

**ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA ELÉCTRICA Y  
ELECTRÓNICA**

**MATERIAL DIDÁCTICO PARA EL ESTUDIO Y SIMULACIÓN DE  
CDMA: APLICACIÓN A COMUNICACIONES MÓVILES**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA Y TELECOMUNICACIONES.**

**ARÉVALO ANCHUNDIA CHRISTIAN DANIEL**  
chi.are.a@gmail.com

**ROJAS URBANO JONNATHAN JAIR**  
rojoas@yahoo.es

**DIRECTOR: DR. ROBIN ÁLVAREZ RUEDA**  
robin.alvarez@epn.edu.ec

**Quito, Mayo de 2011**

## DECLARACIÓN

Nosotros, **Arévalo Anchundia Christian Daniel** y **Rojas Urbano Jonnathan Jair**, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

Christian Daniel Arévalo Anchundia.

---

Jonnathan Jair Rojas Urbano.

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Arévalo Anchundia Christian Daniel y Rojas Urbano Jonnathan Jair, bajo mi supervisión.

---

Dr. Róbin Álvarez Rueda  
DIRECTOR DEL PROYECTO

## **AGRADECIMIENTOS**

Quiero agradecer a mis padres que me han brindado todo su apoyo y paciencia durante toda mi vida. A mis hermanos que me han ayudado con su compañía y consejos. A mi compañero Jonnathan, junto a quien he realizado este proyecto y me ha brindado todo su esfuerzo y dedicación. A mis amigos gabrielinos, que han permitido desarrollarme como persona y con quienes he pasado buenos momentos durante mi vida en el colegio y en la universidad. A mis amigos y amigas de mi carrera que me acompañaron en esta etapa y se han vuelto grandes compañeros. A mis sobrinos que me han brindado sonrisas y alegría desde su nacimiento. Al Dr. Robin Álvarez por su ayuda constante e incondicional durante todo este proyecto. A la Ing. María Soledad Jiménez y al Ing. Tarquino Sánchez, por darnos un gran aporte para el desarrollo y corrección del proyecto. A todos mis amigos y amigas que me han ayudado a crecer y ser mejor persona cada día.

**Christian**

Deseo dejar constancia de mi más profundo y sincero agradecimiento al Dr. Robin Álvarez, como Director de Tesis quien con su ayuda, comprensión y desinteresada colaboración, formó parte de la realización del presente proyecto de titulación. A la Ing. María Soledad Jiménez y al Ing. Tarquino Sánchez, por su tiempo y dedicación prestada para la revisión del presente proyecto de Titulación. Y finalmente a mi compañero Christian por su esfuerzo para concluir el trabajo que nos validará como ingenieros.

**Jonnathan**

## DEDICATORIA

Dedico este proyecto a Dios que me ha brindado una gran vida junto a una gran familia y grandes amigos. A mis padres que me han apoyado en todos los momentos con su apoyo y amor. A mis hermanos, grandes compañeros de mi vida. A mis sobrinos, que desde su nacimiento han sido una de mis más grandes alegrías.

**Christian.**

El presente proyecto de titulación se lo dedico a Dios, pues me ha brindado inteligencia, sabiduría y salud para concluir con éxito mi carrera profesional. A mis padres, por su ejemplo, dedicación, paciencia, sacrificio y amor, pues siempre estuvieron a mi lado en los momentos más difíciles y supieron guiarme. A Martín y Andrea, que se convirtieron en mi motivación para crecer y superarme todos los días. A mis hermanos Javier y Mariuxi por su paciencia, comprensión y apoyo. A mis familiares y amigos por su fuerza y aliento para conseguir este objetivo.

**Jonnathan.**

## CONTENIDO

RESUMEN	xiv
PRESENTACIÓN	xv
<b>INTRODUCCIÓN</b>	
1.1 Ventajas de CDMA	2
1.2 Canal Inalámbrico	2
1.2.1 Ruido Aditivo	3
1.2.2 Ruido Multiplicativo	5
1.3 Codificación del canal y Recepción en CDMA	11
1.3.1 Codificación CRC	11
1.3.2 Codificador Convolutacional	12
1.3.3 Decodificador de Viterbi	19
1.3.4 Interleaving	24
1.4 Receptor Rake 24	
1.5 Correlación Cruzada y Autocorrelación de Series Temporales	25
<b>ANÁLISIS DE SECUENCIAS DENTRO DE CDMA</b>	
2.1 Secuencias No Ortogonales	45
2.1.1 Secuencias Pseudoaleatorias o Secuencias PN	46
2.1.2 Secuencias de Longitud Máxima o M – Secuencias	69
2.1.3 M-Secuencias Preferidas	86
2.1.4 Secuencias Gold	96
2.1.5 Secuencias Kasami	107
2.2 Secuencias Ortogonales	124
2.2.1 Secuencias de Hadamard-Walsh	125
2.2.2 Códigos OVVSF	131

2.3	Secuencias de Sincronización	135
2.3.1	Secuencias de Barker	136

## **ANÁLISIS DE LAS TÉCNICAS DE ESPECTRO ENSANCHADO DE CDMA**

3.1	Historia de los Sistemas de Espectro Ensanchado	142
3.2	Tipos de Sistemas de Espectro Disperso	143
3.3	Técnicas de Modulación Digital Asociadas con Espectro Ensanchado	143
3.4	Análisis de los Sistemas de Secuencia Directa (Direct Sequence)	144
3.4.1	Generalidades de un Sistema DS-SS	144
3.4.2	Análisis Matemático del Transmisor de Sistemas de Secuencia Directa	146
3.4.3	Caracterización de los Sistemas de Espectro Ensanchado de Secuencia Directa	149
3.4.4	Recuperación de la Señal Digital Original	151
3.4.5	Aplicaciones de sistemas DS-SS	179
3.5	Análisis de los Sistemas de Salto de Frecuencia (Frequency Hopping Spread Spectrum): FH-SS	187
3.5.1	Generalidades de los Sistemas FH-SS	187
3.5.2	Tipos de FH-SS	189
3.5.3	Tipos de sistemas FH-SS Aleatorios	190
3.5.4	Simulación y Análisis en MATLAB sobre un sistema FH-SS	191
3.5.5	Simulación de un Sistema Bluetooth que emplea FH-SS en Simulink	200
3.5.6	Caso con Ruido Simulink	210
3.5.7	Caso con Interferencia Simulink	213
3.5.8	Comparación entre un sistema DS-SS y un sistema FH-SS	216

## **DETECCIÓN MULTIUSUARIO Y ESTUDIO DE LA EVOLUCIÓN DE CDMA EN LAS COMUNICACIONES MÓVILES**

4.1	detección multiusuario	218
4.1.1	Detección Óptima Multiusuario	219
4.1.2	Detectores lineales subóptimos	227
4.1.3	Esquemas de combate a Interferencia	231
4.1.4	Técnicas de cancelación de interferencia (IC)	252
4.2	aspectos básicos del ESTÁNDAR IS-95A	257
4.2.1	Configuración de una Red IS-95A	258
4.2.2	Vocoder	261
4.2.3	Códigos utilizados en IS95-A	262
4.2.4	Tasas de transmisión y Rate Sets en IS-95A	265
4.2.5	Potencia Total en el Enlace Forward	266
4.2.6	Control de Potencia	266
4.2.7	Canales Forward y Reverse	270
4.2.8	Canales Forward	271
4.2.9	Canales Reverse	380
4.3	Evolución de cdma en las comunicaciones móviles	415
4.3.1	CDMA/cdmaOne	415
4.3.2	Estándar CDMA2000	417
4.3.3	CDMA2000 1xEV	418

## **CONCLUSIONES Y RECOMENDACIONES**

5.1	Conclusiones	421
5.2	Recomendaciones	429

## **REFERENCIAS BIBLIOGRAFICAS**

## **ANEXOS**



## RESUMEN

Debido a que en la carrera no se estudia la materia de Procesamiento Digital de Señales de forma obligatoria, no se tiene un buen entendimiento de las tecnologías inalámbricas actuales de una forma práctica.

Con el presente proyecto se busca ofrecer a los estudiantes una herramienta sólida de aprendizaje de los aspectos básicos utilizados en las comunicaciones móviles CDMA, a través de material didáctico y simulaciones que permitan un mejor entendimiento de la materia referidos principalmente a los aspectos de Indetectabilidad, Seguridad y Acceso Multiusuario característicos de los sistemas CDMA.

De esta manera se realiza un análisis de los sistemas de espectro ensanchado, características esenciales de los códigos utilizados en sistemas CDMA, para entender la aplicación a sistemas de comunicación móviles a través del estándar IS-95A.

Por lo tanto, este análisis constituirá las bases para el desarrollo de posteriores proyectos sobre tecnologías de tercera y cuarta generación (3G y 4G). Además, se constituirá en una herramienta para el entendimiento de CDMA y su futura aplicación en FPGAs en la Escuela Politécnica Nacional.

## PRESENTACIÓN

El objetivo fundamental del presente proyecto es realizar un análisis y simulación de la técnica de Acceso al Medio CDMA. Enfocándose en los aspectos básicos de esta técnica de una manera didáctica y comprensiva para el lector.

En el capítulo 1, se describe de una manera general a la técnica de Acceso CDMA. Además se mencionan conceptos básicos del canal inalámbrico, codificación del canal, recepción y las funciones de correlación necesarias para entender la teoría de Secuencias.

En el capítulo 2 se realiza una breve descripción de las debilidades de la modulación analógica y digital. Por lo tanto se ve la necesidad de crear un sistema que cumpla con requerimientos de Indetectabilidad, seguridad y que tenga acceso multiusuario. Partiendo de estas premisas, se realiza el análisis y simulación de las secuencias o códigos que se utilizan en sistemas CDMA para conseguir estos objetivos.

En el capítulo 3 se realiza el análisis y simulación de la estructura del transmisor y receptor de las técnicas de Espectro Ensanchado DSSS y FHSS.

En el capítulo 4 se mencionan las técnicas para la detección multiusuario. Además se mencionan los aspectos básicos de la aplicación de CDMA en comunicaciones móviles, a través del estándar IS-95A. Estos aspectos son necesarios para realizar el análisis de los demos implementados en MATLAB 7.0 del canal de tráfico en el enlace Forward y Reverse. El análisis de los demos, se los hace paso a paso para una mejor comprensión.

En el Capítulo 5, se muestran las conclusiones y recomendaciones obtenidas de la realización del proyecto de titulación.

# CAPÍTULO I

## 1. INTRODUCCIÓN

CDMA es una tecnología que permite que un elevado número de comunicaciones de voz o datos, compartan el canal simultáneamente. Este sistema ofrece mejores características que FDMA y TDMA en cuanto a cobertura, calidad y capacidad. Además es más flexible puesto que puede soportar simultáneamente múltiples codificadores de voz, a la vez que nuevos codificadores pueden ser incorporados sin necesidad de hacer mayores cambios en la infraestructura del sistema.

CDMA utiliza la técnica de espectro ensanchado, donde se asignan códigos únicos a cada usuario y se aprovecha de mejor manera el espectro de frecuencia. En el receptor se necesita de este código único para recuperar la información.

CDMA es una tecnología que permite que un elevado número de comunicaciones de voz o datos, compartan el canal simultáneamente. Esto lo logra mediante el uso de códigos únicos para cada usuario que transforman la señal del usuario en una señal de banda ancha. Si el receptor recibe múltiples señales de banda ancha, se usará el código asignado a un usuario en particular para recuperar la información original.

Durante este proceso de recuperación de la información, la potencia de la señal es comprimida dentro del ancho de banda original, mientras que las otras señales de banda ancha del resto de usuarios aparecen como ruido ante la señal deseada.

## 1.1 VENTAJAS DE CDMA

- Mayor capacidad: CDMA permite que un mayor número de usuarios comportan el mismo ancho de banda, al mismo tiempo.
- Seguridad y Privacidad: Es muy difícil capturar y descifrar una señal.
- Reducción del ruido y la interferencia: Al hacer uso de códigos, es posible aumentar la potencia de las señales sin que estas se interfieran.

Para el estudio de CDMA, es necesario conocer algunos conceptos básicos como el medio de transmisión, esquemas de codificación del canal y recepción.

## 1.2 CANAL INALÁMBRICO

Un canal inalámbrico altera la señal transmitida y produce que algunos bits enviados sean diferentes a los bits que llegan hasta el receptor, produciendo una tasa de bits errados.

Las causas de dichas anomalías son muy variadas: térmicas, propias del receptor, otras señales en las mismas frecuencias de la señal transmitida, denominadas interferencias. También existen otras anomalías debido a las características físicas del medio por el cual se propaga la información: reflexiones en múltiples objetos tanto fijos como móviles que están entre el transmisor y el receptor, difracciones en dichos objetos que producen efectos de sombra y penumbra, absorción de aquellos objetos, reflexión en la superficie del terreno que puede ser regular o irregular tanto a nivel microscópico como macroscópico que puede afectar a la señal según la longitud de onda de la señal de información.

Los efectos del canal inalámbrico son estudiados a fondo, tratando de comprender su comportamiento y desarrollando técnicas que permitan reducir su efecto, es decir, reducir la tasa de bits errados.

Dentro de las técnicas que permiten reducir dicha tasa de bits errados, se encuentran: codificación, entrelazado (Interleaving), receptores especializados, etc.

Por lo antes mencionado, el canal inalámbrico puede alterar a la señal que se propaga por él, por causa de dos fenómenos principalmente: ruido multiplicativo y ruido aditivo.

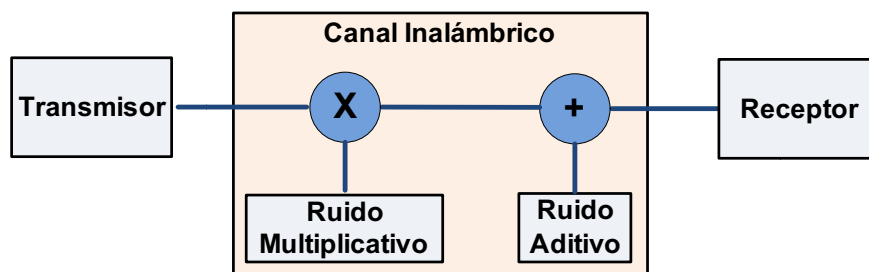
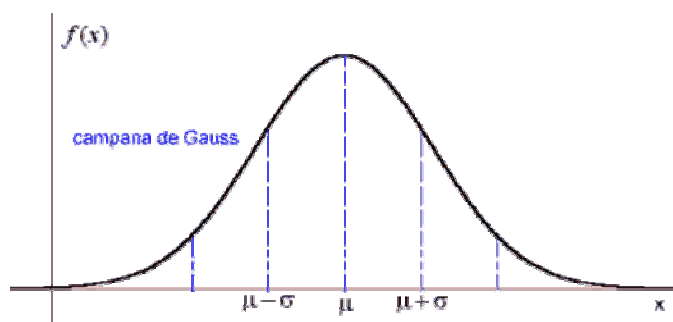


Figura 1.1 Ruidos presentes en Canal Inalámbrico.

### 1.2.1 RUIDO ADITIVO

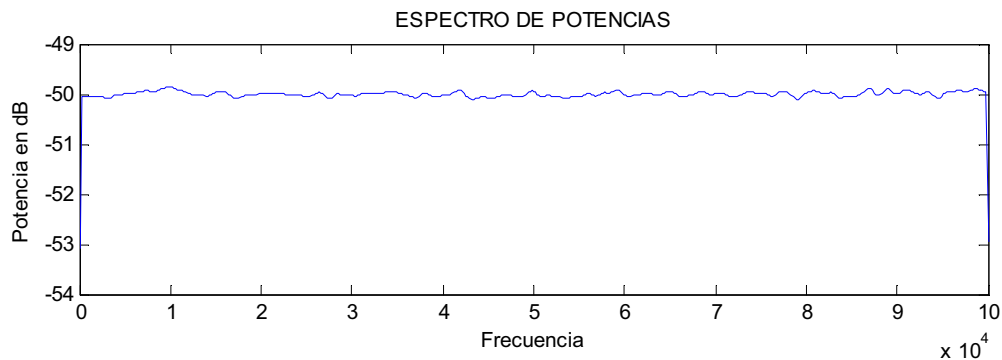
El ruido aditivo, llamado así por sumarse a la señal transmitida, está constituido por contribuciones aleatorias, entre otras de tipo térmico, provenientes de diversas fuentes, cada una de las cuales tiene una función de distribución de probabilidad (FDP) determinada, algunas asimétricas hacia la izquierda, otras simétricas y otras asimétricas hacia la derecha. La suma de todas estas fuentes, por el teorema del límite central, produce una FDP resultante de tipo simétrico por lo que se denomina ruido gaussiano (por parecerse a la distribución Gaussiana o Normal).



$$y = f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Figura 1.2 Función de distribución de Probabilidad Gaussiana.

Como el ruido aditivo es de carácter aleatorio, contiene, en teoría, un rango infinito de componentes de frecuencia, por lo que se denomina blanco (por parecerse al espectro de la luz blanca que ocupa un rango de frecuencias muy amplio correspondiente a la unión de los espectros de todos los colores).



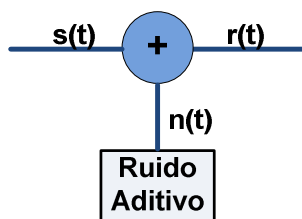
**Figura 1.3** Espectro de potencia del ruido AWGN.

Por lo tanto, la denominación completa es ruido aditivo Gaussiano blanco (Additive White Gaussian Noise o AWGN).<sup>32</sup>

La potencia el ruido se suma a la potencia de la señal transmitida:

$$r(t) = s(t) + n(t) \quad (\text{Ec 1.1})$$

Y las distorsiones que provoque este ruido a la señal dependen de la potencia del mismo en comparación con la potencia de la señal  $s(t)$ .



**Figura 1.4** Ruido Aditivo en la señal.

La calidad de un sistema de comunicación está dada por varios parámetros. Uno de estos es la **relación señal a ruido**. Otro parámetro, es la relación entre Energía de bit ( $E_b$ ) y la potencia del ruido ( $N_0$ ).

En un sistema de comunicación analógico, la relación señal a ruido, es expresada como la relación entre sus potencias:

<sup>32</sup> [http://www.virtual.unal.edu.co/cursos/sedes/manizales/4040051/html/capitulos/cap\\_ii/ruido.html](http://www.virtual.unal.edu.co/cursos/sedes/manizales/4040051/html/capitulos/cap_ii/ruido.html)

$$SNR = \frac{P_S}{P_N} \quad (\text{Ec 1.2})$$

Donde  $P_S$  es potencia de la señal y  $P_N$  es la potencia del ruido.

En un sistema de comunicación digital, se transmiten bits y en este caso, la medida de la calidad de comunicación se mide por la relación entre la energía de bit ( $E_b$ ) y la potencia de ruido ( $N_0$ ). Si se ha introducido modulación multiportadora en la que se agrupan varios bits para conformar un símbolo, la calidad de comunicación se mide por la relación entre la energía de la señal ( $E_s$ ) y la potencia de ruido ( $N_0$ ). La relación entre estos parámetros es la siguiente:

$$\frac{E_s}{N_0} (dB) = \frac{E_b}{N_0} (dB) + 10 \log_{10}(m) \quad (\text{Ec 1.3})$$

Donde  $m$  es el número de bits por cada símbolo.<sup>33, 34</sup>

En los Capítulos 3 y 4 se observará el efecto del ruido aditivo sobre las comunicaciones en sistemas CDMA.

## 1.2.2 RUIDO MULTIPLICATIVO

El ruido multiplicativo se da considerando dos casos:

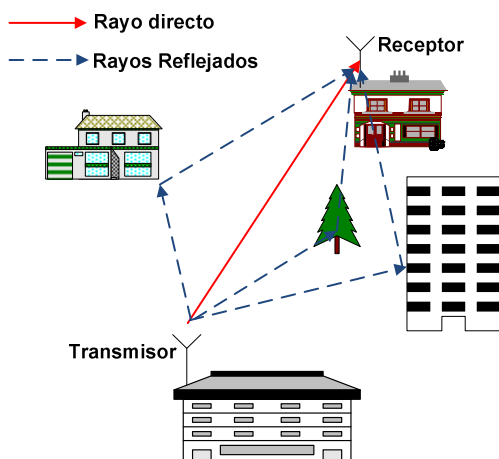
- **Comunicación fija:** Cuando el transmisor y el receptor están fijos.
- **Comunicación móvil:** Cuando el transmisor está fijo y el receptor se mueve.

En el primer caso, la señal que sale desde el transmisor, puede llegar hasta el receptor tanto de manera directa (rayo directo) como por reflexiones en los distintos objetos alrededor al receptor y/o al transmisor (rayos reflejados):

---

<sup>33</sup> <http://sss-mag.com/ebn0.html>

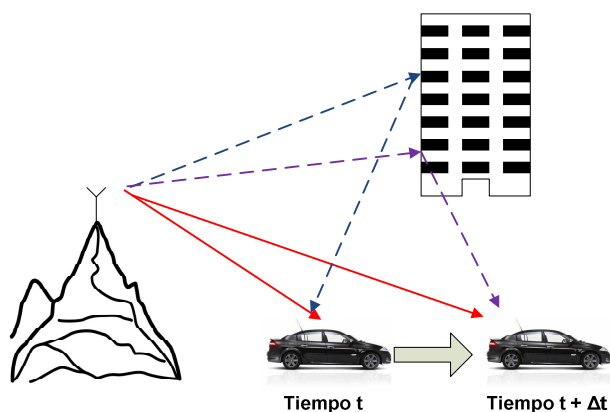
<sup>34</sup> <http://www.slideshare.net/gbermeo/ruido-en-sistemas-de-comunicaciones-presentation>



**Figura 1.5** Escenario de un canal inalámbrico con rayos reflejados de la señal transmitida.

Existe una onda aleatoria de amplitud relativamente pequeña (denominada de pequeña escala) que estaría variando sobre su promedio de valor grande respecto de la onda aleatoria (denominado valor de gran escala).

En el segundo, el escenario será similar en cuanto a la componente de pequeña escala mientras que el valor medio (componente de gran escala) de la señal recibida, dependería de la distancia entre el Tx y el Rx, siendo menor a medida que aumente esta distancia. Este valor promedio se denomina media local.



**Figura 1.6** Desvanecimiento producido por la distancia.

En cuanto a la componente de gran escala, es importante su predicción cuando se realiza el diseño de radioenlaces para conocer las zonas de cobertura cuando estén fijos: la ubicación geográfica del Tx y Rx, las alturas de las antenas, los equipos de comunicación empleados, la zona geográfica de ubicación (clima, tipo



de terreno, etc). Dentro de esto, hay factores fijos que ayudan y otros que se oponen al avance de la onda electromagnética.

En cuanto a la componente de pequeña escala el valor instantáneo de la señal es de tipo estocástico y varía muy rápidamente, por lo que no es de interés. Es de mayor importancia comprender el tipo de Función de Distribución de Probabilidad (FDP) que mejor se ajusta a esta y de ella obtener una medida de su tendencia central (por ejemplo la media) y un valor de su dispersión (por ejemplo la desviación estándar o su varianza).

A diferencia de la mayoría de efectos aleatorios de la naturaleza que se ajustan a una FDP gaussiana o normal, el canal inalámbrico también altera la señal de información de forma que la FDP que mejor se ajusta depende de los dos siguientes casos:

- **Línea de vista:** la FDP que mejor caracteriza a la señal recibida en el receptor es la **FDP de Rice**.
- **Sin línea de vista:** Cuando se realiza la aproximación considerando que la componente de línea de vista tiende a cero, la FDP que mejor caracteriza ésta es una **FDP de Rayleigh**.<sup>35</sup>

### 1.2.2.1 Desvanecimiento y Margen de Desvanecimiento

Al propagarse una onda electromagnética por la atmosfera terrestre, además de la pérdida normal en espacio libre, se presentan pérdidas intermitentes de intensidad de la señal a causa de fenómenos aleatorios.

Esas pérdidas se denominan desvanecimiento (fading) y se producen debido a 3 fenómenos aleatorios:

- Perturbaciones meteorológicas como lluvia, nieve, granizo, etc. Estas influyen en la disponibilidad del enlace.
- Fenómenos físicos como la difracción,

---

<sup>35</sup> [http://neutron.ing.ucv.ve/revista-e/No2/L\\_Chui.html](http://neutron.ing.ucv.ve/revista-e/No2/L_Chui.html)

- A propagación por trayectorias múltiples o multicamino debido a las reflexiones en objetos circundantes al transmisor y al receptor como montañas, edificios, vehículos, personas, etc.
- A una superficie terrestre irregular (sensibilidad del terreno).

Para tener en cuenta el desvanecimiento, se agrega una pérdida adicional a la pérdida en espacio libre. A esta pérdida se le llama margen de desvanecimiento (fading margin).

Existen momentos en los que el desvanecimiento podría ser tan grande que alcanzara al valor de la sensibilidad del receptor produciéndose pérdida de la comunicación y por tanto errores. Estos errores se llaman errores tipo ráfaga. Ante este tipo de errores en ráfaga, los algoritmos de corrección de errores tanto de bloque como convolucionales pierden su eficacia (son eficaces frente a errores dispersos) por lo que es necesario tener una etapa de disseminación de datos denominada entrelazado (interleaving).

### **1.2.2.2 Propagación por Multicamino**

La propagación por multicamino, se da cuando una señal a transmitir que atraviesa el medio de propagación sufre múltiples fenómenos de reflexión y difracción que provocan la separación y dispersión de los rayos físicos que transportan la energía de la señal, es decir, surgen diferentes caminos para una misma señal. Esto ocurre debido a que en el medio de propagación se tiene presencia e interacción dinámica de diferentes obstáculos. La señal recibida por el móvil en un instante dado es el resultado de la combinación de varias réplicas de la señal original que han llegado al punto de recepción a través de varias rutas. La señal se verá afectada en amplitud, fase y tiempo.

Generalmente por cada objeto se tendría un rayo reflejado. Esta cantidad de objetos puede ser muy grande y se convierte en un problema físico matemático muy complejo denominado desvanecimiento por multicamino. Típicamente, si hay línea de vista, el fenómeno es caracterizado por una FDP de Rice y si no hay línea de vista, por una FDP de (Rayleigh).

### 1.2.2.3 Efecto Doppler<sup>36</sup>

El efecto Doppler aparece cuando el receptor esta en movimiento, y este varía en función de dos condiciones:

- **Si el Tx y/o el Rx se acercan:** la onda que emite el Tx con una frecuencia  $f$ , llega al Rx con una frecuencia cada vez menor que  $f$ .
- **Si el Tx y/o el Rx se alejan:** la onda que emite el Tx con una frecuencia  $f$ , llega al Rx con una frecuencia cada vez mayor que  $f$ .

A los rangos de frecuencia generados, se les conoce como **desplazamiento Doppler**.

Debido a que las componentes reflejadas actúan desde distintos ángulos alrededor del Rx móvil, esto causa un rango de desplazamientos Doppler conocido como espectro Doppler. El máximo desplazamiento Doppler corresponderá a la componente que se opone a la dirección del móvil.

Si se considera un canal estático, en donde el transmisor y el receptor están estáticos, el fenómeno Doppler no existe.

### 1.2.2.4 Modelado de un Escenario Multicamino.<sup>37</sup>

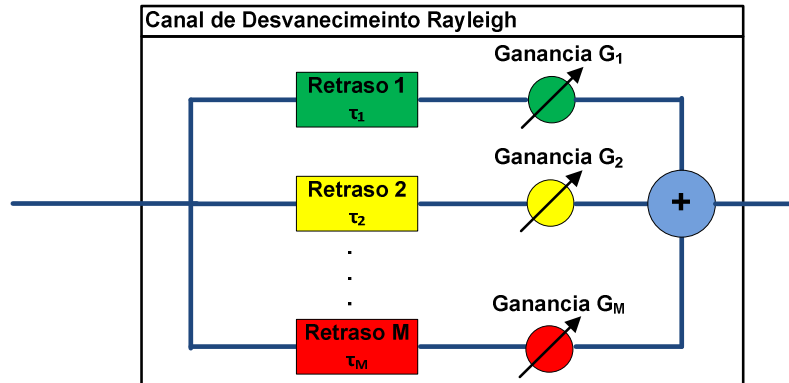
El modelo mediante rayos puede ser visto como una sumatoria de señales: una principal (rayo directo) y otras retardadas y de menor potencia (rayos reflejados). Esta visualización se denomina modelado mediante líneas de retardo (**TDL: *tapped delay line***). En este modelo se representa a cada uno de los caminos de propagación mediante una línea de retardo, con un retardo  $\tau_i$  y ganancia compleja  $a_i$  que se modelan siguiendo las distribuciones de Rice o de Rayleigh con un espectro Doppler definido para cada rayo:

---

<sup>36</sup>

[http://profesores.usfq.edu.ec/renej/Contenidos%20Comunicaciones%20Moviles/Apuntes%20Comunicaciones%20M%F3viles/M%F3viles\\_4.pdf](http://profesores.usfq.edu.ec/renej/Contenidos%20Comunicaciones%20Moviles/Apuntes%20Comunicaciones%20M%F3viles/M%F3viles_4.pdf)

<sup>37</sup> <http://www.mike-willis.com/Tutorial/PF15.htm>



**Figura 1.7** Esquema del modelo de canal multicamino basado en líneas de retardo (TDL).

$$h(t, \tau) = \sum_{i=0}^{M-1} a_i(t) \delta(\tau - \tau_i) \quad (\text{Ec 1.4})$$

Donde  $\tau$  es el retardo temporal y  $t$  la dependencia temporal del canal. La suma total de los retardos representa el  $\tau_{max}$ , y el número de derivaciones dependerá del número de trayectorias de los rayos reflejados y cuánto es el retardo propio de cada uno.

Algunas tecnologías inalámbricas, especifican su desplazamiento Doppler en términos de la velocidad del móvil  $v_f$  (m/s):

$$f_d = \frac{v_f}{c} \quad (\text{Ec 1.5})$$

Donde  $f_d$  es la frecuencia de portadora del transmisor en Hertz y  $c$  es la velocidad de la luz ( $3 \times 10^8$  m/s).

En el Capítulo 4 se observará el efecto del ruido multiplicativo sobre las comunicaciones en sistemas CDMA.

### 1.2.2.5 Efecto del Desvanecimiento Multicanal en la Señal.<sup>38</sup>

Los efectos que puede producir los desvanecimientos multicamino en la señal dependen de las características de la señal. Los desvanecimientos se presentan en la señal de dos maneras:

- Ensanchamiento en tiempo de la señal o dispersión de la señal.
- Variación en tiempo del canal debido al movimiento.

## 1.3 CODIFICACIÓN DEL CANAL Y RECEPCIÓN EN CDMA

La codificación del canal se utiliza para mejorar el rendimiento del sistema de comunicaciones, haciendo la señal más robusta frente a los daños provocados por el ruido y desvanecimiento del canal.

El objetivo de la codificación del canal es reducir la tasa de bits errados (BER), a costa de aumentar el ancho de banda, o el número de bits redundantes que el sistema debe transmitir. Con la codificación de canal, los errores producidos por el canal son mucho más fáciles de encontrar y ser corregidos por el receptor.

Los sistemas CDMA utilizan varias etapas de codificación: CRC, Codificación Convolutional e Interleaver como se observará en el Capítulo 4.

### 1.3.1 CODIFICACIÓN CRC

Un CRC es una función diseñada para detectar cambios accidentales en datos transmitidos. Un dispositivo CRC calcula una secuencia binaria corta y de longitud fija conocida como código CRC o simplemente CRC, para cada trama de datos. El CRC es anexado a la trama de datos. Cuando una trama es leída en el receptor, un dispositivo CRC repite el cálculo de la secuencia; si el nuevo CRC calculado no coincide con el anterior calculado en el transmisor, entonces la trama contiene error de datos y el dispositivo puede tomar acciones correctivas como releer la

---

<sup>38</sup> <http://www.mike-willis.com/Tutorial/PF15.htm>

información o realizar una petición de reenvío de la trama. En caso contrario, la trama ha sido recibida sin errores.

Los sistemas CDMA utilizan la codificación CRC para indicar la calidad de cada trama transmitida.

### 1.3.2 CODIFICADOR CONVOLUCIONAL<sup>39 40</sup>

El codificador convolucional es utilizado para codificar los datos de tal forma que en el receptor se pueda corregir los errores producidos por el canal de comunicaciones.

El codificador convolucional consiste de un conjunto de registros de desplazamiento y la lógica combinacional asociada por medio de la operación XOR. El cálculo de los símbolos codificados depende tanto del conjunto de símbolos de entrada presente, como de algunos símbolos de entrada previos.

Codifica una secuencia de  $k$  bits de entrada en una secuencia de  $n$  bits a diferentes tiempos dependiendo del estado del codificador, es decir que entran  $k$  bits y salen  $n$  bits del codificador. La tasa  $R$  del codificador se define como:

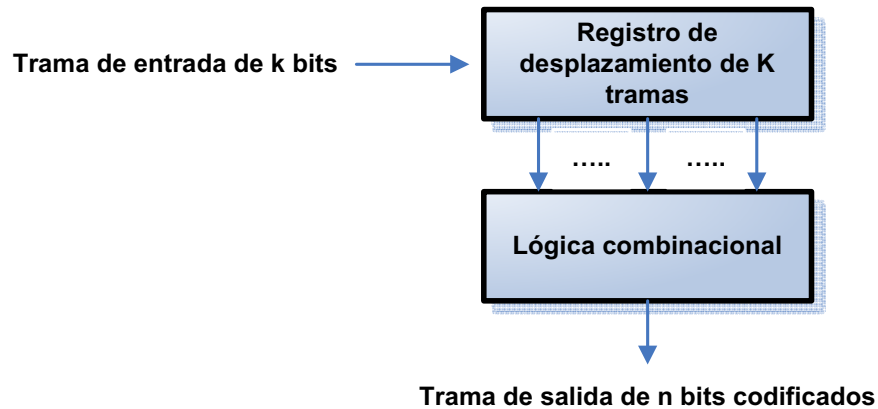
$$R = k / n \quad (\text{Ec 1.6})$$

El registro de desplazamiento consiste de  $K$  etapas (memoria del código o longitud de restricción) donde la entrada binaria es desplazada dentro y a lo largo de los registros,  $k$  bits a la vez. El codificador convolucional se denota como codificador  $(n, k, K)$ . Un diagrama del codificador convolucional se muestra a continuación:

---

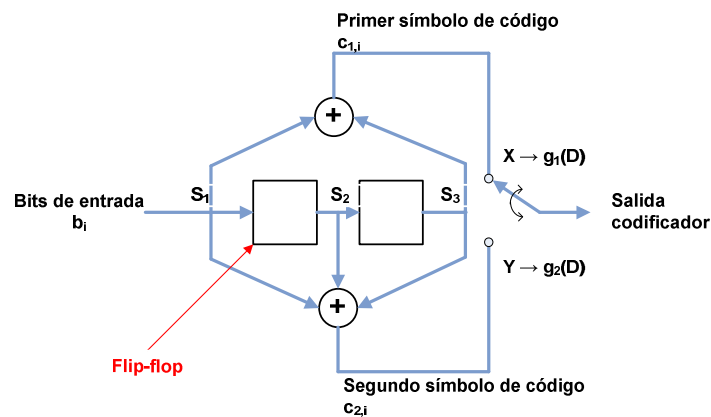
<sup>39</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, p. 107-113.

<sup>40</sup> CDMA Systems Engineering Handbook, Lee, Jhong S.; Miller, Leonard E., p. 848-860.



**Figura 1.8** Diagrama conceptual de codificador convolucional.

A continuación se muestra un codificador convolucional que codifica  $k = 1$  bits de entrada y entrega  $n = 2$  bits de salida, por lo cual su tasa de código es de  $R = 1/2$  y con una memoria del código de  $K = 3$ .



**Figura 1.9** Codificador convolucional (2, 1, 3).

En la figura anterior se considera una secuencia de entrada  $\{b_i\}_{i=0}^{\infty}$ , en donde el codificador convolucional produce 2 flujos (X, Y), los mismos que se calculan utilizando los generadores  $g_1$  y  $g_2$ . Para un caso general de  $n$  flujos los generadores se pueden escribir de la siguiente manera:

$$g_v(D) = \sum_{k=0}^K g_{v,k} D^k \quad (\text{Ec 1.7})$$

Donde  $v = 1, 2, \dots, n$  y  $D$  es una variable que debe ser interpretada como retardo.

En el codificador convolucional mostrado anteriormente, se tienen dos polinomios generadores  $v = 1, 2$ :

$$X \rightarrow g_1(D) = 1 + D^2 \quad (\text{Ec 1.8})$$

$$Y \rightarrow g_2(D) = 1 + D + D^2 \quad (\text{Ec 1.9})$$

Los cuales pueden ser escritos en base octal:

$$X \rightarrow g_1(D) = (101) = 5_{oct} \quad (\text{Ec 1.10})$$

$$Y \rightarrow g_2(D) = (111) = 7_{oct} \quad (\text{Ec 1.11})$$

$S_1, S_2$  y  $S_3$  corresponden a la memoria  $K$  del codificador convolucional y en base a estos valores se obtiene la secuencia codificada.

En la descripción de los codificadores convolucionales suele utilizarse tres representaciones:

- Árboles,
- Trellis
- Diagrama de estados

Las más importantes representaciones son las dos primeras:

### 1.3.2.1 Diagramas de árbol

Para la construcción de un diagrama de árbol se construye la tabla de verdad, que especifica la palabra de código de salida en función del estado de los registros de desplazamiento del codificador.



La profundidad del árbol queda especificada por  $K$  (longitud restringida), y es el número de niveles del árbol hasta que la secuencia se vuelva repetitiva, y se calcula como:

$$X \rightarrow g_1(D) = (101) = 5_{oct} \quad (\text{Ec 1.11})$$

$$P = 2^{(K-1)} \quad (\text{Ec 1.12})$$

Y el número de estados viene dado por:

$$S = 2^{k \cdot (K-1)} \quad (\text{Ec 1.13})$$

El diagrama de árbol se construye considerando los siguientes pasos:

- El estado inicial de los registros de desplazamiento es cero.
- Luego, se avanza en la construcción del árbol de acuerdo a los bits que van entrando al codificador, tomando como regla general que si el bit que entra es un cero se toma la rama superior del árbol, mientras que si el bit que entra es un uno, se toma la rama inferior del árbol.
- El número de ramas se va multiplicado por dos por cada nuevo bit de entrada.
- Al mismo tiempo se va escribiendo el estado de los registros y la salida del codificador.

Para el ejemplo propuesto en la figura 1.9, se construye la tabla de verdad:

**Tabla 1.1** Secuencia de salida del decodificador (2, 1, 3).

$S_1$	$S_2$	$S_3$	$c_1$	$c_2$
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	0
1	1	1	0	1

Donde se diferencia a  $S_1$  como el bit de entrada al codificador convolucional.  $S_2$  y  $S_3$  definen el estado actual del codificador. Por lo tanto hay cuatro tipos de estados para este codificador: **a**, **b**, **c**, **d**.

**Tabla 1.2** Estados del codificador (2, 1, 3).

<b>S2</b>	<b>S3</b>	<b>Estado</b>
0	0	a
0	1	b
1	0	c
1	1	d

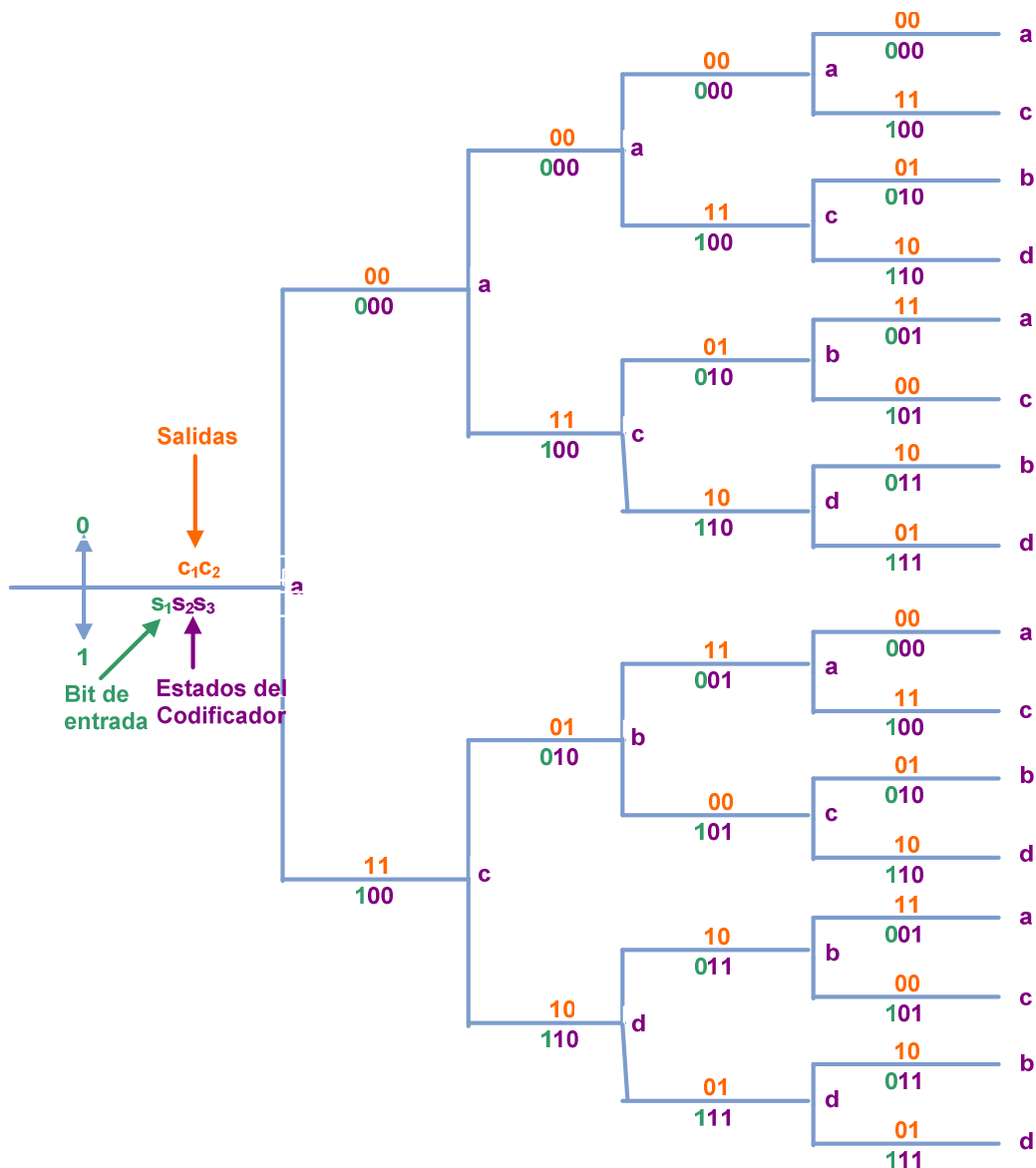
El estado siguiente se define por los valores de  $S_1$  y  $S_2$ , ya que estos se desplazan a la posición de  $S_2$  y  $S_3$  respectivamente, en el siguiente ciclo de reloj, tal como se muestra en la siguiente tabla:

**Tabla 1.3** Estados actuales y siguientes para el decodificador convolucional.

<b>S<sub>1</sub></b>	<b>Estado Actual</b>			<b>Salidas</b>		<b>Estado Siguiente</b>		
	<b>S<sub>2</sub>, S<sub>3</sub></b>			<b>c<sub>1</sub>, c<sub>2</sub></b>		<b>S<sub>2</sub>, S<sub>3</sub></b>		
0	0	0	a	0	0	0	0	a
0	0	1	b	1	1	0	0	a
0	1	0	c	0	1	0	1	b
0	1	1	d	1	0	0	1	b
1	0	0	a	1	1	1	0	c
1	0	1	b	0	0	1	0	c
1	1	0	c	1	0	1	1	d
1	1	1	d	0	1	1	1	d

En la tabla se puede observar que a partir de cualquier estado etiquetado como **a** se producen el mismo par de ramas de salida, salidas 00 y estado **a** y las salidas 11 y estado **c**.

Por lo tanto el diagrama de árbol correspondiente es:



**Figura 1.10** Diagrama de árbol para el codificador (2, 1, 3).

Si se observa este diagrama de árbol, la estructura se repite a sí misma después de la tercera etapa. Esto es debido a que la salida viene dada por el valor del bit actual y dos bits anteriores.

### 1.3.2.2 Diagramas de Trellis

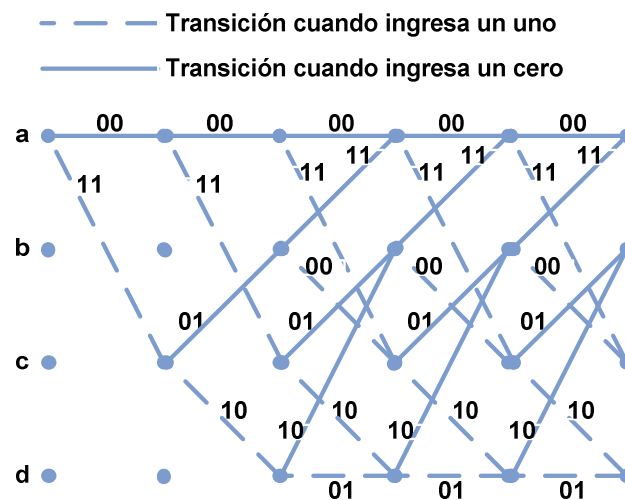
El diagrama de Trellis es un diagrama en forma de red. Cada línea horizontal se corresponde con uno de los estados del codificador. Cada línea vertical se correspondería con uno de los niveles del árbol del código. Este diagrama es el

más utilizado para la descripción de códigos convolucionales, y se construye del hecho de que la estructura del diagrama de árbol es repetitiva. Se parte del estado inicial del codificador en el primer nivel del árbol. A partir de aquí se trazan dos líneas desde este estado. Una para el caso de que la siguiente entrada fuera un **0** y otra para el caso de que fuera un **1**. Estas líneas irán hasta el siguiente nivel del árbol al estado en el que queda el codificador después de haber codificado las correspondientes entradas. Encima de cada una de estas líneas se escribe la salida del codificador para esa codificación. Para la construcción del diagrama de Trellis, se usa como guía el diagrama de árbol y su tabla de verdad:

**Tabla 1.4** Estados actuales y siguientes para el decodificador convolucional.

$S_1$	Estado Actual		Salidas	Estado Siguiente				
	$S_2, S_3$	$S_2, S_3$		$C_1, C_2$	$S_2, S_3$			
0	0	0	a	0	0	0	0	a
0	0	1	b	1	1	0	0	a
0	1	0	c	0	1	0	1	b
0	1	1	d	1	0	0	1	b
1	0	0	a	1	1	1	0	c
1	0	1	b	0	0	1	0	c
1	1	0	c	1	0	1	1	d
1	1	1	d	0	1	1	1	d

El diagrama de Trellis para el codificador convolucional analizado es el siguiente:



**Figura 1.11** Diagrama de Trellis del codificador (2, 1, 3).

### 1.3.3 DECODIFICADOR DE VITERBI <sup>41 42</sup>

La decodificación de una secuencia que ha sido ingresada en un codificador convolucional se realiza mediante el algoritmo de Viterbi. Este algoritmo utiliza el diagrama de Trellis. El proceso de decodificación consiste en buscar un camino en el diagrama de Trellis que de como resultado la secuencia de bits más probable.

Con el algoritmo no se almacenan todas las secuencias que puede generar el codificador convolucional y se basa en el siguiente principio: el mejor camino, con menor distancia de Hamming, que pasa por un determinado nodo, necesariamente incluye el mejor camino desde el principio del diagrama de Trellis hasta este nodo. La distancia de Hamming se define como el número de bits diferentes entre dos palabras del mismo tamaño.

El principio anterior implica que para cada uno de los nodos del diagrama de Trellis sólo es necesario guardar el mejor camino (secuencia) hasta ese nodo. De esta forma, como mucho se tendrán tantos caminos como estados diferentes (el número de estados es  $2^{k \cdot (K-1)}$ ). El mejor camino se escoge en función de la menor métrica. Se define la métrica de bifurcación  $\mu_k$  como la métrica calculada para una ruta entre el estado  $S_k$  y el estado anterior  $S_{k-1}$ . El valor de  $\mu_k$  es calculado para todas las rutas entre los estados  $S_k$  y el estado  $S_{k-1}$ .

Otro parámetro a considerar es la métrica acumulada, que se define como la sumatoria de todas las métricas anteriores más la métrica del estado actual. Y se define como:

$$\Sigma_k = \sum_{t=1}^k \mu_t \quad (\text{Ec 1.14})$$

---

<sup>41</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, p.113-118.

<sup>42</sup> CDMA Systems Engineering Handbook, Lee, Jhong S.; Miller, Leonard E., p. 875-882

Cada estado  $S_k$  posee una métrica acumulada. Estas se van actualizando mediante el algoritmo de Viterbi cada vez que se recibe una nueva palabra de código (incremento de la variable de paso de tiempo  $k$ ).

Los pasos a seguir para la decodificación son:

- En el nivel  $S_k$ , calcular la distancia de Hamming de cada ruta entrante en cada nodo (estado) desde el nodo del nivel  $S_{k-1}$  hasta el nodo del nivel  $k$ .
- Para cada nodo (estado) del diagrama de Trellis en el nivel  $k$ , descartar todas las rutas que entran en el nodo, excepto el de distancia mínima. Cuando a un nodo llegan dos rutas con la misma distancia se toma el superior. Luego se asigna a cada estado la métrica acumulada. El camino elegido se lo denomina como ***ruta superviviente***.
- Pasar al nivel  $k + 1$  y repetir los 2 primeros pasos.

### 1.3.3.1 Ejemplo:

Se asume el codificador convolucional (2, 1, 3) analizado anteriormente. La secuencia a codificar es [0 1 0 1 1]. Después de la codificación se obtiene la secuencia [00 11 10 00 01].

En el receptor llega la secuencia con errores [00 11 11 00 01].

Con el diagrama de Trellis se calculan las métricas para encontrar el mejor camino:

#### **Cálculo para $k = 1$ , nivel 1:**

Cálculo de las métricas:

$$d(00,00) = 0, \quad d(00,11) = 2$$

En este caso la métrica más baja es cero, por lo que esta será la métrica de ruta escogida.

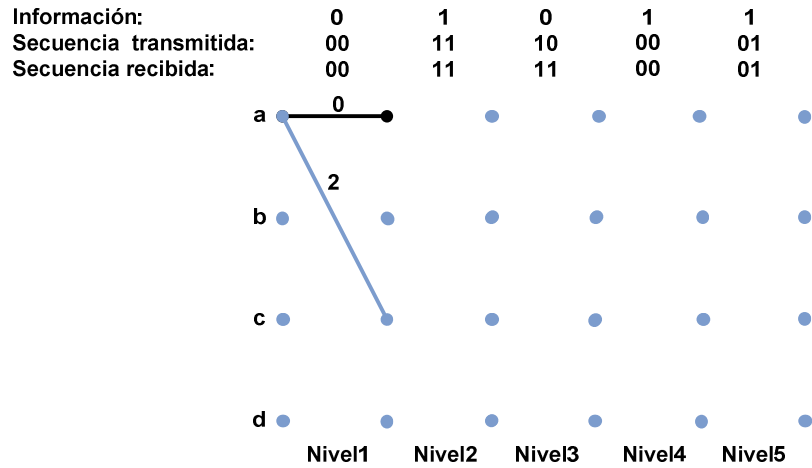


Figura 1.12 Cálculo de métrica para el nivel 1.

### Cálculo para $k = 2$ , nivel 2:

Calculo de la métrica acumulada, como se menciono anteriormente es la sumatoria de las métricas anteriores mas la métrica propia del estado actual.

$$0 + d(11, 00) = 0 + 2 = 2 \quad \mathbf{0 + d(11, 11) = 0 + 0 = 0}$$

En este caso la métrica más baja es cero, por lo que esta será la métrica de ruta escogida.

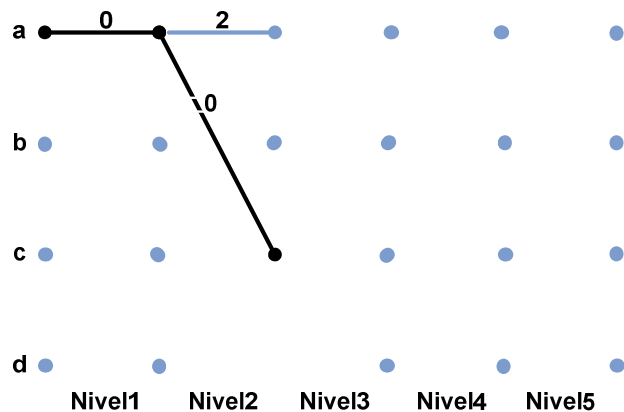


Figura 1.13 Cálculo de métrica para el nivel 2.

**Cálculo para  $k = 3$ , nivel 3:**

Cálculo de la métrica Acumulada:

$$0+d(11,01) = 0+1 = 1 \quad 0+d(11,10) = 0+1 = 1$$

En este caso se tiene dos rutas con igual métrica mínima, por lo que se escoge la ruta superior.

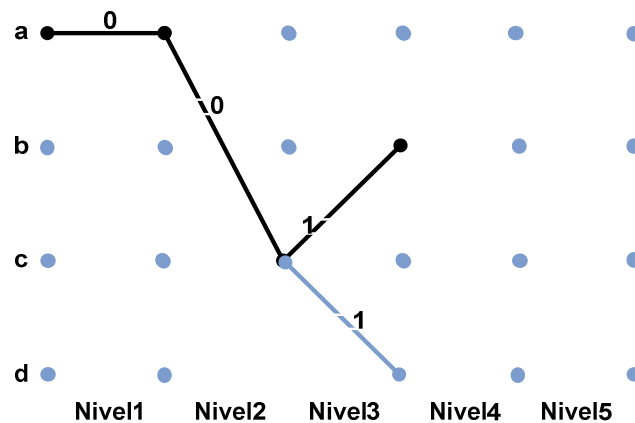


Figura 1.14 Cálculo de métrica para el nivel 3.

**Cálculo para  $k = 4$ , nivel 4:**

Cálculo de la métrica Acumulada:

$$1+d(00,11) = 1+2 = 3 \quad 1+d(00,00) = 1+0 = 1$$

En este caso la métrica más baja es uno, por lo que esta será la métrica de ruta escogida.



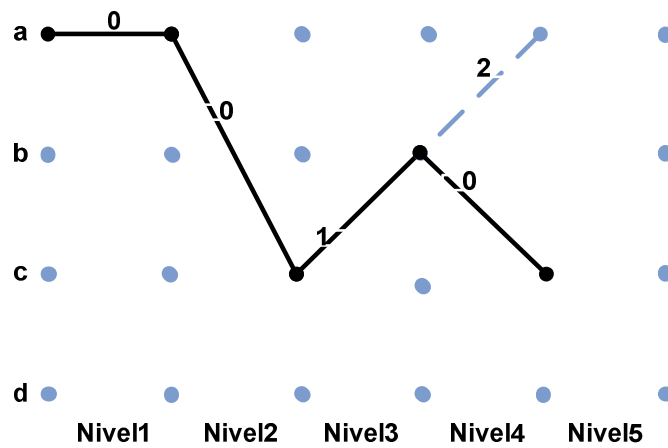


Figura 1.15 Cálculo de métrica para el nivel 4.

### Cálculo para $k = 5$ , nivel 5:

Cálculo de la métrica Acumulada:

$$1+d(01,01) = 1+0 = 1 \quad 1+d(01,10) = 1+2 = 3$$

En este caso la métrica más baja es uno, por lo que esta será la métrica de bifurcación escogida.

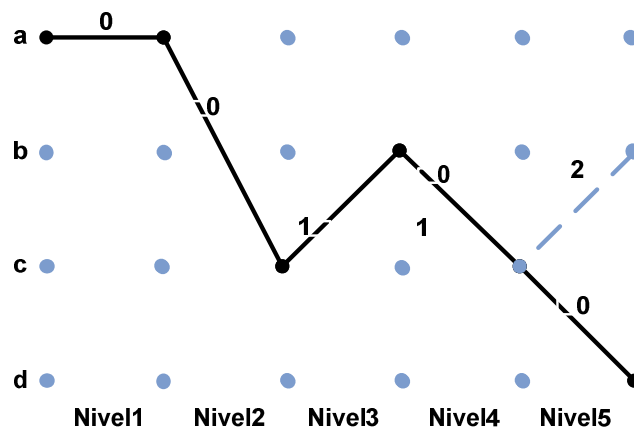


Figura 1.16 Cálculo de métrica para el nivel 5.

Una vez seleccionado el camino, se elige el bit que provoca la transición de estado entre dos niveles comenzando desde el primer nivel hasta el último. La secuencia de bits obtenida es la cadena decodificada:

Nivel	Bit
1	0
2	1
3	0
4	1
5	1

La secuencia decodificada obtenida es [0 1 0 1 1]. Este resultado coincide con los datos originales transmitidos y por lo tanto se han corregido los errores ingresados.

#### 1.3.4 INTERLEAVING.

Como se estudio anteriormente, los códigos de corrección de error son diseñados para combatir los efectos del desvanecimiento para mantener un nivel razonable de potencia. Pero, durante periodos de alto desvanecimiento de la señal se producen ráfagas de errores que hacen que la corrección de errores sea ineficaz.

El Interleaving es la técnica utilizada para dar aleatoriedad a los bits dentro de un mensaje, de tal manera que los errores producidos por el canal puedan ser considerados como aleatorios.

El esquema de interleaving usado en IS-95A se explicará en el capítulo 4.

### 1.4 RECEPTOR RAKE.<sup>43</sup>

Las señales transmitidas en un canal inalámbrico son altamente atenuadas y la información se puede perder en desvanecimientos profundos. La recepción se puede mejorar, con un receptor que contenga dos o más replicas de la misma señal de información. Estas replicas deberían ser transmitidas a través de canales

---

<sup>43</sup> <http://www.wireless.per.nl/reference/chaptr05/cdma/rake.htm>

de desvanecimiento independientes, haciendo que la probabilidad de todo el desvanecimiento sea muy pequeño al mismo tiempo.

El receptor Rake consiste de un número de ramas igual al número de componentes multitrayectoria de la señal, llamados dedos. Cada dedo intenta demodular un camino de la señal multitrayectoria.

El Receptor Rake aprovecha las características de las secuencias de ensanchamiento implementadas en los sistemas CDMA, como se observará en el capítulo 4. Para esto, es importante conocer el concepto de correlación de señales temporales.

## 1.5 CORRELACIÓN CRUZADA Y AUTOCORRELACIÓN DE SERIES TEMPORALES <sup>44 45 46</sup>

Las funciones de correlación proporcionan medidas de la semejanza de una señal  $f(t)$  con ella misma o con otra señal, comparadas con un desplazamiento relativo del tiempo. La correlación es ampliamente utilizada en aplicaciones como la detección de señales dañadas por el ruido del canal de comunicaciones, la estimación de retardos en radar, sincronización, el reconocimiento de patrones, detección y sincronización en comunicaciones digitales, la estimación espectral, etc.

La comparación de dos señales distintas se la conoce como **Correlación Cruzada** y la correlación de una señal con una copia de sí misma se la conoce como **Autocorrelación**.

La función de Correlación periódica discreta de dos señales periódicas se define como:

---

<sup>44</sup> Procesamiento Digital de Señales, PhD Alvarado Moya, José Pablo, p. 57-60.

<sup>45</sup> Señales y Sistemas de Tiempo Discreto, Bertran Albertí, Edward, p. 425-431.

<sup>46</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 40-43.

$$R_{xy}(n) = \sum_{k=-\infty}^{k=\infty} x(k)y(k+n) \quad (\text{Ec 1.15})$$

$$n = 0, \pm 1, \pm 2, \pm 3, \dots$$

Donde el parámetro  $n$  es el desplazamiento o retardo en el tiempo de la señal.

Mientras que la función de Autocorrelación periódica se define como:

$$R_{xx}(n) = \sum_{k=-\infty}^{k=\infty} x(k)x(k+n) \quad (\text{Ec 1.16})$$

$$n = 0, \pm 1, \pm 2, \pm 3, \dots$$

La función de Autocorrelación periódica toma el valor máximo cuando las dos señales se encuentran en fase (desplazamiento  $n=0$ ). Por lo tanto:

$$R_{xx}(0) = E_x \quad (\text{Ec 1.17})$$

Donde  $E_x$  es la energía de la señal discreta  $x(n)$ . La ecuación de arriba tiene una importante aplicación en la detección de datos. Esta sugiere que al tiempo de desplazamiento cero, el nivel de autocorrelación es proporcional a la energía de la señal la cual ayuda al proceso de detección.

Cuando las señales que se correlacionan son periódicas, la función de Correlación cruzada y autocorrelación son simétricas al origen. Para tener la función de correlación en Matlab se realizó la función ***xcorrelacion(sec1,sec2).m***.

La función devuelve dos vectores con los desplazamientos (***ind***) y la función de correlación calculada (***corr***). Para el funcionamiento de la función se considera que las secuencias ingresadas son de la misma longitud.

Para la comprobación del funcionamiento de la función ***xcorrelacion***, se realiza el siguiente ejemplo:

Para calcular la autocorrelación de la secuencia  $[-1 \ 1 \ 1 \ -1 \ -1 \ -1 \ 1]$ , se ingresan los comandos:

```
>> secuencia=[-1 1 1 -1 -1 -1 1];
>> [ind corr]=xcorrelacion(secuencia,secuencia)

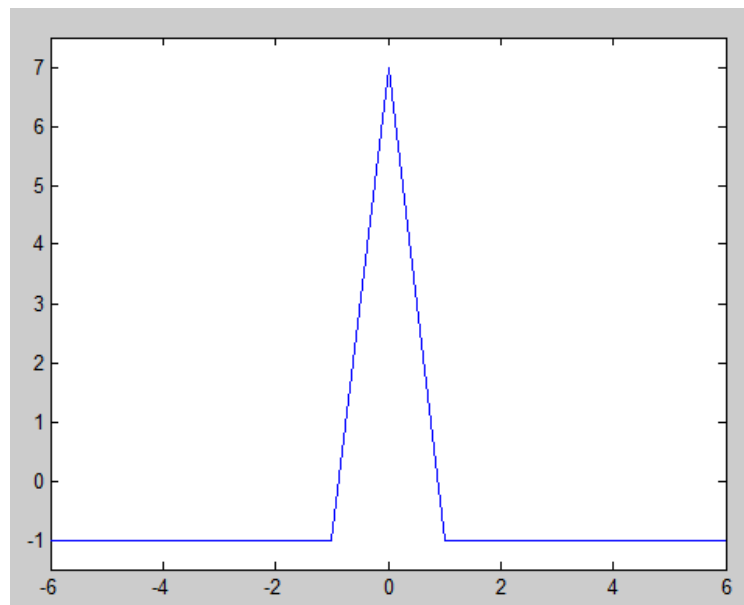
ind =
    -6    -5    -4    -3    -2    -1     0     1     2     3     4     5     6

corr =
    -1    -1    -1    -1    -1    -1     7    -1    -1    -1    -1    -1    -1
```

**Figura 1.17** Comprobación de la función xcorrelacion.

Como se puede ver en la figura, para realizar la Autocorrelación se compara la secuencia consigo misma. La función devuelve los valores de los desplazamientos de la secuencia, así como los valores de la correlación calculados para cada desplazamiento.

Graficando los valores de correlación con los respectivos desplazamientos se tiene:



**Figura 1.18** Autocorrelación de la secuencia  $[-1 \ 1 \ 1 \ -1 \ -1 \ -1 \ 1]$ .

Matlab también tiene una función que calcula la correlación, esta es la función ***xcorr***.

**xcorr**  
Cross-correlation

**Syntax**

```
c = xcorr(x,y)
c = xcorr(x)
c = xcorr(x,y,'option')
c = xcorr(x,'option')
c = xcorr(x,y,maxlags)
c = xcorr(x,maxlags)
c = xcorr(x,y,maxlags,'option')
c = xcorr(x,maxlags,'option')
[c,lags] = xcorr(...)
```

**Description**

**xcorr** estimates the cross-correlation sequence of a random process. Autocorrelation is handled as a special case.

The true cross-correlation sequence is

$$R_{xy}(m) = E\{x_{n+m}y_n^*\} = E\{x_n y_{n-m}^*\}$$

where  $x_n$  and  $y_n$  are jointly stationary random processes,  $-\infty < n < \infty$ , and  $E\{\cdot\}$  is the expected value operator. **xcorr** must estimate the sequence because, in practice, only a finite segment of one realization of the infinite-length random process is available.

**Figura 1.19** Captura de la función xcorr del Help de Matlab.

La función ***xcorr*** estima la correlación cruzada de una secuencia.

