDMA. Esto permite a los móviles tener un tiempo de conversación y de "standby" más largo.

4.2.7: Ancho de Banda según la Demanda

Un canal de CDMA proporciona un recurso común que todos los móviles en un sistema utilizan basado en sus propias necesidades específicas, si están transmitiendo voz, datos, fax, u otras aplicaciones. En un momento dado, la porción de este ancho de banda común que no es usada por un móvil está disponible para el uso por cualquier otro móvil. Esto proporciona una tremenda flexibilidad, que puede explotarse para proporcionar características poderosas, como servicios de velocidad superiores para datos. Además, debido a que los móviles utilizan el ancho de banda común independientemente, estas características pueden coexistir fácilmente en el mismo canal de CDMA.

4.3: Capacidad en CDMA

La capacidad es determinada por el equilibrio entre la SNR requerida para cada usuario, y la ganancia del proceso de esparcido en frecuencia. La figura de mérito de un receptor digital bien diseñado es la dimensión necesaria de relación señal a ruido (SNR)

$$E_b/N_0 \equiv \frac{\text{Energía por bit}}{\text{Densidad espectral de potencia del ruido + interferencia}} \quad (\text{ecuación 4.1})$$

El "ruido" parte de SNR, en un sistema de espectro esparcido es de hecho la suma del ruido térmico y la interferencia de los otros usuarios. La SNR necesaria para lograr una probabilidad de error deseada depende de varios factores, como la codificación usada para corrección de errores, y las condiciones ambientales de multipath y desvanecimiento de la señal. Para los receptores comercialmente usados en CDMA, va típicamente de aproximadamente 3 dB a 9 dB.

La energía por bit está relacionada con la potencia de la señal y la velocidad de los datos:

$$E_b = P_s / R \quad (\text{ecuación 4.2})$$

El término ruido + interferencia es densidad espectral de potencia. Si el espectro de las señales es aproximadamente rectangular, con un ancho de W , entonces la densidad espectral de potencia de ruido + interferencia es

$$N_0 = F_n k_B T_0 + W^{-1} \sum_{\substack{\text{otros} \\ \text{usuarios}}} P_i \quad (\text{ecuación 4.3})$$

donde el primer término representa el nivel del ruido térmico en el receptor ($F_N =$ la figura de ruido del receptor). Volviendo a escribir la ecuación de SNR en términos de la velocidad de los datos y el ancho de banda del espectro esparcido se muestra donde está la ventaja:

$$\left[E_b / (N_0 + I_0) \right]_j = \frac{P_j / R}{N_0 + W^{-1} \sum_i P_i} \quad (\text{ecuación 4.4})$$

La interferencia en esta ecuación es la suma de las señales de todos los usuarios que no son de interés.

Esta ecuación es la clave para entender por qué CDMA no se exploró para el uso en sistemas de múltiple acceso terrestres. También es la clave de la innovación que condujo a un sistema CDMA comercial.

4.3.1: El Problema cerca-lejos

CDMA (y espectro esparcido en general) siempre se desechó por inviable en el ambiente de la radiotelefonía móvil debido a lo que se llamó el problema "cerca-lejos". Siempre se asumió que todas las estaciones emitieran potencia constante. En un sistema de telefonía móvil algunos usuarios pueden localizarse cerca de la estación base, y otros pueden localizarse lejos. La diferencia de pérdidas de propagación entre esos usuarios extremos puede ser varias decenas de dB. Suponga, por ejemplo que sólo dos usuarios están presentes, y que los dos están transmitiendo con suficiente potencia para que el ruido térmico sea despreciable. Entonces la SNR, en dB, es

$$\left[\frac{E_b}{N_0 + I_0} \right]_j = \frac{P_j / R}{N_0 + W^{-1} \sum_i P_i} \quad (\text{ecuación 4.5})$$

Si hay, digamos, 30 dB de diferencia entre las pérdidas del camino más largo y el más corto, hay entonces 60 dB de diferencia entre la SNR del usuario más cercano y el usuario más lejano, porque éstas son las potencias recibidas. Para ajustarse a los usuarios más lejanos, los anchos de banda esparcidos tendrían que ser quizás 40 dB, o 10,000 veces la frecuencia de los datos. Si la velocidad de los datos fuera 10,000 b/s, entonces $W=100\text{MHz}$. La eficiencia espectral es pésima, peor incluso que el más ineficaz sistema TDMA o FDMA. Recíprocamente, si un ancho de banda más razonable es escogido, entonces los usuarios apartados no reciben servicio.

Esta observación fue, durante años, la razón para no intentar ningún sistema de espectro esparcido, salvo en los satélites geosíncronicos, donde las pérdidas en el camino son relativamente pequeñas.

4.3.2: Control de Potencia

La clave para la alta capacidad de CDMA comercial es sumamente simple: Si, en lugar de usar potencia constante, los transmisores pueden controlarse de semejante manera que las potencias recibidas por todos los usuarios sean aproximadamente iguales, entonces se comprenden los beneficios del sistema. Si la potencia recibida se controla, entonces los subscriptores pueden ocupar el mismo espectro, y el esperado beneficio de promediar la interferencia aumenta.

Asumiendo un perfecto control de potencia, el ruido más interferencia es ahora como puede verse en la ecuación 4.6.

$$N_0 + I_0 = N_0 + (N - 1) P_s$$

$$N_0 = F_n k_B T_0$$

(ecuación 4.6)

donde N es el número total de usuarios. La SNR se vuelve

$$\frac{E_b}{N_0 + I_0} = \frac{\frac{P_s}{R}}{N_0 + (N - 1) \frac{P_s}{W}} = \frac{\frac{W}{R}}{\left(\frac{N_0 W}{P_s}\right) + N - 1}$$

(ecuación 4.7)

La capacidad máxima se logra si ajustamos el control de potencia para que la SNR sea exactamente la que necesitamos para una probabilidad de error aceptable. Si fijamos el lado izquierdo de la ecuación 4.7 para una dada SNR en destino y resolvemos para N, encontramos la ecuación de capacidad básica para CDMA:

$$N - 1 = \frac{\frac{W}{R}}{\left[\frac{E_b}{N_0 + I_0}\right]_{destino}} - \frac{N_0}{P_s} \xrightarrow{P_s \rightarrow \infty} \frac{\frac{W}{R}}{\left[\frac{E_b}{N_0 + I_0}\right]_{destino}}$$

(ecuación 4.8)

Usando los números para IS-95A CDMA con la velocidad de 9.6 kbps fija, encontramos

$$K \approx \left(\frac{W}{R}\right)_{dB} - \left(\frac{E_b}{N_0}\right)_{destino, dB} \approx 21.1dB - 6dB = 15.1dB$$

(ecuación 4.9)

o aproximadamente N=32. La SNR designada de 6 dB es una estimación nominal. Una vez que el control de potencia está disponible, el diseñador del sistema y el operador tienen la libertad para cambiar calidad de servicio por capacidad ajustando la SNR en destino. Note que la capacidad y la SNR son recíprocos: una mejora de tres dB en SNR incurre en una pérdida de capacidad en un factor de dos, y viceversa.

Hemos descuidado la diferencia entre N y N-1 en la ecuación 4.9. Esto es conveniente en la matemática, y es normalmente razonable porque la capacidad es lo suficientemente grande.

Hay factores que no hemos tenido en cuenta todavía. Algunas de las cosas que no hemos considerado todavía de hecho ayudan; otras perjudican. Pero en balance, hay una mejora por sobre las tecnologías de banda angosta.

La capacidad sustentable es proporcional al proceso de ganancia, reducida por la SNR requerida. Mientras hay varias consideraciones que tenemos que ver a todavía, hay ya una sugerencia de mejora de capacidad posible. Con E_b/N_0 en el rango de 3-9 dB, la ecuación 4.9 da una capacidad en el intervalo de 16-64 usuarios. En el mismo ancho de banda, un único sector de una sola célula de AMPS tiene sólo 2 canales disponibles.

4.3.3: Capacidad de la Célula

La discusión que lleva a la ecuación 4.9 asume sólo una célula, sin la interferencia de las células vecinas. Uno podría preguntar lo que se ha ganado aquí. La capacidad de una célula de AMPS aislada es igualmente muy alta. De hecho, nada lo detiene a usar todos los canales si no hay ningún vecino; el reuso no se necesita. La capacidad de esa célula de AMPS totalmente usada estaría aproximadamente en 42 canales (1.25 MHz / 30 kHz de espacio por canal). Esto no es muy diferente que el número que nosotros recién calculamos para CDMA.

4.3.4: La ventaja de usar CDMA

Aquí es donde está la gran diferencia. Para encontrar lo que pasa con la interferencia de las células vecinas, tenemos que agregar esa interferencia en la ecuación 4.3 anterior. Pueden encontrarse las matemáticas de esto en algunas de las referencias [por ej. Gilhousen]. Resulta que la fracción de la interferencia del canal inverso que viene de las células vecinas es de alrededor del 60% de la interferencia de la propia célula. Y esta respuesta no es muy sensible a los parámetros del modelo, con tal de que asumamos que los móviles se controlan en potencia de una manera sensata.

Para realizar el cálculo, sólo introduzca el factor de reuso de frecuencia efectivo F , definido como,

$$F = \frac{\textit{Interferencia total}}{\textit{Interferencia de la propia célula}}$$

(ecuación 4.10)

Puede llevarse el efecto de la interferencia adicional a través de la matemática simplemente reemplazando N dondequiera que aparezca por $F \cdot N$. La capacidad, asumiendo que todas las células están igualmente cargadas, es ahora

$$N_{\text{polo}} = \frac{W/R}{\left[\frac{E_b}{(N_0 + I_0)} \right]_{\text{destino}} \bullet F}$$

(ecuación 4.11)

De la ecuación 4.11, es evidente por qué llamamos a F factor de reuso de frecuencia efectivo. Cumple el mismo papel en la ecuación de capacidad de CDMA (4.11) que el factor de reuso de frecuencia K hace en capacidad de los sistemas AMPS. Una célula de CDMA, incluso con la interferencia de las otras células, tiene una capacidad más alta que una célula de AMPS por un factor de $K/F = 7/1.6$ o aproximadamente 4.4. La mejora en una célula sectorizada es aun mas notable, porque la capacidad de CDMA no es muy alterada por la sectorización. Es decir, la ecuación 4.11 se aplica a un sector, con modificaciones sólo pequeñas debido a la filtración de la interferencia entre los sectores, mientras que en AMPS no se gana con la sectorización. En este caso la ganancia es $K/F = 21/1.6$ o aproximadamente 13 veces.

4.3.5: Codificación de voz

Debido a que la interferencia se promedia, cualquier cosa que pueda hacerse para reducir la potencia promedio transmitida mejorará la capacidad. Un objetivo obvio para tal optimización de potencia es la codificación de la voz.

La conversación humana es una fuente de información intermitente. Mediciones realizadas en Laboratorios Bell hace muchos años sugieren que el factor de actividad en una conversación humana está en el rango de 35-40%. Si ese factor de actividad puede traducirse en una reducción de potencia transmitida, entonces es posible un aumento de capacidad de quizás dos veces o más. Esto es de hecho cumplido por las características de los estándares de la interface aéreas y los vocoder utilizados. La capacidad se vuelve

$$N_{\text{polo}} = \frac{W/R}{\left[\frac{E_b}{(N_0 + I_0)} \right]_{\text{destino}} \bullet F \bullet v}$$

(ecuación 4.12)

donde v es el factor de actividad de voz, aproximadamente 0.5. La reducción lograda en la práctica está por debajo del factor de actividad medido porque la potencia de la transmisión en la interface aérea no se reduce a cero durante los periodos ociosos. La ganancia de capacidad por encima de los sistemas AMPS, por las mismas suposiciones anteriores, es ahora de aproximadamente 26 veces, quizás optimista dado la naturaleza cruda del modelo, pero sugestivo de las mejoras sustanciales que son posibles convirtiendo a CDMA.

Los datos de velocidad variable se acomodan a una velocidad de tráfico de datos básica que puede ser reducida en proporciones binarias (1, 1/2, 1/4, y 1/8). La transmisión nunca se reduce a cero debido a que esto presentaría problemas relacionados con la recepción del canal. Actualmente existen dos familias de velocidades posibles : la primera basada en vocoders de 9600 bps, la segunda en 14,400 bps. Éstas, por supuesto, tienen características de capacidad diferentes.

Los vocoders usados en CDMA están estandarizados separadamente de la interface aérea como una opción de servicio. El primer vocoder estandarizado fue el IS-96. La norma IS-96 opera en el juego de velocidades de 9600 bps nominal, y logra un

promedio de velocidad eficaz de datos de algo más de 4kbps, que es aproximadamente el 50% de la velocidad real de la interface aérea (de 8550 bps).

4.3.6: Propagación en múltiples caminos (Multipath)

La capacidad del sistema, tal como podría esperarse, es afectada por los fenómenos de la propagación. Los usuarios de teléfonos celulares analógicos están familiarizados con el desvanecimiento que es tan molesto, sobre todo al realizar movimientos cortos con el móvil (como pararse o cambiar de posición) que obligan a "buscar" el lugar donde la señal sea más intensa. El desvanecimiento (fading) en un vehículo en movimiento es más rápido, siendo causado por el movimiento del vehículo a través de los patrones de interferencia estacionarios, donde la escala espacial del patrón de interferencia es la longitud de onda, aproximadamente un pie. CDMA es mucho más robusto que las tecnologías analógicas en la presencia de multipath, pero afecta su capacidad.

Hay dos preguntas que uno debe hacerse con respecto al desvanecimiento por multipath y CDMA. Primero, ¿bajo qué circunstancias se experimentará en CDMA el desvanecimiento?, Y segundo, ¿cuál es el efecto del desvanecimiento, cuando ocurre, en un canal de CDMA?

¿Cuándo causa fading el multipath?

Cuando los diferentes componentes debidos al multipath son "resueltos" por la forma de onda de CDMA, es decir, cuando sus retrasos están separados por al menos el tiempo de decorrelacion de la forma de onda esparcida en el espectro, entonces pueden ser separados por el receptor. Ellos no interfieren porque cada componente se correlaciona con una demora diferente. Cuando los componentes están separados por menos del tiempo de decorrelacion, entonces no pueden separarse en el receptor, e interfieren entre si, llevando a lo que a veces se llama "flat fading".

El fading también se describe como de tipo Rayleigh o Rician. Rayleigh fading es el resultado de una suma vectorial de componentes de señal múltiples, cada uno con una amplitud al azar. Puede verse alternativamente como una señal cuyas componentes I y Q son Gaussianas. Rayleigh fading muestra profundas caídas de señal.

Si hay una componente fuerte, constante de señal, además de los múltiples componentes aleatorios de Rayleigh fading, entonces se dice que el fading es Rician. Rician fading es típico de situaciones donde hay un camino directo, sin obstrucciones, entre las estaciones, así como también superficies que reflejan y esparcen la señal.

La duración de un "chip" esparcido es $1/1.2288\text{MHz} = 814 \text{ ns}$, que equivale a 244 metros. Diferencias de camino menores de esto llevarán a "flat fading"; y las mayores llevarán a multipath resuelto que será diversidad combinada por el receptor.

Con respecto a la segunda pregunta, los efectos del fading, la respuesta es compleja y es diferente en los vínculos "directo" (de la celda al móvil) e "inverso" (del móvil a la celda). También depende de la velocidad de fading, que a su vez depende de la velocidad a la que viaja la estación móvil. Generalmente el desvanecimiento incrementa la SNR necesaria para una probabilidad de error dada. El incremento puede ser tanto como 6 dB. En el vínculo inverso, el control de potencia mitigará los efectos de fading a velocidad baja; y a velocidades altas tiene poco efecto. A velocidades altas, y en ambos

vínculos, la codificación y el entrelazado se tornan más eficaces cuando el tiempo de desvanecimiento es menor que el lapso del interleaver (1 frame ó 20 mseg.).

4.3.7: Problemas de cobertura

Hay algunas malas noticias que surgen de la ecuación de capacidad de CDMA. A saber, el hecho que la potencia que los móviles necesitan transmitir tiende al infinito cuando la capacidad aumenta. Cuando la potencia requerida aumenta, los móviles al borde de la zona de cobertura empezarán a funcionar sin la suficiente potencia de transmisión. Es decir, les pedirán que transmitan más de lo que su capacidad les permite. La consecuencia práctica de esto es que la carga del sistema realmente debe controlarse para que el área de servicio planeada nunca experimente fallos de cobertura debido a este fenómeno.

Verdaderamente no es un problema porque es una consideración de diseño del sistema. No se puede lograr capacidad máxima y cobertura máxima simultáneamente, porque hay un compromiso entre ambas.

La situación inversa también es posible, es decir, una escasa carga en la célula puede provocar que la misma pueda “crecer” en tamaño, prestando servicio a una zona que no le correspondería según el diseño del sistema. Esto provoca que los límites entre células no sean estrictos, y que un móvil que se encuentra alejado de la estación base pueda ser atendido por la celda vecina si la original se encuentra sobrecargada y la vecina no.

4.3.8: Compromiso Cobertura-capacidad en el Vínculo Inverso

Los modelos simples de capacidad del vínculo inverso muestran que la potencia de RF se eleva con la carga. Hay un límite teórico, o polo de capacidad, logrado cuando todos los usuarios están transmitiendo potencia infinita.

Los sistemas reales deben operar debajo del polo de capacidad porque las estaciones de los usuarios reales tienen un límite superior para la potencia del transmisor. Cuando los usuarios marginales están transmitiendo a potencia máxima, no debe permitirse aumentar la carga. Recíprocamente, si se permite aumentar la carga, entonces las estaciones de los usuarios marginales ya no pueden cerrar sus lazos de control de potencia. Es decir, sus SNRs serán menores a las necesarias. En efecto, la célula se encoge debido a la carga.

Este fenómeno conecta cobertura y carga. El polo de capacidad de una célula sólo depende de la SNR promedio, la ganancia del proceso, y el factor de actividad de voz. El área de cobertura de una célula, es decir, el área en la que todos los usuarios obtienen el SNR designado, depende de la carga relativa al polo de capacidad. Esta relación entre la carga y el área de cobertura es un factor en la elección de las ubicaciones para las células. A uno le gustaría usar el número mínimo de sitios de células para un costo mínimo. Pocas células dispersas pueden ser apropiadas para un nuevo sistema, o una localidad con baja densidad de población. Sin embargo cuando la densidad de carga aumenta, la densidad de las células debe aumentarse. El espaciamiento entre células debe ser suficiente para dar servicio a todos los usuarios, dentro de los límites de diseño del transmisor. Y en realidad, la carga es una variable aleatoria, medida en densidad de Erlang. La calidad de servicio sólo puede medirse en un sentido estadístico. El diseño

del sistema es, así como en sistemas de teléfono fijos, un compromiso entre los requisitos de bajo costo y buena calidad de servicio.

El análisis detallado de la interacción de cobertura y capacidad en Erlang[#] es una cuestión compleja, que involucra diversas características como el control de potencia, el handoff, el desvanecimiento, el comportamiento de los subscriptores, y otros factores, así como las diferencias entre el vínculo directo y el inverso. Pero a un nivel elemental, el vínculo inverso es fácil de analizar, y un modelo simple dilucida algunos fenómenos básicos.

Es conveniente volver a escribir la ecuación de capacidad en términos de un parámetro de carga adimensional. Empezando por la ecuación de capacidad del vínculo inverso.

$$E_b / (N_0 + I_0) = \frac{W/R}{\left(N_0 \frac{W}{P_s} \right) + F v (N-1)}$$

(ecuación 4.13)

donde

$$E_b / (N_0 + I_0) = \text{SNR}_{\text{de funcionamiento}}$$

W = Ancho de banda

R = Velocidad de los datos = 9.6 kbps ó 14.4 kbps

W/R = ganancia del proceso

= 21.1 dB para vocoders de 9.6 kbps

= 19.3 dB para vocoders de 14.4 kbps

$F = \frac{\text{Interferencia total}}{\text{Interferencia de la propia célula}} \approx 1.6$

v = factor de actividad de voz ≈ 0.5

N = número de usuarios activos en el sector

[#] El Erlang es una medida estadística de carga, indica la cantidad de llamadas en un momento dado. Dependerá de la cantidad de suscriptores en la zona, la cantidad de llamadas que realicen (que a su vez dependen del día y la hora) y su duración. El valor de carga en Erlangs y la probabilidad de bloqueo (ocupado) en porcentaje de las llamadas, están relacionadas con la cantidad de canales de RF disponibles. Esto se utiliza durante el diseño del sistema.

El polo de capacidad es la carga que es solución a esta ecuación cuando la potencia de los móviles de los usuarios se aproxima a infinito.

$$N_{\text{polo}} = \frac{W/R}{\left[E_b / (N_0 + I_0) \right]_{\text{destino}} \bullet F \bullet v} \quad (\text{ecuación 4.14})$$

El polo de capacidad para $E_b/N_0 = 6$ dB y el juego de velocidades de 9.6 kbps, está alrededor de $N_{\text{polo}} = 40$.

La relación entre la carga y la potencia toma una forma particularmente simple si nosotros hacemos las siguientes substituciones:

potencia adimensional

$$Z = \frac{N_0 W + N P_s F v}{N_0 W} \quad (\text{ecuación 4.15})$$

carga adimensional

$$L = \frac{N}{N_{\text{polo}}} \quad (\text{ecuación 4.16})$$

La ecuación de capacidad, en términos de estos parámetros adimensionales, se vuelve

$$Z = \frac{1}{Z - L} \quad (\text{ecuación 4.17})$$

Hay dos observaciones importantes de estas ecuaciones:

1. El polo de capacidad sólo depende del E_b/N_0 en destino, la ganancia del proceso, el factor de actividad de voz y el reuso de frecuencia eficaz F . No depende de la figura de ruido del receptor.
2. La figura de ruido del receptor fija su sensibilidad, y la escala espacial del sistema. No afecta la capacidad.

Este análisis no tuvo en cuenta los efectos de un control de potencia imperfecto y la dinámica del sistema. No sería prudente hacer funcionar un sistema demasiado cerca del

polo de capacidad porque las fluctuaciones de carga lo acercarían a éste valor. Las excursiones de carga afectarán adversamente el servicio dado a los subscriptores marginales, posiblemente causando fracasos intermitentes para cerrar sus lazos de control de potencia y una pérdida de Frames excesiva. Cargas de 50% a 75% ($Z = 3$ a 6 dB) son un compromiso apropiado entre carga y cobertura.

4.3.9: Capacidad real

Debido a lo visto anteriormente, la carga debe limitarse para no afectar la cobertura del sistema, por lo que el valor calculado en la ecuación 4.12 no puede alcanzarse.

Además, debe tenerse en cuenta que no todos los canales disponibles se utilizan como canales de tráfico, sino que también son necesarios los canales piloto, de sincronismo y de paging (en el vínculo directo) y los de acceso (en el vínculo inverso).

Durante el Soft Handoff, dos o más celdas atienden simultáneamente una misma llamada, por lo que también debe considerarse su efecto en la capacidad del sistema.

Todo esto lleva a que en la práctica solo estén disponibles alrededor de 20 canales por sector.

Para compararlo con otras tecnologías, se debe tener en cuenta la cantidad de canales en 7 células, con tres sectores cada una, lo que daría un total de 420. Como 1.25 MHz equivale aproximadamente a 42 canales de 30KHz, se puede deducir que CDMA brinda un incremento de capacidad de alrededor de 10 veces con respecto a AMPS.

4.4: Ancho de Banda Elegido

El ancho de banda de 1.25 MHz es un compromiso entre la mitigación por multipath, la complejidad del receptor, y el desempeño del receptor. Anchos de banda más grandes requerirían más correladores Rake para utilizar un mayor fragmento de la energía dispersa por multipath, mientras que velocidades más pequeñas padecerían con mayor probabilidad los efectos de "Flat fading" y la llamada "ley de números grandes".

Es decir, debido a que habría pocos usuarios en cada canal de CDMA, las fluctuaciones de potencia relativas que son proporcionales a $N^{-1/2}$, $N =$ número de usuarios, serían más grandes. Un mayor margen de funcionamiento sería necesario para un funcionamiento estable, trayendo consigo una pérdida en capacidad por MHz.

En el ambiente, la señal se propaga rebotando en objetos tales como edificios, vehículos y accidentes del terreno. Múltiples réplicas de la señal arriban al receptor después de viajar por diferentes caminos, como puede verse en la figura 4.11. Las réplicas llegan con diferentes atenuaciones y retardos.

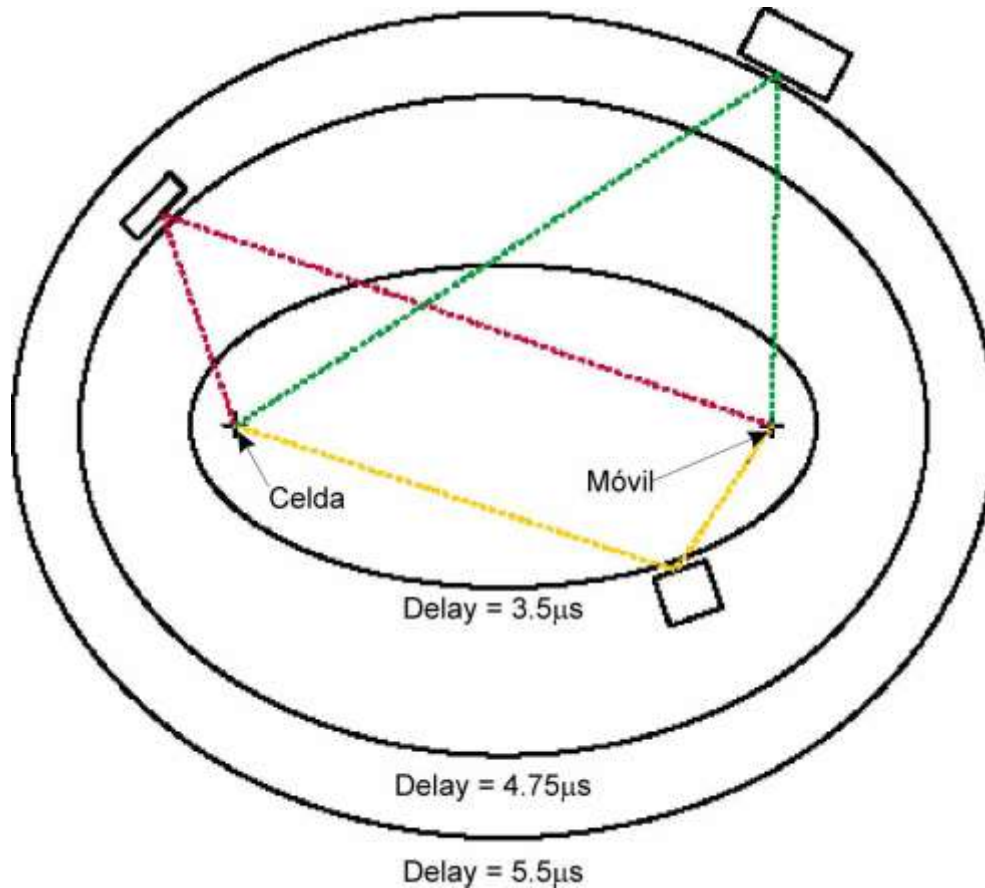


Figura 4.11. Multipath y retardos en el canal.

Las elipses mostrados en la figura son los contornos donde las reflexiones producen un retardo de propagación constante entre la celda y el móvil.

La respuesta al impulso del canal tiene la forma que puede apreciarse en la figura 4.12.

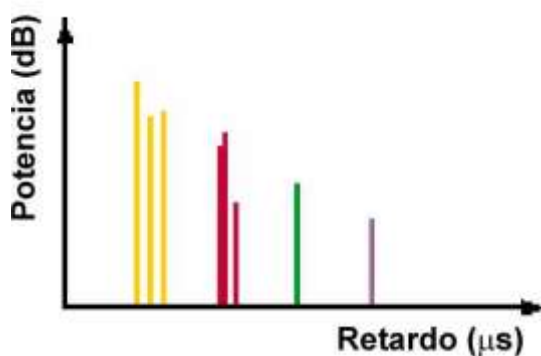


Figura 4.12. Respuesta al impulso del canal

La mayoría de los ambientes muestran una respuesta al impulso con una dispersión temporal total (retardo de tiempo entre la primera y la última réplica de la señal que llegan al receptor) de 10 microsegundos o menos, aunque en lugares con muchos

obstáculos (como puede ser la zona céntrica de una gran ciudad), pueden verse dispersiones de 20 microsegundos o más.

Cada línea de la figura corresponde a un único camino, y no tiene el efecto de desvanecimiento. Pero sólo contiene una pequeña fracción de la potencia total de la señal. Para una señal de banda angosta, las componentes se suman incoherentemente, provocando un único fásor estacionario que cambia cuando el móvil se mueve, produciendo el desvanecimiento.

Una señal tiene una resolución temporal asociada con su ancho de banda:

$$S \approx 1/W$$

(ecuación 4.18)

Un receptor de CDMA forma una ventana temporal tal que las componentes que arriban dentro de la ventana son aceptadas (y sumadas incoherentemente) y las demás son rechazadas. Algunas ventanas temporales se muestran en la figura 4.13.

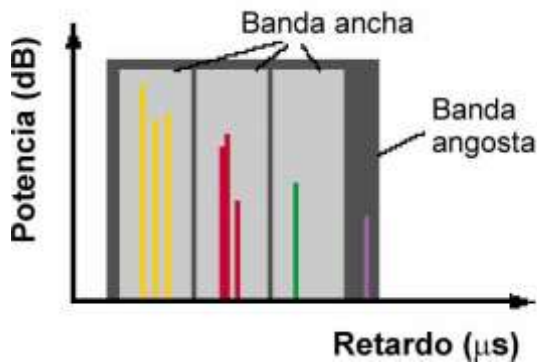


Figura 4.13. Efecto del ancho de banda

Un receptor con un único correlador, aceptará sólo una fracción de la potencia que llega. Si se disponen varios correladores, cada uno con un distinto offset de tiempo (como las 3 ventanas que se ven en la figura anterior), entonces puede aprovecharse una mayor parte de la energía de la señal. Además, las salidas de los correladores pueden sumarse coherentemente, anulando el efecto de desvanecimiento que se ve en señales de banda angosta

El retraso de propagación de Multipath en áreas urbanas está típicamente en el rango de 2-3 microsegundos. Anchos de banda significativamente más grandes que 1 MHz podrían resolver muchos más componentes del multipath. Mientras que habría menos probabilidad de desvanecimiento de cada salida de los correladores, tendría que haber más de ellos. La fracción de la energía en cada uno sería más pequeña, haciendo que el rastreo en tiempo y frecuencia sea más difícil. También la función de búsqueda tendría que encontrar componentes de la señal más pequeños, que requerirían más tiempo de integración y por consiguiente afectarían el desempeño.

Los anchos de banda más pequeños padecen dos problemas. Primero, ellos tenderían a no resolver situaciones comunes de multipath. En este caso, el desvanecimiento sería

muy similar a los sistemas de banda angosta, exhibiendo el llamado desvanecimiento llano (flat fading) de Rayleigh. Y el desempeño global sería afectado adversamente por un desvanecimiento más severo.

Una segunda consecuencia de adoptar un ancho de banda demasiado estrecho es que cada portadora daría servicio a menos usuarios. Si hay menos usuarios por portadora, la ley de grandes números afectaría las fluctuaciones de potencia. Las fluctuaciones en la suma de N variables aleatorias, como la potencia de cada usuario, son proporcionales a $N^{-1/2}$. Con fluctuaciones más grandes, el margen de funcionamiento del sistema se debe aumentar. La capacidad por MHz sería más pobre porque cada canal de CDMA estaría manejando una carga más pequeña que aquella posible con un ancho de banda mas grande.

Una frecuencia esparcida de aproximadamente 2-3 veces el retraso de propagación típico es suficiente para que 2 o 3 correladores en un receptor Rake puedan recoger normalmente casi toda la energía de la señal. Además, puede esperarse alguna independencia estadística entre las salidas de los correladores.

Otra consideración es la introducción de CDMA en la infraestructura de AMPS existente. 1.25 MHz es 1/8 de las originales asignaciones de 10 MHz. El ancho de banda relativamente pequeño le permite a un operador reemplazar aproximadamente el 15% de sus canales de AMPS por un canal de CDMA. Pero ese canal de CDMA tiene muchas veces la capacidad de los canales de AMPS que reemplazó. Esto permite una gradual introducción de CDMA en el sistema.

Y finalmente, teniendo en cuenta las asignaciones de frecuencia para telefonía celular en América del Norte, en la cual uno de los segmentos de la banda sólo tiene 1.5 MHz de ancho, la opción de 1.25 MHz se ve como un compromiso conveniente entre todas estas consideraciones en conflicto.

4.5: Control de Potencia

4.5.1: El Ambiente de Propagación en Telefonía Móvil

Un móvil celular puede estar en cualquier parte de la célula, ya sea justo debajo de la torre de antena como quizás a 30 km de distancia. En un ambiente donde la ley de la propagación es aproximadamente $1/R^4$, como es típico de áreas de servicio celulares, el rango dinámico total de las pérdidas en el camino está en el orden de 80 dB. En un vínculo típico para un sistema IS-95A, esto significa que el transmisor móvil debe variar su potencia de aproximadamente 2.5 nW a 0.25 W.

Además de la pérdida grande por la distancia, la pérdida puede variar también rápidamente debido al desvanecimiento de Rayleigh producido por el multipath. El desvanecimiento de Multipath ocurre cuando la señal llega a la antena receptora después de cruzar rutas diferentes. Si la diferencia de longitud de camino es pequeña comparada a un chip, o aproximadamente 244 m, las señales interfieren entre sí, resultando un desvanecimiento. El desvanecimiento normalmente causa fluctuaciones de 20 a 30 dB en una distancia de una o dos longitudes de onda, 30 a 60 cm. Si el móvil está viajando a 30 m/s, una velocidad típica en una autopista, entonces la frecuencia puede exceder los 100 Hz. Desvanecimientos de esta profundidad, aunque comunes en sistemas de

banda angosta, son menos probables en sistemas de CDMA debido a su habilidad de separar los componentes del multipath en los receptores Rake.

Todavía otra consideración es que los procesos de desvanecimiento para transmisión y recepción, aun cuando tienen correlación, no son necesariamente idénticos debido a los 45 MHz (celular) o 80 MHz (PCS) de diferencia entre frecuencias de los canales directo e inverso.

4.5.2: Control de Potencia en el Vínculo inverso

Control de lazo abierto

El amplio rango dinámico es manejado por una técnica de control de potencia de lazo abierto. El móvil estima la pérdida del camino a la célula midiendo el nivel de la señal recibida. Típicamente esta medida se basa en un AGC (control de ganancia automático) analógico de voltaje. El AGC intenta mantener constante el nivel de la señal de su conversor analógico-a-digital, es decir, la potencia total que entra en sus 1.25 MHz. Esta potencia incluye todo lo que ingresa en el receptor: señal, ruido térmico e interferencia. La potencia medida es ajustada por una corrección de lazo cerrado y entonces controla la potencia transmitida por el móvil de acuerdo con la relación, prescrita en la interface aérea celular:

$$R_x \text{ (dB)} + T_x \text{ (dB)} = -73 \text{ dB (mW)}^2 \quad (\text{ecuación 4.19})$$

Note que ésta es una relación recíproca. Cuando la potencia recibida aumenta, la potencia transmitida disminuye. Su producto, con dimensiones de mWatts^2 , es llamado constante de repunte. En unidades de dB relativo a $(1 \text{ mW})^2$, la constante del repunte es -73. La constante de repunte especificada para PCS, a aproximadamente dos veces la frecuencia, es -76 dB (mW)^2 .

Así, cuando la potencia recibida es, por ejemplo, -90 dBm , tal como podría ser en una célula normal, la potencia del transmisor móvil será $+17 \text{ dBm}$. Es quizás curioso que el móvil, cuando está cerca de una celda, realmente estará transmitiendo menos potencia que la que está recibiendo.

Los ajustes ordenados se transmiten al móvil en mensajes superpuestos en la transmisión. Se usan para ajustar el control de potencia de lazo abierto para las diferentes células y sensibilidades del receptor.

Uno podría cuestionar la decisión de usar la potencia total que se recibe para el control de lazo abierto. Sin embargo, esto se hizo para obtener un tiempo de respuesta más rápido. Es de hecho verdad que la estimación de lazo abierto está a menudo equivocada en varios dB, pero esto es compensado fácilmente por la corrección de lazo cerrado. El problema no podría ser resuelto eliminando el lazo cerrado y haciendo algo más sofisticado el control de lazo abierto, porque la corrección del lazo cerrado todavía se necesitaría para manejar la pérdida debido a la diferencia de frecuencia.