Calculando los valores para la autocorrelación para la secuencia anterior se tiene:

```
>> secuencia=[-1 1 -1 1 -1 1 -1];
>> [corr ind]=xcorr(secuencia,secuencia)

corr =

Columns 1 through 9
    1.0000   -2.0000    3.0000   -4.0000    5.0000   -6.0000    7.0000   -6.0000    5.0000

Columns 10 through 13
   -4.0000    3.0000   -2.0000    1.0000

ind =

   -6    -5    -4    -3    -2    -1     0     1     2     3     4     5     6
```

**Figura 1.20** Autocorrelación de la secuencia [-1 1 1 -1 -1 -1 -1 1] con xcorr.

Graficando la función, se tiene:



Como se puede ver en la tabla anterior, los resultados son los obtenidos en Matlab con la función **xcorr**.

```
>> secuencia=[-1 1 -1 1 -1 1 -1];
>> [corr ind]=xcorr(secuencia,secuencia)

corr =

Columns 1 through 9
    1.0000   -2.0000    3.0000   -4.0000    5.0000   -6.0000    7.0000   -8.0000    5.0000

Columns 10 through 13
   -4.0000    3.0000   -2.0000    1.0000

ind =

   -6    -5    -4    -3    -2    -1     0     1     2     3     4     5     6
```

**Figura 1.22** Autocorrelación de la secuencia [-1 1 1 -1 -1 -1 -1 1] con xcorr.

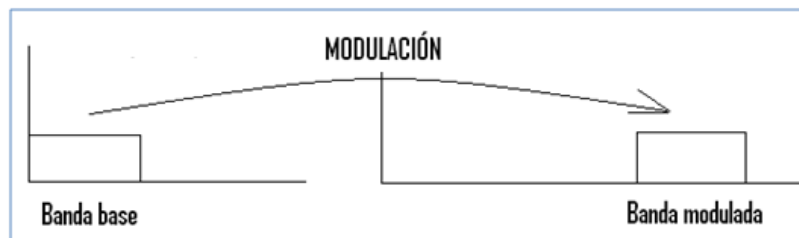
Para realizar el análisis de las secuencias que intervienen en CDMA, es necesario el uso de la función de correlación periódica. Con la función **xcorrelacion** implementada se aplica este concepto y por tanto se la utilizará en el Capítulo 2.



## CAPÍTULO II

### 2. ANÁLISIS DE SECUENCIAS DENTRO DE CDMA

Como se conoce, para poder transmitir un mensaje (**banda base original**) por **vía inalámbrica**, es necesario hacer uso de antenas que transformen la energía eléctrica en electromagnética. Si se intentara transmitir la información original, por ejemplo la voz, el cálculo de la longitud de onda es de muchos kilómetros, siendo inviable la construcción de dicha antena. Por ello es indispensable trasladar el espectro de potencia de la información hacia una zona alta del espectro de frecuencias de modo que el cálculo de las dimensiones de dicha antena, sean realizables. Esta es justamente la **operación de modulación**. En el receptor, la antena recibe la información y por medio del proceso inverso de **demodulación**, se recupera la **banda base** original.



**Figura 2.1** Operación de Modulación.

A estos procedimientos de modulación y demodulación se los resume de la siguiente manera:

#### **Modulación Analógica**

La **información analógica multiplicada por una onda analógica** de mucha mayor frecuencia (**portadora analógica**), genera la misma información pero movida hacia la frecuencia de la portadora y de manera duplicada:

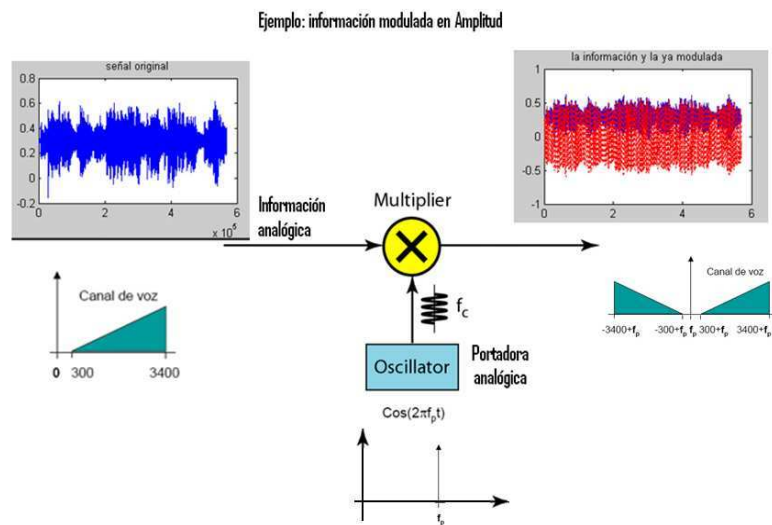


Figura 2.2 Modulación de Amplitud (AM)

### Modulación Digital

Todo es similar al caso anterior solo que la información a transmitir es de tipo digital. Entonces, la **modulación digital** es un caso particular de la **modulación analógica** donde la señal analógica solo toma ciertos valores.

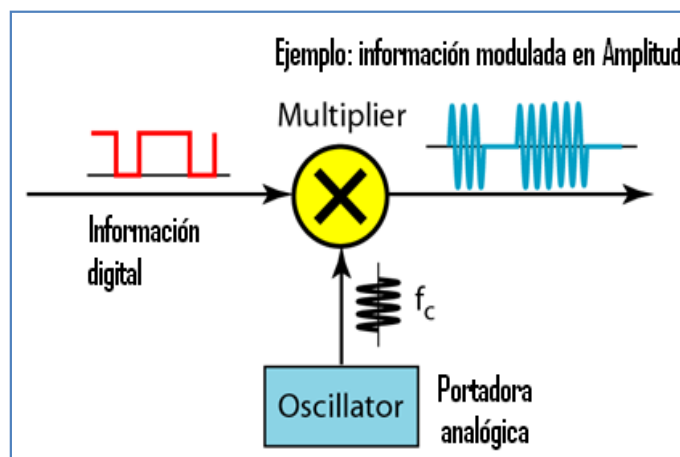
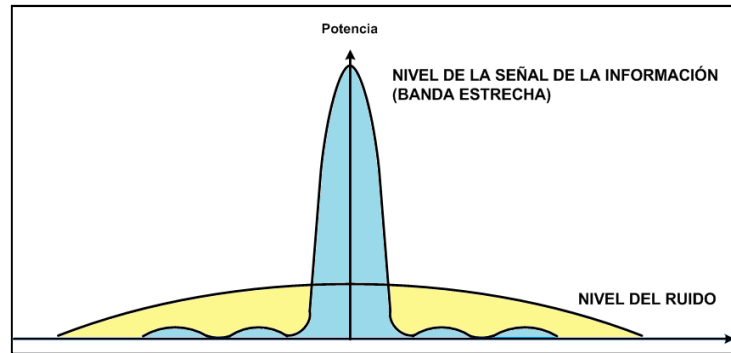


Figura 2.3 Modulación Digital.

En los dos casos anteriores, luego de filtrar una de las bandas, el espectro de la señal a transmitir se vería así:



**Figura 2.4** Espectro de Potencias de una Señal de Información.

En la figura se puede observar que **una señal normal de comunicaciones**, es **susceptible de ser detectada** (ver un pico ubicado en la portadora empleada), **interceptada** (si no tiene algún **código de seguridad** puede ser descubierto el mensaje) y en el caso de que se trate de una guerra electrónica, aún sin tener el código de seguridad, simplemente podría ser **atacada** mediante ruido de modo que evite la comunicación del otro bando.

Por lo anterior, surge la necesidad de idear un nuevo sistema que, de manera ideal, cumpla con los siguientes requerimientos:

- **Indetectabilidad:** que cuando se transmitan datos, tanto en el dominio del tiempo como de la frecuencia, no se note cambio alguno.
- **Seguridad:** que aunque el enemigo intercepte la comunicación, no pueda entenderla. Es decir, que provea resistencia ante interferencia tanto propia como extraña (jammers hostiles).
- **Que tenga acceso múltiple:** que permita que **múltiples usuarios** puedan utilizar el mismo medio de comunicación o canal (**como técnica de acceso al medio**).

A continuación, se dan las ideas fundamentales para cumplir con los anteriores requerimientos:

## ¿Cómo conseguir que la comunicación sea indetectable?

Como ya se planteó, la comunicación no sería detectada solo si antes y después de ella, no se nota cambio alguno en la información. Entonces, es necesario comprender las características del canal inalámbrico aislado:

### Comportamiento en el Dominio del Tiempo<sup>47</sup>

Si analizáramos el comportamiento del canal inalámbrico de manera aislada, como se mencionó en la página 3, se observaría que la señal temporal sigue una **Función de Distribución de Probabilidad (FDP)** de tipo **normal o Gaussiana**. Es decir, si quisiéramos sintetizarla, podríamos emplear la función siguiente:

$$y = f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (\text{Ec 2.1})$$

Donde  $\mu$  es el valor de la **tendencia central** (valor medio o media aritmética en este caso) y  $\sigma$  es el valor de la **dispersión** (desviación estándar en este caso).

En Matlab, la función que genera datos con una FDP normal o Gaussiana es **rand**. Lo que hace esta función es generar datos siguiendo la ecuación anterior.

### **Ejemplo:**

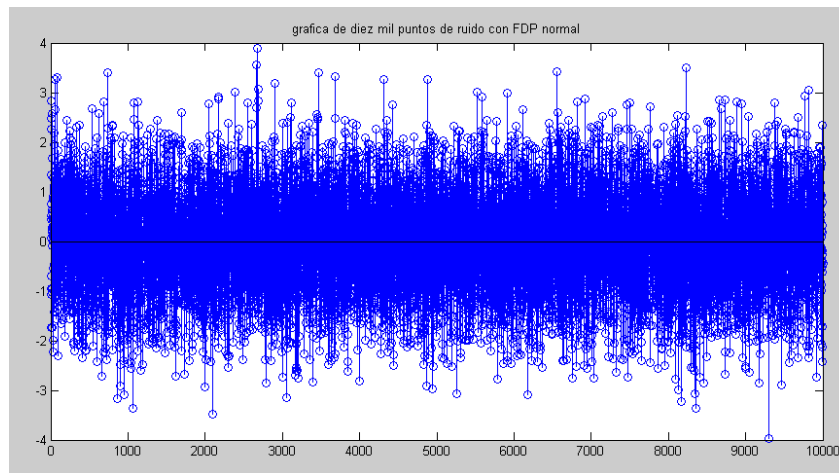
Generar una señal con 10000 puntos con FDP normal de media cero y desviación estándar de 1 y mostrar en una gráfica la serie temporal, verificar los valores de la media aritmética y de la desviación estándar. En otra gráfica muestre su histograma junto con el mejor ajuste Gaussiano. Finalmente, por medio de la **autocorrelación**, verifique que no contenga ninguna periodicidad.

### **Programa cap2\_cod4\_autocorrelacion\_ruido.m (ver Anexos Cap 2)**

---

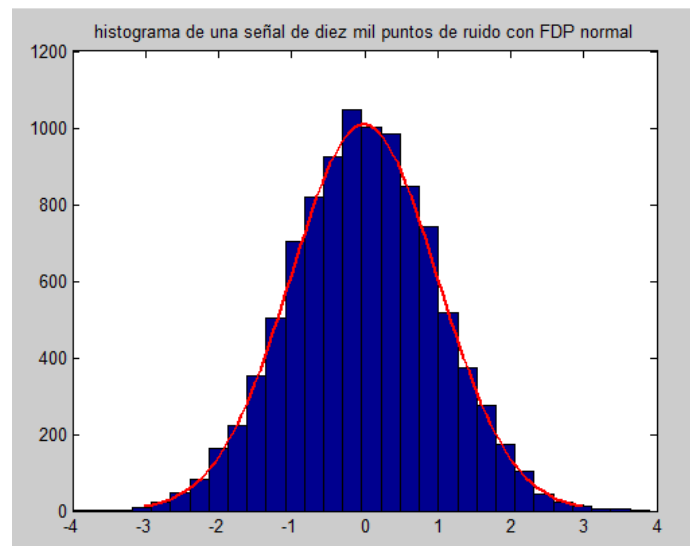
<sup>47</sup> <http://www.dte.us.es/personal/sivianes/tcomu/Alteraciones.pdf>

Se genera 10000 puntos aleatorios para representar el ruido, tal como se muestra a continuación:



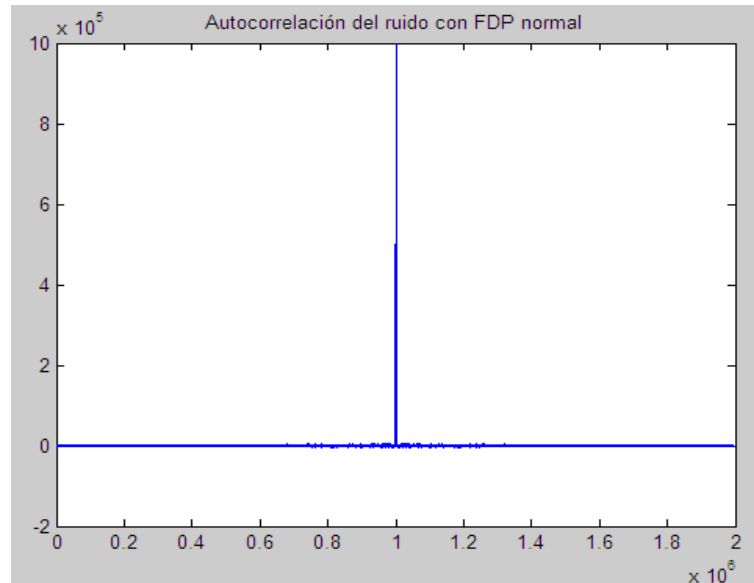
**Figura 2.5** Gráfica de diez mil puntos de ruido con FDP normal en Matlab.

Graficando el histograma, se verifica los valores de los parámetros de la FDP.



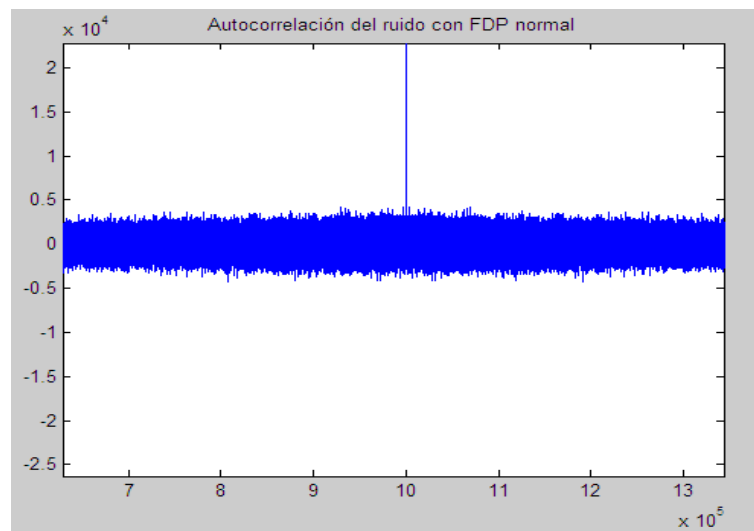
**Figura 2.6** Histograma de una señal de diez mil puntos de ruido con FDP normal.

Ahora calculando la correlación cruzada del ruido se tiene:



**Figura 2.7** Autocorrelación del Ruido con PDF normal.

Si se hace un zoom:



**Figura 2.8** Zoom de la Autocorrelación del ruido con FDP normal.

En la gráfica se puede observar que no existe ninguna periodicidad. Solo existe un pico cuando del desfase es de valor cero ( $k=0$ ). En el resto de puntos la señal no tiene ningún parecido a sí misma. Por lo tanto:

$$R_{xx}(0) = N$$

$$R_{xx}(k) \sim 0 \text{ para } 1 \leq k \leq N - 1$$

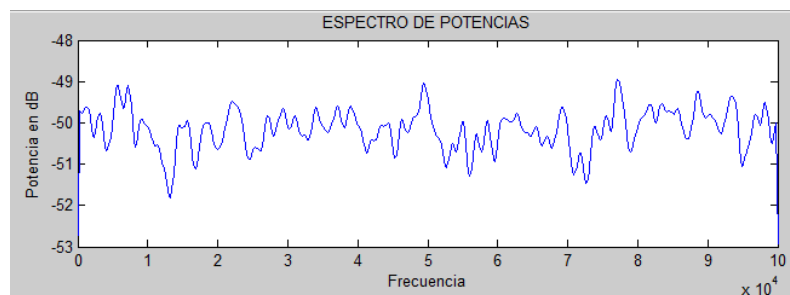
(Ec 2.2)

### Comportamiento en el Dominio de la Frecuencia<sup>48</sup>

Por otro lado, ya que es de carácter *aleatorio*, contiene, en teoría, un *rango infinito de componentes de frecuencia*, por lo que se denomina **blanco** (por parecerse al espectro de la luz blanca que ocupa un rango de frecuencias muy amplio correspondiente a la unión de los espectros de todos los colores). En otras palabras, no existen picos principales y más bien existe la misma potencia en cada una de las componentes de frecuencia: es un espectro plano.

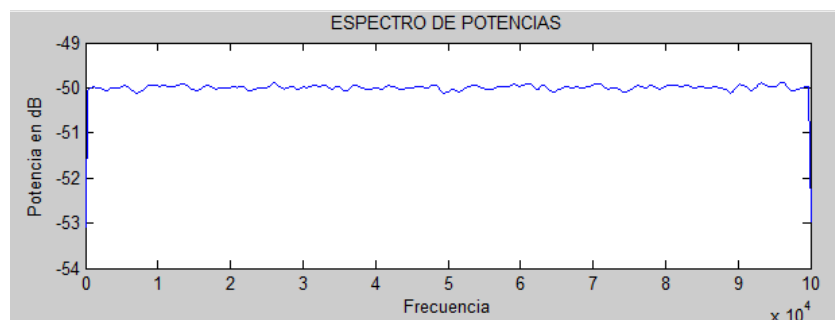
Por lo anterior, la denominación completa es **Ruido Aditivo Gaussiano Blanco (Additive White Gaussian Noise ó AWGN)**. Esto se puede verificar calculando el espectro de potencia del ruido, por ejemplo para a) 10000 puntos y b) para 1000000 de puntos de ruido generados en Matlab.

**Para n = 10000 puntos:**



**Figura 2.9** Espectro de Potencias del Ruido para 10000 puntos.

**Para n = 1000000 puntos:**



**Figura 2.10** Espectro de Potencias del ruido para 1000000 puntos.

<sup>48</sup> <http://www.dte.us.es/personal/sivianes/tcomu/Alteraciones.pdf>

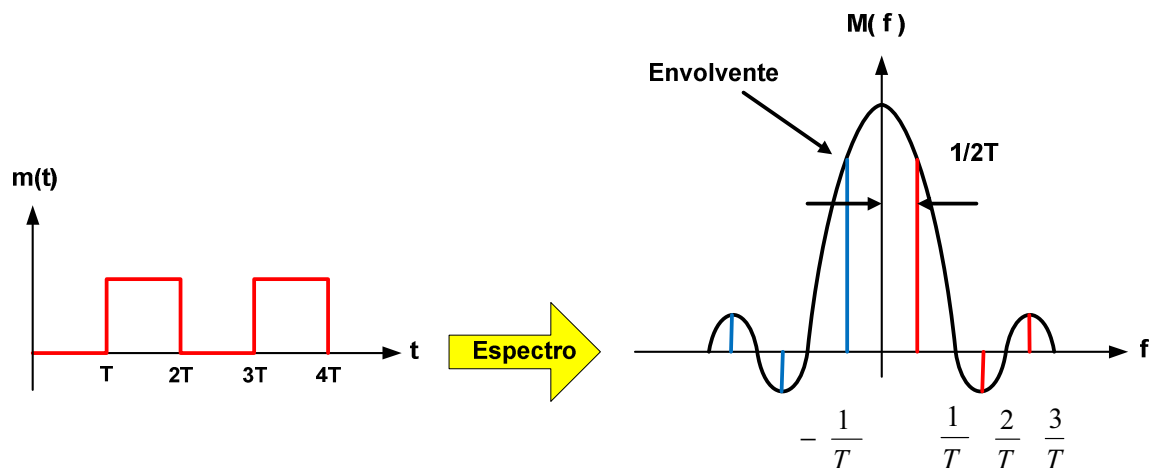
Como se ve, a medida que aumenta el número de puntos, el espectro tiende a ser plano desde menos infinito hasta más infinito. Esto es debido a que al aumentar el **tamaño de ventana**, la Transformada de Fourier aumenta su **resolución en frecuencia** tendiendo a mostrar el **espectro teórico plano**.

### Conclusión:

Después de analizar el comportamiento del ruido aislado, este, en el **dominio del tiempo** es aleatorio (**no contiene ninguna periodicidad**) y sigue una FDP normal o Gaussiana. En el **dominio de la frecuencia**, se verifica que el espectro de potencias tiende a ser plano, es decir, no aparece ningún pico predominante.

**¿Cómo conseguimos que el pico de la comunicación tradicional, disminuya de tamaño a tal punto de confundirse con el espectro del ruido?**

Para poder responder a esta interrogante, vamos a recordar el espectro de potencias de una **señal rectangular**:



**Figura 2.11** Espectro de una Señal Rectangular.

El espectro de una señal rectangular es una Sinc. El ancho de banda según Nyquist necesario para transmitir y recuperar la información en el receptor se define en  $1/T$ .



### ¿Qué pasa si se reduce el tiempo de bit?

Para contestar a esta pregunta, se necesita definir dos señales con distintos tiempo de bit, tal como se muestra en la figura.

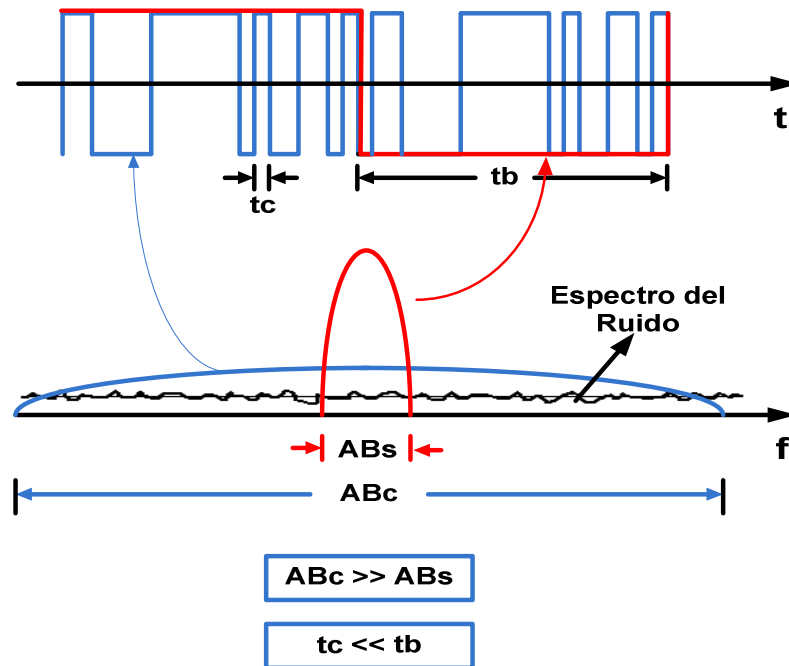


Figura 2.12 Ensanchamiento Espectral.

Se puede ver que, a medida que el período ( $T_s$ ) de la **Señal rectangular** se hace más corto, el ancho de banda ocupado por ella se incrementa  $R = 1/T = 2AB$  (**Teorema de Nyquist**). Verifiquémoslo mediante una simulación en matlab.

**Ejemplo.** Calcular el espectro del pulso rectangular  $f(t)$  siguiente:

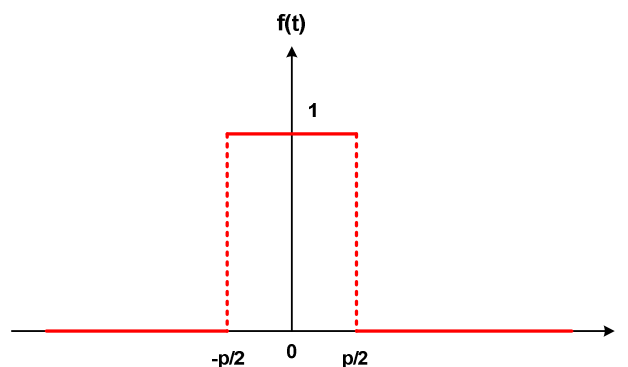


Figura 2.13 Pulso Rectangular.

La función en el dominio del tiempo es:

$$f(t) = \begin{cases} 0 & t < \frac{-p}{2} \\ 1 & \frac{-p}{2} < t < \frac{p}{2} \\ 0 & \frac{p}{2} < t \end{cases} \quad (\text{Ec 2.3})$$

Integrando:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt = \int_{-p/2}^{p/2} e^{-j\omega t} dt = \frac{1}{-j\omega} e^{-j\omega t} \Big|_{-p/2}^{p/2} \quad (\text{Ec 2.4})$$

Usando la formula de Euler:

$$\text{sen}(\omega p/2) = \frac{e^{j\omega p/2} - e^{-j\omega p/2}}{2j} \quad (\text{Ec 2.5})$$

$$F(\omega) = \frac{2\text{sen}(\omega p/2)}{\omega p} = \frac{2\text{sen}(\omega p/2)}{\omega p} * \frac{p/2}{p/2} = p \frac{\text{sen}(\omega p/2)}{\omega p/2} \quad (\text{Ec 2.6})$$

El espectro de magnitud es el módulo de  $F(\omega)$  en función de  $\omega$ .

### Resolución en Matlab:

Para observar el ensanchamiento espectral, se realizó el programa **cap2\_cod5\_espectro\_pulso.m** (ver Anexos Cap 2), donde se implementa el cálculo de la transformada de Fourier correspondiente a un pulso Rectangular de duración  $p$  y amplitud igual a uno. El desarrollo del programa se lo hace en base a variables simbólicas, que se utilizan para la resolución funciones matemáticas.

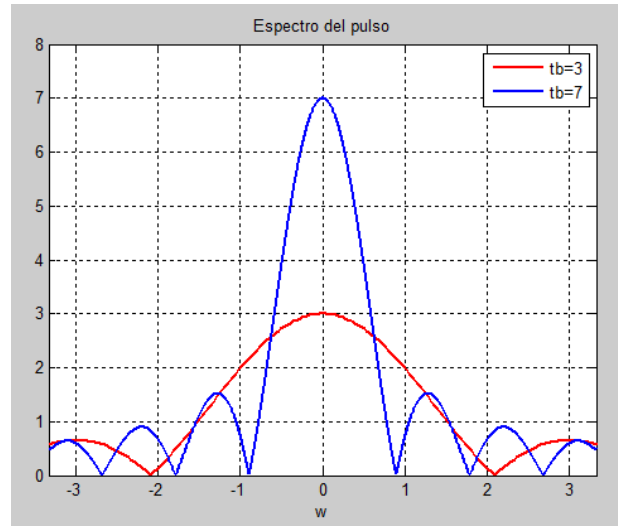
Para  $tb1=3$  y  $tb2=7$ , se tiene lo siguiente:

```

ingrese el menor tiempo de bit: 3
ingrese el mayor tiempo de bit: 7

```

**Figura 2.14** Tiempos de bit ingresados.



**Figura 2.15** Espectros para los pulsos con  $tb_1=3$  y  $tb_2=7$ .

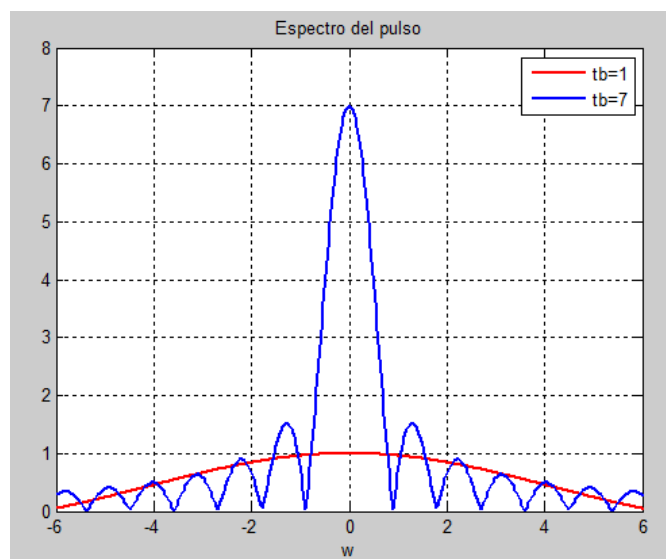
Para  $tb_1=1$  y  $tb_2=7$  se tiene:

```

ingrese el menor tiempo de bit: 1
ingrese el mayor tiempo de bit: 7

```

**Figura 2.16** Tiempos de bit ingresados.



**Figura 2.17** Espectros para los pulsos con  $tb_1=1$  y  $tb_2=7$ .

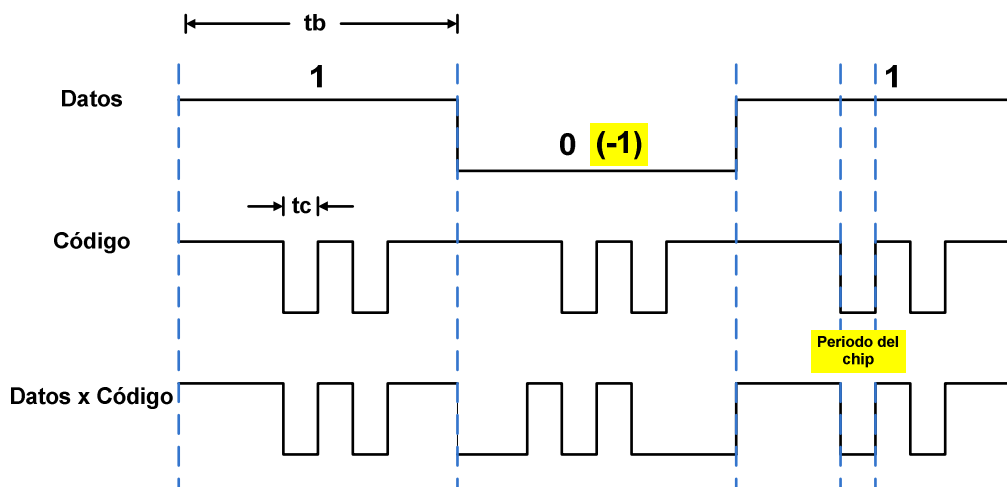
Como se puede notar, el espectro se ensancha y el pico tradicional se hace más corto a medida que el período ( $t_b$ ) de la **Señal rectangular** se hace más pequeño. Consiguiendo así, que a medida que se acorte más el tiempo de bit de la señal, su espectro sea comparable con el espectro ruido (espectro plano).

**Aplicación del fenómeno anterior de ensanchamiento espectral al objetivo de hacer indetectable a una señal**

Para dispersar el espectro, se debería multiplicar la **información** por una secuencia digital cuyo tiempo de un bit, denominado como **chip** ( $T_c$ ) sea mucho más pequeño que el tiempo de bit de la secuencia de información ( $T_b$ ).

$$\text{Secuencia de espectro ensanchado} = \text{Información } (T_b) * \text{Secuencia } (T_c)$$

$$(T_c \ll T_b)$$



**Figura 2.18** Formación de una Señal de Espectro ensanchado.

En este caso, se tendría un **caso especial de modulación** donde la información y la portadora (carrier:  $c$ ) **son digitales**.

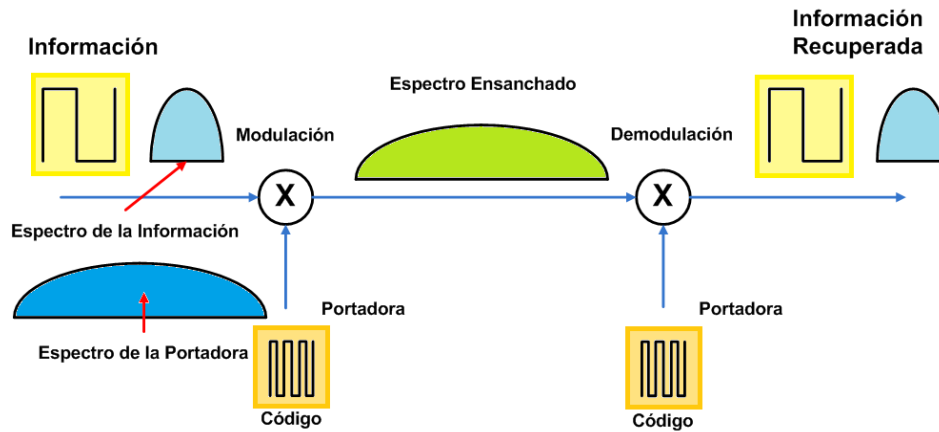


Figura 2.19 Sistema de Espectro Ensanchado.

A la idea anterior, solo falta añadir que la **secuencia o código, tenga características aleatorias** de modo que el espectro de potencias de la **señal resultante (datos x código) sea parecido al espectro del ruido y no muestre periodicidades (no contenga picos de Correlación Cruzada)**. De este modo habremos conseguido una comunicación indetectable.

### ¿Cómo se consigue que el código tenga características aleatorias?

Si la secuencia de código fuera totalmente aleatoria, simplemente el receptor no podría recuperar la señal original. Entonces, se debe investigar secuencias de tipo pseudoaleatorio, es decir, secuencias que se repitan periódicamente de tal forma que puedan ser generadas mediante algún método y permitan retirar su efecto a través del algoritmo opuesto.

### ¿Cómo conseguir que la comunicación sea Segura?

Para que un sistema tenga un cierto nivel de seguridad debe proveer resistencia ante los jammers hostiles y la interferencia propia. Como se conoce, en los sistemas de espectro ensanchado, solo aquel que sepa la secuencia clave, podrá recuperar la información. Para que sea segura la secuencia debe tener un período largo (periodos superiores a 1035 chips). La clave deberá ser igual o más larga que el mensaje. Cuando a una señal se la expande sobre varios megahercios del espectro, su potencia espectral también se ensancha. Esto hace que la potencia

transmitida también se ensanche sobre un extenso ancho de banda y dificulta la detección de forma normal. Así, adivinar la secuencia resulta extremadamente difícil mediante algún método, y se consigue cierto nivel de inmunidad ante los jammers hostiles.

Otra forma de dar seguridad a las comunicaciones mediante el uso de secuencias o códigos, es aumentar la complejidad de la misma. Es decir, realizar operaciones entre secuencias para generar un conjunto de secuencias difíciles de descifrar.

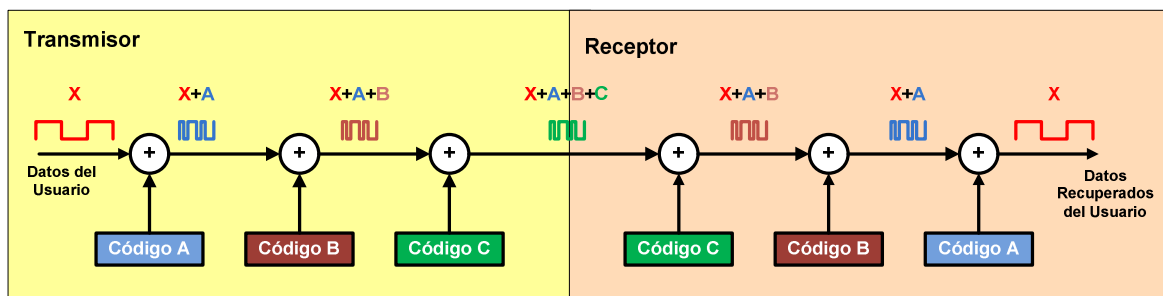


Figura 2.20 Incremento de seguridad a través del uso de mayor número de códigos.

### ¿Cómo conseguir Acceso Multiusuario?

El Acceso Multiusuario se refiere a dar la posibilidad que múltiples usuarios puedan utilizar el mismo medio de comunicación o canal (como técnica de acceso al medio). En la práctica se lo consigue asignando a cada usuario un código diferente. Por lo tanto, en recepción se podrá recuperar las informaciones correspondientes al aplicar el código propio de cada usuario.

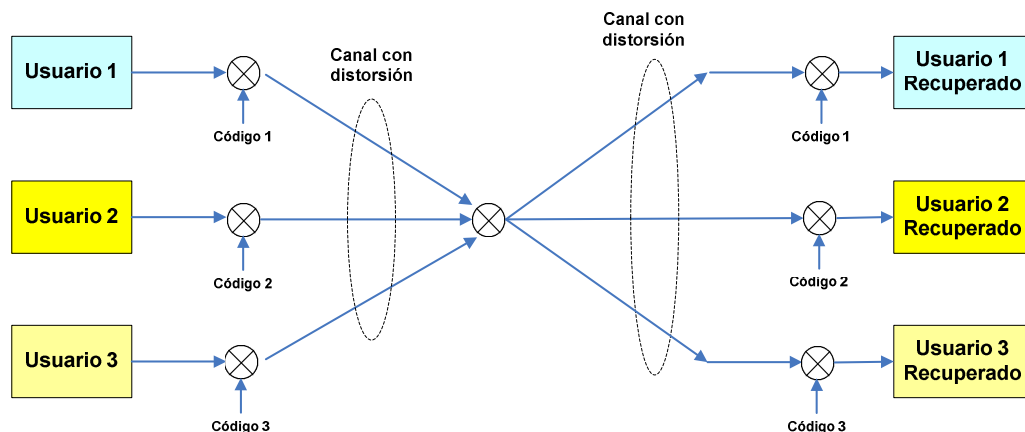


Figura 2.21 Acceso Multiusuario

Cuando varios usuarios tienen acceso al sistema de servicios, cada secuencia de código asignado a un usuario debe ser distinguible de cada secuencia asignada a otros usuarios e idealmente generar poca o ninguna interferencia entre los usuarios que comparten el canal.

La técnica de generación de las secuencias debe tener como objetivo crear una gran familia de secuencias con la finalidad de dar cabida a una gran cantidad de usuarios del sistema. Además las secuencias generadas deben tener un gran impulso de autocorrelación que le permita al sistema mejorar la sincronización y una baja correlación cruzada; pues la probabilidad de error en el desempeño del sistema de comunicación depende de las propiedades de correlación de la secuencia utilizada ya que ésta define la cantidad de interferencia generada por el acceso múltiple de usuarios.

Cada secuencia de código utilizado debe ser distinguible con facilidad de una versión desplazada en el tiempo de sí misma, a fin de permitir la adquisición del código y su seguimiento y, por tanto, la sincronización del sistema.

Aclaradas las inquietudes e ideas planteadas anteriormente, se ve la necesidad de comprender la teoría de secuencias.

Las secuencias que se analizarán se dividen en tres grupos:

- Secuencias No Ortogonales.
- Secuencias Ortogonales.
- Secuencias de Sincronización.

## 2.1 SECUENCIAS NO ORTOGONALES

Como ya se dijo, en la práctica no se podrían utilizar secuencias totalmente aleatorias debido a que no se podrían generar las mismas secuencias tanto en el transmisor como en el receptor. Por tanto, el mensaje original no podría ser recuperado. Entonces, dichas secuencias deberían ser **aparentemente aleatorias**, es decir, **pseudoaleatorias (secuencias PN)**. Para que esto se dé, el período de tales secuencias debería ser muy grande de modo que el enemigo no

podría detectar periodicidades a través de la operación de autocorrelación. Entonces, es necesario comprender la teoría de aquellas secuencias de modo que se empleen las que tengan el **máximo período posible**.

### 2.1.1 SECUENCIAS PSEUDOALEATORIAS O SECUENCIAS PN

Las **secuencias PN** se definen como una secuencia binaria, de cierto **período N**, tan grande que se pueda aproximar a una señal aleatoria.

Dentro de las secuencias PN se define a cada elemento de la misma como un **chip**, cuyo tiempo de duración es mucho menor a la duración de un bit de la secuencia que se quiera codificar o ensanchar. Una secuencia se dice que cumple con los preceptos de ser pseudoaleatoria solo si cumple con los **postulados de Golomb**.

#### 2.1.1.1 Postulados de Golomb<sup>49</sup>

Golomb postuló tres propiedades que se asocian con la **aleatoriedad de secuencias finitas**. Estos tres postulados, conocidos como postulados de Golomb, nos permiten realizar de una manera rápida y práctica, una comprobación de hasta qué punto la secuencia que se quiere estudiar cumple con los preceptos de aleatoriedad.

Las secuencias que cumplen con estos postulados son **conocidas como Secuencias Pseudoaleatorias (Secuencias PN)**.

#### **Primer postulado de Golomb (G1)**

Deberá existir igual número de ceros que de unos. Se acepta como máximo una diferencia igual a la unidad. Esto permitirá que la secuencia en estudio, al igual que el ruido Gaussiano blanco, tenga aproximadamente una **media cero**.

---

<sup>49</sup> [http://www.criptosec.unizar.es/doc/tema\\_c3\\_criptosec.pdf](http://www.criptosec.unizar.es/doc/tema_c3_criptosec.pdf)



**Ejemplo 1:**

Se tiene la secuencia S1 tal como se muestra:

1 1 1 1 0 1 0 1 1 0 0 1 0 0 0

La secuencia S1 consta de 15 bits, de los cuales 8 son 1s y 7 son 0s. Como se puede apreciar la cantidad de unos y ceros difiere en una unidad, por lo tanto S1 cumple con el primer Postulado de Golomb.

**Ejemplo 2:**

Se tiene la secuencia S2 tal como se muestra:

0 1 0 1 1 1 0 0 1 0 0 1 0 0 0 1

La secuencia S2 consta de 16 bits, de los cuales 7 son 1s y 9 son 0s. Como se puede apreciar la cantidad de unos difiere en más de una unidad, por lo tanto la secuencia S2 no cumple con el primer Postulado de Golomb.

**Segundo Postulado de Golomb (G2)**

**Balance en Cadenas de bits:** Dentro de un período de la secuencia se definen cadenas de 1's y cadenas de 0's.

- Para que se dé un balance de cadenas de bits, es obvio que deberá haber **mayor número de rachas cortas que de rachas largas**. Una racha es una secuencia de uno o más dígitos iguales, ubicada entre dos dígitos distintos iguales pero opuestos a los de dicha secuencia.
- **La distribución de las rachas seguirá una progresión geométrica.** Por ejemplo una secuencia  $S_i$  podría tener 8 rachas de longitud uno, 4 de longitud dos, 2 de longitud tres y 1 de longitud cuatro.

En general, en un período T, la mitad del total de rachas de  $S_i$  serán de longitud 1, la cuarta parte de longitud 2, la octava parte de longitud 3, etc.

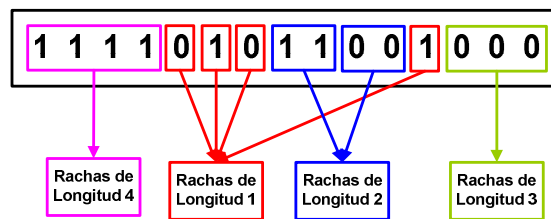
**Ejemplo:**

Se tiene la siguiente secuencia:

Si: 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0

Verificar si cumple con el segundo postulado de Golomb:

En la figura se muestra que en la secuencia de 15 bits podemos distinguir 8 rachas, de las cuales 4 son rachas de longitud 1, 2 rachas de longitud 2, 1 racha de longitud 3 y 1 racha de longitud 4.



**Figura 2.22** Rachas dentro de una secuencia.

Del postulado de Golomb, se debe cumplir:

- Si existen 8 rachas en la secuencia.

Para que se cumpla con el postulado deben existir:

- $8/2 = 4$  rachas de longitud 1.
- $8/4 = 2$  rachas de longitud 2.
- $8/8 = 1$  racha de longitud 3.
- $8/16 = 0,5$  aproximando es igual a 1 racha de longitud 4.

Por lo tanto, la secuencia mostrada cumple con el segundo postulado de Golomb.

### **Tercer postulado de Golomb G3**

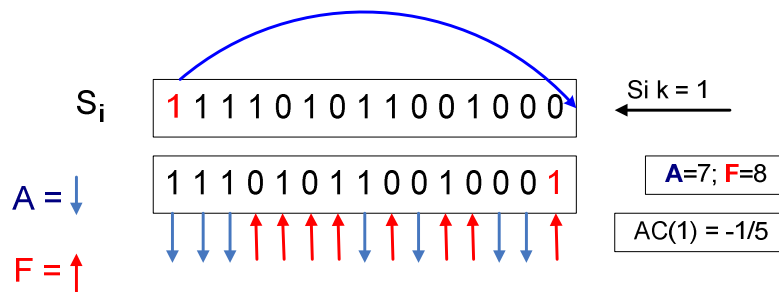
La **autocorrelación fuera de fase  $AC(k)$**  deberá ser **constante** para todo valor de desplazamiento de  $k$  bits.

**Ejemplo:**

Comprobar que para la secuencia  $S_i$  dada a continuación, la **autocorrelación**  $AC(k)$  para todos los valores de  $k$  ( $1 \leq k \leq 14$ ) es constante e igual a  $-1/15$ .

$$S_i = 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0$$

**Solución:** Para realizar la autocorrelación se compara la secuencia original con la secuencia desplazada un bit hacia la izquierda cada vez, tal como se muestra en la figura:



**Figura 2.23** Cálculo de la Correlación fuera de fase de una secuencia.

Para cada desplazamiento  $k$  se verifican los aciertos **A** que están marcados con azul y los fallos **F** en rojo. Luego se utiliza la fórmula para calcular la **autocorrelación fuera de fase**  $AC(k)$ .

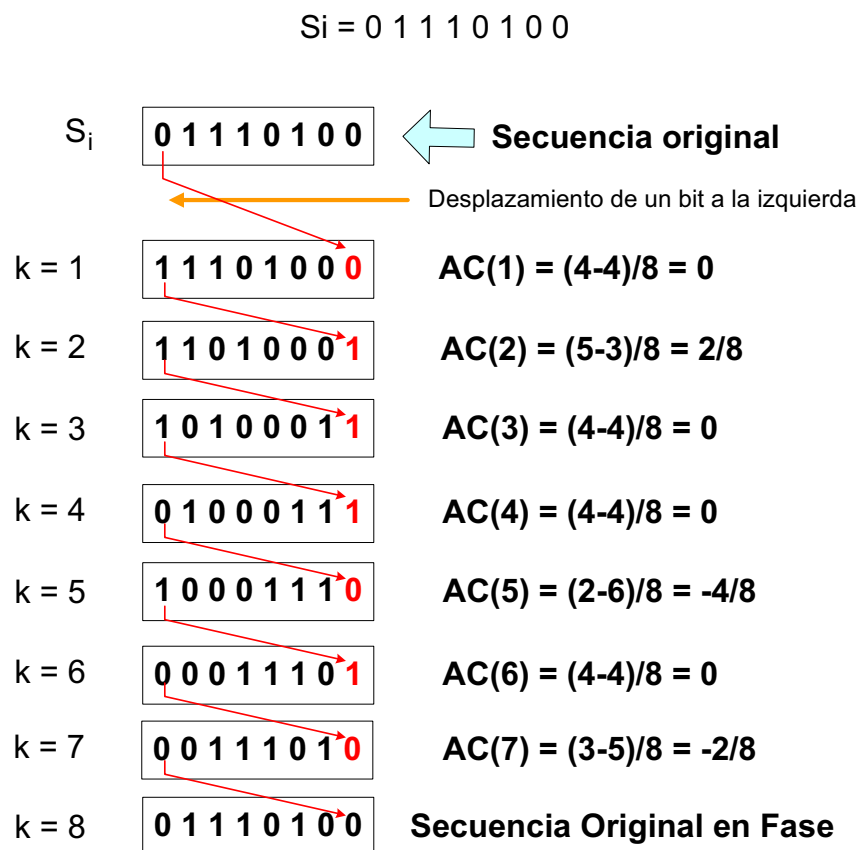
**Tabla 2.1** Autocorrelación fuera de fase  $AC(k)$ .

1	1	1	1	0	1	0	1	1	0	0	1	0	0	0	$AC(k)=(A-F)/T$
1	1	1	0	1	0	1	1	0	0	1	0	0	0	1	$AC(1) = 7-8/15 = -1/15$
1	1	0	1	0	1	1	0	0	1	0	0	0	1	1	$AC(2) = 7-8/15 = -1/15$
1	0	1	0	1	1	0	0	1	0	0	0	1	1	1	$AC(3) = 7-8/15 = -1/15$
0	1	0	1	1	0	0	1	0	0	0	1	1	1	1	$AC(4) = 7-8/15 = -1/15$
1	0	1	1	0	0	1	0	0	0	1	1	1	1	0	$AC(5) = 7-8/15 = -1/15$
0	1	1	0	0	1	0	0	0	1	1	1	1	0	1	$AC(6) = 7-8/15 = -1/15$
1	1	0	0	1	0	0	0	1	1	1	1	1	0	1	$AC(7) = 7-8/15 = -1/15$
1	0	0	1	0	0	0	1	1	1	1	0	1	0	1	$AC(8) = 7-8/15 = -1/15$
0	0	1	0	0	0	1	1	1	1	0	1	0	1	1	$AC(9) = 7-8/15 = -1/15$
0	1	0	0	0	1	1	1	1	0	1	0	1	1	0	$AC(10) = 7-8/15 = -1/15$
1	0	0	0	1	1	1	1	0	1	0	1	1	0	0	$AC(11) = 7-8/15 = -1/15$
0	0	0	1	1	1	1	0	1	0	1	1	0	0	1	$AC(12) = 7-8/15 = -1/15$
0	0	1	1	1	1	0	1	0	1	1	0	0	1	0	$AC(13) = 7-8/15 = -1/15$
0	1	1	1	1	0	1	0	1	1	0	0	1	0	0	$AC(14) = 7-8/15 = -1/15$

Para la secuencia presentada anteriormente, la **autocorrelación** es constante e igual a  $-1/15$ .

### Ejemplo 2:

Encontrar la **autocorrelación**  $AC(k)$  para la secuencias  $S_i$  y comprobar si cumple con el tercer postulado de Golomb.



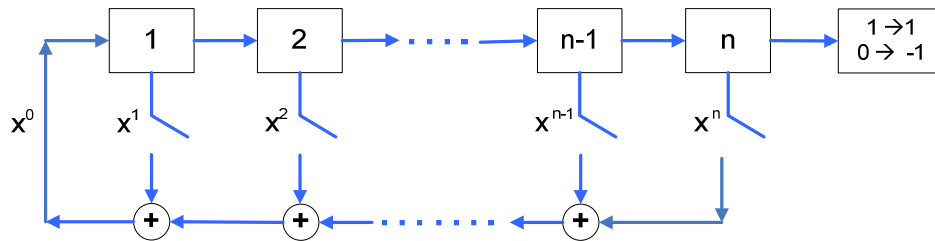
**Figura 2.24** Autocorrelación de la secuencia 0 1 1 1 0 1 0 0.

La secuencia  $S_i = 01110100$  de 8 bits no cumple con G3. Se puede notar que la autocorrelación  $AC(k)$  toma varios valores.

#### 2.1.1.2 Generadores LFSRs de Secuencias PN

Como ya se ha venido comentando, es necesario generar una secuencia pseudoaleatoria, es decir que tenga un período muy grande y que cumpla con los postulados de Golomb. Por la sencillez de implementación, las secuencias PN

pueden ser generadas mediante una **estructura de Registros de Desplazamiento Realimentados Linealmente** (en inglés: **LFSR Linear Feedback Shift Register**) de  $n$  flip flops y una función o tomas especialmente seleccionadas.



**Figura 2.25** Estructura de un Generador LFSR con  $n$  flip flops.

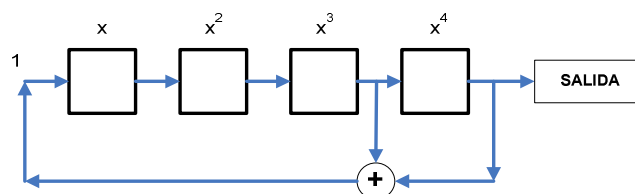
Donde el sumador representa la adición modulo 2 (XOR).

Las realimentaciones o tomas se corresponden con los **coeficientes binarios no nulos de un polinomio  $P(X)$ , de grado  $n$** , perteneciente al campo de Galois binario  $GF(2)$ , de la forma:

$$P(X) = a_n X^n + a_{n-1} X^{n-1} + \dots + a_1 X^1 + a_0; \quad a_i \in (0,1) \quad (\text{Ec 2.7})$$

Donde cada coeficiente  $a_i$  no nulo representa una toma del registro efectuada a la salida de cada etapa. Los coeficientes  $a_0$  y  $a_n$  son siempre iguales a 1. Las tomas se llevan a sumadores modulo 2 concatenados. La salida del último sumador se lleva a la entrada del registro (coeficiente  $a_n = 1$ ). El orden del polinomio es igual al número de elementos del registro.

La siguiente figura muestra un generador LFSR asociado al polinomio  $P(X) = X^4 + X^3 + 1$ .

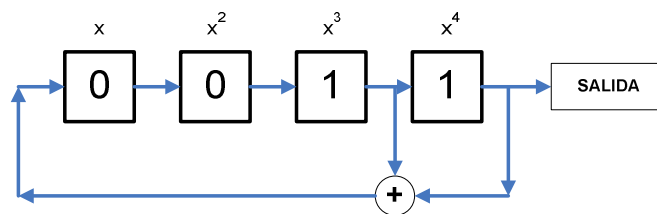


**Figura 2.26** Generador LFSR de cuatro celdas.

## ¿Cómo se genera la Secuencia PN?

A continuación se explica el funcionamiento de este generador mediante un ejemplo.

La siguiente figura muestra un registro de desplazamiento de longitud  $N=4$  que se activa con un reloj común. A cada pulso de reloj el estado de un flip flop se desplaza al siguiente flip flop. El conjunto de registros de desplazamiento consta de una realimentación que ingresa al primer flip flop. La función de la realimentación se obtiene usando la suma en módulo 2 a partir de la salida de varios de los flip flops.



**Figura 2.27** Generador LFSR de cuatro celdas con el estado inicial de 0011.

La secuencia generada está dada por el contenido del flip flop que se ubica más a la derecha o por el que representa el flip flop de mayor orden. Para este ejemplo **el estado inicial** (conocido también como **semilla**) del registro de desplazamiento, está definido tal como se muestra en la figura:

Realizando las operaciones indicadas por las diferentes uniones, teniendo como estados iniciales de los flip flops los ya indicados, a la salida del registro se tiene:

**Tabla 2.2** Semilla Inicial.

Registro N°	1	2	3	4	Salida
Valor inicial	0	0	1	1	1

Realizando  $k$  desplazamientos se obtiene:

**Tabla 2.3** Semillas y Salidas para los k desplazamientos.

k Desplazamientos	Registros				Salida
	1	2	3	4	
1	0	0	0	1	1
2	1	0	0	0	0
3	0	1	0	0	0
4	0	0	1	0	0
5	1	0	0	1	1
6	1	1	0	0	0
7	0	1	1	0	0
8	1	0	1	1	1
9	0	1	0	1	1
10	1	0	1	0	0
11	1	1	1	1	1
12	1	1	1	0	0
13	1	1	1	1	1
14	0	1	1	1	1
15	0	0	1	1	1
Se repite la secuencia					

Se puede notar que al desplazamiento  $k=15$ , **el registro de desplazamiento contiene la semilla inicial** y por lo tanto, nuevamente se generan para otros desplazamientos las mismas salidas y estados del registro de desplazamiento. Debido a esta razón las estructuras LFSR generan secuencias que tarde o temprano se repiten, es decir, **son periódicas**. Es por esta característica que la secuencia resultante se la conoce como **Pseudoaleatoria**.

Para el ejemplo expuesto anteriormente, la secuencia que se obtiene es la siguiente: 110001001101011. Una secuencia cuyo periodo es  $T=15$ .

Del proceso descrito, para el análisis de este tipo de generador, cabe hacer algunas preguntas:

- Con diferentes semillas, ¿qué secuencias se generan y cuál será su período?
- Las secuencias generadas, ¿tendrán propiedades?
- ¿Cumplen con los preceptos de aleatoriedad de Golomb?

### 2.1.1.3 Tipos de LFSR en función de los polinomios generadores asociados<sup>50</sup>

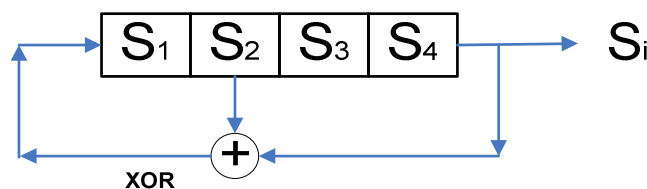
Como ya se comentó, las realimentaciones o tomas se corresponden con los **coeficientes binarios no nulos de un polinomio  $P(X)$ , de grado  $n$** . De este polinomio generador asociado dependerá las respuestas a las preguntas antes planteadas. Por lo tanto, se tienen los siguientes tipos de LFSRs:

- LFSR con polinomios  $f(x)$  factorizables.
- LFSR con polinomios  $f(x)$  irreducibles.
- LFSR con polinomios  $f(x)$  primitivos.

#### Generador LFSR con $f(x)$ factorizable

Un polinomio factorizable es aquel que puede ser factorado en dos o varios términos. Para ilustrarlo, se tiene el siguiente ejemplo:

Se tiene un generador  $f(x)$  factorizable de cuatro celdas ( $n = 4$ ), caracterizado por el polinomio  $f(x) = x^4 + x^2 + 1$ , tal como se muestra en la siguiente figura:



**Figura 2.28** Generador LFSR con  $f(x)$  factorizable.

$f(x)$  es factorizable porque:

$$\text{Sea } f(x_1) = f(x_2) = (x^2+x+1)$$

$$f(x) = f(x_1) \cdot f(x_2) = (x^2+x+1) \cdot (x^2+x+1) = x^4 + x^3+x^2+x^3+x^2+x+x^2+x+1$$

Tras la reducción módulo 2:

---

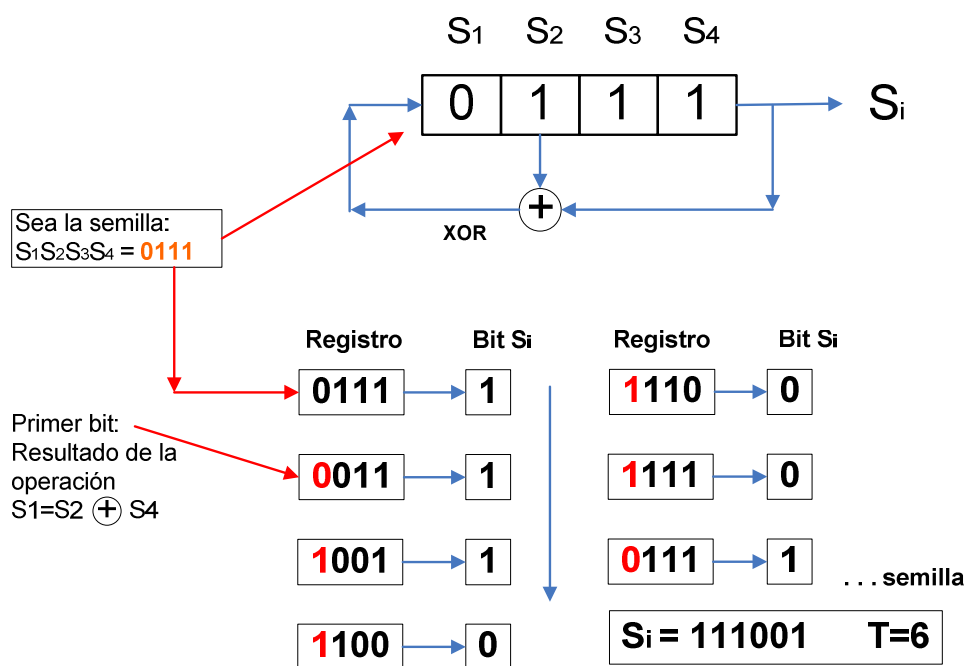
<sup>50</sup> <http://es.scribd.com/doc/4679956/Seguridad-Informatica-y-Criptografia-Sistemas-de-Cifra-en-Flujo>



$$f(x) = x^4 + x^2 + 1$$

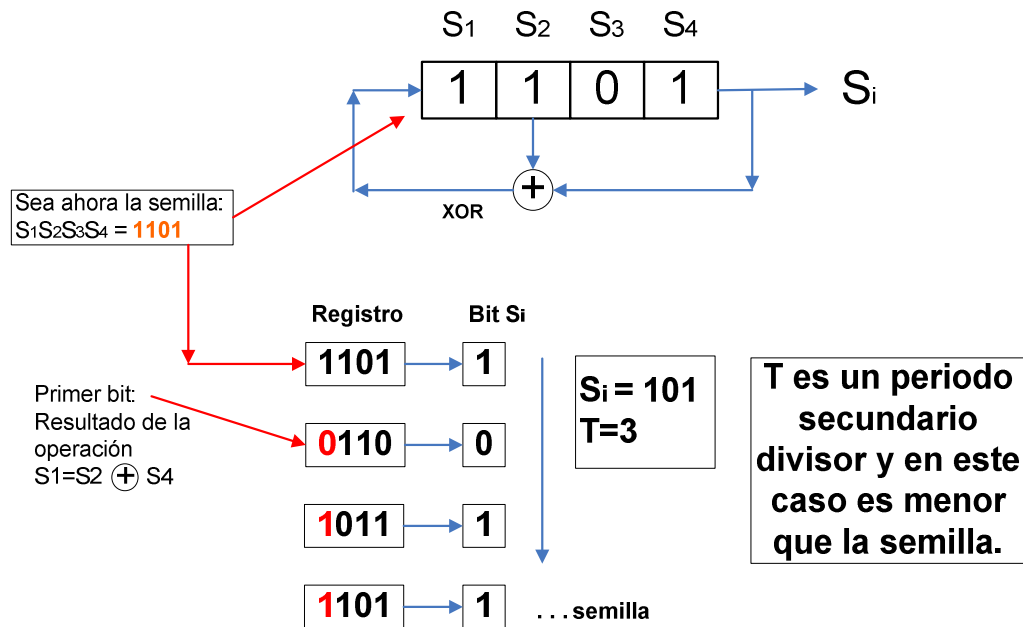
¿Qué periodo tendrá la secuencia generada a partir del LFSR con polinomio factorizable?

Para responder esta interrogante, se realiza un ejemplo. En la figura se tiene un LFSR con el polinomio  $x^4 + x^2 + 1$  y una semilla de 0111.



**Figura 2.29** Generación de una secuencia binaria con un LFSR con  $f(x)$  factorizable asociado y una semilla de 0111.

Como se puede observar la secuencia generada tiene un periodo  $T=6$ . Si ahora se cambia la semilla a 1101, como se muestra en la siguiente figura:



**Figura 2.30** Generación una secuencia binaria con un LFSR con  $f(x)$  factorizable y una semilla de 1101.

Se puede observar que el período es periodo  $T=3$ , que es menor al obtenido anteriormente. Si se analiza para todas las posibles semillas, se obtienen los siguientes períodos:

**Tabla 2.4** Secuencias generadas a partir de un LFSR con polinomio factorizable.

SEMILLA	SECUENCIA	PERIODO T
1 0 0 0	0 0 0 1 0 1	6
0 1 0 0	0 0 1 0 1 0	6
1 0 1 0	0 1 0 1 0 0	6
0 1 0 1	1 0 1 0 0 0	6
0 0 1 0	0 1 0 0 0 1	6
0 0 0 1	1 0 0 0 1 0	6
0 1 1 0	0 1 1	3
1 0 1 1	1 1 0	3
1 1 0 1	1 0 1	3
0 0 1 1	1 1 0 0 1 1	6
1 0 0 1	1 0 0 1 1 1	6
1 1 0 0	0 0 1 1 1 1	6
1 1 1 0	0 1 1 1 1 0	6
1 1 1 1	1 1 1 1 0 0	6
0 1 1 1	1 1 1 0 0 1	6

Se puede ver que para un **polinomio factorizable**, la secuencia resultante y el período de la misma **dependen de la semilla**. De acuerdo a la semilla se generan

tres secuencias diferentes con sus versiones desplazadas. Además existe un período primario ( $T=6$ ) y un período secundario ( $T=3$ ).

### ¿Cumple con los Postulados de Golomb?

Para responder a esta interrogante, se realiza el análisis de las siguientes secuencias:

#### S1: 000101

- Secuencia de Período  $T=6$
- Hay 4 (0) y 2 (1).
- 3 rachas de longitud 1.
- 1 racha de longitud 3.
- 4 rachas en total.

Al diferir en más de 1 la cantidad de (0s) y (1s) no se cumple con el 1er postulado de GOLOMB.

No cumple con la relación del número total de rachas, no cumple con el 2do postulado de GOLOMB.

Análisis de Correlación:

**Tabla 2.5** Autocorrelación fuera de fase para la secuencia 000101.

0	0	0	1	0	1	AC(k)
0	0	1	0	1	0	$AC(1) = (2-4)/6 = -1/3$
0	1	0	1	0	0	$AC(2) = (2-4)/6 = 1/3$
1	0	1	0	0	0	$AC(3) = (2-4)/6 = -1/3$
0	1	0	0	0	1	$AC(4) = (2-4)/6 = 1/3$
1	0	0	0	1	0	$AC(5) = (2-4)/6 = -1/3$

La autocorrelación no es constante. Luego, no cumple el 3er postulado de GOLOMB.

**S2: 011**

- Secuencia de Periodo  $T = 3$ .
- Hay 1(0) y 2 (1).
- 1 rachas de longitud 1.
- 1 rachas de longitud 2.
- 2 rachas en total.

Al diferir en 1 la cantidad de (0s) y (1s) sí se cumple con el 1er postulado de GOLOMB.

Si cumple con la relación al número total de rachas, por lo tanto, cumple con el 2do postulado de GOLOMB.

Análisis de Correlación:

**Tabla 2.6** Autocorrelación fuera de fase para la secuencia 011.

0	1	1	AC(k)
1	1	0	$AC(1)=(1-2)/3=-1/3$
1	0	1	$AC(2)=(1-2)/3=-1/3$

La autocorrelación es constante  $AC(K)= -1/3$ . Luego sí cumple el 3er postulado de GOLOMB.

**S3: 110011**

- Se genera una secuencia con Periodo  $T= 6$ .
- Hay 2(0) y 4 (1)
- 1 racha de longitud 2
- 1 racha de longitud 4
- Hay 1 cadena de ( 0 ) y 1 de ( 1 )
- 2 rachas en total

Al diferir en más de 1 la cantidad de (0s) y (1s) no se cumple con el 1er postulado de GOLOMB.

No cumple con la relación del número total de rachas, no cumple con el 2do postulado de GOLOMB.

Análisis de Correlación:

**Tabla 2.7** Autocorrelación fuera de fase para la secuencia 110011.

1	1	0	0	1	1	AC(K)
1	0	0	1	1	1	$AC(1)=(4-2)/6= 1/3$
0	0	1	1	1	1	$AC(1)=(2-4)/6= -1/3$
0	1	1	1	1	0	$AC(1)=(2-4)/6= -1/3$
1	1	1	1	0	0	$AC(1)=(2-4)/6= -1/3$
1	1	1	0	0	1	$AC(1)=(4-2)/6= 1/3$

La autocorrelación no es constante. Luego no cumple el 3er postulado de GOLOMB.

**Conclusión:**

**Tabla 2.8** Postulados de Golomb para secuencias generadas a partir de LFSR con Polinomio Factorizable.

Secuencia	Postulado 1	Postulado 2	Postulado 3
0 0 0 1 0 1	No cumple.	No cumple.	No cumple.
0 1 1	Cumple.	Cumple.	Cumple.
1 1 0 0 1 1	No cumple.	No cumple.	No cumple.

En cuanto al cumplimiento de los postulados de Golomb se puede notar que la mayoría de las secuencias no cumplen con los postulados de Golomb.

### **Generador LFSR con $f(x)$ irreducible**

Un polinomio Irreducible es aquel polinomio que no puede ser descompuesto en factores. Para ilustrarlo, veamos el siguiente ejemplo:

$$\text{Sea } f(x) = x^4 + x^3 + x^2 + x + 1$$

Es **imposible** factorizarlo en dos o más factores. El esquema del generador es el siguiente:

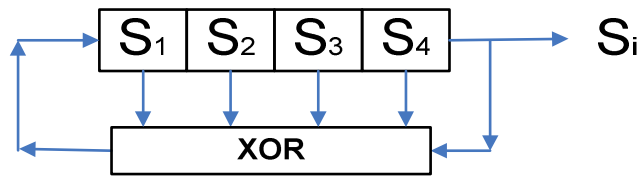


Figura 2.31 Generador LFSR con  $f(x)$  irreducible.

¿Qué periodo tendrá la secuencia generada a partir del LFSR con polinomio Irreducible?

Para responder esta interrogante, se realiza un ejemplo. En la siguiente figura se tiene un LFSR con el polinomio  $x^4 + x^3 + x^2 + x + 1$  y una semilla de 0001.

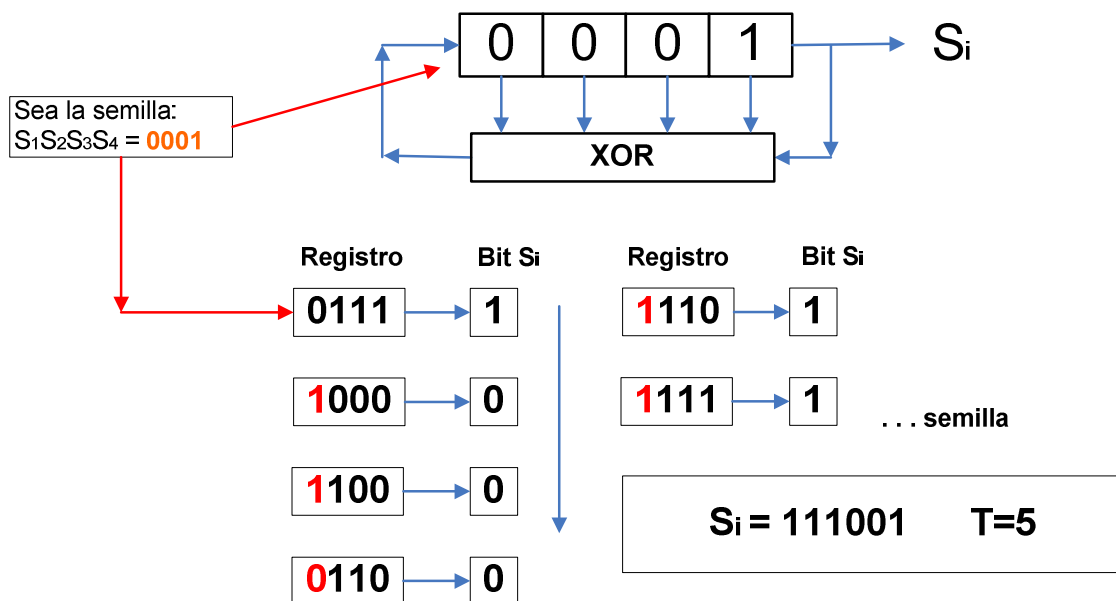


Figura 2.32 Generación de una secuencia binaria con  $f(x)$  irreducible y semilla de 0001.

Variando las semillas se tiene las siguientes secuencias:

**Tabla 2.9** Secuencias generadas por un LFSR con Polinomio Irreducible.

SEMILLA	SECUENCIA	PERIODO T
1 0 0 0	0 0 0 1 1	5
1 1 0 0	0 0 1 1 0	5
0 1 1 0	0 1 1 0 0	5
0 0 1 1	1 1 0 0 0	5
0 0 0 1	1 0 0 0 1	5
0 0 1 0	0 1 0 0 1	5
1 0 0 1	1 0 0 1 0	5
0 1 0 0	0 0 1 0 1	5
1 0 1 0	0 1 0 1 0	5
0 1 0 1	1 0 1 0 0	5
0 1 1 1	1 1 1 0 1	5
1 0 1 1	1 1 0 1 1	5
1 1 0 1	1 0 1 1 1	5
1 1 1 0	0 1 1 1 1	5
1 1 1 1	1 1 1 1 0	5

Como se puede observar se generan tres secuencias distintas con sus respectivas versiones desplazadas. Por lo que la secuencia resultante depende de la semilla. En cuanto al periodo se nota claramente que es constante ( $T=5$ ). Por lo tanto, se generan secuencias con ***periodo constante, independiente de la semilla.***

### ¿Cumple con los Postulados de Golomb?

Para responder a esta interrogante, se realiza el análisis de las siguientes secuencias:

**S1: 0 0 0 1 1**

- Secuencia de Periodo  $T=5$ .
- Hay 3 (0) y 2 (1).
- 1 racha de longitud 2.
- 1 racha de longitud 3.
- 2 rachas en total.

Al diferir en 1 la cantidad de (0s) y (1s). Sí se cumple con el 1<sup>er</sup> postulado de GOLOMB.

No cumple con la relación del número total de rachas, no cumple con el 2<sup>do</sup> postulado de GOLOMB.

Análisis de Autocorrelación.

**Tabla 2.10** Autocorrelación fuera de fase para la secuencia 00011.

0	0	0	1	1	AC(K)
0	0	1	1	0	$AC(1)=(3-1)/5= 2/5$
0	1	1	0	0	$AC(2)=(1-4)/5= -3/5$
1	1	0	0	0	$AC(3)=(1-4)/5= -3/5$
1	0	0	0	1	$AC(4)=(3-2)/5= 1/5$

La auto correlación AC(K) es diferente. Luego no se cumple el 3<sup>er</sup> postulado de GOLOMB.

**S2: 0 1 0 0 1**

- Secuencia de Periodo T=5.
- Hay 3(0) y 2 (1).
- 3 rachas de longitud 1.
- 1 racha de longitud 2.
- 4 rachas en total.

Al diferir en 1 la cantidad de (0s) y (1s) sí se cumple con el 1<sup>er</sup> postulado de GOLOMB.

No cumple con la relación del número total de rachas, no cumple con el 2<sup>do</sup> postulado de GOLOMB.

Análisis de Autocorrelación:

**Tabla 2.11** Autocorrelación fuera de fase para la secuencia 01001.

0	1	0	0	1	AC(K)
1	0	0	1	0	$AC(1)=(1-4)/5= -3/5$
0	0	1	0	1	$AC(2)=(3-2)/5= 1/5$
0	0	1	0	1	$AC(3)=(3-2)/5= 1/5$
1	0	1	0	1	$AC(4)=(1-4)/5= -3/5$



La auto correlación  $AC(k)$  es diferente. Luego no cumple el 3<sup>er</sup> postulado de GOLOMB.

**S3: 1 1 1 0 1**

- Secuencia de Periodo  $T=5$ .
- Hay 1(0) y 4 (1).
- 1 racha de longitud 1.
- 1 racha de longitud 4.
- 2 rachas en total.

Al diferir en más de 1 la cantidad de (0s) y (1s) no se cumple con el 1<sup>er</sup> postulado de GOLOMB.

No cumple con la relación del número total de rachas, no cumple con el 2<sup>do</sup> postulado de GOLOMB.

Análisis de autocorrelación:

**Tabla 2.12** Autocorrelación fuera de fase para la secuencia 11101.

1	1	1	0	1	AC(K)
1	1	0	1	1	$AC(1)=(3-2)/5= 1/5$
1	0	1	1	1	$AC(2)=(3-2)/5= 1/5$
0	1	1	1	1	$AC(3)=(3-2)/5= 1/5$
1	1	1	1	0	$AC(4)=(3-2)/5= 1/5$

La auto correlación es constante  $AC(K)= 1/5$ .Luego sí cumple el 3<sup>er</sup> postulado de GOLOMB.

**Conclusión:**

**Tabla 2.13** Postulados de Golomb para secuencias generadas a partir de LFSR con Polinomio Irreducible.

Secuencia	Postulado 1	Postulado2	Postulado 3
00011	Cumple.	No cumple.	No cumple.
01001	Cumple.	No cumple.	No cumple.
11101	No cumple.	No cumple.	Cumple.

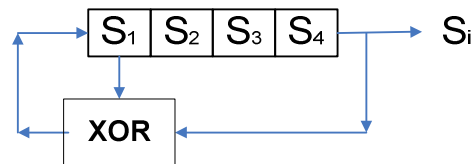
Como se puede ver, la mayoría de las secuencias generadas no cumplen con todos los Postulados de Golomb.

### Generador LFSR con $f(x)$ primitivo

Un polinomio de grado  $n$  es primitivo en  $GF(2^n)$  si y solo si

- Es irreducible (no es factorizable en polinomios menores) y
- Si divide exactamente a  $x^m+1$ , siendo  $m=2^n-1$ . Por definición, todos los polinomios primitivos de grado  $n$  tienen la forma  $x^n+\dots+1$ .

Sea  $f(x) = x^4 + x + 1$



**Figura 2.33** Generación de una secuencia binaria con  $f(x)$  primitivo.

- $f(x)$  no es factorizable como  $f(x_1) \cdot f(x_2)$  en módulo 2.
- Si divide exactamente a  $x^{15}+1$ , entonces el polinomio  $x^4 + x + 1$  es primitivo.

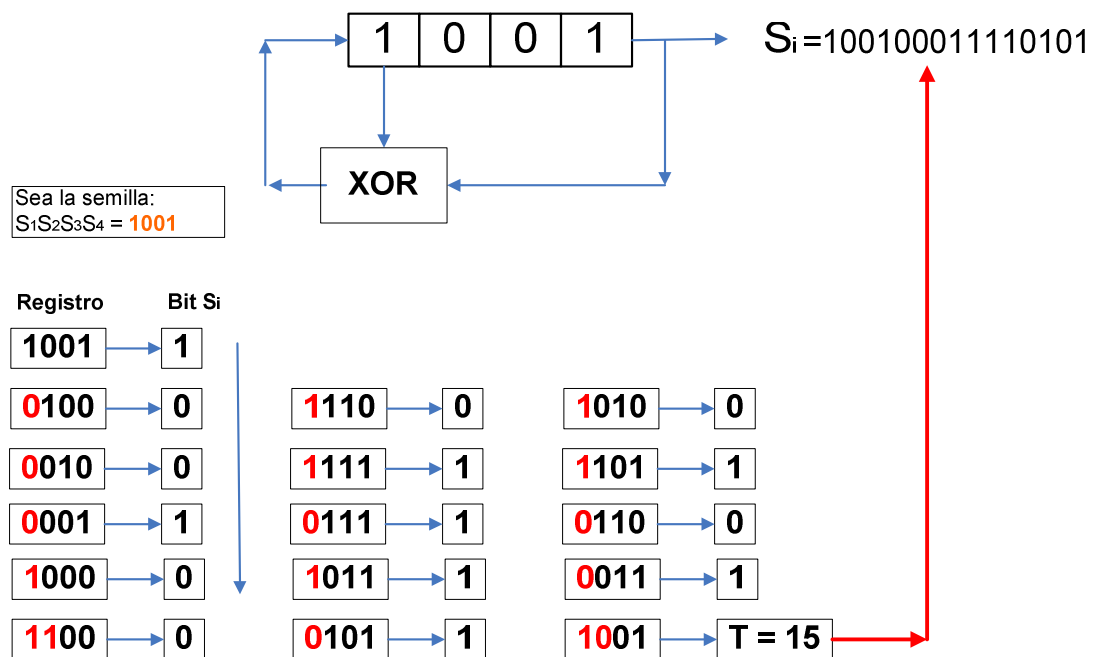
$x^{15}$	$+1$	$x^4+x+1$
$x^{15} + x^{12} + x^{11}$		$x^{11} + x^8 + x^7 + x^5 + x^3 + x^2 + x + 1$
$x^{12} + x^{11}$	$+1$	
$x^{12} + x^9 + x^8$		
$x^{11} + x^9 + x^8$	$+1$	
$x^{11} + x^8 + x^7$		
$x^9 + x^7$	$+1$	
$x^9 + x^6 + x^5$		
$x^7 + x^6 + x^5$	$+1$	
$x^7 + x^4 + x^3$		
$x^6 + x^5 + x^4 + x^3$	$+1$	
$x^6 + x^3 + x^2$		
$x^5 + x^4 + x^2 + 1$		
$x^5 + x^2 + x$		
$x^4 + x + 1$		
$x^4 + x + 1$		
<b>0</b>		<b>Residuo 0 Es primitivo</b>

**Figura 2.34** División entre los polinomios  $x^{15}+1$  para  $x^4 + x + 1$ .

Es una división exacta (residuo=0), por lo tanto, el polinomio  $x^4 + x + 1$  es primitivo.

**¿Qué periodo tendrá la secuencia generada a partir del LFSR con polinomio factorizable?**

Para responder la pregunta, se analiza el siguiente ejemplo:



**Figura 2.35** Generación de una secuencia binaria con  $f(x)$  primitivo y semilla 1001

Se genera una secuencia de periodo:  $T = 15$ .

Para diferentes semillas se generan las siguientes secuencias:

**Tabla 2.14** Secuencias generadas por un LFSR con Polinomio Primitivo.

Semilla	Secuencia	Periodo
1 0 0 1	100100011110101	15
0 1 0 0	001000111101011	15
0 0 1 0	010001111010110	15
0 0 0 1	100011110101100	15
1 0 0 0	000111101011001	15
1 1 0 0	001111010110010	15
1 1 1 0	011110101100100	15
1 1 1 1	111101011001000	15

0 1 1 1	111010110010001	15
1 0 1 1	110101100100011	15
0 1 0 1	101011001000111	15
1 0 1 0	010110010001111	15
1 1 0 1	101100100011110	15
0 1 1 0	011001000111101	15
0 0 1 1	110010001111010	15

Se puede observar que variando las semillas se generan secuencias de periodo  $T = 15$ . Además esta clase de LFSR genera una sola secuencia. Cada semilla genera una versión desplazada de la misma.

Por lo tanto, el periodo  $T$  de la secuencias generada no depende de la semilla y es de valor máximo  $T_{\max} = 2^n - 1$ .

### ¿Cumple con los Postulados de Golomb?

Para responder a esta interrogante, se realiza el análisis de la siguiente secuencia:

**S: 100100011110101**

- Hay 7(0) y 8(1).
- 4 rachas de longitud 1.
- 2 rachas de longitud 2.
- 1 rachas de longitud 3.
- 1 racha de longitud 4.
- 8 rachas en total.

Al diferir en 1 la cantidad de (0s) y (1s), sí se cumple con el 1<sup>er</sup> postulado de GOLOMB.

Sí cumple con la relación del número de rachas de acuerdo al total de rachas, sí cumple con el 2<sup>do</sup> postulado de GOLOMB.

Análisis de Autocorrelación:

**Tabla 2.15** Autocorrelación fuera de fase para la secuencia 100100011110101.

1	0	0	1	0	0	0	1	1	1	1	0	1	0	1	AC(K)
0	0	1	0	0	0	1	1	1	1	0	1	0	1	1	AC(1)=(7-8)/15= -1/15
0	1	0	0	0	1	1	1	1	0	1	0	1	1	0	AC(2)=(7-8)/15= -1/15
1	0	0	0	1	1	1	1	0	1	0	1	1	0	0	AC(3)=(7-8)/15= -1/15
0	0	0	1	1	1	1	0	1	0	1	1	0	0	1	AC(4)=(7-8)/15= -1/15
0	0	1	1	1	1	0	1	0	1	1	0	0	1	0	AC(5)=(7-8)/15= -1/15
0	1	1	1	1	0	1	0	1	1	0	0	1	0	0	AC(6)=(7-8)/15= -1/15
1	1	1	1	0	1	0	1	1	0	0	1	0	0	0	AC(7)=(7-8)/15= -1/15
1	1	1	0	1	0	1	1	0	0	1	0	0	0	1	AC(8)=(7-8)/15= -1/15
1	1	0	1	0	1	1	0	0	1	0	0	0	1	1	AC(9)=(7-8)/15= -1/15
1	0	1	0	1	1	0	0	1	0	0	0	1	1	1	AC(10)=(7-8)/15=-1/15
0	1	0	1	1	0	0	1	0	0	0	1	1	1	1	AC(11)=(7-8)/15=-1/15
1	0	1	1	0	0	1	0	0	0	1	1	1	1	0	AC(12)=(7-8)/15=-1/15
0	1	1	0	0	1	0	0	0	1	1	1	1	0	1	AC(13)=(7-8)/15=-1/15
1	1	0	0	1	0	0	0	1	1	1	1	0	1	0	AC(14)=(7-8)/15=-1/15

La Autocorrelación es constante  $AC(k)=-1/15$ , por lo tanto cumple con el Tercer postulado de Golomb.

**Tabla 2.16** Postulados de Golomb para secuencias generadas a partir de LFSR con Polinomio Primitivo.

Secuencia	Postulado 1	Postulado 2	Postulado 3
100100011110101	Cumple.	Cumple.	Cumple.

Los LFSRs asociados a polinomios primitivos cumplen con todos los postulados de Golomb.

### Características de las secuencias generadas a partir de los Polinomios Primitivos.

- Genera una secuencia de  $2^n - 1$  bits.
- Cumple con el postulado de Golomb 1 (G1):
  - Hay  $2^n$  bits 1 y  $2^n - 1$  bits 0.
- Cumple con el postulado de Golomb 2 (G2):
  - Rachas en Si de un LFSR con  $f(x)$  primitivo.

**Tabla 2.17** Rachas en una secuencia generada con un polinomio Primitivo.

Rachas de Longitud	Rachas de Ceros	Rachas de Unos
1	$2^{n-3}$	$2^{n-3}$
2	$2^{n-4}$	$2^{n-4}$
...	...	...
P	$2^{n-p-2}$	$2^{n-p-2}$
...	...	...
n-2	1	1
n-1	1	0
N	0	1
<b>TOTAL</b>	$2^{n-2}$	$2^{n-2}$

- Cumple con el postulado de Golomb 3 (G3):
  - La correlación fuera de fase  $AC(k)$  es constante.

### **Conclusión respecto de los diferentes tipos de Polinomios**

En la teoría de los LFSRs se menciona que, idealmente, el vector binario de las celdas pasa por todos los estados excepto la cadena de ceros que está prohibida, pues este no produce desplazamiento alguno. Por lo tanto, se producen  $2^n-1$  salidas que corresponden al **periodo máximo que puede tener una secuencia PN**.

Como se vio anteriormente, los generadores asociados a un polinomio factorizable, **la secuencia generada y el periodo de la misma depende de la semilla**. El periodo de las secuencias es menor al periodo máximo  $2^n-1$ . Todas las secuencias no tienen el mismo periodo. Existen secuencias con periodos secundarios divisores del periodo más alto generado y la mayoría de secuencias no cumplen con los Postulados de Golomb.

Para los generadores asociados a un polinomio no factorizable, **la secuencia generada depende de la semilla. El periodo de las secuencias generadas es constante y por lo tanto, es independiente de la semilla; pero es menor al periodo máximo  $2^n-1$** . Además la mayoría de secuencias no cumplen con todos los Postulados de Golomb.

Para el caso de LFSRs asociados a polinomios primitivos, este tipo de generador produce todos los estados lineales posibles del cuerpo de trabajo  $n$ , será el que nos entregue ***una sola secuencia de período máximo  $T_{max} = 2^n - 1$ , que cumple con todos los postulados de Golomb, independiente de la semilla.***

***Ya que el objetivo inicialmente planteado fue de obtener el período más grande posible, entonces, las estructuras que conviene utilizar para la generación de secuencias PN son aquellas que se asocian con un polinomio primitivo.***

Las secuencias generadas con este tipo de estructuras son conocidas como ***Secuencias de Longitud Máxima o M – Secuencias.***

### 2.1.2 SECUENCIAS DE LONGITUD MÁXIMA O M – SECUENCIAS<sup>51</sup>

Como ya se vio anteriormente, para que una comunicación permanezca invisible, debía cumplir con dos requisitos: que el pico de su espectro de potencias sea lo más pequeño posible: esto quedó ya ***cumplido*** mediante el proceso de ensanchamiento. Por otro lado, no debería ser detectado mediante algún algoritmo de búsqueda de periodicidades como el de correlación: esto quedaría ***resuelto*** mediante el ***empleo de M-Secuencias*** ya que su período es tan largo (***período máximo***  $T_{max}=2^n-1$ ) que en uno de ellos, la secuencia parece aleatoria. Esto produce a su vez un espectro de potencias plano, similar al del ruido Gaussiano blanco. Con todo esto, se garantiza que antes y después de la comunicación, el canal quede lo más similar posible evitando que el enemigo lo detecte y por tanto lo intervenga o lo sabotee.

#### **Propiedades de las secuencias de longitud máxima (m-secuencias)**

- 1) Dado un registro de  $n$  filp flops, el número de secuencias de longitud  $2^n-1$  es limitado, tal como se muestra en la siguiente tabla. Por ejemplo para

---

<sup>51</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p.211-213.

$n=4$ , solo hay 2 secuencias de longitud 15; para  $n=5$  y  $n=6$  hay solo 6 secuencias. Habrá más secuencias si  $n$  es un número primo.

**Tabla 2.18** Polinomios Primitivos para generar M-secuencias.

Número de etapas del Registro de Desplazamiento $n$	Tamaño de la secuencia $N=2^n-1$	Número de m-secuencias	Polinomio Generador
2	3	1	$1+x+x^2$
3	7	2	$1+x+x^2$ $1+x^2+x^3$
4	15	2	$1+x+x^4$ $1+x^3+x^4$ $1+x^2+x^5$ $1+x^3+x^5$
5	31	6	$1+x+x^2+x^3+x^5$ $1+x+x^2+x^4+x^5$ $1+x+x^3+x^4+x^5$ $1+x^2+x^3+x^4+x^5$ $1+x+x^6$ $1+x+x^3+x^4+x^6$ $1+x^5+x^6$
6	63	6	$1+x+x^2+x^5+x^6$ $1+x^2+x^3+x^5+x^6$ $1+x+x^4+x^5+x^6$ $1+x+x^7$ $1+x^3+x^7$ $1+x+x^2+x^3+x^7$ $1+x^4+x^7$ $1+x^2+x^3+x^4+x^7$ $1+x+x^2+x^5+x^7$ $1+x+x^3+x^5+x^7$ $1+x^3+x^4+x^5+x^7$
7	127	18	$1+x^2+x^3+x^4+x^5+x^7$ $1+x^6+x^7$ $1+x+x^3+x^6+x^7$ $1+x+x^4+x^6+x^7$ $1+x^2+x^4+x^6+x^7$ $1+x^2+x^5+x^6+x^7$ $1+x+x^2+x^3+x^5+x^6+x^7$ $1+x^4+x^5+x^6+x^7$ $1+x+x^2+x^4+x^5+x^6+x^7$ $1+x+x^2+x^3+x^4+x^5+x^6+x^7$



			$1+x^2+x^3+x^4+x^8$
			$1+x+x^3+x^5+x^8$
			$1+x^2+x^3+x^5+x^8$
			$1+x^2+x^3+x^6+x^8$
			$1+x^2+x^3+x^4+x^6+x^8$
			$1+x+x^5+x^6+x^8$
			$1+x^2+x^5+x^6+x^8$
			$1+x^3+x^5+x^6+x^8$
8	255	16	$1+x^4+x^5+x^6+x^8$
			$1+x+x^2+x^7+x^8$
			$1+x^2+x^3+x^7+x^8$
			$1+x^3+x^5+x^7+x^8$
			$1+x+x^6+x^7+x^8$
			$1+x+x^2+x^3+x^6+x^7+x^8$
			$1+x+x^2+x^3+x^6+x^7+x^8$
			$1+x+x^2+x^5+x^6+x^7+x^8$
			$1+x^2+x^4+x^5+x^6+x^7+x^8$
9	511	48	$1+x^4+x^9$ (ejemplo)
10	1023	60	$1+x^3+x^{10}$ (ejemplo)

- 2) Si una m-secuencia se suma módulo 2 con una réplica de ella misma desplazada, la suma es otra réplica de la misma secuencia con otro desplazamiento.
- 3) M-secuencias del mismo tamaño pueden ser derivadas de otra mediante el proceso de la decimación. La decimación consiste en tomar cada  $q^{\text{th}}$  elemento de la secuencia para formar una **nueva secuencia binaria**. La secuencia generada de esta manera, depende del valor escogido de  $q$ . Consecuentemente, **no todos los valores que tome  $q$ , ni todas las secuencias decimadas, formarán una nueva m-secuencia**.
- 4) Correlación Parcial: Cuando el período de la m-secuencia es muy largo, sobre todo comparado con la longitud del mensaje, que se lo conocerá como  $M$ , se puede asumir independencia entre un elemento  $a_n$  y uno  $a_{n+k}$ , de forma que el valor medio de la correlación parcial resulta ser igual a  $M$  para  $k=0, N, 2N, \dots$  y cero para el resto.

Como se sabe, solamente los polinomios primitivos generan secuencias de Longitud máxima. Actualmente los polinomios primitivos se encuentran tabulados

en muchos libros. Si no se tiene a la mano estas tablas, Matlab también provee un grupo de estos polinomios a través de una función específica.

### ¿Cómo se obtienen los polinomios primitivos en MATLAB?

Se los obtiene mediante la función **Primpoly**: la captura de la ayuda es la siguiente:

```

primpoly
Find primitive polynomials for a Galois field
Syntax
pr = primpoly(m)
pr = primpoly(m,opt)
pr = primpoly(m...,'nodisplay')

```

**Figura 2.36** Función Primpoly de Matlab para encontrar los polinomios primitivos.

Retorna los polinomios primitivos del campo de Galois  $GF(2^m)$ , donde  $m$  es un entero entre 2 y 16. La ventana de comandos muestra el polinomio usando  $D$  como variable.

**Ejemplo:** Obtener los polinomios primitivos asociados a un registro de desplazamiento de 8 flip flops.

```

m = 8;
polinomios=primpoly(m,'all')%polinomios primitivos para m=8.

```

El programa anterior genera el siguiente resultado:

```

Primitive polynomial(s) =

D^8+D^4+D^3+D^2+1
D^8+D^5+D^3+D^1+1
D^8+D^5+D^3+D^2+1
D^8+D^6+D^3+D^2+1
D^8+D^6+D^4+D^3+D^2+D^1+1
D^8+D^6+D^5+D^1+1
D^8+D^6+D^5+D^2+1
D^8+D^6+D^5+D^3+1
D^8+D^6+D^5+D^4+1
D^8+D^7+D^2+D^1+1
D^8+D^7+D^3+D^2+1
D^8+D^7+D^5+D^3+1
D^8+D^7+D^6+D^1+1
D^8+D^7+D^6+D^3+D^2+D^1+1
D^8+D^7+D^6+D^5+D^2+D^1+1
D^8+D^7+D^6+D^5+D^4+D^2+1

```

**Figura 2.37** Polinomios primitivos de grado  $m=8$  en Matlab.

Como se puede ver en el resultado anterior, existen 16 polinomios primitivos de grado  $m=8$ .

Comprendida la teoría de las secuencias PN y M-secuencias, a continuación se detallan algunos programas que permiten obtener estas secuencias en Matlab.

### Generación de Secuencias PN en Matlab

Ya que no existe una función que permita generar la secuencia PN directamente, se desarrolló el programa **cap2\_cod6\_secuencias\_pn.m** (ver **Anexos Cap 2**), en el cual se deben ingresar los coeficientes  $a_n$  del polinomio que define al generador LFSR, de forma **descendente**. El polinomio asociado a un generador LFSR está dado de la siguiente manera:

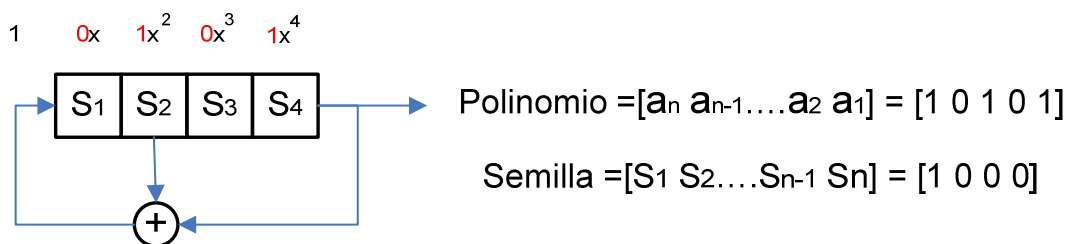
$$P(X)=a_nX^n+a_{n-1}X^{n-1}+ \dots + a_1X^1+a_0 ; \quad a_i \in (0,1)$$

Por ejemplo para el polinomio  $P(X)=X^4+X^2+1$  los valores de los coeficientes  $a_n$  son:

$$P(X)=1X^4+0X^3+1X^2+0X^1+1$$

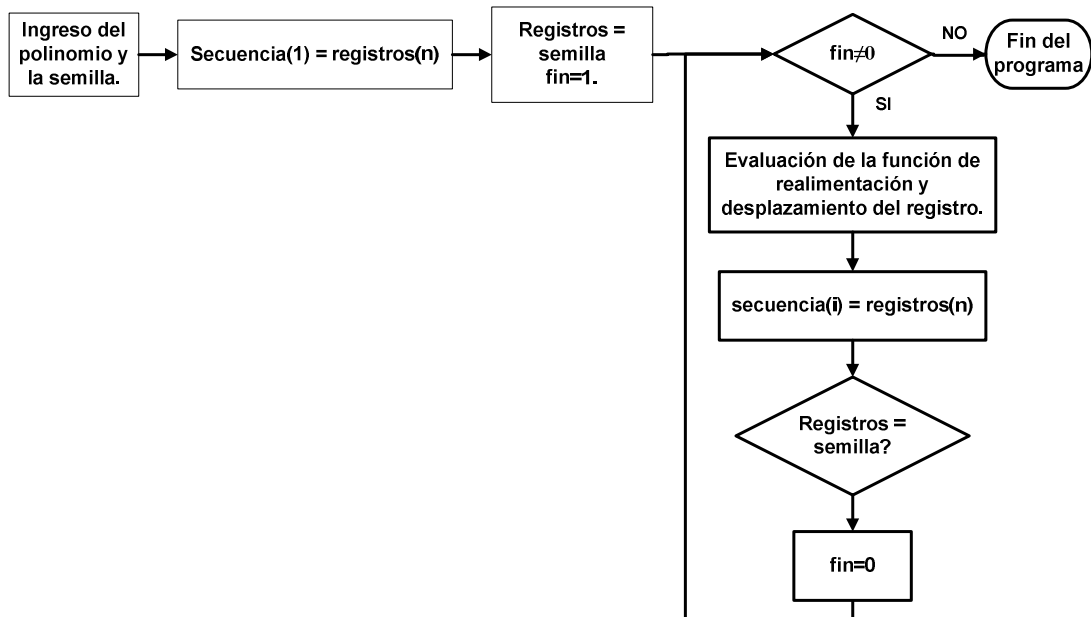
Por lo tanto el vector a ingresar será: [1 0 1 0 1].

Además se debe ingresar la semilla. La semilla define el estado inicial de los flip flops del registro de desplazamiento. Se debe ingresar un vector que contenga los valores de la semilla de forma **ascendente**. El primer valor de la semilla corresponde al flip flop de menor orden (en este caso  $S_1$ ). Para este ejemplo el valor de la semilla a ingresar es: [1 0 0 0].



**Figura 2.38** Definición de los parámetros del programa cap2\_cod6\_secuencias\_pn.m

El diagrama para la generación de la secuencia PN es el siguiente:



**Figura 2.39** Diagrama para la generación de Secuencias PN.

La siguiente figura muestra el resultado que se obtiene al ejecutar el programa *cap2\_cod6\_secuencias\_pn.m*.

```

Ingrese el polinomio del LFSR: [1 0 1 0 1]
Semilla: [1 0 0 0]
La secuencia generada es:

secuencia =

    0    0    0    1    0    1
  
```

**Figura 2.40** Resultado del Programa *cap2\_cod6\_secuencias\_pn.m*

### **Generación de M-secuencias en Matlab**

En un apartado anterior, se vio cómo generar un período de un registro de desplazamiento cualquiera. En este apartado se verá el procedimiento para generar un período de una m-secuencia.

Para generar un período de una m-secuencia, se puede emplear el programa *cap2\_cod6\_secuencias\_m.m*. La única diferencia es que ahora el polinomio

ingresado debe ser de tipo **primitivo**, pues solo los polinomios primitivos generan m-secuencias.

Para simplificar la generación de las secuencias pn, al programa **cap2\_cod6\_secuencias\_pn.m** se lo convirtió en la función **sec\_pn(polynomio).m** (ver **Anexos Cap 2**), donde se debe ingresar solamente el polinomio del LFSR generador. La semilla está establecida como un vector de n-1 ceros y un 1; siendo n el grado del polinomio o numero de celdas del registro de desplazamiento.

Como ya se explicó anteriormente, la **función directa de Matlab** para encontrar los polinomios primos asociados a un registro de desplazamiento de N flip flops, es **primpoly**:

**Ejemplo:** Generar un período de una m-secuencia asociado a un registro de desplazamiento de n=5. En la siguiente figura se generan los polinomios primitivos de grado n=5 con la función primpoly.

```
>> n=5;
>> primpoly(n,'all')

Primitive polynomial(s) =

D^5+D^2+1
D^5+D^3+1
D^5+D^3+D^2+D^1+1
D^5+D^4+D^2+D^1+1
D^5+D^4+D^3+D^1+1
D^5+D^4+D^3+D^2+1
```

**Figura 2.41** Polinomios primitivos de grado n=5 con la función primpoly.

Tomando uno de los polinomios de la figura y empleando la función **sec\_pn(polynomio).m**, se puede generar un periodo de una m-secuencia. Por ejemplo para el polinomio  $x^5+x^2+1$ , se tiene una semilla de [0 0 0 0 1]. Por lo tanto la m-secuencia generada en Matlab es:

```

>> polinomio=[1 0 0 1 0 1];
>> m_secuencia=sec_pn(polinomio)

m_secuencia =

Columns 1 through 15
    1     0     0     0     0     1     0     1     0     1     1     1     0     1     1

Columns 16 through 30
    0     0     0     1     1     1     1     1     0     0     1     1     0     1     0

Column 31
    0

```

**Figura 2.42** Generación de una M-secuencia en Matlab.

Del resultado anterior se observa que el periodo de la m-secuencia para el polinomio  $x^5+x^2+1$  es máximo e igual a  $N=2^5-1=31$ .

### **Seguridad de la comunicación utilizando M-secuencias**

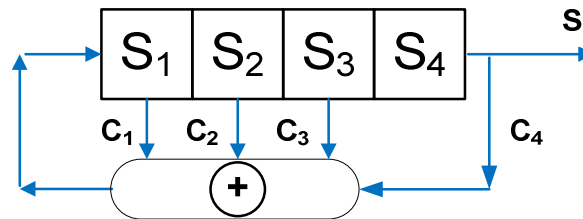
Una de las características importantes que debe tener una secuencia, es que ésta sea indescifrable por un usuario no autorizado. Como se pudo ver anteriormente las m-secuencias son generadas por LFSR con un polinomio primitivo asociado. Pero esta estructura de generación tiene una debilidad y puede ser recreada mediante la resolución de un sistema de ecuaciones a partir de  $2 \cdot n$  bits consecutivos de un m-secuencia de  $2^n-1$  bits, donde  $n$  es el número de celdas del generador LFSR.

Por ejemplo, si se conoce sólo  $2 \cdot 10 = 20$  bits en un sistema de 10 celdas con un período  $2^{10}-1 = 1.023$ , es posible encontrar las conexiones de las celdas o valores de  $C_i$  y generar la secuencia completa  $S_i$ .

Esta debilidad es la que usa el ataque conocido como algoritmo de ***Berlekamp-Massey***.

### Ejemplo de Ataque de Berlekamp-Massey <sup>52</sup>

Para conocer el algoritmo del ataque de Berlekamp-Massey se considera un generador LFSR de cuatro celdas. Si se conocen  $2 \cdot n = 8$  bits  $b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8$  del LFSR  $C_1 C_2 C_3 C_4$ .



**Figura 2.43** Ejemplo de ataque de Berlekamp-Massey con LFSR de 4 celdas.

En un generador LFSR primero se transmite  $S_4 S_3 S_2 S_1$  (semilla) y luego los bits  $S_5 S_6 S_7 S_8$ . Si se asignan valores de esos  $2 \cdot n = 8$  bits  $b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 = S_4 S_3 S_2 S_1 S_5 S_6 S_7 S_8$  es posible resolver este sistema de ecuaciones:

$$S_5 = C_1 \cdot S_1 \oplus C_2 \cdot S_2 \oplus C_3 \cdot S_3 \oplus C_4 \cdot S_4$$

$$S_6 = C_1 \cdot S_5 \oplus C_2 \cdot S_1 \oplus C_3 \cdot S_2 \oplus C_4 \cdot S_3$$

(Ec 2.8)

$$S_7 = C_1 \cdot S_6 \oplus C_2 \cdot S_5 \oplus C_3 \cdot S_1 \oplus C_4 \cdot S_2$$

$$S_8 = C_1 \cdot S_7 \oplus C_2 \cdot S_6 \oplus C_3 \cdot S_5 \oplus C_4 \cdot S_1$$

Si  $b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 = 11001000$  son correlativos y como hay cuatro celdas y primero se transmite la semilla, entonces:

$$S_1 = 0 \quad S_5 = 1$$

$$S_2 = 0 \quad S_6 = 0$$

(Ec 2.9)

$$S_3 = 1 \quad S_7 = 0$$

<sup>52</sup> <http://www.scribd.com/doc/4679956/Seguridad-Informatica-y-Criptografia-Sistemas-de-Cifra-en-Flujo>

$$S_4 = 1 \quad S_8 = 0$$

Reemplazando en el sistema de ecuaciones, se tiene:

$$1 = C_1 \cdot 0 \oplus C_2 \cdot 0 \oplus C_3 \cdot 1 \oplus C_4 \cdot 1$$

$$0 = C_1 \cdot 1 \oplus C_2 \cdot 0 \oplus C_3 \cdot 0 \oplus C_4 \cdot 1$$

(Ec 2.10)

$$0 = C_1 \cdot 0 \oplus C_2 \cdot 1 \oplus C_3 \cdot 0 \oplus C_4 \cdot 0$$

$$0 = C_1 \cdot 0 \oplus C_2 \cdot 0 \oplus C_3 \cdot 1 \oplus C_4 \cdot 0$$

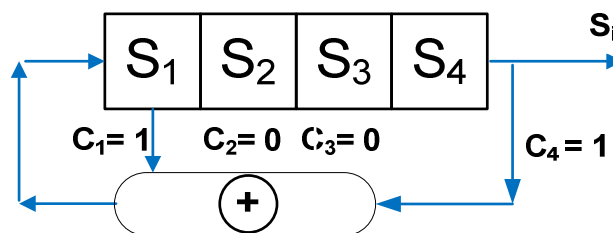
Resolviendo el sistema de ecuaciones, se obtienen las conexiones  $C_i$  del Generador LFSR.

$$C_1 = 1 \quad C_2 = 0$$

(Ec 2.11)

$$C_3 = 0 \quad C_4 = 1$$

Por lo tanto, el polinomio generador es:  $x^4 + x + 1$ .



**Figura 2.44** LFSR calculado con ataque de Berlekamp-Massey.

Como se puede ver, ahora ya se conoce la estructura del Generador LFSR, y por lo tanto se puede generar la  $m$ -secuencia y queda demostrada la vulnerabilidad de la información al utilizar este tipo de secuencias.

### **Simulación en MATLAB del Ataque de Berlekamp-Massey**

**Programa Berlekamp.m (ver Anexos Cap 2)**



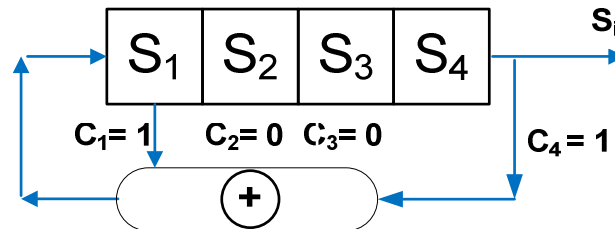
Para esta simulación, se utiliza el ejemplo anteriormente explicado del Ataque Berlekamp-Massey, donde solo se conocen los valores 11001000 de una secuencia PN generada con un LFSR de 4 celdas.

```

Grado del polinomio generador: 4
Bits correlativos: [1 1 0 0 1 0 0 0]
El polinomio generador es:
    1    1    0    0    1
  
```

**Figura 2.45** Resultado del Algoritmo de Berlekamp Massey.

Por lo tanto, el polinomio generador es:  $1+1x+0x^2+0x^3+1x^4$



**Figura 2.46** Cálculo de elementos desconocidos.

### Conclusión:

Las m-secuencias son vulnerables ante ataques de usuarios no autorizados. Por lo tanto, se deben buscar secuencias que sean más resistentes a ataques como el algoritmo de Berlekamp-Massey. Una solución a estos ataques es la combinación de dos o más secuencias para aumentar la complejidad de la secuencia resultante.

### Utilización de las secuencias como técnica de acceso multiusuario

Ahora es preciso plantear un nuevo objetivo para las secuencias. El mismo, es el de conseguir que la modulación mediante espectro ensanchado, sirva para múltiples usuarios y se transforme en una nueva técnica de acceso al medio.

Como se verá en el capítulo 3, si se desea obtener un sistema basado en códigos que permita implementar un **sistema multiusuario**, es necesario que la

**correlación cruzada** entre las secuencias participantes **sea baja** y que la **autocorrelación** de cada una de ellas **sea alta** de modo que cuando se desee recibir la comunicación de un usuario determinado, se aplique el código correspondiente y solo en este caso (autocorrelación) se dispare un pico grande. En los otros casos (correlación cruzada), se disparará un pico muy pequeño. De esta manera, se habrá creado un sistema que permita reconocer a un usuario deseado.

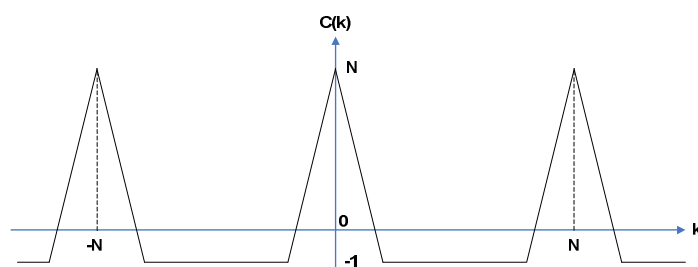
Con estas ideas, es preciso analizar las funciones de Autocorrelación y Correlación Cruzada de las M-secuencias para determinar qué conjunto de secuencias son aptas para conformar un sistema multiusuario.

### Autocorrelación y Correlación Cruzada de las M – secuencias

La función de **autocorrelación** de estas **M-secuencias** es periódica y tiene dos valores. Para una secuencia  $\{a_k\}$  de 1s y -1s la Autocorrelación  $C(k)$  se calcula como:

$$C(k) = \sum_{n=1}^N a_n a_{n+k} = \begin{cases} N & k = 0, N, 2N \\ -1 & \text{el resto} \end{cases}$$

Siendo  $N=2^n-1$ , el período de la M-secuencia.



**Figura 2.47** Autocorrelación de una M-secuencia.

Esto está asociado con el hecho de que una secuencia periódica al ser comparada con cualquier versión desplazada de ella, tendrá una diferencia entre acuerdos y desacuerdos igual a 1.

Para la **Autocorrelación normalizada**, la cual se define como  $C(k)/N$ , sus valores serán 1 para  $k=0$  ( $,N, 2N$ ) y  $-1/N$  para el resto de valores de  $k$ .

Como se ve, la Autocorrelación de una m-secuencia, tiene el mismo comportamiento que el del ruido (solo se parece a sí misma en desplazamiento cero). Esto es justamente lo que se buscaba para hacer que el espectro se parezca al del ruido, es decir, desparramado a lo largo de todas las frecuencias (espectro disperso).

**Ejemplo:** Dada la M-secuencia, encontrar el gráfico de autocorrelación.

$$\text{M-Secuencia} = [1 \ 1 \ -1 \ -1 \ -1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ 1 \ -1 \ 1 \ 1]$$

Para cuando la M-secuencia se encuentra en fase, es decir, se compara con la misma sin desplazamiento alguno, la Autocorrelación toma el siguiente valor:

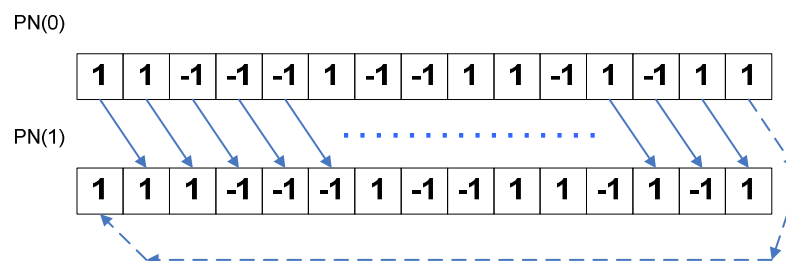
$$\text{PN} = \{a_1, a_2, \dots, a_{15}\} =$$

1	1	-1	-1	-1	1	-1	-1	1	1	-1	1	-1	1	1
---	---	----	----	----	---	----	----	---	---	----	---	----	---	---

**Figura 2.48** M-Secuencia sin desplazamiento ( $k=0$ )

$$C[0] = \sum_i a_i a_i = 15$$

La Autocorrelación, toma el valor máximo de 15. Para obtener el resto de valores de Autocorrelación, se debe desplazar la secuencia PN  $k$  veces y compararla con la secuencia original, tal como se muestra en la figura:



$$C[1] = \sum_i a_i a_{i+1} = -1$$

**Figura 2.49** Cálculo de la Autocorrelación de una M-secuencia.

Repitiendo el proceso para los otros valores de desplazamiento, se obtienen 15 valores que se los puede graficar. Resolviendo el ejemplo en Matlab, se tiene:

```
%definición de la m-secuencia
sec1=[1 1 -1 -1 -1 1 -1 -1 1 1 -1 1 -1 1 1];
sec2=sec1; %para realizar la autocorrelación sec2=sec1
[ind,corr]=xcorrelacion(sec1,sec2) %llamada a la función que
realiza la correlación.
plot(ind,corr) %graficar la función de autocorrelación para la m-
secuencia.
grid on
title('Autocorrelación de M-Secuencias')
```

Las figuras muestran los resultados obtenidos en Matlab para el análisis de Autocorrelación de una m-secuencia.

```
ind =

Columns 1 through 15
-14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0

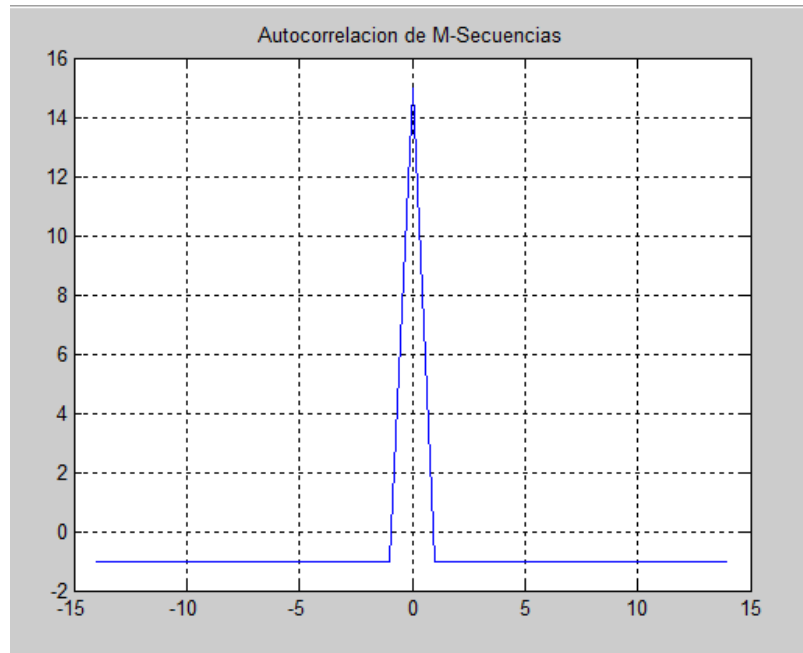
Columns 16 through 29
1 2 3 4 5 6 7 8 9 10 11 12 13 14

corr =

Columns 1 through 15
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 15

Columns 16 through 29
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

**Figura 2.50** Cálculo de la Autocorrelación de una M-secuencia en Matlab.



**Figura 2.51** Gráfica de la Autocorrelación de una M-secuencia en Matlab

Se puede observar claramente que la Autocorrelación de una M-secuencia tiene un pico máximo  $N=15$  solamente cuando la m-secuencia no se encuentra desplazada, para cualquier desplazamiento, toma el valor de  $-1$ .

Una secuencia PN y en particular las m-secuencias, deberían tener un comportamiento similar al ruido. Para **m-secuencias correspondientes a registros de muchos flip flops**, al tener un **período muy largo**, la tasa  $Ac(0)/Ac(k) = -1/N$  es muy pequeña y en este caso, se acerca a una secuencia PN ideal.

En algunas aplicaciones, la **correlación cruzada** de una secuencia PN, es tan importante como la autocorrelación. Por ejemplo, como se verá más adelante en el CAP. 3, en CDMA, a cada usuario se le asigna una secuencia particular. Idealmente, las secuencias PN deberían ser todas **mutuamente ortogonales** de modo que el nivel de interferencia experimentado por un usuario debido a la transmisión de los otros usuarios debería ser cero. Sin embargo, las secuencias PN usadas en la práctica, exhiben alguna correlación. Para el caso particular de las **m-secuencias**, la **correlación cruzada** entre un par de m-secuencias del mismo período, puede tener picos grandes.

Por ejemplo, para  $n=4$ , existen dos polinomios primitivos generadores, por lo tanto dos m-secuencias. Si se hace la correlación cruzada de las dos secuencias se tiene:

```
>> n=4;
>> primpoly(n,'all')

Primitive polynomial(s) =

D^4+D^1+1
D^4+D^3+1
```

**Figura 2.52** Polinomios primitivos para  $n=4$ .

De la figura se obtienen los dos polinomios primitivos:  $x^4+x+1$  y  $x^4+x^3+1$ , los cuales se los utiliza para generar las m-secuencias en Matlab.

```
>> pol1=[1 0 0 1 1];
>> pol2=[1 1 0 0 1];
>> msec1=bipolar(sec_pn(pol1))

msec1 =

-1  1  1  1 -1 -1 -1 -1  1 -1  1 -1 -1  1  1

>> msec2=bipolar(sec_pn(pol2))

msec2 =

-1  1  1  1 -1  1  1 -1 -1  1 -1  1 -1 -1 -1
```

**Figura 2.53** M-secuencia obtenidas en Matlab.

Haciendo la correlación cruzada entre las dos m-secuencias se tiene:

```
>> [ind corr]=xcorrelacion(msec1,msec2);
>> corr

corr =

Columns 1 through 15
  3  -1  7  3  -5  -1  3  7  -5  3  -5  -5  -1  -1  -1

Columns 16 through 29
 -1  -1  -5  -5  3  -5  7  3  -1  -5  3  7  -1  3
```

**Figura 2.54** Correlación Cruzada entre M-secuencias con  $n=4$ .

Analíticamente se observa que para estas secuencias existen tres valores de Correlación cruzada, siendo el máximo valor o pico de correlación Cruzada, 7. Ahora, gráficamente se tiene:



**Figura 2.55** Correlación Cruzada entre M-secuencias con  $n=4$ .

Como se sabe, la autocorrelación de una M-secuencia tiene un único pico. Para estas secuencias el pico tiene el valor de la longitud de la secuencia,  $N = 2^n - 1$ . Por lo tanto, el pico de Autocorrelación es  $N=15$ .

Si relacionamos el pico de Autocorrelación con el pico máximo de Correlación cruzada:

$$\frac{Ac_{max}}{Ac} = \frac{7}{15} = 0,4667$$

Se puede notar que el pico de Correlación cruzada entre las dos secuencias es significativamente grande respecto al de Autocorrelación (casi la mitad).

En la siguiente tabla, se lista el número de m-secuencias de longitud o periodo  $N = 2^n - 1$  para valores de  $3 \leq n \leq 12$  ( $n$  es el número de etapas que tiene el generador LFSR), el valor de correlación cruzada máxima ( $R_{max}$ ) entre pares de m-secuencias y la relación del valor de correlación cruzada máxima respecto al valor máximo de Autocorrelación  $Ac(0)$ :

**Tabla 2.19** Correlación de las M-secuencias.

n	N = 2 <sup>n</sup> -1	m-secuencias		
		Número de M-secuencias	Rmax	Rmax/Ac(0)
3	7	2	5	0,71
4	15	2	7	0,47
5	31	6	11	0,35
6	63	6	23	0,36
7	127	18	41	0,32
8	255	16	95	0,37
9	511	48	113	0,22
10	1023	60	383	0,37
11	2047	176	287	0,14
12	4095	144	1407	0,34

Se observa que el número de m-secuencias (de periodo  $T = 2^n - 1$ ), aumenta rápidamente con n.

Nuevamente se observa que para la mayoría de las secuencias, el pico de la correlación cruzada (Rmax), tiene un valor grande comparado con el pico de la función de Autocorrelación (Ac(0)).

Para que un grupo de secuencias **sea apto** para comunicaciones multiusuario, es necesario que la función de correlación cruzada entre dos secuencias cualesquiera de la misma familia sea pequeña, por lo que deben buscarse subconjuntos de secuencias que tengan esta propiedad. Tal como se verá a continuación.

### 2.1.3 M-SECUENCIAS PREFERIDAS<sup>53</sup>

La correlación cruzada entre dos secuencias de longitud máxima generadas por dos polinomios primitivos diferentes pueden tener tres, cuatro o más valores, lo que dificulta encontrar un grupo suficientemente grande de secuencias para comunicaciones multiusuario.

---

<sup>53</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p.216-217.



En los años 60, Gold y Kasami probaron que existían ciertos pares de m-secuencias con  $n \neq 4$  y sus múltiplos, **denominadas secuencias preferidas**, que tienen 3 posibles valores de Correlación Cruzada:

$$R = [-1, -t(n), t(n) - 2] \quad (\text{Ec 2.12})$$

Donde:

$$t(n) = 1 + 2^{\frac{n+1}{2}} \text{ para } n \text{ impar} \quad (\text{Ec 2.13})$$

$$t(n) = 1 + 2^{\frac{n+2}{2}} \text{ para } n = 2 \bmod 4$$

Por lo tanto, en este tipo de secuencias, se cumple la siguiente desigualdad:

$$R \leq t(n) \quad (\text{Ec 2.14})$$

Donde R es la función de Correlación cruzada.

### ¿Cómo saber que dos m-secuencias son preferidas?

Hay dos formas para saber si un par de m-secuencias son preferidas:

#### Primera forma:

- Verificar que el grado del polinomio generador,  $n$ , no sea divisible para 4.
- Observar que el par de m-secuencias, tengan tres valores de Correlación Cruzada. Y que cumplan con la desigualdad  $R \leq t(n)$ .

#### Segunda forma:

- Verificar que el grado del polinomio generador,  $n$ , no sea divisible para 4.
- El par designado puede ser seleccionado como la m-secuencia  $u$  y su versión decimada  $v=u[q]$ . Donde:
  - $q$  es impar.
  - $q = 2^k + 1$  o  $q = 2^{2k} - 2^k + 1$ .
  - $k$  es típicamente toma los valores de 1 y 2. Estos valores de  $k$  hacen que  $q=3, 5, 13$ .
  - $v$  es obtenida por el muestreo de cada  $q^{\text{th}}$  símbolo de  $u$ .

- El máximo común divisor de n y k debe tomar los siguientes valores:

$$\gcd(n, k) = \begin{cases} 1, & n \equiv 1 \pmod{2}; \quad n \text{ es impar} \\ 2, & n \equiv 2 \pmod{4}; \quad n \text{ es par y no es divisible para 4} \end{cases} \quad (\text{Ec 2.15})$$

### Ejemplo

#### Primera forma:

Se consideran los siguiente polinomios generadores,  $x^3+x+1$  y  $x^3+x^2+1$ .

Con una semilla = 001 para los dos generadores, se obtienen las siguientes secuencias:

$$\text{msec1}=[0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1] \text{ y } \text{msec2}=[0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1].$$

Para realizar la correlación cruzada de las secuencias, éstas deben estar en forma bipolar. Utilizando una lógica bipolar negativa se tienen:

$$\text{msec1}=[1 \ 1 \ -1 \ -1 \ -1 \ 1 \ -1] \text{ y } \text{msec2}=[1 \ 1 \ -1 \ 1 \ -1 \ -1 \ -1].$$

Ahora veamos si estas secuencias son preferidas:

1. Se puede notar que  $n=3$  y por lo tanto se cumple que el grado de los polinomios no es divisible para 4.
2. Se debe verificar, entonces que las dos m-secuencias cumplen con el **valor máximo de correlación cruzada** y con que la **correlación cruzada solo tenga tres valores**, es decir:

$$R \leq t(n)$$

(Ec 2.16)

$$R=[-1, -t(n), t(n)-2]$$

Para  $n=3$ ; n impar:

$$t(n) = 2^{\frac{n+1}{2}} + 1 = 2^{\frac{3+1}{2}} + 1 = 5 \quad (\text{Ec 2.17})$$

Por lo tanto:

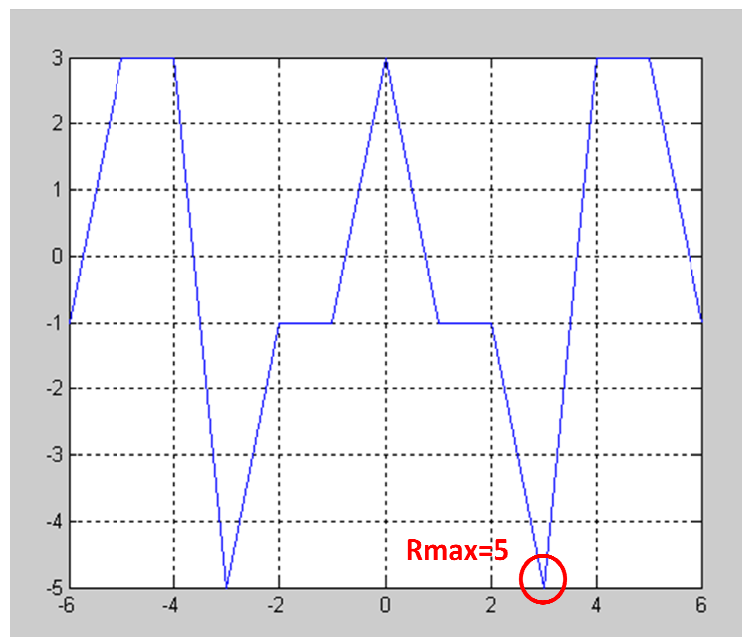
$$R \leq 5 \quad (\text{Ec 2.18})$$

Y los valores de correlación cruzada pueden ser:

$$R = [-1, -5, 3] \quad (\text{Ec 2.19})$$

Empleando la función de Matlab ***xcorrelacion*** desarrollada en un apartado anterior, se puede verificar que los valores de correlación cruzada corresponden a los valores teóricos.