Control de lazo cerrado

El control de lazo cerrado es una especie de "sintonía fina" del lazo abierto. La celda mide la relación E_b/N_0 recibida y la compara a un valor fijo (que puede ser ajustado dinámicamente). Si el valor de E_b/N_0 está por encima del valor fijo, entonces se envía una orden de "bajar"; si está por debajo, se envía una el orden de "subir" al móvil. El móvil ajusta su potencia aumentando o bajando su potencia, relativa a la estimación de lazo abierto, en aproximadamente un dB por cada orden. No hay ninguna orden de "no haga nada" para ajustar las órdenes a un solo bit. Un continuo "no haga nada" tiene que ser transmitido como alternativos "subir" y "bajar". Las órdenes se envían una vez cada 1.25 ms, a una frecuencia de 800 correcciones por segundo.

Éste es un mecanismo de control muy rápido (800 dB por segundo), y ha demostrado trabajar muy bien en la práctica. Las mediciones de errores de E_b/N_0 en un sistema de gran envergadura mostraron que los errores eran aproximadamente normales, con una desviación estandar de aproximadamente 2 dB. Uno esperaría errores distribuídos normalmente debido al hecho que los ajustes son deltas constantes en dB.

Para desvanecimientos más rápidos que el lazo cerrado, son muy eficaces la codificación y el entrelazado aplicados a la voz. A bajas frecuencias el entrelazado puede ser menos eficaz, pero entonces el control de potencia es sumamente robusto.

Los bits de control de potencia se envían sin codificar y se insertan en la transmisión del canal de tráfico directo, en lugar de dos símbolos.

La decisión para enviar los bits de control sin codificar es, como el lazo abierto, para lograr una velocidad de respuesta mas rápida. Cualquier codificación de estos bits producirían un retraso del proceso y, por lo tanto, un lazo más lento.

El rango dinámico del control de lazo cerrado es 24 dB relativo al de lazo abierto.

Soft Handoff

Durante el soft handoff es crucial que el móvil transmita potencia controlado por la célula que está recibiendo la mejor señal, para que se transmita la mínima potencia necesaria. Éste es un requisito importante para lograr el máximo de capacidad en el sistema. Por esta razón, cada célula y sector que participan en un handoff hace una determinación distinta del bit de control de potencia que se envía. El móvil los procesa separadamente, y realiza una operación lógica "or" entre ellos para determinar si debe "bajar". Es decir, si cualquiera de los participantes del handoff (puede haber más de dos) dice "abajo" el móvil reduce su potencia.

Una variación especial se permite en esta regla. En algunas circunstancias los sectores participantes pueden ver las decisiones de los otros sectores y tomar una decisión en conjunto, que entonces transmiten los dos. Éste normalmente es el caso cuando el handoff es entre los sectores de una misma célula. El mismo equipo manejando ambas ramas del handoff. En este caso, el móvil está informado que los bits de control son idénticos. Puede conseguir diversidad combinando entonces los bits, en lugar de bits separados y una lógica "or".

4.5.3: Control de Potencia en el Vínculo Directo

Los requisitos en el vínculo directo son menos severos. Mientras las pérdidas en el camino sufren las mismas variaciones debido a desvanecimientos y sombras que en el

canal inverso, mientras que el nivel de la señal total sea adecuado, debido a que la interferencia (los usuarios de la misma célula) viene de la misma fuente, la señal deseada y la interferencia tienden a desaparecer juntas.

Esto no es verdad para la interferencia debido a otra célula, sin embargo, cuando exista el potencial para una interferencia dañina, habrá comenzado un handoff, así que ya existirá un vínculo alternativo.

Existen dos técnicas para el control de potencia. IS-95A (celular) y el Juego 1 de velocidades de J-STD-008 (PCS) especifican sólo un control basado en el mensaje. Es decir, cuando la estación móvil determina, debido a un porcentaje de frames erróneos excesivo, que la calidad de señal es pobre, envía un informe a la celda. Este método es relativamente lento, debido al retraso del proceso. El juego de velocidades 2 (14,400 bps), incorpora un mecanismo de control de potencia más rápido. Cada frame de tráfico inverso incorpora un bit con el que informa a la celda, con un insignificante (2 frames, es decir 40 msec.) retardo en el proceso.

4.6: Handoff

El acto de transferir el vínculo de RF con un móvil de una celda a otra es llamado handoff (o handover). La implementación del handoff es sumamente diferente entre las normas de banda angosta y las de CDMA.

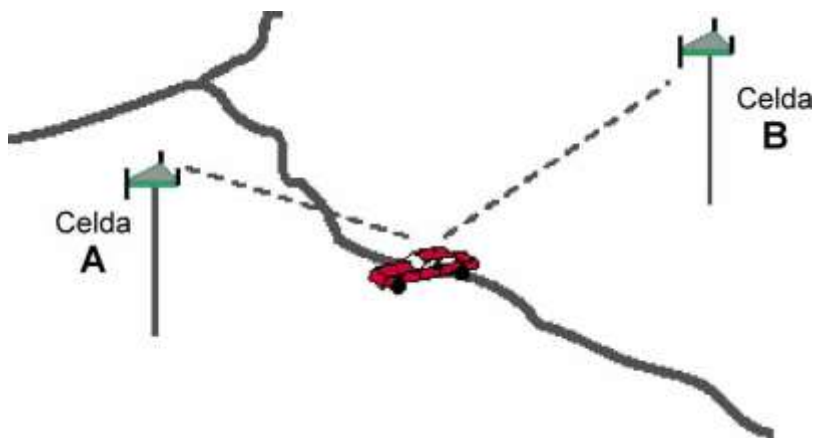


Fig. 4.14: Límite de las coberturas de cada celda.

Los handoffs de AMPS frecuentemente fallan, causando la caída de la llamada. Esto contribuye a la percepción de una pobre calidad de servicio. Es más, cada handoff es precedido y seguido por largos intervalos de una pobre calidad del vínculo, produciendo un molesto ruido y distorsión. Por comparación, CDMA se diseña específicamente no sólo para reducir fracasos del handoff sino también para proporcionar un servicio sin interrupciones. CDMA mantiene una buena calidad de voz en todo momento: antes de, durante, y después del handoff. Los handoffs son imperceptibles, incluso para oyentes experimentados.

4.6.1: Frecuencia de Handoff

¿Que tan problemático es esto? Después de todo, si los handoffs son raros, hacerlos pobremente es una consideración poco importante.

El número de células transitadas durante una llamada puede estimarse crudamente con las suposiciones siguientes:

- Las células son redondas con un radio uniforme R.
- La trayectoria de la estación móvil es una línea recta.
- Un móvil entra en cada célula por una ubicación aleatoria en la periferia, con una dirección aleatoria dentro de 90 grados la radial.

Con estas suposiciones el camino promedio dentro de una célula se demuestra fácilmente que es casi $\pi R/2$. El número de células transitadas durante una llamada de duración T, por un vehículo con velocidad v se estima así crudamente a aproximadamente $0.65 \cdot v \cdot T/R$. Algunos casos representativos se muestran en la tabla para llamadas de tres minutos.

Caso	Velocidad (Km/h)	Radio de Célula (Km)	Handoffs
Vehículo en ruta rural	104	16	0.33
Vehículo en ruta urbana	104	1.6	3.25
Vehículo en calle urbana	48	1	2.4
Peatón en calle urbana	2.4	1	0.12
Peatón, en microcélula	2.4	0.1	1.20

El número de handoffs puede ser en cualquier parte de quizás uno cada 8-10 llamadas a quizás 3-4 por llamada. Aunque ésta es sólo una estimación cruda, indica que el handoff es tan frecuente que un buen desempeño es importante.

Es quizás notable que los radios de célula en CDMA pueden ser considerablemente más grandes que los radios de las células de AMPS. Esto reduce la frecuencia del handoff simplemente en base a la geometría, independiente de las otras ventajas en el desempeño que presenta.

4.6.2: Pasos en un Handoff

Independientemente de la tecnología utilizada, los siguientes pasos son parte del handoff de cualquier llamada.

1. Empieza en un estado donde sólo una célula está atendiendo la llamada en cuestión.
2. Se produce si las condiciones de vínculo de RF entre el móvil y la célula vieja está deteriorándose, y si hay un vínculo potencialmente mejor a una nueva célula candidata.

3. Se informa a la célula candidata del handoff inminente, incluyendo los parámetros necesarios para identificar el móvil y ejecuta el handoff.
4. Se comunica al móvil para empezar a ejecutar el handoff.
5. La nueva célula empieza a atender al móvil
6. El móvil comienza a usar la nueva célula
7. Se ingresa al estado de medio-handoff (sólo en CDMA)
8. El móvil interrumpe el uso de la célula vieja
9. La célula vieja deja de dar servicio al móvil
10. Acabando en un estado donde una célula, la nueva, está atendiendo la llamada en cuestión

El funcionamiento real es mucho más complejo, pero éstos son los eventos principales. Algunos pasos traen consigo actividades diferentes en AMPS que en CDMA. Por ejemplo, los handoffs de CDMA normalmente no requieren sintonizar una frecuencia; en cambio requieren cambiar el código del canal que se utiliza (en el canal Directo). Ningún ajuste, ya sea de frecuencia o de código del canal se requiere en el canal de CDMA Inverso en ningún momento.

4.6.3: Proceso de Handoff Tradicional

El Handoff en AMPS tiene varios aspectos significantes que crean problemas:

- Es "duro", significando que la comunicación se interrumpe brevemente mientras el control se transfiere. Los móviles pueden sintonizar sólo un par de canales a la vez, y se requiere una interrupción porque las células adyacentes nunca usan los mismos juegos de frecuencias.
- No hay diversidad durante el handoff porque toda la comunicación es con una celda o con la otra. En ningún momento hay una comunicación simultánea con más de una celda.
- Las celdas, no los móviles, hacen las mediciones de calidad de señal que llevan a la decisión para hacer un handoff.

El disparo del handoff en un sistema AMPS puede deberse a varias cosas:

- Nivel absoluto de la señal recibida medida por la celda actual. Esto puede ser engañoso porque la medición no distingue entre la señal deseada y la interferencia, que a veces puede ser grande. Si es así, la medición puede ser errónea y fallar el inicio del handoff.
- Diferencia de potencia de señal entre la célula actual y una célula candidata. Esto tiende a trabajar mejor que el umbral absoluto, porque tiende a quitar los efectos de las variaciones de ganancia de la antena del móvil, ubicación, y potencia de salida.

- Silencio medido, por ejemplo, por la relación señal a ruido de pos detección. Esto no parece ser muy usado en la práctica.

El mecanismo de disparo debe anticiparse a la necesidad del handoff con suficiente margen para la ejecución. No es posible hacer esto con perfecta fiabilidad basándose solo en mediciones de intensidad de señal. Si el disparador es demasiado sensible, entonces se comenzarán handoffs cuando no se necesitan. Las Móviles que transiten áreas de baja intensidad de la señal, pueden activar un handoff bien dentro de una célula. Los handoffs falsos reducirán un poco la capacidad del sistema, y aumentará la probabilidad de un fracaso por otras razones, como falta de recursos en la célula designada.

El concepto de un handoff abrupto es fundamentalmente insuficiente porque las mediciones de intensidad de señal estáticas no pueden predecir el futuro. Aun cuando uno pudiera hacer suposiciones sobre la dinámica de la estación móvil, extraer información sobre ese movimiento de la señal de comunicación es imposible. Es más, calcular la pérdida de propagación por la ubicación de la estación es un problema sumamente difícil, incluso en teoría. Hacer semejante cálculo en una fracción de segundo no es ni remotamente práctico.

La solución que generalmente ha sido adoptada por los fabricantes de la infraestructura es monitorear la intensidad de la señal (RSSI) en la célula actual. Cuando el RSSI cae debajo de un determinado umbral, entonces se piden mediciones a las células candidatas al handoff predeterminadas. Después de reunir los informes de las mediciones se inicia la comunicación para llevarlo a cabo. Incluso cuando esto funciona, es poco satisfactorio porque el umbral de RSSI inicial tiene que ser puesto tan bajo que la calidad de la voz generalmente se ha deteriorado mucho antes de que el proceso se active.

Los handoffs resultantes fallan a menudo, siendo las siguientes algunas de las causas más frecuentes:

- No había ningún canal disponible en la célula designada debido a la carga.
- La medida de RSSI en una célula candidata era errónea debido a la interferencia, conduciendo a una selección incorrecta del destino.
- La calidad del vínculo se deterioró tanto antes del handoff que la comunicación necesaria para llevarlo a cabo falla.
- El cronómetro de 5 segundos que interrumpe la llamada cuando no se recibe señal, expira antes de que el handoff se pueda ejecutar.
- El móvil realmente no estaba en un límite entre dos células sino en un agujero de cobertura dentro de una célula, así que ningún destino de handoff podría encontrarse.

Y la calidad de la voz justo antes de y justo después del handoff, incluso cuando tiene éxito, es a menudo pobre porque el límite del handoff es el punto donde hay grandes pérdidas en ambas células. Nada en el diseño del sistema puede recuperar la gran pérdida en los límites del handoff.

4.6.4: Proceso de Handoff en CDMA

El handoff de CDMA difiere de AMPS en varios aspectos importantes:

- Es "suave", significando que la comunicación no es interrumpida por el handoff. Esto a veces se llama "echa antes de la interrupción." Pero es más que eso.
- El handoff no es abrupto, sino que es un estado de la llamada prolongado durante el cual hay comunicación por medio de dos o más estaciones de base. La diversidad de comunicación de multi-canal mejora el desempeño del vínculo durante el handoff. La ganancia de diversidad compensa parcialmente la gran pérdida en el camino que es propia del límite de la célula.
- La medición de la señal que activa el handoff es realizada por las estaciones móviles, no las estaciones de base.

No hay ningún límite de handoff en CDMA sino una región de handoff. El handoff distribuido mitiga la mayoría de los defectos del handoff duro estilo AMPS. Por ejemplo, una decisión de que debe comenzarse un handoff sólo significa que otra estación de base se agrega al juego activo de estaciones de base para este móvil. Los handoff pueden ser completados ya sea por el traspaso completo del móvil a la nueva célula, o volviendo el móvil a la célula original. En cualquier caso la llamada no está nunca en riesgo debido al fracaso del vínculo.

La diversidad durante el handoff mejora el desempeño del vínculo al punto que no sólo no hay una interrupción, sino que ni siquiera es perceptible en la calidad de voz, incluso para observadores experimentados. Esto reduce también muchísimo la probabilidad de llamadas caídas debido a fracasos de comunicación que rompen la coordinación del handoff.

4.6.5: Detalles del Handoff

La buena actuación del handoff en CDMA no es solamente una sutileza; es completamente esencial. Un móvil que está vinculado a una estación de base cuando otra está más cerca en términos de pérdidas en el camino, estará transmitiendo más potencia de la que sería necesaria usando la célula correcta. El hecho que aquel móvil, y otros como él, transmitan potencia en exceso, levanta el nivel global de interferencia. Los altos niveles de interferencia globales aumentan el factor de reuso eficaz, y así reducen la capacidad. Deben reducirse los handoffs sucios, tardíos, o lentos a un número mínimo.

Por otro lado, hay un efecto perjudicial de handoff, debido a la asimetría en el diseño del control de potencia entre los vínculos directos e inversos. El control de potencia del vínculo inverso es más rápido y exacto que en el directo.

El handoff suave requiere que múltiples celdas transmitan el mismo tráfico al móvil en cuestión. Los múltiples canales activos levantan el nivel de interferencia global al móvil, como en el eslabón inverso, esto aumenta el factor de reuso de frecuencia eficaz y así reduce la capacidad.

4.6.6: Soft Handoff

El handoff suave de CDMA es un estado de la llamada en que dos o más estaciones están atendiendo al mismo móvil. Esas estaciones pueden ser distintos sectores de células separadas, o múltiples sectores de la misma célula, o cualquier combinación de éstos. El handoff entre sectores, de hecho, es muy común debido a los patrones de ganancia de antena y las extravagancias de la propagación en un área urbana.

4.6.7: Canal Directo

Cada célula participante en un handoff suave transmite el mismo flujo de tráfico al móvil, bit a bit. Ellos lo hacen así en cualquier canal disponible. Cada estación de base escoge un canal simplemente en base a disponibilidad. La estación móvil debe llevar a cabo, en su receptor Rake, múltiples búsquedas de las que son capaces "sintonizando" cualquiera de los 63 canales (códigos) disponibles. Las salidas de los buscadores Rake deben combinarse para un buen desempeño de E_b/N_0 . La presencia de un piloto en el Canal de CDMA Directo permite combinar coherentemente en forma óptima las salidas de los Rake.

4.6.8: Control de potencia en el canal inverso

Los bits de control de potencia inversos están incluidos en el Canal de CDMA Directo. Éstos ocurren en posiciones pseudo aleatorias cada 1.25 ms de intervalo, o 16 veces por Frame. Cada bit se interpreta como un orden para levantar o bajar la potencia con que se transmite.

Cada estación de base toma decisiones de control de potencia independientemente. La estación móvil es responsable de demodular los bits levantar o bajar su potencia. La meta del control de potencia es mantener el vínculo inverso transmitiendo a la mínima potencia posible para un desempeño adecuado. El móvil debe interpretar los bits que discreparán a menudo entre sí, y sólo aumentar la potencia si todas las celdas en el handoff lo requieren; si cualquier participante dice "abajo" entonces el móvil debe reducir su potencia.

4.6.9: Canal inverso

La modulación del Canal de Tráfico Inverso en CDMA es única para cada móvil. No hay nada en la codificación ni en la modulación que dependa de forma alguna de las celdas que estén atendiendo al móvil. El móvil entonces no tiene necesidad de hacer nada especial en el handoff, aparte de la interpretación apropiada de los bits de control de potencia.

La combinación de las señales del vínculo inverso recibidas por los distintos sectores (o células) no se especifica en los estándares de la interface. Sin embargo las consideraciones prácticas animan el uso de diversidad en la recepción. Es decir, cada celda demodula, reordena y descifra el tráfico independientemente. Cuando los Frames de tráfico de los participantes del handoff llegan a la interface de la red, la calidad de los Frames puede compararse y el mejor será escogido para transmitirse en la red. Esto a

veces se llama diversidad de selección. Usa la mejor de las copias disponibles de cada Frame de tráfico.

4.6.10: Softer handoff

Todos esto se modifica levemente si los participantes en un handoff suave son sectores de la misma célula. Esta situación ha llegado para ser conocida como "softer" handoff. Las celdas permiten combinarse a los sectores para tener un único receptor con visibilidad de múltiples sectores. Pueden hacerse tales combinaciones símbolo por símbolo, en lugar de por selección de Frames enteros.

Además, un único módem común a todos los sectores es capaz de enviar bits de control de potencia idénticos a todos. Esto está permitido por las interfaces aéreas. El mensaje con la instrucción de iniciar el handoff contiene un campo que indica que las estaciones están transmitiendo los mismos bits de control de potencia para que el móvil pueda combinarlos.

Para el móvil, un "softer" handoff es idéntico al "soft" handoff salvo el tratamiento de los bits de control de potencia.

4.6.11: ¿Cómo se inicia un "soft" handoff?

Búsqueda del Piloto

Se dice que CDMA usa Mobile Assisted Handoff (MAHO). En la práctica esto significa que la estación móvil continuamente busca el código del piloto usando un correlador PN específicamente designado para este propósito. La universalidad del código del piloto (el Código Corto) facilita la búsqueda. Todo las estaciones de base usan el mismo código. La estación móvil puede realizar búsquedas con diferentes hipótesis de tiempo sin tener que cambiar la secuencia PN.

Si el móvil ya tiene una noción del sistema de tiempo, cuando ya está involucrado en una llamada, entonces puede informar el offset relativo de un piloto recientemente encontrado. Lo que distingue las estaciones entre sí es la fase de sus pilotos. El periodo de cada piloto es 26.667 ms. Están separados por un mínimo de 64 chips, aproximadamente 52 ms o 15 km a la velocidad de luz. Esto normalmente será suficiente para que un offset del piloto identifique inequívocamente a la estación que él ha encontrado.

Umbral de detección

El móvil informa los pilotos en base a su relación piloto a interferencia (PIR). El PIR (llamado, extrañamente, E_c/I_0 en mucha literatura y normas) se compara a un umbral absoluto para determinar cuando debe informarse como un candidato de handoff. Ese umbral es un parámetro que el móvil obtiene de los mensajes transmitidos por las estaciones de base. Cuando un piloto cruza el primer umbral, T_ADD, entonces su presencia se informa, por medio de un mensaje, a la red. La red normalmente agregará esa estación de base al así llamado Juego Activo, es decir, el juego de estaciones de base que son participantes en el handoff suave del móvil en cuestión.

El segundo umbral no es absoluto sino relativo. Se compara la diferencia entre el PIR más grande en el juego activo y los PIRs de todos los otros miembros. Cuando cualquiera de ellos cae debajo de este umbral, T_DROP, entonces otro mensaje se transmite. El resultado normal es que la estación en cuestión se retirará del Juego Activo, y eso se informará al móvil mediante una instrucción determinada.

El efecto de los dos umbrales, uno absoluto, el otro relativo, es asegurar que cualquier estación que puede contribuir de alguna manera significativa al SNR global después de combinarse, esté en el juego activo. Recíprocamente, una estación sólo se deja caer cuando está muy por debajo de la mejor. Este esquema de dos umbrales es muy eficaz en la práctica. Su único posible inconveniente es que a veces provoca demasiados handoffs. Demasiados handoffs reducen la capacidad debido al número de Canales de Tráfico Directos necesarios para realizarlo. También impacta al número de elementos del canal (módemes de CDMA) necesario en las estaciones.

4.6.12: ¿Usa CDMA siempre el Soft Handoff?

No. Pero lo realiza así siempre que sea posible porque la actuación es muy superior a otras formas de handoff. Hay varias formas de handoff que no puede hacerse "suavemente".

4.6.13: Handoffs a distintas frecuencias

Si hay múltiples frecuencias portadoras de CDMA activas, entonces los handoffs entre ellas deben ser duros. Mientras el handoff inter-frecuencias es físicamente posible, se tomó la decisión en las comisiones de estandarización para no requerirlo. Llevarlo a cabo complicaría muchísimo la estación móvil. Se requerirían múltiples (por lo menos dos) sintetizadores de frecuencia independientes. Las búsquedas de los pilotos tendrían que abarcar todas las frecuencias activas. Todo esto aumentaría muchísimo el costo del terminal subscriptor, esto en un mercado muy manejado por los costos. Mientras que los handoff inter-frecuencia suaves mejorarían la actuación, se consideró que el beneficio no justifica el costo.

Los handoffs inter-frecuencias duros probablemente se logran mejor haciéndolos dentro de una misma célula, en lugar de intentarlo con una célula vecina. Los handoff de inter-frecuencia pueden ejecutarse primero dentro de la misma célula, donde el móvil ya está sincronizado, seguido inmediatamente por un handoff suave a la vecina, sin necesidad del cambio de frecuencia.

4.6.14: Digital a Analógico

No hay una manera práctica de realizar un handoff suave desde un sistema de CDMA a uno analógico, este handoff es inherentemente duro.

El caso inverso, si bien es físicamente realizable, su complejidad (y el costo que ésta significa) hace que no se justifique.

Capítulo V: Simulación del Canal Directo

Se denomina Canal Directo al vínculo de RF por el cual se transmite el mensaje desde la celda hacia el teléfono móvil.

La norma IS-95A especifica detalladamente la generación de las señales de espectro esparcido. El standard, sin embargo, no contiene detalles sobre la demodulación, por lo que cada fabricante realiza su propio diseño, pudiendo variar éste entre las distintas empresas. Sin embargo, la base del sistema sigue siendo el desarrollo realizado por Qualcomm.

5.1: Transmisión

El diagrama en bloques del transmisor puede verse a continuación.

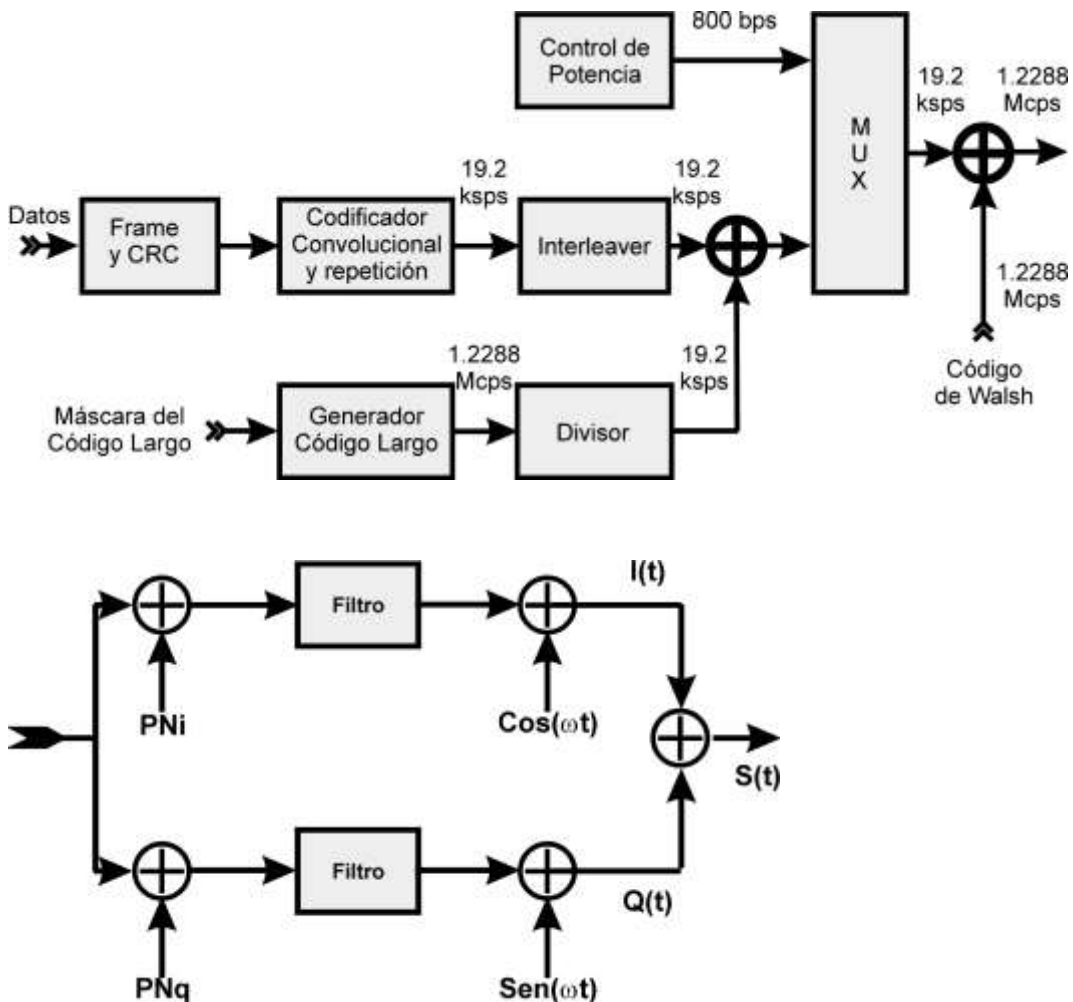


Fig. 5.1: Generación de un Canal Directo.

Existen 64 canales lógicos por cada portadora. Esto incluye a un canal Piloto, un canal de Sincronismo, 7 canales de Paging y los canales de Tráfico.

5.1.1: Generación de los Frames

Los Frames son generados por un Vocoder de velocidad variable, que produce uno cada 20 mseg. , usando una técnica CELP (code excited linear prediction). Estos Frames pueden tener una velocidad full, $\frac{1}{2}$, $\frac{1}{4}$ o $\frac{1}{8}$, dependiendo de la actividad de la voz. Tanto en Celular como en PCS pueden usarse los juegos de velocidades 1 (9600 bps) o 2 (14400 bps). La calidad de este último es superior, pero esto reduce la capacidad. Cabe aclarar que pueden usarse los dos tipos de Vocoder al mismo tiempo en el mismo sistema, brindando así una mayor calidad a algunos usuarios.

Cada Frame consta de tres partes:

- Una cantidad de bits de información, que dependerá del Vocoder y de la velocidad a la que está trabajando en ese momento.
- Bits de código CRC, cuya cantidad dependerá también del Vocoder y de la velocidad. Estos bits son verificados cuando se recupera el mensaje y sirven para decidir si el Frame recibido es correcto o no.
- Tail bits, que para todos los casos son 8 bits en cero. Estos bits son necesarios para vaciar los registros del codificador convolucional que se utiliza más adelante.

La composición de los distintos Frames puede verse en las figuras 5.2a y 5.2b

Full (9600 bps)	172 bits	12 bits CRC	8 Tail bits
$\frac{1}{2}$ (4800 bps)	80 bits	8 bits CRC	8 Tail bits
$\frac{1}{4}$ (2400 bps)	40 bits		8 Tail bits
$\frac{1}{8}$ (1200 bps)	16 bits		8 Tail bits

Fig. 5.2a: Composición de los Frames del Vocoder tipo 1

Full (14400 bps)	268 bits	12 bits CRC	8 Tail bits
$\frac{1}{2}$ (7200 bps)	126 bits	10 bits CRC	8 Tail bits
$\frac{1}{4}$ (3600 bps)	56 bits	8 bits CRC	8 Tail bits
$\frac{1}{8}$ (1800 bps)	22 bits	8 bits CRC	8 Tail bits

Fig. 5.2b: Composición de los Frames del Vocoder tipo 2

El primer bit de información para el Frame de velocidad full (en el vocoder de tipo 1) indica si los bits corresponden a voz (bit en uno), o si se están transmitiendo datos de

señalización también (bit en cero). En este último caso se utilizan los 3 bits siguientes para indicar de qué manera se comparte el Frame. Hay varias posibilidades:

- Todos los bits son de señalización, en este caso no se comparte el Frame con un mensaje (por ejemplo, puede ser la orden de finalizar la comunicación, cuando el otro usuario corta).
- Los bits de señalización son 88 cuando se completa un Frame de velocidad $\frac{1}{2}$ para transformarlo en uno de velocidad full ($80 + 88 + 4 = 172$).
- Los bits de señalización son 128 cuando se completa un Frame de velocidad $\frac{1}{4}$ para transformarlo en uno de velocidad full ($40 + 128 + 4 = 172$).
- Los bits de señalización son 168 cuando se completa un Frame de velocidad $\frac{1}{8}$ para transformarlo en uno de velocidad full ($16 + 168 + 4 = 172$).

5.1.2: Codificación CRC

El código cíclico que se agrega en cada Frame se utiliza para verificar la validez de la palabra recibida. Si se detecta un error, este ya no podrá ser corregido, y se desecha el Frame directamente, ya que en una transmisión de voz es preferible no reproducir nada antes que reproducir algo que se sabe erróneo. Esto puede apreciarse durante una conversación, e incluso en otros sistemas que trabajan de la misma manera (por ejemplo en la televisión, ya que el formato digital MPEG2 que se utiliza para transmisiones satelitales trata al sonido en una forma muy similar). En CDMA estos “silencios” no son tan evidentes, debido a que el móvil informa a la celda cuando hay una excesiva pérdida de Frames para que ésta aumente la potencia con la que transmite.

Los polinomios generadores para un Vocoder de tipo 1 son:

$$G(x) = x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^4 + x + 1$$

$$G(x) = x^8 + x^7 + x^4 + x^3 + x + 1$$

(ecuación 5.1)

Para las velocidades full y $\frac{1}{2}$ respectivamente.

Para las velocidades $\frac{1}{4}$ y $\frac{1}{8}$, la codificación CRC no es utilizada por dos motivos.

- Es más difícil que el Frame recibido sea inválido, debido a la repetición.
- Los Frames corresponden a silencios en la conversación, por lo que no es necesario “borrarlos”.

El siguiente es el procedimiento que realiza la codificación CRC del frame a transmitirse.

Can es el n° de canal (0,1 o 2), Longitud es la longitud del mensaje y CRCbits es la cantidad de bits de código cíclico que se agregan al mensaje (depende de la longitud del frame).

```
Public Sub CodificarCRC(Can As Integer, Longitud As Integer, CRCbits As Integer)
    Dim Reali As Boolean      'realimentación del registro que genera el CRC.
    Dim Reg As Object        'el registro que realiza la codificación.
    Set BasedeDatos1 = Workspaces(0).OpenDatabase(Archivo)
    Set Tabla1 = BasedeDatos1.TableDefs("CRC")
```

```

Set Registro1 = BasedeDatos1.OpenRecordset("CRC")
'Según la velocidad del Frame, se crea el registro de desplazamientos necesario.
Select Case Longitud
Case 172
'Reg de 12 bits.
Set Reg = New Cregistro9600
Case 80
'Reg de 8 bits.
Set Reg = New CRegistro4800
Case Else
'No se codifica.
Registro1.MoveFirst
For N = 1 To Longitud
Registro1.Edit
Registro1(3 + Can) = Registro1(Can)
Registro1.Update
Registro1.MoveNext
Next
Exit Sub
End Select
Registro1.MoveFirst
'Para cada bit.
For N = 1 To Longitud
'Lo guardo en el campo "Codificado" de la tabla "CRC" y lo ingreso al registro de desplazamientos.
Registro1.Edit
Registro1(3 + Can) = Registro1(Can)
'La realimentación de Reg será la or exclusiva entre el estado del último bit de Reg y la entrada.
Reali = Registro1(Can) Xor Reg.Estado(CRCbits - 1)
'Se desplaza, con el valor calculado como realimentación y como entrada.
Reg.Desplazar Reali, Reali
Registro1.Update
Registro1.MoveNext
Next
'Agrega los bits almacenados en el registro de desplazamiento al final del mensaje.
For N = Longitud + 1 To Longitud + CRCbits
'Si estamos al final de la tabla.
If Registro1.EOF Then
'Se agrega un nuevo registro donde guardar el dato.
Registro1.AddNew
'Se agrega el contenido del registro "Reg".
Registro1(3 + Can) = Reg.Estado(CRCbits - 1)
Registro1.Update
'Si quedan registros en la tabla.
Else
'Se edita el registro actual.
Registro1.Edit
'Se agrega el contenido del registro "Reg".
Registro1(3 + Can) = Reg.Estado(CRCbits - 1)
'Se actualiza el valor en la base de datos.
Registro1.Update
'Se pasa a la posición siguiente.
Registro1.MoveNext
End If
'Se desplaza el registro "Reg" sin realimentarlo.
Reg.Desplazar 0, 0
Next
BasedeDatos1.Close
End Sub

```

La clase CRegistro9600 es el registro de desplazamientos que se utiliza para la codificación cuando el frame a transmitirse es de 9600 bps, y se define de la siguiente manera:

```

Private R(11) As Boolean      'el registro de desplazamientos que se usará.
Private N As Integer         'contador.
'Este procedimiento inicializa la clase CRegistro9600.
Public Sub Class_Initialize()
    'Se pone en cero el registro.
    For N = 0 To 11
        R(N) = 0
    Next
End Sub
'Este procedimiento desplaza el registro.Realim es el valor que se usará como realimentación, y Entrada
el nuevo dato que se ingresa al registro.
Public Sub Desplazar(ByVal Realim As Boolean, Entrada As Boolean)
    R(11) = R(10) Xor Realim
    R(10) = R(9) Xor Realim
    R(9) = R(8) Xor Realim
    R(8) = R(7) Xor Realim
    R(7) = R(6)
    R(6) = R(5)
    R(5) = R(4)
    R(4) = R(3) Xor Realim
    R(3) = R(2)
    R(2) = R(1)
    R(1) = R(0) Xor Realim
    R(0) = Entrada
End Sub
'Este procedimiento se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación.
Public Property Let Estado(ByVal Indice As Integer, Valor As Boolean)
    R(Indice) = Valor
End Property
'Este procedimiento se usa al recuperar el valor de la propiedad, en la parte derecha de una asignación.
Public Property Get Estado(ByVal Indice As Integer) As Boolean
    Estado = R(Indice)
End Property
'Este procedimiento devuelve el valor de la propiedad "Estados", que se utiliza para verificar si el mensaje
es válido al decodificarlo.
Public Property Get Estados() As Integer
    Estados = 0
    For N = 0 To 11
        If R(N) Then
            Estados = Estados + 2 ^ N
        End If
    Next
End Property

```

La clase CRegistro4800 se utiliza para codificar un frame de 4800 bps, su estructura es la siguiente:

```

Private R(7) As Boolean      'el registro de desplazamientos que se usará.
Private N As Integer         'contador.
'Este procedimiento inicializa la clase CRegistro4800.
Public Sub Class_Initialize()

```

```

'Se pone en cero el registro.
For N = 0 To 7
  R(N) = 0
Next
End Sub
'Este procedimiento desplaza el registro. Realim es el valor que se usará como realimentación, y Entrada
el nuevo dato que se ingresa al registro.
Public Sub Desplazar(ByVal Realim As Boolean, Entrada As Boolean)
  R(7) = R(6) Xor Realim
  R(6) = R(5)
  R(5) = R(4)
  R(4) = R(3) Xor Realim
  R(3) = R(2) Xor Realim
  R(2) = R(1)
  R(1) = R(0) Xor Realim
  R(0) = Entrada
End Sub
'Este procedimiento se usa al asignar un valor a la propiedad, en la parte izquierda de una asignación.
Public Property Let Estado(ByVal Indice As Integer, Valor As Boolean)
  R(Indice) = Valor
End Property
'Este procedimiento se usa al recuperar el valor de la propiedad, en la parte derecha de una asignación.
Public Property Get Estado(ByVal Indice As Integer) As Boolean
  Estado = R(Indice)
End Property
'Este procedimiento devuelve el valor de la propiedad "Estados", que se utiliza para verificar si el mensaje
es válido al decodificarlo.
Public Property Get Estados() As Integer
  Estados = 0
  For N = 0 To 7
    If R(N) Then
      Estados = Estados + 2 ^ N
    End If
  Next
End Property

```

5.1.3: Codificación Convolutiva

Para asegurar una gran capacidad de corrección de errores, se utiliza un poderoso codificador convolutiva, que provee la redundancia necesaria para que el receptor pueda corregir una gran cantidad de errores.

La codificación se realiza con un Nivel de Memoria $K = 9$, cuyo esquema puede verse en la figura. Aumentar el Nivel de Memoria incrementa exponencialmente la complejidad, por lo que no es conveniente llevarlo más allá de 9 porque el aumento en la ganancia del código no lo justifica. Otras tecnologías de Telecomunicaciones utilizan $K = 4$ ó 5 .

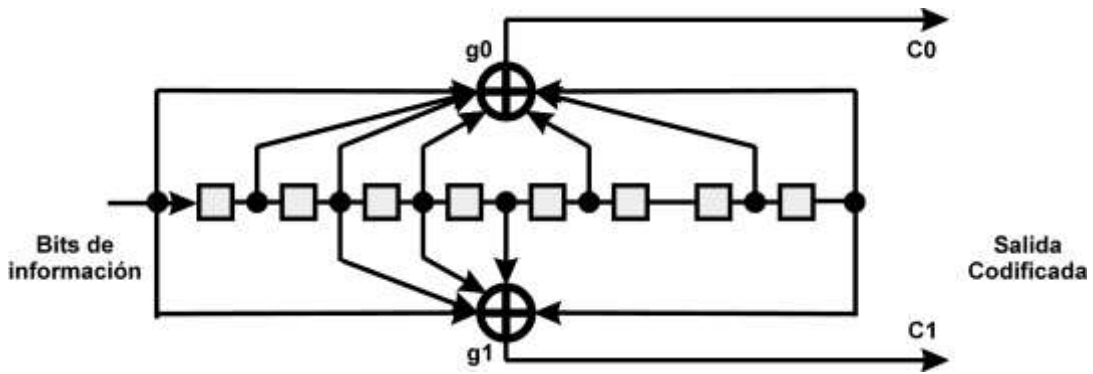


Fig. 5.3: Codificador convolucional usado en el canal directo.

Cuando se utiliza un Vocoder de Tipo 1 (9600 bps), la codificación se realiza a una velocidad $R = \frac{1}{2}$, obteniendo 19200 bps a la salida.

Al utilizar un Vocoder del Tipo 2 (14400 bps), se codifica con una velocidad $R = \frac{3}{4}$, para obtener los mismos 19200 bps a la salida. Esto se logra transmitiendo solo 4 de cada 6 símbolos generados por el codificador, como puede apreciarse en la figura 5.4.

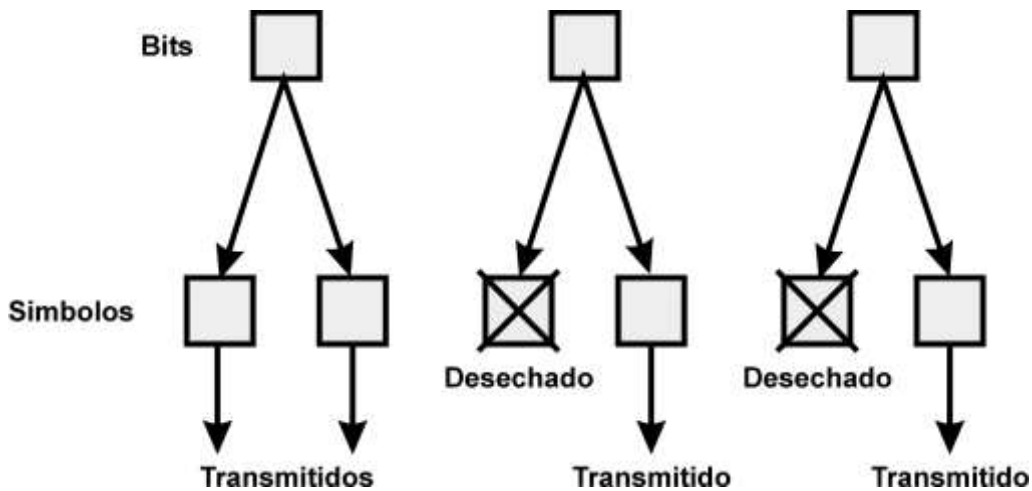


Fig. 5.4: Bits transmitidos para un vocoder de tipo 2

La ganancia de codificación para $R = \frac{1}{2}$ es de aproximadamente 4 db para una Probabilidad Binaria de Error de 10^{-3} , en un ambiente AWGN. Para $R = \frac{3}{4}$, esta ganancia se reduce a 2.5 db.

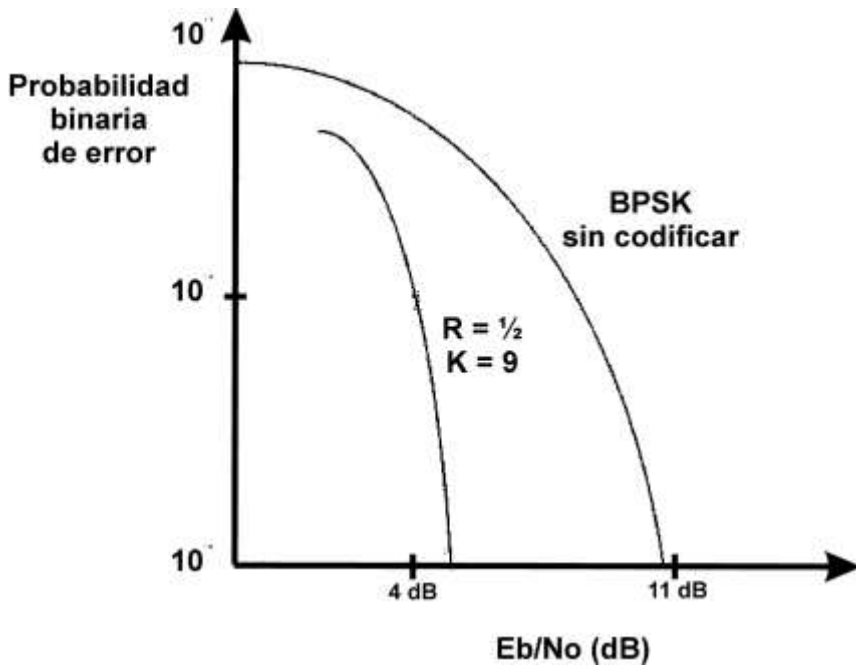


Fig. 5.5: Ganancia de codificación

El siguiente procedimiento realiza la codificación convolucional de cada Frame y la repetición del mensaje para llevar la velocidad a 19200 en todos los casos. Can es el n° de canal (0, 1 o 2) y Longitud es la longitud del Frame.

```

Public Sub CodifConvolucional(Can As Integer, Longitud As Integer)
    Dim Repeticion As Integer           'cantidad de veces que debe repetirse
    Dim I As Integer                   'contador.
    Dim Reg As New CRegistroConvol     'registro de desplazamientos de clase
    Dim BaseDatos1 As Database         'Base de datos
    Dim Tabla1 As TableDef             'Tabla de definiciones de campo
    Dim Registro1 As Recordset         'Registro de datos
    Dim Tabla2 As TableDef             'Tabla de definiciones de campo
    Dim Registro2 As Recordset         'Registro de datos

    Set BaseDatos1 = Workspaces(0).OpenDatabase("Archivos")
    Set Tabla1 = BaseDatos1.TableDefs("CRC")
    Set Registro1 = BaseDatos1.OpenRecordset("CRC")
    Set Tabla2 = BaseDatos1.TableDefs("Convolucional")
    Set Registro2 = BaseDatos1.OpenRecordset("Convolucional")

    'Según la longitud del Frame, será la repetición aplicada.
    Select Case Longitud
        Case 192
            Repeticion = 1
        Case 96
            Repeticion = 2
        Case 48
            Repeticion = 4
        Case 24
            Repeticion = 8
    End Select
    Registro1.MoveFirst
    'Para cada bit
    For N = 1 To Longitud
        'Se lo ingresa al registro convolucional
        Reg.Entrada = (Registro1(3 + Can))
    Next N
End Sub

```

'Las salidas del registro convolucional serán la palabra codificada, repitiéndose tantas veces como sea necesario

```

For I = 1 To Repeticion
  If Registro2.EOF Then
    Registro2.AddNew
    Registro2(Can) = Reg.C0
    Registro2.Update
  Else
    Registro2.Edit
    Registro2(Can) = Reg.C0
    Registro2.Update
    Registro2.MoveNext
  End If
Next
For I = 1 To Repeticion
  If Registro2.EOF Then
    Registro2.AddNew
    Registro2(Can) = Reg.C1
    Registro2.Update
  Else
    Registro2.Edit
    Registro2(Can) = Reg.C1
    Registro2.Update
    Registro2.MoveNext
  End If
Next
Registro1.MoveNext
'Se desplaza el registro convolucional.
Reg.Desplazar
Next
BasedeDatos1.Close
Set Reg = Nothing
End Sub

```

La clase CRegistroConvol es el registro de desplazamientos que se utiliza para la codificación convolucional, y se define de la siguiente manera:

```

Private R(8) As Boolean      'el registro de desplazamientos.
Private N As Integer        'contador.
Public Entrada As Boolean   'variable que se usa como una propiedad del objeto.
'Este procedimiento desplaza los bits por el registro.
Public Sub Desplazar()
  'Cada bit pasa al lugar siguiente en el registro.
  For N = 8 To 2 Step -1
    R(N) = R(N - 1)
  Next
  'El primer lugar del registro es ocupado por el bit de entrada.
  R(1) = Entrada
End Sub
'Este procedimiento devuelve el primero de los bits codificados.
Public Property Get C0() As Boolean
  'Calcula la salida según las conexiones del registro.
  C0 = Entrada Xor R(1) Xor R(2) Xor R(3) Xor R(4) Xor R(5) Xor R(7) Xor R(8)
End Property
'Este procedimiento devuelve el segundo de los bits codificados.
Public Property Get C1() As Boolean
  'Calcula la salida según las conexiones del registro.

```



```

C1 = Entrada Xor R(2) Xor R(4) Xor R(5) Xor R(8)
End Property
'Este procedimiento devuelve el estado del registro.
Public Property Get Estados() As Integer
    Estados = 0
    For N = 1 To 8
        If R(N) Then
            Estados = Estados + 2 ^ (N - 1)
        End If
    Next
End Property
'Este procedimiento establece el dato contenido en "Valor" como estado del registro.
Public Property Let Estados(Valor As Integer)
    'Para cada bit del registro.
    For N = 1 To 8
        'Lo pone en cero o uno segun "Valor".
        R(N) = Valor And 2 ^ (N - 1)
    Next
End Property

```

5.1.4: Repetición

Cuando los Frames generados por el Vocoder son de velocidades $\frac{1}{2}$, $\frac{1}{4}$ ó $\frac{1}{8}$, se realiza una repetición de los símbolos generados por el codificador convolucional (2, 4 y 8 veces, respectivamente) para lograr una velocidad constante de 19200 bps independientemente de la velocidad del Frame. Esta redundancia reduce la potencia requerida, disminuyendo la interferencia a otros usuarios y aumentando la capacidad del sistema.

5.1.5: Interleaving

El paso siguiente es el Interleaving, que es un intercambio en el orden en que serán transmitidos los símbolos. Se realiza en bloques de 20 mseg. Esto es exactamente la duración de un Frame, por lo que no se realiza mas allá de los límites de cada Frame. Los 384 bits, que corresponden a la velocidad de 19200 bps, se reordenan siguiendo siempre el mismo arreglo, y luego son reordenados en el receptor.

La razón fundamental para realizar este proceso, es que los errores que pueden introducirse durante la transmisión suelen aparecer en “bursts”, mientras que la corrección de errores es mucho mas eficiente si éstos aparecen espaciados al azar. Desordenar los símbolos antes de la transmisión tiene el efecto de “blanquear” el canal. Los errores que ocurren en “bursts” aparecen aleatoriamente esparcidos cuando los símbolos son reordenados en el receptor. Esto da como resultado una mayor ganancia de codificación.

El siguiente es el procedimiento que intercala los bits según la matriz de interleaving.

```

Public Sub Intercalar()
    Dim Canal As Integer           'n° de canal.
    Dim Marca As String           'guarda un valor de "Bookmark" para poder regresar a ese registro.
    Dim Aux(2) As Boolean         'se usan como auxiliares para intercalar los valores.
    'Se abre la base de datos "Auxiliar", donde se encuentra el orden en que se debe hacer el interleaving.
    ChDir App.Path

```

```

Set BasedeDatos1 = Workspaces(0).OpenDatabase("Auxiliar")
Set Tabla1 = BasedeDatos1.TableDefs("Interleaver")
'Este registro se usará para leer el orden en que deben intercalarse los bits.
Set Registro1 = BasedeDatos1.OpenRecordset("Interleaver")
Set BasedeDatos2 = Workspaces(0).OpenDatabase("Archivo")
Set Tabla2 = BasedeDatos2.TableDefs("Convolutacional")
Set Registro2 = BasedeDatos2.OpenRecordset("Convolutacional")
Registro1.MoveFirst
'Para cada bit del Frame.
For N = 1 To 384
  'Para cada canal, se guarda el valor en una variable auxiliar.
  For Canal = 0 To 2
    Aux(Canal) = Registro2(Canal)
  Next
  'Se guarda el valor de Bookmark.
  Marca = Registro2.Bookmark
  'Se mueve el registro activo hasta el lugar indicado.
  Registro2.Move Registro1("Desordenar") - N
  Registro2.Edit
  'Se escribe el valor donde debe ir.
  For Canal = 0 To 2
    Registro2(3 + Canal) = Aux(Canal)
  Next
  Registro2.Update
  'Se vuelve al lugar original, para seguir por el próximo.
  Registro2.Bookmark = Marca
  Registro2.MoveNext
  Registro1.MoveNext
Next
BasedeDatos1.Close
BasedeDatos2.Close
End Sub

```

Es de notar que este procedimiento realiza el procedimiento para los 3 canales simultáneamente, no como los anteriores (codificación CRC y Convolutacional) que se llaman con el canal a codificar como parámetro.

5.1.6: Scrambling

El paso siguiente es el llamado "Scrambling", consiste en la multiplicación de los 19200 símbolos por segundo por una señal pseudo-aleatoria que es también generada a 19200 chips por segundo. Cada símbolo se suma módulo-2 con un chip de la secuencia pseudo-aleatoria.

Esta secuencia es producida por el generador de Código Largo, que es un registro de desplazamientos de 42 bits, esto da como resultado una secuencia con un período de $2^{42} - 1$. El polinomio generador de la secuencia pseudo-aleatoria es el siguiente:

$$G(x) = X^{42} + X^{35} + X^{33} + X^{31} + X^{27} + X^{22} + X^{21} + X^{19} + X^{18} + X^{17} + X^{16} + X^{10} + X^7 + X^6 + X^5 + X^3 + X^2 + X + 1$$

(ecuación 5.2)

El generador es enmascarado usando una máscara de 42 bits como se muestra en la figura 5.6, esta máscara es para asegurar la privacidad de la llamada.

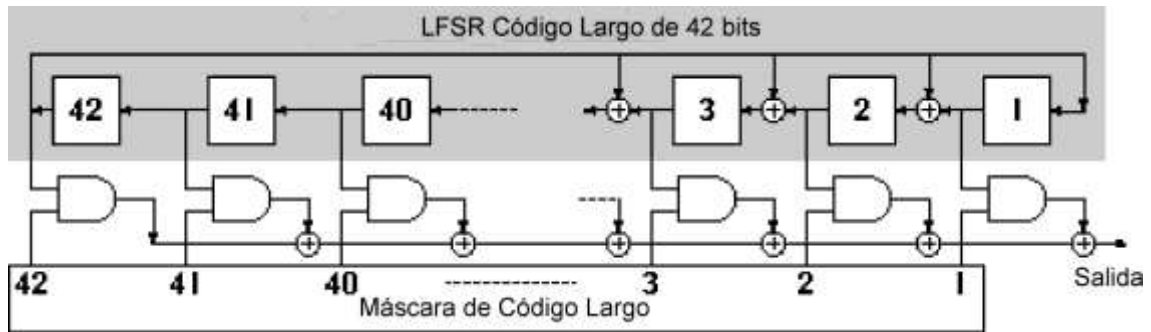


Fig. 5.6: Enmascaramiento del Código Largo

Puede demostrarse, teniendo en cuenta las propiedades de los registros de desplazamientos, que el resultado de este proceso es que para cada máscara se obtiene una distinta fase de la secuencia generada por el registro.

Los 10 bits de mayor orden de esta máscara son fijos, mientras que los restantes 32 bits se obtienen del número de serie electrónico del teléfono móvil.

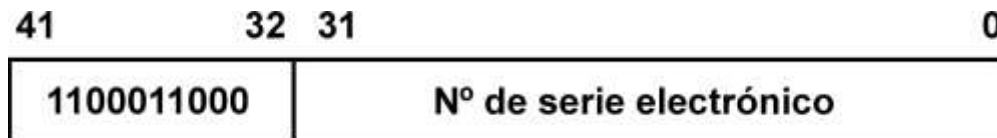


Fig. 5.7: Long Code Mask (Máscara del código largo)

El generador del Código Largo produce 1.2288 Mcps, como solo se necesitan 19200 chips por segundo se usa un divisor para bajar la velocidad de la secuencia pseudo-aleatoria. Esto se obtiene seleccionando el primer chip en cada período, como se puede apreciar en la figura 5.8.

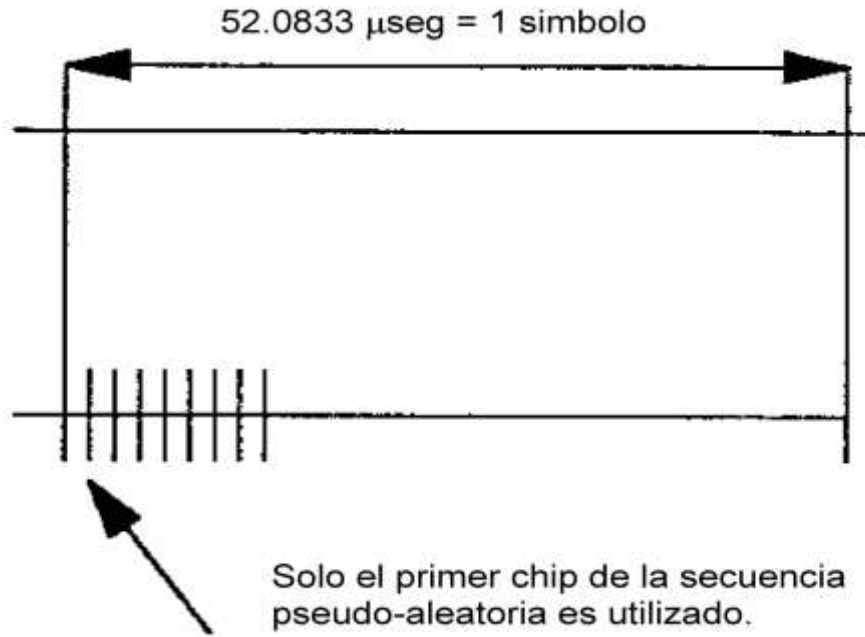


Fig. 5.8: Chips utilizados para el Scrambling.

El siguiente procedimiento genera la secuencia pseudoaleatoria PNLC, con la que se enmascarará el mensaje.

Can es el n° del canal que se quiere simular.

```
Public Sub GenerarPNLC(Can As Integer)
    Dim LCSR As New CLongCode 'registro de desplazamientos de clase CLongCode.
    Set BasedeDatos1 = Workspaces(0).OpenDatabase(Archivo)
    Set Tabla1 = BasedeDatos1.TableDefs("Convolucional")
    Set Registro1 = BasedeDatos1.OpenRecordset("Convolucional")
    'Inicializa el registro de desplazamientos, con el canal como parámetro.
    LCSR.Iniciar Can
    Registro1.MoveFirst
    'Genera la cantidad de bits necesaria y la guarda en la Base de Datos.
    For N = 1 To Tabla1.RecordCount
        Registro1.Edit
        LCSR.Desplazar
        Registro1(6 + Can) = LCSR.Salida
        Registro1.Update
        Registro1.MoveNext
    Next
    BasedeDatos1.Close
    Set LCSR = Nothing
End Sub
```

La clase CLongCode es el registro que se usa en la rutina anterior para poder generar el código largo, y se define como puede verse a continuación.

```
Private N As Integer 'contador.
Private R(1 To 42) As Boolean 'registro de desplazamientos.
Private G(0 To 42) As Boolean 'conexiones de la realimentación en el registro (True = conectado).
```

```

Private LCM(1 To 42) As Boolean 'máscara utilizada para la transmisión del mensaje.
Dim BasedeDatos As Database
Dim Tabla As TableDef
Dim Registro As Object
'Este procedimiento inicializa el registro para generar el Código Largo.
'Can es el n° del canal.
Public Sub Iniciar(Can As Integer)
    'Se abre la base de datos "Setup", donde están los datos necesarios.
    Set BasedeDatos = Workspaces(0).OpenDatabase(Setup)
    Set Tabla = BasedeDatos.TableDefs("LongCode")
    Set Registro = BasedeDatos.OpenRecordset("LongCode")
    'Inicializa los valores del registro con la fase actual, las conexiones
    'y la máscara (LCM) con el valor que corresponde según el canal.
    Registro.MoveFirst
    For N = 1 To 42
        LCM(N) = Registro(Can + 1)
        R(N) = Registro("Reg")
        G(N) = Registro("Conexiones")
        Registro.MoveNext
    Next
    BasedeDatos.Close
End Sub
'Este procedimiento desplaza el registro.
Public Sub Desplazar()
    Dim Realim As Boolean
    'Se realimenta el bit 42 del registro.
    Realim = R(42)
    'Se desplazan los bits por el registro, teniendo en cuenta las realimentaciones.
    For N = 42 To 2 Step -1
        R(N) = R(N - 1) Xor (Realim And G(N - 1))
    Next
    R(1) = Realim
End Sub
'Obtiene el bit de salida para el desplazamiento actual.
Public Property Get Salida() As Boolean
    Salida = False
    'La salida es la or exclusiva entre el contenido del registro y la máscara (LCM) utilizada para la
    transmisión del mensaje.
    For N = 1 To 42
        Salida = Salida Xor (LCM(N) And R(N))
    Next
End Property

```

El siguiente procedimiento enmascara cada mensaje con su PNLC (que ya fuera generado por el procedimiento anterior), mediante una or exclusiva.

```

Public Sub Scrambling()
    Dim Canal As Integer 'n° de canal.
    Set BasedeDatos1 = Workspaces(0).OpenDatabase(Archivo)
    Set Tabla1 = BasedeDatos1.TableDefs("Convolutacional")
    Set Registro1 = BasedeDatos1.OpenRecordset("Convolutacional")
    Registro1.MoveFirst
    'Para cada bit.
    For N = 1 To Tabla1.RecordCount
        Registro1.Edit
        'Para cada canal.
        For Canal = 0 To 2

```

```

'Se hace la or exclusiva con el "Código Largo" para enmascarar el mensaje.
Registro1(9 + Canal) = Registro1(3 + Canal) Xor Registro1(6 + Canal)
Next
Registro1.Update
Registro1.MoveNext
Next
BasedeDatos1.Close
End Sub

```

5.1.7: Esparcido

La señal, que en este punto tiene una velocidad de 19200 bps, debe llevarse a un ancho de banda de 1.25 MHz. De esta manera se realiza la división de canales en la transmisión, ya que a cada usuario se le asigna un código de Walsh diferente. Esto es fácil de realizar porque el número de chips por símbolo de mensaje es grande, ya que tomando una velocidad de 1.2288 Mcps se obtienen 64 chips por cada símbolo. Los códigos de Walsh son 64 secuencias, de 64 chips cada una, ortogonales entre sí. Esto permite una perfecta separación entre canales, ya que todos los usuarios en un mismo sector comparten el mismo "timing" y un mismo "Código Corto", permitiendo que los canales sean ortogonales entre sí al arribar al receptor, pues provienen de una misma antena. Esto se realiza, entonces, multiplicando cada símbolo del mensaje que se transmitirá por los 64 chips del código de Walsh que se asignó a esa llamada.

Los canales que son utilizados por el sistema tienen asignados para sí determinados códigos de Walsh, el canal Piloto corresponde al código de Walsh 0, el canal de sincronismo está en el código 32, y los canales de Paging son los códigos 1 al 7. El resto de los códigos se utilizan como canales de tráfico, y son asignados a cada usuario al iniciarse una llamada. Los canales de Paging no usados pueden asignarse como canales de tráfico, aunque no es necesario porque el límite en la capacidad del sistema no es la cantidad de canales lógicos.

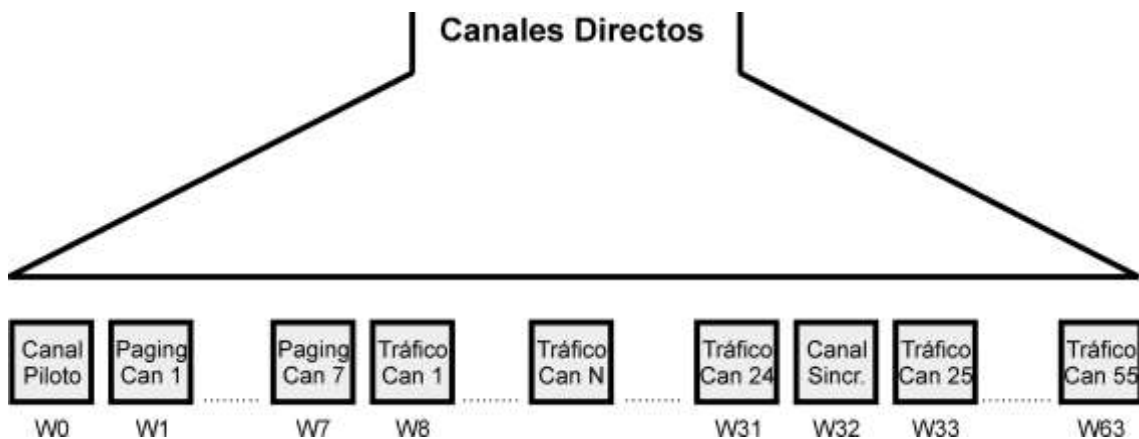


Fig. 5.9: Canales Directos de CDMA.

El siguiente procedimiento realiza la multiplicación de cada bit por el código de Walsh correspondiente, lo que produce un esparcido en frecuencia.

Can es el nº de canal que se simulará y Codigo es el nº del Código de Walsh a utilizar.

```

Public Sub Walsh(Can As Integer, Codigo As Integer)
    Dim WalshCode(63) As Boolean    'se usa para almacenar el código de Walsh correspondiente.
    Dim I As Integer                'contador.
    ChDir App.Path
    'Se abre la base de datos "Auxiliar".
    Set BasedeDatos1 = Workspaces(0).OpenDatabase("Auxiliar")
    Set Tabla1 = BasedeDatos1.TableDefs("Walsh")
    'En esta tabla se encuentran los 64 códigos de Walsh.
    Set Registro1 = BasedeDatos1.OpenRecordset("Walsh")
    'Se busca el código de Walsh correspondiente al canal.
    Registro1.MoveFirst
    Registro1.Move Codigo
    For N = 0 To 63
        'Se guarda el valor del código.
        WalshCode(N) = Registro1(N)
    Next
    Set BasedeDatos1 = Workspaces(0).OpenDatabase(Archivo)
    Set Tabla1 = BasedeDatos1.TableDefs("Convolucional")
    Set Registro1 = BasedeDatos1.OpenRecordset("Convolucional")
    Set Tabla2 = BasedeDatos1.TableDefs("Walsh")
    Set Registro2 = BasedeDatos1.OpenRecordset("Walsh")
    Registro1.MoveFirst
    'Se multiplica cada bit por el código de walsh correspondiente.
    For N = 1 To Tabla1.RecordCount
        For I = 0 To 63
            If Registro2.EOF Then
                Registro2.AddNew
                Registro2(Can) = WalshCode(I) Xor Registro1(9 + Can)
                Registro2.Update
            Else
                Registro2.Edit
                Registro2(Can) = WalshCode(I) Xor Registro1(9 + Can)
                Registro2.Update
                Registro2.MoveNext
            End If
        Next
        Registro1.MoveNext
    Next
    BasedeDatos1.Close
End Sub

```

5.1.8: Modulación en cuadratura

Una vez que se ha multiplicado el mensaje por el código de Walsh asignado, se modula en cuadratura. Toda la información es enviada en ambas componentes (I, Q). Esto asegura que la interferencia mutua sea homogénea en fase. Cada componente se obtiene multiplicando el mensaje por un código pseudo-aleatorio, llamado "Código Corto". Hay un código corto para la componente **I** y otro distinto para la componente **Q**, con distintos generadores y baja correlación entre ellos. En la figura 5.10 puede verse la lógica que genera las dos componentes.

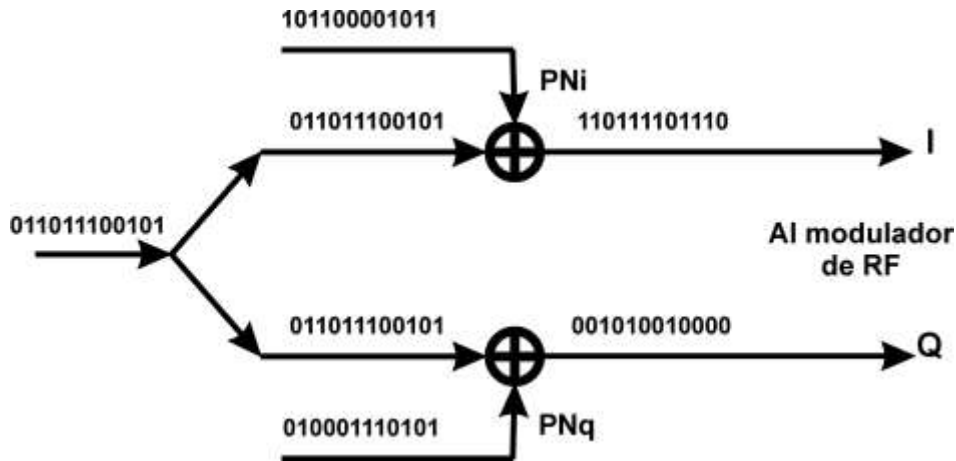


Fig. 5.10: Generación de las componentes I y Q .

Este código es usado para aislar los sectores entre sí, permitiendo la reutilización de los códigos de Walsh en cada sector. Esto se logra mediante un código común a todas las estaciones de base, que al estar sincronizadas entre sí pueden transmitir diferentes fases de la secuencia, permitiendo así que los móviles puedan distinguirlas.

La autocorrelación de una secuencia generada por un LFSR (Linear Feedback Shift Register) tiene la forma que se ve en la figura 5.11.

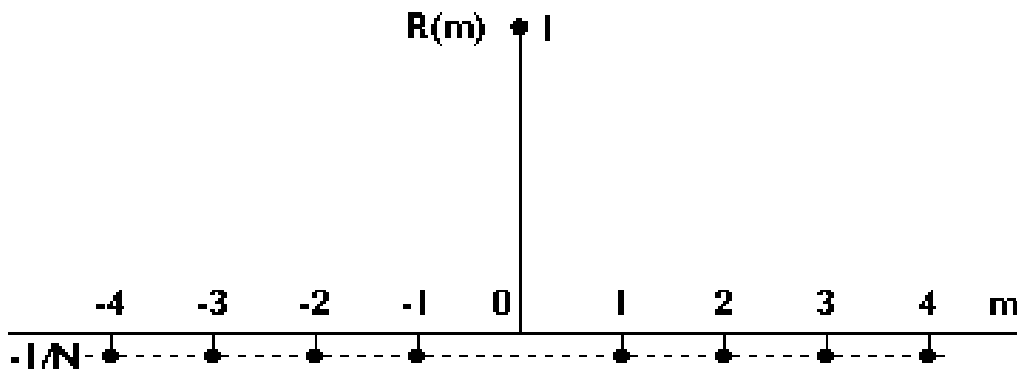


Fig. 5.11: Autocorrelación de las secuencias ($N = \text{período}$).

La separación mínima está relacionada con la cantidad de células que componen el sistema, y debe ser mayor que el retardo de propagación mas grande que pueda ocurrir en el sistema.

El offset mínimo permitido es de 64 chips (equivale a 15.6 km por retardos de propagación). Los operadores pueden elegir un offset mayor, usando generalmente 128 o 256 chips de offset.

El código corto tiene un período 2^{15} , que equivale a $80/3 = 26.667$ ms a una frecuencia de 1.2288 MHz. Esta longitud es un compromiso entre el tiempo de búsqueda y el número de offsets disponibles para el sistema.

Así, con un offset de 64 chips entre estaciones, hay 512 posibles fases. Como el período de la secuencia es $2^{15} - 1 = 32767$, debe aumentarse agregando un cero extra en determinado lugar de la secuencia (después de una corrida de 14 0's). Esto hace iguales la cantidad de unos y ceros, y aunque deteriora un poco la propiedad de autocorrelación

que vimos anteriormente (fig 9), es necesario para lograr un sistema con fases equidistantes entre los distintos sectores.

Los polinomios generadores para las secuencias **I** y **Q** son los siguientes:

$$P_I(X) = X^{15} + X^{13} + X^9 + X^8 + X^7 + X^5 + 1$$

$$P_Q(X) = X^{15} + X^{12} + X^{11} + X^{10} + X^6 + X^5 + X^4 + X^3 + 1$$

(ecuación 5.3)

El siguiente procedimiento genera las secuencias pseudoaleatorias que identifican a cada sector (código corto), y las utiliza para enviar la información en cuadratura.