```
%se definen las m-secuencias con las que se generaron las
secuencias gold
msec1=[1 1 -1 -1 -1 1 -1];
msec2=[1 1 -1 1 -1 -1 -1];
%llamada a la función xcorrelacion para calcular la
correlación cruzada.
[ind,corr]=xcorrelacion(msec1,msec2)
%grafica la función de correlación cruzada
plot(ind, corr)
grid on
```



**Figura 2.56** Correlación Cruzada entre un par de m-secuencias preferidas con  $n=3$ .

En la gráfica se ve que el valor máximo absoluto de la ***correlación cruzada*** entre las m-secuencias tiene un valor de  **$R=5$** . La correlación cruzada tiene 3 valores (3,-1 y -5). ***Por lo tanto este par de secuencias son preferidas.***

### Segunda Forma:

Se consideran el siguiente polinomio generador:  $x^3+x+1$  (cuyo grado no es divisible para 4.)

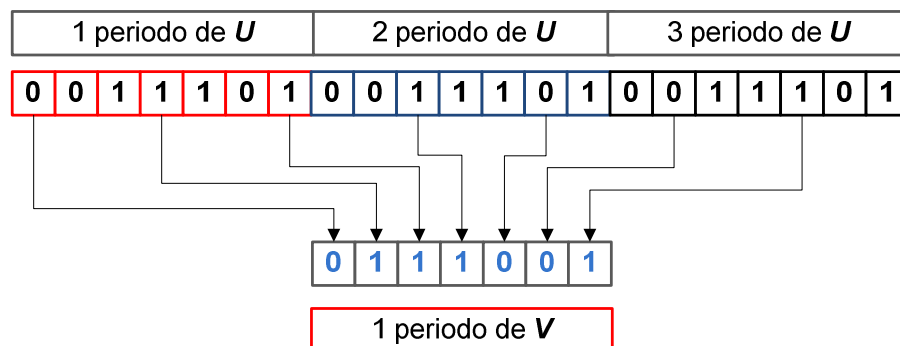
Con una semilla = 001 se obtienen la siguiente secuencia: msec1=[0 0 1 1 1 0 1] a la cual se la conocerá como la secuencia  $u$ . El par designado puede ser seleccionado como la  $m$ -secuencia  $u$  y su versión decimada  $v=u[q]$ . Donde se debe cumplir:

- $q$  es impar.
- $q = 2^k+1$  o  $q = 2^{2k} - 2^k + 1$ .
- Para generar secuencias preferidas  $k$  es típicamente toma los valores de 1 y 2. Estos valores de  $k$  hacen que  $q=3, 5, 13$ .
- **$v$  es obtenida por el muestreo de cada  $q^{\text{th}}$  símbolo de  $u$ .**
- El máximo común divisor de  $n$  y  $k$  debe tomar los siguientes valores:

$$\gcd(n, k) = \begin{cases} 1, & n \equiv 1 \pmod{2}; \quad n \text{ es impar} \\ 2, & n \equiv 2 \pmod{4}; \quad n \text{ es par y no es divisible para 4} \end{cases} \quad (\text{Ec 2.20})$$

**Primer Caso:** Seleccionando  $k=1$ , se tiene que  $q=3$ . El  $\gcd(3,1)=1$ . Por lo tanto se cumple con las condiciones para poder generar la secuencia  $v$ .

$v=u(3)$ . Es decir que la secuencia  $v$  se forma de muestrear cada 3<sup>er</sup> símbolo de  $u$ , tal como se muestra a continuación:



**Figura 2.57** Obtención de una secuencia  $v$  mediante la decimación de la secuencia  $u$  para  $k=1$  y  $q=3$ .

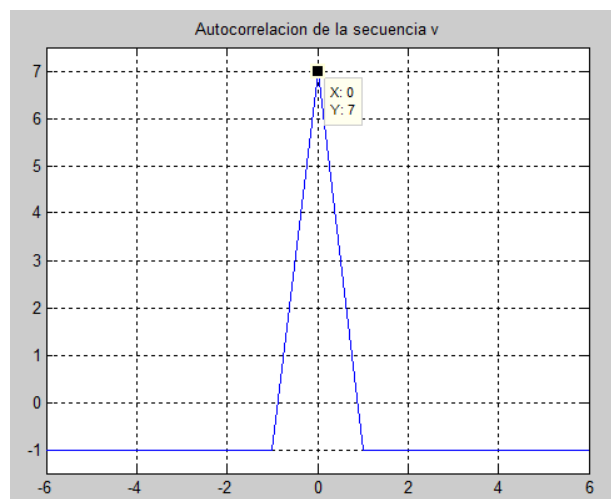
Se puede observar que para generar una secuencia con la misma cantidad de elementos, tomando  $q$  muestras de otra, se deben generar  $q$  periodos. El valor de  $q$  es muy importante, pues no todos los valores de  $q$  permiten obtener  $m$ -secuencias.

Para verificar si  $v$  es un  $m$ -secuencia, se puede realizar la Autocorrelación de esta secuencia y observar si presenta las características de un  $m$ -secuencia.

$$v=u[3]=[0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1] \Rightarrow [1 \ -1 \ -1 \ -1 \ 1 \ 1 \ -1]$$

En Matlab se tiene:

```
v=[1 -1 -1 -1 1 1 -1];
[ind corr]=xcorrelacion(v,v);
plot(ind, corr)
title('Autocorrelación de la secuencia v')
```

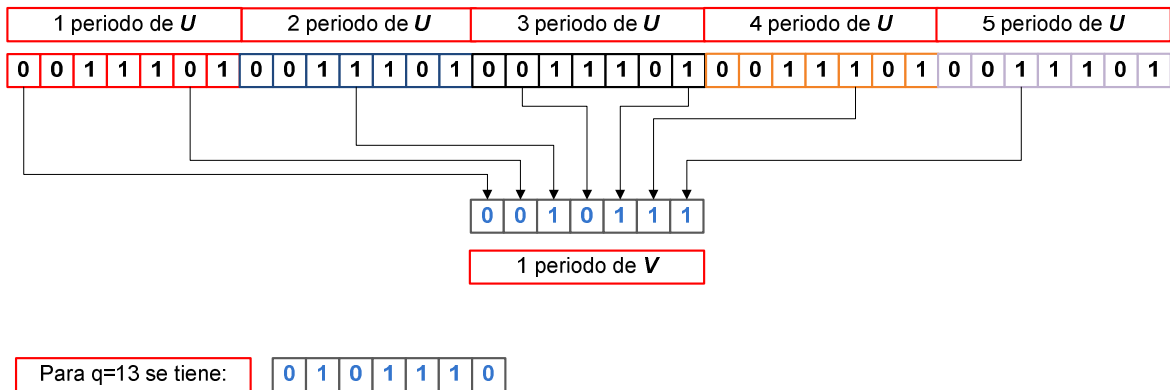


**Figura 2.58** Autocorrelación de la secuencia  $v$ .

La Autocorrelación presenta el pico máximo cuando está en fase, para las demás fases es constante e igual a -1. Por lo tanto, para  $q=3$  se ha generado una  $m$ -secuencia. De esta manera, se han cumplido con todas las condiciones para formar un **par de secuencias preferidas**. Se puede concluir entonces, que las secuencias  $u$  y  $v$  **son secuencias preferidas**.

**Segundo Caso:** Para  $k=2$  se tienen dos valores de  $q$ , siendo  $q_1=5$  y  $q_2=13$ . Por lo tanto el  $\gcd(3,2)=1$ . Entonces es posible encontrar un par preferido mediante la

decimación de la secuencia  $u$  para  $q_1=5$  y  $q_2=13$ . Por lo tanto,  $v=u(5)$ ,  $v=u(13)$  respectivamente. Es decir que la secuencia  $v$  se forma de muestrear cada 5<sup>to</sup> y 13<sup>avo</sup> símbolo de  $u$ , tal como se muestra a continuación:



**Figura 2.59** Obtención de una secuencia  $v$  mediante la decimación de la secuencia  $u$  para  $k=2$ ,  $q_1=5$  y  $q_2=13$ .

Como se puede ver en la figura se obtienen las siguientes secuencias:

$$V_1 = u[5] = [0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1] \Rightarrow [1 \ 1 \ -1 \ 1 \ -1 \ -1 \ -1]$$

$$V_2 = u[13] = [0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0] \Rightarrow [1 \ -1 \ 1 \ -1 \ -1 \ -1 \ 1]$$

**¿Estas secuencias son distintas entre sí?**

Para responder esta interrogante, debemos realizar la correlación cruzada de estas dos secuencias. Resolviendo en Matlab, se tiene:

```
v1=[1 1 -1 1 -1 -1 -1];
v2=[1 -1 1 -1 -1 -1 1];
[ind corr]=xcorrelacion(v1,v2);
plot(ind, corr)
title('Correlación Cruzada entre v1 y secuencia v2')
```



**Figura 2.60** Correlación Cruzada entre las secuencias v1 y v2.

Como se puede ver en la figura, existe un pico máximo de correlación cruzada, cuando la secuencia v2 se desplazada una vez en referencia a la secuencia v1. Para los demás desplazamientos la correlación cruzada es -1. Dada esta característica de la Correlación Cruzada, se puede afirmar que en realidad v1 y v2 **son la misma m-secuencia con diferente fase**, en ese caso están desfasa por un bit v2 respecto a v1.

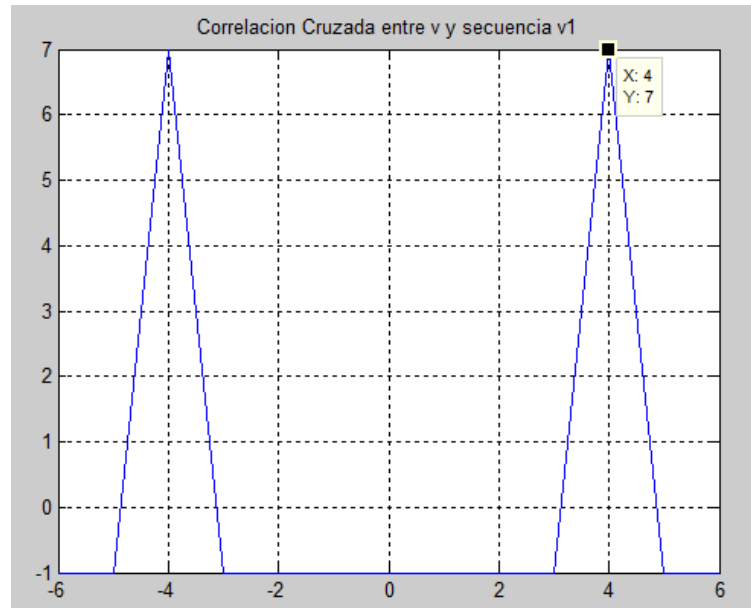
**¿Las secuencias obtenida para q1=5 y q=13 son diferentes a la secuencia obtenida para q=3?**

Como se vio en la pregunta anterior, para q1=5 y q=13, sólo se generó una m-secuencia a partir de la decimación. Ahora, Resolviendo la correlación cruzada de **v=u[3]** y **v1=[5]** se tiene:

$$v=[0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1] \Rightarrow [1 \ -1 \ -1 \ -1 \ 1 \ 1 \ -1]$$

$$v1=u[5]=[0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1] \Rightarrow [1 \ 1 \ -1 \ 1 \ -1 \ -1 \ -1]$$

```
v=[1 -1 -1 -1 1 1 -1];
v1=[1 1 -1 1 -1 -1 -1];
[ind corr]=xcorrelacion(v,v1);
plot(ind, corr)
title('Correlación Cruzada entre v y secuencia v1')
```



**Figura 2.61** Correlación cruzada entre las secuencias  $v$  y  $v1$ .

Nuevamente se presentan las características de Correlación cruzada para un  $m$ -secuencia desplazada. Por lo tanto, las secuencias generadas ***son la misma  $m$ -secuencia con diferente fase una de otra.***

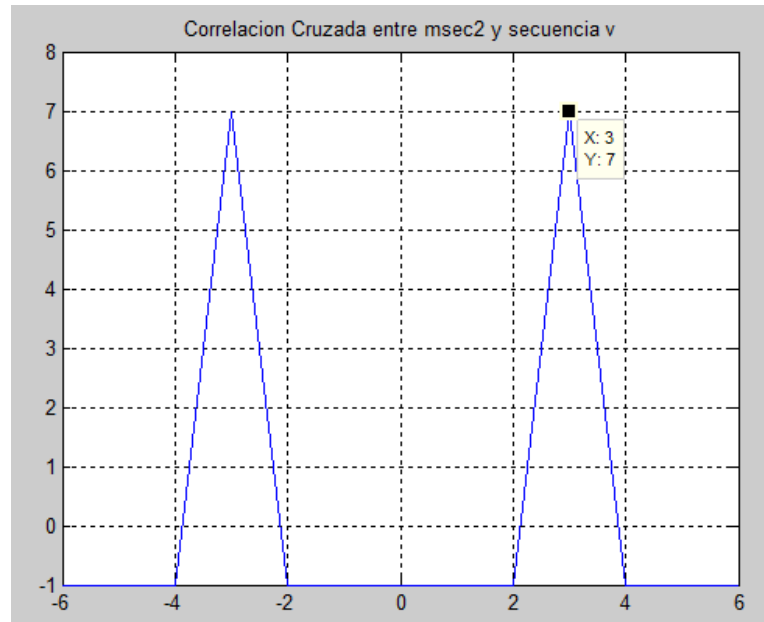
De la misma manera, si comparamos con la secuencia generada en el apartado anterior por el polinomio  $x^3+x^2+1$  y la semilla=001, con la secuencia  $v=u[3]$  generada a partir del proceso de decimación, se encuentra lo siguiente:

$$\text{msec2}=[0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1] \Rightarrow [1 \ 1 \ -1 \ 1 \ -1 \ -1 \ -1]$$

$$v=[0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1] \Rightarrow [1 \ -1 \ -1 \ -1 \ 1 \ 1 \ -1]$$

```
msec2=[1 1 -1 1 -1 -1 -1];
v=[1 -1 -1 -1 1 1 -1];
[ind corr]=xcorrelacion(msec2,v);
plot(ind, corr)
grid on
title('Correlación Cruzada entre msec2 y secuencia v')
```





**Figura 2.62** Correlación Cruzada entre msec2 y la secuencia v.

La figura muestra que **v es una versión desplazada de msec2.**

### Conclusión:

**Solo existe un par de secuencias preferidas para  $n=3$ ,** las cuales son generadas a partir de los polinomios  $x^3+x+1$  y  $x^3+x^2+1$ , o mediante la decimación de cualquier m-secuencia generada por cualquier polinomio generador para  $n=3$ .

La siguiente tabla muestra algunos pares de polinomios que generan secuencias preferidas.

**Tabla 2.20** Polinomios Preferidos.

n	$N=2^n-1$	Polinomio Preferido 1	Polinomio Preferido 2
5	31	$x^5+x^2+1$	$x^5+x^4+x^3+x^2+1$
6	63	$x^6+x+1$	$x^6+x^5+x^2+x+1$
7	127	$x^7+x^3+1$	$x^7+x^3+x^2+x+1$
9	511	$x^9+x^4+1$	$x^9+x^6+x^4+x^3+1$
10	1023	$x^{10}+x^3+1$	$x^{10}+x^8+x^3+x^2+1$
11	2047	$x^{11}+x^2+1$	$x^{11}+x^8+x^5+x^2+1$

Como se puede apreciar, si bien es posible seleccionar un pequeño conjunto de m-secuencias que tengan una función de **correlación cruzada baja**, este número

de secuencias es demasiado pequeño para aplicaciones prácticas de comunicaciones *multiusuario*. Consecuentemente, **las m-secuencias (incluidas las secuencias preferidas), no son apropiadas para poder distinguir muchos usuarios dentro de un sistema de comunicación CDMA.**

Hay que recalcar la importancia que tienen las **secuencias preferidas**, pues éstas, **son las precursoras de nuevos tipos de secuencias, cuya autocorrelación es alta y su correlación cruzada es baja**, como se verá a continuación.

#### 2.1.4 SECUENCIAS GOLD <sup>54 55</sup>

Las secuencias de Gold son una familia de secuencias que tienen buenas características de **autocorrelación** y de **correlación cruzada**. Una Secuencia Gold se genera mediante la suma módulo-2 de dos M-Secuencias de igual longitud que son parte de un conjunto especial de secuencias conocidas como **secuencias preferidas**.

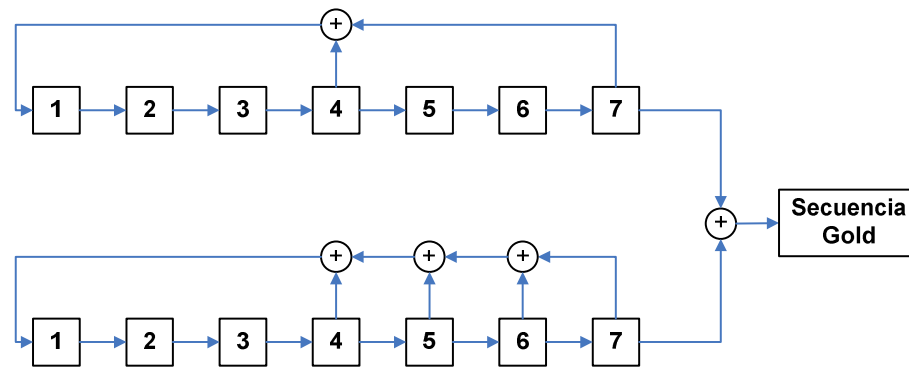
La familia completa de las secuencias Gold se forma realizando la suma módulo 2 (**XOR**) de una m-secuencia sin desplazamientos y las respectivas versiones desplazadas de la otra m-secuencia. De esta manera se generan  $2^n+1$  secuencias diferentes cada una con periodo igual a  $2^n-1$ , cuya correlación cruzada es baja.

**Un ejemplo** de la combinación de **dos secuencias preferidas** para generar una secuencia Gold es el siguiente:

---

<sup>54</sup> The Next Generation CDMA Technologies, Chen, Hsiao-Hwa, p. 144-145.

<sup>55</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 218-220.



**Figura 2.63** Generador de Secuencias Gold

Los polinomios asociados a las secuencias preferidas son  $x^7+x^4+1$  y  $x^7+x^6+x^5+x^4+1$ .

Un generador de Gold, tiene algunas características interesantes. A continuación se describen algunas de ellas:

- A diferencia del **generador de *m*-secuencias**, que sólo es capaz de generar una **única secuencia**, el **generador Gold** es capaz de producir distintos códigos, dependiendo de las **condiciones iniciales**.
- Todas las secuencias que se generan con la estructura anterior, se denominarán **familia de códigos de Gold**.
- Todas las secuencias generadas a partir de un mismo generador de Gold poseen el mismo período y tienen idénticas propiedades de Autocorrelación y Correlación Cruzada.
- La multiplicación de cualesquiera dos secuencias {a} y {b} de una determinada familia de códigos Gold, resulta en otra secuencia perteneciente a la misma familia.

### **Análisis de Autocorrelación y Correlación Cruzada de las Secuencias Gold**

Debido a que las secuencias Gold **se generan a partir de un par de secuencias preferidas**, heredan las características de la función de **correlación cruzada**, obteniendo los tres valores definidos por  $[-1, -t(n), t(n)-2]$ , donde:

$$t(n) = 1 + 2^{\frac{n+1}{2}} \text{ para } n \text{ impar} \quad (\text{Ec 2.21})$$

$$t(n) = 1 + 2^{\frac{n+2}{2}} \text{ para } n = 2 \bmod 4$$

La función de **autocorrelación** toma **cuatro valores** definidos por **un valor pico igual a la longitud de la secuencia Gold ( $2^n-1$ )** y los **tres valores de correlación cruzada** definidos anteriormente, es decir,  $[N, -1, -t(n), t(n)-2]$ , siendo  $N=2^n-1$ .

Por lo tanto, se puede notar que la **correlación cruzada** y la **autocorrelación (excepto cuando  $k=0$ )** entre dos secuencias cualesquiera  $\{a\}$  y  $\{b\}$  de la familia de Gold cumple con la siguiente desigualdad:

$$R \leq \begin{cases} \left\lceil 2^{\frac{n+1}{2}} + 1 \right\rceil, & \text{si } n \text{ es impar} \\ \left\lceil 2^{\frac{n+2}{2}} + 1 \right\rceil, & \text{si } n \text{ es par} \end{cases} \quad (\text{Ec 2.22})$$

En la siguiente tabla se hace una comparación en cuanto a valores de correlación cruzada entre m-secuencias y secuencias de Gold.

**Tabla 2.21** Comparación de Correlación entre las M-secuencias y las Secuencias de Gold

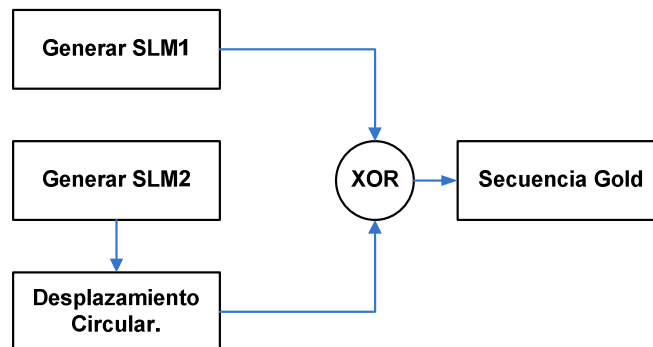
n	N = $2^n-1$	m-secuencias			Secuencias de Gold	
		Número	Rmax	Rmax/R(0)	Rmax	Rmax/R(0)
3	7	2	5	0,71	5	0,71
5	31	6	11	0,35	9	0,29
6	63	6	23	0,36	17	0,27
7	127	18	41	0,32	17	0,13
9	511	48	113	0,22	33	0,06
10	1023	60	383	0,37	65	0,06
11	2047	176	287	0,14	65	0,03

Se nota claramente la disminución de la correlación cruzada en las secuencias Gold.

### **Procedimiento para obtener Secuencias de Gold**

Se generan a partir de la XOR de una Secuencia de Longitud Máxima (SLM1) de periodo N con las todas las versiones desplazadas (**Corrimiento Circular**) de otra

Secuencia de Longitud Máxima (SLM2), La condición que deben cumplir las dos secuencias SLM, es que sean **secuencias preferidas**.

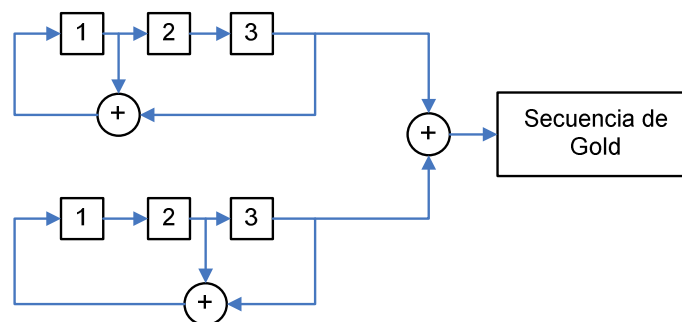


**Figura 2.64** Diagrama de un generador de secuencias Gold.

A continuación se muestra un ejemplo de la configuración con la cual, a partir de dos polinomios generadores de un par de **secuencias preferidas** se puede generar una familia de secuencias de Gold.

### Ejemplo de un generador de secuencias de Gold:

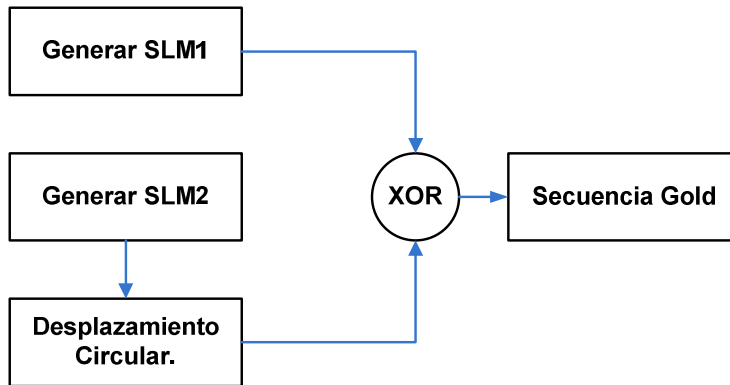
Un ejemplo de la combinación de dos **secuencias preferidas** para generar una secuencia Gold es el siguiente:



**Figura 2.65** Generadores LFSRs de **secuencias preferidas** para obtener la secuencia de Gold.

Los polinomios que definen al generador son:  $x^3+x+1$  y  $x^3+x^2+1$ . Además se puede notar que  $n=3$ . Por lo tanto generarán secuencias de periodo  $N=2^3-1=7$ .

Para generar las secuencias la familia completa de secuencias Gold se toma en cuenta el siguiente esquema:



**Figura 2.66** Diagrama de un generador de secuencias Gold.

Hay que notar que la secuencia de longitud máxima 1 (**SLM1**) o **m-secuencia 1 permanece constante** y que en la **secuencia de longitud máxima 2 (SLM2) se realiza un desplazamiento circular** para la generación de la familia de secuencias Gold del ejemplo, tal como se muestra en la siguiente tabla. Las secuencias SLM1 y SLM2 se obtienen con una semilla de 001.

**Tabla 2.22** Generación de la Familia de Secuencias Gold.

Secuencia SLM1	Secuencia SLM2	Secuencia Gold
0 0 1 1 1 0 1	0 0 1 0 1 1 1	0 0 0 1 0 1 0
0 0 1 1 1 0 1	1 0 0 1 0 1 1	1 0 1 0 1 1 0
0 0 1 1 1 0 1	1 1 0 0 1 0 1	1 1 1 1 0 0 0
0 0 1 1 1 0 1	1 1 1 0 0 1 0	1 1 0 1 1 1 1
0 0 1 1 1 0 1	0 1 1 1 0 0 1	0 1 0 0 1 0 0
0 0 1 1 1 0 1	1 0 1 1 1 0 0	1 0 0 0 0 0 1
0 0 1 1 1 0 1	0 1 0 1 1 1 0	0 1 1 0 0 1 1

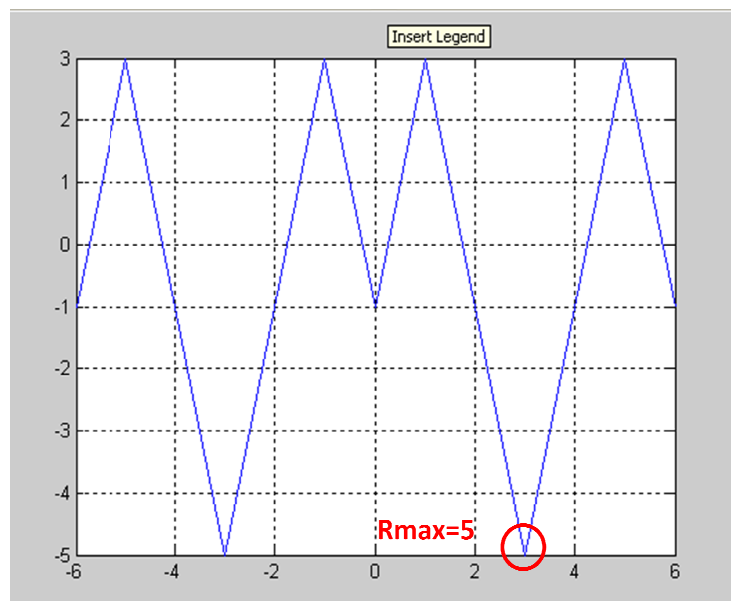
Realizando este procedimiento se llegan a tener  $2^n-1$  secuencias de Gold; pero una familia tiene  $2^n+1$  secuencias. Las dos secuencias que faltan para completar la familia son las secuencias SLM1 y SLM2. Por lo tanto se han generado  $2^n+1=2^3+1=9$  secuencias.

Verificar la **correlación cruzada** entre los dos primeros códigos de Gold.

Por ejemplo [0 0 0 1 0 1 0] y [1 0 1 0 1 1 0]. Para esto se debe pasar las secuencias binarias a bipolares ( $\pm 1$ ) usando una lógica negativa. Entonces se tiene las siguientes secuencias: [1 1 1 -1 1 -1 1] y [-1 1 -1 1 -1 -1 1]

En Matlab, mediante la función ya implementada ***xcorrelacion***, se puede realizar la ***correlación cruzada*** entre estas secuencias Gold con el siguiente programa:

```
%Se definen las secuencias Gold
gold1= [1 1 1 -1 1 -1 1];
gold2= [-1 1 -1 1 -1 -1 1];
%llamada a la función xcorrelacion para calcular la correlación
cruzada.
[ind,corr]=xcorrelacion(gold1,gold2)
%grafica la función de correlación cruzada
plot(ind, corr)
grid on
```

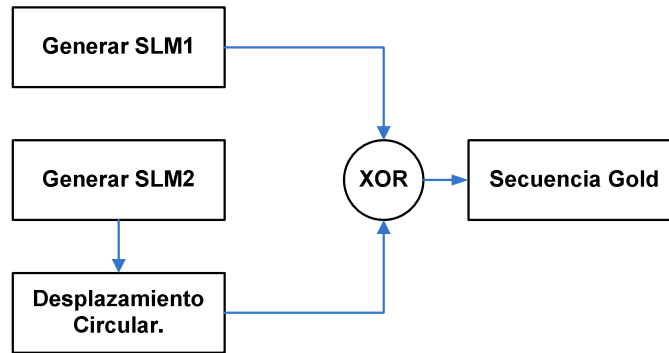


**Figura 2.67** Correlación cruzada entre las secuencias Gold [1 1 1 -1 1 -1 1] y [-1 1 -1 1 -1 -1 1]

Se puede ver que el valor máximo absoluto de la función de correlación cruzada es  $R=5$  y que existen solo **tres** valores de correlación cruzada, los cuales son [3, -1, -5]. Estos valores se producen a manera de herencia de la correlación cruzada de las secuencias preferidas.

### **Generación de Secuencias Gold en Matlab**

Como se vio en el ejemplo anterior, para generar  $2^n+1$  secuencias de Gold a partir de las secuencias generadas por los dos polinomios se hace un corrimiento circular a una de las dos secuencias tal y como se muestra en la figura:



**Figura 2.68** Esquema de un Generador de Secuencias Gold.

Utilizando el esquema anterior se generan  $2^n - 1$  secuencias de Gold, las dos que faltan para completar las  $2^n + 1$  que se pueden generar son SLM1 y SLM2. Para la generación de secuencias Gold se realizó el programa *cap2\_cod7\_secuencias\_gold.m* (ver *Anexos Cap 2*). La familia de secuencias Gold se almacena en la variable `gold`. El script también muestra la autocorrelación de dos secuencias Gold de la familia generada, así como de su correlación cruzada. Para obtener las gráficas de la Autocorrelación y correlación cruzada, se deben escoger previamente el par de secuencias gold que se desee analizar.

### Resultados:

Los polinomios que definen al generador son:  $x^3 + x + 1$  y  $x^3 + x^2 + 1$  y se obtienen 9 secuencias Gold. Las dos primeras secuencias de la familia de secuencias Gold corresponden a las secuencias de longitud máxima (SLM1 y SLM2). Las 7 restantes secuencias se generan a partir de SLM1 y los desplazamientos de SLM2.

```

Secuencias Gold generadas:

gold =

     1     1    -1    -1    -1     1    -1
     1     1    -1     1    -1    -1    -1
     1     1     1    -1     1    -1     1
    -1     1    -1     1    -1    -1     1
    -1    -1    -1    -1     1     1     1
    -1    -1     1    -1    -1    -1    -1
     1    -1     1     1    -1     1     1
    -1     1     1     1     1     1    -1
     1    -1    -1     1     1    -1    -1
  
```

**Figura 2.69** Familia de secuencias Gold generadas en Matlab.



En el programa se pueden escoger dos secuencias de entre la familia de secuencias Gold para realizar el cálculo de la Autocorrelación y Correlación cruzada respectivamente. Escogiendo las secuencias 3 y 4, se obtiene:

```

Indice Secuencia Gold1: 3
Indice Secuencia Gold2: 4

gold1 =

    1    1    1   -1    1   -1    1

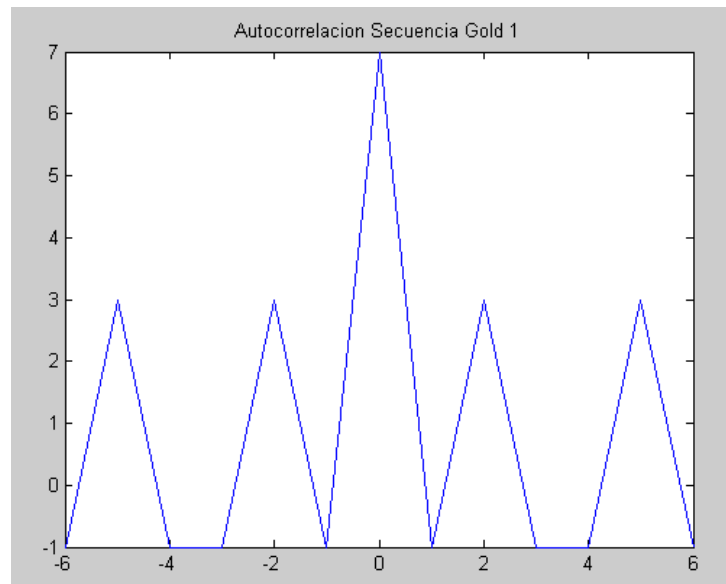
gold2 =

   -1    1   -1    1   -1   -1    1

```

**Figura 2.70** Secuencias Gold escogidas para verificar la Correlación Cruzada y Autocorrelación.

Secuencia Gold 3 = [1 1 1 -1 1 -1 1].



**Figura 2.71** Autocorrelación de la secuencia Gold 3 de la familia de secuencias Gold generadas

Secuencia Gold 4 = [-1 1 -1 1 -1 -1 1]

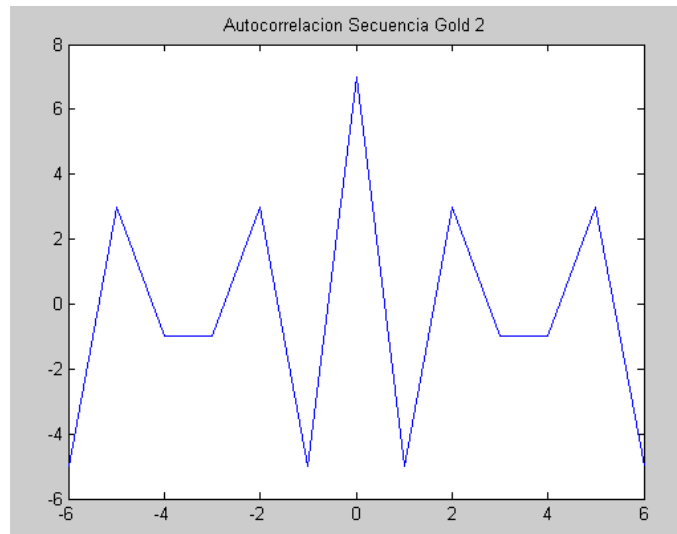


Figura 2.72 Autocorrelación de la secuencia Gold 4 de la familia de secuencias Gold generadas.

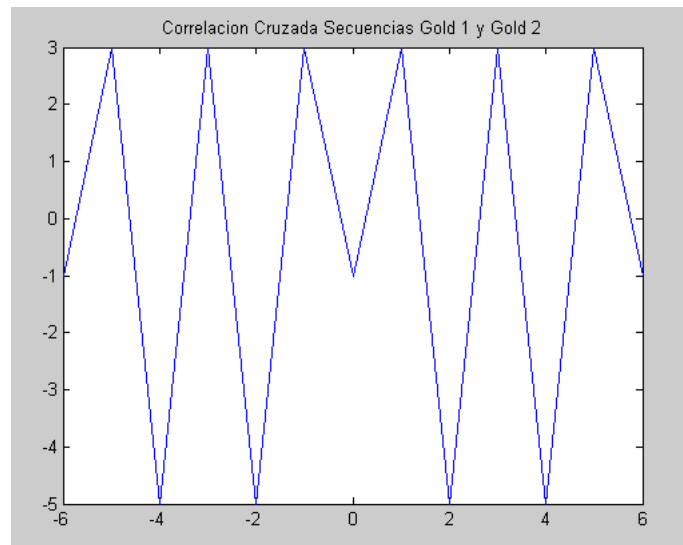


Figura 2.73 Correlación cruzada entre las secuencias Gold 3 y 4.

Se puede ver que la autocorrelación de una secuencia Gold tiene cuatro valores

**Ejemplo 2:** Los polinomios de grado  $n=7$ ,  $p_1(x) = x^7+x^4+1$  y  $p_2(x) = x^7+x^6+x^5+x^3+1$  cumplen con las condiciones dadas por el teorema de Gold. La generación de una secuencia de Gold a partir de la xor de las secuencias de longitud máxima generadas por cada uno de estos polinomios se ilustra en la siguiente figura:

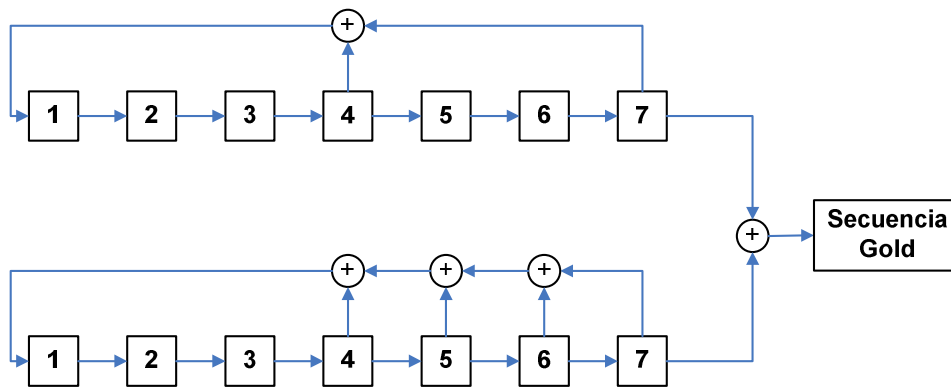


Figura 2.74 Generador de Secuencias Gold con  $p_1(x) = x^7+x^4+1$  y  $p_2(x) = x^7+x^6+x^5+x^3+1$

### Resolución en Matlab.

Ahora se definen los polinomios  $p_1(x) = x^7+x^4+1$  y  $p_2(x) = x^7+x^6+x^5+x^4+1$  en el programa *cap2\_cod7\_secuencias\_gold.m*.

```
pol1=[1 0 0 1 0 0 0 1]; %x7+0x6+0x5+x4+0x3+0x2+0x+1
pol2=[1 1 1 1 0 0 0 1]; %x7+x6+x5+x4+0x3+0x2+0x+1
```

### Resultados:

Para este ejemplo tomamos las secuencias Gold 3 y 4 de la familia de secuencias Gold. La familia de secuencias generadas está almacenada en la variable gold en forma de matriz. Para este ejemplo se genera una familia de  $2^7+1= 129$  secuencias Gold de longitud 127.

Name	Value	Class
g1	3	double
g2	127	double
gold	<129x127 double>	double
gold1	<1x127 double>	double
gold2	<1x127 double>	double

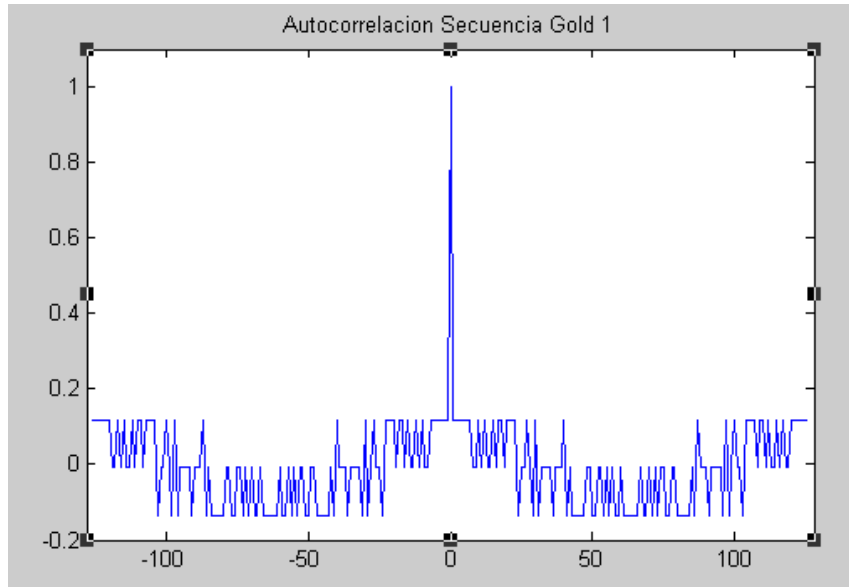
Figura 2.75 Número de Secuencias Gold generadas a partir de  $p_1(x) = x^7+x^4+1$  y  $p_2(x) = x^7+x^6+x^5+x^3+1$

Las secuencias escogidas se guardan en las variables gold1 y gold2 respectivamente. Esto se lo hace ingresando el número de la secuencia que se requiere para el análisis en el workspace, como se muestra a continuación:

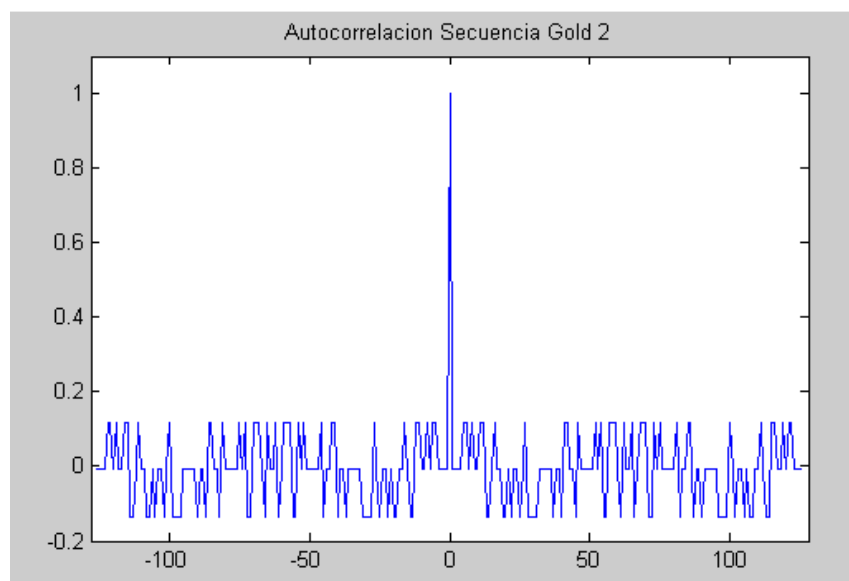
```
Indice Secuencia Gold1: 3
Indice Secuencia Gold2: 4
```

**Figura 2.76** Secuencias escogidas para el análisis de Autocorrelación y correlación cruzada.

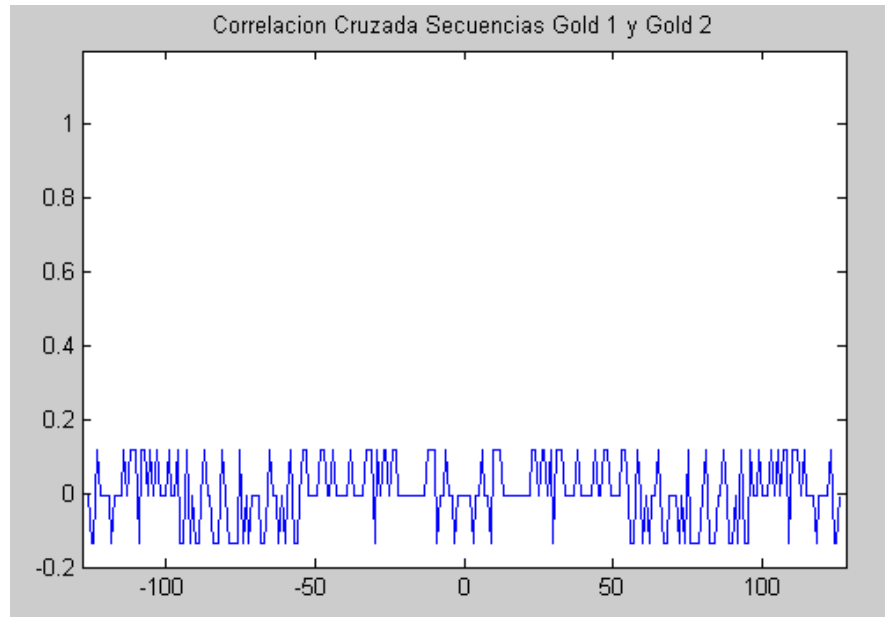
Las figuras siguientes muestran las secuencias Gold escogidas con su Autocorrelación y correlación cruzada normalizada.



**Figura 2.77** Autocorrelación normalizada de la secuencia Gold 3.



**Figura 2.78** Autocorrelación normalizada de la secuencia Gold 4.



**Figura 2.79** Correlación cruzada entre dos secuencias Gold con  $n=7$ .

### Conclusiones:

- Se puede notar que las secuencias Gold generadas a partir de un par de secuencias preferidas tienen cuatro valores de Autocorrelación y tres valores de correlación cruzada.
- Cuando las secuencias Gold son de mayor longitud, la correlación cruzada se aproxima a la correlación de cruzada del ruido. Mientras sea de mayor longitud, los tres valores de correlación cruzada tienen a ser cero.

### 2.1.5 SECUENCIAS KASAMI <sup>56 57</sup>

Como se vio en el caso de las secuencias de Gold, su objetivo fue de obtener secuencias con mejores propiedades de autocorrelación y correlación cruzada. Algo similar hizo Kasami. Este tipo de secuencias son muy importantes ya que

---

<sup>56</sup> The Next Generation CDMA Technologies, Chen, Hsiao-Hwa, p. 145-146.

<sup>57</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 221-222.

presentan valores de **correlación cruzada muy bajos, siendo más bajos que los picos en las secuencias Gold.**

Hay dos Sets de secuencias Kasami: **Set Pequeño y el Set Grande.**

### 2.1.5.1 Set pequeño de Secuencias Kasami

El set pequeño de Secuencias Kasami se forma a partir de una m-secuencia **u**, **generada solamente por un polinomio de grado par.**

El proceso para generar este Set corto de secuencias Kasami es el siguiente:

Se parte de una m-secuencia **u**, con periodo  **$N=2^n-1$** , donde **n** es únicamente par. Esta es decimada cada  **$q=2^{n/2}+1$**  elementos para formar una **nueva secuencia binaria w cuyo periodo es  $2^{n/2}-1$ .**

**La suma modulo 2 de u y algún cambio cíclico de w desde 0 a  $2^{n/2}-2$  genera una secuencia Kasami con periodo  $2^n-1$ .**

**El Set pequeño de secuencias Kasami** se define mediante las siguientes fórmulas, en las cuales **T** denota el operador de los desplazamientos a la izquierda de la secuencia, **m** es el parámetro de desplazamiento para **w** y  $\oplus$  denota la adición modulo 2.

$$K_s = \begin{cases} u & m = -1 \\ u \oplus T^m w & m = 0, 1, \dots, 2^{n/2} - 2 \end{cases} \quad (\text{Ec 2.23})$$

Entonces el conjunto de **Secuencias Kasami pequeña** contiene  $2^{n/2}$  secuencias.

La **correlación cruzada** entre dos secuencias del **set corto de secuencias Kasami** puede tomar los siguientes valores:

$$-s(n), -1, s(n)-2; \text{ siendo } s(n) = 2^{n/2} + 1 \quad (\text{Ec 2.24})$$

Mientras que la **Autocorrelación** toma cuatro valores: los tres de la Correlación cruzada y un pico de tamaño N cuando la secuencia está en fase ( $k=0$ ).

$$-s(n), -1 \text{ o } [s(n)-2] N, \text{ siendo } N=2^n-1 \quad (\text{Ec 2.25})$$

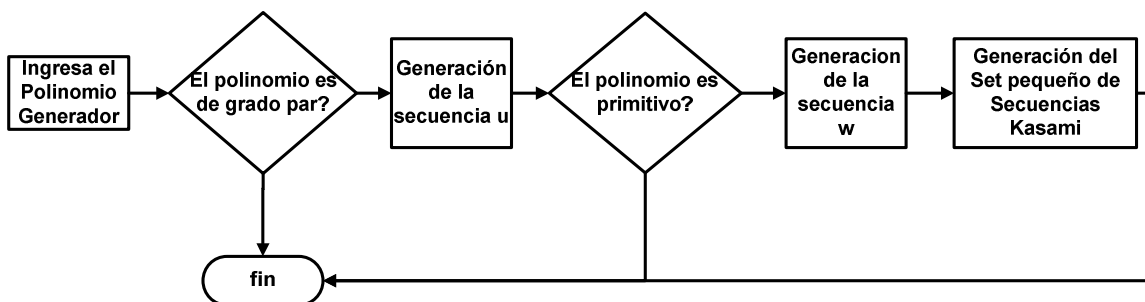
A continuación se listan algunos polinomios generadores para un Set de Secuencias Kasami Pequeño:

**Tabla 2.23** Polinomios generadores del set corto de secuencias Kasami.

<b>n</b>	<b>N</b>	<b>Polinomio</b>
4	15	$x^4+x+1$
6	63	$x^6+x+1$
8	255	$x^8+x^4+x^3+x^2+1$
10	1023	$x^{10}+x^3+1$
12	4095	$x^{12}+x^6+x^4+x+1$

### Generación del set pequeño de Secuencias Kasami en Matlab

La generación del set corto de secuencias Kasami se realiza con el programa *cap2\_cod8\_kasami\_pequena.m* (ver Anexos Cap 2), el cual se basa en el siguiente diagrama:



**Figura 2.80** Diagrama para generar el Set Pequeño de Secuencias Kasami.

Las secuencias Kasami se formarán de la XOR entre la  $m$ -secuencia  $u$  y una versión desplazada de la secuencia  $w$ .

## Resultados:

Como polinomio generador se ingresará el que se muestra en la siguiente tabla.

**Tabla 2.24** Polinomio generador de grado  $n=4$  para producir un Set Pequeño de Secuencias Kasami.

<b>n</b>	<b>N</b>	<b>Polinomio</b>	<b>Set</b>
4	15	$x^4+x+1$	Pequeño

El vector a ingresar será: [1 0 0 1 1].

```
Ingrese el Polinomio Generador: [1 0 0 1 1]
>> u
u =
    1    0    0    0    1    1    1    1    0    1    0    1    1    0    0
>> w
w =
    1    1    0    1    1    0    1    1    0    1    1    0    1    1    0
>> kasami
kasami =
   -1    1    1    1   -1   -1   -1   -1    1   -1    1   -1   -1    1    1
    1   -1    1   -1    1   -1    1    1    1    1   -1   -1    1   -1    1
    1    1   -1   -1   -1    1    1   -1   -1    1    1    1    1    1   -1
   -1   -1   -1    1    1    1   -1    1   -1   -1   -1    1   -1   -1   -1
```

**Figura 2.81** Set corto de Secuencias Kasami a partir del polinomio  $p(x)=x^4+x+1$ .

Se puede observar que  $u$  es una  $m$ -secuencia con  $n=4$  y  $N=15$ .  $w$  está formada muestreando cada quinto elemento de  $u$ , comenzando desde el primer elemento,  $w=[1\ 1\ 0]$ . Por lo tanto se genera una familia de  $2^{n/2}=4$  secuencias Kasami.

## Verificación de la Correlación Cruzada y Autocorrelación

Para  $n=4$ :

$$s(n)=2^{4/2}+1=5 \quad (\text{Ec 2.26})$$

Por lo tanto los valores de correlación cruzada serán:



$$-s(n), -1 \text{ o } [s(n)-2] = -5, -1, 3$$

(Ec 2.27)

```

>> sec1=kasami(2,:)

sec1 =

     1     -1     1     -1     1     -1     1     1     1     1     -1     -1     1     -1     1

>> sec2=kasami(3,:)

sec2 =

     1     1     -1     -1     -1     1     1     -1     -1     1     1     1     1     1     -1

>> [ind, corr]=xcorrelacion(sec1,sec2);
>> corr

corr =

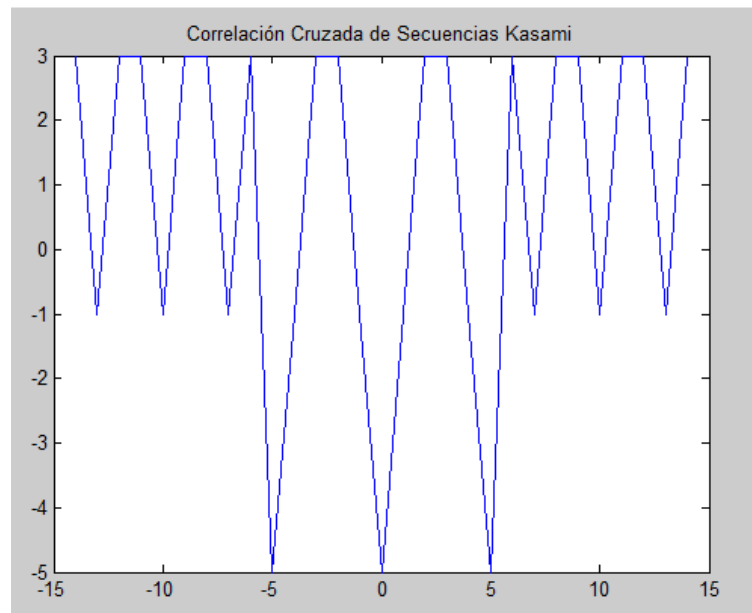
Columns 1 through 15
|
     3     -1     3     3     -1     3     3     -1     3     -5     -1     3     3     -1     -5

Columns 16 through 29
-1     3     3     -1     -5     3     -1     3     3     -1     3     3     -1     3

>> plot(ind,corr)

```

**Figura 2.82** Correlación Cruzada de dos secuencias Kasami en Matlab.



**Figura 2.83** Gráfica de la Correlación Cruzada de dos secuencias Kasami.

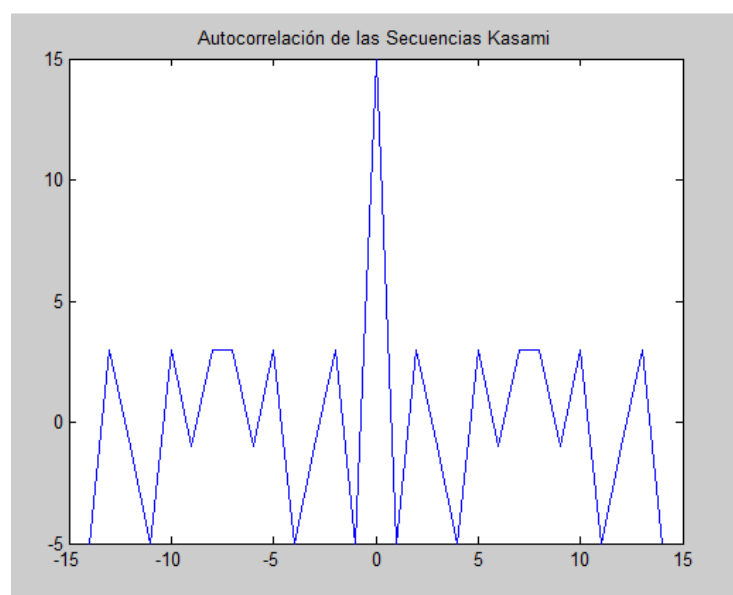
Como se puede ver la **correlación cruzada** entre dos secuencias Kasami cumplen con las definiciones presentadas anteriormente.

La **autocorrelación**, presentará **cuatro** valores, los tres definidos para la correlación cruzada y un pico del tamaño de la secuencia, cuando se encuentre en fase ( $k=0$ ). Por lo tanto los valores de la autocorrelación de una secuencia Kasami deben ser:  $[-s(n) -1 s(n)-2 N]$  con  $s(n)=2^{n/2}+1$ .

Para  $n=4$ , los valores de Autocorrelación deben ser:  $[-5 -1 3 15]$ .

```
>> sec1=kasami(2,:)
sec1 =
     1     -1     1     -1     1     -1     1     1     1     1     -1     -1     1     -1     1
>> [ind, corr]=xcorrelacion(sec1,sec1);
>> corr
corr =
Columns 1 through 15
    -5     3     -1     -5     3     -1     3     3     -1     3     -5     -1     3     -5     15
Columns 16 through 29
    -5     3     -1     -5     3     -1     3     3     -1     3     -5     -1     3     -5
>> plot(ind,corr)
```

**Figura 2.84** Autocorrelación de una secuencia Kasami.



**Figura 2.85** Gráfica de la Autocorrelación de una Secuencia Kasami.

Se comprueba que la **autocorrelación** tiene cuatro valores [-5 -1 3 15].

### 2.1.5.2 Set Grande de Secuencias Kasami.

Para generar un Set Grande de secuencias Kasami, igual que el caso anterior, se parte de una  $m$ -secuencia  $u$ , con periodo  $N=2^n-1$ . Esta es decimada cada  $q=2^{n/2}+1$  elementos para formar una **secuencia binaria  $w$**  cuyo periodo es  $2^{n/2}-1$ . De la misma manera se conforma una **nueva secuencia binaria  $v$** , decimando la secuencia  $u$  por un factor de  $q=2^{(n+2)/2}+1$ .

Resulta interesante la obtención de la secuencia  $v$ , ya que al decimar  $u$  por  $q=2^{(n+2)/2}+1$  se asegura que **la secuencia  $v$  sea primeramente un  $m$ -secuencia y además que sea el par preferido para la secuencia  $u$ . Así, las secuencias  $u$  y  $v$  forman un par de secuencias preferidas**. Por lo tanto, la XOR de  $u$  y algún desplazamiento de  $v$ , **formaran un subconjunto de secuencias Gold dentro del Set Grande de secuencias Kasami**.

Este tipo de secuencias se restringe para **polinomios primitivos generadores de grado par que no sean divisibles para 4**, pues los grados múltiplos de 4 solo pueden formar un set de secuencias Kasami pequeño. Además como se vio en el apartado de secuencias preferidas, los polinomios de grado 4 y sus múltiplos no tienen un par preferido.

El Set Largo de Secuencias Kasami se define mediante las siguientes fórmulas, en las cuales  **$T$  denota el operador de los desplazamientos a la izquierda** de la secuencia,  **$m$**  es el parámetro de desplazamiento para  $w$ ,  **$k$**  es el parámetro de desplazamiento de  $v$  y  $\oplus$  denota la adición modulo 2. Un **Set de secuencias Kasami Grande** de pueden ser definidas como sigue:

$$K_s = \begin{cases} u & k = -2, m = -1 \\ v & k = -1, m = -1 \\ u \oplus T^k v & k = 0, 1, \dots, 2^n - 2, m = -1 \\ u \oplus T^m w & k = -2, m = 0, 1, \dots, 2^{\frac{n}{2}} - 2 \\ v \oplus T^m w & k = -1, m = 0, 1, \dots, 2^{\frac{n}{2}} - 2 \\ u \oplus T^k v \oplus T^m w & k = 0, 1, \dots, 2^n - 2, m = 0, 1, \dots, 2^{\frac{n}{2}} - 2 \end{cases} \quad (\text{Ec 2.28})$$

Entonces, el Set de Secuencias Kasami grande tiene  $2^{n/2}(2^n+1)$  secuencias.

Los valores de la función de **correlación cruzada** de estas secuencias toman los valores:

$$\{-t(n), -s(n), -1, s(n)-2, t(n)-2\} \quad (\text{Ec 2.29})$$

Donde:

$$t(n)=2^{(n+2)/2}+1 \quad (\text{Ec 2.30})$$

$$s(n)=2^{n/2}+1 \quad (\text{Ec 2.31})$$

La **autocorrelación** de las **Secuencias Kasami grandes puede tener hasta seis** valores:

$$\{-t(n), -s(n), -1, s(n)-2, t(n)-2, N\} \quad (\text{Ec 2.32})$$

$$N=2^n-1$$

En la siguiente tabla se listan algunos polinomios generadores de secuencias Kasami.

**Tabla 2.25** Polinomios Generadores del Set Grande de Secuencias Kasami.

n	N	Polinomio
6	63	$x^6+x+1$
10	1023	$x^{10}+x^3+1$

Se puede notar que el grado de los polinomios generadores es par y no son múltiplos de 4.

Resulta interesante comparar las secuencias Kasami con las secuencias Gold. La comparación se la muestra en la siguiente tabla.

**Tabla 2.26** Comparación entre Secuencias Kasami y Gold.

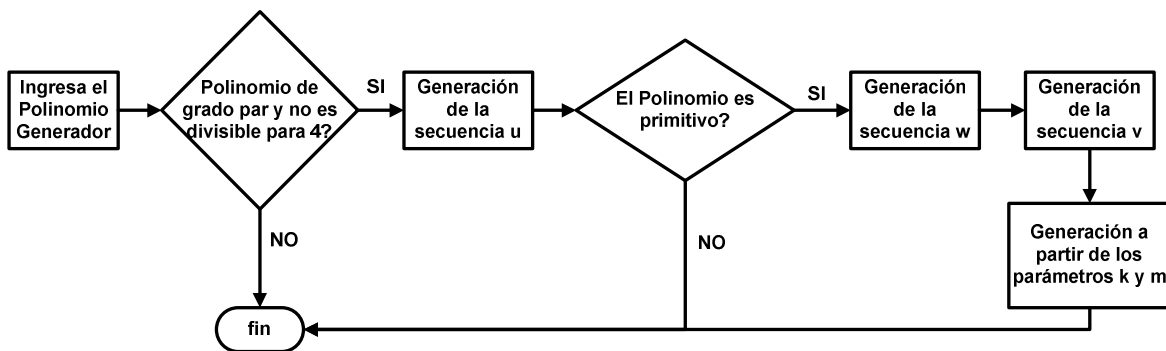
	<b>Set Corto de Secuencias Kasami.</b>	<b>Set grande se secuencias Kasami.</b>	<b>Secuencias Gold.</b>
<b>Periodo de la secuencia</b>	$2^n-1$	$2^n-1$	$2^n-1$
<b>Tamaño de la familia</b>	$2^{n/2}$	$2^{n/2}(2^n+1)$	$2^n-1$
<b>Valores de n</b>	Par	Par no múltiplo de 4	Impar o Par no múltiplo de 4
<b>Máximo valor de correlación entre cualquier par de secuencias</b>	$2^{n/2}+1$	$2^{(n+2)/2}+1$	$2^{(n+2)/2}+1$

Por ejemplo, para  $n=6$ , el tamaño de una secuencia Kasami es 63 bits, el tamaño de un set pequeño es 8 secuencias, el tamaño de un set grande es de 520 secuencias y el tamaño de la familia de secuencias Gold es 65 secuencias. Los valores de correlación cruzada son 9 para el set pequeño de secuencias Kasami, 17 para el set grande de secuencias Kasami y 17 para la familia de secuencias Gold.

Por lo tanto el conjunto grande de secuencias Kasami genera mayor cantidad de secuencias para asignar a los usuarios que la familia de secuencias Gold. Además que el máximo valor de correlación cruzada es el mismo que en las secuencias Gold. Mientras que el set de secuencias Kasami pequeño tiene mejores características de correlación cruzada que el Set de Secuencias Kasami grande y las secuencias Gold.

### Generación del Set Grande de Secuencias Kasami en Matlab

La generación del set grande de secuencias Kasami se realiza con el programa *cap2\_cod9\_kasami\_grande.m* (ver *Anexos Cap 2*), el cual se basa en el siguiente diagrama:



**Figura 2.86** Diagrama para generar el Set Largo de Secuencias Kasami.

Se debe ingresar un vector que contenga los coeficientes  $a_n$  del polinomio en forma descendente, es decir:  $P=[a_n a_{n-1} \dots a_1 a_0]$ .

Los valores de  $k$  y  $m$  deben tomar los valores entre los rangos:  $-2 \leq k \leq 2^n - 2$  y  $-1 \leq m \leq 2^{n/2} - 2$ . Si los valores que se ingresan en  $k$  y  $m$  corresponden al rango respectivo se realizará la operación que generará la secuencia Kasami.

**EJEMPLO:** Para el ejemplo se considera como polinomio generador a  $P(x)=x^6+x+1$ . Por lo tanto el vector a ingresar será:  $[1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1]$ . Para  $k=-2$  y  $m=-1$ , se genera la  $m$ -secuencia  $u$ .

```

Ingrese el Polinomio Generador: [1 0 0 0 0 1 1]
Ingrese el valor de k: -2
Ingrese el valor de m: -1
La Secuencia Kasami es:
Columns 1 through 15

-1    1    1    1    1    1   -1   -1   -1   -1   -1   -1    1   -1    1

Columns 16 through 30

-1    1   -1   -1    1    1   -1   -1    1   -1   -1   -1    1   -1   -1

Columns 31 through 45

 1   -1    1    1   -1    1    1   -1   -1   -1    1    1    1   -1    1

Columns 46 through 60

-1   -1   -1   -1    1    1   -1    1   -1    1    1    1   -1   -1    1

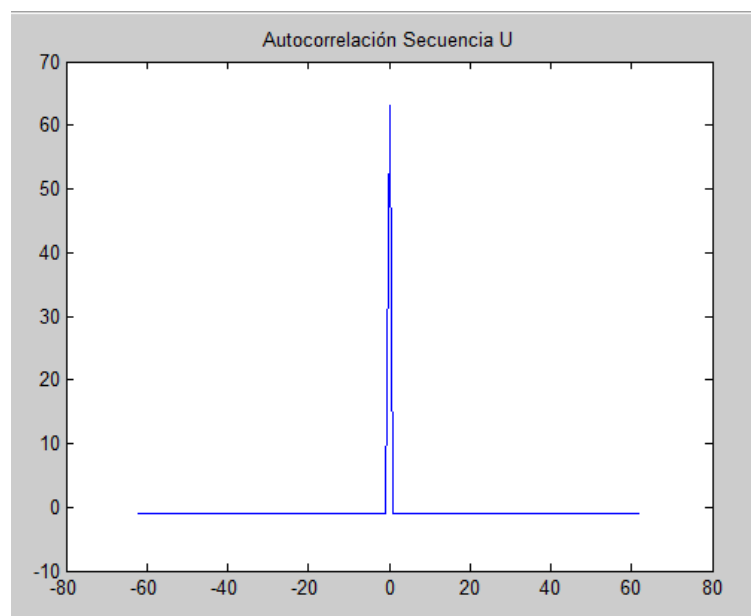
Columns 61 through 63

 1    1    1

```

**Figura 2.87** Generación de la secuencia Kasami para  $k=-2$  y  $m=-1$ .

Realizando la **autocorrelación** para la secuencia Kasami obtenida, se puede ver que efectivamente,  $u$  es una  $m$ -secuencia.



**Figura 2.88** Autocorrelación para la secuencia Kasami con  $k=-2$  y  $m=-1$ .

Ahora, se comprueba que la secuencia  $u$  y  $v$  son un par preferidos de secuencias.

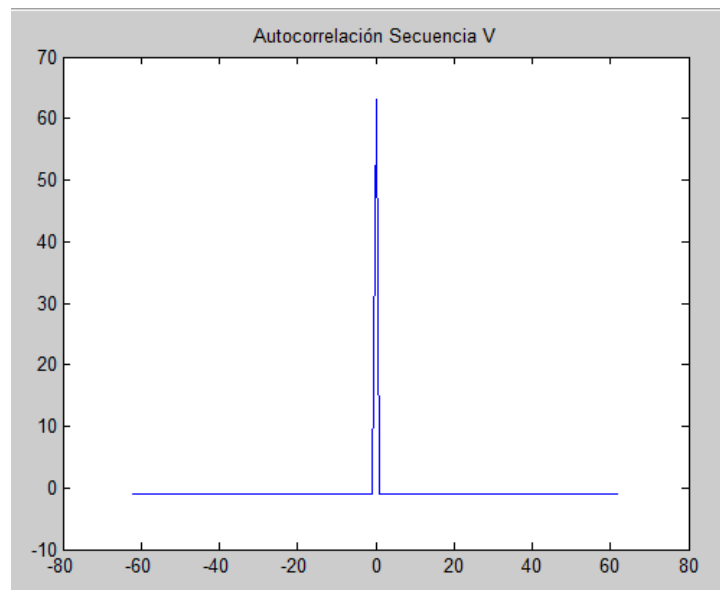
```

Ingrese el Polinomio Generador: [1 0 0 0 1 1]
Ingrese el valor de k: -1
Ingrese el valor de m: -1
La Secuencia Kasami es:
  Columns 1 through 15
    -1    -1    -1    -1     1    -1    -1     1    -1     1     1     1    -1     1     1
  Columns 16 through 30
     1     1    -1     1    -1    -1     1     1    -1     1    -1     1    -1     1     1
  Columns 31 through 45
    -1     1     1    -1    -1    -1    -1     1     1     1     1     1    -1    -1     1
  Columns 46 through 60
    -1    -1    -1     1     1    -1    -1     1     1     1    -1    -1    -1     1    -1
  Columns 61 through 63
     1    -1    -1

```

**Figura 2.89** Secuencia Kasami con  $k=-1$  y  $m=-1$ .

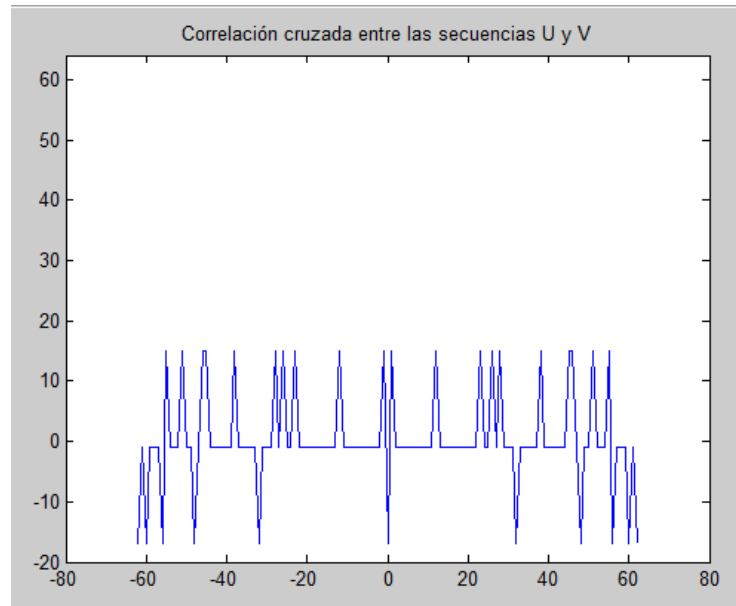
La gráfica de la **autocorrelación** para la secuencia  $v$  es la siguiente:



**Figura 2.90** Autocorrelación para la Secuencia Kasami con  $k=-1$  y  $m=-1$ .

Haciendo la correlación cruzada entre las dos secuencias se tiene:





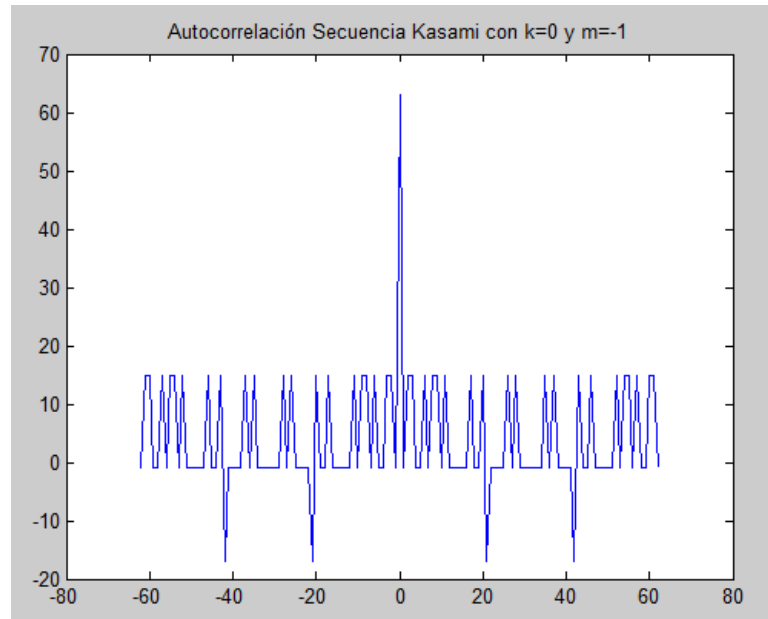
**Figura 2.91** Correlación Cruzada entre las secuencias U y V.

Se puede notar que la **correlación cruzada** entre las dos secuencias presentan **tres** valores: [-17 -1 15]. **Se comprueba que son un par de secuencias preferidas**, y por lo tanto, el Set Largo de Secuencias Kasami contendrá secuencias Gold generadas por el par de secuencias preferidas u y v. Por ejemplo para  $k=0 \dots 2^n-2$  y  $m=-1$ , se generan las secuencias gold. Si  $k=0$  y  $m=-1$  se tiene:

```

Ingrese el Polinomio Generador: [1 0 0 0 0 1 1]
Ingrese el valor de k: 0
Ingrese el valor de m: -1
La Secuencia Kasami es:
  Columns 1 through 15
    1  -1  -1  -1  1  -1  1  -1  1  -1  -1  -1  -1  -1  1
  Columns 16 through 30
   -1  1  1  -1  -1  -1  -1  -1  -1  -1  1  -1  -1  -1  -1
  Columns 31 through 45
   -1  -1  1  -1  1  -1  -1  -1  -1  -1  1  1  -1  1  1
  Columns 46 through 60
    1  1  1  -1  1  -1  1  1  -1  1  -1  -1  1  -1  -1
  Columns 61 through 63
    1  -1  -1
  
```

**Figura 2.92** Secuencia Kasami con  $k=0$  y  $m=-1$ .



**Figura 2.93** Autocorrelación de la Secuencia Kasami con  $k=0$  y  $m=-1$ .

Se puede ver que la **autocorrelación** de esta secuencia Kasami tiene cuatro valores  $[-17 -1 15 63]$ . Estos valores son los típicos de una secuencia Gold para  $n=6$ , con lo que se puede observar que dentro del Set Largo de secuencias Kasami, se tienen secuencias Gold. Consecuentemente, los valores de la **correlación cruzada** entre las secuencias generadas para  $k=0,1,\dots,2^n-2$  y  $m=-1$ , tomarán **tres** valores  $[-17 -1 15 63]$

En general, entre los subgrupos que conforman el Set Largo de secuencias Kasami (a excepción de entre las secuencias del subgrupo que conforman las secuencias Gold) los valores de la correlación cruzada, tomará cinco valores expuestos anteriormente:  $\{-t(n), -s(n), -1, s(n)-2, t(n)-2\}$ . Donde:  $t(n)=2^{(n+2)/2}+1$  y  $s(n)=2^{n/2}+1$

Para  $n=6$ , los valores que debe tomar la función de **autocorrelación y correlación cruzada** deben ser:

Autocorrelación:  $[-17 -9 -1 7 15 63]$ .

Correlación cruzada:  $[-17 -9 -1 7 15]$ .