La variable booleana "Sectores" indica si hay que simular dos sectores de la celda. Como puede verse en el código, se utiliza un offset de 512 chips entre los sectores simulados.

```
Public Sub Cuadratura(Sectores As Boolean)
    Dim SCSR(1) As New CShortCode      'registro de desplazamientos de clase CShortCode.
    Dim Sector As Integer              'sector de la celda que transmitirá el mensaje (0 ó 1).
    Dim ContI As Integer               'contador de ceros en la componente "I".
    Dim ContQ As Integer               'contador de ceros en la componente "Q".
    Dim Can As Integer
    Set BasedeDatos1 = Workspaces(0).OpenDatabase(Archivo)
    Set Tabla1 = BasedeDatos1.TableDefs("Walsh")
    Set Registro1 = BasedeDatos1.OpenRecordset("Walsh")
    'Primer sector de la celda.
    Sector = 0
    Do
        'Genera 512 chips de offset para el segundo sector, mediante el desplazamiento del registro
        correspondiente.
        For N = 1 To 512 * Sector
            SCSR(Sector).DesplazarI
            SCSR(Sector).DesplazarQ
        Next
        Registro1.MoveFirst
        'Se inicializan los contadores de ceros.
        ContI = 0
        ContQ = 0
        'Para cada bit
        For N = 1 To Tabla1.RecordCount
            Registro1.Edit
            'Genera el bit I del "Código Corto". Si hubo una corrida de 14 ceros seguidos, el proximo valor
            será otro cero. Sino el próximo valor es la salida del registro "SCSR" luego de un desplazamiento.
            If ContI = 14 Then
                Registro1(3 + 2 * Sector) = 0
            Else
                SCSR(Sector).DesplazarI
                Registro1(3 + 2 * Sector) = SCSR(Sector).SalidaI
            End If
            'Genera el bit Q del "Código Corto"
            If ContQ = 14 Then
                Registro1(4 + 2 * Sector) = 0
            Else
```

```

SCSR(Sector).DesplazarQ
Registro1(4 + 2 * Sector) = SCSR(Sector).SalidaQ
End If
'Se lleva la cuenta de los ceros para poder agregar otro desués de una corrida de 14.
If SCSR(Sector).SalidaI = False And ContI < 14 Then
    ContI = ContI + 1
Else
    ContI = 0
End If
If SCSR(Sector).SalidaQ = False And ContQ < 14 Then
    ContQ = ContQ + 1
Else
    ContQ = 0
End If
'Para cada canal, se obtienen las componentes en fase y en cuadratura.
'Si el canal 2 corresponde a otro sector.
If Sectores Then
    'Si se está simulando el sector 0,
    If Sector = 0 Then
        'Se simulan los 2 primeros canales.
        For Can = 0 To 1
            Registro1(7 + 2 * Can) = Registro1("SortI 0") Xor Registro1(Can)
            Registro1(8 + 2 * Can) = Registro1("SortQ 0") Xor Registro1(Can)
        Next
        'Si se está simulando el otro sector, se calcula el último canal.
    Else
        Registro1("CuadraturaI 2") = Registro1("SortI 1") Xor Registro1("Esparcido 2")
        Registro1("CuadraturaQ 2") = Registro1("SortQ 1") Xor Registro1("Esparcido 2")
    End If
    'Si los tres canales se transmiten por el mismo sector.
    Else
        'Se simulan para los 3 canales
        For Can = 0 To 2
            Registro1(7 + 2 * Can) = Registro1("SortI 0") Xor Registro1(Can)
            Registro1(8 + 2 * Can) = Registro1("SortQ 0") Xor Registro1(Can)
        Next
    End If
    Registro1.Update
    Registro1.MoveNext
Next
'Se pasa al otro sector
Sector = Sector + 1
'Se repite el bucle si hay dos sectores para simular y solo se simuló el primero.
Loop While Sectores And Sector <= 1
BasedeDatos1.Close
End Sub

```

La clase CShortCode, que puede verse a continuación, se utiliza para generar las dos componentes del código corto.

```

Private N As Integer           'contador.
Private RI(1 To 15) As Boolean 'registro de desplazamientos para la componente "I".
Private RQ(1 To 15) As Boolean 'registro de desplazamientos para la componente "Q".
Private GI(0 To 15) As Boolean 'conexiones de realimentación del registro "I".
Private GQ(0 To 15) As Boolean 'conexiones de realimentación del registro "Q".
'Este procedimiento inicializa los registros para generar las dos componentes del Código Corto.
Private Sub Class_Initialize()

```

'Se abre la base de datos "Setup", donde están los datos necesarios.

```
Set BasedeDatos = Workspaces(0).OpenDatabase(Setup)
```

```
Set Tabla = BasedeDatos.TableDefs("ShortCode")
```

```
Set Registro = BasedeDatos.OpenRecordset("ShortCode")
```

'Inicializa los valores de los registros con la fase actual y las conexiones.

```
Registro.MoveFirst
```

```
For N = 1 To 42
```

```
    RI(N) = Registro("RegI")
```

```
    RQ(N) = Registro("RegQ")
```

```
    GI(N) = Registro("ConexionesI")
```

```
    GQ(N) = Registro("ConexionesQ")
```

```
    Registro.MoveNext
```

```
Next
```

```
BasedeDatos.Close
```

```
End Sub
```

'Este procedimiento produce el desplazamiento en el registro que genera la componente "I" del Código Corto.

```
Public Sub DesplazarI()
```

```
    Dim RealimI As Boolean
```

'Se realimenta el bit 15 del registro.

```
    RealimI = RI(15)
```

'Se desplazan los bits por el registro, teniendo en cuenta las realimentaciones.

```
    For N = 15 To 2 Step -1
```

```
        RI(N) = RI(N - 1) Xor RealimI And GI(N - 1)
```

```
    Next
```

```
    RI(1) = RealimI
```

```
End Sub
```

'Este procedimiento produce el desplazamiento en el registro que genera la componente "Q" del Código Corto.

```
Public Sub DesplazarQ()
```

```
    Dim RealimQ As Boolean
```

'Se realimenta el bit 15 del registro.

```
    RealimQ = RQ(15)
```

'Se desplazan los bits por el registro, teniendo en cuenta las realimentaciones.

```
    For N = 15 To 2 Step -1
```

```
        RQ(N) = RQ(N - 1) Xor RealimQ And GQ(N - 1)
```

```
    Next
```

```
    RQ(1) = RealimQ
```

```
End Sub
```

'Este procedimiento obtiene el bit de salida para la componente "I".

```
Public Property Get SalidaI() As Boolean
```

```
    SalidaI = RI(15)
```

```
End Property
```

'Este procedimiento obtiene el bit de salida para la componente "Q".

```
Public Property Get SalidaQ() As Boolean
```

```
    SalidaQ = RQ(15)
```

```
End Property
```

5.1.9: Modulación

El diagrama en bloques del modulador puede verse en la figura 5.12. Las secuencias **I** y **Q** se utilizan para generar una señal QPSK, mediante dos portadoras en cuadratura. Los "Moduladores de impulsos" generan una salida (-1, 1) para una entrada binaria (0, 1).

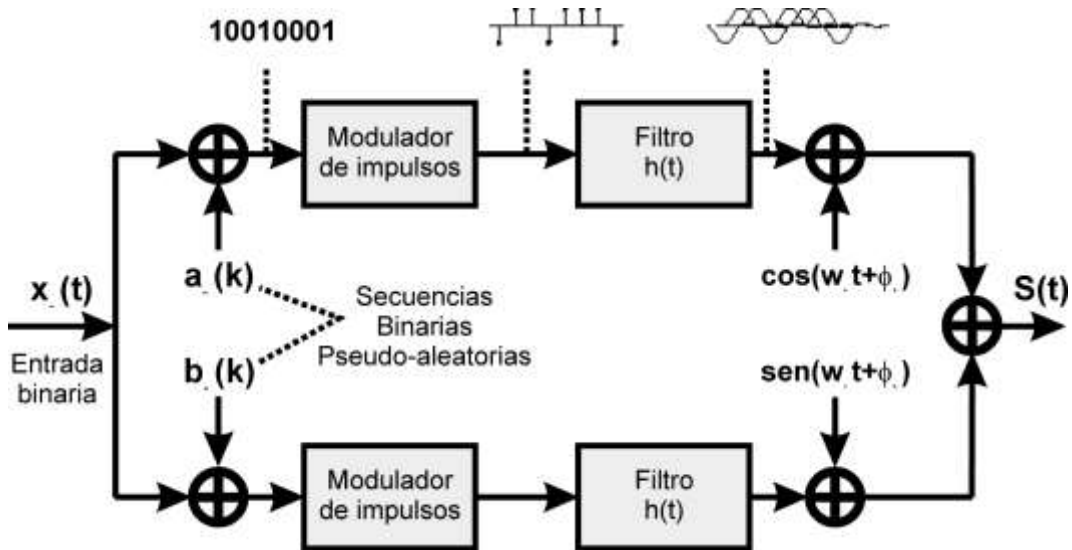


Fig. 5.12: Modulador QPSK.

El siguiente procedimiento simula la salida del filtro, generando pulsos en forma de sinc con los mensajes de cada canal.

```

Public Sub Pulsos()
    Dim DatosI(3, 6) As Integer 'guarda los datos que serán transmitidos
    Dim DatosQ(3, 6) As Integer 'en fase y en cuadratura, en formato polar.
    Dim I As Integer 'contador.
    Dim Can As Integer 'nº de canal.
    Set BasedeDatos1 = Workspaces(0).OpenDatabase(Archivo)
    Set Tabla1 = BasedeDatos1.TableDefs("Walsh")
    'En este registro se podrán leer los datos de la tabla "Walsh", que contiene los valores de cada mensaje,
    en sus componentes "I" y "Q".
    Set Registro1 = BasedeDatos1.OpenRecordset("Walsh")
    Set Tabla2 = BasedeDatos1.TableDefs("RF")
    'Este registro se usará para guardar en la tabla "RF" los valores de los pulsos para cada muestra
    simulada.
    Set Registro2 = BasedeDatos1.OpenRecordset("RF")
    Registro1.MoveFirst
    'Para cada bit de mensaje (se suma 5 para poder generar completo el último pulso).
    For I = 1 To Tabla1.RecordCount + 5
        'Se da el valor que indicará la barra de progreso.
        FormEspera.ProgressBar.Value = I / (Tabla1.RecordCount + 5) * 50
        'Para cada canal
        For Can = 0 To 2
            'Se desplazan los datos ya leídos, para leer el nuevo.
            For N = 6 To 2 Step -1
                DatosI(Can, N) = DatosI(Can, N - 1)
                DatosQ(Can, N) = DatosQ(Can, N - 1)
            Next
            'Si se terminaron los datos (ocurre para los últimos 5 bits) se leen como ceros.
            If Registro1.EOF Then
                DatosI(Can, 1) = 0
                DatosQ(Can, 1) = 0
            Else
                'Sino, se leen como 1 o -1 para la componente en fase del canal.
                If Registro1(7 + 2 * Can) Then

```

```

        DatosI(Can, 1) = 1
    Else
        DatosI(Can, 1) = -1
    End If
    'Igualmente para la componente en cuadratura.
    If Registro1(8 + 2 * Can) Then
        DatosQ(Can, 1) = 1
    Else
        DatosQ(Can, 1) = -1
    End If
End If
End If
Next
Registro2.AddNew
'Para cada canal.
For Can = 0 To 2
    'Se guardan en la columna correspondiente los valores de los pulsos "I" y "Q" cuando la sinc tiene
    su valor máximo, que coincide con un cero para los otros bits del mensaje.
    Registro2(2 * Can) = DatosI(Can, 4)
    Registro2(2 * Can + 1) = DatosQ(Can, 4)
Next
Registro2.Update
'Para los demás valores debe tenerse en cuenta los bits anteriores y posteriores del mensaje, así como
también el valor del respectivo pulso en ese instante.
For N = 1 To 31
    Registro2.AddNew
    For Can = 0 To 2
        Registro2(2 * Can) = DatosI(Can, 1) * P(96 - N) + DatosI(Can, 2) * P(64 - N) + DatosI(Can, 3)
        * P(32 - N) + DatosI(Can, 4) * P(N) + DatosI(Can, 5) * P(32 + N) + DatosI(Can, 6) * P(64 + N)
        Registro2(2 * Can + 1) = DatosQ(Can, 1) * P(96 - N) + DatosQ(Can, 2) * P(64 - N) +
        DatosQ(Can, 3) * P(32 - N) + DatosQ(Can, 4) * P(N) + DatosQ(Can, 5) * P(32 + N) + DatosQ(Can, 6) *
        P(64 + N)
    Next
    Registro2.Update
Next
'Se mueve el registro para buscar el bit siguiente, si todavía hay datos que leer.
If Not Registro1.EOF Then
    Registro1.MoveNext
End If
Next
Registro2.AddNew
'Se agrega el cero final para cada canal
For Can = 0 To 2
    Registro2(2 * Can) = 0
    Registro2(2 * Can + 1) = 0
Next
Registro2.Update
BasedeDatos1.Close
End Sub

```

La siguiente función es la que devuelve el valor del pulso (una función sinc) para un instante dado.

N es el valor de muestra, entre 0 y 96 (es una función par).

```

Public Function P(N As Integer) As Single
    Const R = 1228800          'ancho de banda del sistema.
    Const Dt = 2.54313151E-08 'intervalo entre muestras elegido.
    If N = 0 Then

```

```

P = 1
Else
  P = Sin(Pi * R * N * Dt) / (Pi * R * N * Dt)
End If
End Function
    
```

Las componentes **I** y **Q** producen entonces cambios de fase en la portadora, como puede apreciarse en la figura 5.13. Cuando los valores de ambas componentes cambian simultáneamente, se produce un cambio de fase de 180°.

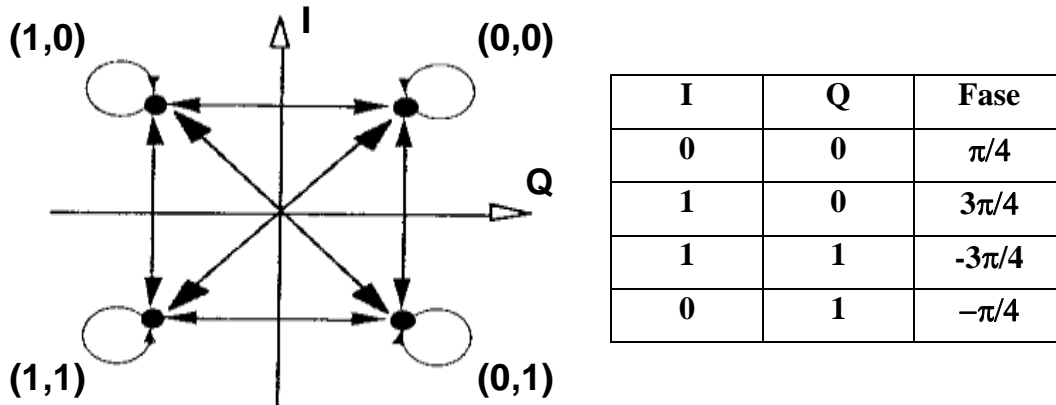


Fig. 5.13: Cambios de fase en la portadora.

La forma de onda debe estar acotada a un ancho de banda especificado, por lo que debe utilizarse un filtro pasabajos con un ancho de banda de 615 KHz.

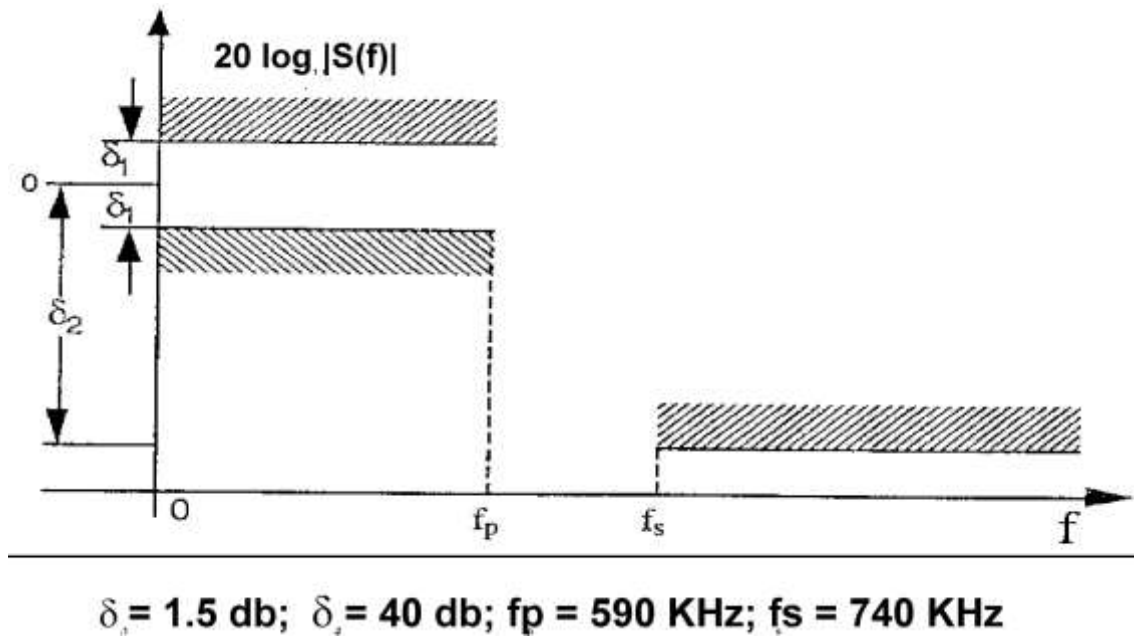


Fig. 5.14: Características del filtro pasabajos especificado por el standard.

El siguiente procedimiento realiza la modulación de cada canal, sumándolos luego para obtener la salida.

Public Sub Modulación()

Dim I As Long 'contador.

Dim Can As Integer 'n° de canal.

Dim Aux As Single 'variable auxiliar donde se acumula la suma de los 3 canales.

Set BasedeDatos1 = Workspaces(0).OpenDatabase(Archivo)

Set Tabla1 = BasedeDatos1.TableDefs("RF")

Set Registro1 = BasedeDatos1.OpenRecordset("RF")

Registro1.MoveFirst

'Para cada valor

For I = 1 To (Tabla1.RecordCount - 1) / 8

'Se da el valor que indicará la barra de progreso.

FormEspera.ProgressBar.Value = 50 + I / ((Tabla1.RecordCount - 1) / 8) * 50

'Se tomarán 8 puntos por cada ciclo de portadora (se simulan cuatro ciclos enteros por cada dato).

For N = 0 To 7

'Se inicializa la variable auxiliar.

Aux = 0

'Se modulan las componentes en fase y en cuadratura de cada canal y se acumula la suma.

For Can = 0 To 2

Aux = Aux + Registro1(2 * Can) * Cos(Pi / 4 * N) + Registro1(2 * Can + 1) * Sin(Pi / 4 * N)

Next

'Se guarda el resultado en la base de datos.

Registro1.Edit

Registro1("Salida") = Aux

Registro1.Update

Registro1.MoveNext

Next

Next

'Se agrega el último valor

Registro1.Edit

Registro1("Salida") = 0

Registro1.Update

BasedeDatos1.Close

End Sub

5.1.10: Canal Piloto

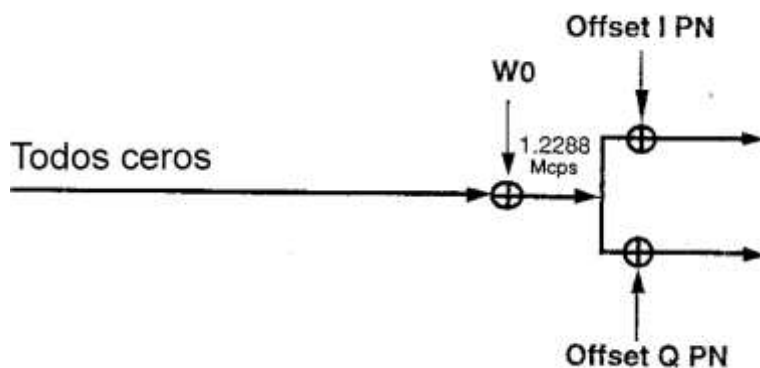


Fig. 5.15: Generación del Canal Piloto.

El canal Piloto no contiene ninguna información, es una secuencia de ceros modulada con el código de Walsh cero y el Código Corto con el offset correspondiente al sector. La demodulación del Piloto permite al móvil obtener una referencia de tiempo, fase e intensidad de la señal. Como todos los canales en una portadora comparten el mismo "timing", con idénticas características de propagación, esta referencia que brinda el Piloto puede utilizarse para demodular los demás canales en forma coherente.

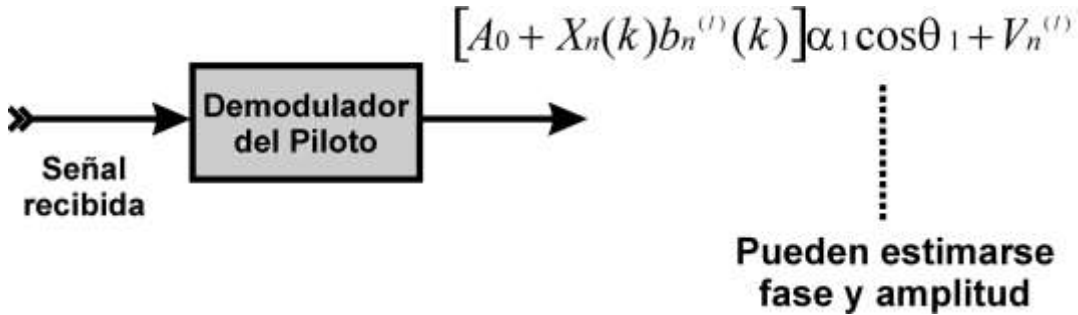


Fig. 5.16: Demodulación del Canal Piloto.

Además de utilizarse como una referencia de fase para demodular coherentemente, el Piloto también se usa para decidir cuando debe iniciarse un handoff, por lo que su amplitud debe ser cuidadosamente controlada porque fija el límite entre las distintas células del sistema.

Como todas las celdas usan el mismo Código Corto, todos los Pilotos tendrán la misma forma de onda, distinguiéndose entre sí (e identificando a cada sector) por el offset. El período de estas secuencias (26.667ms) facilita una rápida búsqueda por parte de los móviles. La información obtenida del Piloto también se utiliza para asignar a los receptores Rake el retardo apropiado.

5.1.11: Canal de Sincronismo

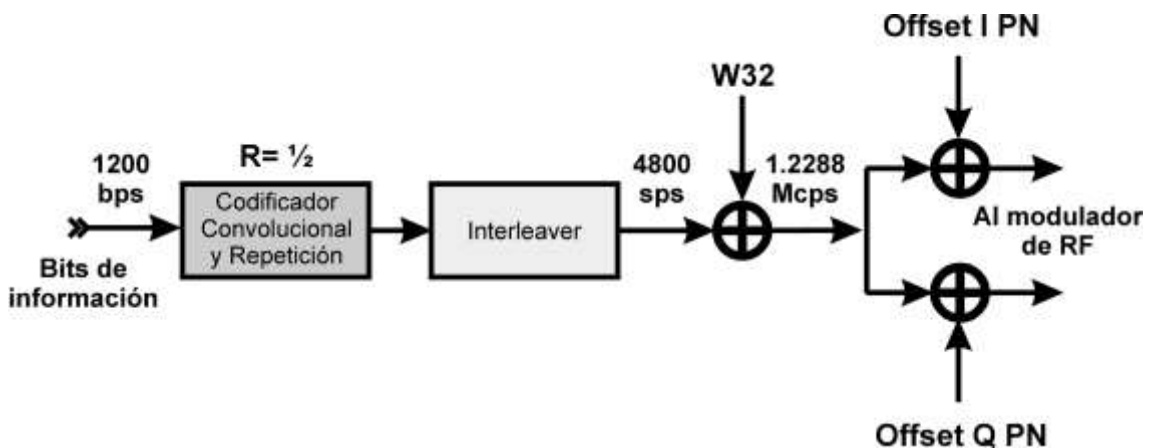


Fig. 5.17: Generación del Canal de Sincronismo.

El canal de Sincronismo transmite un mensaje a una velocidad de 1200 bps, que será codificado y repetido de la misma manera que un canal de tráfico, excepto el proceso de "Scrambling", y al cual se le asigna siempre el código de Walsh 32.

El mensaje transmitido es la información necesaria para que el móvil se sincronice con la celda, como el offset del Piloto, el estado del Código Largo e información sobre el "timing" y la configuración del sistema. Una vez que se recibe el Canal de Sincronismo, el móvil está en condiciones de leer los canales de "Paging".

5.1.12: Canales de Paging

La generación de un canal de Paging es similar a la de los canales de tráfico, con la excepción del control de potencia.

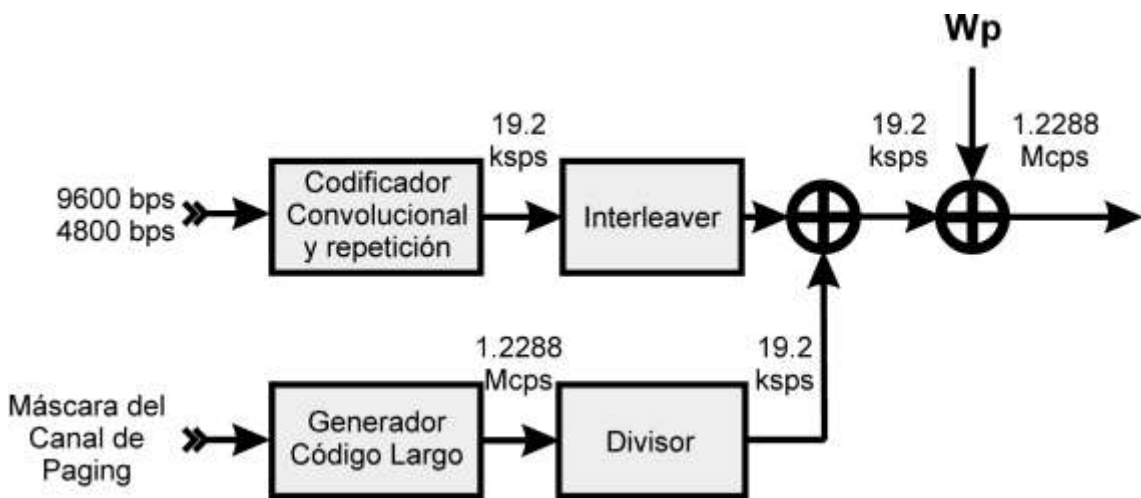
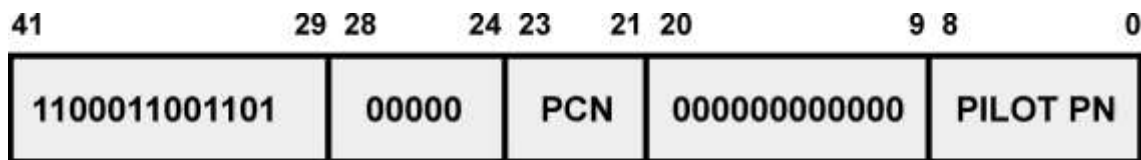


Fig. 5.18: Generación de un Canal de Paging.

La máscara que se utiliza para el Código Largo contiene los datos del número de canal de Paging correspondiente y el offset del Código Corto que identifica al sector. Estos datos ya son conocidos por el móvil, que puede decodificar el canal después de haber recibido el Canal de Sincronismo.



PCN: Paging Chanel Number.

PILOT PN: Offset que identifica al sector.

Fig. 5.19: Máscara del Código Largo de un Canal de Paging.

El Canal de Paging es el vehículo para la comunicación con los móviles cuando no están asignados a un canal de tráfico. Su principal propósito es notificar a los móviles

cuando reciben una llamada. También responde los mensajes que envía el móvil por el Canal de Acceso, ya sea como respuesta a una llamada entrante o cuando el móvil quiere originar una llamada. De esta manera se inicia la llamada, comunicándole al móvil el canal de tráfico que se le ha asignado. Una vez que este canal ha sido asignado, la comunicación entre el móvil y la celda se hace por esta vía, junto con la transmisión de voz.

Para aumentar el rendimiento de las baterías de los teléfonos móviles, se le asigna a cada uno un determinado "slot" de tiempo en el cual puede recibir un mensaje, no siendo necesario que el móvil esté continuamente recibiendo el canal.

5.2: Características del canal de transmisión

El canal de transmisión puede representarse por medio del modelo que se ilustra en la figura. Este modelo consta de atenuaciones y retardos diferentes para cada usuario, con ruido térmico Gaussiano presente en el receptor, donde una posible interferencia es representada por $q(t)$.

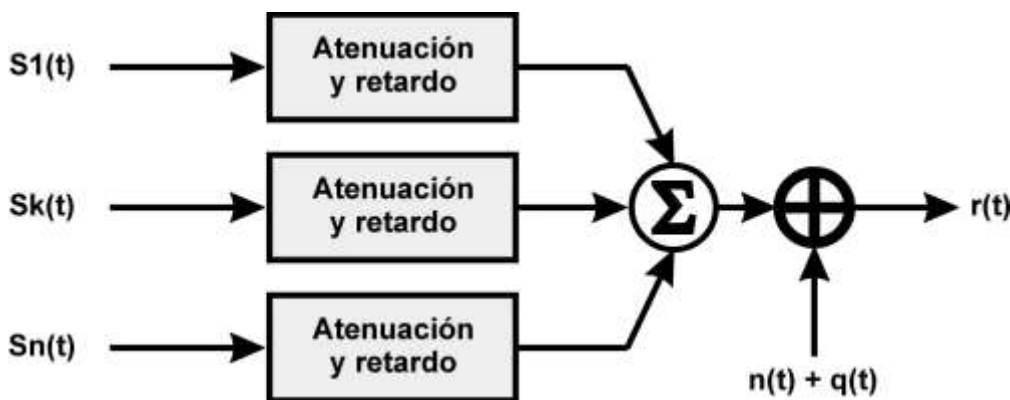


Fig. 5.20: Modelo del Canal de Transmisión.

En el modelo se tienen en cuenta solo tres componentes del multipath, debido a que son los que pueden ser resueltos por el receptor Rake, que tiene tres elementos.

Las demás componentes pueden despreciarse por varios motivos. En primer lugar, se buscan las componentes más fuertes para recibirlas. Además, las componentes que podrían ser resueltas por el receptor y no son tenidas en cuenta, se comportan de la misma forma que la interferencia de los demás usuarios, aportando poco al ruido de fondo debido a su magnitud. Las componentes que presentan un retardo muy pequeño no pueden ser resueltas por el receptor, introduciendo interferencia intersimbólica, cuyo efecto también puede compararse a la interferencia de otros usuarios.

El siguiente procedimiento simula el efecto de los distintos rebotes de la señal, que provocan el arribo de múltiples copias de la señal transmitida.

Los parámetros son las distintas amplitudes y retardos de tiempo de las señales que llegarán.

```

Public Sub Retardos(Intensidad() As Single, Retardo() As Integer)
    Dim Sumar As Boolean 'indica si el registro "2" debe sumarse al contenido actual.
    Set BasedeDatos = Workspaces(0).OpenDatabase(Archivo)
    Set Tabla = BasedeDatos.TableDefs("RF")
    'Se abren 3 registros para desplazarse por la tabla "RF".
    Set Registro0 = BasedeDatos.OpenRecordset("RF")
    Set Registro1 = BasedeDatos.OpenRecordset("RF")
    Set Registro2 = BasedeDatos.OpenRecordset("RF")
    'El registro "0" empieza por el primer registro de "RF".
    Registro0.MoveFirst
    'El registro "1" se desplaza de acuerdo al retardo correspondiente.
    Registro1.MoveFirst
    Registro1.Move Retardo(1)
    'El registro "2", además de desplazarse, pone en cero la columna "Llegada". Esto es necesario porque
    puede haber una simulación anterior del canal.
    Registro2.MoveFirst
    For N = 1 To Retardo(2)
        Registro2.Edit
        Registro2("Llegada") = 0
        Registro2.Update
        Registro2.MoveNext
    Next
    'Para cada instante.
    For N = 1 To 786593
        'Se da el valor que indicará la barra de progreso.
        FormEspera.ProgressBar.Value = N / 786593 * 100
        'Se suma el primer arribo, con amplitud 1.
        Registro0.Edit
        Registro0("Llegada") = Registro0("Llegada") + Registro0("Salida")
        Registro0.Update
        'Si el registro "1" ha llegado al final de la tabla.
        If Registro1.EOF Then
            'Se vuelve al principio.
            Registro1.MoveFirst
        End If
        'Se suma el segundo arribo, con la amplitud correspondiente.
        Registro1.Edit
        Registro1("Llegada") = Intensidad(1) * Registro0("Salida") + Registro1("Llegada")
        Registro1.Update
        'Si el registro "2" ha llegado al final de la tabla.
        If Registro2.EOF Then
            'Se vuelve al principio.
            Registro2.MoveFirst
            'El valor se deberá sumar al ya guardado.
            Sumar = True
        End If
        Registro2.Edit
        'Si ya se volvió al principio de la tabla.
        If Sumar Then
            'Se suma al valor ya guardado el nuevo arribo con su intensidad.
            Registro2("Llegada") = Intensidad(2) * Registro0("Salida") + Registro2("Llegada")
        Else
            Registro2("Llegada") = Intensidad(2) * Registro0("Salida")
        End If
        Registro2.Update
        Registro0.MoveNext
        Registro1.MoveNext
        Registro2.MoveNext
    Next
End Sub

```

```

Next
BasedeDatos.Close
End Sub

```

Como se simula solamente un frame, los retardos no pueden extender la señal mas allá del mismo. Puede verse en el código anterior que esto se solucionó volviendo al principio del frame, por lo que cada dato será la suma de las tres componentes defasadas.

El siguiente procedimiento simula las condiciones de ruido en el canal de transmisión. Los parámetros son la Intensidad del ruido blanco presente en todo momento, y la duración y el comienzo de un pulso que se utiliza para simular la pérdida de una cantidad de bits determinada.

```

Public Sub Ruidos(Intensidad As Integer, Duracion As Long, Comienzo As Long)
  Set BasedeDatos = Workspaces(0).OpenDatabase(Archivo)
  Set Tabla = BasedeDatos.TableDefs("RF")
  Set Registro1 = BasedeDatos.OpenRecordset("RF")
  Registro1.MoveFirst
  'Para cada instante.
  For N = 1 To Tabla.RecordCount
    'Se da el valor que indicará la barra de progreso.
    FormEspera.ProgressBar.Value = N / Tabla.RecordCount * 100
    Registro1.Edit
    'Si se está en el intervalo en el que se desea destruir el mensaje.
    If N > Comienzo And N < Comienzo + Duracion Then
      'Se le da un valor constante que no depende de la señal transmitida.
      Registro1("Con Ruido") = 10
    Else
      'Se le agrega ruido con la intensidad deseada a la señal (el valor se genera aleatoriamente).
      Registro1("Con Ruido") = Registro1("Llegada") + 2 * Intensidad * (Rnd - 0.5)
    End If
    Registro1.Update
    Registro1.MoveNext
  Next
  BasedeDatos.Close
  Unload FormEspera
End Sub

```

5.3: Recepción

Para la recepción se implementa un sistema como el de la figura 5.21. El "Searcher" busca las tres componentes mas fuertes del multipath, utilizando para esto el Piloto, y le asigna una a cada correlador. Las salidas de estos correladores se suman coherentemente, debido a que se conocen los retardos.

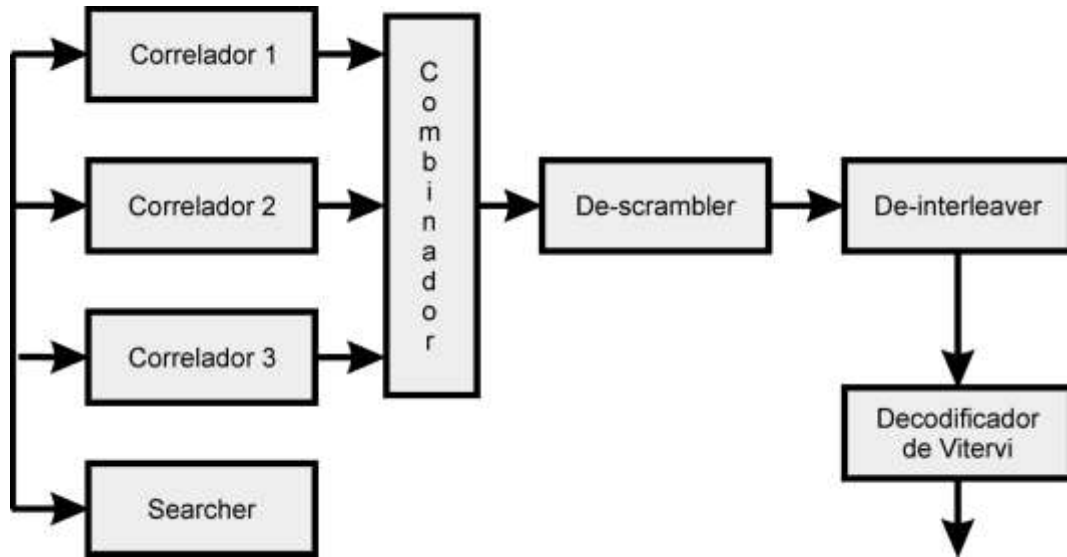


Fig. 5.21: Diagrama en bloques del Receptor.

Cada uno de estos correladores demodula una señal QPSK, funcionando como lo sugiere la figura 5.22, donde la fase ϕ_k corresponde a la fase asignada al correlador, $h(-t)$ es el filtro adaptado y las secuencias a_n y b_n son las que corresponden al Código Corto con el offset que identifica al sector.

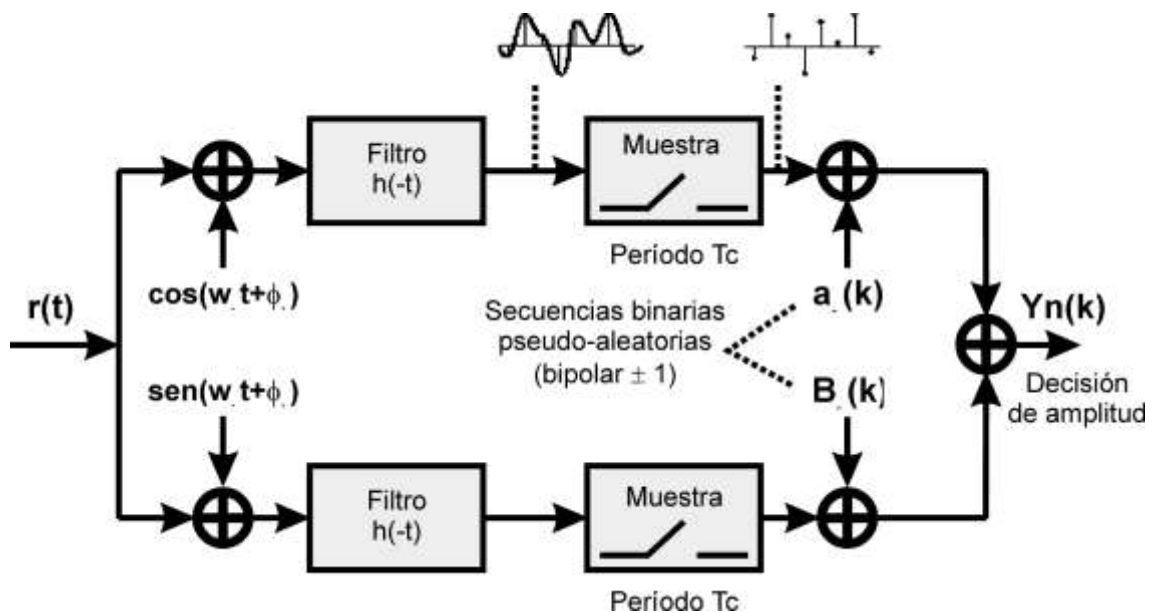


Fig. 5.22: Demodulador QPSK.

Las salidas son coherentemente combinadas, y se multiplican por el código de Walsh asignado al canal de tráfico, para separar la señal deseada del resto de los usuarios que comparten la portadora.

El receptor debe entonces realizar el proceso de "De-Scrambling", para lo cual simplemente multiplica (mediante una or exclusiva) el mensaje recibido por el Código

Largo. Para esto, el receptor ya conoce el estado del generador y su propio ESN (número de serie electrónico) con que fuera enmascarado en el transmisor.

El proceso de “De-Interleaving” es el reordenamiento de los bits para volverlos a su ubicación original.

El siguiente paso es la decodificación del mensaje, que fuera codificado en el transmisor mediante el codificador convolucional de la figura 3. Esto se logra implementando un decodificador de Viterbi. El principal problema es que el receptor desconoce la velocidad del frame original, por lo que debe decodificar para los cuatro casos, y elegir luego el más probable aplicando el criterio de máxima similitud.

Hay todavía un paso más antes de transformar el frame en una señal de voz, que es verificar la validez del mismo mediante el código cíclico que fuera agregado al mensaje. En caso de que el frame decodificado no sea válido, se desecha directamente. Cuando la cantidad de frames desechados es importante, se avisa a la celda, para que se aumente la potencia y se asegure una calidad de voz preestablecida.

El siguiente procedimiento simula la recepción de la señal que arriba, teniendo en cuenta los retardos en el canal, y realiza el filtrado. El resultado puede pensarse como la salida del filtro $h(-t)$ en la figura 21, con la diferencia de que ya se han sumado las tres componentes del multipath.

```
Public Sub Recibo()
Dim Retardo(2) As Integer      'retardos del 2º y 3º arribo.
Dim Intensidad(2) As Single   'intensidades del 2º y 3º arribo.
Dim Aire(193) As Single       'señal generada al combinar coherentemente los 3 arribos.
Dim AI As Single              'se usa para obtener la componente en fase, filtrada.
Dim AQ As Single              'se usa para obtener la componente en cuadratura, filtrada.
ChDir App.Path
Set BasedeDatos1 = Workspaces(0).OpenDatabase("Setup")
Set Tabla1 = BasedeDatos1.TableDefs("Canal")
'Se buscan los datos en la tabla "Canal".
Set Registro1 = BasedeDatos1.OpenRecordset("Canal")
Registro1.MoveFirst
'Se leen los valores para el segundo arribo.
Retardo(1) = Registro1("Retardo") * 39000
Intensidad(1) = Registro1("Intensidad") / 10
Registro1.MoveNext
'Se leen los valores para el tercer arribo.
Retardo(2) = Registro1("Retardo") * 39000
Intensidad(2) = Registro1("Intensidad") / 10
BasedeDatos1.Close
Set BasedeDatos1 = Workspaces(0).OpenDatabase(Archivo)
Set Tabla1 = BasedeDatos1.TableDefs("RF")
'Los registros "0", "1" y "2" se usan para leer en la tabla "RF".
Set Registro0 = BasedeDatos1.OpenRecordset("RF")
Set Registro1 = BasedeDatos1.OpenRecordset("RF")
Set Registro2 = BasedeDatos1.OpenRecordset("RF")
Set Tabla2 = BasedeDatos1.TableDefs("Recibido")
'El registro "3" se usa para escribir los resultados en la tabla "Recibido".
Set Registro3 = BasedeDatos1.OpenRecordset("Recibido")
'Cada registro se desplaza según el retardo que corresponde.
Registro0.MoveFirst
Registro1.MoveFirst
Registro1.Move Retardo(1)
Registro2.MoveFirst
```

```

Registro2.Move Retardo(2)
'Para los primeros 161 valores de señal recibidos.
For N = 0 To 160
  'Se combinan los 3 arribos en forma coherente.
  Aire(N) = Registro0("Con Ruido") + Registro1("Con ruido") / Intensidad(1) + Registro2("Con
ruido") / Intensidad(2)
  Registro0.MoveNext
  Registro1.MoveNext
  Registro2.MoveNext
Next
'Para recuperar cada bit.
For I = 1 To (Tabla1.RecordCount - 161) / 32
  'Se da el valor que indicará la barra de progreso.
  FormRecepcion.ProgressBar.Value = I / ((Tabla1.RecordCount - 161) / 32) * 50
  'Se inicializan las variables que acumularán el valor de cada componente.
  AI = 0
  AQ = 0
  'Para cada 32 muestras.
  For N = 161 To 192
    'Si un registro llegó al final de la tabla, se lo vuelve al principio.
    If Registro1.EOF Then
      Registro1.MoveFirst
    End If
    If Registro2.EOF Then
      Registro2.MoveFirst
    End If
    'Se combinan los 3 arribos en forma coherente.
    Aire(N) = Registro0("Con Ruido") + Registro1("Con Ruido") / Intensidad(1) + Registro2("Con
Ruido") / Intensidad(2)
    Registro0.MoveNext
    Registro1.MoveNext
    Registro2.MoveNext
  Next
  'Para cada valor combinado.
  For N = 0 To 192
    'Se acumula su producto con la forma del pulso, realizando así el filtrado por correlación.
    AI = AI + Aire(N) * Cos(Pi / 4 * N) * P(N - 96)
    AQ = AQ + Aire(N) * Sin(Pi / 4 * N) * P(N - 96)
  Next
  'Se guardan los valores en las columnas correspondientes de la tabla "Recibido"
  Registro3.AddNew
  Registro3("SampleI") = AI
  Registro3("SampleQ") = AQ
  Registro3.Update
  'Se desplazan los valores para recuperar el siguiente bit.
  For N = 0 To 160
    Aire(N) = Aire(N + 32)
  Next
Next
BasedeDatos1.Close
End Sub

```

El siguiente procedimiento recupera el canal elegido, mediante su código de Walsh. Elegido es el n° de canal a recuperar.

```

Public Sub Recuperar(Elegido As Integer)
Dim Walsh(63) As Boolean      'almacena los 64 bits del código de Walsh correspondiente.

```

```

Dim SCSR As New CShortCode 'registro de desplazamientos de clase CShortCode.
Dim Canal(2) As Integer 'n° de canal.
Dim SortI As Single 'componente en fase del código corto generado.
Dim SortQ As Single 'componente en cuadratura del código corto generado.
Dim Suma As Single 'acumula el producto de la señal recibida por el código de Walsh del
canal que se desea recuperar.
Dim ContI As Integer 'contador de ceros en la componente "I".
Dim ContQ As Integer 'contador de ceros en la componente "Q".
ChDir App.Path
Set BasedeDatos1 = Workspaces(0).OpenDatabase("Setup")
Set Tabla1 = BasedeDatos1.TableDefs("Mensaje")
Set Registro1 = BasedeDatos1.OpenRecordset("Mensaje")
Registro1.MoveFirst
'Se leen los canales transmitidos.
For N = 0 To 2
    Canal(N) = Registro1("Canal")
    Registro1.MoveNext
Next
BasedeDatos1.Close
Set BasedeDatos1 = Workspaces(0).OpenDatabase("Auxiliar")
Set Tabla1 = BasedeDatos1.TableDefs("Walsh")
'Se abre la tabla "Walsh" de la base de datos auxiliar.
Set Registro1 = BasedeDatos1.OpenRecordset("Walsh")
Registro1.MoveFirst
'Se busca el código de Walsh del canal que se va a recuperar.
Registro1.Move Canal(Elegido)
'Se lo almacena en la variable correspondiente.
For N = 0 To 63
    Walsh(N) = Registro1(N)
Next
BasedeDatos1.Close
Set BasedeDatos1 = Workspaces(0).OpenDatabase(Archivo)
Set Tabla1 = BasedeDatos1.TableDefs("Recibido")
'Se abre la tabla que contiene la señal recibida y filtrada.
Set Registro1 = BasedeDatos1.OpenRecordset("Recibido")
Set Tabla2 = BasedeDatos1.TableDefs("Recuperado" & Elegido)
'Se abre la tabla correspondiente al canal elegido para escribir el mensaje recuperado.
Set Registro2 = BasedeDatos1.OpenRecordset("Recuperado" & Elegido)
Registro1.MoveFirst
'Se inicializan los contadores de ceros.
ContI = 0
ContQ = 0
'Si se quiere recuperar el canal "2" y fué transmitido por el otro sector de la celda.
If Canal(1) = Canal(2) And Elegido = 2 Then
    'Se desplaza el registro que genera el "Código Corto" 'para generar los 512 chips de offset
necesarios.
    For N = 1 To 512
        SCSR.DesplazarI
        SCSR.DesplazarQ
    Next
End If
'Para recuperar cada bit.
For N = 1 To Tabla1.RecordCount / 64
    'Se da el valor que indicará la barra de progreso.
    FormRecepcion.ProgressBar.Value = 50 + N / (Tabla1.RecordCount / 64) * 10
    'Se inicializa el acumulador.
    Suma = 0
    'Para cada bit del Código de Walsh.

```



```

For I = 0 To 63
  'Se genera la componente "I" del código corto. Si hubo una corrida de 14 ceros seguidos, el
  proximo valor será otro cero. Sino el próximo valor es la salida del registro "SCSR" luego de un
  desplazamiento.
  If ContI = 14 Then
    SortI = 0
  Else
    SCSR.DesplazarI
    SortI = SCSR.SalidaI
  End If
  'Se genera la componente "Q" del código corto.
  If ContQ = 14 Then
    SortQ = 0
  Else
    SCSR.DesplazarQ
    SortQ = SCSR.SalidaQ
  End If
  'Se lleva la cuenta de los ceros para agregar otro después de una corrida de 14.
  If SCSR.SalidaI = False And ContI < 14 Then
    ContI = ContI + 1
  Else
    ContI = 0
  End If
  If SCSR.SalidaQ = False And ContQ < 14 Then
    ContQ = ContQ + 1
  Else
    ContQ = 0
  End If
  Registro1.Edit
  'Según el producto del código corto por el código de Walsh, se acumula la suma o la resta del
  mensaje recibido.
  If SortI Xor Walsh(I) Then
    Suma = Suma - Registro1("SampleI")
  Else
    Suma = Suma + Registro1("SampleI")
  End If
  If SortQ Xor Walsh(I) Then
    Suma = Suma - Registro1("SampleQ")
  Else
    Suma = Suma + Registro1("SampleQ")
  End If
  Registro1.MoveNext
Next
Registro2.AddNew
'Si la suma es positiva, se toma el bit como un "1".
Registro2("Decision") = Suma > 0
Registro2.Update
Next
Set SCSR = Nothing
BasedeDatos1.Close
End Sub

```

La clase CShortCode ya fue vista con anterioridad, en la sección “Modulación en cuadratura”.

El siguiente procedimiento genera el Código Largo que se usó para enmascarar el mensaje, y que ahora se necesita para recuperar el mensaje original.

Elección es el canal que se está recuperando.

```
Public Sub PNLC(Eleccion As Integer)
    Dim LCSR As New CLongCode      'registro de desplazamiento de clase CLongCode.
    Set BasedeDatos1 = Workspaces(0).OpenDatabase(Archivo)
    Set Tabla1 = BasedeDatos1.TableDefs("Recuperado" & Eleccion)
    'Se abre la tabla que se está usando para recuperar el mensaje.
    Set Registro1 = BasedeDatos1.OpenRecordset("Recuperado" & Eleccion)
    Registro1.MoveFirst
    'Se inicializa el registro, con el canal que se desea recuperar como parámetro.
    LCSR.Iniciar Eleccion
    'Para cada bit.
    For N = 1 To Tabla1.RecordCount
        'Se desplaza el registro.
        LCSR.Desplazar
        Registro1.Edit
        'El valor del bit de "Código Largo" es la salida del registro.
        Registro1("PNCode") = LCSR.Salida
        'Se multiplica por el bit del mensaje recuperado para poder descifrarlo.
        Registro1("DeScrambled") = Registro1("PNCode") Xor Registro1("Decision")
        Registro1.Update
        Registro1.MoveNext
    Next
    BasedeDatos1.Close
    Set LCSR = Nothing
End Sub
```

La clase CLongCode ya fue vista con anterioridad, en la sección “Scrambling”.

El siguiente procedimiento reordena los bits del mensaje, que fueran intercalados en el transmisor.

Elección es el canal que se está recuperando.

```
Public Sub Reordenar(Eleccion As Integer)
    Dim Marca As String           'guarda un valor de "Bookmark" para poder regresar a ese registro.
    Dim Aux As Boolean            'se usa como auxiliar para intercalar los valores.
    ChDir App.Path
    Set BasedeDatos1 = Workspaces(0).OpenDatabase("Auxiliar")
    Set Tabla1 = BasedeDatos1.TableDefs("Interleaver")
    'Se abre la tabla "Interleaver", que contiene el orden para reordenar los bits.
    Set Registro1 = BasedeDatos1.OpenRecordset("Interleaver")
    Set BasedeDatos2 = Workspaces(0).OpenDatabase(Archivo)
    Set Tabla2 = BasedeDatos2.TableDefs("Recuperado" & Eleccion)
    Set Registro2 = BasedeDatos2.OpenRecordset("Recuperado" & Eleccion)
    Registro2.MoveFirst
    Registro1.MoveFirst
    'Para cada bit.
    For N = 1 To 384
        'Se guarda el bit en la variable auxiliar
        Aux = Registro2("DeScrambled")
        'Se guarda el valor de Bookmark.
        Marca = Registro2.Bookmark
        'Se mueve el registro activo hasta el lugar indicado.
        Registro2.Move Registro1("Reordenar") - N
        Registro2.Edit
        'Se escribe el valor donde debe ir.
    Next
```

```

Registro2("Reordenado") = Aux
Registro2.Update
'Se vuelve al lugar original, para seguir por el próximo.
Registro2.Bookmark = Marca
Registro2.MoveNext
Registro1.MoveNext
Next
BasedeDatos1.Close
BasedeDatos2.Close
End Sub

```

El siguiente procedimiento decodifica el mensaje recibido, aplicando el algoritmo de decodificación convolucional de Viterbi.

Repeticion es la cantidad de veces que se repitió el mensaje, y Eleccion es el canal que se está decodificando. Las salidas "SCamino" y "SDistancia" son la secuencia decodificada y su distancia.

```

Public Sub Deco(Repeticion As Integer, Eleccion As Integer, SCamino As String, SDistancia As String)
Dim R As Integer           'contador.
Dim Señal0 As Boolean      'el primero de los dos bits de secuencia codificada que arribaron en el
                           'instante actual.
Dim Señal1 As Boolean      'el segundo bit de secuencia codificada en arribar.
Dim Dist0 As Integer       'se usa para calcular la distancia con la que se llega a cada estado al
                           'decodificar un cero.
Dim Dist1 As Integer       'se usa para calcular la distancia con la que se llega a cada estado al
                           'decodificar un uno.
Dim Aux As Integer         'variable auxiliar.
Dim Distancia(2, 256) As String 'guardará las distancias entre la secuencia recibida y la que se va
                           'guardando en cada estado, para el instante actual y el próximo.
Dim Camino(2, 256) As String 'guardarán las secuencias con menor distancia respecto de la recibida,
                           'para cada estado y para el instante actual y el próximo.
Dim Actual As Integer      'estas variables indican cual es la columna que corresponde
Dim Proximo As Integer     'al instante actual y cual al próximo en las matrices de las secuencias
                           'y las distancias.
Dim Cam(8) As String       'guarda las secuencias decodificadas para cada repetición.
Dim Dist(8) As Integer     'guarda las distancias para cada secuencia decodificada.
Dim Iguales(8) As Integer  'guarda la cantidad de secuencias coincidentes que se detectaron.
Dim D(8) As Integer        'contiene la suma de las distancias de las secuencias coincidentes.
Dim Maximo As Integer      'máxima cantidad de coincidencias entre las secuencias.
Dim Minimo As Integer      'mínima distancia para un grupo de secuencias coincidentes.
Set BasedeDatos1 = Workspaces(0).OpenDatabase(Archivo)
Set Tabla1 = BasedeDatos1.TableDefs("Recuperado" & Eleccion)
Set Registro1 = BasedeDatos1.OpenRecordset("Recuperado" & Eleccion)
ChDir App.Path
Set BasedeDatos2 = Workspaces(0).OpenDatabase("Auxiliar")
Set Tabla2 = BasedeDatos2.TableDefs("Convolucional")
'Se abre la tabla "Convolucional", que se necesita para decodificar el Frame.
Set Registro2 = BasedeDatos2.OpenRecordset("Convolucional")
'Inicializa los valores.
Actual = 0
Proximo = 1
'Para cada repetición del mensaje.
For R = 1 To Repeticion
'Se inicializan las variables que guardarán las secuencias.
For N = 0 To 255
Camino(0, N) = ""
Camino(1, N) = ""

```

```

Next
'Se inicializan las variables que guardarán las distancias. Se les da un valor alto para que se acepte la
primer secuencia con la que se llegue a ese estado como próximo.
Distancia(0, 0) = 0
Distancia(1, 0) = 100
For I = 1 To 255
  Distancia(0, I) = 100
  Distancia(1, I) = 100
Next
Registro1.MoveFirst
'Se empieza por el primer bit de la repetición.
Registro1.Move R - 1
'Para cada bit de la repetición.
For I = 1 To 192 / Repeticion
  'Se lee el primer bit.
  Señal0 = Registro1("Reordenado")
  'Se busca el siguiente bit que corresponde a la repetición.
  Registro1.Move Repeticion
  'Se lee el segundo bit.
  Señal1 = Registro1("Reordenado")
  Registro2.MoveFirst
  'Para cada estado.
  For N = 0 To 255
    'Si es un estado posible (hay una secuencia para llegar hasta él).
    If Distancia(Actual, N) < 100 Then
      'Se les dá el valor de distancia actual a las variables que calcularán la distancia para los
      próximos estados.
      Dist0 = Distancia(Actual, N)
      Dist1 = Distancia(Actual, N)
      'Si el bit recibido es distinto al que debería haberse transmitido en primer lugar si el bit
      original era un uno.
      If Señal0 <> Registro2("Salida 0 true") Then
        'Se incrementa en 1 la distancia para el próximo estado.
        Dist0 = Dist0 + 1
      End If
      'Si el bit recibido es distinto al que debería haberse transmitido en primer lugar si el bit
      original era un cero.
      If Señal0 <> Registro2("Salida 0 false") Then
        'Se incrementa en 1 la distancia para el próximo estado.
        Dist1 = Dist1 + 1
      End If
      'Si el bit recibido es distinto al que debería haberse transmitido en segundo lugar si el bit
      original era un uno.
      If Señal1 <> Registro2("Salida 1 true") Then
        'Se incrementa en 1 la distancia para el próximo estado.
        Dist0 = Dist0 + 1
      End If
      'Si el bit recibido es distinto al que debería haberse transmitido en segundo lugar si el bit
      original era un cero.
      If Señal1 <> Registro2("Salida 1 false") Then
        'Se incrementa en 1 la distancia para el próximo estado.
        Dist1 = Dist1 + 1
      End If
      'Si, suponiendo decodificar un cero, la distancia calculada para el próximo estado es menor
      que la ya calculada con otra secuencia.
      If Dist0 < Distancia(Proximo, Registro2("Siguiete true")) Then
        'Se guarda la distancia para el próximo estado.
        Distancia(Proximo, Registro2("Siguiete true")) = Dist0

```

```

        'Se guarda la secuencia por la que se llega al estado.
        Camino(Proximo, Registro2("Siguiete true")) = Camino(Actual, N) & "1"
    End If
    'Si, suponiendo decodificar un uno, la distancia calculada para el próximo estado es menor
    que la ya calculada con otra secuencia.
    If Dist1 < Distancia(Proximo, Registro2("Siguiete false")) Then
        'Se guarda la distancia para el próximo estado.
        Distancia(Proximo, Registro2("Siguiete false")) = Dist1
        'Se guarda la secuencia por la que se llega al estado.
        Camino(Proximo, Registro2("Siguiete false")) = Camino(Actual, N) & "0"
    End If
    'Se pone un valor alto en la distancia actual, porque la columna será luego utilizada como
    distancia para el próximo instante.
    Distancia(Actual, N) = 100
    End If
    'Se pasa al siguiente estado.
    Registro2.MoveNext
Next N
'Se intercambian los valores de "Proximo" y "Actual", utilizando la variable auxiliar.
Aux = Actual
Actual = Proximo
Proximo = Aux
'Se busca el siguiente bit correspondiente a la repetición.
Registro1.Move Repeticion
Next I
'Se vuelve al primer estado.
Registro2.MoveFirst
'Se guarda la secuencia decodificada (se sabe que el último estado es el cero).
Cam(R) = Camino(Actual, 0)
'Se guarda la distancia correspondiente.
Dist(R) = Distancia(Actual, 0)
Next R
'Para cada repetición, se inicializan las variables.
For R = 1 To Repeticion
    Iguales(R) = 0
    D(R) = 0
Next
'Para cada repetición, se compara la secuencia con las demás.
For R = 1 To Repeticion
    For N = 1 To Repeticion
        'Si coinciden.
        If Cam(R) = Cam(N) Then
            'Se incrementa el contador de coincidencias correspondiente.
            Iguales(R) = Iguales(R) + 1
            'Se suman las distancias.
            D(R) = D(R) + Dist(N)
        End If
    Next N
Next R
'Se busca la secuencia que tenga un máximo de coincidencias.
Maximo = Iguales(1)
SCamino = Cam(1)
Minimo = D(1)
For R = 2 To Repeticion
    If Maximo < Iguales(R) Then
        Maximo = Iguales(R)
        SCamino = Cam(R)
        Minimo = D(R)
    End If
Next R

```

```

End If
Next
'Para cada repetición.
For R = 1 To Repeticion
'Si la secuencia tiene el máximo de coincidencias y la mínima distancia, será la secuencia
decodificada.
If Iguales(R) = Maximo And D(R) < Minimo Then
SCamino = Cam(R)
Minimo = D(R)
End If
Next
'La distancia será la suma de las distancias para cada repetición.
SDistancia = 0
For R = 1 To Repeticion
SDistancia = SDistancia + Dist(R)
Next
Basedatos1.Close
Basedatos2.Close
End Sub

```

El procedimiento anterior realiza la decodificación para una determinada velocidad del frame, que se pasa como parámetro. El código que llama a este procedimiento para los 4 casos posibles y luego elige el más probable puede verse a continuación.

```

'Para cada velocidad posible.
For Vel = 0 To 3
'Se da el valor que indicará la barra de progreso.
ProgressBar.Value = 60 + 10 * Vel
'Se llama al procedimiento "Deco", que decodifica el frame recibido.
Deco 2 ^ Vel, Canal, Cam(Vel), Dist(Vel)
Next
'Compara las distancias para elegir el camino mas probable.
Menor = Dist(0)
Camino = Cam(0)
For Vel = 1 To 3
If Dist(Vel) < Menor Then
Menor = Dist(Vel)
Camino = Cam(Vel)
End If
Next
'Lo guarda en la base de datos
Guardar Camino, Canal

```

El siguiente procedimiento verifica el mensaje calculando el síndrome. Las salidas son Válido, que indica la validez del Frame, y CRCbits, que es la cantidad de bits de código CRC. Eleccion es el n° del canal que se decodificó.

```

Public Sub Verificar(Valido As Boolean, CRCbits As Integer, Eleccion As Integer)
Dim Reg As Object 'registro usado para decodificar.
Dim Reali As Boolean 'valor que se realimenta en el registro CRC.
Set Basedatos1 = Workspaces(0).OpenDatabase(Archivo)
Set Tabla1 = Basedatos1.TableDefs("Decodificado" & Eleccion)
Set Registro1 = Basedatos1.OpenRecordset("Decodificado" & Eleccion)
'Según la cantidad de bits del Frame, inicializo el registro y la cantidad de bits que corresponden al
código CRC.
Select Case Tabla1.RecordCount

```

```
Case 192
  Set Reg = New CRegistro9600: CRCbits = 12
Case 96
  Set Reg = New CRegistro4800: CRCbits = 8
Case Else
  'En estos casos no hubo codificación CRC.
  CRCbits = 0: Valido = True: Exit Sub
End Select
Registro1.MoveFirst
'Para cada bit de mensaje (excepto los tail bits).
For N = 1 To Tabla1.RecordCount - 8
  'Desplazo el mensaje por el registro para obtener el síndrome.
  Reali = Reg.Estado(CRCbits - 1)
  Reg.Desplazar Reali, Registro1("Frame") Xor Reali
  Registro1.MoveNext
Next
'Si el síndrome es cero.
If Reg.Estados = 0 Then
  'El frame recibido es válido.
  Valido = True
Else
  Valido = False
End If
Set Reg = Nothing
BasedeDatos1.Close
End Sub
```

Las clases CRegistro9600 y Cregistro4800 ya se vieron anteriormente, en la sección “Codificación CRC”.

El código completo del programa puede encontrarse en el directorio “Proyecto” del CD.

Capítulo VI: Funcionamiento del Canal Inverso

Se denomina Canal Inverso al vínculo de RF por el cual se transmite el mensaje desde el teléfono móvil a la celda. La generación de los canales de tráfico inversos es diferente en varios aspectos a la generación de los canales directos.

El teléfono móvil transmite a una frecuencia menor a la que transmite la estación de base, esta diferencia es de 45 MHz en el servicio celular, y 80 MHz en el servicio PCS. En el Canal Inverso, cada móvil transmite usando un código único que lo distingue de los demás usuarios.

El diagrama en bloques del transmisor puede verse en la figura 6.1.

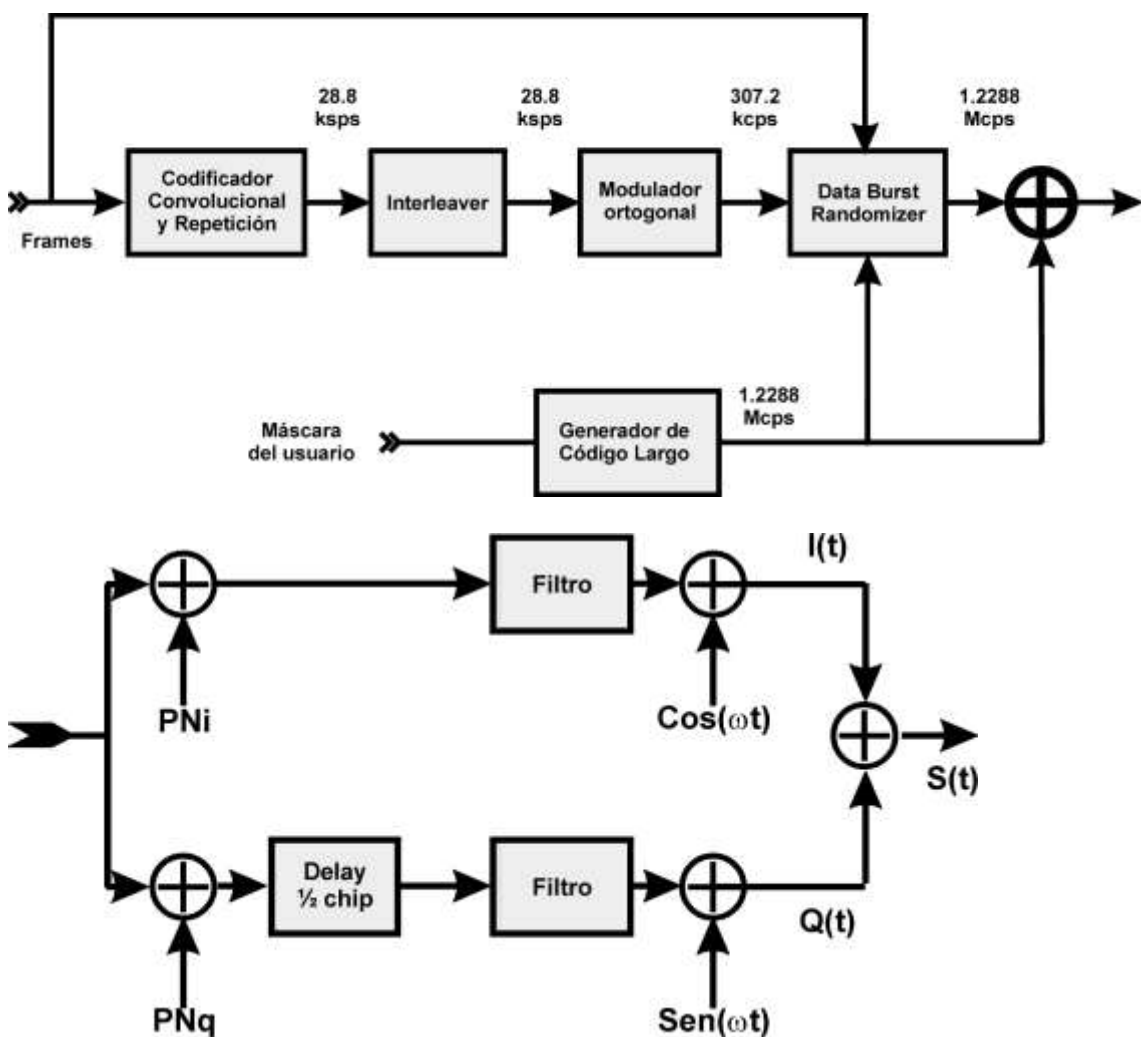


Fig. 6.1: Generación de un Canal Inverso.

6.1: Generación de los frames

Los Frames tienen la misma forma que los utilizados en los canales de tráfico directos, con la misma codificación CRC y los Tail Bits, soportando también ambos tipos de Vocoders.

Full (9600 bps)	172 bits	12 bits CRC	8 Tail bits
1/2 (4800 bps)	80 bits	8 bits CRC	8 Tail bits
1/4 (2400 bps)	40 bits		8 Tail bits
1/8 (1200 bps)	16 bits		8 Tail bits

Fig.6.2a: Frames generados por un Vocoder de tipo 1.

Full (14400 bps)	268 bits	12 bits CRC	8 Tail bits
1/2 (7200 bps)	126 bits	10 bits CRC	8 Tail bits
1/4 (3600 bps)	56 bits	8 bits CRC	8 Tail bits
1/8 (1800 bps)	22 bits	8 bits CRC	8 Tail bits

Fig.6.2b: Frames generados por un Vocoder de tipo 2.

6.2: Codificación CRC

La codificación se realiza de la misma forma que en el canal directo, siendo los polinomios generadores los ya vistos anteriormente.

$$G(x) = x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^4 + x + 1$$

$$G(x) = x^8 + x^7 + x^4 + x^3 + x + 1$$

(ecuación 6.1)

6.3: Codificación convolucional

La codificación convolucional se realiza con un Nivel de Memoria $K = 9$, en el caso de utilizarse un vocoder de tipo 1 (9600 bps), el codificador tiene una velocidad 1/3 (3 bits de salida para cada entrada) como puede verse en la figura 6.4, mientras que si se utiliza un vocoder de tipo 2 (14400 bps), el codificador convolucional es el mismo que el usado para el canal directo, como puede verse en la figura 6.3.

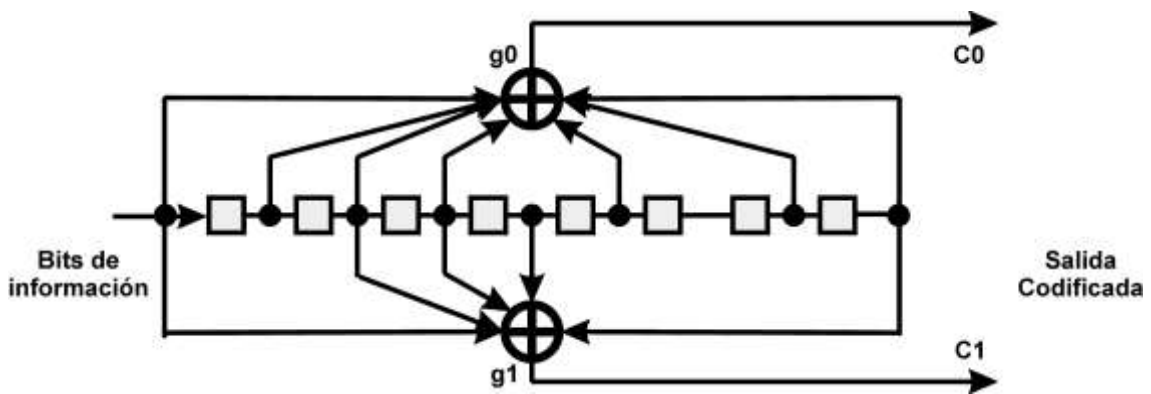


Fig. 6.3: Codificador convolucional de velocidad 1/2.

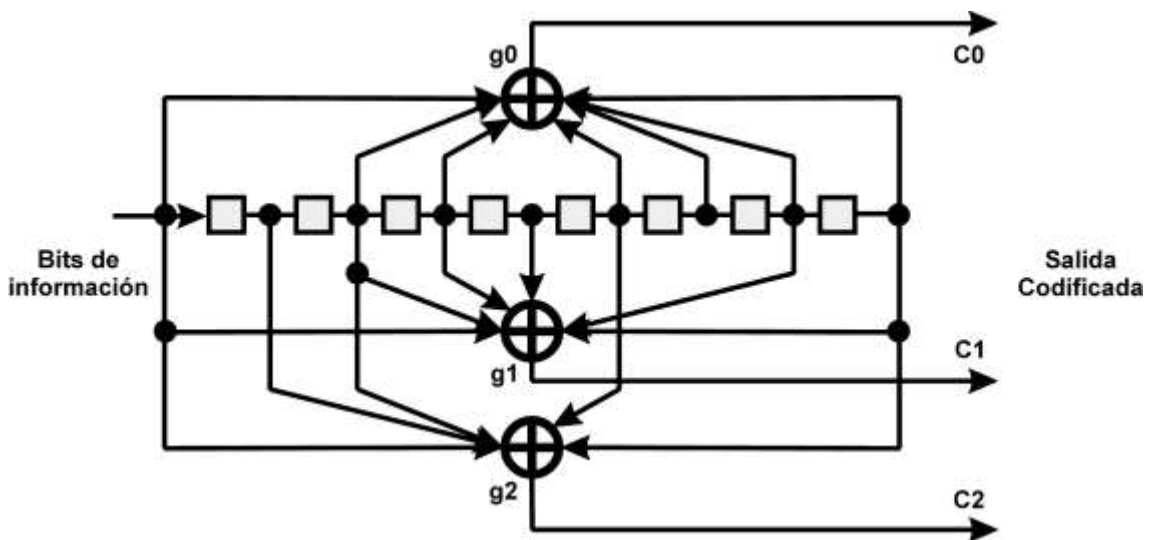


Fig. 6.4: Codificador convolucional de velocidad 1/3.

Los bits codificados son repetidos luego, en un proceso similar al efectuado en el Canal Directo, dando como resultado una velocidad constante (28800 bps), independientemente de la velocidad del Frame. Es de notar que, debido a los codificadores utilizados, esta velocidad es también independiente del tipo de Vocoder que se utiliza, no necesitándose descartar algunos bits como se hace en el Canal Directo. El proceso de repetición se realiza por conveniencia, pero los símbolos repetidos no serán transmitidos. El llamado "Data burst randomizer", seleccionará luego una copia de cada símbolo para transmitirlo, y apagará el transmisor para los símbolos redundantes.