```

Ingrese el Polinomio Generador: [1 0 0 0 0 1 1]
Ingrese el valor de k: 1
Ingrese el valor de m: 1
La Secuencia Kasami es:
Columns 1 through 15
-1    1    1   -1    1    1    1    1   -1    1   -1   -1    1   -1   -1
Columns 16 through 30
1   -1    1   -1    1   -1    1    1    1   -1    1    1   -1    1    1
Columns 31 through 45
1   -1    1    1    1    1    1    1   -1   -1   -1    1    1   -1   -1
Columns 46 through 60
1    1   -1   -1   -1    1   -1    1   -1   -1   -1    1   -1    1   -1
Columns 61 through 63
1    1   -1

```

**Figura 2.94** Secuencia Kasami con  $k=1$  y  $m=1$ .

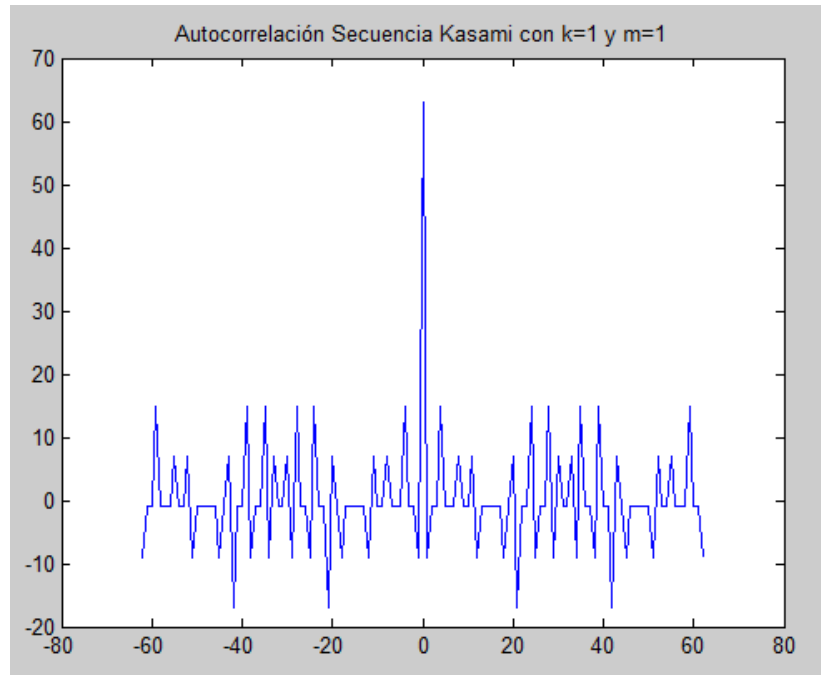
La grafica de la Autocorrelación de la secuencia Kasami presenta los valores de  $[-17 -9 -1 7 15 63]$ . Las siguientes figuras muestran los valores de la Autocorrelación obtenidos analíticamente y gráficamente en Matlab.

```

>> corr
corr =
Columns 1 through 15
-9   -1   -1   15   -1   -1   -1    7   -1   -1    7   -9   -1   -1   -1
Columns 16 through 30
-1   -1   -9   -1    7  -17   -1   -1   15   -9   -1   -1   15   -9    7
Columns 31 through 45
-1   -1    7   -9   15   -1   -1   -9   15   -1   -1  -17    7   -1   -9
Columns 46 through 60
-1   -1   -1   -1   -1   -9    7   -1   -1    7   -1   -1   -1   15   -1
Columns 61 through 75
-1   -9   63   -9   -1   -1   15   -1   -1   -1    7   -1   -1    7   -9

```

**Figura 2.95** Valores de la Autocorrelación de una secuencia Kasami con  $k=1$  y  $m=1$ .



**Figura 2.96** Autocorrelación de la Secuencia Kasami con  $k=1$  y  $m=1$ .

Para el análisis de la **correlación cruzada** entre las secuencias Kasami, se realizó la función **sec\_kasami(pol,k,m).m** que utiliza script del programa **cap2\_cod9\_kasami\_grande.m**. Siendo **pol** el polinomio primitivo generador de las secuencias Kasami y, **k** y **m** los índices de desplazamiento para las generar las secuencias Kasami.

```
clear all, clc
pol=[1 0 0 0 0 1 1];
k1=-1;
m1=-1;
k2=1;
m2=4;
sec1=sec_kasami(pol,k1,m1);
sec2=sec_kasami(pol,k2,m2);
[ind corr]=xcorrelacion(sec1,sec2);
plot(ind,corr)
title('Correlación Cruzada entre Secuencias Kasami')
```

```

>> corr
corr =

Columns 1 through 15
    15    -9    -1    -1    -1     7    -9    -9     7    15    -9    -1    -1    -9    -1

Columns 16 through 30
     7    -1    -1     7    -1    -1    -1     7    -1    -1    -1   -17    -1    15    -1

Columns 31 through 45
   -17     7    -9    -1    -9     7    -9    -9    15    -1    -1    15     7    -1    -1

Columns 46 through 60
     7    -1    -1    15    -1    -9    -1     7     7    -9     7    15    -1    -9    -9

Columns 61 through 75
    -9     7    -9     7    -9    -9    -9    -1    15     7    -9     7     7    -1    -9

```

Figura 2.97 Valores para la correlación cruzada entre secuencias Kasami.

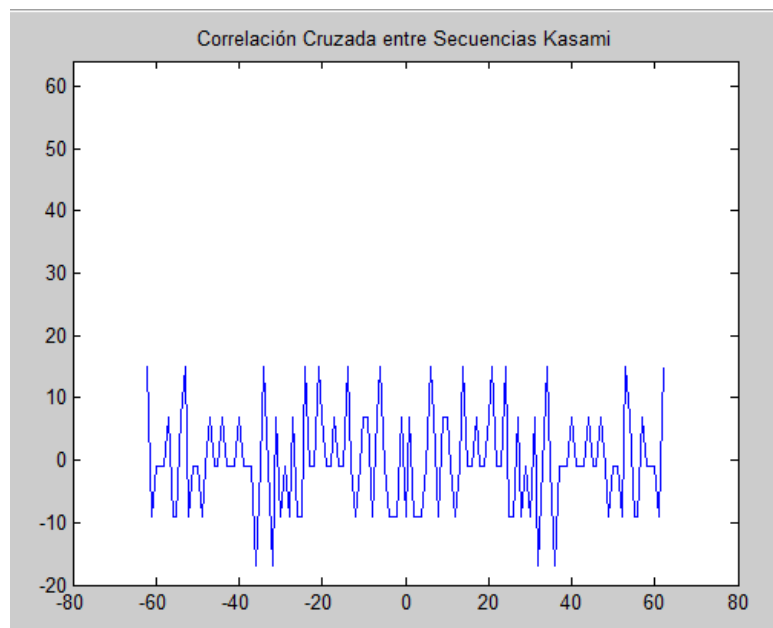


Figura 2.98 Correlación Cruzada entre Secuencias Kasami.

Se puede constatar que la **correlación cruzada** toma cinco valores:

$[-17 \ -9 \ -1 \ 7 \ 15]$

## 2.2 SECUENCIAS ORTOGONALES

Como se explicó anteriormente en un sistema multiusuario cuando se desee recuperar la información de un usuario determinado, habría que emplear la secuencia correspondiente. Solo en este caso la **correlación cruzada debería ser distinta de cero (máxima autocorrelación)**, de lo contrario debería ser cero (**correlación cruzada nula**). Además, idealmente **se debería cumplir que todas las secuencias fuesen perfectamente ortogonales entre sí**, lo que garantizaría la total separación de la información de los usuarios. Entonces existe la necesidad de investigar nuevos tipos de secuencias que cumplan con el objetivo de que sean ortogonales y que cumplan con el criterio de correlación cruzada.

### ¿Qué es la ortogonalidad?

Si se consideran dos vectores, estos son **ortogonales** si su **producto escalar** es **cero**, es decir:

$$A \cdot B = |A||B| \cos \theta = 0 \quad (\text{Ec 2.33})$$

Visto desde el **punto de vista de la correlación**, se puede **redefinir** la ortogonalidad para la **correlación cruzada en el origen**, si se cumple que el **producto interno o producto punto** de las secuencias es nulo:

$$R_{ij} = \sum_{n=0}^{N-1} x[n]y[n] = 0 \quad (\text{Ec 2.34})$$

Existen **dos tipos** de códigos o secuencias ortogonales pueden ser:

- Secuencias de Walsh.
- Códigos OVSF.

### 2.2.1 SECUENCIAS DE HADAMARD-WALSH <sup>58 59 60</sup>

En 1923, J.L Walsh definió un sistema de funciones ortogonales. La característica más importante de los **códigos de Walsh** es la **perfecta ortogonalidad entre sus códigos**, y es por ello que se utilizan en aplicaciones de comunicaciones. Las secuencias de Walsh son utilizadas en IS-95A y UMTS como códigos de canalización en el Uplink y en el Downlink.

#### Generación de las secuencias de Walsh

Los vectores de Walsh se generan a partir de la **matriz binaria de Hadamard**, donde siguiendo una estructura de conformación se pueden crear palabras código de **máxima autocorrelación** consigo misma y de **correlación cruzada cero** con las otras palabras código generadas. Es decir, se obtiene un alfabeto código con palabras ortogonales entre sí.

La rutina de creación consiste en definir una **matriz inicial de Hadamard** asignando el valor cero (0) de inicio. Por ejemplo de la **matriz de Walsh de orden cero** con valor cero (0) será:

$$W_0 = [0] \quad (\text{Ec 2.35})$$

Esta matriz dará origen a la de orden dos, modelando la nueva matriz con los mismos valores a excepción de la posición  $W_2(2,2)$ , donde se coloca la **matriz complemento**, tal como se ve:

$$W_2 = \begin{bmatrix} W_0 & W_0 \\ W_0 & \overline{W_0} \end{bmatrix} \quad (\text{Ec 2.36})$$

Donde  $\overline{W_0}$  es la matriz complemento.

---

<sup>58</sup> The Next Generation CDMA Technologies, Chen, Hsiao-Hwa, p. 49.

<sup>59</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 222-224.

<sup>60</sup> CDMA Systems Engineering Handbook, Lee, Jhong S.; Miller, Leonard E., p. 425-456.

Reemplazando los elementos en la matriz tenemos:

$$W_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (\text{Ec 2.37})$$

La **matriz de Walsh de orden cuatro** se obtiene con la misma rutina a partir de la orden dos:

$$W_4 = \begin{bmatrix} W_2 & W_2 \\ W_2 & \overline{W_2} \end{bmatrix} \quad (\text{Ec 2.38})$$

Reemplazando:

$$W_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad (\text{Ec 2.39})$$

Cada fila o columna de una matriz de Hadamard es una secuencia de Walsh. Para  $W_4$  se tienen los siguientes códigos:

**Tabla 2.27** Secuencias de Walsh generadas para  $N = 4$ .

Número de Secuencia Walsh	Secuencia Walsh
1	0 0 0 0
2	0 1 0 1
3	0 0 1 1
4	0 1 1 0

Con el mismo criterio se pueden generar matrices de órdenes superiores, donde la matriz de Hadamard se construye con la siguiente fórmula recursiva:

$$W_{2N} = \begin{bmatrix} W_N & W_N \\ W_N & \overline{W_N} \end{bmatrix} \quad (\text{Ec 2.40})$$



### Propiedades de las secuencias de Walsh:

Si se define la secuencia de Walsh  $W_i$  como la  $i$ -ésima fila o columna de una matriz de Hadamard, se pueden derivar las siguientes propiedades:

- La longitud de las secuencias de Walsh son siempre potencia de 2.
- Siempre hay  $N$  secuencias diferentes de longitud  $N$ .
- Las secuencias de Walsh son mutuamente ortogonales si están sincronizadas.
- Si dos secuencias de Walsh tienen desplazamiento en el tiempo, la función de correlación cruzada puede tomar valores mayores que el pico de la función de Autocorrelación. Aunque también es posible que la función de correlación cruzada tome un valor de cero incluso cuando existe cualquier desplazamiento en el tiempo.

### Generación de las secuencias Walsh en Matlab

En Matlab existe una **función llamada Hadamard(N)** que genera la matriz de Hadamard de orden  $N \times N$ , donde  $N$  debe ser un potencia de 2. Los elementos de la matriz responden a una **lógica bipolar negativa**, por lo tanto serán 1 o -1.

```
>> hadamard(1)

ans =

     1

>> hadamard(2)

ans =

     1     1
     1    -1

>> hadamard(4)

ans =

     1     1     1     1
     1    -1     1    -1
     1     1    -1    -1
     1    -1    -1     1
```

**Figura 2.99** Generación de Secuencias Walsh

La figura muestra la generación en Matlab de las matrices de Hadamard con  $N=1,2,4$ . Donde cada fila o columna es un código Walsh. Para obtener solamente un código basta ingresar lo siguiente:

```
%Genera un matriz hadamard de 8x8
walsh=hadamard(8)
%se escoge el código perteneciente a la cuarta fila:
walsh1=walsh(4, :)
```

```
>> walsh=hadamard(8)

walsh =

     1     1     1     1     1     1     1     1
     1    -1     1    -1     1    -1     1    -1
     1     1    -1    -1     1     1    -1    -1
     1    -1    -1     1     1     1    -1    -1
     1     1     1     1    -1    -1    -1    -1
     1    -1     1    -1    -1     1    -1     1
     1     1    -1    -1    -1    -1     1     1
     1    -1    -1     1    -1     1     1    -1

>> walsh1=walsh(4, :)

walsh1 =

     1    -1    -1     1     1     1    -1    -1
```

Figura 2.100 Obtención de un código Walsh en Matlab.

### Correlación Cruzada y Autocorrelación de las Secuencias Walsh

La **Autocorrelación** de las secuencias Walsh **no tiene buenas características** debido a que no presenta un pico único. Esto hace que se dificulte la detección del inicio de una secuencia Walsh en el receptor si no existe alguna sincronización externa.

La principal ventaja de estas secuencias es que **tienen correlación cruzada nula siempre y cuando el desfase entre las secuencias sea nulo**, es decir, cuando están completamente alineadas. Para desplazamientos distintos de cero el valor de la correlación cruzada depende fuertemente de las secuencias particulares utilizadas.

La **correlación cruzada** puede tomar valores mayores que el pico de la función de Autocorrelación e incluso un valor de cero cuando existe cualquier

desplazamiento en el tiempo. Debido a esta razón, las secuencias Walsh son óptimas solo para **comunicaciones sincrónicas**.

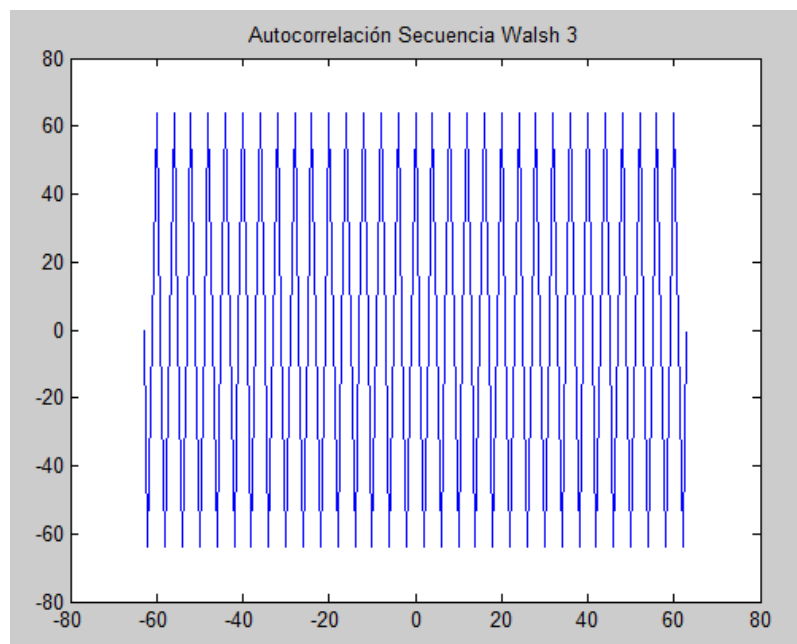
Para visualizar lo expuesto anteriormente se desarrolló el programa **cap2\_cod10\_secuencias\_walsh.m** (ver **Anexos Cap 2**), en el cual se ingresa la longitud de las secuencias Walsh y se escoge dos secuencias para realizar y graficar las funciones de Autocorrelación y Correlación Cruzada.

### Resultados:

La siguiente figura muestra el ingreso de los parámetros para generar secuencias Walsh de longitud 64 y realizar las gráficas de los valores de la Autocorrelación y Correlación cruzada simulando distintos desfases entre un par de secuencias escogidos. Las secuencias que se escogieron para el análisis fueron: 3, 33 y 56.

```
Ingrese el orden N del código de walsh: 64
Escoja dos secuencias entre 1 y 64
Ingrese el indice de la primera secuencia walsh: 3
Ingrese el indice de la segunda secuencia walsh: 33
Ingrese el indice de la tercera secuencia walsh: 56
```

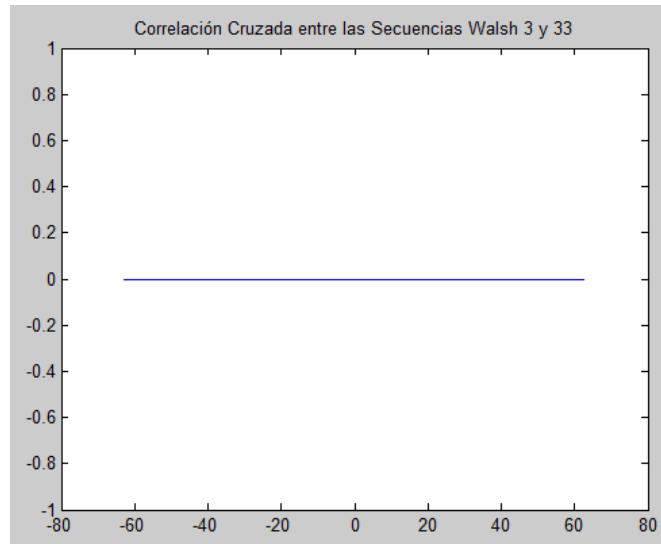
**Figura 2.101** Ingreso de los parámetros para generar las secuencias Walsh y realizar la Autocorrelación y Correlación cruzada.



**Figura 2.102** Autocorrelación de la secuencia Walsh 3.

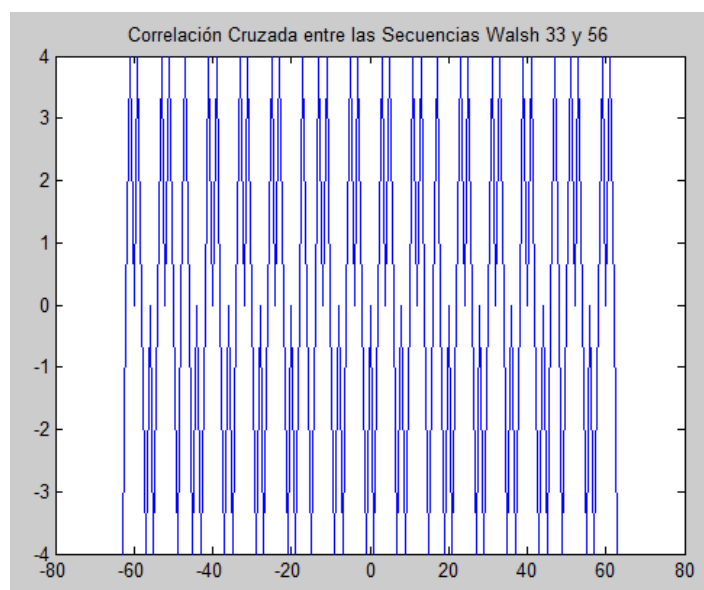
Se puede notar que la Autocorrelación de la secuencia Walsh 3 **tiene varios picos** entre 64 y -64. La presencia de varios picos en la Autocorrelación de una secuencia, dificultan la detección de la secuencia en un receptor asincrónico.

Haciendo la correlación Cruzada entre las secuencias Walsh 3 y 33, y entre las secuencias Walsh 33 y 56, se obtiene lo siguiente:



**Figura 2.103** Correlación cruzada entre las secuencias Walsh 3 y 33.

Para este caso, las secuencias Walsh 3 y 33, mantienen la **correlación cruzada** en **cero**, en cualquier desplazamiento en el tiempo



**Figura 2.104** Correlación cruzada entre las secuencias Walsh 33 y 56.

Se puede ver que la Correlación cruzada toma valores entre 4, 0 y -4 de forma periódica, en cualquier desplazamiento en el tiempo de las secuencias. La presencia de estos picos demuestra que en comunicaciones asincrónicas, los códigos pierden sus características de ortogonalidad. Debido a esta razón, las secuencias Walsh son óptimas solo para **comunicaciones sincrónicas**. Donde la **correlación cruzada es estrictamente cero**. Por lo tanto, es más fácil determinar los datos de un usuario en el receptor.

Otro método para generar códigos ortogonales es utilizando una estructura de árbol con un factor de ensanchamiento variable, a estos códigos se los conoce como **OVSF (Ortogonal Variable Spread Factor)**.

## 2.2.2 CÓDIGOS OVSF (ORTOGONAL VARIABLE SPREADING FACTOR)

En los sistemas CDMA convencionales usados para transmitir voz en sistemas celulares, la velocidad es una constante. Correspondientemente, los códigos utilizados tienen una longitud fija.

Con el paso del tiempo, los servicios de datos se han incrementado significativamente en las redes celulares. De hecho, la principal característica de los sistemas de tercera generación es soportar Calidad de Servicio (QoS) diferenciada para las nuevas aplicaciones multimedia, las cuales son típicamente de velocidad variable. Soportar servicios de datos a altas velocidades se puede conseguir mediante el empleo de múltiples códigos para una conexión y por lo tanto se requiere de múltiples transceivers para cada nodo, incrementando la complejidad en el hardware del sistema de comunicaciones.

Motivados por soportar servicios de tasas variables a un costo bajo, se desarrolló un código de longitud variable, conocido como **códigos OVSF (Ortogonal Variable Spread Factor)**.

La idea de utilizar el OVSF es permitir a los códigos tener longitudes variables y asignar los códigos más pequeños a las tasas de transmisión más altas. De esta manera, usando un código OVSF, solo se requiere de un transceiver por nodo.

En muchas fuentes se puede encontrar que se habla de los códigos Walsh y de los códigos OVSF por separado, pero si se realiza una comparación de los códigos OVSF con los *Walsh-Hadamard*, veríamos que contienen las mismas secuencias de código. La única diferencia entre estos códigos es como están indexados, es decir, el orden en el que se crean las secuencias. Para generar secuencias OVSF se considera una matriz  $N \times N$ ,  $C_N$ , la cual se forma recursivamente de la siguiente manera:<sup>61</sup>

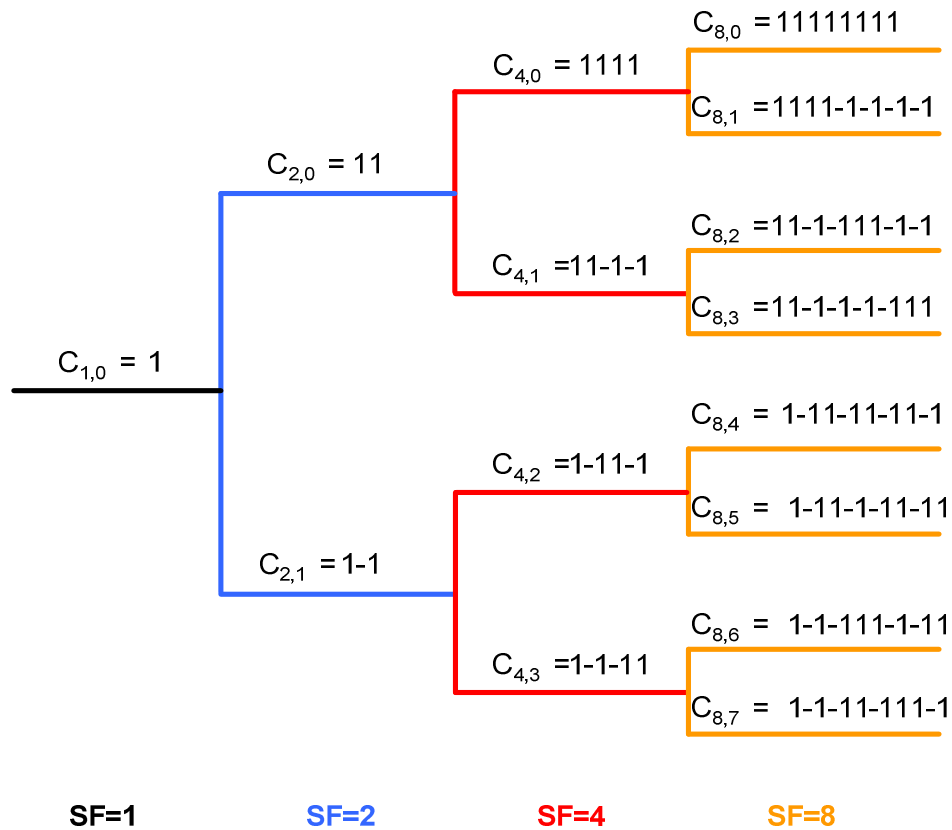
- Primero se define  $C_1=[1]$ .
- Luego se asume que  $C_N$  está definida y que  $C_N(k)$  denota la  $k^{\text{th}}$  fila de  $C_N$ .  $C_N$  es definida solamente para  $N$  potencia de 2
- Ahora  $C_{2N}$  se define como:

$$C_{2N} = \begin{bmatrix} C_N(0) & C_N(0) \\ C_N(0) & -C_N(0) \\ C_N(1) & C_N(1) \\ C_N(1) & -C_N(1) \\ \dots & \dots \\ C_N(N-1) & C_N(N-1) \\ C_N(N-1) & -C_N(N-1) \end{bmatrix} \quad (\text{Ec 2.41})$$

La matriz  $C_{2N}$  sigue la condición de que las filas de  $C_N$  son ortogonales una de otra. Por lo tanto, los códigos OVSF son definidos recursivamente por una estructura de árbol, tal como se muestra en la **figura 2.105**, donde se muestran cuatro capas o ramas de la estructura de árbol. En los códigos OVSF se deben especificar dos parámetros: el factor de esparcimiento (SF), el cual es la longitud del código, y el índice del código, el cual debe ser un entero en el rango de 1 a  $N$ .

---

<sup>61</sup> The Next Generation CDMA Technologies, Lee, Jhong S.; Miller, Leonard E., p. 136-141.



**Figura 2.105** Estructura de árbol para la generación de códigos OVFS.

Se puede observar que comenzando en  $C_{1,0}=1$ , una serie de  $2^k$  códigos de dispersión con longitud de  $2^k$  chips son generados en la  $k$ th capa. Además, se puede notar que cualquier par de códigos de diferentes capas son ortogonales, excepto en el caso de que uno de los códigos sea el código madre del otro.

Debido a que se debe mantener la ortogonalidad los códigos válidos para los usuarios dentro de una celda se restringen. Por ejemplo, si el código  $C_{4,1}$  en el árbol es asignado a un usuario, los códigos  $C_{1,0}$ ,  $C_{2,0}$ ,  $C_{8,2}$ ,  $C_{8,3}$ , y así sucesivamente, no pueden ser asignados a ningún otro usuario en la misma celda. En general, todos los códigos padres e hijos de un código específico, no pueden ser usados en una misma celda para mantener la ortogonalidad entre usuarios.

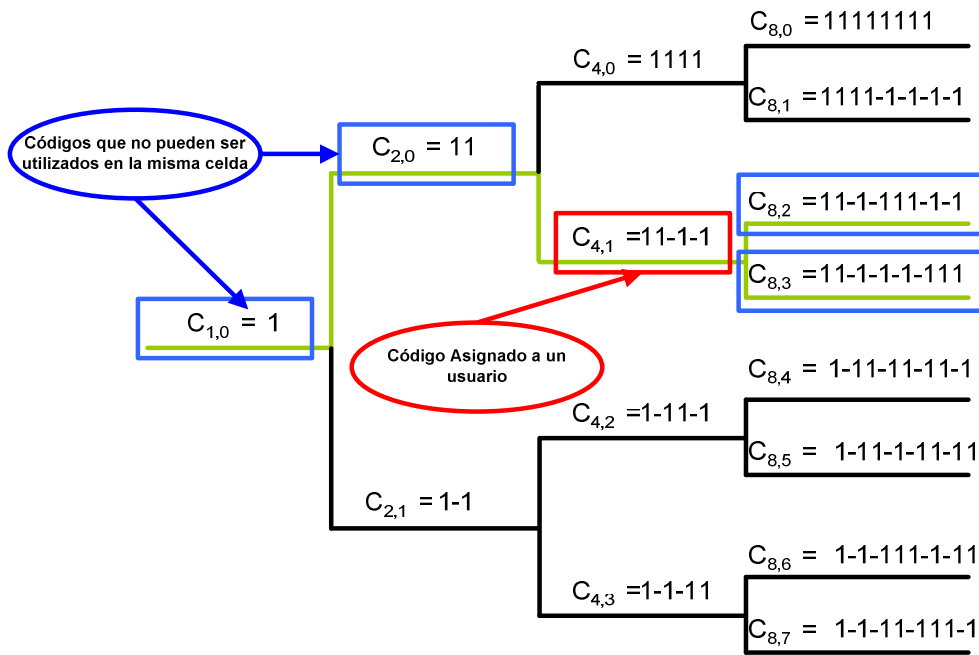


Figura 2.106 Asignación de los códigos OVSF a los usuarios dentro de una celda.

**Generación de los códigos OVSF en Matlab**

**Programa cap2\_cod11\_ovsf (ver Anexos Cap 2)**

Para la generación de los códigos OVSF se usa la siguiente forma recursiva a partir del factor de esparcimiento (SF) y el índice del código.

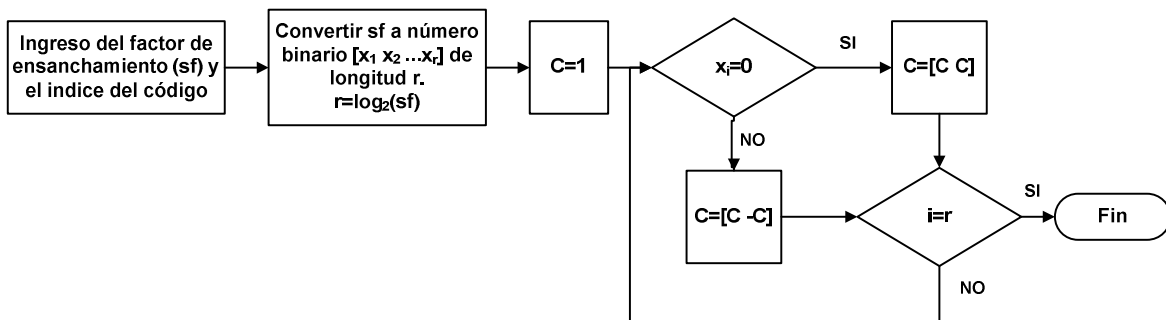


Figura 2.107 Diagrama para generar un código OVSF.



**Resultados:**

```

Ingrese el factor de esanchamiento: 8
Ingrese el indice del código: 5
El codigo OVSF de sf 8 y k= 5 es:

c =

      1      -1      1      -1      -1      1      -1      1
>>

```

**Figura 2.108** Código OVSF con factor de esparcimiento 8 e índice 5.

Se puede notar que le código corresponde a  $C_{8,5}$  de la estructura de árbol para la generación de los códigos OVSF.

**Propiedades de los códigos OVSF**

Los OVSF tienen dos características principales.

- La propiedad de ortogonalidad de los OVSF asegura que diferentes usuarios de la misma celda no interfieran con alguna otra, pues preservan su ortogonalidad entre **diferentes tasas y factores de esparcimiento**.
- El aspecto variable de OVSF soporta diferentes tasas de datos del mismo código de árbol: bajas tasas de datos pueden ser codificadas con OVSFs largos, mientras que altas tasas de datos son codificadas con OVSFs cortos. Un bit codificado con un OVSF de longitud 256 puede ser representado por 256 chips, mientras que un bit codificado con un OVSF de longitud 4 puede ser representado por 4 chips. Usando un OVSF largo se tiene la ventaja de sumar redundancia para transmitir la información.

**2.3 SECUENCIAS DE SINCRONIZACIÓN**

Cómo se pudo ver en los códigos anteriores como los códigos Walsh y OVSF, solo sirven para escenarios de comunicaciones sincronizadas. Ahora, para aplicaciones donde no se pueda tener dicha sincronización es importante

investigar nuevos tipos de secuencias que permitan tener una **autocorrelación maximizada**, como son las secuencias de **Barker**.

### 2.3.1 SECUENCIAS DE BARKER

Las secuencias o códigos de Barker se caracterizan por su función de correlación; que es 1 para todos los desplazamientos excepto en el desplazamiento cero. Por esta razón, estos códigos son usados comúnmente como **preámbulo para la sincronización de tramas** en la comunicación digital. La función de Autocorrelación es impulsiva, lo mismo que en las m-secuencias. Por lo tanto los códigos de Barker son también ideales para la detección de datos de información. Desafortunadamente, los códigos de Barker solo existen para los tamaños de  $L = 2, 3, 4, 5, 7, 11, 13$ , y por lo tanto ofrecen un proceso de ganancia limitado.

Los códigos de Barker fueron desarrollados originalmente para las emisiones de radar. Por ejemplo, pueden ser usados como un preámbulo de una secuencia PN larga con un solo propósito, simplificar la sincronización. La más notable propiedad de los códigos de Barker es que el menor pico de su función de Autocorrelación siempre consiste es 1. Las secuencias de Barker no son producto de los LFSRs, si no que son **hard coded**, es decir, son generados de manera manual de modo que, en el proceso de investigación, produzcan la máxima autocorrelación posible. La siguiente tabla presenta la lista completa de los códigos de Barker:

**Tabla 2.28** Códigos de Barker de longitud L.

Longitud del Código de Barker (L)	Código de Barker
1	-1
2	-1 1
3	-1 -1 1
4	-1 -1 1 -1
5	-1 -1 -1 1 -1
7	-1 -1 -1 1 1 -1 1
11	-1 -1 -1 1 1 1 -1 1 1 -1 1
13	-1 -1 -1 -1 -1 1 1 -1 -1 1 -1 1 -1

### Generación de los códigos Barker en Matlab

Para el análisis de las secuencias de Barker se desarrolló la función ***Barker(longitud).m*** (ver Anexos Cap 2), donde longitud es el tamaño de la secuencia Barker a generar.

Para Analizar las propiedades de Autocorrelación de las secuencias de Barker, basta ingresar el siguiente código:

```
longitud=13;
secuencia=barker(longitud)
[ind corr]=xcorrelacion(secuencia,secuencia);
plot(ind,corr)
title('Autocorrelación de la Secuencia Barker 13')
```

```
>> secuencia
secuencia =
    -1    -1    -1    -1    -1     1     1    -1    -1     1    -1     1    -1

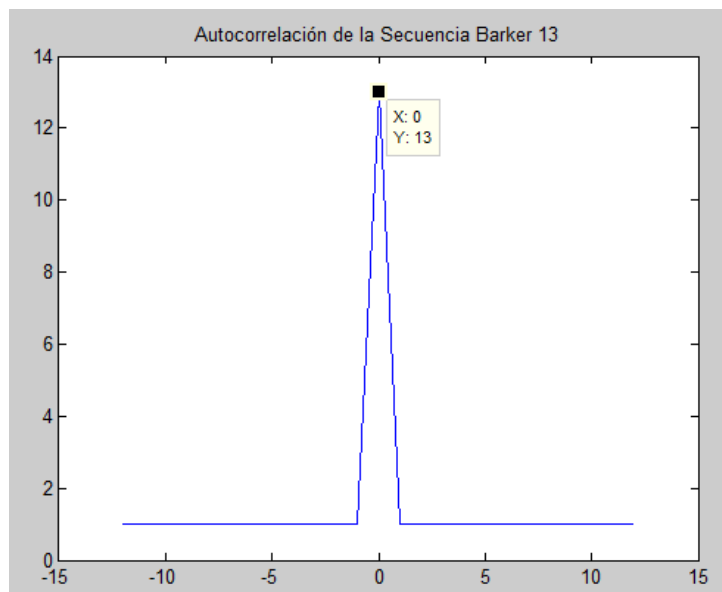
>> corr
corr =

Columns 1 through 15
     1     1     1     1     1     1     1     1     1     1     1     1    13     1     1

Columns 16 through 25
     1     1     1     1     1     1     1     1     1     1
```

**Figura 2.109** Secuencia Barker de longitud 13 y sus valores de Autocorrelación.

Se nota que la Autocorrelación se parece a la que tiene un m-secuencia. La diferencia radica en los valor 1 que toma para los desplazamientos de la Secuencia Barker.



**Figura 2.110** Gráfica de la Autocorrelación de la secuencia Barker de longitud 13.

Esta propiedad de Autocorrelación hace que en aplicaciones de sistemas de espectro ensanchado, el espectro sea mucho más uniforme y por lo tanto el desempeño de los receptores mejora notablemente.

Como se explicó anteriormente, las características de autocorrelación este tipo de secuencias son muy utilizadas ya sea como preámbulo de una trama de información para simplificar la sincronización, como secuencia ensanchadora del estándar 802.11a/b, o en aplicaciones identificación de objetos como el radar.

A continuación se realizará un ejemplo de aplicación de las secuencias Barker.

### **Determinación de la distancia de un blanco <sup>62</sup>**

La distancia de un blanco es determinada por el tiempo que le toma a la señal electromagnética transmitida por el radar en viajar de ida y de vuelta en la atmósfera. A este tiempo se le conoce como tiempo de retardo  $t_r$ . La energía electromagnética en el espacio libre viaja a la velocidad de la luz  $c = 3 \times 10^8 \text{ m/s}$ . Por lo tanto, el tiempo para que la señal viaje hacia la posición del blanco a una distancia  $D$  y regrese al radar es  $2D/c$ . Entonces la distancia del blanco es:

---

<sup>62</sup> Modelado de un Radar Doppler de Pulsos (PDR), Arrambí Díaz, Marcial, p. 58-59.

$$D = \frac{ct_r}{2} \quad (\text{Ec 2.42})$$

Por lo tanto, para obtener el tiempo de retardo de la señal se recurre a la correlación cruzada, la cual es máxima cuando la señal transmitida y la señal recibida son similares. Al correlacionar estas señales, se está obteniendo el tiempo preciso cuando las señales presentan la máxima similitud entre ellas. Este tiempo es el retardo entre de la señal recibida respecto a la señal transmitida.

La función de correlación, se puede escribir de la siguiente forma:

$$r_{s,s_r}(t_r) = \sum_{k=-\infty}^{\infty} s(k)s_r(k - t_r) \quad (\text{Ec 2.43})$$

Donde  $s(n)$  es la señal transmitida y  $s_r(k-t_r)$  es la señal reflejada. Además la señal  $s_r(n)$  es:

$$s_r(n) = \alpha s(n - t_r) + W(n) \quad (\text{Ec 2.44})$$

Donde  $W(n)$  es el ruido aditivo del canal.

De esta manera se sabe que la señal reflejada posee un retardo y ruido por lo que al correlacionarla con la señal transmitida, se tendrá una señal que en algún tiempo  $t_r$  tendrá un valor máximo de amplitud. **Para incrementar la precisión en el cálculo del tiempo de retraso  $t_r$ , y a su vez la distancia del blanco, se utiliza el código de Barker.**

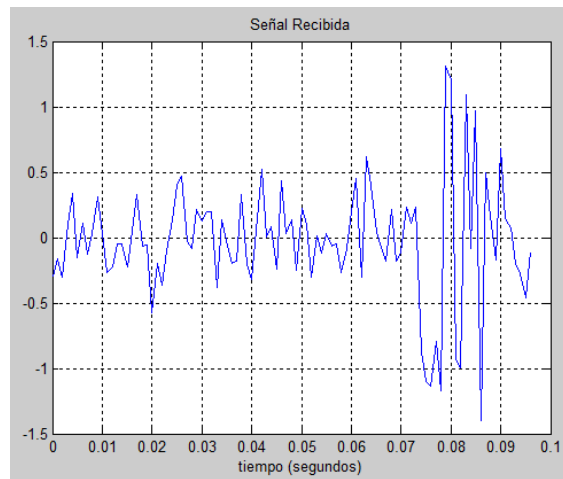
Para realizar el ejemplo expuesto, se realizó el programa **cap2\_cod12\_ejemplo\_barker.m** (ver Anexos Cap 2), donde se define una secuencia Barker de longitud 13 con la función **barker(longitud).m**

También se define un retraso aleatorio de la señal para realizar la correlación cruzada y detectar la señal entre 30 y 100 tiempos de bit. Se define un tiempo de bit  $t_b=1$  ms. La señal transmitida es una secuencia Barker y la señal recibida será la secuencia Barker retrasada más el ruido de un canal AWGN. Se calcula la

correlación cruzada para determinar el retraso que ha sufrido la señal transmitida originalmente y se lo muestra gráficamente.

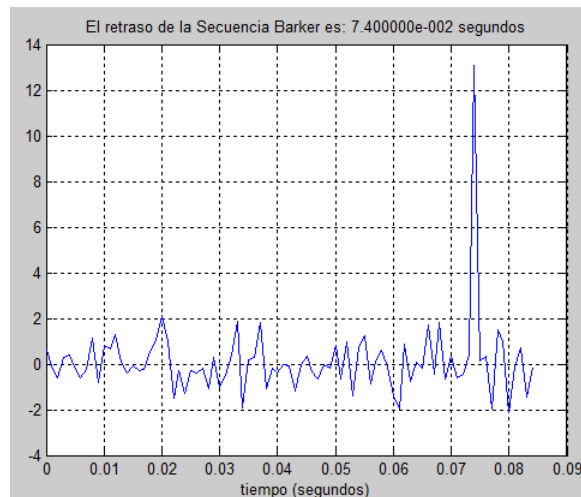
## Resultados:

*Señal Recibida con Ruido.*



**Figura 2.111** Señal Recibida en el Radar.

*Retraso de la Secuencia Barker.*



**Figura 2.112** Correlación cruzada entre la Señal Transmitida y la Señal Recibida por el Radar.

Se puede ver claramente, para este ejemplo, que se produce un pico de amplitud máximo a  $t=74$  ms. Para saber si el resultado es el esperado, se revisa el valor calculado y guardado en la variable **delay**.

```
>> delay  
  
delay =  
  
    74
```

**Figura 2.113** Valor de delay calculado aleatoriamente.

Por lo tanto el algoritmo propuesto es válido para calcular la distancia a un blanco a través de la correlación cruzada de las secuencias Barker. Además se nota claramente la importancia de las características de autocorrelación y correlación cruzada de códigos, para ser aplicados de una manera eficiente.

## CAPÍTULO III

### 3. ANÁLISIS DE LAS TÉCNICAS DE ESPECTRO ENSANCHADO DE CDMA

En el capítulo 2 se mostraron las ideas fundamentales para conseguir que el pico del espectro de potencias se reduzca y a la vez se ensanche. Ahora se mostrará su aplicación fundamental a comunicaciones digitales: el proceso de ensanchamiento espectral en el transmisor y desensanchamiento o recuperación en el receptor. En este capítulo profundizará en las aplicaciones a comunicaciones digitales basadas en espectro ensanchado.

#### 3.1 HISTORIA DE LOS SISTEMAS DE ESPECTRO ENSANCHADO

Históricamente, en 1942 Hedy Lamarr y George Antheil presentaron una patente que contiene los fundamentos sobre la teoría de espectro disperso <sup>63</sup> en los cuales al dispersar el espectro a través del uso de un código personalizado, se lograba un aumento del ancho de banda ocupado y a la vez una reducción del pico espectral de modo que ahora la comunicación se podría hacer invisible. También se fortalece la transmisión frente al ruido y a las interferencias intencionales o no. Su sistema fue la base para desarrollar una nueva técnica conocida como **Frequency Hopping** <sup>64</sup>. En la década de los 90s aparecieron los primeros equipos que explotan estos principios.

---

<sup>63</sup> <http://sss-mag.com/shistory.html>

<sup>64</sup> <http://www.paristransatlantic.com/antheil/mainpage/patent.html>



Hoy en día, la tecnología de espectro disperso se aplica en muchos campos y es, junto a OFDM y MIMO, la base fundamentes de todo tipo de comunicaciones inalámbricas: telefonía celular y redes de datos inalámbricas en general.

### 3.2 Tipos de Sistemas de Espectro **DISPERSO**

Existen 3 tipos básicos de sistemas de espectro disperso los cuales pueden combinarse para aprovechar ciertas ventajas de unos u otros sistemas:

- Espectro ensanchado por Secuencia Directa (Direct Sequence Spread Spectrum: DS/SS).
- Espectro ensanchado por Salto en Frecuencia (Frequency Hopping Spread Spectrum: FH/SS).
- Espectro ensanchado por Salto en Tiempo (Time Hopping Spread Spectrum-TH/SS).

Como ya se anotó, actualmente existen también esquemas híbridos que explotan al máximo las capacidades de aquellos sistemas.

### 3.3 **TÉCNICAS DE MODULACIÓN DIGITAL ASOCIADAS CON ESPECTRO ENSANCHADO**

**PSK:** asociada generalmente con DS/SS y es apropiada en aplicaciones donde la coherencia de fase entre TX y Rx puede ser mantenida en un intervalo de tiempo que abarca varios símbolos o bits.

**FSK:** generalmente usada con FH/SS y es apropiada en aplicaciones donde la coherencia de fase entre TX y Rx no puede ser mantenida debido a las características cambiantes del canal de comunicación.

### 3.4 ANÁLISIS DE LOS SISTEMAS DE SECUENCIA DIRECTA (DIRECT SEQUENCE): DS-SS

#### 3.4.1 GENERALIDADES DE UN SISTEMA DS-SS<sup>65</sup>

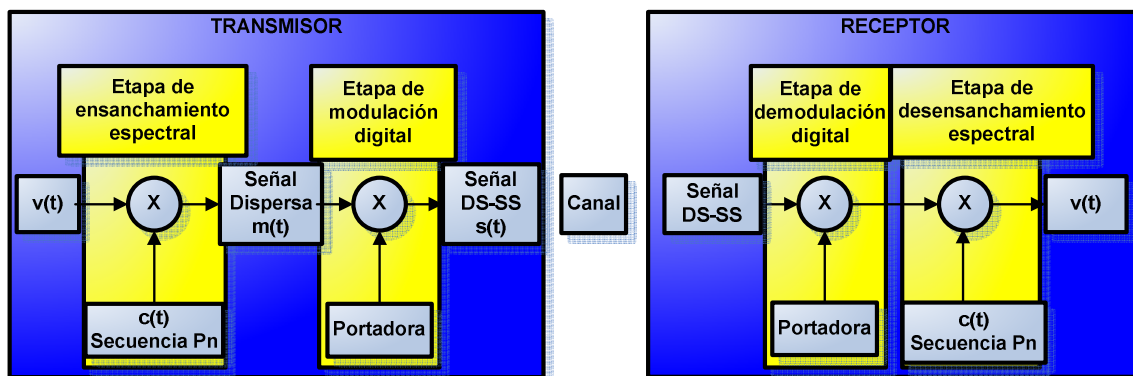


Figura 3.1 Sistema DSSS.

#### 3.4.1.1 Transmisor

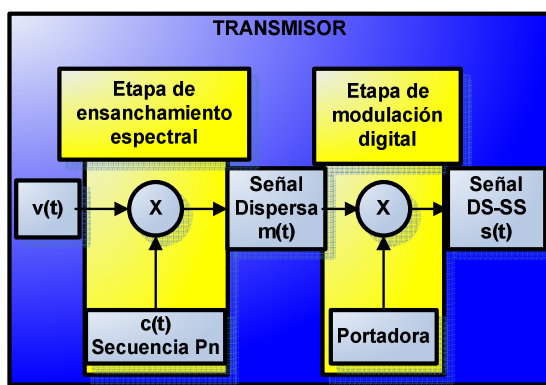


Figura 3.2 Transmisor DSSS.

En este tipo de modulación, la señal digital de información atraviesa por dos etapas:

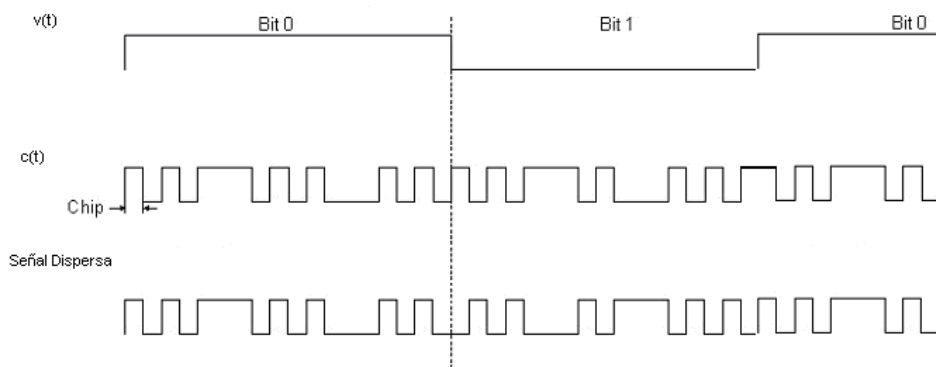
- Es modulada mediante una secuencia pseudoaleatoria, que tiene una velocidad mucho mayor que la de la señal de información (etapa de ensanchamiento espectral).

<sup>65</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 161-164.

b) Luego es transmitida utilizando algún tipo de modulación digital como se observa en la figura anterior. En esta etapa se utiliza una portadora analógica donde se introducirá la información de amplitud, fase o frecuencia.

### **a) Etapa de Ensanchamiento Espectral**

La etapa de ensanchamiento espectral ha sido explicada en detalle en el capítulo 2.

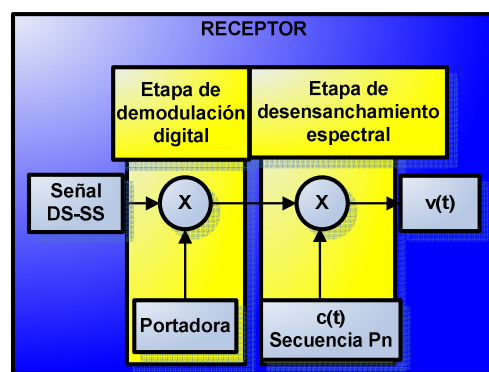


**Figura 3.3** Ejemplo para un  $v(t)$  dado.

### **b) Etapa de Modulación Digital**

Luego de multiplicar el mensaje por la secuencia pseudoaleatoria, el siguiente paso consiste en la modulación de esta señal usando cualquier tipo de modulación digital convencional, siendo PSK la más usual.

#### **3.4.1.2 Recuperación de la Señal Digital original (Receptor)**



**Figura 3.4** Receptor DSSS simplificado.

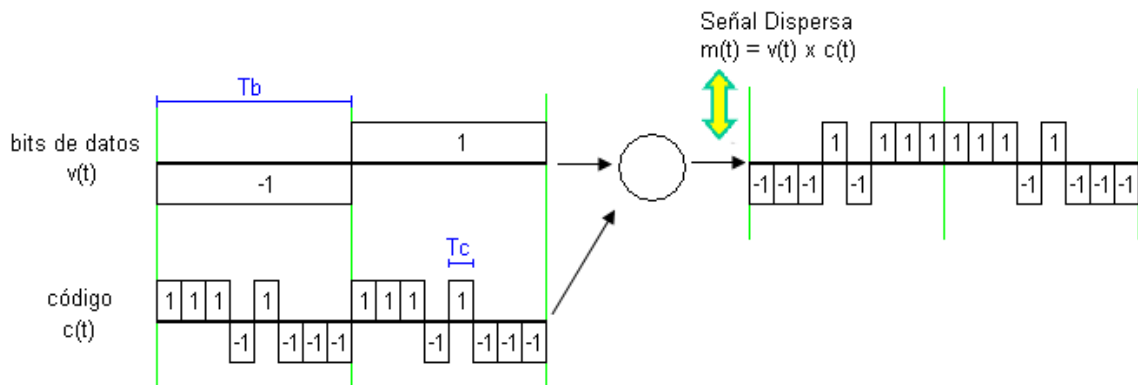
Para recuperar la señal original, se sigue el proceso opuesto.

Es decir, para recuperar la señal original, se siguen los siguientes pasos:

- Se multiplica la señal por la misma portadora analógica utilizada en el transmisor (demodulación).
- La señal demodulada, se multiplica por una réplica de la secuencia PN utilizada en el transmisor (DESEXPANSOR O DESPREADER).

### 3.4.2 ANÁLISIS MATEMÁTICO DEL TRANSMISOR DE SISTEMAS DE SECUENCIA DIRECTA <sup>66</sup>

#### 3.4.2.1 Etapa de Ensanchamiento Espectral



**Figura 3.5** Operación de etapa de ensanchamiento espectral.

Se considera una secuencia de bits de información (banda base) con una tasa  $R$  (bits/seg) y un tiempo de bit  $T_b = 1/R$ .

Esta señal de información es denotada como  $v(t)$ . Su ancho de banda mínimo requerido es:  $1/T_b = R$

Por otro lado la secuencia pseudoaleatoria se denota como  $c(t)$ .

Su ancho de banda mínimo requerido es:  $1/T_c$ .

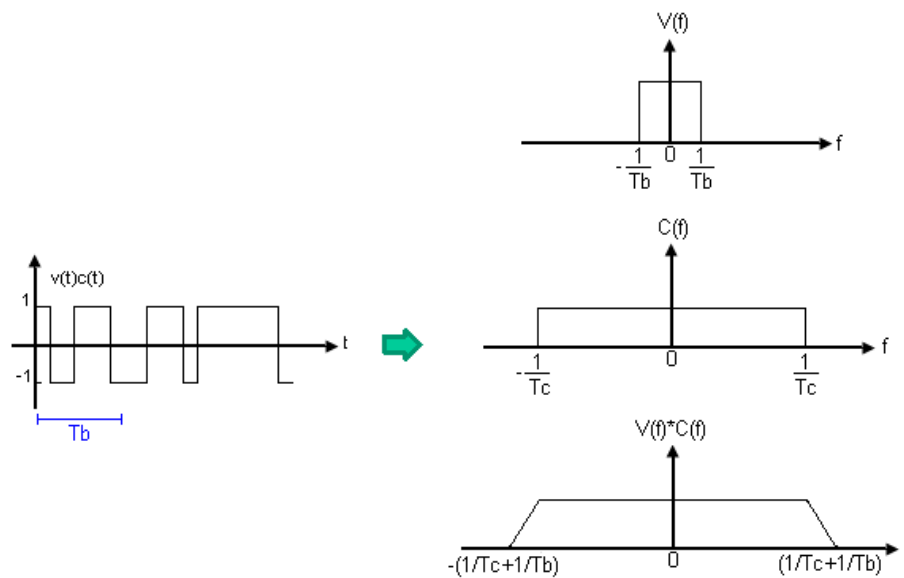
<sup>66</sup> Contemporary Communication Systems using MATLAB, Proakis, John; Salehi, Masoud, p. 391-394.

La señal resultante de multiplicar  $v(t)$  y  $c(t)$  produce un ensanchamiento o dispersión del ancho de banda de la información:

$$m(t) = v(t) \times c(t) \quad (\text{Ec 3.1})$$

Visto desde el punto de vista de la frecuencia:

$$M(f) = V(f) * C(f) \quad (\text{Ec 3.2})$$



**Figura 3.6** Ensanchamiento de Espectro, visto en el dominio de la frecuencia.

Esto se puede comprobar con el ejemplo de ensanchamiento de espectro aplicado en el capítulo 2.

### 3.4.2.2 Etapa de Modulación: Generación de la Señal PSK

Como ya se explicó, la señal dispersa  $m(t) = v(t) \times c(t)$  debe ser multiplicada por una portadora analógica  $\text{sen}(2\pi f_o t)$ :

$$s(t) = v(t)c(t)A_c \text{sen}(2\pi f_o t) \quad (\text{Ec 3.3})$$

Entonces se genera una onda modulada en amplitud con portadora suprimida.

Como  $v(t)c(t) = \pm 1$  para cualquier valor de  $t$ , la ecuación anterior, se reduce a:

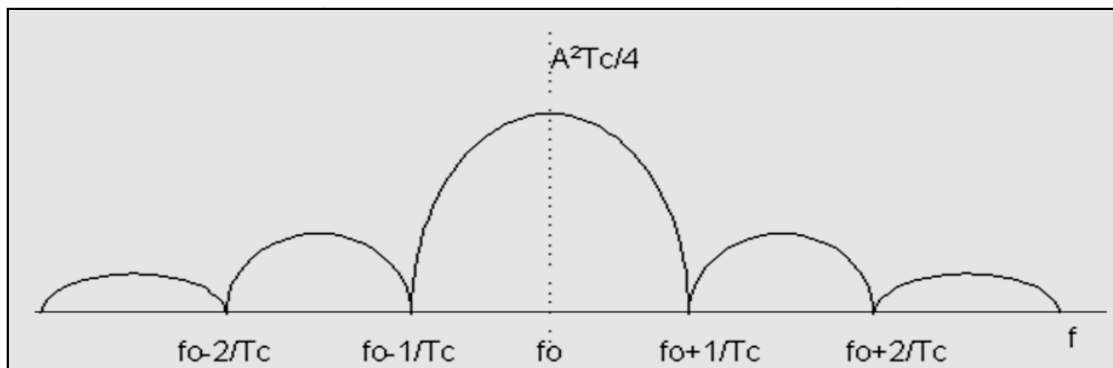
$$s(t) = \pm A_c \text{sen}(2\pi f_o t) \quad (\text{Ec 3.4})$$

Por tanto, la señal generada es una señal B-PSK cuya fase varía al ritmo de  $1/T_c$ .

Se puede demostrar que la función de densidad espectral de potencia de la señal DSSS BPSK es:

$$\frac{A^2 T_c}{4} \{ \text{sinc}^2(f - f_o) T_c + \text{sinc}^2(f + f_o) T_c \} \quad (\text{Ec 3.5})$$

y se muestra a continuación:



**Figura 3.7** Densidad Espectral de Potencias de la señal DSSS BPSK.

Si se toma el lóbulo principal, el ancho de banda de la señal modulada es  $2/T_c$  o  $2 \times F$  (chip).

En otras palabras, se obtiene la misma señal ensanchada solo que trasladada en el espectro hacia la zona de la portadora analógica. Por lo tanto, el ancho de banda de esta señal modulada resulta mucho mayor que el ancho de banda que produciría modular directamente los datos originales.

### 3.4.2.3 Potencia de la Señal Modulada en los Sistemas de Espectro Ensanchado

Al final del modulador se tiene que:

$$\begin{aligned}
 P &= \frac{1}{T} \int_{-T/2}^{T/2} A_c^2 \text{sen}^2(2\pi f_o t) m^2(t) dt \\
 P &= \frac{1}{T} \int_{-T/2}^{T/2} A_c^2 \frac{1 - \cos(4\pi f_o t)}{2} m^2(t) dt \\
 P &= A_c^2 \left\{ \frac{1}{T} \int_{-T/2}^{T/2} \frac{m^2(t)}{2} dt - \frac{1}{T} \int_{-T/2}^{T/2} m^2(t) \frac{\cos(4\pi f_o t)}{2} dt \right\} \\
 \frac{1}{T} \int_{-T/2}^{T/2} m^2(t) \frac{\cos(4\pi f_o t)}{2} dt &= 0
 \end{aligned}$$

$$\begin{aligned}
 P &= A_c^2 \frac{1}{T} \int_{-T/2}^{T/2} \frac{m^2(t)}{2} dt = \frac{A_c^2}{2} \frac{1}{T} \int_{-T/2}^{T/2} m^2(t) dt \\
 \frac{1}{T} \int_{-T/2}^{T/2} m^2(t) dt &= P_m
 \end{aligned}$$

$$P = \frac{A_c^2}{2} P_m \quad (\text{Ec 3.6})$$

Como conclusión, la potencia de la onda ya modulada depende tanto de la amplitud de la portadora como de la potencia del mensaje o información.

Como un caso particular,  $m(t)$  vale  $+1L$  ó  $-1L$ .

Entonces,  $m^2(t) = 1$  (siempre), y por tanto la potencia se reduciría a:

$$P = A_c^2 / 2 \rightarrow A_c = \sqrt{2P} \quad (\text{Ec 3.7})$$

### 3.4.3 CARACTERIZACIÓN DE LOS SISTEMAS DE ESPECTRO ENSANCHADO DE SECUENCIA DIRECTA<sup>67</sup>

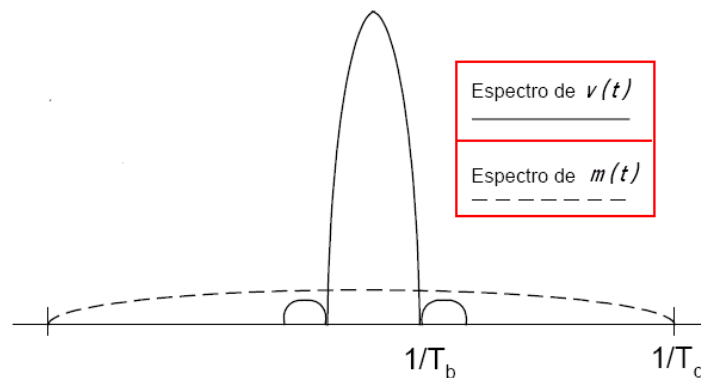
Para caracterizar los sistemas de espectro disperso se usan dos criterios:

- a) La **ganancia de procesado** ( $G_p$ ).
- b) El **margen sobre la interferencia** (jamming margin,  $J$ ).

---

<sup>67</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 167-184.

### 3.4.3.1 Ganancia de Procesado



**Figura 3.8** Espectros de los datos originales y de la secuencia ensanchada.

Normalmente se cumple que  $T_c < T_b$  y por tanto el ancho de banda de la señal  $m(t)$  se incrementa respecto al espectro de  $v(t)$  en un **factor llamado ganancia de procesado,  $G_p$** , que se define como:

$$G_p = \frac{T_b}{T_c} \quad (\text{Ec 3.8})$$

$T_b / T_c$  es usualmente seleccionada para ser un **entero** y representa el número de chips de la secuencia PN por bit de información. Otra interpretación de este valor, es que representa el número de transiciones de fase en la señal transmitida en un intervalo de bit  $T_b$ .

### 3.4.3.2 Margen de Interferencia (Jamming Margin, J)

El margen de interferencia indica la máxima interferencia que puede tolerar el sistema, con una tasa de error aceptable.

$$J = G_p - \text{SNR}_{\min} \quad (\text{Ec 3.9})$$

Donde  $\text{SNR}_{\min}$  es la mínima relación señal a ruido aceptable a la entrada del receptor.



El principal problema de los sistemas DS-SS es que el rechazo a la interferencia depende del valor de correlación que resulte entre el código original y la secuencia interferente.

### 3.4.4 RECUPERACIÓN DE LA SEÑAL DIGITAL ORIGINAL<sup>68</sup>

#### 3.4.4.1 Caso ideal sin Ruido y sin Interferencias

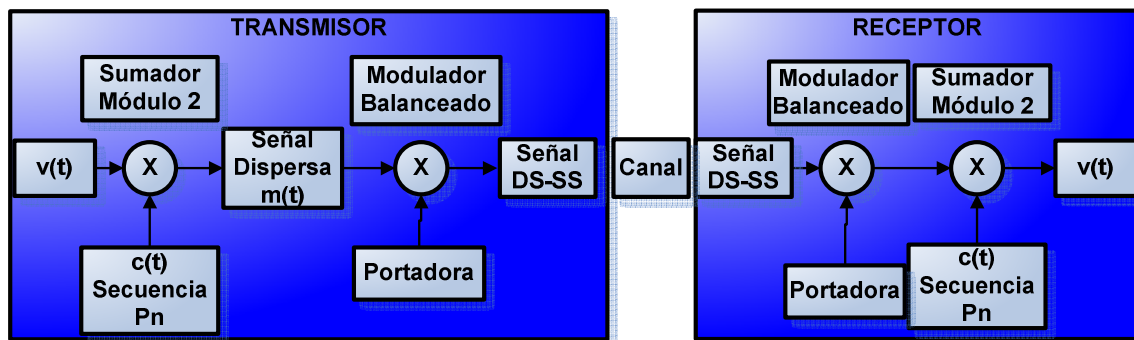


Figura 3.9 Sistema DSSS con canal ideal.

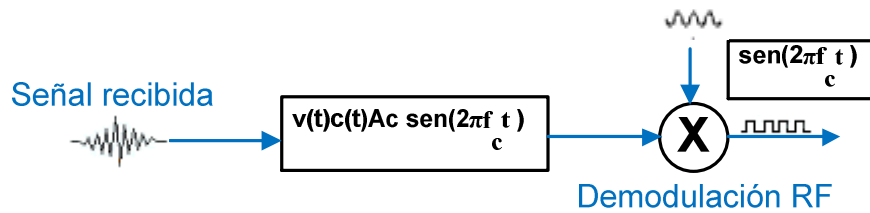
Ya que el canal no afecta a la señal transmitida, en el receptor se obtiene:

$$v(t)c(t)A_c \text{sen}(2\pi f_c t) \quad (\text{Ec 3.20})$$

Para poder recuperar la información original, se debe realizar los algoritmos opuestos hechos en el transmisor:

- a) Demodulación: retirar efecto de la portadora analógica.

<sup>68</sup> Contemporary Communication Systems using MATLAB, Proakis, John; Salehi, Masoud, p. 395-397.



**Figura 3.10** Recepción DSSS con canal ideal.

Para realizar este proceso de demodulación, se multiplica por una portadora local de la misma frecuencia que la portadora del transmisor. Así se obtiene:

$$v(t)c(t)A_c \sin(2\pi f_o t) \times \sin(2\pi f_o t) = v(t)c(t)A_c \sin^2(2\pi f_o t) \quad (\text{Ec 3.11})$$

Por la identidad trigonométrica:

$$\sin^2(2\pi f_o t) = \frac{1 - \cos(4\pi f_o t)}{2} \quad (\text{Ec 3.12})$$

Se obtiene:

$$v(t)c(t)A_c \sin^2(2\pi f_o t) = v(t)c(t)A_c \left( \frac{1 - \cos(4\pi f_o t)}{2} \right) \quad (\text{Ec 3.13})$$

Se observa que se obtiene un término  $v(t) \times c(t)$  y otro ubicado en alta frecuencia. Por tanto, la recuperación de  $s(t) = v(t) \times c(t)$  se realiza mediante un **filtro pasa banda**:

$$\frac{A_c}{2} v(t)c(t) \quad (\text{Ec 3.14})$$

b) Para obtener los datos originales  $v(t)$ , se debe multiplicar  $s(t)$  por una réplica del código (secuencia PN) del transmisor:

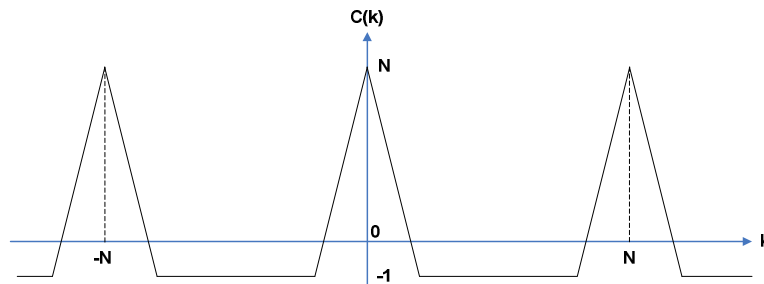
$$\frac{A_c}{2} v(t)c(t) \times c(t) = \frac{A_c}{2} v(t) \quad (\text{Ec 3.15})$$

¿Pero cómo puede darse esto?

El objetivo anterior, solo se consigue teniendo una **clase muy especial de secuencias** cuya **autocorrelación** sea igual a **una constante** la cual puede ser removida para obtener así los datos originales:

$$\begin{aligned} &= \frac{A_c}{2} v(t) \times k_1 \\ &= v(t) \times K \end{aligned} \quad (\text{Ec 3.16})$$

Como se estudió en el capítulo anterior, dicha clase de secuencias son las secuencias PN cuya autocorrelación es:

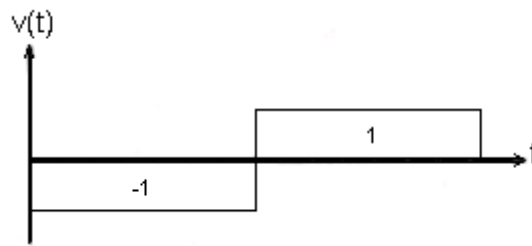


**Figura 3.11** Autocorrelación de una M-secuencia.

Como resultado de este procedimiento, se obtiene el mismo mensaje original, con una variación en amplitud.

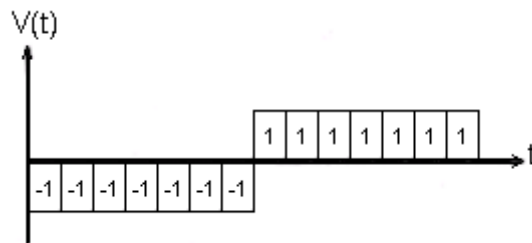
Cabe recalcar que en el caso práctico, la señal  $v(t)$  desensanchada tiene que pasar por un integrador para que se sumen los chips que conforman cada bit de la señal recuperada y luego se necesita realizar una toma de decisión para recuperar los bits originalmente transmitidos.

Por ejemplo, siendo la señal  $v(t)$ :



**Figura 3.12** Señal  $v(t)$  de datos originales.

Pero luego de la multiplicación por la señal ensanchadora en el receptor se obtiene la señal  $V(t)$ :



**Figura 3.13** Señal  $V(t)$  luego del desensanchamiento en el receptor.

Para recuperar los datos originales con el integrador y la toma de decisión se realiza la operación:

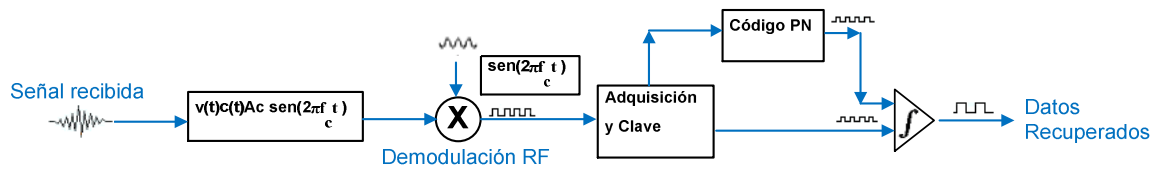
$$z(t) = \int_0^{T_b} V(t) dt \quad (\text{Ec 3.17})$$

$$z(t) = \pm T_b + \varepsilon$$

si  $z > 0$  entonces  $v'(t)$  es 1

si  $z < 0$  entonces  $v'(t)$  es -1

siendo  $v'(t)$  la señal recuperada e idéntica a  $v(t)$  del transmisor.



**Figura 3.14** Recuperación de datos originales.

NOTA: se puede ver que se para que lo anterior se cumpla, se supone un sincronismo perfecto al momento de multiplicar la señal que llega al receptor por la secuencia pseudoaleatoria. Este es uno de los problemas más complicados de resolver y solo se citarán algunas ideas al respecto:

### **Sincronismo**

El objetivo del sistema de sincronismo es encargarse de alinear la señal recibida, con la copia local de la secuencia de código que genera el receptor.

Este mecanismo tiene dos fases diferenciadas: el ajuste grueso o fase de adquisición propiamente dicha, y el ajuste fino o fase de seguimiento.

La dificultad del sincronismo radica en la corta duración del periodo de chip de las secuencias de código. En CDMA se requiere que el sincronismo funcione correctamente al menos a nivel de chip.

#### **3.4.4.2 Simulación y Análisis en MATLAB sobre un sistema DSSS en Banda Base**

##### ***Programa cap3\_cod1\_espectro\_PN.m (ver Anexos Cap 3)***

En este ejemplo se genera un sistema DSSS en banda base (no se realiza la simulación de la modulación PSK de la señal). Para el transmisor se genera un número finito de bits de datos conocidos, a través de una secuencia de Barker y de chips, a través de un generador de secuencia M. Por cada bit de datos se genera la secuencia M.

Se puede analizar y observar cómo se produce el ensanchamiento de una señal en el transmisor y su recuperación en el receptor.

Se analizará este sistema utilizando los siguientes casos de canal: Canal ideal, donde no se considera ruido e interferencia. Se observan a las señales en cada etapa del ensanchamiento y desensanchamiento.

El tiempo de bit  $t_b$  para la simulación es de 1 segundo y el tiempo de chip  $t_c$  es igual a  $t_b/N_c$ , siendo  $N_c = 63$  el número de chips por bit, entonces  $t_c = 1/63$  segundos. La ganancia de procesamiento  $G_p$  es igual a  $t_b/t_c$ . Entonces  $G_p = 63$ .

**Sistema DSSS.** Se simulan tanto las señales producidas en el transmisor como las señales recuperadas en el receptor.

El sistema consta de los siguientes bloques.

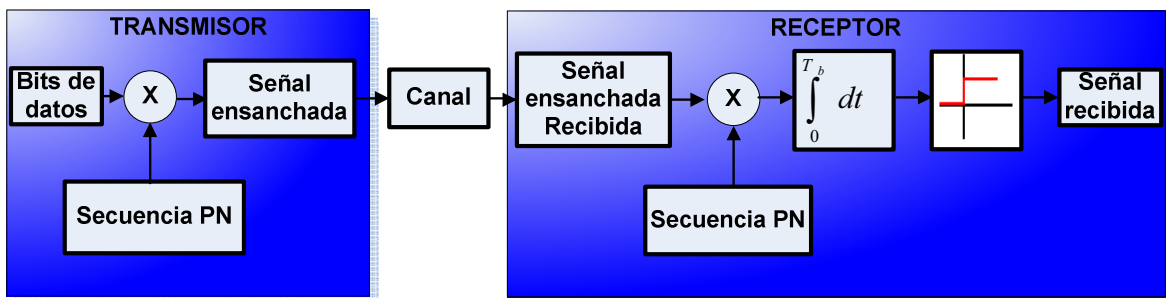


Figura 3.15 cap3\_cod1\_espectro\_PN.m, Sistema DSSS.

## **1. Implementación del Transmisor**

### **1.1 Generación de los bits de datos**

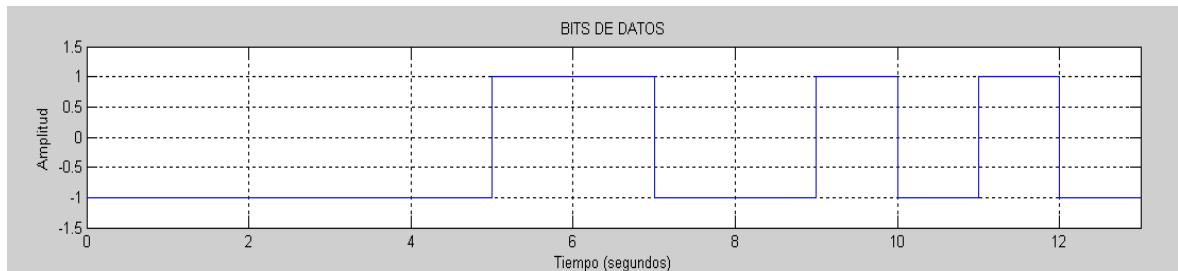
Se genera una secuencia de bits conocidos, a través de una secuencia de Barker de longitud 13. Esta secuencia es: [-1 -1 -1 -1 -1 1 1 -1 -1 1 -1 1 -1] en forma bipolar con lógica negativa.

#### **Resultados:**

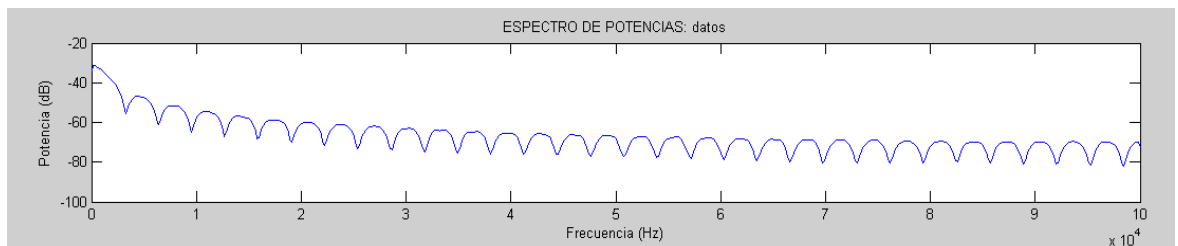
El total de datos generados es de 13 bits de forma bipolar. La secuencia generada siempre es la misma y se muestra a continuación:

```
>> dato
dato =
    -1    -1    -1    -1    -1     1     1    -1    -1     1    -1     1    -1
```

**Figura 3.16** Resultado de la generación de bits de datos.



**Figura 3.17** Gráfica de bits de datos.



**Figura 3.18** Espectro de potencia de datos

## **1.2 Generador de la Secuencia M**

La generación de la secuencia M se realiza con la función ***sec\_pn(polynomio).m*** implementada en el capítulo 2. Para esta secuencia se necesita ingresar el polinomio primitivo generador de la secuencia en forma de vector en el parámetro ***polynomio***. El polinomio generador implementado en la simulación es  $x^6 + x + 1$  o ***[1 0 0 0 0 1 1]*** de forma vectorial y se almacena en la variable ***polynomio***.

### **Resultados:**

Se genera el vector ***secuencia\_pn*** con longitud 63.

```

secuencia_pn =

Columns 1 through 17
   -1    1    1    1    1    1   -1   -1   -1   -1   -1   -1    1   -1    1   -1

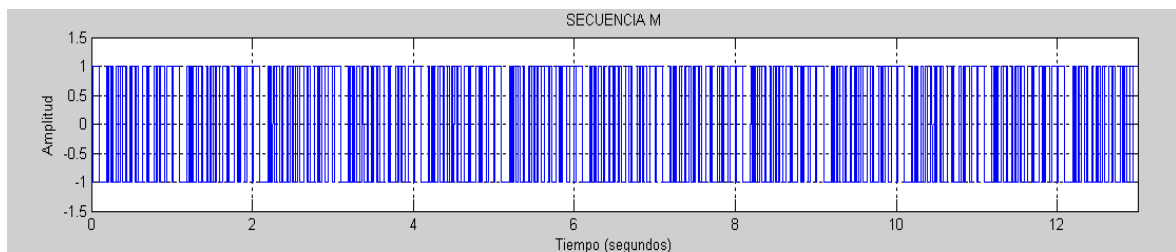
Columns 18 through 34
   -1   -1    1    1   -1   -1    1   -1   -1   -1    1   -1   -1    1   -1    1

Columns 35 through 51
   -1    1    1   -1   -1   -1    1    1    1   -1    1   -1   -1   -1   -1    1

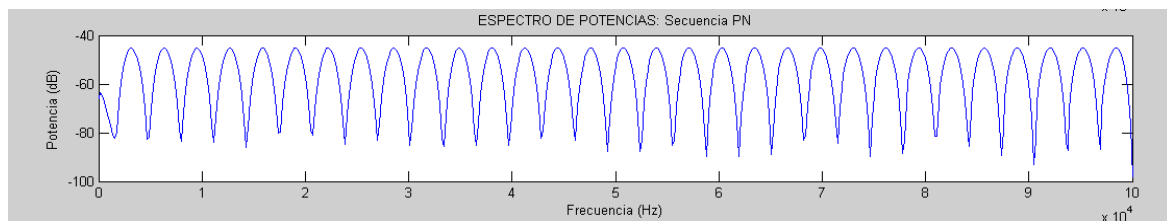
Columns 52 through 63
   -1    1   -1    1    1    1   -1   -1    1    1    1    1

```

**Figura 3.19** Resultado de la generación de la secuencia M.



**Figura 3.20** Gráfica de la secuencia M.



**Figura 3.21** Espectro de potencia de la secuencia M.

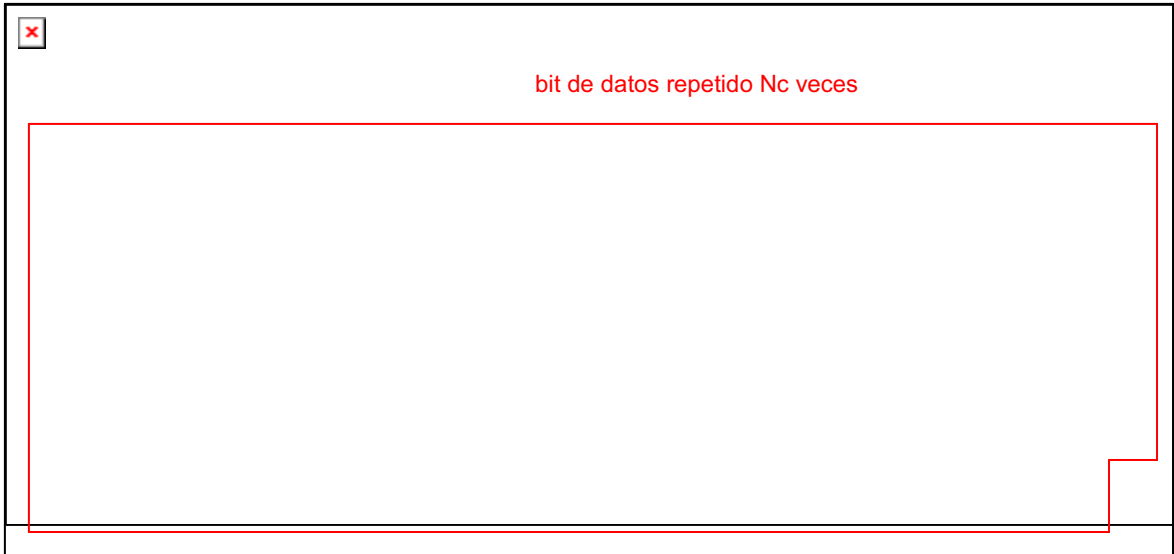
### **1.3 Multiplicador para obtener la señal Ensanchada**

Para realizar la multiplicación de la señal de datos con la secuencia M primero se necesita repetir los datos en la variable **dato**  $N_c$  veces, donde  $N_c$  es el número de chips por bit o la longitud de la secuencia M. Esto se almacenará en la variable **datos**.

#### **Resultados:**

En la variable **datos** se almacenarán el dato repetido  $N_c$  veces y se generarán  $13 \times 63 = 819$  valores:

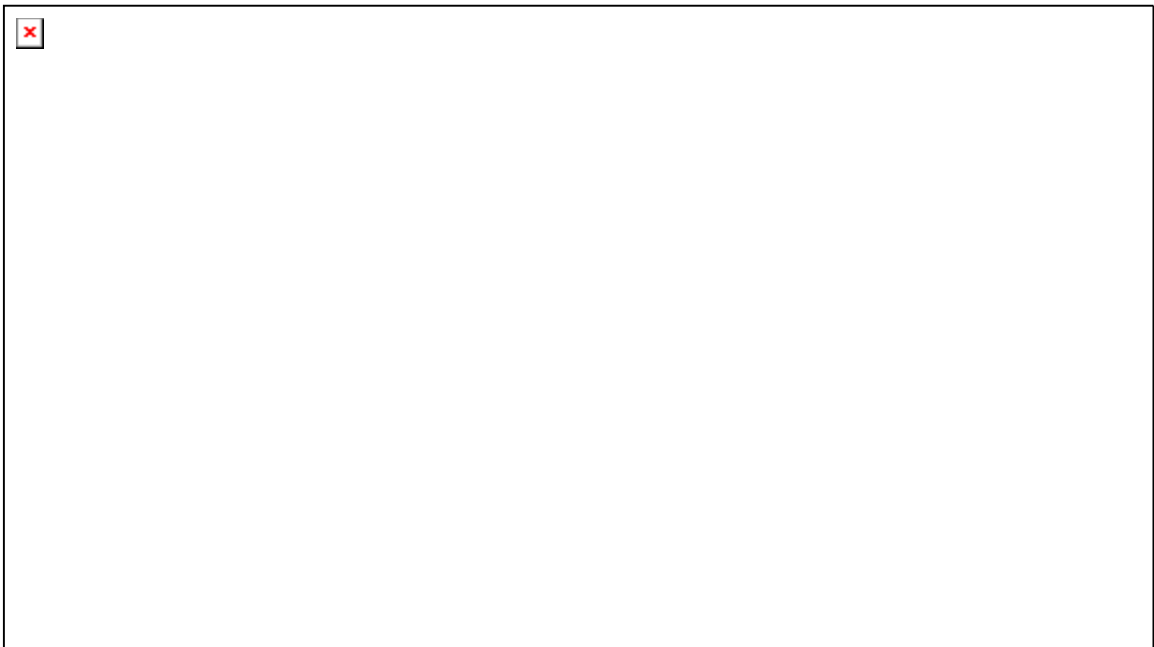




**Figura 3.22** Resultado de repetición de bits de datos  $N_c$  veces cada uno.

Luego, la secuencia  $M$  debe repetirse por cada bit para poder realizar la multiplicación de la señal de datos y la secuencia  $M$  y tener como resultado la señal ensanchada.

En la variable ***PN*** se almacenarán el dato repetido  $N_c$  veces y se generarán  $63 \times 13 = 819$  valores:



**Figura 3.23** Resultado de repetición de secuencia  $PN$  por cada bit.

Para obtener la señal ensanchada en MATLAB, basta con multiplicar los elementos correspondientes de los vectores datos y PN.

En la variable **senal\_ensanchada** se guarda el resultado de la multiplicación de los elementos correspondientes de los vectores datos y PN.

```
>> senal_ensanchada
senal_ensanchada =

Columns 1 through 16
    1    -1    -1    -1    -1    -1     1     1     1     1     1     1    -1     1    -1     1

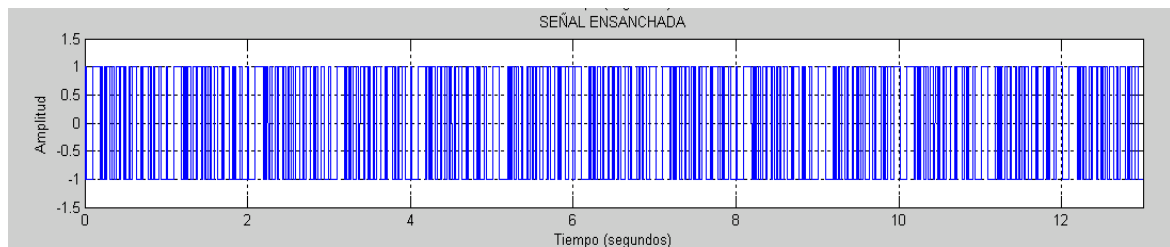
Columns 17 through 32
   -1     1     1    -1    -1     1     1    -1     1     1     1    -1     1     1    -1     1

Columns 33 through 48
   -1    -1     1    -1    -1     1     1     1    -1    -1    -1     1    -1     1     1     1

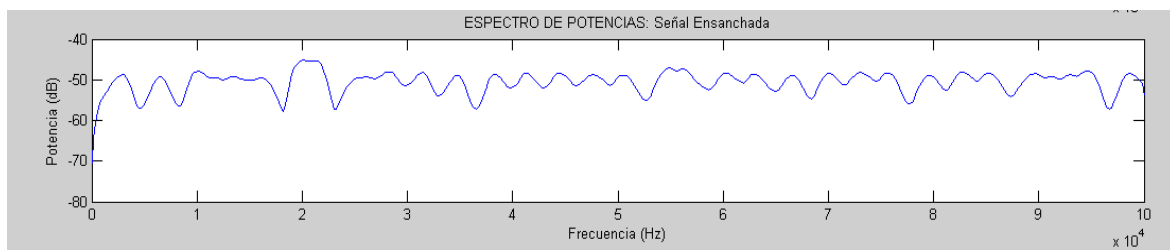
Columns 49 through 64
     1    -1    -1     1    -1     1    -1    -1    -1     1     1    -1    -1    -1    -1     1

Columns 65 through 80
   -1    -1    -1    -1    -1     1     1     1     1     1     1    -1     1    -1     1    -1
```

**Figura 3.24** Resultado de la multiplicación de bits de datos con secuencia PN.

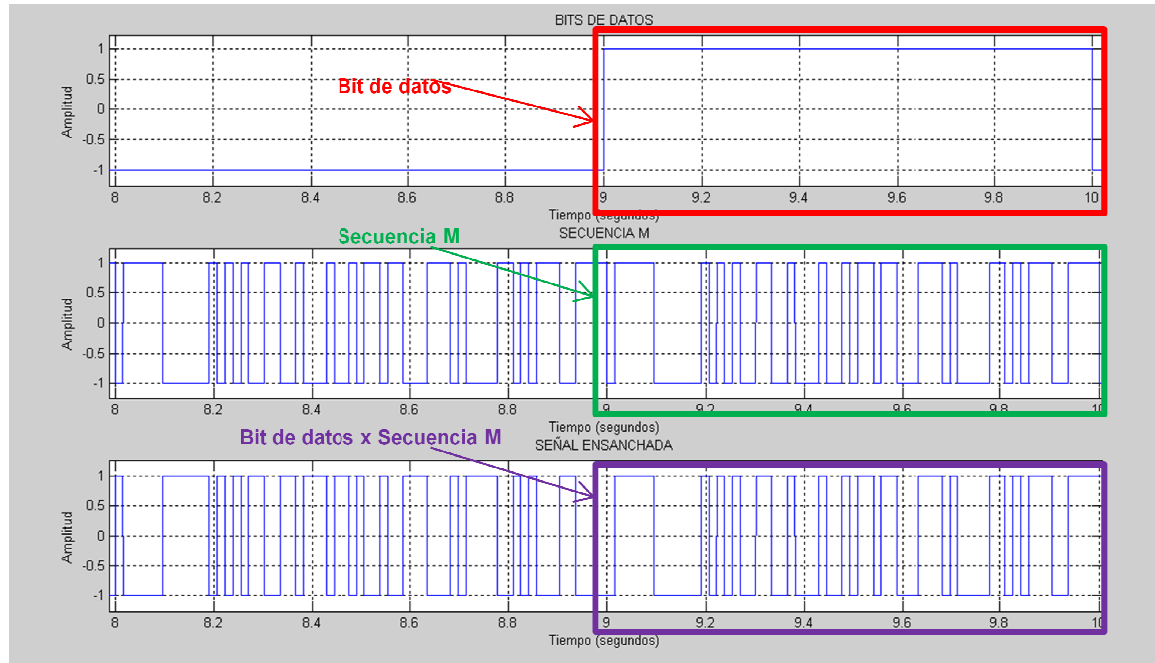


**Figura 3.25** Gráfica de la multiplicación de bits de datos con secuencia PN.



**Figura 3.26** Espectro de potencia de la señal ensanchada.

## Gráficas de las señales del Transmisor

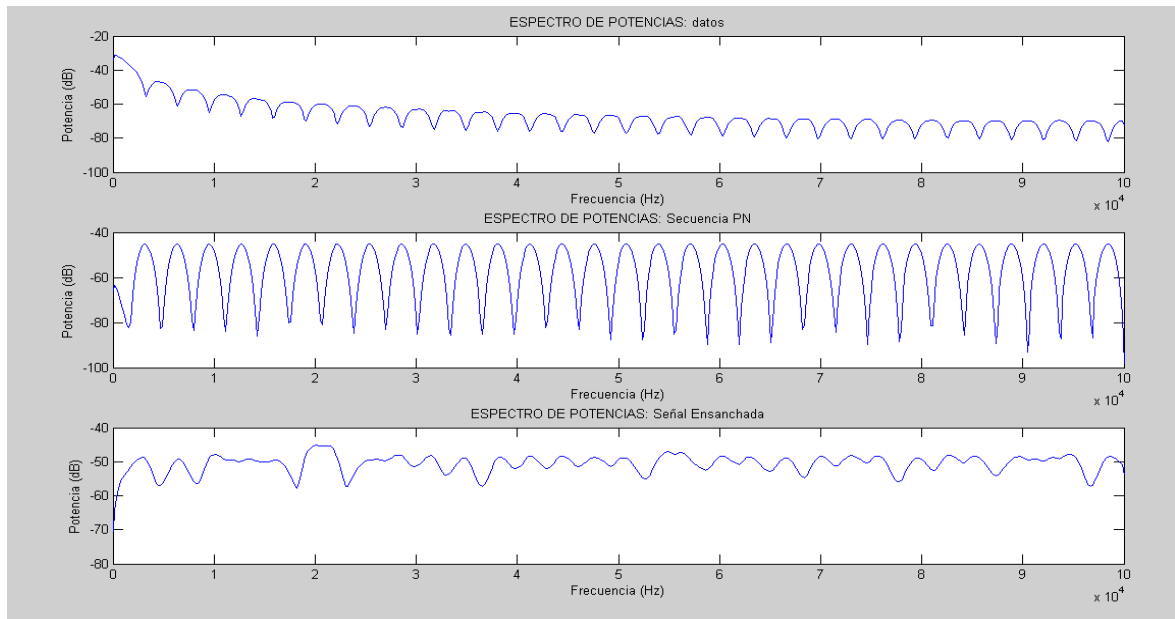


**Figura 3.27** Visualización de las señales obtenidas del transmisor con la simulación.

### Comentarios

Se puede observar que el tiempo de chips es mucho menor que el tiempo de bits. Por cada bit de datos se produce la secuencia PN, es decir que por cada tiempo de bit hay 63 chips.

## **Gráficas de los espectros de potencia del Transmisor**



**Figura 3.28** Visualización de los espectros de potencia del transmisor.

### **Comentarios**

Se puede observar que el espectro de potencia de la señal de datos es menor que el espectro de la secuencia PN. En la gráfica de la secuencia PN se observa que hay ciertas periodicidades. El espectro de potencia de la señal ensanchada tiene una apariencia de ruido.

### **2. Canal**

En este caso el canal es ideal; por lo tanto la señal ensanchada no será contaminada con el ruido propio del canal ni con interferencia.

### **3 Implementación del Receptor**

#### **3.1 Recepción de la señal ensanchada y secuencia PN**

Con las consideraciones de canal ideal, la señal ensanchada es la misma que se guardó en la variable ***senal\_ensanchada***.

De igual manera, la secuencia PN debe ser la misma que se usó en el transmisor para poder recuperar la señal original. El vector que se necesita se encuentra guardado en la variable **PN**.

### **3.2 Multiplicador para desensanchar la señal**

Para obtener la señal desensanchada en Matlab, basta con multiplicar los elementos correspondientes de los vectores **senal\_ensanchada** y **PN**.



**Figura 3.29** Gráfica de la señal ensanchada por secuencia M.

### **Resultados:**

El resultado se almacena en la variable **senal\_recuperada**.

```
>> senal_recuperada
senal_recuperada =
Columns 1 through 16
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
Columns 17 through 32
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
Columns 33 through 48
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
Columns 49 through 64
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
Columns 65 through 80
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

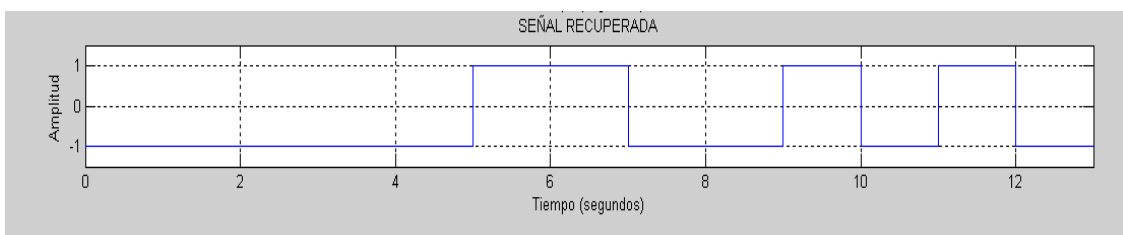
**Figura 3.30** Señal Recuperada.

### **3.3 Integrador y toma de decisión**

Se puede observar que la señal recuperada es la misma que la señal original.

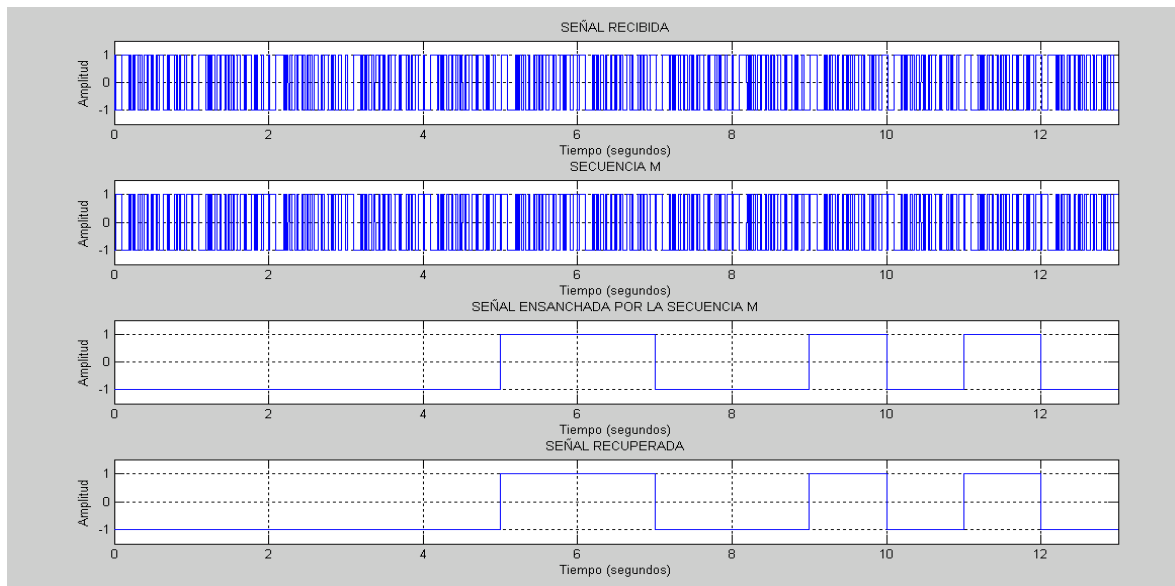
```
>> dato_recibido
dato_recibido =
-1 -1 -1 -1 -1 1 1 -1 -1 1 -1 1 -1
```

**Figura 3.31** Datos recuperados.



**Figura 3.32** Gráfica de la señal recuperada.

### **Gráfica de las señales del Receptor**



**Figura 3.33** Visualización de las señales obtenidas del receptor con la simulación.

### **Comentarios**

Se puede observar que se utiliza la misma secuencia M implementada en el transmisor y que la señal recuperada es idéntica a la señal original del transmisor.

### **Conclusión:**

La señal recibida es igual a la señal de origen. Por lo tanto, se ha conseguido enviar datos a través de la Técnica de Espectro Ensanchado.

#### **3.4.4.3 Simulación de un Sistema DS-SS con Canal Ideal en Simulink**

##### ***Modelo cap3\_modelo1\_DSSS.mdl (ver Anexos Cap 3)***

A continuación se realiza un ejemplo del transmisor y receptor en un sistema DSSS con canal ideal:

Se modula a la señal utilizando modulación BPSK (Binary phase shift keying). La señal de transmisión tiene una velocidad de 1 Mbps. Se utiliza la técnica de **secuencia directa** con un generador de código PN, con una tasa de transmisión de 4 Mbps para ensanchar el espectro de la señal original:

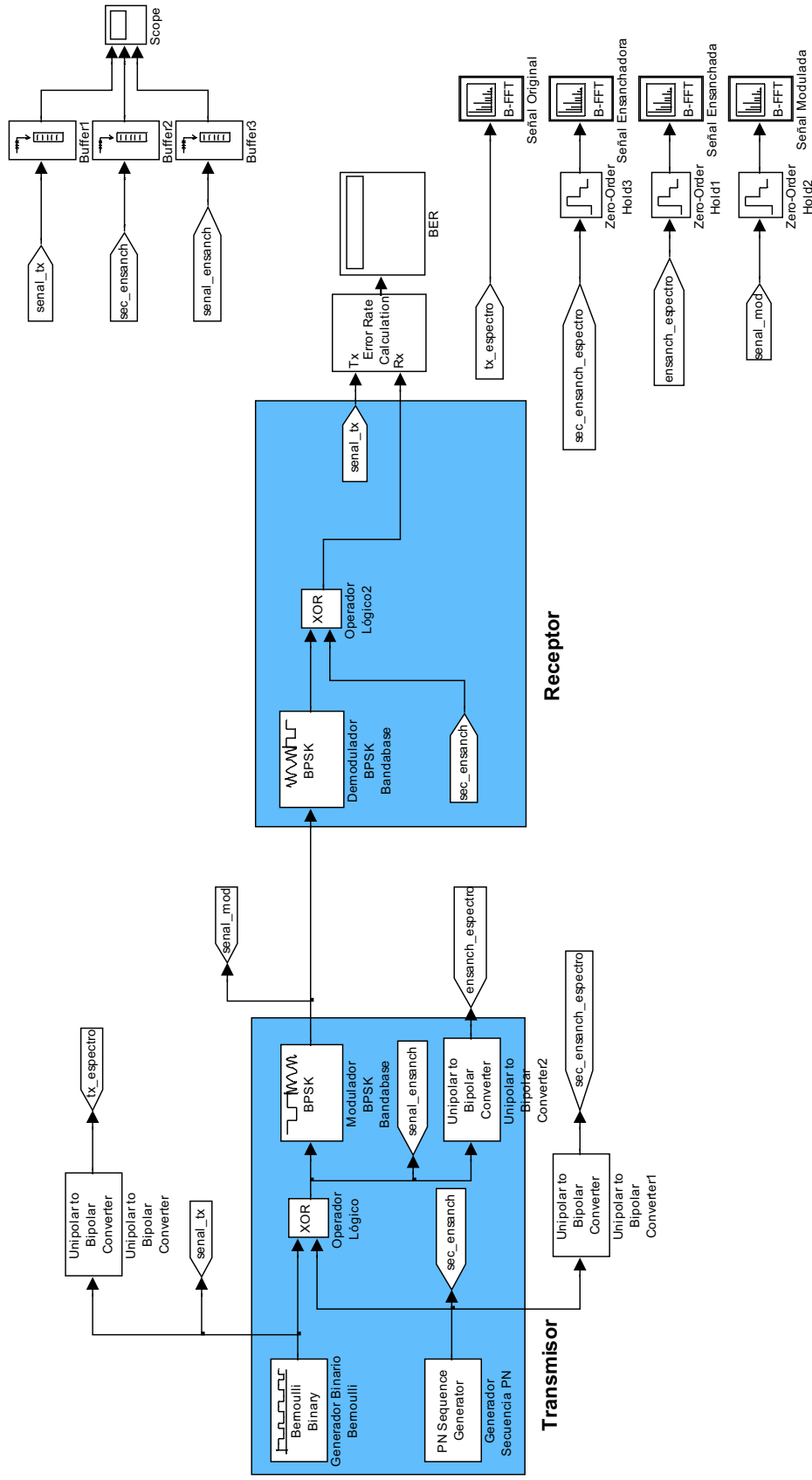


Figura 3.34 cap3\_modelo1\_DSSS.mdl, Sistema DSSS con Simulink



## Transmisor

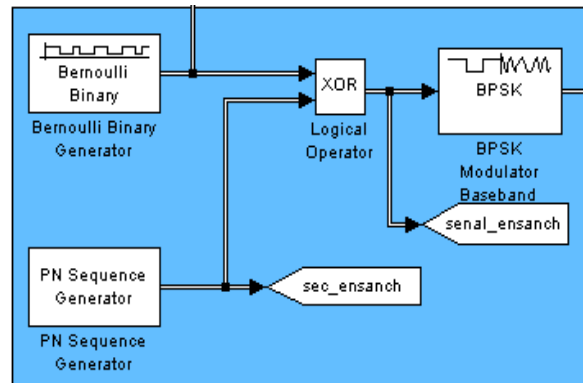


Figura 3.35 Transmisor DSSS en Simulink

## Generación de Datos

El transmisor consta de un generador de datos binarios de Bernoulli. Este generador está configurado con una velocidad de 1 Mbps, y genera tramas de 1 bit cada una. El tipo de salida es booleanas (0s y 1s).

El bloque Bernoulli Binary Generator genera números binarios aleatorios utilizando una distribución de Bernoulli. La distribución de Bernoulli, con parámetro  $p$ , produce ceros (0s) con probabilidad  $p$  y unos (1s) con probabilidad  $1-p$ . Esta distribución tiene valor de media  $1-p$  y varianza  $p(1-p)$ .

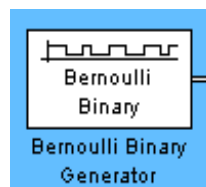


Figura 3.36 Bloque Bernoulli Binary Generator

## Resultados:

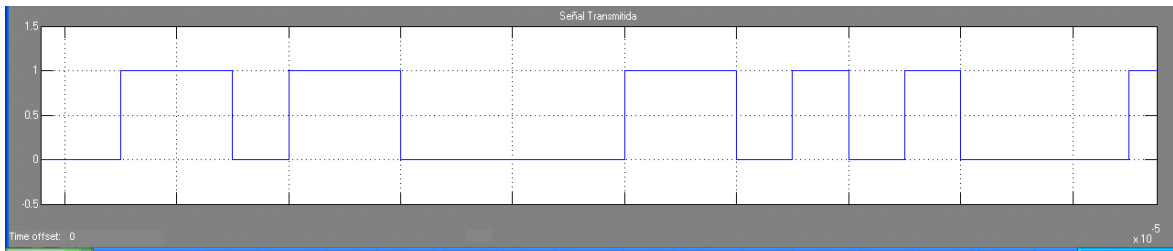


Figura 3.37 Señales de Datos Originales del Transmisor DSSS.

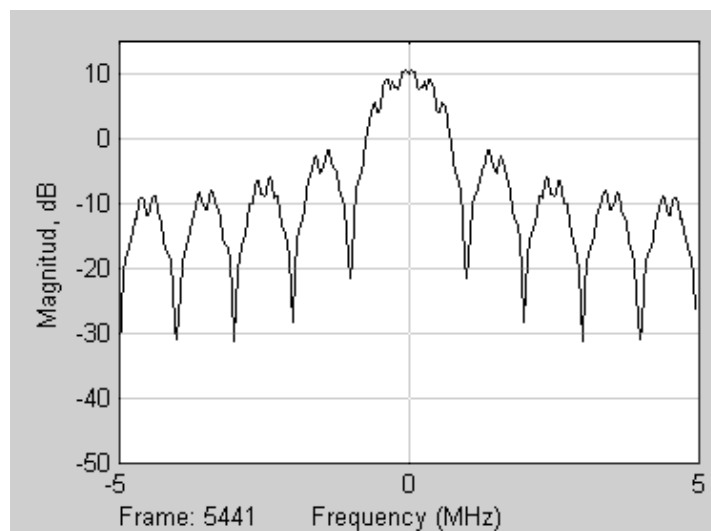


Figura 3.38 Espectro de Potencia de Señal de datos.

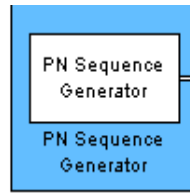
## Comentario

Se puede observar la señal de datos originales de forma unipolar. El espectro de la señal resulta una función sinc.

## Código Expansor

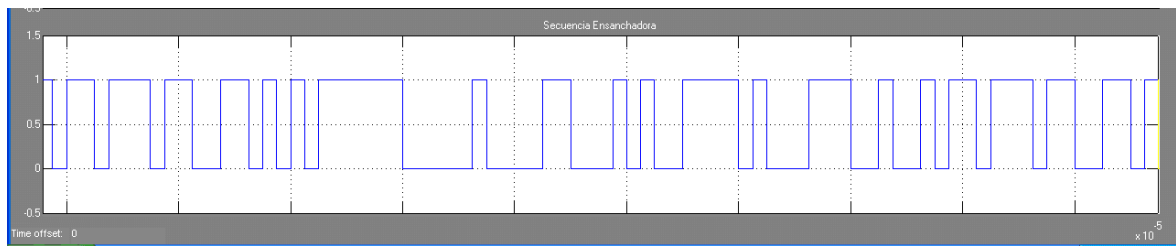
Para generar el código expansor, se utiliza el bloque de generador de código PN. Se utiliza el polinomio generador **primitivo**  $x^6 + x + 1$ .

El generador tiene una velocidad de transmisión de 4 Mbps.

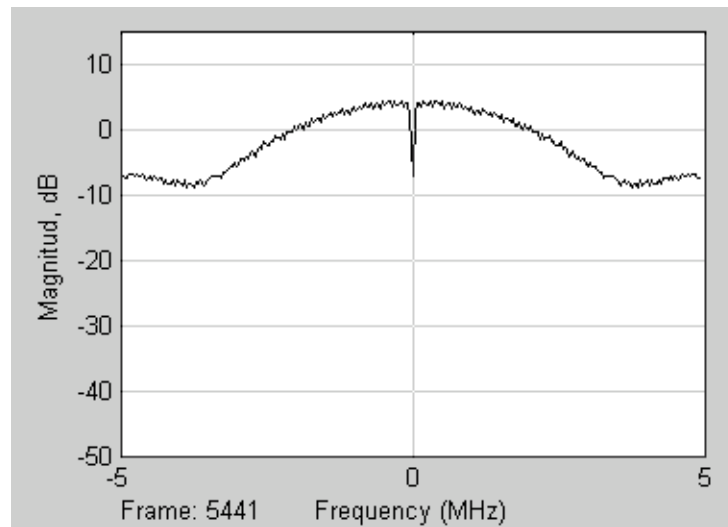


**Figura 3.39** Bloque PN Sequence Generator.

### Resultados:



**Figura 3.40** Señales ensanchadora del Transmisor DSSS.



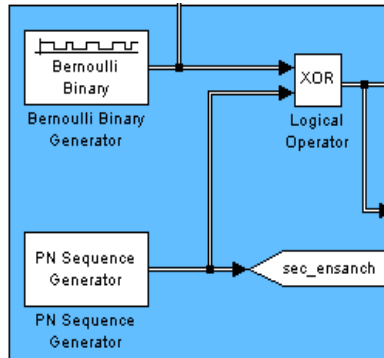
**Figura 3.41** Espectro de Potencia de Secuencia ensanchadora.

### Comentario

Se puede observar que la secuencia ensanchadora posee un tiempo de chip menor que la secuencia original de datos y por tanto una mayor espectro de potencia.

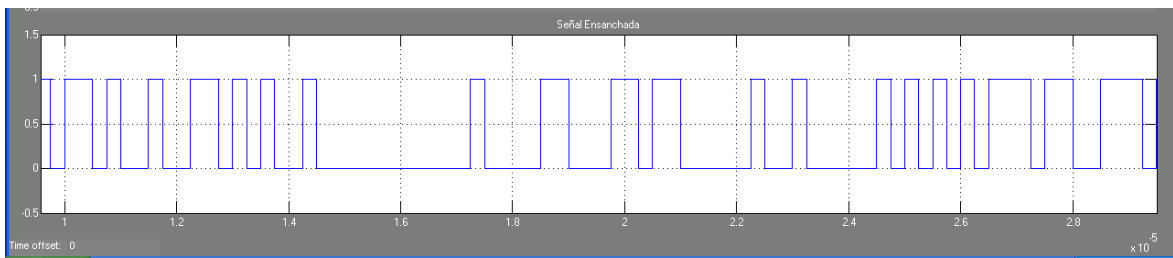
### Ensanchamiento de los datos

Para realizar el ensanchamiento, se realiza la operación lógica XOR entre el resultado del generador de datos con el resultado del generador de secuencia PN.

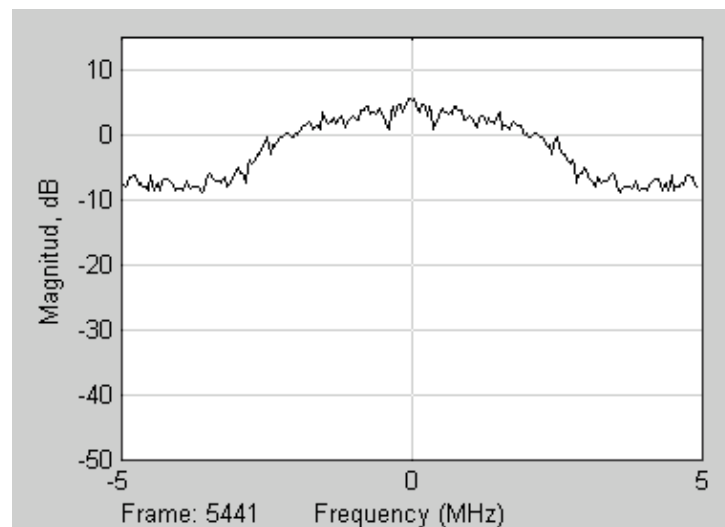


**Figura 3.42** Ensanchamiento de la señal de datos.

### Resultados:



**Figura 3.43** Señales ensanchada del Transmisor DSSS.



**Figura 3.44** Espectro de Potencia de Señal ensanchada.

## Comentario

Se puede observar que la señal ensanchada tiene el mismo tiempo de chip que la secuencia ensanchadora y por tanto es menor que el tiempo de bit de los datos originales. También se puede observar que el espectro de potencia ha aumentado en ancho, 4 veces en comparación del espectro de potencia de los datos originales debido a que el tiempo de chip es menor que el tiempo de bit.

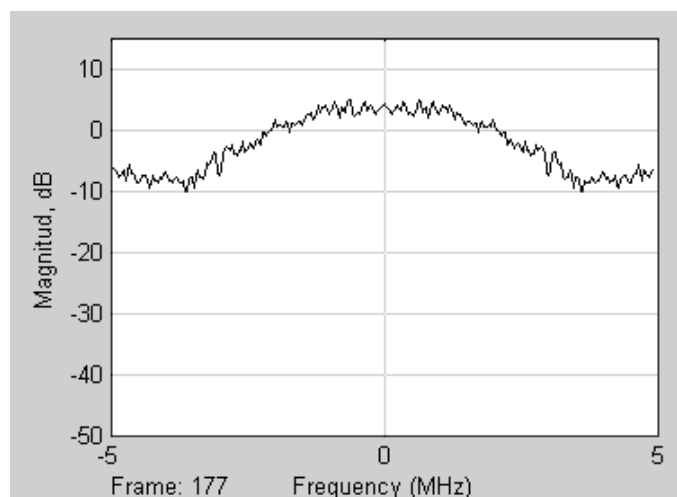
## Modulación

Luego se procede a realizar la modulación. Como se mencionó anteriormente se realiza modulación BPSK, muy utilizada con sistemas DSSS.



**Figura 3.45** Bloque BPSK Modulator Baseband.

## Resultados:



**Figura 3.46** Espectro de Potencia de Señal modulada.

## Comentario

Se puede observar que el modulador BPSK, como con el espectro de la señal tiene una Gp de 4.

**Canal:** en este caso, se considera canal *ideal* y por tanto no hay cambios entre la señal enviada y la señal recibida.

## Receptor

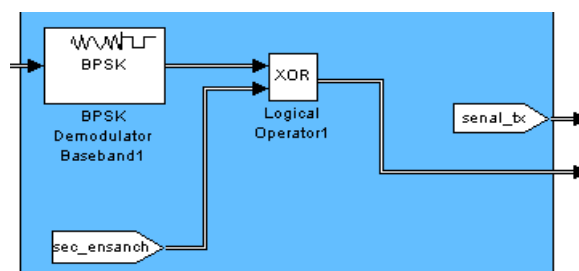


Figura 3.47 Receptor DSSS en Simulink

Para demodular la señal recibida, se utiliza el **bloque BPSK demodulator baseband**.

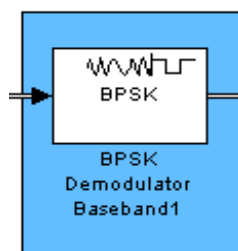


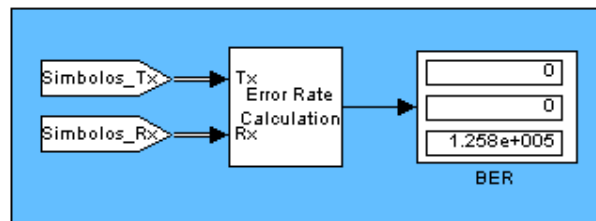
Figura 3.48 Bloque BPSK Demodulator Baseband.

Para **desensanchar** los datos originales, se necesita un generador de secuencia PN **idéntico** al del transmisor y además se considera que está totalmente sincronizado. En la **simulación** no hay ningún problema ya que se utiliza el mismo generador utilizado en el transmisor. En la **práctica** habría que considerar un método para conseguir dicha sincronización precisa.

Para realizar el desensanchamiento de la señal, se utiliza la operación lógica XOR entre la señal recuperada del demodulador BPSK con la secuencia PN. Con esto se recuperan los datos originales.

### BER

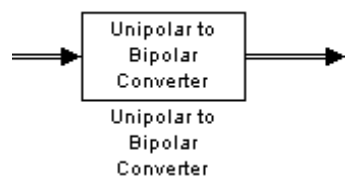
Con el **bloque Error rate calculation** se calcula el BER de la transmisión y recepción. No existe retardo en la simulación.



**Figura 3.49** Bloque Error Rate Calculation.

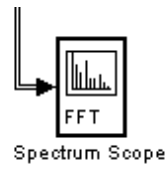
### Espectros de Potencia

Para la visualización de los espectros de las señales primeramente se utilizó el bloque **Unipolar to Bipolar Converter** para transformar a las señales de forma unipolar (1s y 0s) a forma bipolar con lógica negativa (-1s y 1s respectivamente). Con esto se logra eliminar las componentes DC que se producen en una señal unipolar.



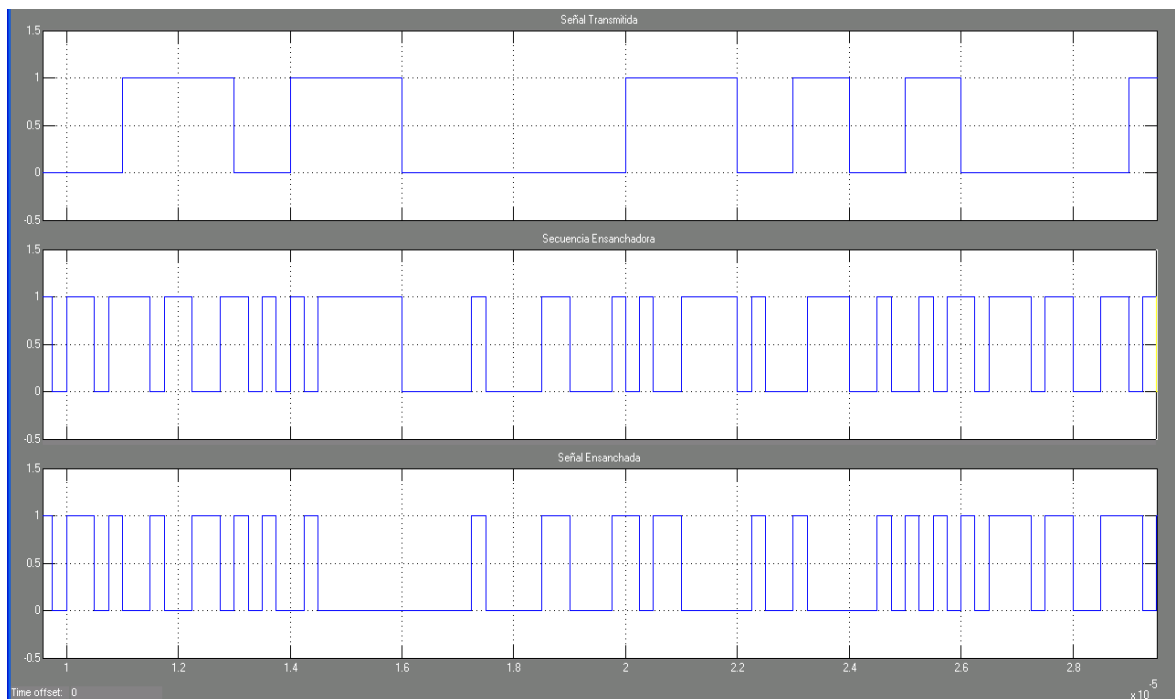
**Figura 3.50** Bloque Unipolar to Bipolar Converter.

Luego, se utilizó el **bloque Spectrum Scope**:



**Figura 3.51** Bloque Spectrum Scope.

Las principales señales del sistema se pueden observar en el osciloscopio de la simulación:



**Figura 3.52** Señales del Transmisor DSSS.



### 3.4.4.4 Análisis Matemático del Comportamiento de un Sistema DS-SS frente al Ruido<sup>69</sup>

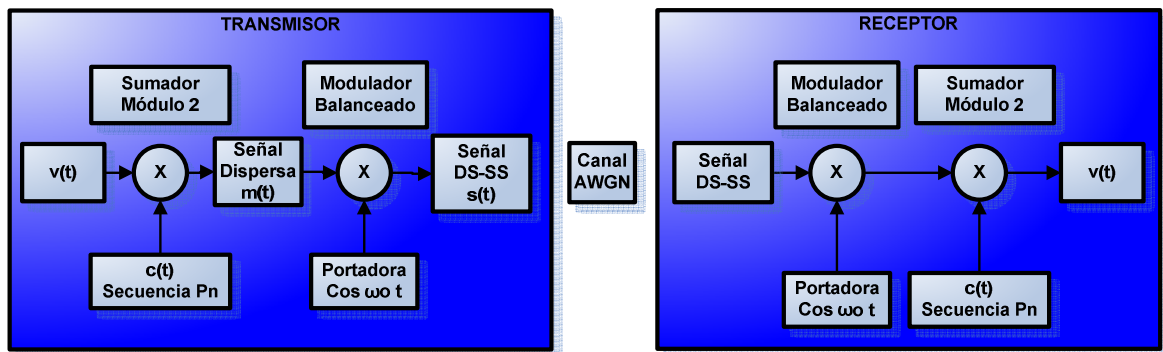


Figura 3.53 Sistema DSSS con canal AWGN.

#### Procedimiento:

Un mensaje  $v(t)$  es multiplicado por una secuencia PN  $c(t)$  y luego modulado de tal manera que:

$$v(t)c(t)\text{sen}(2\pi f_o t) \quad (\text{Ec 3.18})$$

Esta señal se contamina **con un canal de tipo aditivo: ruido gaussiano blanco** de media cero:

$$v(t)c(t)\text{sen}(2\pi f_o t) + n_i(t) \quad (\text{Ec 3.19})$$

Como se conoce, para recuperar la información en el receptor, se deben realizar los siguientes pasos:

**Demodulación:** Se multiplica por la misma portadora empleada en el transmisor. Así se obtiene:

<sup>69</sup> Contemporary Communication Systems using MATLAB, Proakis, John; Salehi, Masoud, p. 395-397.

$$[v(t)c(t)\text{sen}(2\pi f_o t) + n_i(t)]\text{sen}(2\pi f_o t) = v(t)c(t)\text{sen}^2(2\pi f_o t) + n_i(t)\text{sen}(2\pi f_o t) \quad (\text{Ec 3.20})$$

Por la identidad trigonométrica:  $\text{sen}^2(2\pi f_o t) = \frac{1 - \cos(4\pi f_o t)}{2}$

Se obtiene:

$$\begin{aligned} &= v(t)c(t)\frac{1 - \cos(4\pi f_o t)}{2} + n_i(t)\text{sen}(2\pi f_o t) \\ &= 0.5v(t)c(t) - 0.5v(t)c(t)\cos(4\pi f_o t) + n_i(t)\text{sen}(2\pi f_o t) \end{aligned} \quad (\text{Ec 3.21})$$

Como se puede ver, se obtiene el término  $v(t) \times c(t)$  que se desea recuperar en esta etapa y además dos términos en la frecuencia de la portadora. Para dejar solo el término en baja frecuencias, se emplea un **filtro pasabanda**, luego de lo cual se obtiene:

$$0.5v(t)c(t) \quad (\text{Ec 3.22})$$

**Desenchamamiento:** Para recuperar  $v(t)$  se multiplica a esta señal por la misma secuencia  $c(t)$  empleada en el transmisor.

$$\begin{aligned} 0.5v(t)c(t) \times c(t) &= K_1 v(t)c(t)c(t) = K_1 v(t)K_2 \\ &= v(t)K \end{aligned} \quad (\text{Ec 3.23})$$

Igual que el caso anterior, para que esto se dé, el **tipo de secuencias** debe ser tal que la operación de **autocorrelación** debe dar una **constante**.

Como se observa se tiene el mismo mensaje original, con una variación en amplitud.

### 3.4.4.5 Simulación de un sistema DSSS con Ruido en Simulink

**Modelo cap3\_modelo2\_DSSS\_ruido.mdl (ver Anexos Cap 3)**

A continuación se realiza un ejemplo del transmisor y receptor en un sistema DSSS con un canal con **ruido gaussiano blanco**:

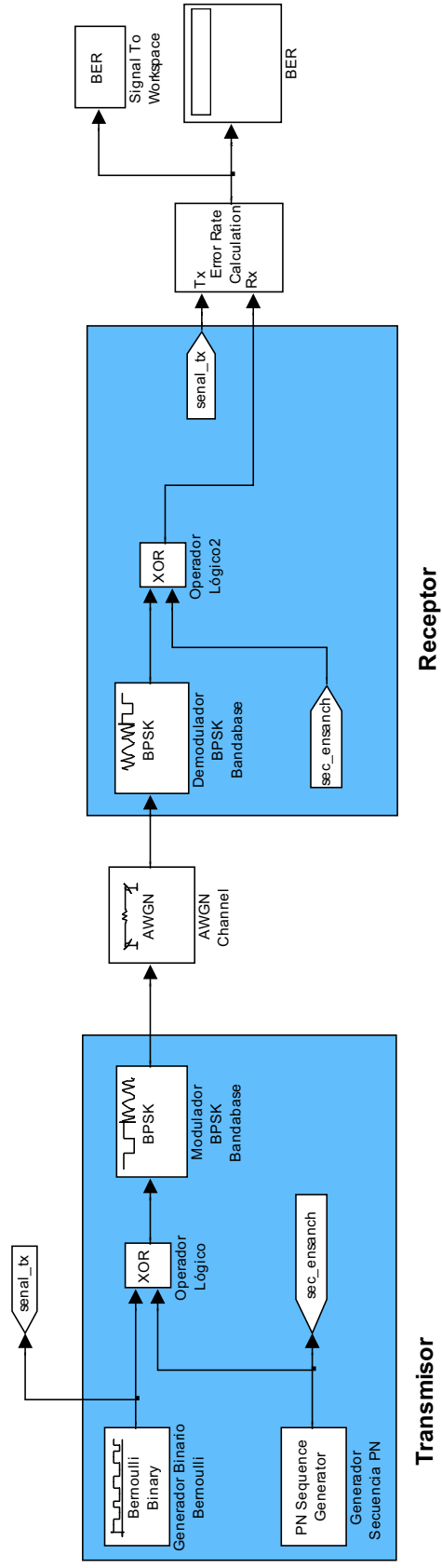
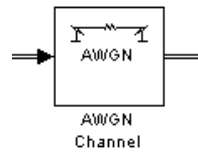


Figura 3.54 cap3\_modelo2\_DSSS\_ruido.mdl, Sistema DSSS con canal AWGN en Simulink.