6.4: Interleaving

El proceso de intreleaving es similar al que se usa para el Canal Directo, realizando el intercambio entre símbolos que corresponden a un mismo Frame. La diferencia radica en el arreglo usado para desordenar los símbolos, porque a una velocidad de 28800 bps,

hay a esta altura 576 símbolos por cada Frame de 20 mseg. que se transmite, en lugar de los 384 que hay cuando la velocidad es de 19200 bps.

6.5: Modulación ortogonal

Las estaciones de base deben demodular la transmisión del móvil en forma no coherente. Para mejorar esta demodulación, se utiliza un esquema de modulación ortogonal, utilizando un juego de señales ortogonales entre sí para transmitirlos en lugar de los símbolos codificados en formato bipolar. Para esto, la duración de la señal debe ser tan grande como sea posible, pero dentro del tiempo de coherencia del canal (tiempo durante el cual el canal es relativamente estable). Para este propósito se eligió usar los códigos de Walsh que, a diferencia del canal directo donde proveen aislamiento entre usuarios, proveerán aislamiento entre símbolos.

El juego de señales ortogonales contiene 64 señales posibles. La información a ser modulada se divide en grupos de 6 símbolos. Estos 6 símbolos corresponden en formato binario a un valor entre 0 y 63 ($2^6 = 64$), utilizándose entonces para seleccionar el código de Walsh que será transmitido. Esto incrementa el ancho de banda a 307.2 KHz ($28.8 \text{ k} * 64 / 6 = 307.2 \text{ k}$).

6.6: Data Burst Randomizer

Para aprovechar los períodos de baja actividad de voz, el vocoder reduce su velocidad permitiendo la transmisión de la señal con una menor potencia promedio. En un Canal de Tráfico Directo, esto se logra repitiendo los símbolos y transmitiéndolos a una menor potencia. La desventaja de este método es que dispersa la energía en el tiempo, haciendo más lento el proceso de recolectarla en el receptor. Esta es la razón de que no pueda utilizarse en el Canal Inverso, ya que se implementa un control de potencia muy rápido que es incompatible con este retardo.

En el Canal de Tráfico Inverso, el teléfono móvil no transmite la información redundante, ya que cuando esta es preproducida por el repetidor de símbolos, el "Data Burst Randomizer" apaga el transmisor, reduciendo la potencia promedio transmitida. El apagado de transmisor se realiza eligiendo cual copia se transmitirá en forma pseudo-aleatoria.

6.7: Separación de usuarios

El Canal Inverso, a diferencia de lo que sucede en el Canal Directo, no utiliza códigos estrictamente ortogonales para separar a los usuarios en canales lógicos. Lo que se utiliza son distintas fases de un código pseudo-aleatorio con un período muy grande, que da como resultado una correlación aceptablemente pequeña, aunque no sea cero.

Todos los canales en el vínculo inverso se aíslan entre sí usando el Código Largo. Debido a los millones de posibles offsets de este código (todas las $2^{42} - 1$ posibles fases están disponibles como canales lógicos), es posible asignar uno fijo a cada usuario, usando como máscara del Código Largo a la misma que se utiliza en el Canal Directo.

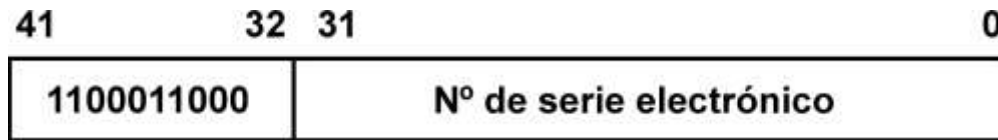


Fig. 6.5: Máscara del Código Largo para un Canal de Trafico Inverso.

Esto facilita el proceso de handoff, ya que los móviles no deben cambiar nada en su proceso de codificación y modulación al pasar a otra célula.

La señal es multiplicada por el Código Largo, aumentando así su ancho de banda de 307.2KHz a 1.23 MHz. Esto se realiza multiplicando cada chip del código de Walsh generado por la modulación ortogonal por 4 chips de Código Largo.

El período de este Código Largo es de 3.6 millones de segundos, algo mas de 41 días.

Aún cuando hay un gran número de fases posibles para el Código Largo, hay una ambigüedad potencial entre unidades que tengan máscaras similares, que producen offsets pequeños entre los diferentes móviles.

Esto hace que no pueda distinguirse un offset de x microsegundos de un retardo de x microsegundos (de un móvil que se encuentre más alejado de la antena). Este es el motivo por el cual se agrega el Código Corto.

La combinación del Código Corto con el Código Largo da por resultado una "supersecuencia", que será también periódica pero con un período más largo ($2^{42} * 2^{15} = 2^{57}$, aproximadamente 3700 años). El resultado es que para cualquier offset de la secuencia del Código Largo, al agregarle el Código Corto, se obtiene una fase diferente de la "supersecuencia". Así, estaremos usando 2^{42} posibles fases de esta "supersecuencia" de período 2^{57} , con lo que se consigue mejorar la ambigüedad de tiempo de 1 chip a 2^{15} chips de retardo.

Al igual que en el Canal Directo, se utilizan dos secuencias (**I** y **Q**) para generar una modulación en cuadratura.

El Código Corto es el mismo utilizado para separar los sectores en el Canal Directo, pero en el Canal Inverso se utiliza sin ningún offset.

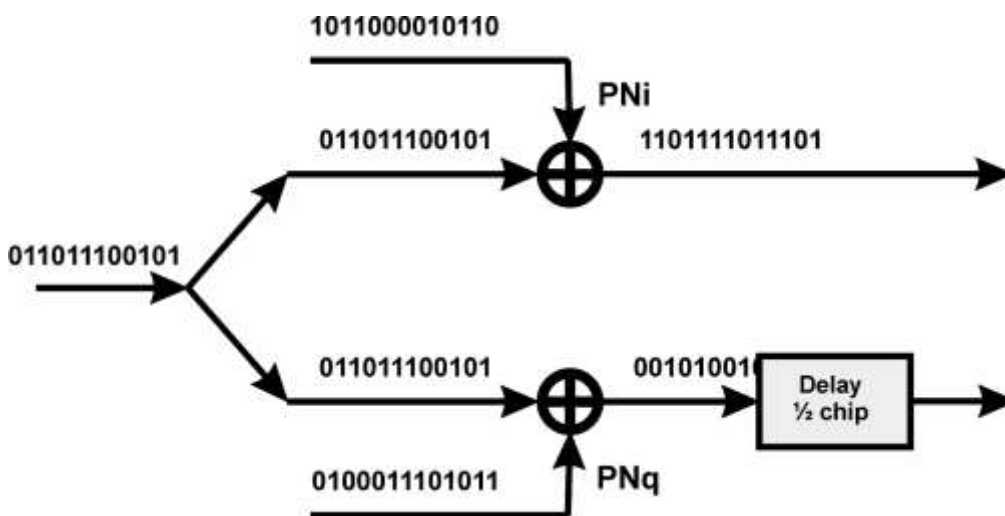


Fig. 6.6: Modulación del Canal Inverso.

6.8: Modulación de RF

La diferencia con el Canal Directo es que la componente **Q** se retrasa $\frac{1}{2}$ chip, produciendo offset QPSK. Esto se hace para reducir los requerimientos del amplificador de potencia de las unidades móviles.

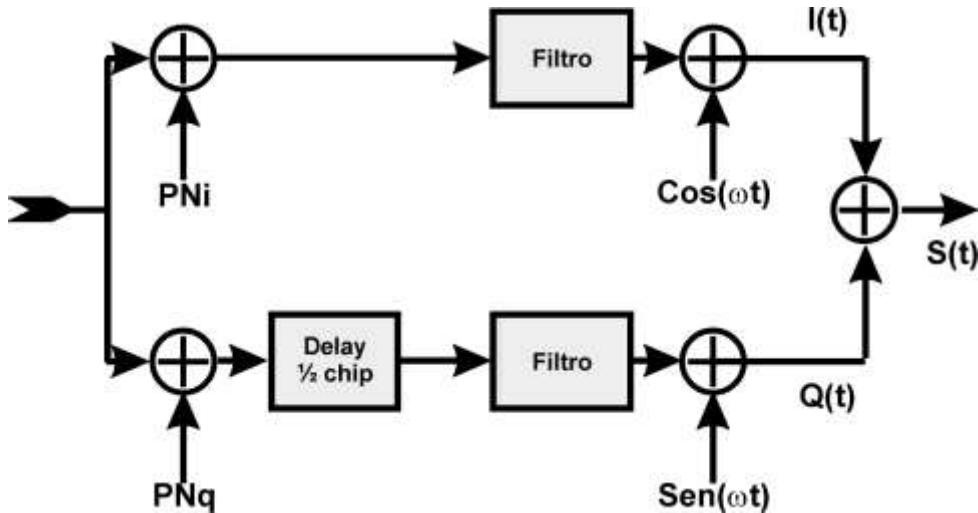


Fig. 6.7: Modulación Offset QPSK.

6.9: Generación del Canal de Acceso

El canal de Acceso se genera de la misma forma que un Canal de Tráfico Inverso con una excepción: el “Data Burst Randomizer” no se utiliza. Esto se debe a que no hay actividad de voz en el Canal de Acceso, que transmite a 4800 bps. Cada frame dura 20 mseg. y contiene 88 bits de información y 8 tail bits.

El generador de Código Largo es enmascarado como puede verse en la figura 6.8.

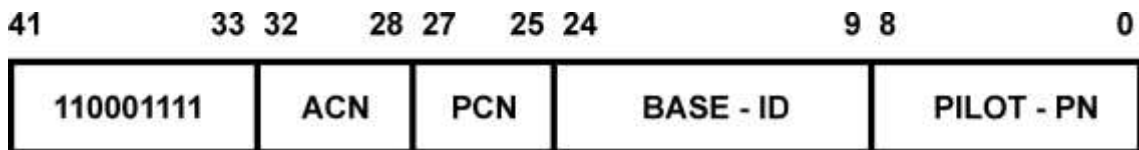


Fig. 6.8: Máscara del Canal de Acceso.

En donde:

ACN: Número de Canal de Acceso.

PCN: Número de Canal de Paging.

BASE – ID: Identificación de la Estación de Base.

PILOT – PN: PN offset del Canal Directo.

Este canal es el vehículo para comunicarse cuando los móviles no están asignados a un canal de tráfico. Su propósito es comunicar el origen de llamadas desde el móvil y responder a los mensajes de Paging. Un acceso exitoso es seguido por la asignación de un canal de tráfico.

6.10: Recepción

Para la recepción se implementa un sistema como el que se puede observar en la figura 8. La estación de base también implementa un receptor Rake, en este caso con 4 elementos (Fingers). La función de búsqueda está distribuida en los 4 Fingers, identificando las componentes de multipath más fuertes y demodulándolas con el offset indicado. Los correladores utilizan una OR exclusiva con las secuencias con que fuera modulado (el Código Corto y el Código Largo) para separar la señal deseada del resto de los usuarios que comparten la misma frecuencia, y realizan una FHT (Fast Hadamard Transform) para detectar los símbolos que fueran modulados ortogonalmente mediante los códigos de Walsh.

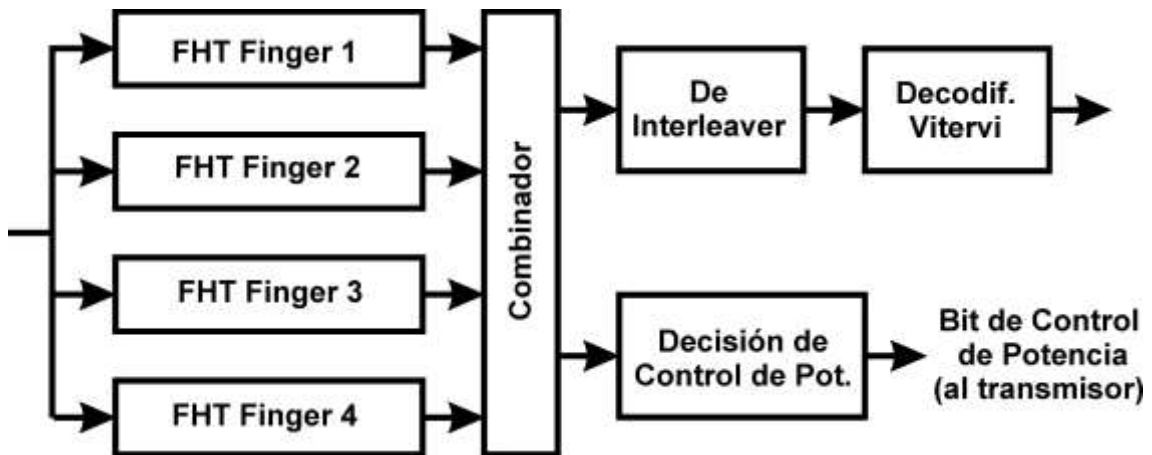


Fig. 6.9: Diagrama en bloques del Receptor.

Las salidas son combinadas en forma incoherente, y luego se realiza el proceso de “De-Interleaving” y la decodificación convolucional, implementando un decodificador de Viterbi.

También se mide la calidad de la señal recibida, para decidir si se envía al móvil una orden de aumentar o de disminuir su potencia.

Capítulo VII: Manual de operación

El programa realiza una simulación de la transmisión de mensajes (voz o datos) desde la Estación Base hasta el Teléfono Móvil, para el sistema de Telefonía Celular CDMA (Code Division Multiple Access).

Para esto se simula primero la transmisión de tres mensajes diferentes, que son previamente elegidos por el usuario, por tres canales distintos. Luego se simulan las características del canal, también elegidas por el usuario, como atenuaciones, retardos y ruidos. Finalmente se puede seleccionar cual de los mensajes se desea recibir y se lo decodifica.

7.1: Como realizar la simulación

Para realizar la simulación, primero debe seleccionarse Nuevo en el Menú Archivo. De esta manera ya se pueden elegir los mensajes a transmitir, para lo cual debe seleccionar Mensaje en el Menú Transmisor, aquí veremos el Formulario Mensajes, en el cual podemos introducir los mensajes a enviar por los tres canales.

Al seleccionar Mensaje en el menú Transmisor (o presionando Ctrl+M), aparece el formulario que puede verse en la figura 7.1.

Mensajes

Canal 1 | Canal 2 | Canal 3

Mensaje para el canal 1

B8944C0CDB06DC5FD0F5

Mensaje a transmitir por el canal (1 Frame), en hexadecimal

Velocidad del frame 4800

Canal a utilizarse 9

Nº de serie electrónico del teléfono (32 bits):
0000011100000000
0000011001011100

Fig. 7.1: Formulario mensajes.

Las tres fichas corresponden a los tres canales que se simularán transmitir, debiendo introducir los siguientes datos:

El mensaje propiamente dicho, que se introducirá en forma hexadecimal y cuyo tamaño dependerá de la velocidad que se haya seleccionado.

La velocidad del Frame, que se selecciona de una lista (9600, 4800, 2400, 1200).

El canal que se usará para el mensaje, número entre 0 y 64 que se selecciona de una lista y corresponde al código de Walsh que se utilizará.

El N° de serie electrónico (ESN) del teléfono móvil que recibirá el mensaje.

El botón Random completa aleatoriamente tanto el mensaje como el N° de serie, en el caso de que estén vacíos o a medio completar. Esto también se realiza en forma automática al presionar el botón Aceptar.

Los datos sólo se actualizarán una vez que se presione el botón Aceptar, por lo que para deshacer los cambios basta con presionar Cancelar o Esc.

Una vez hecho esto se puede simular la transmisión, seleccionando Transmitir en el menú Transmisor. De esta manera se realiza todo el proceso efectuado por la Estación Base con cada mensaje hasta la salida al aire del mismo. Mientras se realiza la simulación, aparece una ventana indicando el progreso de la misma, ya que la simulación es lenta y tarda unos cuantos minutos.

Una vez simulada la transmisión, se pueden elegir las características del Canal de Transmisión, seleccionando Características en el Menú Canal. Aquí aparecerá un formulario con el título Características del canal de transmisión, en donde pueden elegirse los parámetros del canal con los cuales se simularán los Retardos y el Ruido, que se encuentran también en el Menú Canal.

Al seleccionar la opción Características en el Menú Canal, aparece un formulario con tres fichas.

The image shows a software dialog box titled "Características del canal de transmisión". It features three tabs: "Primer Rebote", "Segundo Rebote", and "Ruido". The "Primer Rebote" tab is currently active. Below the tabs, the text "Seleccione los siguientes parámetros:" is displayed. There are two input fields: "Retardo en microsegundos" with a dropdown menu showing the value "3", and "Intensidad del rebote" with a dropdown menu showing the value "0.8". At the bottom of the dialog, there are three buttons: "Aceptar", "Cancelar", and "Ayuda".

Fig. 7.2: Formulario características del canal. Fichas correspondientes al multipath.

Las dos primeras fichas, que tienen el aspecto que se ve en la figura 7.2, corresponden a las características del canal con respecto a los retardos, en las cuales se pueden seleccionar la intensidad y el retardo de tiempo de los dos rebotes más fuertes, con respecto al primero en arribar.



The image shows a software dialog box titled "Características del canal de transmisión" (Transmission Channel Characteristics). It has three tabs: "Primer Rebote", "Segundo Rebote", and "Ruido". The "Ruido" tab is selected. The dialog contains the following elements:

- A header: "Seleccione los siguientes parámetros:" (Select the following parameters:)
- A dropdown menu for "Nivel del ruido blanco" (White noise level) with the value "3" selected.
- A dropdown menu for "Duración del pulso de ruido" (Noise pulse duration) with the value "2" selected.
- A slider control for "Comienzo del pulso de ruido" (Noise pulse start) with a small vertical bar indicating the start position.
- Three buttons at the bottom: "Aceptar" (Accept), "Cancelar" (Cancel), and "Ayuda" (Help).

Fig. 7.3: Formulario características del canal. Ficha correspondiente al ruido.

En la última ficha, que puede verse en la figura 7.3, se seleccionan las características del ruido en el canal. El "nivel de ruido blanco" es el valor pico de ruido aleatorio que se suma a la señal transmitida, el valor que se selecciona es en veces respecto de la señal. El pulso de ruido, del que se eligen la duración y el instante en el que comienza, se usa para destruir una parte del frame transmitido, permitiendo ver el funcionamiento de la corrección de errores en el sistema. Tanto para la duración como para el comienzo, el valor seleccionado es en milisegundos.

Por último, una vez simulada la transmisión y los efectos del canal sobre la señal transmitida, podemos recibir cualquiera de los mensajes que hemos transmitido. Para esto debemos seleccionar el Menú Receptor, donde podremos elegir el canal que deseamos recibir. Aquí el programa realiza la recepción y decodificación de la misma forma que lo hace el teléfono móvil y presenta el mensaje recuperado.

7.2: Como ver los resultados

El programa genera, como resultado de cada simulación, una base de datos donde puede verse todo el proceso. Para poder abrir esta base de datos se necesita tener instalado Microsoft Access, u otro programa similar.

Abriendo el archivo generado, pueden verse a continuación las tablas existentes, como lo muestra la figura 7.4.

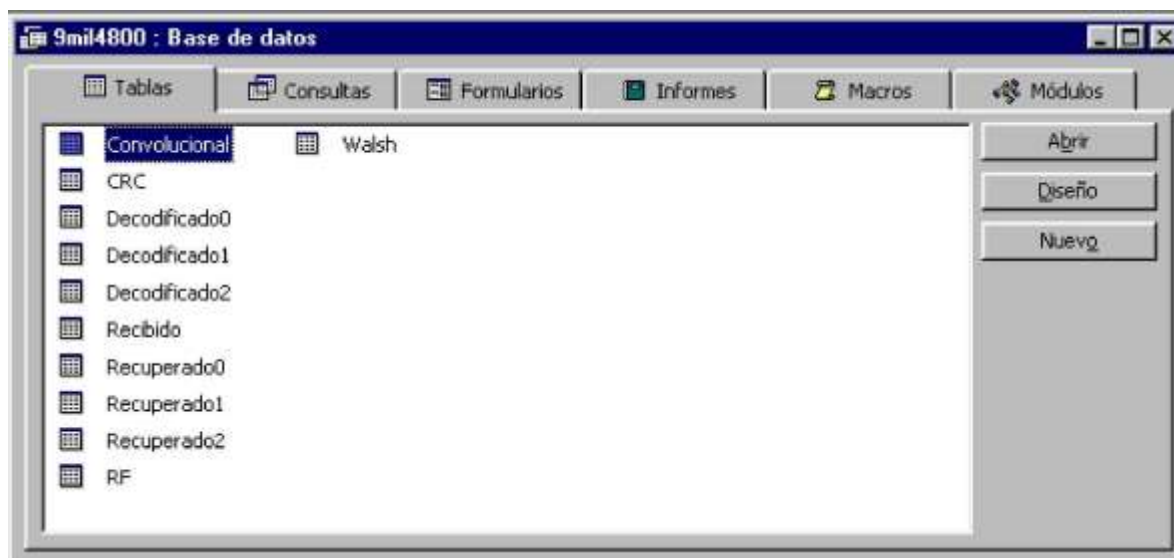


Figura 7.4: Tablas de la base de datos, vistas con Access.

7.2.1: Tabla “CRC”

La tabla “CRC” contiene las 6 columnas que pueden verse en la figura 7.5, y la cantidad de registros dependerá de la longitud de los mensajes que se han simulado.

Mensaje0	Mensaje1	Mensaje2	Codificado0	Codificado1	Codificado2
0	-1	0	0	-1	0
-1	-1	0	-1	-1	0
-1	-1	-1	-1	-1	-1
-1	-1	0	-1	-1	0
0	-1	0	0	-1	0
0	-1	0	0	-1	0
0	0	-1	0	0	-1
-1	-1	0	-1	-1	0
0	-1	-1	0	-1	-1
0	0	-1	0	0	-1

Fig. 7.5: Tabla CRC

Las columnas “Mensaje0”, “Mensaje1” y “Mensaje2”, contienen los bits correspondientes a los mensajes cuya transmisión se ha simulado. Cada columna contiene, en formato binario, el mensaje que se ingresó en forma hexadecimal para simular la transmisión.

Las columnas “Codificado0”, “Codificado1” y “Codificado2”, contienen los 3 Frames que se simuló transmitir. Cada Frame está compuesto del mensaje propiamente dicho, los bits de código CRC (si es necesario) y 8 ceros.

7.2.2: Tabla “Convolutional”

La tabla “Convolutional” contiene las 12 columnas que pueden verse en las figuras 7.6a y 7.6b. La cantidad de registros es siempre de 384, ya que junto con la codificación convolutional se realiza la repetición.

	Codificado 0	Codificado 1	Codificado 2	Intercalado 0	Intercalado 1	Intercalado 2
	-1	0	0	-1	0	0
	-1	0	0	-1	0	-1
	-1	0	0	-1	-1	-1
	-1	0	0	-1	0	0
	-1	-1	0	-1	0	0
	-1	-1	0	-1	-1	-1
	0	-1	0	0	0	0
	0	-1	0	-1	0	-1

Fig. 7.6a: Tabla Convolutional

	PNCode 0	PNCode 1	PNCode 2	Scrambled 0	Scrambled 1	Scrambled 2
	0	0	0	-1	0	0
	0	0	0	-1	0	-1
	-1	0	0	0	-1	-1
	-1	-1	0	0	-1	0
	-1	0	0	0	0	0
	-1	-1	-1	0	0	0
	0	0	0	0	0	0
	-1	-1	0	0	-1	-1

Fig. 7.6b: Tabla Convolutional

Las columnas “Codificado” contienen los mensajes después de ser codificados convolutionalmente.

Las columnas “Intercalado” contienen los mensajes una vez intercalados. Este proceso, llamado Interleaving, se realiza para mejorar el desempeño de la codificación convolutional.

Las columnas “PNCode” contienen las secuencias de Código Largo utilizadas para multiplicar cada mensaje.

Las columnas “Scrambled” contienen el producto (or exclusiva) entre las columnas “Intercalado” y “PNCode”.

7.2.3: Tabla “Walsh”

La tabla “Walsh” contiene 13 columnas, como puede verse en las figuras 7.7a y 7.7b.

	Esparcido 0	Esparcido 1	Esparcido 2	SortI 0	SortQ 0	SortI 1	SortQ 1
	-1	0	0	0	-1	0	0
	0	-1	-1	0	0	0	0
	-1	0	-1	-1	0	0	0
	0	-1	0	0	0	0	0
	-1	-1	-1	-1	-1	0	0
	0	0	0	-1	-1	0	0
	-1	-1	0	-1	0	0	0
	0	0	-1	0	0	0	0

Fig. 7.7a: Tabla Walsh

	CuadraturaI 0	CuadraturaQ 0	CuadraturaI 1	CuadraturaQ 1	CuadraturaI 2	CuadraturaQ 2
	-1	0	0	-1	0	-1
	0	0	-1	-1	-1	-1
	0	-1	-1	0	0	-1
	0	0	-1	-1	0	0
	0	0	0	0	0	0
	-1	-1	-1	-1	-1	-1
	0	-1	0	-1	-1	0
	0	0	0	0	-1	-1

Fig. 7.7b: Tabla Walsh

Las columnas “Esparcido” corresponden a la multiplicación de cada bit de la columna “Scrambled” de la tabla “Convolutacional” por su correspondiente Código de Walsh. Esto explica la cantidad de registros en la tabla, que será $64 * 384 = 24576$.

Las columnas “SortI 0” y “SortQ 0” contienen respectivamente las secuencias pseudoaleatorias “I” y “Q” del código corto, que identifican a cada sector de cada celda y se usan para enviar la información en cuadratura. La existencia de dos pares de estas columnas es necesaria cuando se simula enviar un mensaje por otro sector de la celda (en el caso de la figura no es necesario, puede verse que hay dos columnas vacías).

Las columnas “CuadraturaI” y “CuadraturaQ” son el resultado de multiplicar cada columna “Esparcido” por sus correspondientes “SortI” y “SortQ”. Estos son los bits que serán filtrados y modulados para transmitirse.

7.2.4: Tabla “Rf”

La tabla “RF” puede verse en las figuras 7.8a y 7.8b. Consta de 9 columnas, y la cantidad de registros se debe al hecho de que se simuló una modulación tal que entran 4 ciclos de portadora por cada bit y se tuvieron en cuenta 8 puntos para cada ciclo de portadora.

	PulsosI 0	PulsosQ 0	PulsosI 1	PulsosQ 1	PulsosI 2	PulsosQ 2
▶	0	0	0	0	0	0
	0,01054143	-0,01054143	-0,01054143	0,01054143	-0,01054143	0,01054143
	0,02117204	-0,02117204	-0,02117204	0,02117204	-0,02117204	0,02117204
	0,03182518	-0,03182518	-0,03182518	0,03182518	-0,03182518	0,03182518
	0,04240012	-0,04240012	-0,04240012	0,04240012	-0,04240012	0,04240012
	0,05279467	-0,05279467	-0,05279467	0,05279467	-0,05279467	0,05279467
	0,06290615	-0,06290615	-0,06290615	0,06290615	-0,06290615	0,06290615
	0,07263228	-0,07263228	-0,07263228	0,07263228	-0,07263228	0,07263228

Fig. 7.8a: Tabla RF

	Salida	Llegada	Con Ruido
▶	0	-0,3254588	1,730016
	-3,72694E-07	0,7733424	1,107583
	0,02117098	1,460881	2,256067
	0,04500761	1,348278	-0,7560971
	0,04240436	0,4723045	-1,508215
	9,33283E-06	-0,7072486	2,040153
	-0,06289672	-1,537443	-6,397267
	-0,1027175	-1,523014	1,084222

Fig. 7.8b: Tabla RF

Las columnas “PulsosI” y “PulsosQ” corresponden a la salida del filtro para las señales “CuadraturaI” y “CuadraturaQ” de la tabla “Walsh”, que se simula por correlación con la función Sinc correspondiente al ancho de banda del filtro.

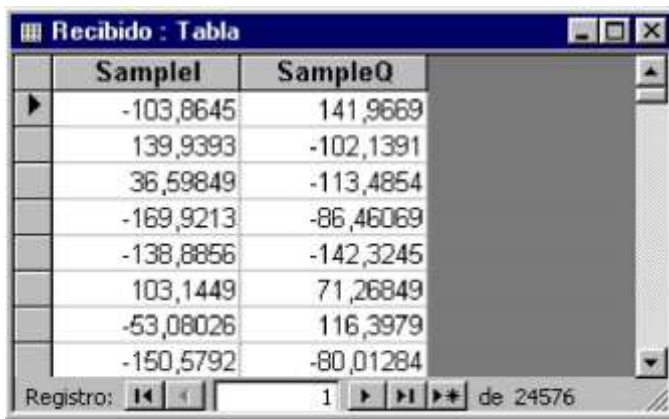
La columna “Salida” es la suma de los mensajes transmitidos por los 3 canales simulados, modulados en cuadratura. Esto es, la suma de cada columna “Pulsos” modulada con un coseno o un seno, según sea la componente en fase o en cuadratura.

La columna “Llegada” es el resultado de los distintos retardos y atenuaciones en el canal de transmisión, cuyas condiciones se han simulado de acuerdo a los parámetros ingresados en el formulario “Características del canal de transmisión”.

La columna “Con Ruido” es el resultado de sumarle al canal un valor de ruido blanco y un intervalo en el cual se destruye una corrida de bits, para poder luego ver el funcionamiento del decodificador convolucional. Estos valores también fueron elegidos por el usuario en el formulario “Características del canal de transmisión”.

7.2.5: TABLA “RECIBIDO”

Esta tabla consta de solo 2 columnas, como puede verse en la figura 7.9. Aquí se puede apreciar que la cantidad de registros en la tabla vuelve a ser 24576.



	SampleI	SampleQ
▶	-103,8645	141,9669
	139,9393	-102,1391
	36,59849	-113,4854
	-169,9213	-86,46069
	-138,8856	-142,3245
	103,1449	71,26849
	-53,08026	116,3979
	-150,5792	-80,01284

Registro: 1 de 24576

Fig. 7.9: Tabla Recibido

Las columnas “SampleI” y “SampleQ” son el resultado de demodular y filtrar la señal que se recibe, que es la columna “Con Ruido” de la tabla “RF”. Para esto se tienen en cuenta los 3 arribos para sumarlos coherentemente, y se dividen en componente en fase y en cuadratura al demodular. El filtrado se realiza por correlación con el pulso Sinc correspondiente al ancho de banda del filtro.

7.2.6: Tabla “Recuperado”

Para cada mensaje que se recupera, se genera la tabla “Recuperado” correspondiente, como puede verse en la base de datos que resulta de la simulación.

Cada una de estas tablas tiene la estructura que se ve en la figura 7.10, constando de 4 columnas.

La columna “Decisión” contiene los bits de mensaje recuperados. Para tomar la decisión se tienen en cuenta los valores de la tabla “Recibido”, conjuntamente con los valores del Código Corto correspondientes al sector y el Código de Walsh que corresponde al canal que se está recibiendo.

La columna “PNCode” contiene la secuencia del Código Largo que se usó para transmitir el mensaje, que ahora es generada por el receptor para poder descifrarlo.

La columna “DeScrambled” es el resultado de multiplicar las columnas “Decisión” y “PNCode”.

La columna “Reordenado” es el resultado de realizar el proceso inverso al Interleaving realizado durante la transmisión.

	Decision	PNCode	DeScrambled	Reordenado
	-1	0	-1	-1
	-1	0	-1	-1
	-1	0	-1	-1
	0	0	0	0
	0	-1	-1	-1
	0	-1	-1	0
	-1	-1	0	-1
	0	-1	-1	-1

Fig. 7.10: Tabla Recuperado

Esta tabla puede compararse con la tabla “Convolutacional” que se genera durante la transmisión, correspondiendo las columnas “Decisión”, “PNCode”, “DeScrambled” y “Reordenado” a las columnas “Scrambled”, “PNCode”, “Intercalado” y “Codificado” de dicha tabla respectivamente.

7.2.7: Tabla “Decodificado”

Para cada mensaje que se decodifica, se genera la tabla “Decodificado” correspondiente, como puede verse en la base de datos.

Cada una de estas tablas tiene la estructura que se ve en la figura 7.11, constando de 2 columnas. La cantidad de registros dependerá de la cantidad de bits del Frame recuperado.

	Frame	Mensaje
	0	0
	-1	-1
	-1	-1
	-1	-1
	-1	-1
	-1	-1
	-1	-1
	0	0

Fig. 7.11: Tabla Decodificado

La columna “Frame”, contiene los bits del Frame que se decodificó. Estos se obtienen procesando la columna “Reordenado” de la tabla “Recuperado” mediante el Algoritmo de Viterbi.

Esta tabla puede compararse con la tabla “CRC”, correspondiendo las columnas “Frame” y “Mensaje” a las columnas “Codificado” y “Mensaje” de dicha tabla respectivamente.

La columna “Mensaje” corresponde al mensaje propiamente dicho, y es igual a la columna “Frame” salvo en los bits que corresponden al código CRC (si lo hubiera). Si esta columna estuviera vacía (todos ceros), significa que se detectó mediante el código cíclico que el Frame decodificado no era válido. Este caso puede observarse en la figura 7.12.



	Frame	Mensaje
▶	-1	0
	0	0
	0	0
	-1	0
	-1	0
	0	0
	0	0
	0	0

Registro: 1 de 192

Fig. 7.12: Tabla Decodificado. Frame inválido.

Capítulo VIII: Conclusiones

A continuación se analizan algunos ejemplos, que pueden encontrarse en el directorio “Ejemplos” del CD, y que nos servirán para sacar algunas conclusiones sobre el funcionamiento del programa y del canal directo. Pueden analizarse los distintos procesos y ver que cantidad de errores pueden corregir el sistema

Ejemplo 1

El archivo “Ejemplo 1” es una base de datos donde se simuló la transmisión de los siguientes frames:

<i>Canal</i>	<i>Velocidad</i>	<i>Mensaje</i>	<i>Código De Walsh</i>
1	1200	A48F	11
2	4800	B8944C0CDB06DC5FD0F5	15
3	9600	09749A44DD9FE3BF381FA0911C40464FF6422A66B5A	15

Puede observarse en la tabla “CRC” que en las primeras tres columnas están los mensajes a transmitir, en binario. Por ejemplo, en la primera columna (Mensaje 0) se ve el mensaje A48F, que en binario es: 1010010010001111 (16 bits). En las dos columnas siguientes se observan los dos mensajes restantes, de 80 y 172 bits respectivamente (los ceros que siguen solo completan la tabla, ya que los registros no pueden estar vacíos).

Las tres columnas siguientes corresponden a los frames completos, con el código CRC correspondiente y los tail bits. Puede verse en la columna “Codificado 1” que los 8 bits de código CRC que se agregaron al mensaje se encuentran entre las filas 81 y 88, siendo su valor para el mensaje que se ha simulado: 01000111. En la columna “Codificado 2” se ven los 12 bits de código CRC entre las filas 173 y 184, y su valor en este caso es el siguiente: 110010000101.

Puede verse también que los 8 últimos datos, entre las filas 185 y 192, son los ceros correspondientes a los tail bits.

En la tabla “Convolutiva” se pueden observar en las tres primeras columnas la salida del codificador convolutiva para cada frame, donde se ve claramente la repetición de los símbolos de acuerdo a la velocidad de cada frame. Así, en la primera columna se aprecia que los bits se repiten 8 veces, ya que los datos se ven repetidos al menos 8 veces. En la columna siguiente se ven los datos repetidos de a pares, mientras que en la tercera no hay repetición.

Las 3 columnas siguientes (“Intercalado”) muestran el resultado del Interleaving para cada caso, pudiendo apreciarse que los datos ya no se repiten. Puede seguirse el proceso teniendo en cuenta el orden en que se realiza el proceso (ver la tabla “Interleaving”, en la base de datos “Auxiliar”), y seguir a cada símbolo hasta su nueva ubicación.

Las columnas siguientes (“PNCode” 0, 1 y 2), corresponden al Código Largo por el que se multiplicará cada mensaje. Puede observarse que son distintos entre sí, ya que cada

uno está enmascarado con el ESN (Electronic Serial Number) del teléfono móvil que corresponde.

Las tres últimas columnas (“Scrambled”) corresponden al mensaje ya multiplicado por el Código Largo correspondiente, donde puede observarse que cada símbolo es el resultado de una or exclusiva entre las columnas “Intercalado” y “PNCode” correspondientes.

En la tabla “Walsh”, puede observarse en las 3 primeras columnas el efecto de multiplicar cada símbolo a transmitir por el código de Walsh correspondiente. Estos códigos pueden verse en la tabla “Walsh” de la base de datos “Auxiliar”. Se aprecia claramente a lo largo de las columnas que cuando el símbolo a transmitir (columna “Scrambled” de la tabla “Convolutacional”) es un cero, se ve el código de Walsh correspondiente, mientras que cuando el símbolo a transmitir era un uno, se ve la versión negada del mismo código de Walsh. Como se eligió el mismo código de Walsh (el 15) para los canales 2 y 3, lo que corresponde a simular que el tercer mensaje se transmite por otro sector de la celda, puede verse claramente que las columnas 2 y 3 de la tabla muestran los mismos 64 valores cuando los símbolos a transmitir por los dos canales son iguales y muestran 64 valores distintos cuando estos símbolos son diferentes.

En las columnas “SortI 0” y “SortQ 0” se ven las componentes en fase y en cuadratura del Código Corto, mientras que las siguientes (“SortI 1” y “SortQ 1”) corresponden al Código Corto para el otro sector de la celda que se simula. Como el Código Corto es único y lo que varía es el offset, recordando que se simuló un offset de 512 chips entre estos dos sectores (se desplazó 512 veces el registro que genera el Código Corto para el segundo sector), puede verse en las columnas correspondientes este offset, ya que a partir del valor 513 de las columnas “SortI 0” y “SortQ 0” se ven los mismos valores que en las columnas “SortI 1” y “SortQ 1” desde el primer valor.

Las columnas siguientes corresponden a multiplicar cada una de las primeras tres columnas por su correspondiente Código Corto, obteniendo los mensajes para las componentes en fase y en cuadratura. Puede observarse como, después de aplicar el Código Corto, ya no hay ninguna relación entre los datos de los canales 2 y 3.

En la tabla “RF”, puede verse en las columnas “PulsosI” y “PulsosQ” el resultado de pasar los datos a transmitir (en formato bipolar) por el filtro (simulado mediante una función sinc, de la que se toma hasta el 3º cruce por cero). Se toman 32 muestras por cada chip, y puede observarse en la tabla como cada 32 valores hay un 1 o un -1, correspondiendo al centro de la sinc para el chip y a un valor nulo de la misma para los demás.

En las siguientes figuras puede verse la forma de onda de los pulsos, correspondiendo a las filas 161 – 481 de la tabla “RF”.

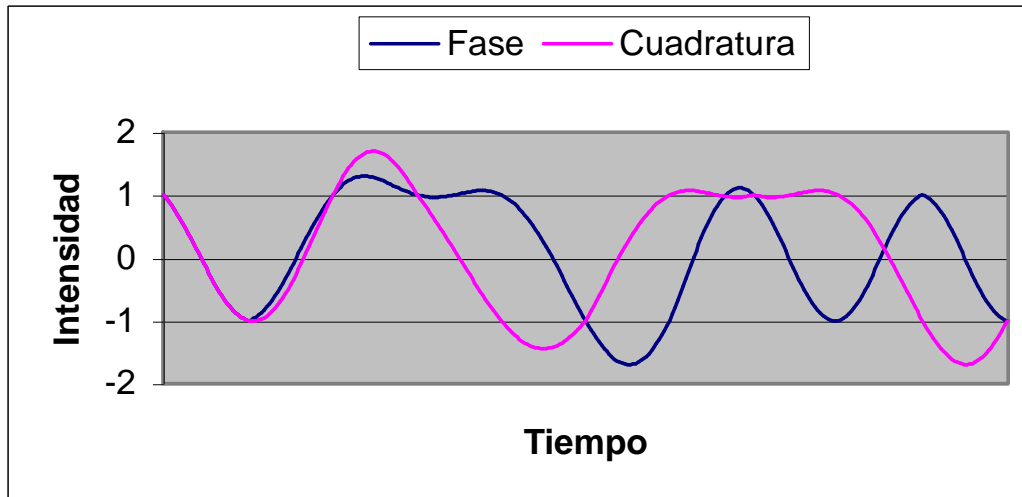


Fig. 8.1: Canal 0. Fase: 10111001010. Cuadratura: 10110011100.

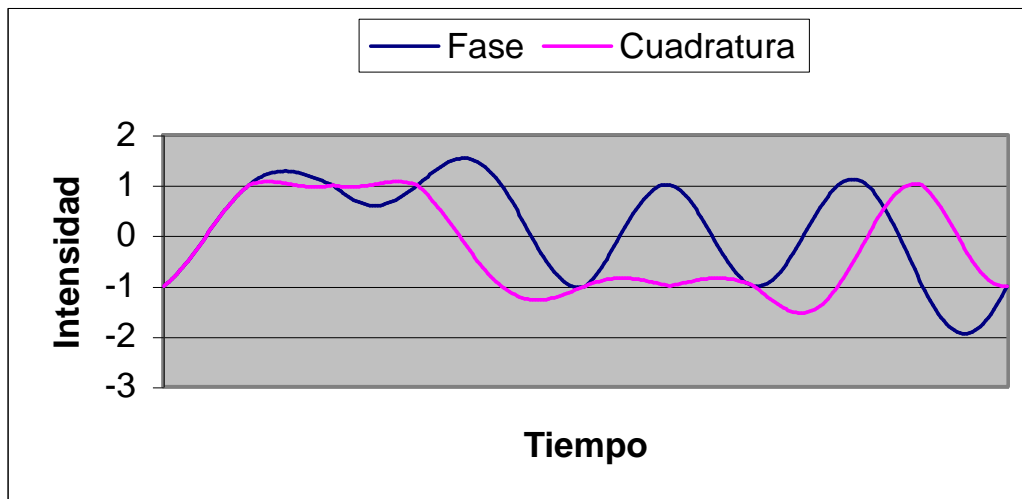


Fig 8. 2: Canal 1. Fase: 01111010100. Cuadratura: 01110000010.

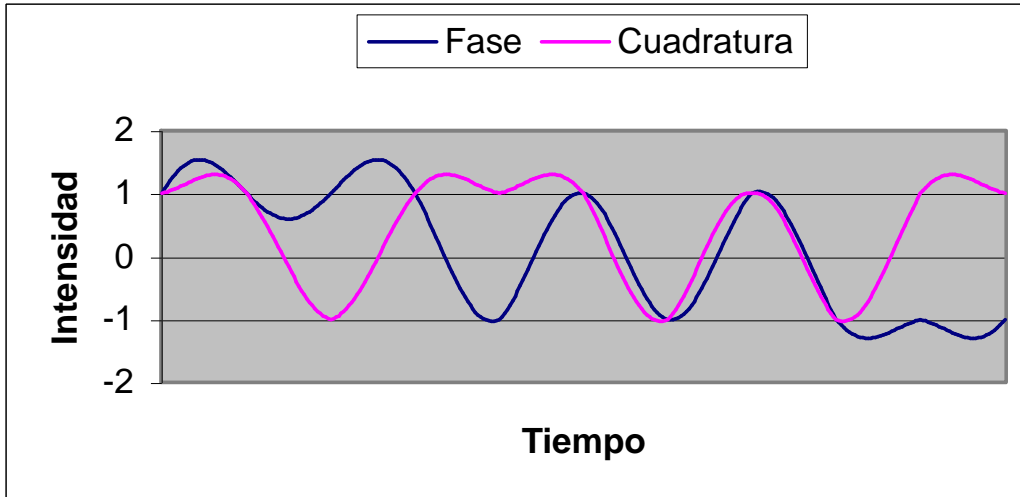


Fig. 8.3: Canal 2. Fase:11110101000. Cuadratura: 11011101011.

La columna “Salida” de la tabla es la suma de las anteriores, previamente multiplicadas por un coseno o un seno, según sea la componente en fase o en cuadratura. Para la simulación se moduló con una frecuencia tal que entran exactamente 4 ciclos de la portadora por cada chip, para tomar 8 puntos por ciclo sin hacer mas extensa aún la tabla. El resultado de esta suma puede verse en el gráfico 4. Puede apreciarse como en ciertos momentos las señales se suman y en ciertos momentos se restan. Al transmitirse más canales, este efecto se equilibra, resultando una señal mas plana.

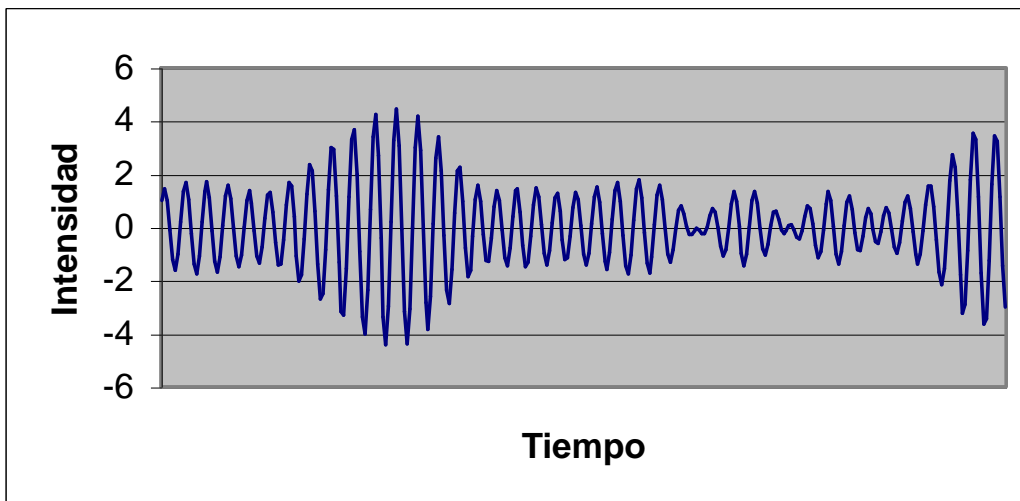


Fig. 8.4. Suma de los 3 canales modulados.

En la columna “Llegada”, se puede ver la misma señal al llegar al teléfono móvil, teniendo en cuenta el efecto del Multipath en el canal de transmisión, con los valores

que se eligió para simular (6 μ s de retardo y 0.8 de intensidad para el primer rebote, y 11 μ s y 0.5 para el segundo). Esta columna puede verse en el siguiente gráfico, donde se observa que la señal tiende a anularse en algunos puntos y reforzarse en otros.

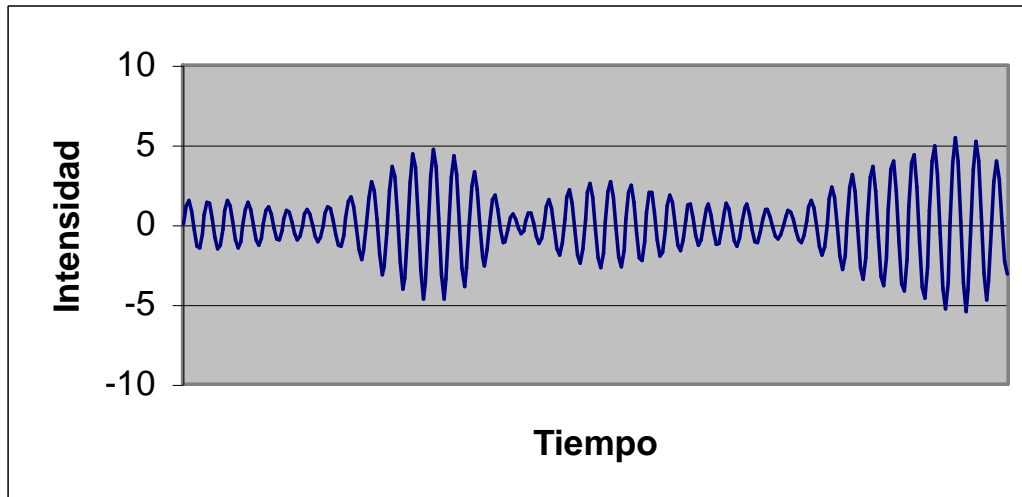


Fig. 8.5: Efecto del Multipath en el canal de transmisión.

La última columna de la tabla muestra la señal mas el ruido que se ha simulado. En este caso no se ha simulado ningún pulso de ruido para destruir una parte del mensaje. La forma de onda puede verse en el gráfico siguiente, donde se ve la señal totalmente enmascarada por el ruido que fue simulado.

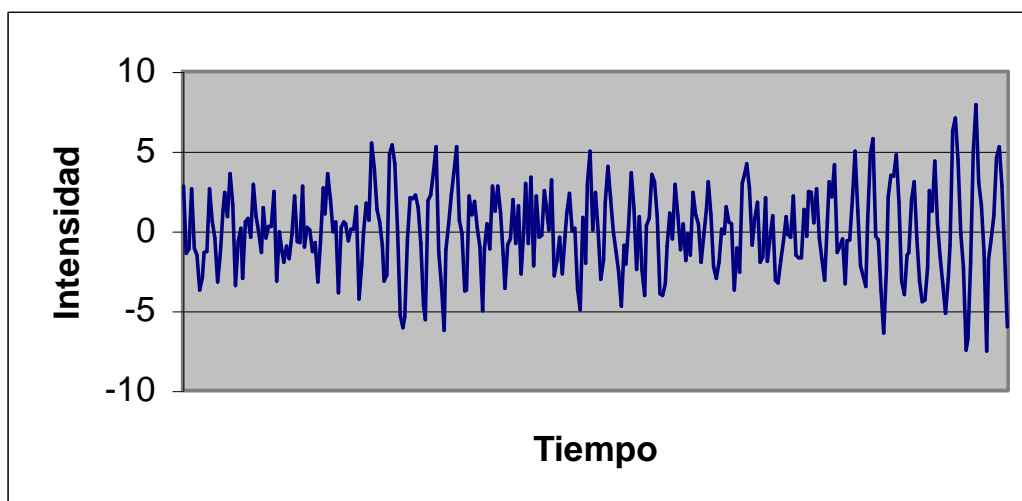


Fig. 8.6: Señal mas ruido en el receptor.

La tabla “Recibido” tiene solo dos columnas. Las columnas “SampleI” y “SampleQ” son el resultado de demodular y filtrar la señal que se recibe, que es la columna “Con Ruido” de la tabla “RF”. Para esto se tienen en cuenta los 3 arribos para sumarlos coherentemente, y se dividen en componente en fase y en cuadratura al demodular. El filtrado se realiza por correlación con el pulso Sinc correspondiente al ancho de banda del filtro.

La tabla “Recuperado”, de la que hay una por cada canal transmitido, tiene como primera columna “Decisión”, que es el resultado de multiplicar el valor de salida del filtro (tabla “Recibido”) por el Código Corto correspondiente al sector y el Código de Walsh correspondiente al canal, separando así los mensajes. La columna siguiente es el Código Largo, también enmascarado con el ESN del teléfono móvil. La columna “DeScrambled” es el resultado de multiplicar las dos anteriores, mientras que la última es el resultado de reordenar los símbolos, mediante el proceso inverso al Interleaving realizado en el transmisor. El orden en que se realiza puede verse también en la tabla “Interleaving” de la base de datos Auxiliar.

Por último, la tabla “Decodificado” muestra en la primera columna los frames que fueran decodificados mediante el algoritmo de Viterbi y en la segunda el mensaje original (Sin el código cíclico).

Ejemplo 2

La base de datos Ejemplo 2 muestra la simulación de un pulso de ruido que destruye una parte del frame, para poder observar como trabaja la corrección de errores.

Los mensajes que se simuló transmitir son los siguientes:

<i>Canal</i>	<i>Velocidad</i>	<i>Mensaje</i>	<i>Código de Walsh</i>
1	9600	B8944C0CDB06DC5FD0F58C09749A44DD9FE3BF381FA	9
2	9600	A1C258D98FE25BB8944C0CDB06DC5FD0F58C09749A4	35
3	9600	D9FE3BF381FA0911C40464FF6422A66B5A32917E4C6	15

Se simuló un pulso de 4 ms de duración, que comienza a los 2 ms del frame.

En la columna “Con Ruido” de la tabla “RF”, puede verse como se destruyó el mensaje entre las filas 78001 y 234000, donde se les dio el valor 1 a todos.

En las tablas “Recuperado” pueden verse los errores introducidos, ya que comparando la columna “Decisión” con la columna “Scrambled” correspondiente en la tabla “Convolucional”, puede verse que estos ocurren desde la fila 39 hasta la 114. Para observar el efecto del Interleaving, se debe comparar la columna “Reordenado” con la columna “Codificado” correspondiente en la tabla “Convolucional”, pudiendo verse claramente como los errores se dispersan a lo largo de todo el mensaje.

En la tabla “Decodificado” podemos ver el frame que se decodifica mediante el algoritmo de Viterbi, aquí puede observarse que solo se decodificó correctamente el frame transmitido por uno de los canales, de lo que se deduce que estamos en el límite de la cantidad de errores que puede corregir el código.

Ejemplos 3 y 4

En estos ejemplos se simuló la transmisión de frames de 4800 bps, para buscar el límite en la cantidad de errores que puede corregir el código.

Los mensajes a simular en el Ejemplo 3 son los siguientes:

<i>Canal</i>	<i>Velocidad</i>	<i>Mensaje</i>	<i>Código de Walsh</i>
1	4800	59A24FD762BC658DFAE0	11
2	4800	B8944C0CDB06DC5FD0F5	15
3	4800	6F089B272C3077505F97	15

Mientras que en el ejemplo 4 solo se han cambiado los códigos de Walsh por los valores 11, 39 y 18 respectivamente.

En el Ejemplo 3 se destruyen 12 ms y en el Ejemplo 4 se destruyen 6 ms con idéntico resultado, es decir, algunos mensajes se recibieron correctamente y otros no. La diferencia se encuentra en el sector del frame que fue afectado, ya que en el Ejemplo 3 se inicia desde cero, mientras que en el Ejemplo 4 se inicia a los 7 ms.

Esto se explica fácilmente observando el proceso de Interleaving. Como puede verse en la base de datos Auxiliar, los primeros 192 bits corresponden a números impares (1, 197, 97, 289, 49, 241, 145...), mientras que los pares comienzan a partir del 193 (4, 196, 100, 292...), por lo que al reordenarse, los errores que ocurrieron en la primera mitad del mensaje quedan en los símbolos impares, y los que ocurren en la segunda mitad quedan en los símbolos pares. Debido a esto, como los símbolos fueron repetidos en el transmisor (2 veces en este caso), los errores en la primer mitad del Ejemplo 1 solo afectan a una de los símbolos repetidos, por lo que se puede decodificar el mensaje perfectamente.

Así podemos suponer que se pueden destruir hasta 3 ms de cada repetición.

El motivo por el cual esto no es totalmente cierto puede verse en el Ejemplo 4, donde se puede ver que el primer canal fue decodificado mal aunque solo 2 ms de una de las repeticiones fueron destruidos. Esto se debe a que el decodificador de Viterbi no conoce la velocidad original del frame transmitido, y debe decodificar para los cuatro casos y elegir el más probable mediante el criterio de máxima similitud. Si abrimos la tabla "Decodificado0" veremos que se decodificó un frame de 192 bits (9600 bps), y luego se lo desechó por erróneo al comprobar el código cíclico. Es evidente que la cantidad de errores fue tan grande como para elegir un erróneamente un frame de 9600 bps en lugar del original.

Ejemplo 5

En este ejemplo se simula la transmisión de tres frames de 2400 bps, para buscar el límite en la cantidad de errores que puede corregir el sistema. Los mensajes que se simuló transmitir son los siguientes:

Canal	Velocidad	Mensaje	<i>Código de Walsh</i>
1	2400	492D3A5CB8	13
2	2400	D0F58C0974	45
3	2400	44DD9FE3BF	20

En el canal se simuló destruir 12 ms de mensaje, a partir de los 4 ms.

Como puede verse en las tablas "Decodificado", mientras en una se decodificó correctamente el mensaje, en las dos restantes se decodificaron erróneamente frames de 192 bits (9600 bps), que luego fueron desechados al verificar el código cíclico.

Todos estos ejemplos también pueden abrirse desde el programa, pudiendo modificar algunos parámetros del canal de transmisión y simular a partir de ahí para ver los efectos de estos cambios.

Capítulo IX: Bibliografía

IS-95A: The cdmaOne Standard. CDMA Training Department. Qualcomm. 1997.

Mobile Cellular Telecommunications. Analog and Digital Systems. William C. Y. Lee. Mc Graw-Hill. Second edition.1995.

The Mobile Communications Handbook. Jerry D. Gibson. CRC Press.1996.

CDMA. Principles of Spread Spectrum Communication. Andrew J. Viterbi. Addison Wesley Wireless Communications Series. 1997.

Manual de Telefonía. Telefonía Fija y Móvil. José Manul Huidobro.Editorial Paraninfo. 1998.

Sistemas de Comunicación Móvil. Una introducción. Domingo Lara Rodríguez, David Muñoz Rodríguez y Salvador Rosas García. Alfaomega. 1995.

CDMA Technology and Benefits. A introduction to the Benefits of CDMA for Wireless Telephony. Cellular Infraestructure Group. Motorola, Inc. 1996.

Apéndice I: Asignación de frecuencias

El servicio de telefonía celular ocupa la porción de espectro que se ve en la figura 1. El teléfono móvil transmite a una frecuencia 45 MHz menor que la celda. Originalmente cada operador tenía asignada una banda de 10 MHz en cada dirección, que corresponden a las bandas A y B de la figura, pero luego se agregaron las 3 bandas restantes, por lo que cada uno cuenta con 12.5 MHz de ancho de banda en cada dirección.

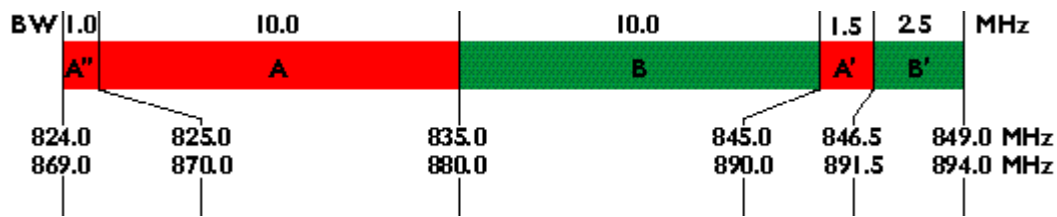


Fig. 1: Banda celular.

PCS utiliza la porción de espectro que se ve en la figura 2. Hay en total 60 MHz asignados en cada dirección. La división que se ve en la figura corresponde a EEUU. En Argentina el espectro se divide entre 4 operadores, dos de los cuales tienen asignados 10 MHz de ancho de banda y los otros dos tienen 20 MHz. Corresponde el mayor ancho de banda a los operadores que en la zona no tienen asignado espectro en la banda celular.

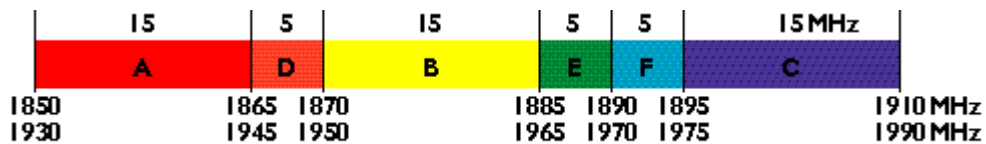


Fig. 2: Banda PCS

Apéndice II: Códigos Convolucionales.

Existen dos tipos de codificación para el control de error en un canal, los códigos de bloques y los códigos convolucionales. Se ha analizado hasta ahora las características de los códigos de bloques. En este capítulo se realiza el análisis de los códigos convolucionales. En un código convolucional los n bits de la salida del codificador dependen no solo de las k bits de entrada, sino también de K bloques de información de entrada previos. Un código convolucional (n, k, K) se implementa con k entradas que ingresan a un circuito secuencial de n salidas con un nivel de memoria K . Típicamente n y k son números enteros siendo normalmente $k < n$. El nivel de memoria K debe hacerse grande para aumentar la capacidad de detección de error del código.

Codificación en códigos convolucionales.

La figura 1 representa un codificador para generar un código convolucional.

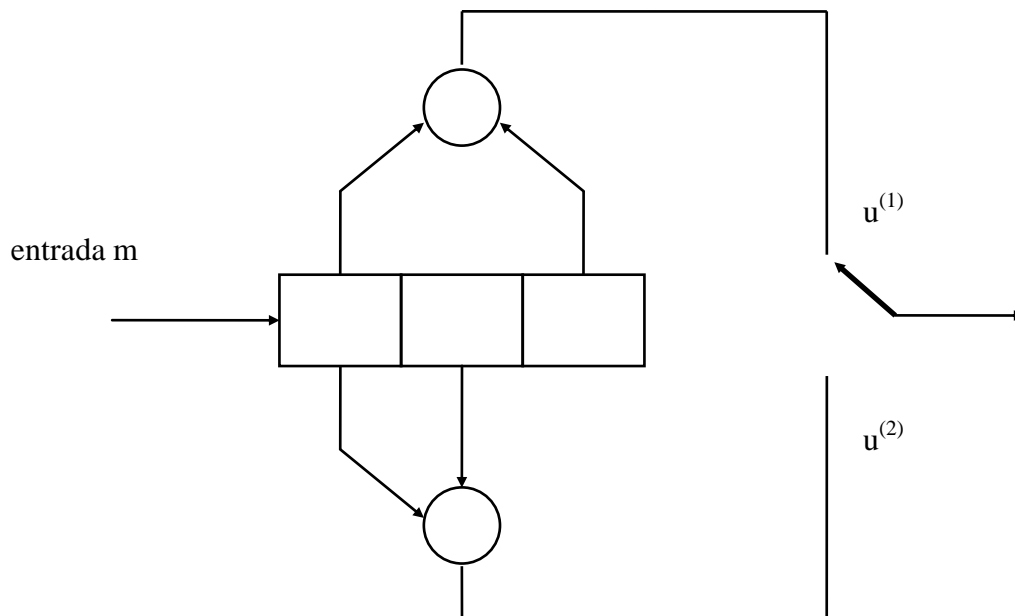


Figura 1.

Este será un código convolucional binario $(2,1,3)$. El vector de entrada es una secuencia que tiene la forma:

$$m = (m_0, m_1, m_2, \dots) \quad (\text{ecuación 1})$$

El vector secuencia de salida u se configura con las secuencias de salida $u^{(1)}$ y $u^{(2)}$:

$$\begin{aligned} \mathbf{u}^{(1)} &= (u_0^{(1)}, u_1^{(1)}, u_2^{(1)}, \dots) \\ \mathbf{u}^{(2)} &= (u_0^{(2)}, u_1^{(2)}, u_2^{(2)}, \dots) \end{aligned}$$

(ecuación 2)

Un esquema general de un codificador convolucional emplea kK registros que operan en modulo-2. El parámetro K se denomina longitud de restricción. Consiste básicamente de una medida de la forma en que los bits previos afectan a los que se están generando en la salida. En la figura se ve que las secuencias de entrada de k bits son desplazadas hacia la derecha en las primeras k etapas de los registros. La salida es muestreada para generar la secuencia de salida del código. Existen n bits de salida por cada k bits de entrada, por lo que se puede definir la velocidad del código como k/n siendo $k < n$. La mayoría de los códigos convolucionales emplean $k = 1$ de forma que los bits de entrada son ingresados de forma serie. Para este tipo de codificador convolucional la velocidad de código es $1/n$. El registro del sistema que tiene kK etapas, tendrá en este caso entonces solo K etapas. La longitud K entonces se define en cantidad de bits para este caso.

Representación de un codificador convolucional.

La figura 1 muestra la llamada representación de conexiones que se emplea en estos códigos. Existen diferentes forma de representación de la información y de la forma en que una secuencia de m bits de entrada genera una secuencia u de salida.

Las representaciones mas conocidas son el gráfico de conexiones, los polinomios o vectores de conexión, el diagrama de estados, el diagrama de árbol, y el trellis.

Representación de conexiones.

Como se ve en la figura 1, que representa un codificador convolucional (2,1,3) cada bit de los m de la secuencia de entrada se desplaza a los registros en cada cambio de clock, y la salida es muestreada dos veces generando los valores de la secuencia de salida. Los valores de la salida dependen de la forma en que las conexiones están realizadas. La secuencia de salida será diferente si las conexiones de los registros se cambian. Una descripción de un código convolucional emplea un conjunto de vectores conexión donde la existencia de la misma se representa con un uno, mientras que la ausencia de conexión se representa con un cero. Por ejemplo para el codificador de la figura los vectores de conexión g_1 y g_2 que describen las ramas superior e inferior son respectivamente:

$$\begin{aligned} g_1 &= 1 \ 0 \ 1 \\ g_2 &= 1 \ 1 \ 0 \end{aligned}$$

La secuencia de salida para un dado vector de entrada se analiza en la siguiente tabla.

El vector de entrada es $m = (1\ 0\ 0\ 1\ 0\ 1)$. Se analizan los valores de los contenidos del registro, el estado actual, el estado siguiente o próximo y las secuencias de salida $u^{(1)}$ y $u^{(2)}$:

entrada m_i	registros	estado en t_i	estado en t_{i+1}	$u^{(1)}$	$u^{(2)}$
-	0 0 0	0 0	0 0	-	-
1	1 0 0	0 0	1 0	1	1
0	0 1 0	1 0	0 1	0	1
0	0 0 1	0 1	0 0	1	0
1	1 0 0	0 0	1 0	1	1
0	0 1 0	1 0	0 1	0	1
1	1 0 1	0 1	1 0	0	1
0	0 1 0	1 0	0 1	0	1
0	0 0 1	0 1	0 0	1	0

La secuencia de salida es entonces $u = 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0$

Representación polinomial.

Otra forma de representar un codificador convolucional es usando polinomios generadores. Cada polinomio es de grado $K-1$ o menor y describe las conexiones de forma similar a la representación de conexiones. Los coeficientes de estos polinomios son iguales a uno cuando la conexión existe, y a cero cuando no existe. Para el codificador de la figura por ejemplo los polinomios generadores de la parte superior e inferior son respectivamente, g_1 y g_2 :

$$g_1(X) = 1 + X^2$$

$$g_2(X) = 1 + X$$

Donde el termino de mas bajo orden en el polinomio corresponde a la etapa del registro. Dado un vector de entrada $m = (1\ 0\ 0\ 1\ 0\ 1)$, el vector de salida se calcula de la siguiente forma:

$$m(X) \cdot g_1(X) = (1 + X^3 + X^5)(1 + X^2) = 1 + X^2 + X^3 + X^7$$

$$m(X) \cdot g_2(X) = (1 + X^3 + X^5)(1 + X) = 1 + X + X^3 + X^4 + X^5 + X^6$$

$$u(X) = (1,1) + (0,1)X + (1,0)X^2 + (1,1)X^3 + (0,1)X^4 + (0,1)X^5 + (0,1)X^6 + (1,0)X^7$$

La secuencia de salida es entonces $u = 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0$

Representación por diagrama de estados.

El estado de un código convolucional $1/n$ se define como el contenido de los $K-1$ registros de estado que se encuentran a la derecha. De esta manera los $K-1$ registros de estado de la izquierda constituyen el próximo estado. El diagrama de estados es una forma de representación de la evolución de las secuencias de estado para estos códigos. El codificador analizado en este ejemplo tiene un diagrama de estados como el que se ve en la figura 2.

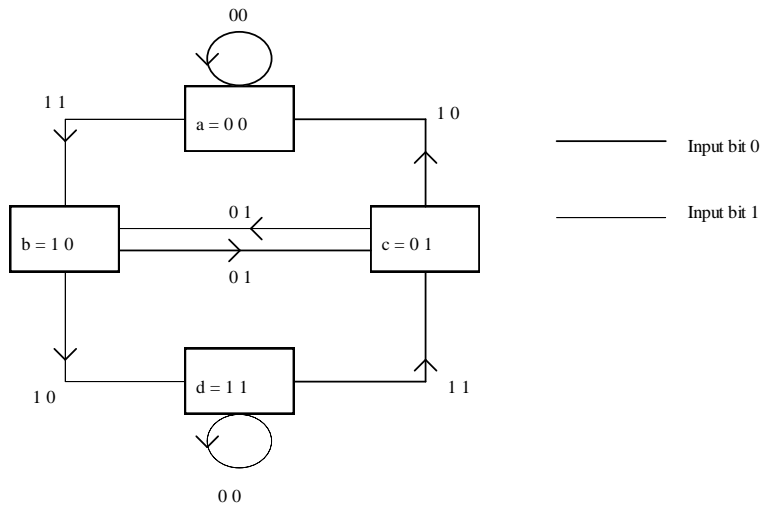


Figura 2

El sistema se analiza considerando que existen cuatro estados representados por $a = 00$, $b = 10$, $c = 01$ y $d = 11$. Existen solo dos transiciones que emergen de cada estado y que tienen que ver con las dos posibilidades de entrada binaria existentes, '1' o '0'. La convención empleada es tal que un '1' lógico de entrada se representa con líneas de puntos, mientras que un '0' lógico de entrada se representa con líneas llenas. Una característica importante de estos códigos se hace evidente de la observación del diagrama de estado. No es posible pasar de un estado a otro arbitrario. Esto resulta del mecanismo de memoria que el sistema tiene.

El diagrama en árbol.

El diagrama de estados permite conocer la transición entre estados, pero no aporta información acerca de la historia del sistema es decir, de los estados y transiciones iniciales.

El diagrama de árbol tiene en cuenta la dimensión del tiempo, poniendo en evidencia la forma en que la secuencia realmente sucede. El diagrama de árbol para la secuencia de entrada $m = (1\ 0\ 0\ 1\ 0\ 1)$ que entra al código convolucional de la figura es el que se muestra en la figura 3.

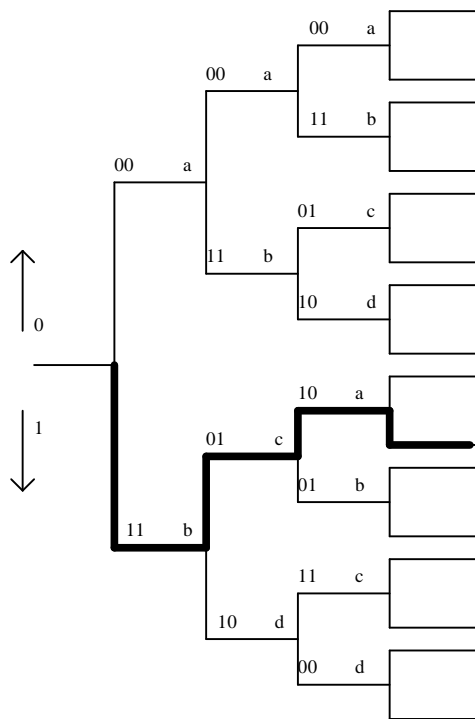


Figura 3

Cada bit de entrada mueve la secuencia hacia arriba o hacia abajo dependiendo de su valor. Un bit de entrada '0' se representa como un movimiento hacia arriba. Un bit de entrada '1' se representa como un movimiento de la secuencia hacia abajo. En este ejemplo el sistema se inicia en el estado $a = 00$. La secuencia evoluciona hacia abajo yendo al estado $b = 01$ con una salida 11, luego se genera la salida 01 yendo al estado $d = 11$, y así sucesivamente. Esta representación pone en evidencia la forma en que la secuencia se inicia, así como la dependencia del estado inicial. Sin embargo la realización de este esquema se torna impracticable para largas secuencias. Existe por otra parte algún tipo de repetición en la formación de las secuencias que hacen de este gráfico una representación redundante.

Una representación mejorada de estos códigos es la llamada representación por trellis.

Trellis para un código convolucional.

Una propiedad importante de los códigos convolucionales es observada en la representación de árbol hecha previamente. El sistema tiene, a partir de determinado momento una repetición en su estructura. La estructura del árbol se repite después de K ramas, donde K es la constante de longitud asociada a estos códigos. La siguiente figura es la representación en trellis del código convolucional estudiado en este ejemplo:

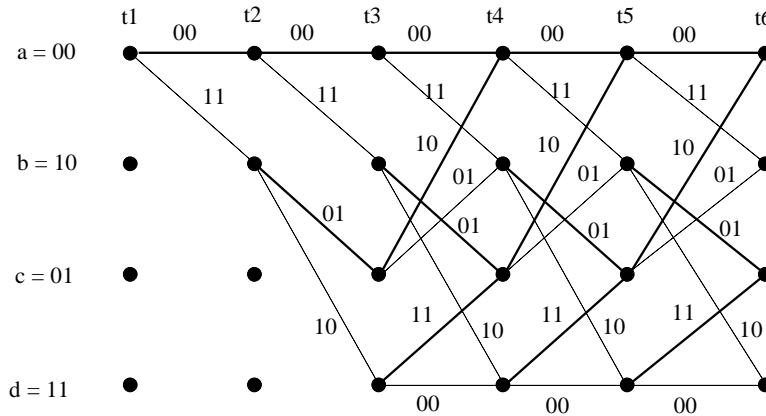


Figura 4.

Se utiliza la misma convención para los bits de entrada que se empleó en el diagrama de árbol. Las líneas sólidas representan un bit de entrada ‘0’ mientras que líneas punteadas se usan para representar un bit de entrada ‘1’. Existen 2^{K-1} posibles estados para ser representados. Como se observa, la estructura se torna periódica después del instante t_4 . Existen también dos ramas que emergen de cada estado pertenecientes a los dos posibilidades de bits de entrada, y dos ramas que arriban a cada estado.

Decodificación por máxima similitud (MLD: Maximum Likelihood Detection).

Para una cierta secuencia de salida de un código convolucional $U^{(m)}$ existe una secuencia recibida, denominada Z , que resulta de la aparición de errores por efecto del ruido sobre la secuencia original U . El decodificador óptimo es tal que compara las probabilidades condicionales $P(Z/U^{(m)})$ de que la secuencia recibida Z corresponda a una secuencia conocida $U^{(m)}$ y toma la decisión de elegir la secuencia que tiene la mayor probabilidad de coincidir con Z :

$$P(Z / U^{(m)}) = \max_{U^{(m)}} P(Z / U^{(m)}) \tag{ecuación 3}$$

Este es el criterio de máxima similitud. Esta obviamente de acuerdo con el hecho intuitivo de asignar la secuencia conocida en el decodificador que más se parece a la recibida.

La aplicación de este criterio a los códigos convolucionales enfrenta el hecho que un gran número de secuencias posibles debe ser considerada. Para una secuencia de L bits se tienen 2^L posibilidades. El decodificador de máxima similitud elige una secuencia $U^{(m)}$ del conjunto posible, que tenga la mayor probabilidad de parecerse al vector recibido.

Si el canal no tiene memoria, el ruido es aditivo blanco y gaussiano, cada símbolo está afectado de una forma independiente de los otros. Para un código convolucional de

velocidad $1/n$ la función probabilidad que mide la similitud respecto del vector recibido es:

$$P(Z/U^{(m)}) = \prod_{i=1}^{\infty} P(Z_i / U_i^{(m)}) = \prod_{i=1}^{\infty} \cdot \prod_{j=1}^n P(z_{ji} / u_{ji}^{(m)})$$

(ecuación 4)

Donde Z_i es la i -ésima rama de la secuencia recibida Z , $U_i^{(m)}$ es la i -ésima rama de la secuencia de una palabra de código $U^{(m)}$, z_{ij} es el j -ésimo símbolo de código de Z_i , y $u_{ji}^{(m)}$ es el j -ésimo símbolo de código de $U_i^{(m)}$ donde cada rama esta constituida por n símbolos de código. El proceso de decodificación consiste en la elección de una secuencia que maximice la función similitud. Este problema se analiza a continuación.

Algoritmo de decodificación convolucional de Viterbi.

El algoritmo de decodificación de Viterbi realiza la detección por máxima similitud. Para eso utiliza las propiedades del trellis de un código convolucional. El algoritmo intenta reducir la complejidad del calculo evitando tener en cuenta la totalidad de las secuencias posibles. El procedimiento consiste en calcular la distancia entre la señal recibida en el instante t_i y los caminos o ramas entrantes del trellis en ese instante en el estado analizado. En la medida que este criterio se va aplicando se evalúa la secuencia que tiene la menor distancia respecto de la recibida, de forma que la secuencia de máxima similitud finalmente aparece. El camino o secuencia de máxima similitud, que es al mismo tiempo aquel que presenta la menor distancia respecto del recibido, se denomina camino o secuencia sobreviviente. Decir entonces que la secuencia es la de máxima similitud, es también decir que se trata de aquella secuencia que tiene la menor distancia respecto de la recibida.

Se analiza un ejemplo para una mejor comprensión del mecanismo de decodificación. Se emplea el criterio de distancia denominado de Hamming. Es decir, la distancia entre dos palabras será el número de elementos diferentes que entre ellas exista.

Ejemplo: Dada la secuencia de entrada $Z = 11\ 01\ 11\ 11\ 01\ \dots$ correspondiente al código convolucional de la figura, se aplica el algoritmo de Viterbi para decodificarla. El primer paso de este proceso es determinar la distancia de Hamming entre el vector recibido y la salida correspondiente a cada estado del sistema.

Input data :	1	0	0	1	0	1
Transmitted word U :	1 1	0 1	1 0	1 1	0 1	0 1
Received sequence Z:	1 1	0 1	1 1	1 1	0 1	0 1

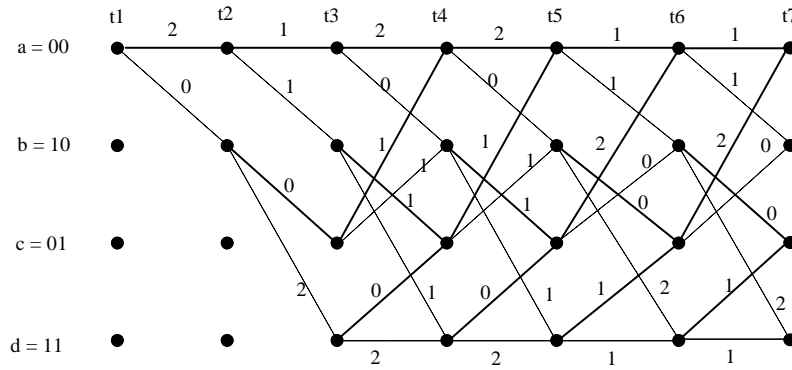


Figura 5

Uno de las dos ramas que arriban a cada estado puede ser eliminada por tener mayor distancia que la otra. Las decisiones comienzan a tomarse cuando dos secuencias diferentes arriban al mismo estado por dos o mas caminos distintos. Por ejemplo, esto se realiza en el instante t_4 .

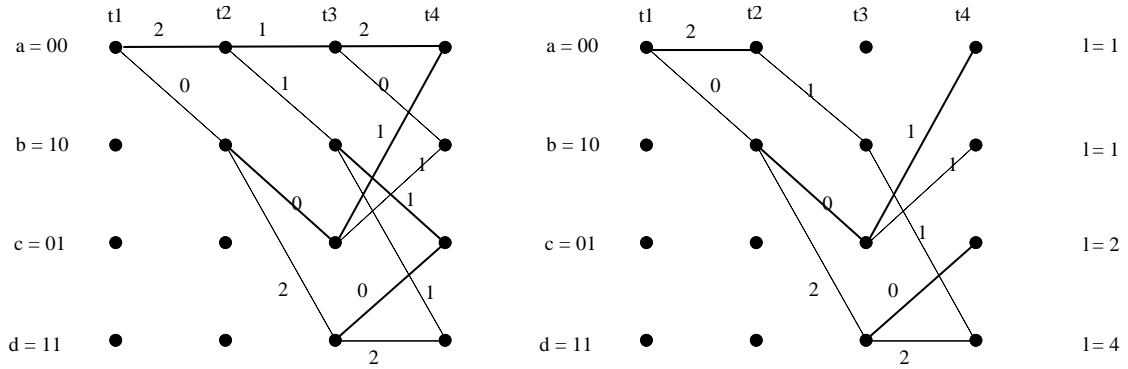


Figura 6

En el instante t_4 se toman decisiones sobre los caminos que arriban a cada estado, y se eliminan aquellos que tienen mayor distancia que el sobreviviente. El procedimiento se repite sucesivamente en cada estado, teniendo en cuenta las decisiones previas, lo que simplifica el proceso de selección. La secuencia detectada se ve en la figura 7.

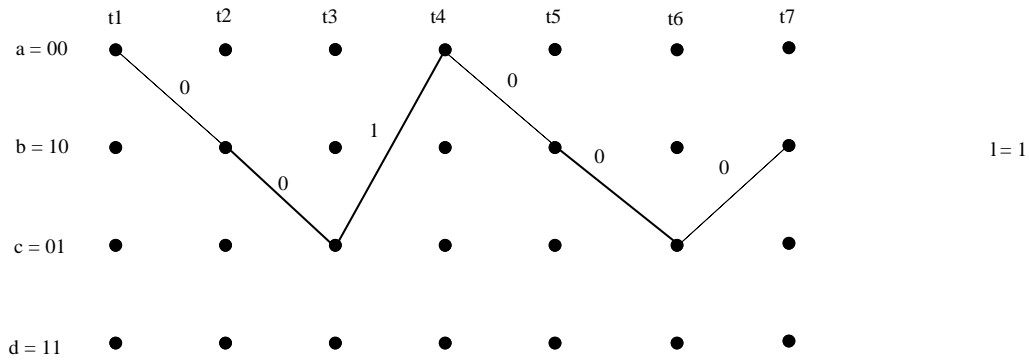


Figura 7.

La decisión tomada en el instante t_7 permite conocer la secuencia de entrada que originó la secuencia recibida. La secuencia elegida es aquella que tiene la menor distancia acumulada. Observando entonces la secuencia decodificada, y por inspección del tipo de línea que tiene cada rama asociada, se puede determinar si la entrada fue un '1' o un '0'.

La secuencia decodificada es entonces 1 0 0 1 0 1. El sistema produjo la corrección de un error en la secuencia de entrada Z.

Propiedades de distancia de los códigos convolucionales.

Una de las características de todo sistema de codificación es el parámetro distancia mínima, que resulta ser la menor entre todos los valores de distancia entre los elementos del código. Para el caso de los códigos lineales esa distancia puede tenerse en cuenta calculando la distancia mínima entre cada vector y el vector "todos ceros", o vector nulo. Para el caso de los códigos convolucionales la distancia entre vectores no es tan claramente definida, debido a que la transmisión no es por bloques, sino más bien por secuencias con cierto contenido de memoria. Sin embargo se puede suponer que el sistema está fuera de posibilidad de corregir errores cuando estando en el estado "todos ceros", la secuencia originalmente emitida es de ceros consecutivos, y por efecto del ruido la secuencia se aparta de ser una cadena de ceros, y no tiene posibilidad de retornar al estado "todos ceros". Este evento será el de error. Luego, la secuencia que tenga la mínima distancia, y que partiendo del estado cero, retorna a él después de una serie de transiciones, será la secuencia que define la distancia mínima del código.

El procedimiento para definir la distancia mínima de un código convolucional consiste en calcular la distancia para todos los caminos que salen y retornan al y del estado cero. Aquel con distancia mínima define la distancia del código. Esta distancia se la denomina *mínima distancia libre*, d_{free} . La capacidad de corrección del sistema, o sea el número t de errores que puede corregir, estará dada entonces por la expresión:

$$t = \text{parte entera de } \left\{ \frac{d_{\text{free}} - 1}{2} \right\}$$

(ecuación 5)

Esto es solo cierto en algunas condiciones, como son que la longitud de decodificación sea suficiente, y que el valor de K sea no muy elevado.

Apéndice III: Códigos Cíclicos.

Los códigos cíclicos son una importante clase de códigos de bloques lineales. La idea de su diseño es la facilidad de ser implementados usando registros de desplazamiento que emplean realimentación. La decodificación por síndrome también se realiza empleando registros de desplazamiento. La matemática utilizada para generar estos códigos es una extensión de la suma y multiplicación módulo-2.

Un código de bloques lineal se dice código cíclico si para una palabra de código

$$U = (u_0 \ u_1 \ \dots \ u_{n-1}) \quad (\text{ecuación 1})$$

se tiene siempre que existe otra palabra de código de la forma:

$$U^{(1)} = (u_{n-1} \ u_0 \ u_1 \ \dots \ u_{n-2}) \quad (\text{ecuación 2})$$

La notación empleada significa que $U^{(1)}$ es la versión de U que se obtiene desplazando en un registro en forma cíclica esta palabra. El último bit pasa a ser el primero, el primero el segundo y así sucesivamente. En general el desplazamiento de bits puede ser mayor implicando que cualquier rotación de la palabra original genera otra palabra del mismo código.

Las palabras del código pueden ser representadas como un polinomio:

$$U(x) = u_0 + u_1X + u_2X^2 + \dots + u_{n-1}X^{n-1} \quad (\text{ecuación 3})$$

En esta representación polinomial, el exponente del término independiente X significa sencillamente la posición que el bit tiene dentro de la palabra. Los coeficientes son términos binarios de manera que la presencia o ausencia de el término X^j significa que el bit en la posición j es un uno o un cero respectivamente.

La utilidad de esta representación será evidente más adelante para la definición de las propiedades de estos códigos.

Estructura algebraica de los códigos cíclicos.

Si se expresa a las palabras del código en forma polinomial puede decirse que una palabra de código es básicamente un polinomio de orden $(n-1)$. Puede demostrarse que si $U(X)$ es una palabra de código expresada como polinomio, entonces $U^{(i)}(X)$ que es el resto de la división del polinomio $X^i U(X)$ por el polinomio $X^n + 1$ es también una palabra de código. Esto significa que:

$$\frac{X^i U(X)}{X^n + 1} = q(X) + \frac{U^{(i)}(X)}{X^n + 1} \quad (\text{ecuación 4})$$

Si se multiplica por $X^n + 1$:

$$X^i U(X) = q(X)(X^n + 1) + U^{(i)}(X) \quad (\text{ecuación 5})$$

En esta operación se dice que una nueva palabra de código se genera por división del polinomio $X^i U(X)$ sobre $X^n + 1$, tomando el resto de la división. Este tipo de operación es también llamada función modulo. esto es, la palabra del código generada de esta forma es

$$U^{(i)}(X) = X^i U(X) \text{ modulo } (X^n + 1) \quad (\text{ecuación 6})$$

Como se sabe, x modulo y es el resto de la división de x sobre y .

Si por ejemplo:

$$U(X) = u_0 + u_1 X + u_2 X^2 + \dots + u_{n-1} X^{n-1} \quad (\text{ecuación 7})$$

entonces un desplazamiento de esta palabra es:

$$X \cdot U(X) = u_0 X + u_1 X^2 + u_2 X^3 + \dots + u_{n-1} X^n \quad (\text{ecuación 8})$$

Sumando y restando apropiadamente u_{n-1} en operaciones modulo-2 obtenemos:

$$\begin{aligned} X \cdot U(X) &= u_{n-1} + u_0 X + u_1 X^2 + u_2 X^3 + \dots + u_{n-2} X^{n-1} + u_{n-1} X^n + u_{n-1} \\ &= U^{(1)}(X) + u_{n-1}(X^n + 1) \end{aligned} \quad (\text{ecuación 9})$$

En general se puede decir que

$$U^{(i)}(X) = X^i U(x) \text{ modulo } (X^n + 1) \quad (\text{ecuación 10})$$

Ejemplo:

Si la palabra (1 1 0 1) pertenece a un código cíclico, expresela en forma polinomial, y determine por división polinomial otra palabra del mismo código.

$$U(X) = 1 + X + X^3 \quad (\text{ecuación 11})$$

la palabra es expresada como un polinomio donde los bits menos significativos se toman sobre el lado izquierdo de la expresión vectorial correspondientes a los términos en X de menor grado. Si por ejemplo el desplazamiento fuera e orden triple, $i = 3$, el polinomio resultante sería:

$$X^i U(X) = X^3 + X^4 + X^6 \quad (\text{ecuación 12})$$

Si se divide $X^3 U(X)$ por X^4+1 ,

$$\begin{array}{r} X^6 + X^4 + X^3 \quad | \quad X^4 + 1 \\ X^2 + 1 \\ \hline X^6 X^2 \\ \hline X^4 + X^3 + X^2 \\ X^4 + 1 \\ \hline X^3 + X^2 + 1 \end{array}$$

El polinomio $1+X^2+X^3$ representa a la palabra (1 0 1 1), que resulta ser la palabra original (1 1 0 1) sobre la que se realizaron tres desplazamientos sobre la derecha.

Propiedades de los códigos cíclicos binarios.

Los códigos cíclicos pueden ser generados haciendo uso de un polinomio generador. El polinomio generador $g(X)$ para un código (n,k) cíclico lineal de bloques tiene la forma:

$$g(X) = g_0 + g_1 X + g_2 X^2 + \dots + g_r X^r \quad (\text{ecuación 13})$$

donde el primer y el ultimo coeficiente deben ser igual a uno, es decir $g_0 = g_r = 1$. Cualquier palabra del código se genera por multiplicación del vector de entrada por el polinomio generador:

$$U(X) = m(X).g(X) \quad (\text{ecuación 14})$$

donde:

$$m(X) = m_0 + m_1X + m_2X^2 + \dots + m_{n-r-1}X^{n-r-1} \quad (\text{ecuación 15})$$

Existen 2^{n-r} palabras de código, y a su vez 2^k vectores de mensaje. Si se desea tener una relación biunívoca entre las palabras codificadas y las de mensaje debe suceder que:

$$n - r = k, \quad \text{o} \quad r = n - k.$$

Por lo tanto $g(X)$ debe ser un polinomio de orden $(n - k)$. Cada palabra del código, expresada en forma polinomial se expresa entonces como:

$$U(X) = (m_0 + m_1X + m_2X^2 + \dots + m_{k-1}X^{k-1})g(X) \quad (\text{ecuación 16})$$

De la expresión anterior se deduce que U es una palabra de código si la división del polinomio $U(X)$ por $g(X)$ tiene resto cero. También es demostrable que un polinomio generador $g(X)$ de un código cíclico (n,k) es factor del polinomio X^n+1 . Entonces deberá cumplirse que

$$X^n + 1 = g(X)h(X) \quad (\text{ecuación 17})$$

Se puede verificar como ejemplo que :

$$X^7 + 1 = (1 + X + X^3)(1 + X + X^2 + X^4)$$

Así entonces si el polinomio generador es por ejemplo $g(X) = 1+X+X^3$ el código generado será un código cíclico lineal de bloques $(n,k) = (7,4)$. El polinomio alternativo sería $g(X) = 1+X+X^2+X^4$, que crearía un código lineal de bloques de tipo cíclico $(7,3)$. Cualquier polinomio $g(X)$ de grado $(n - k)$ que sea factor de X^n+1 genera de forma única un código (n,k) cíclico.

Forma sistemática de codificación.

La forma sistemática de un código de bloques tiene por característica el mostrar de forma transparente en una parte identificada de la palabra codificada cual ha sido la palabra de mensaje que la origina. Un vector de mensaje se puede expresar en forma polinomial como:

$$m(X) = m_0 + m_1X + m_2X^2 + \dots + m_{k-1}X^{k-1} \quad (\text{ecuación 18})$$

Las palabras de un código cíclico de tipo sistemático tienen la propiedad de poseer los bits de la palabra de mensaje en el lado derecho, mientras que los bits de control de paridad se ubican en el lado izquierdo. Para desplazar los bits de mensaje sobre la derecha es suficiente con multiplicar el vector de mensaje $m(X)$ por X^{n-k} :

$$X^{n-k}m(X) = m_0X^{n-k} + m_1X^{n-k+1} + \dots + m_{k-1}X^{n-1} \quad (\text{ecuación 19})$$

Si este polinomio es dividido por $g(X)$ la expresión final sería:

$$X^{n-k}m(X) = q(X)g(X) + r(X) \quad (\text{ecuación 20})$$

donde $r(X)$ es igual a:

$$r(X) = r_0 + r_1X + r_2X^2 + \dots + r_{n-k-1}X^{n-k-1} \quad (\text{ecuación 21})$$

El polinomio $r(X)$ puede entonces expresarse de acuerdo a la ecuación 20 como la operación modulo sobre $g(X)$:

$$r(X) = X^{n-k}m(X) \text{ modulo } g(X) \quad (\text{ecuación 22})$$

Sumando $r(X)$ a ambos lados de la expresión se obtiene:

$$X^{n-k}m(X) + r(X) = q(X)g(X) = U(X) \quad (\text{ecuación 23})$$

De la ecuación se deduce que el polinomio del lado derecho es un polinomio que representa a una palabra de código, debido a que puede expresarse como el producto de cierto polinomio $q(X)$ multiplicado por $g(X)$. El resto de la división entre este polinomio y $g(X)$ es por lo tanto cero.

Así el polinomio que representa un vector de código adopta la forma sistemática:

$$r(X) + X^{n-k}m(X) = r_0 + r_1X + r_2X^2 + \dots + r_{n-k-1}X^{n-k-1} + m_0X^{n-k} + m_1X^{n-k+1} + \dots + m_{k-1}X^{n-1} \quad (\text{ecuación 24})$$

La palabra de código asociada

$$U = (r_0 \ r_1 \ \dots \ r_{n-k-1}, m_0 \ m_1 \ \dots \ m_{k-1}) \quad (\text{ecuación 25})$$

Que aparece presentada en forma sistemática.

Ejemplo: Determinar el vector de código correspondiente a la palabra de mensaje $m(1 \ 0 \ 1 \ 1)$ si el polinomio generador es $g(X) = 1+X+X^3$, en un código cíclico (7,4).

El polinomio se obtiene por división de $X^{n-k} \cdot m(X)$ por $g(X)$.

$$m(X) = 1+X^2+X^3.$$

$$X^3 \cdot m(X) = X^3+X^5+X^6.$$

Dividiendo se tiene:

$$X^3+X^5+X^6 = (1+X+X^2+X^3)(1+X+X^3) + 1 = q(X) \cdot g(X) + 1$$

$$U(X) = r(X) + X^3 m(X) = 1 + X^3 + X^5 + X^6$$

$$U = (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1)$$

Implementación circuital de divisores de polinomios.

La codificación en códigos cíclicos de bloques de tipo lineal requiere de la división polinómica. Esta operación puede realizarse circuitalmente con registros de desplazamiento realimentados.

Dados los polinomios $v(X)$ y $g(X)$:

$$V(X) = v_0 + v_1 X + v_2 X^2 + \dots + v_m X^m \quad (\text{ecuación 26})$$

$$g(X) = g_0 + g_1 X + g_2 X^2 + \dots + g_r X^r \quad (\text{ecuación 27})$$

Siendo en general $m \geq r$.

La división de estos polinomios establece el valor del cociente y del resto de la misma:

$$\frac{V(X)}{g(X)} = q(X) + \frac{r(X)}{g(X)} \quad (\text{ecuación 28})$$

Un circuito como el de la figura 1 realiza la operación de división.

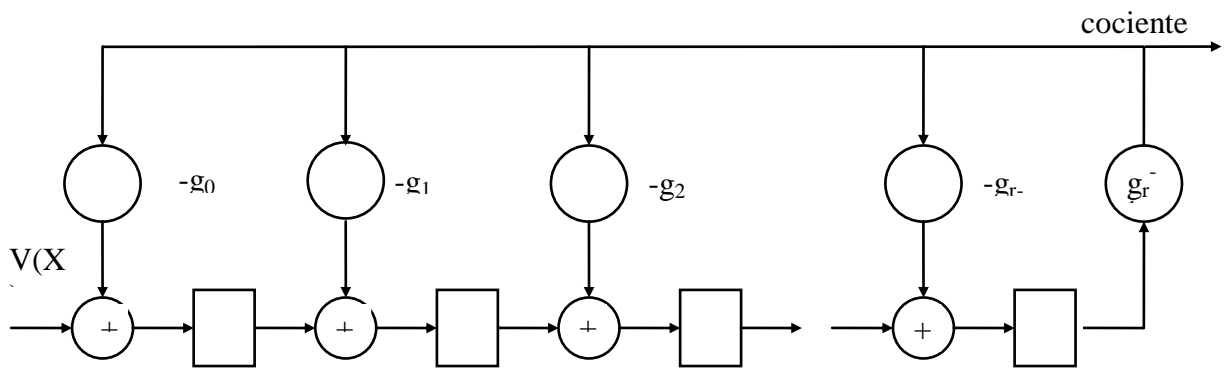


Figura 1

Las etapas iniciales de los registros son llevadas a cero. Los primeros r desplazamientos generan el ingreso de los bits mas significativos de V(X). Luego del ingreso del r-esimo bit la salida del cociente es $g_r^{-1} v_m$. Este es el termino de mas alto orden del cociente. Para cada coeficiente del cociente q_i el polinomio $q_i g(X)$ tiene que ser restado del dividendo.

Las conexiones de realimentacion realizan esta sustracción. La diferencia entre los r bits de la izquierda que permanecen en los registros y los términos de realimentacion $q_i g(X)$ se realiza en cada desplazamiento y aparece como los contenidos del registro. Después de m+1 desplazamientos en el registro, el cociente se presenta en serie sobre el terminal de salida, y el resto de la división reside en los registros de desplazamiento.

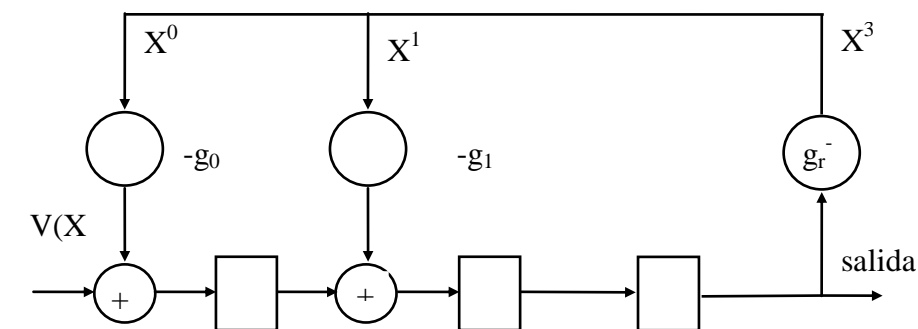
Ejemplo:

Efectuar la división del polinomio $V(X) = X^3 + X^5 + X^6$ por el polinomio generador $g(X) = 1 + X + X^3$

La operación a realizar por el circuito es de la forma:

$$\frac{X^3 + X^5 + X^6}{1 + X + X^3} = q(X) + \frac{r(X)}{1 + X + X^3}$$

El circuito de división se presenta en la figura 2.



0 0 0 1 0 1

Figura 2

La evolución de los registros es la siguiente:

E								Desp l		R				salida
0	0	0	1	0	1	1		1		0	0	0		-
	0	0	0	1	0	1		1		1	0	0		0
		0	0	0	1	0		2		1	1	0		0
			0	0	0	1		3		0	1	1		0
				0	0	0		4		0	1	1		1
					0	0		5		1	1	1		1
						0		6		1	0	1		1
						-		7		1	0	0		1

Después del cuarto desplazamiento los coeficientes de la salida cociente q_i son 1 1 1 1, que corresponde al polinomio cociente $q(X) = 1+X+X^2+X^3$. Los coeficientes del resto son 1 0 0, y el polinomio resto es $r(X) = 1$.

Codificación sistemática con un registro de desplazamiento de (n-k) etapas.

La codificación de una palabra de código cíclico en forma sistemática consiste de la realización de la división del polinomio $X^{n-k} m(X)$ por $g(X)$. El resto de esta división genera los bits de paridad. La palabra de mensaje tiene que ser entonces desplazada hacia arriba para permitir la ubicación de los bits de paridad. El desplazamiento hacia arriba de la palabra de mensajes (n-k) posiciones es una operación trivial previa que no necesita ser tenida en cuenta en la división en si. En realidad solo los bits de paridad se calculan. Posteriormente se los ubica en la posición adecuada dentro de la palabra de acuerdo a la forma sistemática de codificación. El polinomio de los bits de paridad es obtenido como el resto de la división después de n desplazamientos sobre los (n-k) registros de desplazamiento. Los primeros (n-k) desplazamientos son simplemente desplazamientos para llenar los registros. No habrá realimentación hasta que el registro mas a la derecha no se haya llenado. El termino realimentado a la izquierda es la suma del termino de la etapa a la derecha con la entrada. Esta suma se genera solo si $g_0 = g_{n-k} = 1$ para cualquier polinomio generador $g(X)$. El polinomio generador es igual a:

$$g(X) = 1 + g_1 X + g_2 X^2 + \dots + g_{n-k-1} X^{n-k-1} + X^{n-k}$$

(ecuación 29)

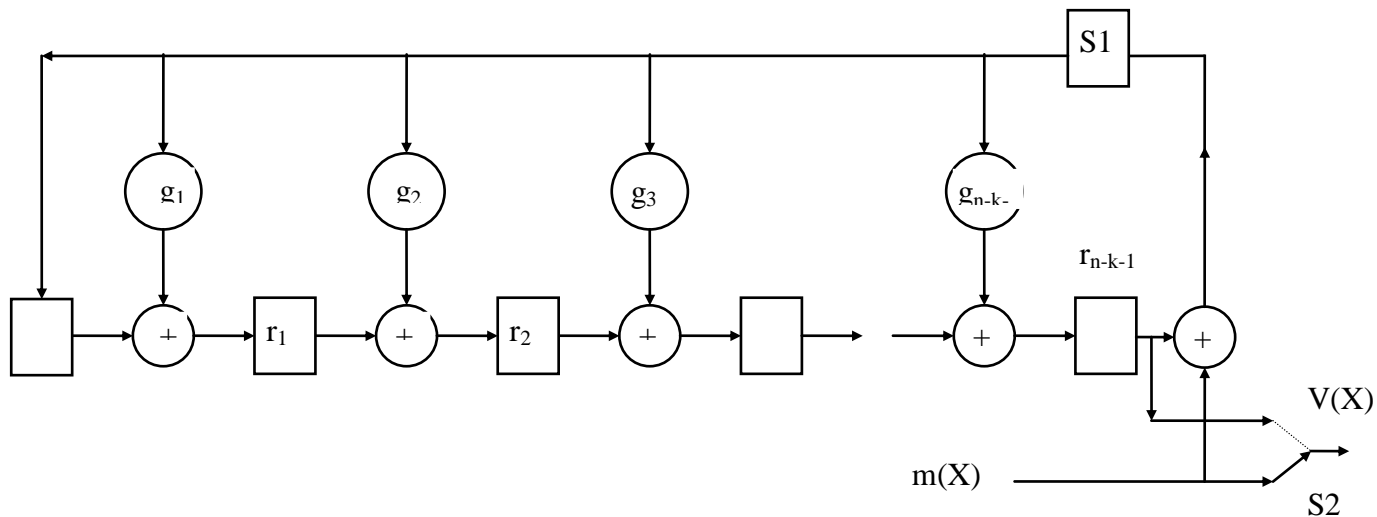


Figura 3

El procedimiento de codificación es el siguiente:

Cuando el switch S1 está cerrado los primeros k desplazamientos tienen lugar. Esto permite la transmisión de los bits de mensaje en las $(n-k)$ etapas del registro de desplazamiento.

El switch S2 está en la posición inferior permite la transmisión de los bits de mensaje sobre la salida durante los primeros k desplazamientos.

Luego de transmitir los primeros k bits el switch S1 se abre y el switch S2 se conecta al borne superior.

Los restantes $(n-k)$ desplazamientos limpian los registros desarrollando la palabra de bits de paridad sobre la salida.

El número total de desplazamientos es n , y los contenidos del registro de salida es la palabra de código, que en forma polinómica es $r(X) + X^{n-k} \cdot m(X)$.

Ejemplo:

Con el circuito de la figura 5.9 obtenga la palabra de código correspondiente al vector de mensaje $m = (1\ 0\ 1\ 1)$ en un código $(7,4)$ cíclico lineal de bloques. Emplee el polinomio generador $g(X) = 1 + X + X^3$.

$$m = (1\ 0\ 1\ 1)$$

$$m(X) = 1 + X^2 + X^3$$

$$r(X) = (X^3 + X^5 + X^6) \text{ modulo } (1 + X + X^3)$$

Para los primeros 3 registros de cambio de codificación las operaciones son las siguientes:

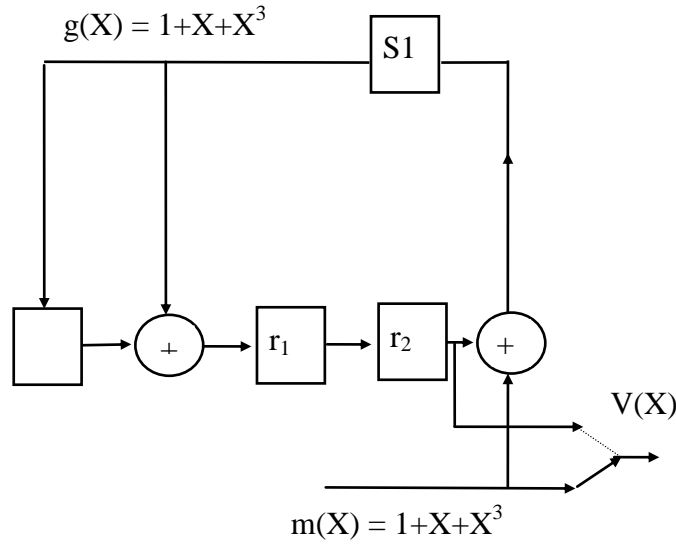


Figura 4

E					despl.		R				Sal.
1	0	1	1		0		0	0	0		-
	1	0	1		1		1	1	0		1
		1	0		2		1	0	1		1
			1		3		1	0	0		0
			-		4		1	0	0		1

Después del cuarto desplazamiento el switch S1 se abre, el switch S2 se coloca en la posición superior, y los bits de paridad se desplazan sobre la salida. El vector de salida es $U = (1\ 0\ 0\ 1\ 0\ 1\ 1)$, que en forma polinomial es igual a $U(X) = 1 + X^3 + X^5 + X^6$.

Detección de error con un registro de desplazamiento de (n-k) etapas.

Para un dado vector de código expresado polinomialmente $U(X)$ el ruido presente en el canal de transmisión lo transforma en una versión con errores $Z(X)$. Por ser palabra de código U corresponde a un polinomio $U(X)$ que es múltiplo del polinomio generador, es decir:

$$U(X) = g(X).m(X) \tag{ecuación 30}$$

el vector con errores esta representado por un polinomio $Z(X)$ de forma tal que:

$$Z(X) = U(X) + e(X) \tag{ecuación 31}$$

Donde $e(X)$ es el polinomio que representa el patrón de error existente. La detección de error se realiza también por síndrome. El decodificador realiza la división del vector recibido por $g(X)$ y observa si el resto de la división es cero. Por lo tanto el síndrome $S(X)$ es el resto de la división de $Z(X)$ por $g(X)$:

$$Z(X) = q(X).g(X) + S(X) \tag{ecuación 32}$$

Donde $S(X)$ es un polinomio de grado $(n-k-1)$ o menor. Combinando las expresiones anteriores se tiene:

$$e(X) = [m(X) + q(X)]g(X) + S(X) \tag{ecuación 33}$$

El síndrome calculado dividiendo $Z(X)$ por $g(X)$ es el mismo que el calculado dividiendo $e(X)$ por $g(X)$. El síndrome contiene entonces toda la información necesaria para la corrección del error. La detección de errores implica entonces hacer una división polinómica. El circuito de la figura realiza tal división.

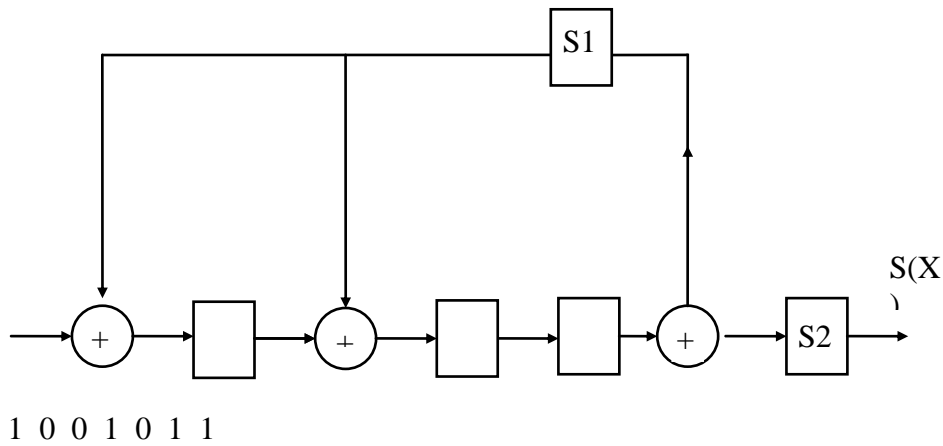


Figura 6.

El circuito de decodificación es similar al de codificación. El primer proceso sucede con el switch $S1$ cerrado, y el $S2$ abierto. El vector recibido es desplazado sobre los registros, que inicialmente están en valor cero. Luego de que el vector de entrada ingresa a los registros, el contenido de los registros es el síndrome. Luego el switch $S1$ se abre, y el $S2$ se cierra, permitiendo desplazar el vector sobre la salida. La siguiente tabla describe el desarrollo de la decodificación:

E								Despl.		R		
1	0	0	1	0	1	1		0		0	0	0
	1	0	0	1	0	1		1		1	0	0
		1	0	0	1	0		2		1	1	0
			1	0	0	1		3		0	1	1
				1	0	0		4		0	1	1
					1	0		5		1	1	1
						1		6		1	0	1
						-		7		0	0	0

Si el síndrome decodificado es el vector todos ceros, la palabra decodificada se acepta como válida. Para corregir errores se puede confeccionar una tabla de síndromes y patrones y corregir de la misma forma que se hace en todo código de bloques lineal.



RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-
NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).