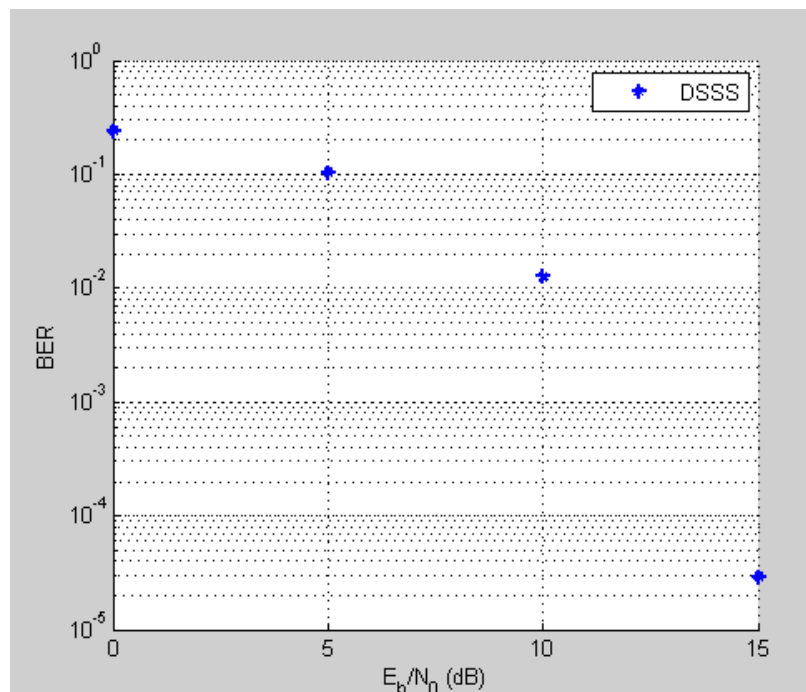
Para simular un canal con ruido gaussiano blanco, se utiliza el **bloque AWGN Channel**. Este bloque añade ruido gaussiano blanco.



**Figura 3.55** Bloque AWGN Channel.

Para la visualización del BER de la señal DSSS se utiliza la herramienta GUI de MATLAB BERTool.

### **Resultados:**



**Figura 3.56** BER VS  $E_b/N_0$  obtenido con BER Tool.

### **Comentarios**

Se puede observar que el Sistema DSSS presenta cierta resistencia al ruido debido a las características mencionadas anteriormente. El sistema DSSS no elimina por completo los efectos del ruido sobre la información transmitida, pero

como se puede observar, el BER generado con una tasa de transmisión alta en la simulación es bastante bajo.

Se debe considerar que no se está utilizando codificación de canal para mitigar los efectos del ruido sobre un canal inalámbrico.

### 3.4.5 APLICACIONES DE SISTEMAS DS-SS

#### 3.4.5.1 Resistencia para Interferencia y Jammers Hostiles

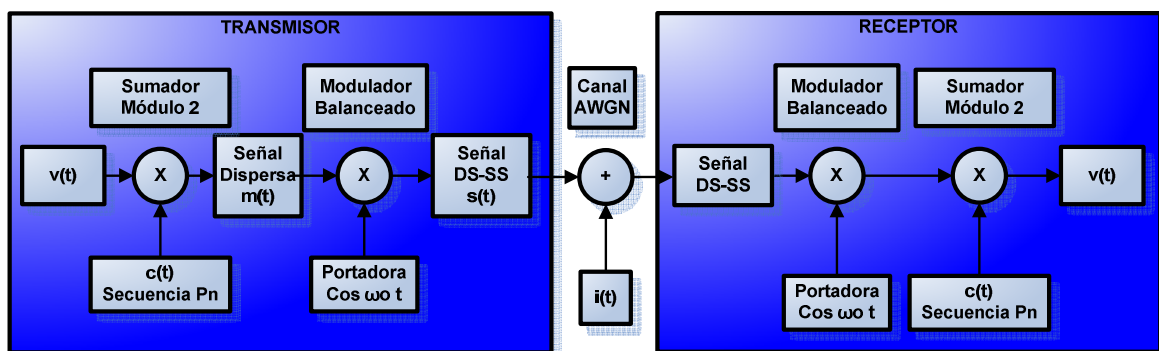


Figura 3.57 Sistema DSSS con interferencia.

Suponiendo que en vez del ruido gaussiano blanco, en el canal existe una señal de interferencia  $i(t)$ . Se conoce que la señal DS-SS es igual a:

$$s(t) = v(t)c(t) \cos(2\pi f_o t)$$

Esta señal se contamina **con una señal de interferencia** cuya potencia es mayor:

$$v(t)c(t) \sin(2\pi f_o t) + i(t) \quad (\text{Ec 3.24})$$

Como se conoce, para recuperar la información en el receptor, se deben realizar los siguientes pasos:

**Demodulación:** Se multiplica por la misma portadora empleada en el transmisor. Así se obtiene:

$$[v(t)c(t)\text{sen}(2\pi f_o t) + i(t)]\text{sen}(2\pi f_o t) = v(t)c(t)\text{sen}^2(2\pi f_o t) + i(t)\text{sen}(2\pi f_o t) \quad (\text{Ec 3.25})$$

Por la identidad trigonométrica:  $\text{sen}^2(2\pi f_o t) = \frac{1 - \cos(4\pi f_o t)}{2}$

Se obtiene:

$$\begin{aligned} &= v(t)c(t)\frac{1 - \cos(4\pi f_o t)}{2} + i(t)\text{sen}(2\pi f_o t) \\ &= 0.5v(t)c(t) - 0.5v(t)c(t)\cos(4\pi f_o t) + i(t)\text{sen}(2\pi f_o t) \end{aligned} \quad (\text{Ec 3.26})$$

Como se puede ver, se obtiene el término  $v(t) \times c(t)$  que se desea recuperar en esta etapa y además dos términos en la frecuencia de la portadora. Para dejar solo el término en baja frecuencias, se emplea un **filtro pasa banda**, luego de lo cual se obtiene:

$$0.5v(t)c(t) \quad (\text{Ec 3.27})$$

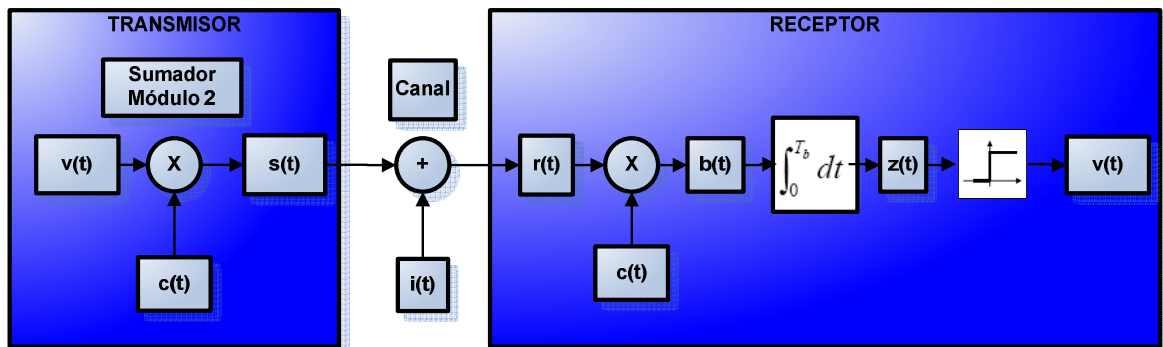
**Desenchamamiento:** Para recuperar  $v(t)$  se multiplica a esta señal por la misma secuencia  $c(t)$  empleada en el transmisor.

$$\begin{aligned} 0.5v(t)c(t) \times c(t) &= K_1 v(t)c(t)c(t) = K_1 v(t)K_2 \\ &= v(t)K \end{aligned} \quad (\text{Ec 3.28})$$

Igual que el caso con ruido, para que esto se dé, el **tipo de secuencias** debe ser tal que la operación de **autocorrelación** debe dar una **constante**.

Como se observa se tiene el mismo mensaje original, con una variación en amplitud. Como con el ruido, el sistema DSSS presenta una gran ventaja frente a la señal de interferencia. Uno de los problemas que se puede presentar es la presencia de demasiadas señales de interferencia que se sumarían a la recepción de la señal ensanchada y dificultarían la recuperación de la señal ensanchada.

Para analizar de forma directa la resistencia contra jammers hostiles, no se toma en cuenta la etapa de modulación digital:<sup>70</sup>



**Figura 3.58** Sistema DSSS sin modulación digital y con interferencia.

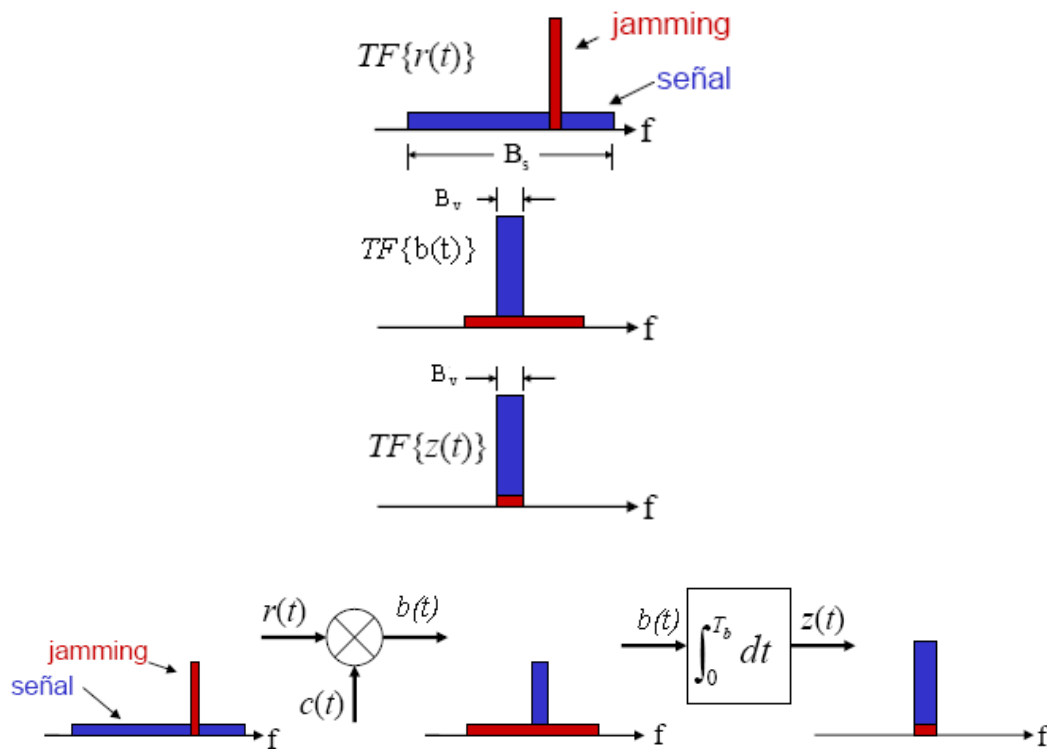
Al receptor le llega la señal:

$$r(t) = v(t) c(t) + i(t) \quad (\text{Ec 3.29})$$

Para realizar el desensanchado de la señal se realiza la operación:

$$\begin{aligned} b(t) &= v(t) c^2(t) + i(t) c(t) \\ &= v(t) + i(t) c(t) \end{aligned} \quad (\text{Ec 3.30})$$

<sup>70</sup> Contemporary Communication Systems using MATLAB, Proakis, John; Salehi, Masoud, p. 395-397.



**Figura 3.59** Recuperación de señal ensanchada contra interferencia.

$$r(t) = v(t)c(t) + i(t)$$

$$b(t) = v(t)c^2(t) + i(t)c(t)$$

$$b(t) = v(t) + i(t)c(t)$$

(Ec 3.31)

$$z(t) = \int_0^{T_b} v(t)dt + \int_0^{T_b} i(t)c(t)dt$$

$$z(t) = \pm T_b + \varepsilon$$

$i(t)$  se ensancha debido a la multiplicación por la secuencia  $c(t)$ .

### 3.4.5.2 Simulación de un sistema DSSS con Interferencia en Simulink

**Modelo cap3\_modelo3\_DSSS\_interferencia.mdl (ver Anexos Cap 3)**

A continuación se realiza un ejemplo del transmisor y receptor en un sistema DSSS con una señal de interferencia DSSS con un generador de secuencia PN diferente:



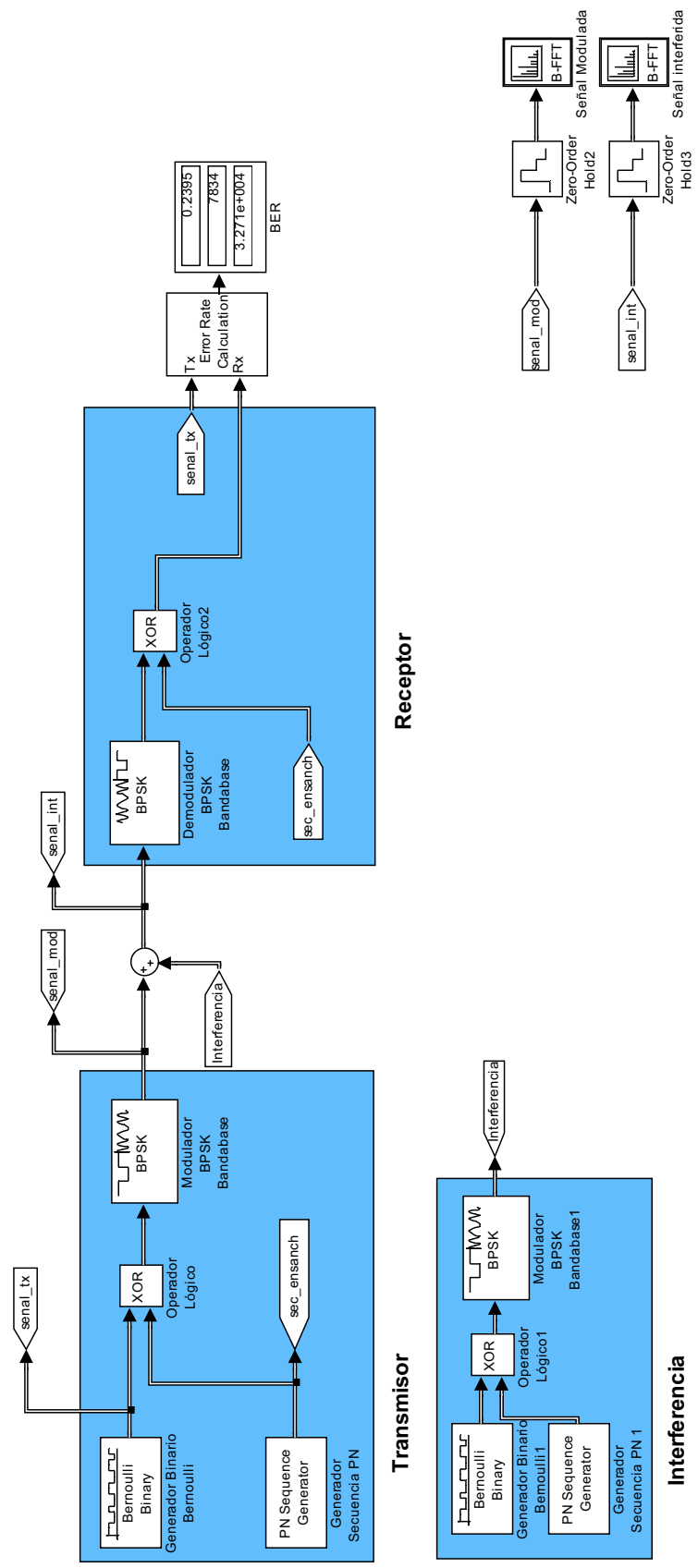
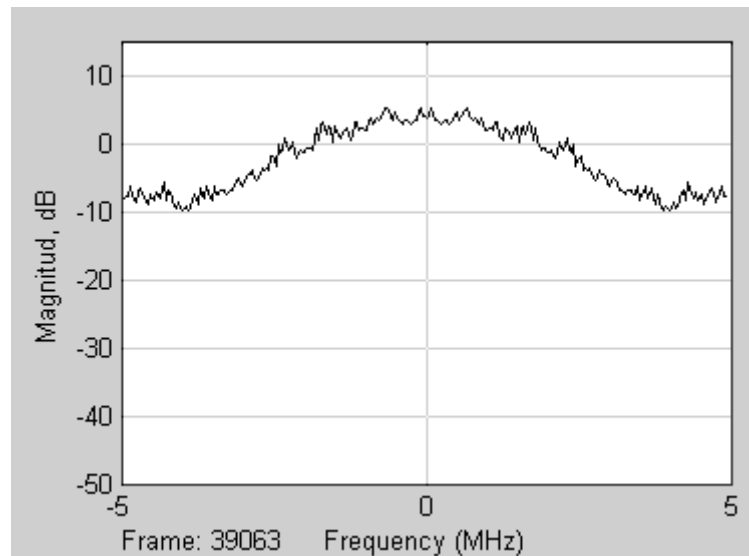
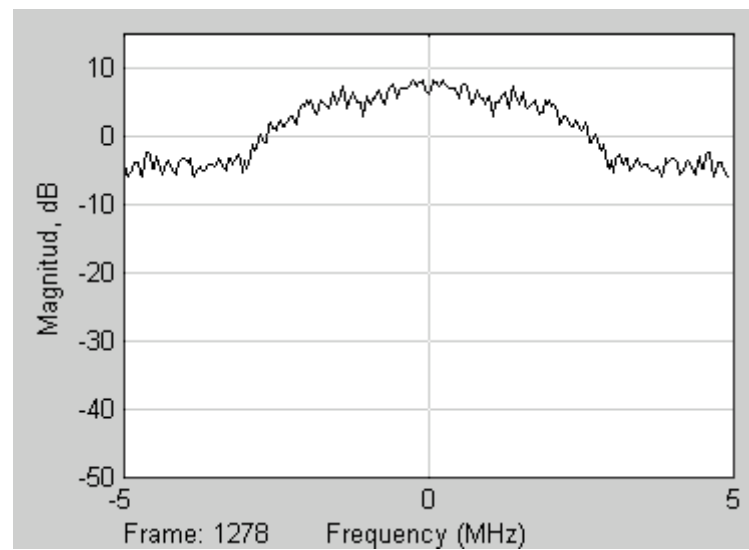


Figura 3.60 cap3\_modelo3\_DSSS\_interferencia.mdl, Sistema DSSS con interferencia de características similares.



**Figura 3.61** Espectro de Potencia de Señal Modulada.



**Figura 3.62** Espectro de Potencia de Señal Interferida.

Como se observa en la figura del sistema, el BER de la señal es 0.2537 con un generador de secuencia  $x^6 + x^4 + x^3 + x + 1$ , [1 0 1 1 0 1 1]. Se puede observar que el espectro de la señal interferida es el resultado de la suma de los dos espectros (señal de datos y señal interferente).

Con dos sistemas de interferencia se obtiene un BER de 0.3175 con un generador de secuencia  $x^6 + x^5 + 1$ , [1 1 0 0 0 0 1].

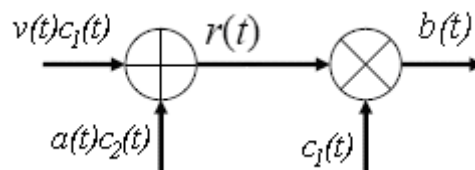
Con 3 sistemas de interferencia se obtiene un BER de 0.3409 con un generador de secuencia  $x^6 + x^5 + x^2 + x + 1$ , [1 1 0 0 1 1 1]. Como se observa, existe cierto número de errores que se incrementa con el número de señales interferentes.

### 3.4.5.3 Privacidad y Detectabilidad

La señal transmitida solo puede ser desensanchada si el código  $c(t)$  es conocido en el receptor. Siendo:

$$r(t) = v(t)c_1(t) \quad (\text{Ec 3.3})$$

Añadiendo otro usuario al sistema cuya señal ensanchada sea igual a  $a(t)c_2(t)$ , su efecto será aditivo:



**Figura 3.63** Recuperación con señal de jamming.

$$r(t) = v(t)c_1(t) + a(t)c_2(t)$$

$$b(t) = v(t)c_1^2(t) + a(t)c_1(t)c_2(t)$$

$$b(t) = v(t) + a(t)c_1(t)c_2(t)$$

(Ec 3.33)

$$z(t) = \int_0^{T_b} v(t)dt + \int_0^{T_b} a(t)c_1(t)c_2(t)dt$$

$$z(t) = \pm T_b + \mathcal{E}$$

Donde  $\int_0^{T_b} c_1(t)c_2(t)dt$  es la función de correlación cruzada entre  $c_1(t)$  y  $c_2(t)$  la cual debe ser baja para poder recuperar con mayor facilidad los datos del usuario respectivo.

Como se pudo observar, solo el receptor que posea el código  $c_1(t)$  podrá recuperar la señal. Las otras señales serán tratadas como interferencia.

Sistemas CDMA utilizan hasta tres diferentes tipos de código para ofrecer mayor complejidad a la señal transmitida. El receptor deberá conocer los tres tipos de código para poder recuperar la información y esto aumenta la seguridad de los datos transmitidos. Por su baja densidad espectral de potencia es difícil detectar e interceptar la señal ensanchada.

#### 3.4.5.4 Acceso Multiusuario

El Acceso Multiusuario se refiere a dar la posibilidad que múltiples usuarios puedan utilizar el mismo medio de comunicación o canal (como técnica de acceso al medio). Si a cada usuario se asigna un código diferente, en recepción se podrá recuperar las informaciones correspondientes, al aplicar el código propio de cada usuario. Cuando se quiere recuperar el mensaje del Usuario 1, se aplica el Código 1, cuando se desea recuperar el mensaje del Usuario 2, se aplica el Código 2, etc.

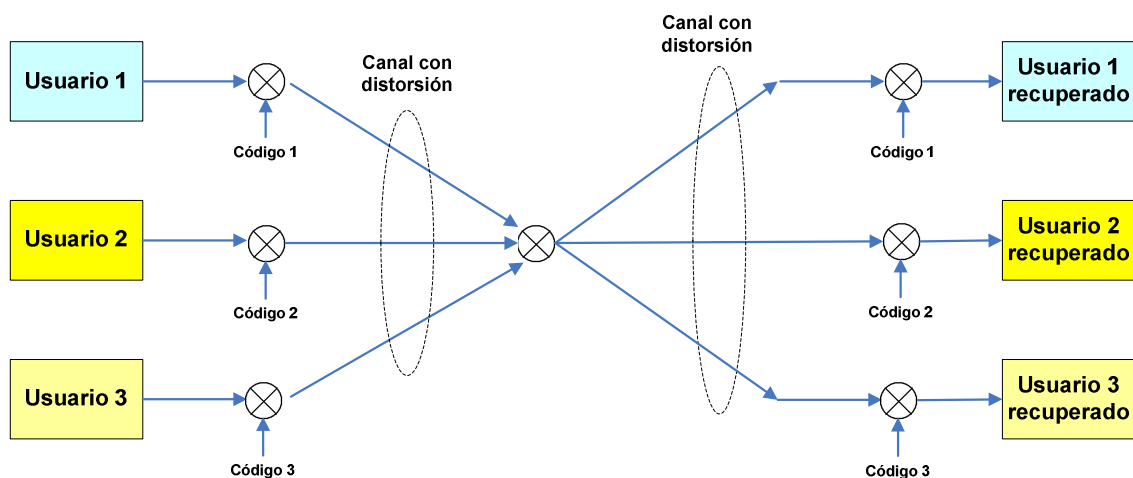


Figura 3.64 Acceso Multiusuario

Cuando varios usuarios tienen acceso al sistema de servicios, cada secuencia de código asignado a un usuario debe ser distinguible de cada secuencia asignada a otros usuarios e idealmente generar poca o ninguna interferencia entre los usuarios que comparten el canal.

Se debe crear una gran familia de secuencias con la finalidad de dar cabida a una gran cantidad de usuarios del sistema. Las secuencias generadas deben tener un gran impulso de autocorrelación que le permitan al sistema mejorar la sincronización y una baja correlación cruzada para reducir la interferencia por el acceso múltiple.

Cada secuencia de código utilizado debe ser distinguible con facilidad de una versión desplazada en el tiempo de sí misma, a fin de permitir la adquisición del código y su seguimiento y, por tanto, la sincronización del sistema. Como se estudió en el Capítulo 2, existen tipos de secuencias como Gold, Kasami, Walsh y OVFS que cumplen con las características mencionadas y que permiten realizar sistemas DSSS multiusuario prácticos.

### **3.5 ANÁLISIS DE LOS SISTEMAS DE SALTO DE FRECUENCIA (FREQUENCY HOPPING-SPREAD SPECTRUM): FH-SS**

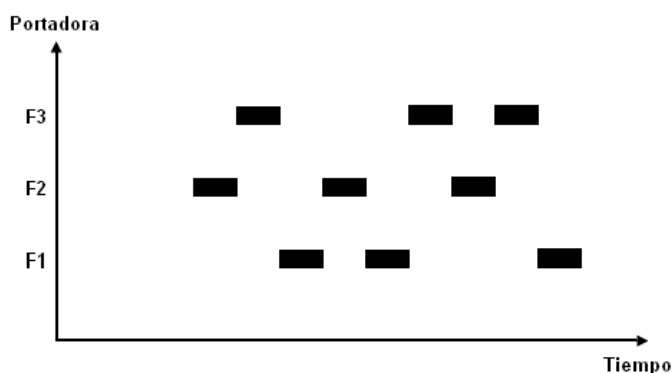
#### **3.5.1 GENERALIDADES DE LOS SISTEMAS FH-SS<sup>71</sup>**

La idea central de la técnica de espectro expandido consiste en transmitir la información en una zona que va cambiando en el tiempo. Para conseguir esto, la frecuencia de la portadora de transmisión debe ir saltando entre canales disponibles dentro del ancho de banda expandido disponible. La comunicación entre el Transmisor y el Receptor en un sistema FHSS, permanece en uno de esos canales durante un tiempo llamado “dwell time” y luego salta a otro canal.

---

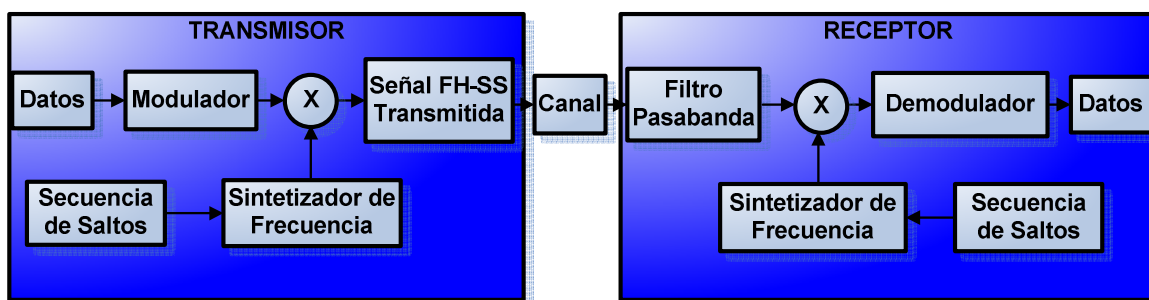
<sup>71</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 164-165.

El salto entre uno y otro canal se realiza mediante un modulador cuya frecuencia portadora es generada por un sintetizador de frecuencias de modo que la frecuencia es fijada por un generador de secuencia pseudoaleatorio (cíclico o aleatorio). Por medio de una técnica de análisis tiempo-frecuencia ideal, los saltos de la frecuencia portadora, podrían lucir como se muestra a continuación:



**Figura 3.65** Saltos de frecuencia portadora en el tiempo.

El transmisor y el receptor FH-SS siguen manteniendo las mismas ideas de pseudoaleatoriedad. La diferencia es que esta vez es aplicada a la frecuencia portadora. De este modo, el ensanchamiento espectral es producido porque el ancho de banda ocupado por la información, debido a aquellos saltos espectrales, es mucho más ancho que el ocupado por la información. Un esquema básico de dicho sistema de comunicaciones sería como el siguiente:



**Figura 3.66** Sistema FHSS básico.

Como se puede ver, el transmisor FHSS básico consiste de un modulador FSK o PSK y un sintetizador de frecuencia. El sintetizador de frecuencia genera frecuencias de portadora de acuerdo a un generador de secuencia de saltos

(cíclico o aleatorio) y el resultado es multiplicado por los datos modulados para generar la señal FHSS.

En el receptor FHSS básico, la señal recibida es primero filtrada utilizando un filtro pasabanda y luego multiplicado por una réplica del sintetizador de frecuencia. Esta salida se pasa por el demodulador correspondiente.

### **3.5.2 TIPOS DE FH-SS<sup>72</sup>**

Dependiendo de la forma cómo varía la frecuencia en un sistema FS/SS, se definen el FH cíclico y el FH aleatorio.

#### **3.5.2.1 Frequency Hopping Cíclico**

Las frecuencias varían de una manera secuencial y repetitiva.

Provee buena diversidad de frecuencia pero no muestra buen desempeño frente a interferencia.

#### **3.5.2.2 Frequency Hopping Aleatorio**

La frecuencia cambia según una secuencia pseudoaleatoria.

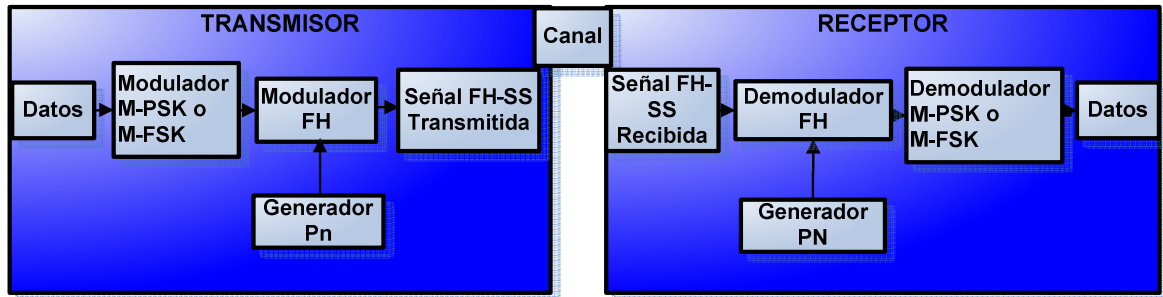
Esta tiene mejor desempeño frente a las interferencias.

Usualmente la modulación es M-FSK ó M-PSK.

Esta señal modulada es multiplicada por un tono de una de entre varias posibles frecuencias generadas aleatoriamente.

---

<sup>72</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 164-167.



**Figura 3.67** Sistema FHSS aleatorio.

### 3.5.3 TIPOS DE SISTEMAS FH-SS ALEATORIOS

Los sistemas FHSS Aleatorios, dependiendo de la velocidad de cambio en la frecuencia de portadora, se clasifican en:

Salto de frecuencia rápido (Fast frequency hopping, F-FH): cuando se transmite un símbolo de datos completo o una fracción del mismo dentro del “dwell time” de un canal.

Salto de frecuencia lento (Slow frequency hopping, S-FH): cuando se transmite más de un símbolo de datos dentro del “dwell time” de un canal.

#### 3.5.3.1 Comparación entre F-FH y S-FH

S-FH: menor inmunidad a interferencias que F-FH pero más barato.

F-FH: mayor inmunidad a interferencias que S-FH pero más caro.



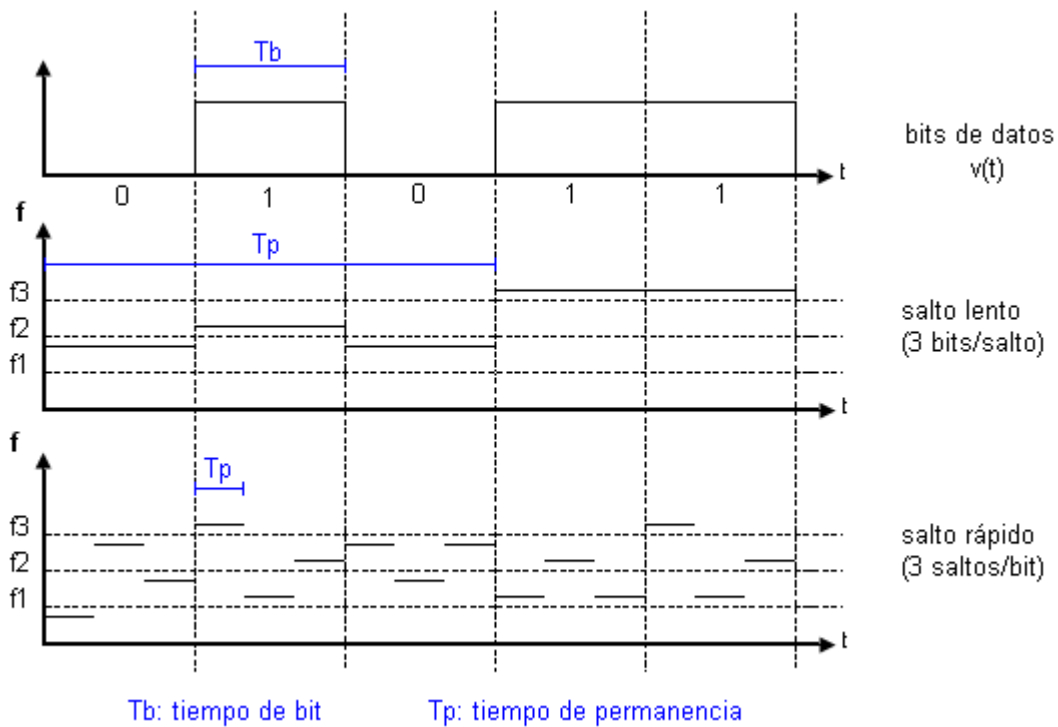


Figura 3.68 Ejemplos de F-FH y S-FH.

### 3.5.4 SIMULACIÓN Y ANÁLISIS EN MATLAB SOBRE UN SISTEMA FH-SS

**Programa cap3\_cod2\_fhss.m (ver Anexos Cap 3)**

#### Caso Ideal

En este ejemplo se genera un número finito de bits de datos para poder analizar y observar cómo se produce el salto de frecuencias de una señal FHSS en el transmisor y su recuperación en el receptor. Se simula una modulación B-PSK de la señal.

Se analizará este sistema utilizando los siguientes casos de canal:

Canal ideal, donde no se considera ruido e interferencia.

Se observan a las señales tanto en el transmisor como en el receptor.

**Sistema FHSS.** Se simulan tanto las señales producidas en el transmisor como las señales recuperadas en el receptor.

El sistema consta de los siguientes bloques.

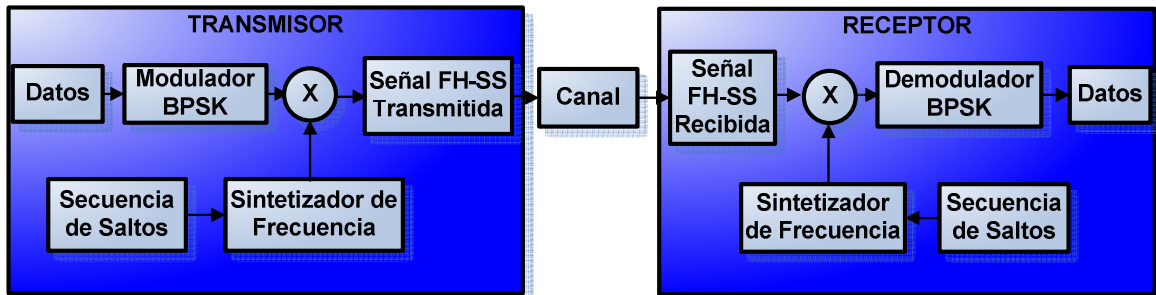


Figura 3.69 cap3\_cod2\_fhss.m.

## 1. Implementación del Transmisor

### 1.1 Generación de los bits de datos

Se genera un número de bits finito. Para la simulación se generan 25 bits con un tiempo de bit igual a 1 segundo. La señal de datos unipolar es convertida a bipolar con lógica negativa.

#### **Resultados:**

Los datos generados se almacenan en la variable **dato**.

A continuación se indican los resultados.

```
>> dato
dato =
Columns 1 through 16
    1    0    0    0    0    1    0    1    0    0    0    1    0    0    0    1
Columns 17 through 25
    0    1    0    1    0    1    0    0    0
```

Figura 3.70 Resultado de la generación de bits de datos.

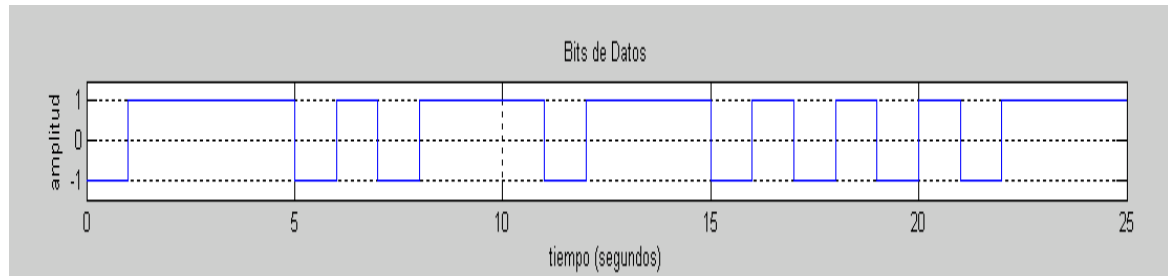


Figura 3.71 Gráfica de señal de datos.

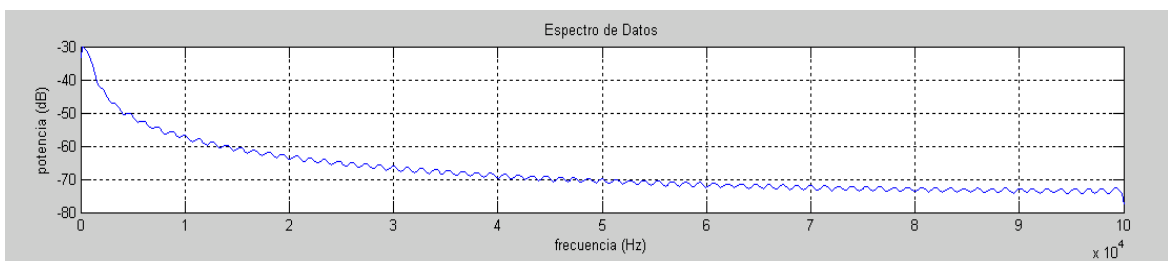


Figura 3.72 Espectro de datos.

En la variable **senalbin** se almacenarán los bits de datos de forma bipolar, cada uno repetido  $N_c$  veces.

```
>> senalbin
senalbin =

Columns 1 through 17
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
Columns 18 through 34
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
Columns 35 through 51
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
Columns 52 through 68
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
Columns 69 through 85
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

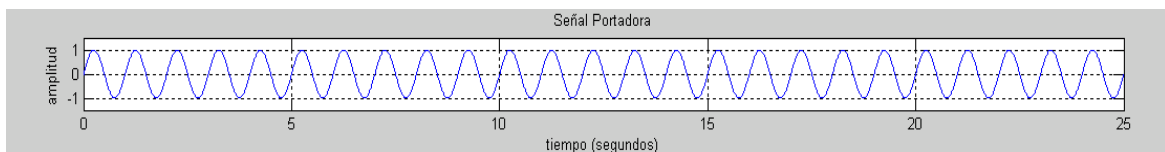
Figura 3.73 Resultado de repetición de bits de datos.

## 1.2 Modulación BPSK

Para realizar la modulación BPSK primero se necesita generar una portadora.

La generación de la portadora se da en base al vector tiempo **ts** obtenido anteriormente. Se genera una señal seno con una frecuencia de 1 Hz.

### Resultados:



**Figura 3.74** Gráfica de señal portadora.

Para realizar la modulación BPSK se multiplica la señal **datos** por la señal portadora.

### Resultados:



**Figura 3.75** Gráfica de señal BPSK.

## 1.3 Sintetizador de Frecuencia con Secuencia de Saltos

El salto de frecuencia se realiza cada tiempo de bit, es decir, que el salto de frecuencia se realizará cada 150 puntos de la simulación. Para realizar el sintetizador de frecuencia, se generan 6 diferentes portadoras de: 12 Hz, 6 Hz, 4 Hz, 3 Hz, 2 Hz y 1 Hz.

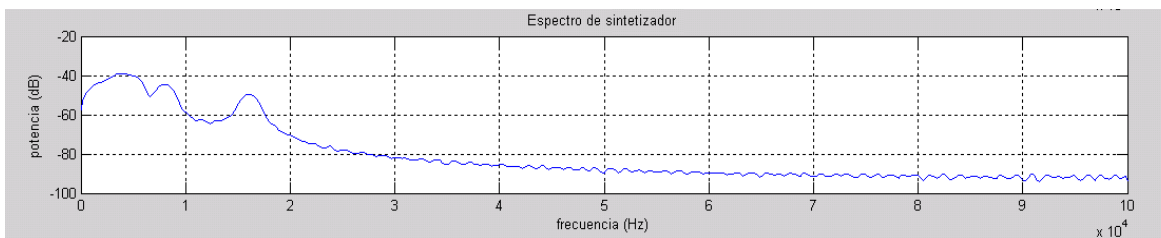
En la variable **secuencia\_salto** se almacena la secuencia de frecuencias que se ha generado para la simulación:

```
>> secuencia_salto
secuencia_salto =
Columns 1 through 17
     4     5     4     3     4     4     5     4     6     2     5     2     1     4     3     3     4
Columns 18 through 25
     5     3     4     3     6     5     2     4
```

**Figura 3.76** Secuencia de frecuencias de salto.



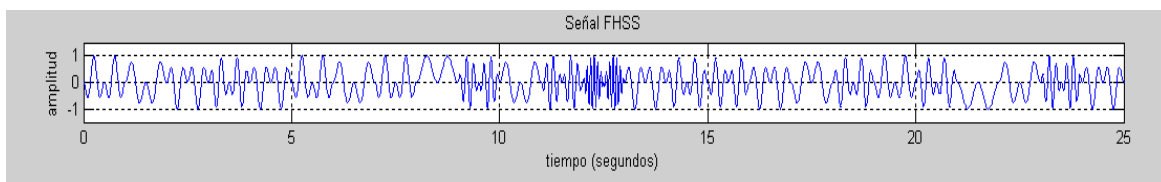
**Figura 3.77** Gráfica de la señal del Sintetizador.



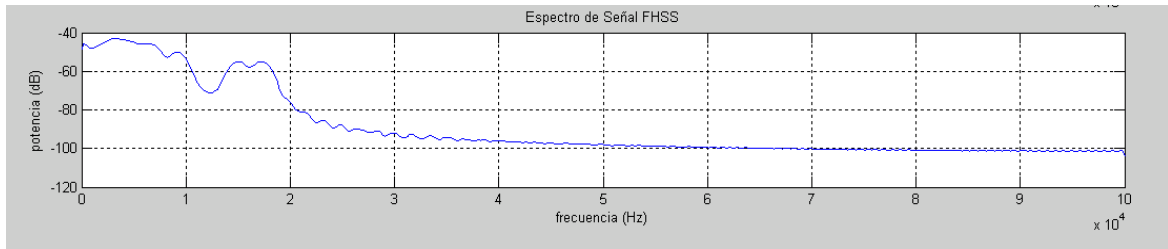
**Figura 3.78** Espectro de potencia de la señal del sintetizador.

### **1.4 Generación de Señal FHSS**

Para la generación de la señal FHSS se realiza la multiplicación entre la señal modulada BPSK con la señal del sintetizador de frecuencia.

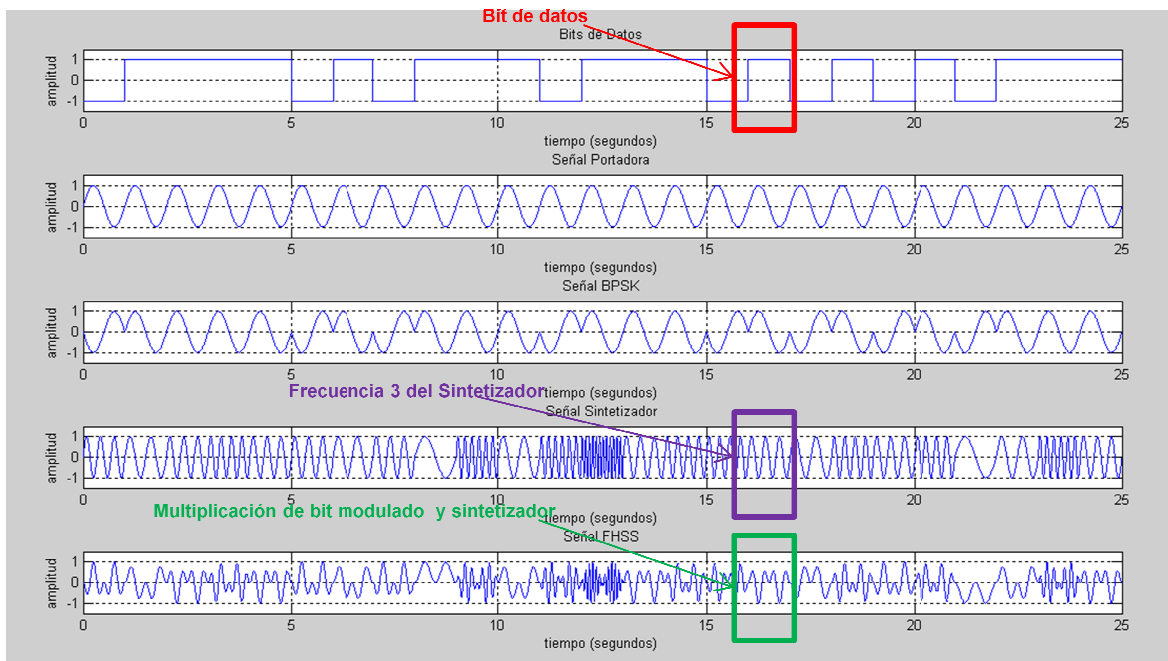


**Figura 3.79** Gráfica de la señal FHSS.



**Figura 3.80** Espectro de potencia de la señal FHSS.

### Gráficas de las señales del Transmisor

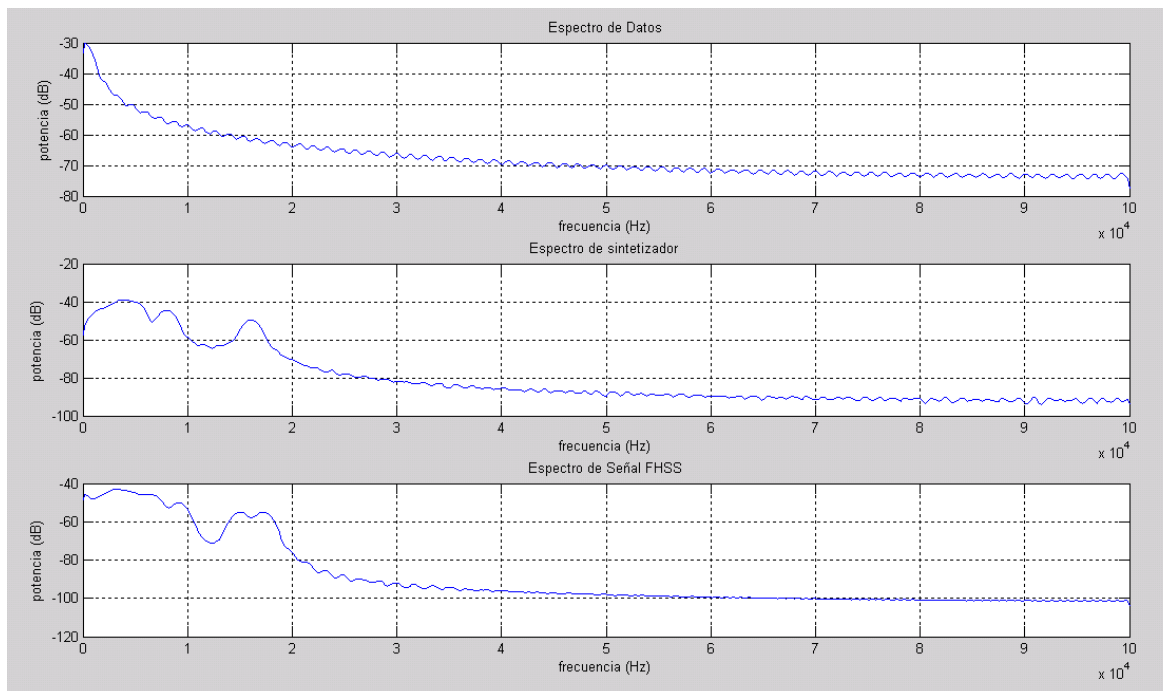


**Figura 3.81** Visualización de las señales obtenidas del transmisor con la simulación.

### Comentarios

Se puede observar que la señal del sintetizador consta de señales con diferente frecuencia y la señal resultante FHSS presenta las mismas características.

### **Gráficas de los espectros de potencia del Transmisor**



**Figura 3.82** Visualización de los espectros de potencia del transmisor.

### **Comentarios**

Se puede observar que el espectro de potencia de la señal de datos es menor que el espectro de la señal FHSS. El espectro de potencia de la señal FHSS no presenta una apariencia de ruido a diferencia de una señal DSSS, debido a que el número de frecuencias utilizadas en el sintetizador es bajo. Aumentando el número de estas, el espectro tenderá a ser plano.

### **2. Canal**

En este caso el canal es ideal; por lo tanto la señal ensanchada no será contaminada con el ruido propio del canal ni con interferencia.

### **3 Implementación del Receptor**

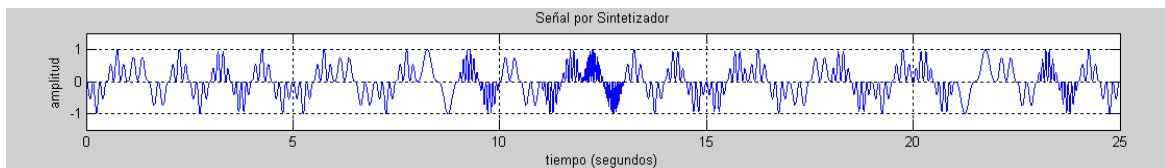
#### **3.1 Recepción de la señal FHSS**

Con las consideraciones de canal ideal, la señal FHSS es la misma que se guardó en la variable *senal\_fhss*.

De igual manera, la secuencia de saltos debe ser la misma que se usó en el transmisor y por tanto debe ser la misma señal del sintetizador de frecuencia, la cual se encuentra en la variable *senal\_ensanchadora*.

#### **3.2 Multiplicador de señal recuperada por sintetizador de frecuencia**

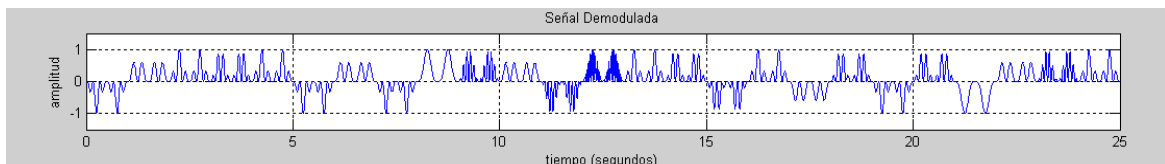
Para recuperar la señal BPSK, se multiplica la señal recibida por la señal del sintetizador de frecuencia, la cual debe ser la misma que en el transmisor, para esto se utiliza la variable *senal\_ensanchadora*:



**Figura 3.83** Gráfica de la señal recuperada por la señal del sintetizador.

#### **3.3 Demodulación BPSK**

Para realizar la demodulación BPSK, se realiza la multiplicación de la señal, por la misma señal portadora utilizada en el transmisor:



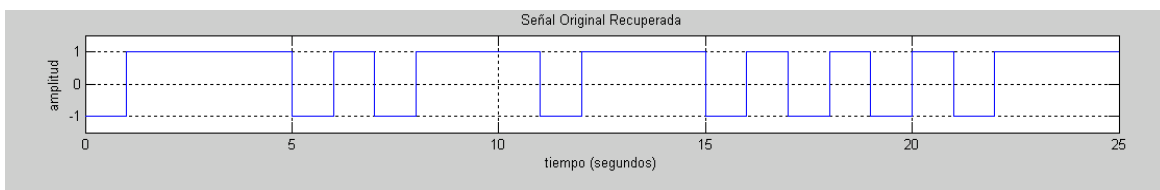
**Figura 3.84** Gráfica de la señal demodulada.



### **3.4 Integrador y toma de decisión**

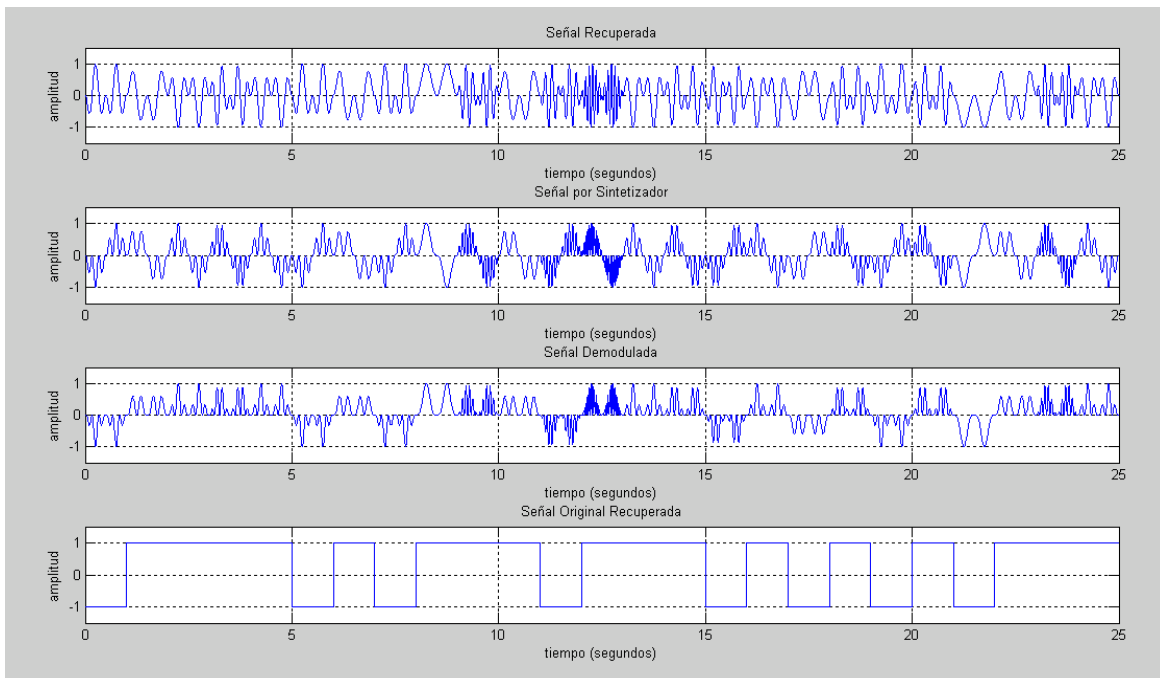
```
>> dato_recibido
dato_recibido =
Columns 1 through 17
    1    0    0    0    0    1    0    1    0    0    0    1    0    0    0    1    0
Columns 18 through 25
    1    0    1    0    1    0    0    0
```

**Figura 3.85** Datos recuperados.



**Figura 3.86** Gráfica de los datos recuperados.

### **Gráfica de las señales del Receptor**



**Figura 3.87** Visualización de las señales obtenidas del receptor con la simulación.

## Comentarios

Se puede observar que se utiliza la misma señal del sintetizador de frecuencia implementada en el transmisor y que la señal recuperada es idéntica a la señal original del transmisor.

### 3.5.5 SIMULACIÓN DE UN SISTEMA BLUETOOTH QUE EMPLEA FH-SS EN SIMULINK

#### **Modelo cap3\_modelo4\_FHSS.mdl (ver Anexos Cap 3)**

A continuación se realiza un ejemplo del transmisor y receptor en un sistema Bluetooth que utiliza FHSS. El modelo realizado se basó en el demo **Bluetooth Frequency Hopping** implementado en el **Communications Blockset** de Matlab.

Se modula señales utilizando modulación GFSK (Gaussian frequency shift keying) sobre un canal de radio con una capacidad máxima de 1 Mbps.

Se utiliza la técnica de salto de frecuencia sobre un rango de frecuencia de 79 MHz para evitar interferencias con otros dispositivos que trabajan en la misma banda. El transmisor divide el tiempo de transmisión en ranuras de 625 microsegundos y utiliza una nueva frecuencia de salto para cada ranura. Debido a que la capacidad del enlace es de 1Mbps, en cada ranura se van a producir 625 bits.

Datos:

$$V_{tx} = 1 \text{ Mbps}$$

$$\text{dwell time} = 625 \text{ us}$$

$$N^{\circ} \text{ saltos} = 1/625 = 1600 \text{ saltos /seg}$$

$$N^{\circ} \text{ de bits por ranura} = 1 \text{ Mbps} * 625 \mu\text{s} = 625 \text{ bits/ranura}$$

La modulación GFSK para salto de frecuencia se realiza con un bloque de modulación CPM y un bloque de modulación M-FSK. Más adelante se explicará su razón. Se utiliza un bloque demodulador M-FSK para realizar la demodulación de la señal FH-SS.

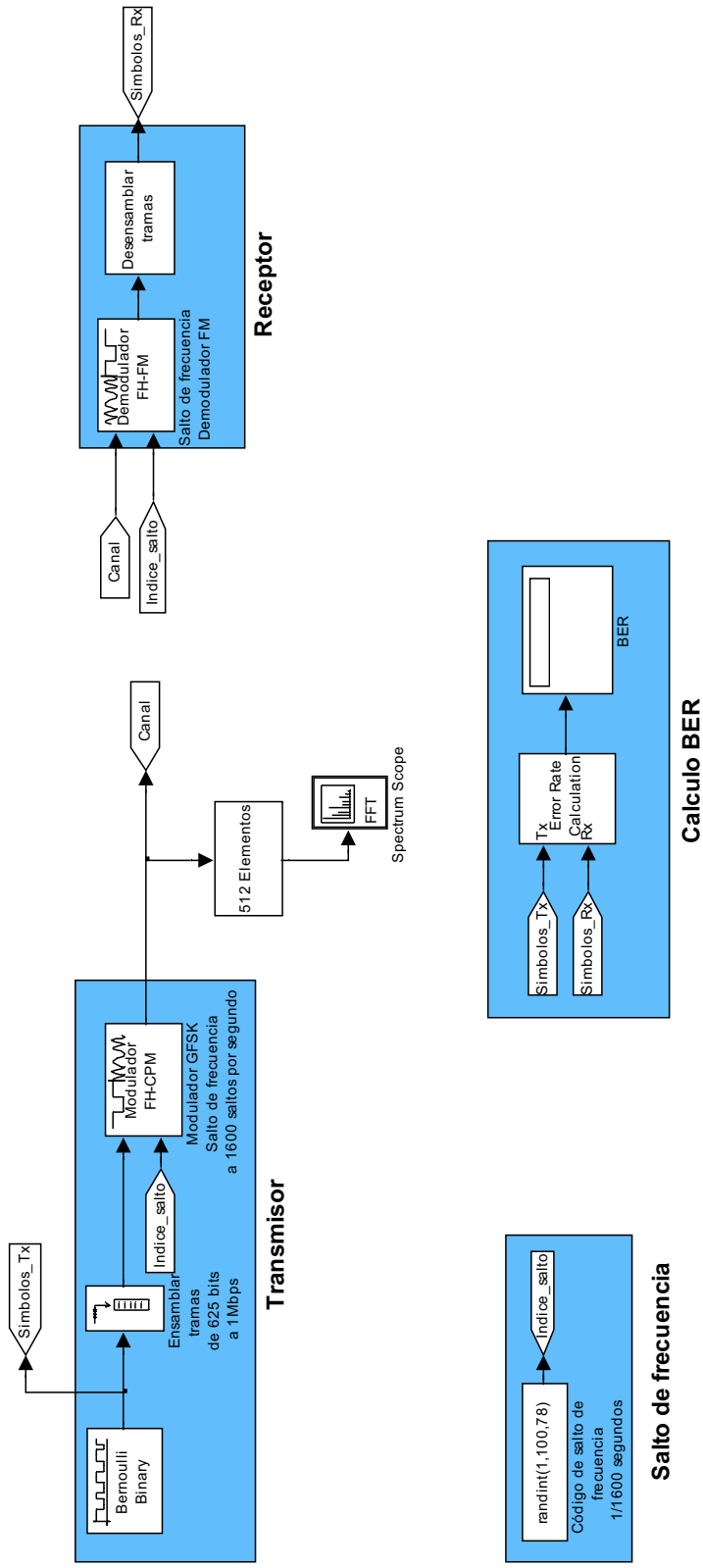
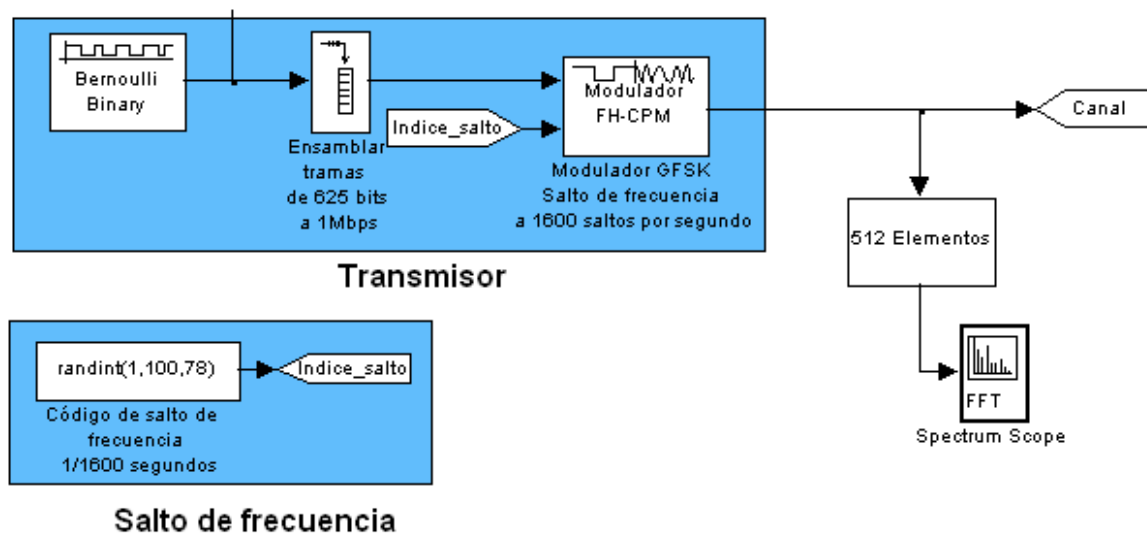


Figura 3.88 cap3\_modelo4\_FHSS.mdl, Sistema Bluetooth en Simulink.

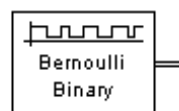
## Transmisor



**Figura 3.89** Transmisor de Sistema FHSS

## Generación de Datos

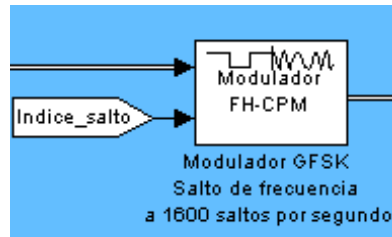
El transmisor consta de un generador de datos binarios de Bernoulli. Este generador está configurado con una tasa de transmisión de 1 Mbps.



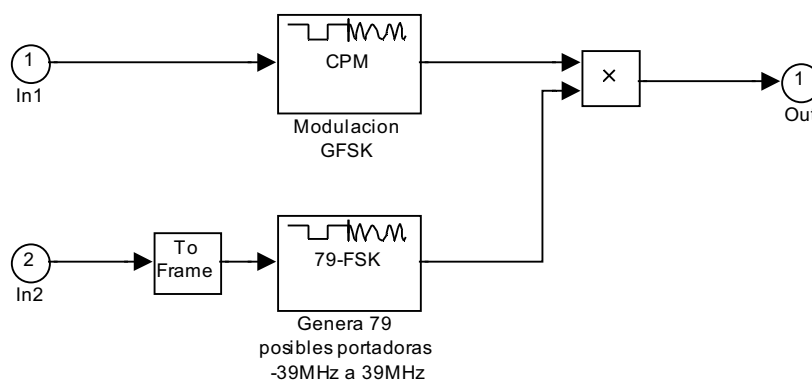
**Figura 3.90** Bloque Bernoulli Binary Generator.

Con esta velocidad de transmisión, se necesita generar tramas de 625 bits para cada ranura de 625  $\mu\text{s}$ .

Luego de armadas las tramas de 625 bits, se procede a realizar la modulación. Como se mencionó anteriormente se realiza modulación GFSK debido a su eficiencia espectral.



**Figura 3.91** Bloque de Modulación.



**Figura 3.92** Bloque de Modulación Expandido.

## **Modulación**

En la modulación por desplazamiento de frecuencia gaussiana GFSK (Gaussian Frequency Shift Keying) se representa a los 1s lógicos mediante un incremento de la frecuencia de la onda portadora, y los 0s lógicos mediante un decremento de la misma.

GFSK es una versión mejorada de la modulación FSK (Frequency Shift Keying), pues introduce ventajas de la técnica de filtrado para **reformado de pulsos** (pulse reshaping). Esta emplea filtros tales como: coseno alzado, raíz de coseno alzado o gaussiano. Esto hará que en lugar de transmitir bits que contienen cambios bruscos de nivel, se transmitan formas de onda mucho más suave que contienen menor cantidad de armónicos.

Antes de modular la señal, es pasada por un **filtro de reformado de pulsos gaussiano** ya que, como se dijo anteriormente, presenta una buena respuesta en

el dominio del tiempo. Con esto se consigue un espectro de energía más estrecho de la señal modulada, reduciendo el ISI y permitiendo así mayores velocidades de transmisión que FSK.

Para realizar esta modulación, en Simulink se utiliza el bloque de modulación CPM, en donde se puede escoger el filtro a utilizar. En la modulación CPM la fase de la portadora es modulada de forma continua. La discontinuidad de fase requiere un gran porcentaje de potencia, lo que da una eficiencia espectral pobre. La continuidad de fase de CPM posee una gran eficiencia espectral que permite tener velocidades de transmisión altas.

Una gran ventaja de la modulación CPM es la gran variedad de formas para detectar la señal, desde baratas y de calidad aceptable, hasta caras y de muy buena calidad.

Cada símbolo es modulado, cambiando gradualmente la fase de la portadora. La fase inicial de cada símbolo es determinada por la fase de todos los símbolos previamente transmitidos, conocida como memoria de fase.

Al índice de modulación se lo conoce como  $\beta = 0.5$ . En la modulación CPM, específicamente en la modulación CPFSK, es el parámetro fijo que marca la separación entre frecuencias instantáneas transmitidas.

Para la modulación CPFSK, la señal modulada es igual a:

$$s(t) = \cos\left(2\pi\left(f_c \pm \frac{\beta}{2T}\right)t + \theta\right), 0 \leq t \leq T \quad (\text{Ec 3.34})$$

$$f_1 = f_c - \frac{\beta}{2T} = f_c - \frac{1}{4T} \quad (\text{Ec 3.35})$$

$$f_2 = f_c + \frac{\beta}{2T} = f_c + \frac{1}{4T} \quad (\text{Ec 3.36})$$

Si se transmite la frecuencia  $f_1$ , se tiene un incremento de fase en un periodo de símbolo:

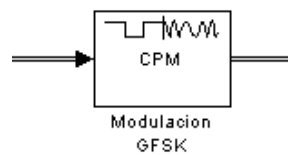
$$\Delta\theta_{f_1} = 2\pi\left(f_c - \frac{\beta}{2T}\right)T = 2\pi\left(f_c - \frac{0.5}{2T}\right)T = 2\pi\left(\frac{1}{T} - \frac{1}{2*2T}\right)T = 2\pi - \frac{\pi}{2} = \frac{\pi}{2} \quad (\text{Ec 3.37})$$

Para  $f_2$ :

$$\Delta\theta_{f_2} = 2\pi\left(f_c + \frac{\beta}{2T}\right)T = 2\pi\left(f_c + \frac{0.5}{2T}\right)T = 2\pi\left(\frac{1}{T} + \frac{1}{2*2T}\right)T = 2\pi + \frac{\pi}{2} = \frac{3\pi}{2} \quad (\text{Ec 3.38})$$

La fase inicial  $\theta_n$  elegida para dar continuidad en un símbolo es la del anterior símbolo  $\theta_{n-1}$  incrementada en  $\pi/2$  o  $3\pi/2$  dependiendo de la frecuencia ( $f_1$  o  $f_2$ ) transmitida en el símbolo anterior.

En esta simulación emplea un filtro de reformado de pulsos gaussiano para simular la modulación GFSK.

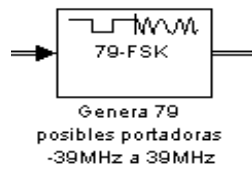


**Figura 3.93** Bloque CPM Modulation.

### **Sintetizador de Frecuencia**

Para el sintetizador de frecuencia se utiliza un modulador 79-FSK. Se disponen de 78 canales, cada uno de 1 MHz más el canal de guarda. Para la selección de canal se utiliza un generador aleatorio de números enteros desde el 1 al 78.

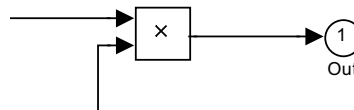
Como sintetizador, se utiliza el bloque de modulación M-FSK, en este caso 79, como se necesitan 78 canales y el canal de guarda.



**Figura 3.94** Bloque M-FSK.

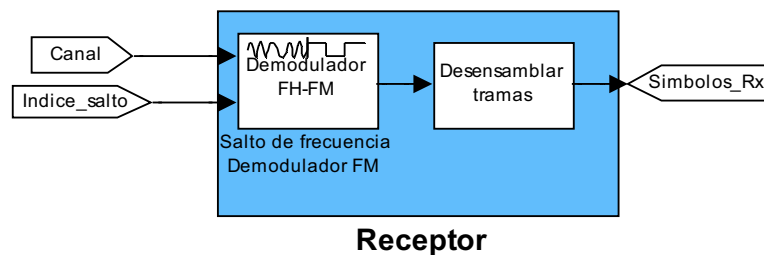
### Ensanchamiento

Para realizar el ensanchamiento del espectro y salto de frecuencia, se multiplica el resultado del modulador CPM con el resultado del modulador 79-FSK.



**Figura 3.95** Multiplicador para obtención de señal FHSS.

### Receptor



**Figura 3.96** Receptor de sistema FHSS

Para recuperar la señal, se necesita un sintetizador de frecuencia con las mismas características que el sintetizador utilizado en el transmisor. Para esto se utiliza un bloque de modulador 79 FSK, que tiene como entrada el mismo generador de números aleatorios que el transmisor. Esto se realiza para simular que se conoce la secuencia de saltos de canal tanto en el transmisor como en el receptor, y que ambos están sincronizados.



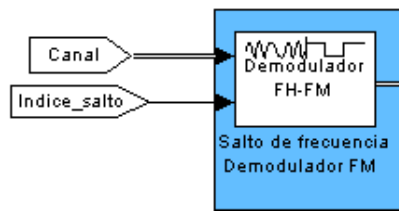


Figura 3.97 Bloque de Demodulación.

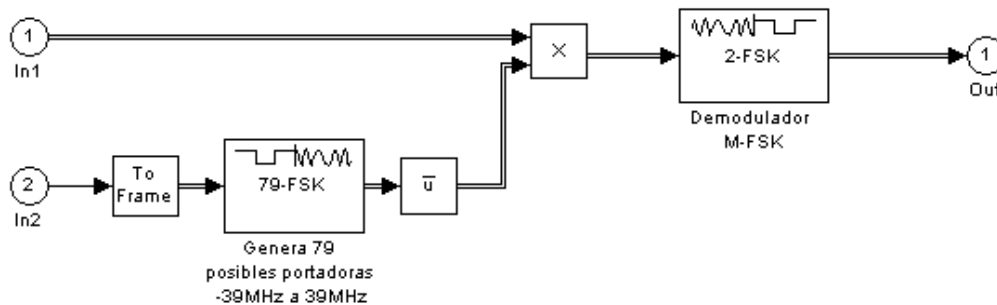


Figura 3.98 Bloque de Demodulación expandido

Como el modulador 79-FSK en el transmisor produce una salida compleja, se necesita en el receptor que el sintetizador, produzca la conjugada del mismo resultado para obtener el módulo de la señal del sintetizador multiplicado por la señal del modulador CPM. Para esto, luego del modulador 79-FSK, se utiliza el bloque **Math Function**, en donde se escoge la operación conjugada (**conj**). El resultado es multiplicado por la señal recibida del transmisor. Matemáticamente se expresa:

$$X * Y = Z$$

$$Z * \bar{Y} = X * Y * \bar{Y} = X * |Y| \quad (\text{Ec 3.39})$$

Siendo

X = Señal modulada CPM

Y = Señal del Sintetizador 79-FSK

Z = Señal FHSS

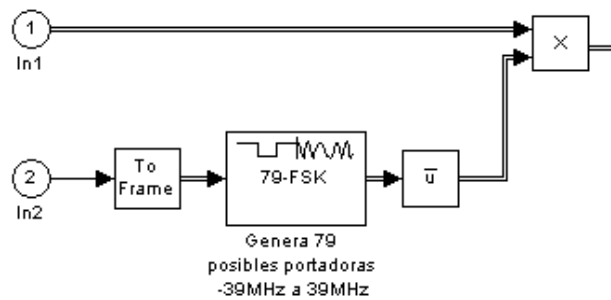


Figura 3.99 Recuperación de la señal GFSK

### Demodulación

La señal obtenida se demodula por medio del bloque Demodulator M-FSK, que permite obtener buenos resultados de BER. Como se mencionó antes, la modulación CPM ofrece buenos resultados con receptores no muy complejos.

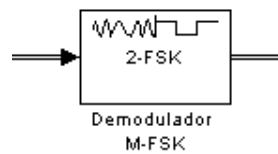


Figura 3.100 Bloque M-FSK Demodulator.

### BER

Con el **bloque Error rate calculation** se calcula el BER de la transmisión y recepción.

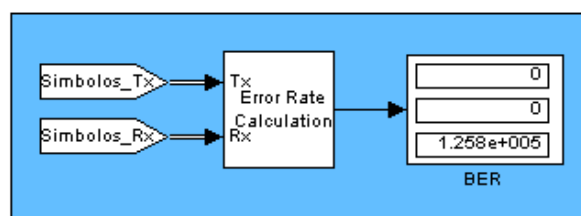
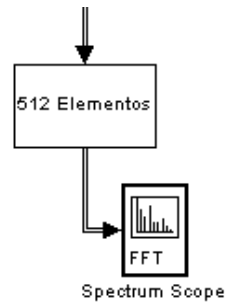


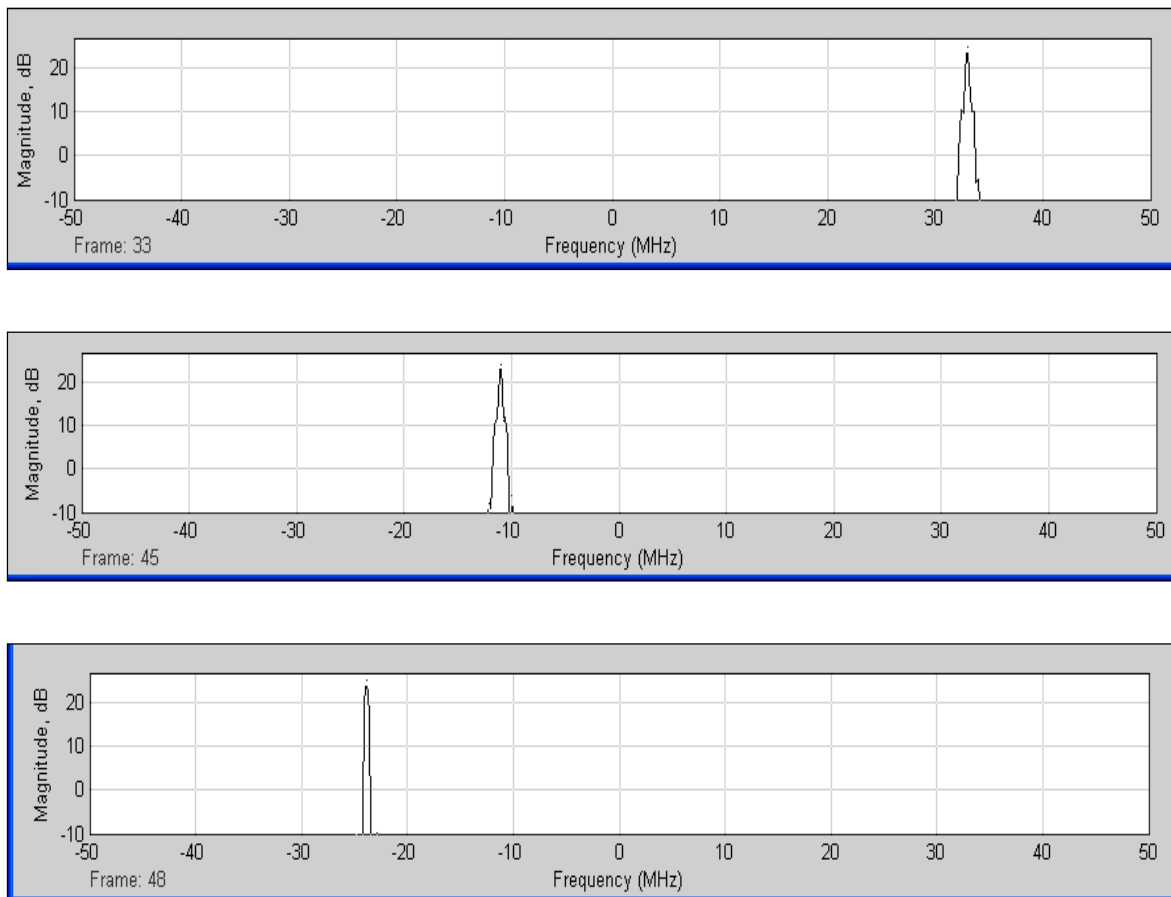
Figura 3.101 Bloque Error Rate Calculation.

### Visualización de saltos de frecuencia

Para la visualización de los saltos de frecuencia de la señal del transmisor, se utiliza el bloque Spectrum Scope. Primero se arman tramas de 512 elementos para luego ser enviadas al osciloscopio.



**Figura 3.102** Bloques para la visualización del Espectro de Potencia.



**Figura 3.103** Saltos del espectro de la señal FHSS.

### 3.5.6 CASO CON RUIDO SIMULINK

***Modelo cap3\_modelo5\_FHSS\_ruido.mdl (ver Anexos Cap 3)***

A continuación se realiza un ejemplo del transmisor y receptor en un sistema FHSS con canal ruidoso:

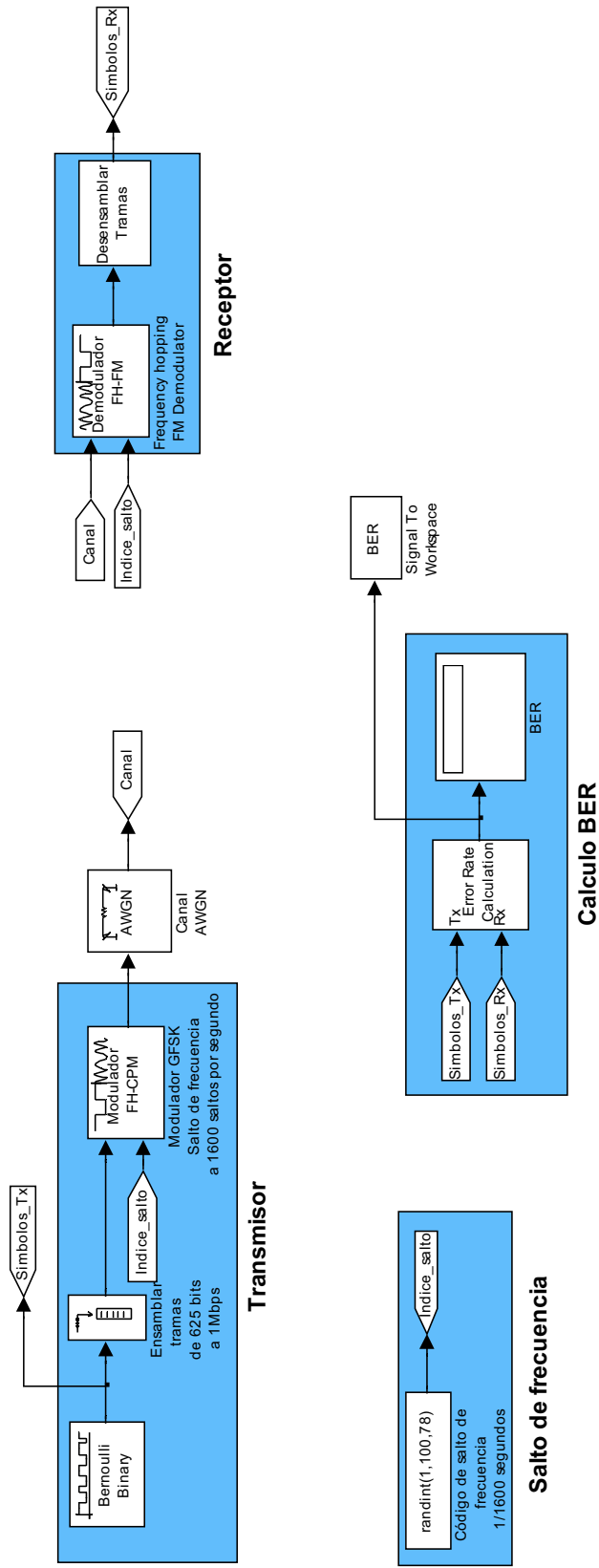
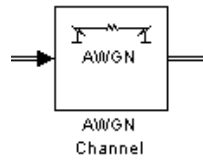


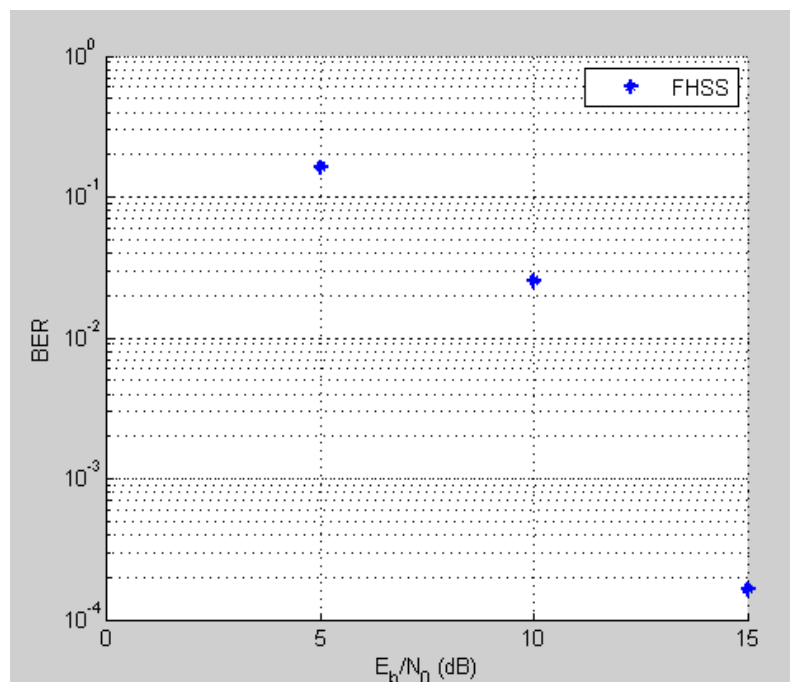
Figura 3.104 cap3\_modelo5\_FHSS\_ruido.mdl, Sistema Bluetooth con canal AWGN en Simulink.

Para simular un canal con ruido blanco gaussiano, se utiliza el bloque AWGN Channel. Este bloque añade ruido gaussiano blanco.



**Figura 3.105** Bloque AWGN Channel.

Para la visualización del BER de la señal FHSS se utiliza la herramienta GUI de MATLAB BERTool.



**Figura 3.106** BER VS  $E_s/N_0$  obtenido con BER Tool.

### **Comentarios**

Se puede observar que el Sistema FH-SS presenta cierta resistencia al ruido menor que un sistema DS-SS. El sistema FH-SS no elimina por completo los efectos del ruido sobre la información transmitida.

Como con la simulación de un sistema DS-SS, no se está utilizando codificación de canal para mitigar los efectos del ruido sobre un canal inalámbrico.

### **3.5.7 CASO CON INTERFERENCIA SIMULINK**

***Modelo cap3\_modelo6\_FHSS\_interferencia.mdl (ver Anexos Cap 3)***

A continuación se realiza un ejemplo del transmisor y receptor en un sistema FHSS con una señal de interferencia FHSS.

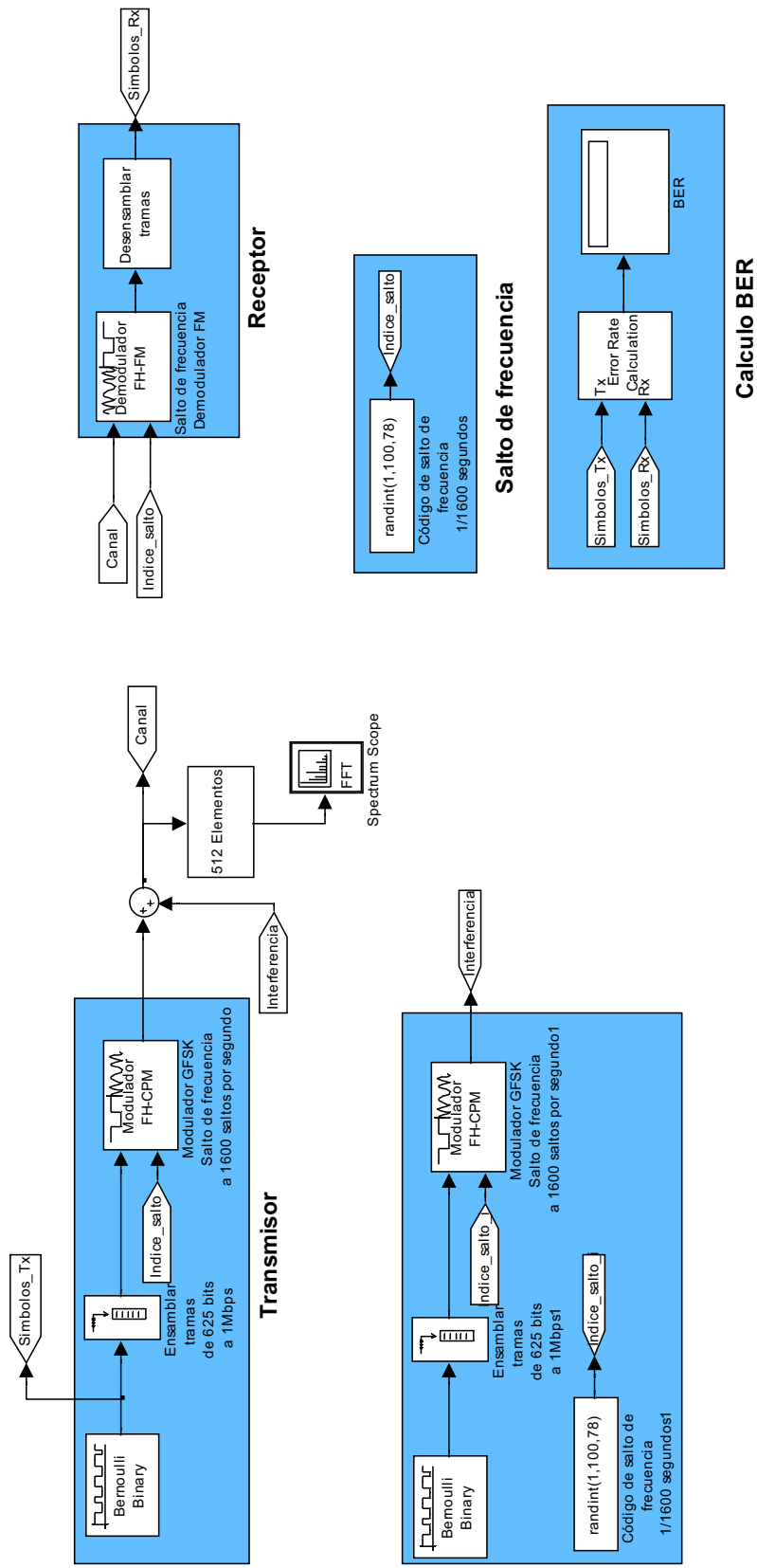
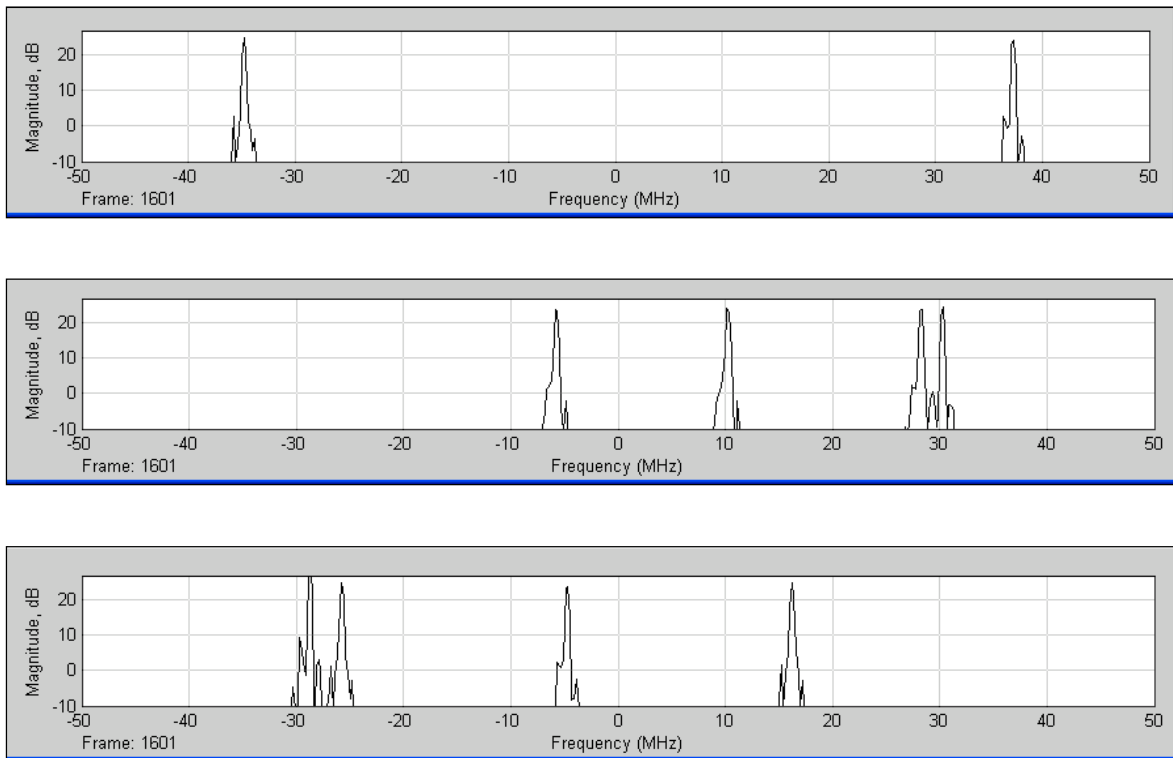


Figura 3.107 cap3\_modelo6\_FHSS\_interferencia.mdl, Sistema Bluetooth con interferencia en Simulink





**Figura 3.108** Espectro de Potencia de Señal e interferencias a diferentes tiempos.

Como se observa, el BER de la señal con una señal de interferencia es 0.

Para 2 sistemas se obtuvo un BER de 0.

Para 3 sistemas se obtuvo un BER de 0.

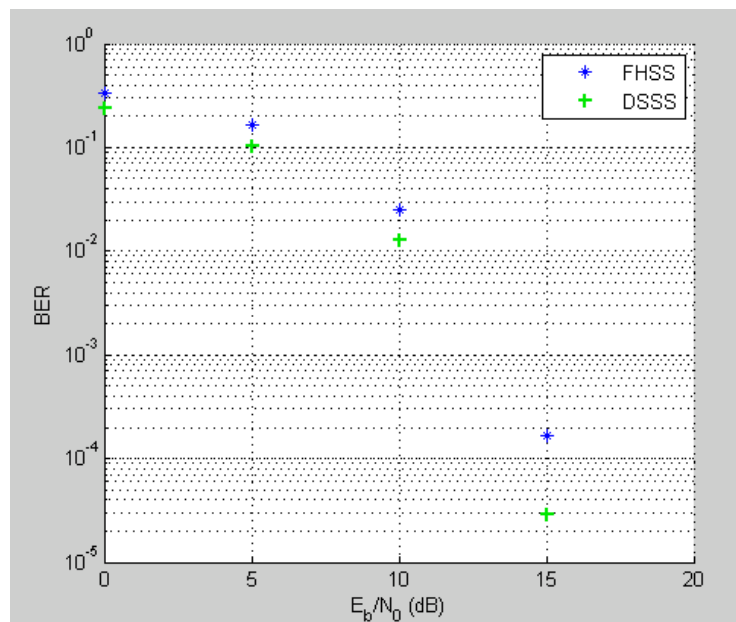
Para 4 sistemas se obtuvo un BER de 0.

Para 5 sistemas se obtuvo un BER de 0.

Como se observa, no existe error entre sistemas FHSS, debido a que las señales de interferencia no poseen una secuencia de salto similar a la señal transmitida y no se superponen la una con la otra. Como se utiliza un ancho de banda total de 78 MHz con canales de 1 MHz, es poco probable que haya la superposición de una señal con otra de características similares.

### 3.5.8 COMPARACIÓN ENTRE UN SISTEMA DS-SS Y UN SISTEMA FH-SS

A continuación se puede observar el resultado del cálculo del BER, en la simulación de un sistema DS-SS y un sistema FH-SS. El BER resultante del sistema DSSS es menor.



**Figura 3.109** Comparación de BER VS  $E_s/N_0$  de Sistema DSSS y FHSS.

DSSS tiene la ventaja de proveer tasas de transmisión mayores que FHSS, pero es afectada por reflexiones. Como se puede observar en las simulaciones, el sistema DSSS no es muy afectado por la presencia del ruido, a diferencia del sistema FHSS.

FHSS es una tecnología muy robusta con poca influencia de reflexiones u otras radio estaciones. El número de sistemas activos simultáneos en la misma área geográfica es mucho mayor que con sistemas DSSS. Como se pudo observar en las simulaciones, el sistema FHSS no es muy afectado por la presencia de sistemas interferentes de similares características.

Una de las mayores ventajas de los sistemas DSSS es su simplicidad tanto en el ensanchamiento de la señal como en su modulación, en comparación de los sistemas FHSS que generalmente utilizan una modulación FSK.<sup>83</sup>

---

<sup>83</sup> Frequency Hopping Spread Spectrum (FHSS) vs. Direct Sequence Spread Spectrum (DSSS) in Broadband Wireless Access (BWA) and Wireless LAN (WLAN), Schwartz, Sorin M.

## CAPÍTULO IV

### 4. DETECCIÓN MULTIUSUARIO Y ESTUDIO DE LA EVOLUCIÓN DE CDMA EN LAS COMUNICACIONES MÓVILES

#### 4.1 DETECCIÓN MULTIUSUARIO

En los sistemas CDMA todos los usuarios transmiten simultáneamente y en la misma banda. La señal de transmisión es modulada bit a bit por un código cuyo ancho de banda es mayor a la de la señal original (espectro ensanchado). Cada usuario posee un código particular; normalmente estos códigos son ortogonales entre sí, lo que permite detectar a un usuario por correlación.

No obstante debido a muchos factores tales como el ruido, el desvanecimiento de la señal, el efecto multitrayectoria, la interferencia intersimbólica (ISI), la detección convencional posee muchos errores, a pesar de que se utilizan técnicas de codificación e interleaving, que ayudan a reducir el efecto del error, a costa de la eficiencia del canal.

Existen sistemas que utilizan códigos cuasi ortogonales, en los cuales la interferencia entre usuarios, conocida como MAI (Multiple Access Interference) es mayor.

Con las técnicas de detección multiusuario se puede mitigar tanto el ISI como el MAI, permitiendo así una detección más eficiente, que a su vez permite incrementar la capacidad del usuario sin necesidad de aumentar el ancho de banda.

#### 4.1.1 DETECCIÓN ÓPTIMA MULTIUSUARIO <sup>84 85</sup>

Los receptores óptimos para los sistemas de acceso múltiple CDMA son designados de acuerdo a dos estrategias diferentes: la estrategia óptima individual usada para minimizar la probabilidad de error de cada usuario individual en el grupo. La otra estrategia, llamada detección óptima conjunta, maximiza la probabilidad a posteriori de los usuarios activos que comparten el canal CDMA. Ambas estrategias no toman las mismas decisiones bajo altos niveles de ruido.

En la transmisión sincrónica CDMA, cada usuario transmite un símbolo en un intervalo de tiempo dado el cual interfiere con otros símbolos de usuarios en el mismo intervalo de tiempo. Se considera un usuario  $K$  del canal básico CDMA, donde cada usuario está transmitiendo un bloque de datos de tamaño  $2M+1$ , y sean los símbolos transmitidos en el  $i$ -ésimo intervalo de tiempo  $\mathbf{b}(i)$  dados por:

$$\mathbf{b}(i) = [b_1(i), \dots, b_k(i), \dots, b_K(i)]^T \quad (\text{Ec 4.1})$$

Donde  $i = -M, -M+1, \dots, 0, \dots, M-1, M$

Luego, todos los símbolos transmitidos por todos los usuarios,  $B$ , están dados por:

$$B = [b(-M), \dots, b(0), \dots, b(M)] \quad (\text{Ec 4.2})$$

La señal de entrada del receptor desde un canal con ruido Gaussiano es:

$$r(t) = s(t, B) + n(t) \quad (\text{Ec 4.3})$$

Donde  $n(t)$  es el ruido Gaussiano y :

---

<sup>84</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 402-414.

<sup>85</sup> Multiuser Detection for CDMA Systems, Duel-Hallen, Alexandra; Holtzman, Jack; Zvonar, Zoran.

$$s(t, B) = \sum_{i=-M}^M \sum_{k=1}^K b_k(i) \sqrt{E_k(i)} C_k(t - iT - \tau_k) \quad (\text{Ec 4.4})$$

Para el k-teavo usuario del i-teavo intervalo de tiempo,  $E_k$  es la energía recibida,  $\tau_k$  es el retraso y  $C_k(t)$  es la forma de onda del usuario normalizada, que es cero fuera del intervalo  $[0.T]$ .

La detección óptima que maximiza la decisión en los símbolos transmitidos basada en la observación de la señal recibida  $r(t)$  en cada intervalo de tiempo, es para maximizar la probabilidad a posteriori, criterio MAP.

Usando la regla de Bayes, la probabilidad a posteriori está dada por:

$$P(B|r(t)) = P(r(t)|B)P[B]/P(r(t)) \quad (\text{Ec 4.5})$$

Para símbolos de igual probabilidad,  $P[B]$  es constante igual a  $\frac{1}{2M+1}$  por lo tanto  $P(r(t))$  es independiente de aquellos símbolos que son transmitidos. Consecuentemente el detector diseñado el criterio de MAP y el detector diseñado usando el criterio de máxima probabilidad, toman las mismas decisiones. Por lo tanto se tiene:

$$P(B|r(t)) = P(r(t)|B) \quad (\text{Ec 4.6})$$

El detector óptimo buscará para la secuencia de símbolos  $B$ , que maximiza,

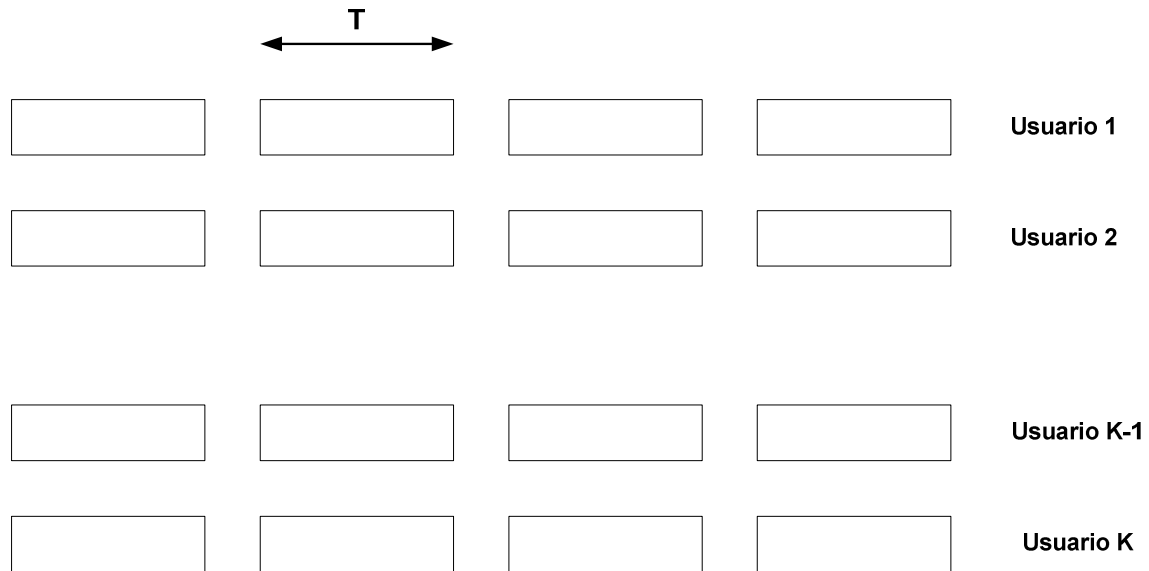
$$P(r(t)) = C \exp[-\Lambda(B)/2\sigma^2] \quad (\text{Ec 4.7})$$

Donde  $C$  es una constante independiente de  $B$ ,  $\sigma^2$  es la varianza de ruido y  $\Lambda(B)$  es la función de probabilidad.

#### 4.1.1.1 Detector Óptimo Sincrónico

En el caso sincrónico, todo retraso  $\tau_k$  para  $k=1, \dots, K$  es igual a cero y es suficiente para separar cada intervalo de símbolo considerado en **Ecuación 4.4**. Así que la sumatoria  $\sum_{i=-M}^M$  puede ser reducida y por lo tanto  $B$  será reemplazada por  $b$ . Consecuentemente la función de probabilidad en el canal sincrónico puede

ser obtenida separadamente para cada intervalo de símbolo, y el detector es llamado **one shot detector**.



**Figura 4.1** Símbolos multiusuario alineados para la transmisión CDMA sincrónica

Así se puede derivar el criterio para un intervalo de tiempo  $i=0$ . La función de probabilidad  $\Lambda(\mathbf{b})$  es dada por:

$$\Lambda(\mathbf{b}) = \int_0^T \left[ r(t) - \sum_{k=1}^K \sqrt{E_k(0)} b_k(0) C_k(t) \right]^2 dt \quad (\text{Ec 4.8})$$

El detector óptimo escogerá  $\mathbf{b}$  que minimiza  $\Lambda(\mathbf{b})$  en la **Ecuación 4.8**. Expandiendo la **Ecuación 4.8** se obtiene:

$$\begin{aligned} \Lambda(\mathbf{b}) = & \int_0^T r^2(t) dt - 2 \sum_{k=1}^K \sqrt{E_k(0)} b_k(0) \int_0^T r(t) C_k(t) dt \\ & + \sum_{k=1}^K \sum_{j=1}^K \sqrt{E_k(0)} \sqrt{E_j(0)} b_k(0) b_j(0) \int_0^T C_k(t) C_j(t) dt \end{aligned} \quad (\text{Ec 4.9})$$

El término incluyendo  $r^2(t)$  en la ecuación anterior, es común para todos los símbolos  $\mathbf{b}$  y pueden ser pasados por alto sin hacer ninguna diferencia para la

reducción de **Ecuación 4.8**. Además  $\int_0^T r(t)C_k(t)dt = y_k(0)$  con  $[k = 1, \dots, K]$  representa la salida del filtro adaptado para el k-ésimo usuario y  $\int_0^T C_k(t)C_j(t)dt = \rho_{kj}(0)$  con  $[k = 1, \dots, K]$  es la correlación cruzada de los usuarios  $k$  y  $j$ .

Por tanto la **Ecuación 4.9** puede ser simplificada a:

$$\Lambda(b) = \int_0^T r^2(t)dt - 2 \sum_{k=1}^K \sqrt{E_k(0)} b_k(0) y_k(0) + \sum_{k=1}^K \sum_{j=1}^K \sqrt{E_k(0)} \sqrt{E_j(0)} b_k(0) b_j(0) \rho_{kj}(0) \quad (\text{Ec 4.10})$$

Sea:

$$\Omega(b) = 2 \sum_{k=1}^K \sqrt{E_k(0)} b_k(0) y_k(0) - \sum_{k=1}^K \sum_{j=1}^K \sqrt{E_k(0)} \sqrt{E_j(0)} b_k(0) b_j(0) \rho_{kj}(0) \quad (\text{Ec 4.11})$$

Por lo tanto función de reducción de probabilidad en **Ecuación 4.10** es equivalente a maximizar  $\Omega(b)$  en **Ecuación 4.11**. Ahora se escribe la **Ecuación 4.11** en forma de matriz como métricas de correlación:

$$\Omega(b) = 2b^T E y - b^T H b \quad (\text{Ec 4.12})$$

Donde:

$$y = [y_1, \dots, y_k, \dots, y_K]^T \quad (\text{Ec 4.13})$$

E es una matriz diagonal K x K de las energías recibidas dadas por:

$$E = \text{diag}[\sqrt{E_1}, \dots, \sqrt{E_k}, \dots, \sqrt{E_K}] \quad (\text{Ec 4.14})$$

$$H = E R E \quad (\text{Ec 4.15})$$



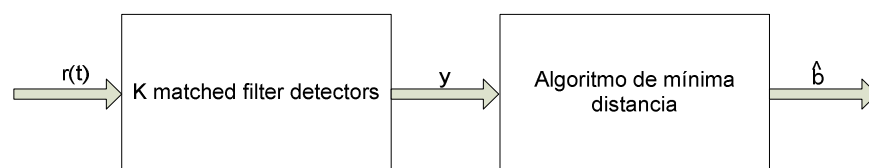
La matriz de correlación cruzada normalizada  $\mathbf{R}$  tiene los elementos de la diagonal igual a 1, y los elementos  $(k, j)$  igual a  $\rho_{kj}(0)$ . Por lo tanto el criterio para la detección óptima multiusuario en el canal CDMA sincrónico Gaussiano, el cual maximiza las métricas de correlación está dado por:

$$\hat{\mathbf{b}} = \text{Arg. max.} \{2\mathbf{b}^T \mathbf{E} \mathbf{y} - \mathbf{b}^T \mathbf{H} \mathbf{b}\} \quad (\text{Ec 4.16})$$

$$\mathbf{b} \in [+1, -1]^K$$

Como se puede observar en la ecuación anterior el detector óptimo debe tener conocimiento de las energías  $\mathbf{E}$  de las señales recibidas y las firmas de usuario. El detector óptimo sincrónico calcula los  $2^K$  valores de las métricas de correlación en la **Ecuación 4.15** para los  $2^K$  posibles elecciones de los símbolos transmitidos desde los  $K$  usuarios y selecciona el símbolo correspondiente a la métrica más larga de correlación. En la práctica el detector sincrónico óptimo consiste de un banco de  $K$  detectores de filtros adaptados seguidos por un algoritmo que selecciona la secuencia que maximiza la métrica de correlación o escogiendo la secuencia correspondiente a la mínima distancia Eduliana.

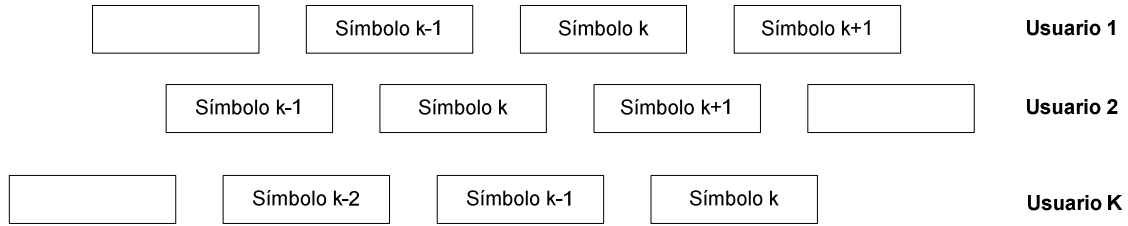
La siguiente figura muestra el receptor óptimo para un sistema CDMA DSSS sincrónico



**Figura 4.2** Receptor óptimo para un sistema CDMA.

#### 4.1.1.2 Detector óptimo asincrónico

La siguiente figura muestra un posible arreglo de símbolos de múltiples usuarios, en el canal asincrónico de CDMA.



**Figura 4.3** Arreglo de símbolos de múltiples usuarios, en el canal asincrónico de CDMA.

Como se puede ver en la figura, cada símbolo deseado está sobrepuesto exactamente por dos símbolos consecutivos de cada usuario indeseado (interferente). Considerando que  $K$  usuarios comparten el canal CDMA asincrónico, cada uno está transmitiendo  $(2M+1)$  símbolos.

Las decisiones óptimas multiusuario del detector son obtenidas por un detector de secuencia de probabilidad máxima que selecciona la secuencia más probable de símbolos transmitidos para las observaciones dadas. La función de probabilidad  $\Lambda(b)$  puede ser calculada utilizando la siguiente integral:

$$\Lambda(b) = \int_{-MT}^{MT+2T} \left[ r(t) - \sum_{k=1}^K \sum_{i=-M}^K \sqrt{E_k(i)} b_k(i) C_k(t - iT - \tau_k) \right]^2 dt \quad (\text{Ec 4.17})$$

$$\begin{aligned} &= \int_{-MT}^{MT+2T} r(t)^2 dt - 2 \sum_{k=1}^K \sum_{i=-M}^K \sqrt{E_k(i)} b_k(i) \int_{-MT}^{MT+2T} r(t) C_k(t - iT - \tau_k) dt \\ &+ \sum_{k=1}^K \sum_{j=1}^K \sum_{i=-M}^M \sum_{m=-M}^M \sqrt{E_k(i) E_j(m)} b_k(i) b_j(m) \int_{-MT}^{MT+2T} C_k(t - iT - \tau_k) C_j(t - mT - \tau_j) dt \end{aligned} \quad (\text{Ec 4.18})$$

Donde  $\mathbf{b}$  es el vector  $K(2M+1)$  representando las secuencias de datos desde los  $K$  usuarios, donde:

$$\begin{aligned}
 & \begin{matrix} b_1(-M) & b_1(-M+1) & b_1(0) & b_1(M-1) & b_1(M) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ b_k(-M) & b_k(-M+1) & b_k(0) & b_k(M-1) & b_k(M) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ b_K(-M) & b_K(-M+1) & b_K(0) & b_K(M-1) & b_K(M) \end{matrix} & \text{(Ec 4.19)}
 \end{aligned}$$

$$b(i) = [b_1(i) \ b_2(2) \ \dots \ \dots \ \dots \ b_K(i)]^T \quad \text{(Ec 4.20)}$$

Como en el caso sincrónico,  $r(t)^2$  es común para todos los usuarios y la integral que involucra a  $r(t)^2$  puede ser ignorada. Se define a la energía de la señal por una matriz diagonal  $\mathbf{E}$  de  $(2M+1)K \times (2M+1)K$ , dada por:

$$E = \text{diag}([\sqrt{E_1(-M)}, \dots, \sqrt{E_K(-M)}, \dots, \sqrt{E_1(M)}, \dots, \sqrt{E_K(M)}]) \quad \text{(Ec 4.21)}$$

La matriz  $\mathbf{R}(\mathbf{m})$   $K \times K$  de correlación cruzada normalizada con elementos:

$$R_{kj}(m) = \int_{-\infty}^{\infty} C_k(t - \tau_k) C_j(t - mT - \tau_j) dt \quad \text{(Ec 4.22)}$$

Ahora desde que las formas de onda de las firmas de los usuarios son ceros fuera del intervalo de tiempo  $[0 \ T]$ , entonces:

$$\begin{aligned}
 R(m) &= 0 \quad |m| > 1 \\
 R(-m) &= R^T(m)
 \end{aligned} \quad \text{(Ec 4.23)}$$

Se define la matriz cuadrada  $\mathbf{R}$  de  $(2M+1)K$ , que representa las matrices  $\mathbf{R}(\mathbf{m})$ , tal que:

$$\mathfrak{R} = \begin{matrix} R(0) & R^T(1) & 0 & \dots & \dots & 0 \\ R(1) & R(0) & R^T(1) & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & R(1) & R(0) & R^T(1) \\ 0 & 0 & 0 & 0 & R(1) & R(0) \end{matrix} \quad (\text{Ec 4.24})$$

La matriz  $\mathfrak{R}$  se define como la matriz **block-Toeplitz simétrica**. La función de probabilidad en  $\Lambda(b)$  en la **Ecuación 4.17** puede se expresada en términos de la métrica de correlación que involucra  $K$  detectores de Filtro Adaptado, uno para cada usuario. La salida se define por  $(2M+1)K$  el vector  $y$ .

$$y = \mathfrak{R}Eb + n \quad (\text{Ec 4.25})$$

$$n = [n^T(-M)n^T(-M+1) \dots \dots \dots n^T(-M)]^T \quad (\text{Ec 4.26})$$

Donde:

$$n(i) = [n_1(i)n_2(i) \dots \dots \dots n_K(i)]^T \quad (\text{Ec 4.27})$$

Los vectores de ruido Gaussiano  $n(i)$  tienen una media cero y una matriz de autocorrelación, la cual depende de la matriz de correlación cruzada  $\mathbf{R}(i)$ :

$$E[n(i)n^T(k)] = \sigma^2 R(i-k) \quad (\text{Ec 4.28})$$

Donde  $\mathbf{R}(i-k)$  está definida en la **Ecuación 4.22**, y por lo tanto, es diferente de cero solo cuando  $|i-k| \leq 1$ .

Siguiendo un proceso similar al utilizado en el caso sincrónico, el criterio para la detección multiusuario óptima en un canal Gaussiano sincrónico es maximizar las métricas de correlación, dadas por:

$$\hat{b} = \text{Arg. max.} \{2b^T E y - b^T E \mathfrak{R} E b\} \quad (\text{Ec 4.29})$$

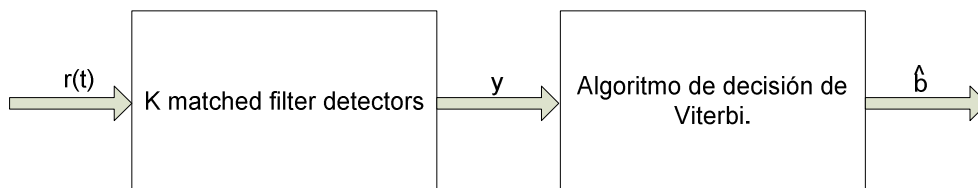
$$b \in [+1, -1]^K$$

Un acercamiento directo para la solución de optimización, será seleccionar las secuencias  $\mathbf{b}$  representadas por el vector de tamaño  $2(M+1)K$  de todos los vectores  $2^{(2M+1)K}$  posibles, los cuales indican el costo de cálculo, que dependen exponencialmente tanto de  $K$  como de  $M$ .

En los sistemas CDMA actuales tanto  $K$  como  $M$  son largos, así que, el acercamiento directo para el problema de optimización es impráctico.

Una solución eficiente para la optimización de la ecuación anterior es propuesta en Verdú, utilizando el algoritmo de Viterbi, el cual selecciona una secuencia  $\mathbf{b}$  que maximiza la métrica de correlación en la ecuación anterior, la cual hace que el costo del cálculo se incremente exponencialmente con  $K$  y  $M$ .

El algoritmo de decisión óptima propuesta por Verdú consiste de  $K$  detectores de filtro adaptado, seguido por un algoritmo de decisión de probabilidad máxima de Viterbi.



**Figura 4.4** Detector óptimo para Sistemas CDMA sincrónicos.

## 4.1.2 DETECTORES LINEALES SUBÓPTIMOS<sup>86 87</sup>

### 4.1.2.1 Detector Decorrelador

La salida  $\mathbf{y}$  del receptor de filtro adaptado para  $K$  usuarios en un sistema CDMA sincrónico puede ser expresado de la forma de matriz:

<sup>86</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 416-422.

<sup>87</sup> Multiuser Detection for CDMA Systems, Duel-Hallen, Alexandra; Holtzman, Jack; Zvonar, Zoran.

$$y = RA b + n \quad (\text{Ec 4.30})$$

Donde  $\mathbf{R}$  es la matriz de correlación cruzada normalizada de orden  $K \times K$  para las formas de onda de firma de  $K$  usuarios y el vector  $\mathbf{n}$  representa muestras de ruido AWGN. Sea la energía recibida el  $k$ -avo usuario,  $E_k$  tal que las energías de las señales de los usuarios puedan ser escritas como  $\mathbf{A}$ , donde:

$$A = \text{diag}[\sqrt{E_1}, \sqrt{E_2}, \dots, \sqrt{E_K}] \quad (\text{Ec 4.31})$$

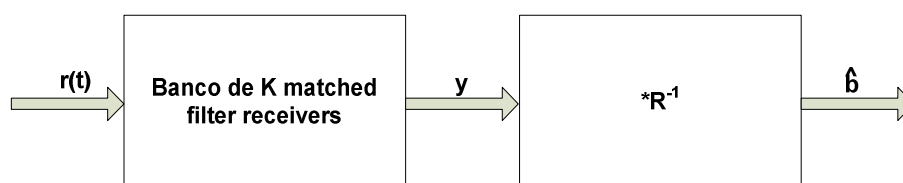
Donde  $\text{diag}[J]$  es una matriz diagonal. Ahora multiplicando ambos lados de la **Ecuación 4.30** con la inversa de  $\mathbf{R}$  (denotada como  $\mathbf{R}^{-1}$ ), asumiendo que  $\mathbf{R}^{-1}$  existe, eso quiere decir que  $\mathbf{R}$  es una matriz cuadrada.

$$\mathbf{R}^{-1}y = \mathbf{R}^{-1}RA b + \mathbf{R}^{-1}n \quad (\text{Ec 4.32})$$

Ya que  $\mathbf{R}^{-1}\mathbf{R} = \mathbf{I}$  donde  $\mathbf{I}$  es la matriz identidad y  $\mathbf{IAb} = \mathbf{Ab}$ , se puede simplificar la **Ecuación 4.31**:

$$\mathbf{R}^{-1}y = Ab + \mathbf{R}^{-1}n \quad (\text{Ec 4.33})$$

El decorrelador descrito en la ecuación anterior transforma de forma lineal, las salidas del receptor de filtro adaptado convencional en secuencias transmitidas ajustadas  $\mathbf{Ab}$  más ruido. A continuación se muestra un diagrama del receptor decorrelador.



**Figura 4.5** Receptor Decorrelador para un sistema CDMA sincrónico

El receptor decorrelador elimina la interferencia multiacceso asociada con la detección de filtro adaptado pero cambia (generalmente incrementa) el nivel de ruido de fondo. La ventaja significativa del receptor decorrelador es su baja complejidad (incrementa linealmente con  $K$ ) comparado con el receptor óptimo (la

complejidad aumenta exponencialmente con  $K$ ). Sin embargo necesita el conocimiento de las formas de onda de firma, información de sincronización y las energías de todos los usuarios.

El receptor decorrelador para un sistema CDMA sincrónico puede ser derivado utilizando un acercamiento similar pero la correlación  $\mathfrak{R}$  es dependiente del tiempo dada por la **Ecuación 4.24**. y  $\mathbf{A}$  es reemplazada por  $\mathbf{E}$  así que multiplicar ambos lados de **Ecuación 4.25** por  $\mathfrak{R}^{-1}$  se obtiene:

$$\mathfrak{R}^{-1}y = Eb + \mathfrak{R}^{-1}n \quad (\text{Ec 4.34})$$

Las **Ecuación 4.33** y **Ecuación 4.34** muestran la salida de los receptores decorreladores consisten de dos términos: uno debido a las señales útiles de los usuarios que son  $\mathbf{Ab}$  y  $\mathbf{Eb}$  y el otro término debido al ruido de fondo  $\mathbf{R}^{-1}n$  y  $\mathfrak{R}^{-1}n$  con media cero. La componente de ruido del receptor en un sistema CDMA sincrónico tiene varianza dada por:

$$E[(\mathfrak{R}^{-1}n)(\mathfrak{R}^{-1}n)^T] = E[\mathbf{R}^{-1}n n^T \mathbf{R}^{-1}] \quad (\text{Ec 4.35})$$

Claramente se ve que la señal útil del  $k$ -teavo usuario es ortogonal a las señales interferentes, significando que el receptor decorrelador es resistente al efecto Near-Far.

#### 4.1.2.2 Detección de Error Cuadrático Medio Mínimo (Minimum Mean Square Error , MMSE)

Una forma eficiente para estimar una variable aleatoria  $\mathbf{b}$  basada en la observación de  $\mathbf{y}$ , es encontrar la función  $\mathbf{b}(\mathbf{y})$  cuyo error cuadrático medio mínimo (MMSE) es:

$$E\left([b - b(\mathbf{y})]^2\right) \quad (\text{Ec 4.36})$$

El detector MMSE aplica una transformación lineal a la salida ( $\mathbf{y}$ ) del detector convencional de un banco de filtros adaptados para minimizar la diferencia entre la secuencia de símbolos transmitidos  $\mathbf{b}$  y la secuencia de símbolos estimados  $\hat{\mathbf{b}}$

definida por la función  $\mathbf{b}(\mathbf{y})$ . La optimización de la función  $\mathbf{b}(\mathbf{y})$  se logra escogiendo la matriz  $\mathbf{W}$  de orden  $K \times K$  que logra:

$$\min_{\mathbf{W} \in R} E(\|\mathbf{b} - \mathbf{W}\mathbf{y}\|^2) \quad (\text{Ec 4.37})$$

Donde  $\mathbf{y}$  está dada por  $\mathbf{y} = \mathbf{R}\mathbf{A}\mathbf{b} + \mathbf{n}$ . Ya que  $\|\mathbf{X}\|^2 = \text{trace}\{\mathbf{X}\mathbf{X}^T\} = \text{trace}\{\text{cov}\{\mathbf{X}\}\}$ , la **Ecuación 4.37** puede ser escrita como:

$$\min_{\mathbf{W} \in R} E(\|\mathbf{b} - \mathbf{W}\mathbf{y}\|^2) = \min_{\mathbf{W} \in R} \text{trace}\{\text{cov}\{\mathbf{b} - \mathbf{W}\mathbf{y}\}\} \quad (\text{Ec 4.38})$$

La covarianza del vector de error  $[\text{cov}\{\mathbf{b} - \mathbf{W}\mathbf{y}\}]$  está dado por:

$$\begin{aligned} \text{cov}\{\mathbf{b} - \mathbf{W}\mathbf{y}\} &= E[(\mathbf{b} - \mathbf{W}\mathbf{y})(\mathbf{b} - \mathbf{W}\mathbf{y})^T]^T \\ \text{cov}\{\mathbf{b} - \mathbf{W}\mathbf{y}\} &= [\mathbf{I} + \sigma^{-2} \mathbf{A}\mathbf{R}\mathbf{A}]^{-1} + (\mathbf{W} - \bar{\mathbf{W}})(\mathbf{R}\mathbf{A}^2\mathbf{R} + \sigma^2\mathbf{R})(\mathbf{W} - \bar{\mathbf{W}})^T \end{aligned} \quad (\text{Ec 4.39})$$

Se asume que  $\mathbf{A}$  no es singular y:

$$\bar{\mathbf{W}} \equiv \mathbf{A}^{-1}[\mathbf{R} + \sigma^2\mathbf{A}^{-2}]^{-1} \quad (\text{Ec 4.40})$$

Ya que el segundo término de **Ecuación 4.39** es siempre positivo entonces la **Ecuación 4.40** se vuelve:

$$\min_{\mathbf{W} \in R} E(\|\mathbf{b} - \mathbf{W}\mathbf{y}\|^2) = \text{trace}\left\{[\mathbf{I} + \sigma^{-2} \mathbf{A}\mathbf{R}\mathbf{A}]^{-1}\right\} \quad (\text{Ec 4.41})$$

Consecuentemente la **Ecuación 4.40** alcanza la suma mínima de los errores cuadráticos medios de manera que la secuencia de símbolos estimados del  $k^{\text{th}}$  usuario está dada por:

$$\begin{aligned} \hat{b}_k &= \text{sgn}(\bar{\mathbf{W}}\mathbf{y})_k \\ \hat{b}_k &= \text{sgn}\left(\mathbf{A}_k^{-1}\left([\mathbf{R} + \sigma^2\mathbf{A}^{-2}]^{-1}\mathbf{y}\right)_k\right) \end{aligned} \quad (\text{Ec 4.42})$$

Se asume que  $\mathbf{A}_k^{-1}$  es la unidad, **Ecuación 4.42** puede ser simplificado como:

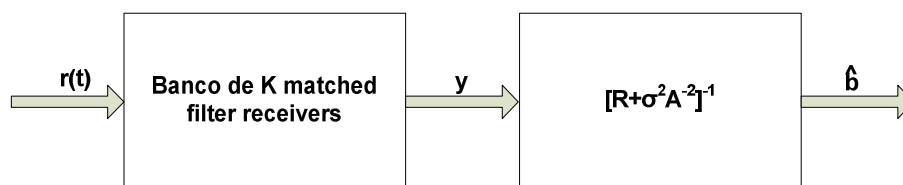


$$\hat{b}_k = \text{sgn}\left(\left([R + \sigma^2 A^{-2}]^{-1} y\right)_k\right) \quad (\text{Ec 4.43})$$

Del análisis previo, se concluye que el receptor MMSE sustituye la transformación  $\mathbf{R}^{-1}$  en el receptor correlador por:

$$[R + \sigma^2 A^{-2}]^{-1} \quad (\text{Ec 4.44})$$

A continuación se muestra un diagrama de un receptor MMSE para un sistema CDMA sincrónico:



**Figura 4.6** Receptor MMSE para un sistema CDMA sincrónico.

#### 4.1.3 ESQUEMAS DE COMBATE A INTERFERENCIA<sup>88 89</sup>

La capacidad de los sistemas de espectro ensanchado multiusuario está limitada por la interferencia de múltiple acceso (MAI). Los esquemas de los detectores óptimos y subóptimos mitigan los efectos de la interferencia multiusuario, sin embargo las estrategias que se utilizan implican un costo de cálculo adicional comparados con un receptor de filtro adaptado convencional, y esto compromete la calidad de servicio. Una alternativa efectiva y una técnica ampliamente utilizada para reducir el desvanecimiento, es la diversidad de antenas. El enfoque clásico es el uso de antenas en el receptor y realizar la combinación de la señal o, seleccionar una conmutación de antenas para mejorar el BER de la señal recibida.

<sup>88</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 423-442.

<sup>89</sup> Multiuser Detection for CDMA Systems, Duel-Hallen, Alexandra; Holtzman, Jack; Zvonar, Zoran.

El principal problema con el esquema de diversidad en los receptores móviles es que hace que los receptores sean más grandes y más caros. Por lo tanto estas técnicas son aplicadas solo en las estaciones base, ya que cada estación base sirve a cientos o miles de estaciones móviles. Esto, por lo tanto, es mucho más económico añadir equipamiento a las estaciones bases que a las estaciones móviles. Por esta razón, los esquemas de diversidad de transmisión para los downlinks y esquemas de diversidad de recepción para los uplinks son muy atractivos para los sistemas de comunicación móvil.

Ahora se realiza un enfoque sobre las técnicas para utilizar múltiples antenas en el transmisor y/o en el receptor. Esto puede reducir la interferencia. Consecuentemente, la capacidad del sistema y la cobertura de la red se incrementan. Un esquema importante en estas técnicas es la antena inteligente, que controla inteligentemente la radiación hacia un usuario deseado y reduce significativamente la interferencia con otros usuarios.

#### **4.1.3.1 Antenas Inteligentes**

Una antena es un puerto donde la energía electromagnética es radiada desde un transmisor de radio al espacio. Esta energía es adquirida desde el espacio usando una antena receptora adecuada. Tradicionalmente las antenas son clasificadas de acuerdo a su patrón de radiación como omnidireccionales o direccionales.

Las antenas omnidireccionales tienen una ganancia igual en todas las direcciones y son conocidas como antenas isotrópicas. Las antenas omnidireccionales son comúnmente usadas tanto en las estaciones base como en las estaciones móviles. La radiación omnidireccional llega al usuario deseado con un porcentaje pequeño de toda la energía radiada. Debido a esta limitación, para mejorar el rendimiento de un usuario deseado solo puede lograrse aumentando el nivel de potencia del transmisor, lo que a su vez aumenta la interferencia para otros usuarios en la misma o en celdas adyacentes.

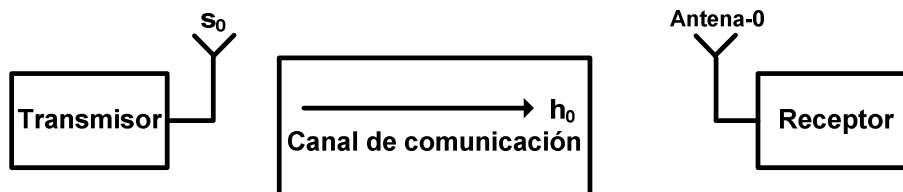
Las antenas direccionales tienen direcciones de transmisión y recepción fijas, lo que significa que tienen una mayor ganancia en esa dirección y menor en otras.

Son empleadas en las estaciones base. Un sistema de antenas sectorizadas divide a un área en sectores que son cubiertos con antenas direccionales, que se encuentran en la misma estación base. Operacionalmente cada sector puede ser tratado como una celda diferente en el sistema.

La necesidad de transmitir eficientemente a un grupo grande de usuarios en una celda determinada, sin empeorar la interferencia de otros usuarios, guía a la evolución de las antenas inteligentes. Un sistema de antenas inteligentes consiste de múltiples antenas cuyas salidas son procesadas en el transmisor/receptor para mitigar la interferencia y el desvanecimiento.

La inteligencia de la antena está medida por su capacidad de procesamiento digital de la señal.

En un sistema de comunicación inalámbrico convencional se utiliza una antena transmisora a la entrada del canal y una antena receptora a la salida del canal. Este arreglo de antenas es llamado de una Sola Entrada una Sola Salida (Simple Input Simple Output, SISO).



**Figura 4.7** Sistema de Comunicación Inalámbrico Convencional SISO

Dos o más antenas pueden ser usadas en el transmisor o en el receptor o en ambos. Este sistema de antenas es llamado una Sola Entrada y Múltiples Salidas (Single in Multiple Out, SIMO); Múltiples Entradas y una Sola Salida (Multiple In Single Out MISO), y Múltiples Entradas y Múltiples Salidas (Multiple In Multiple Out, MIMO).

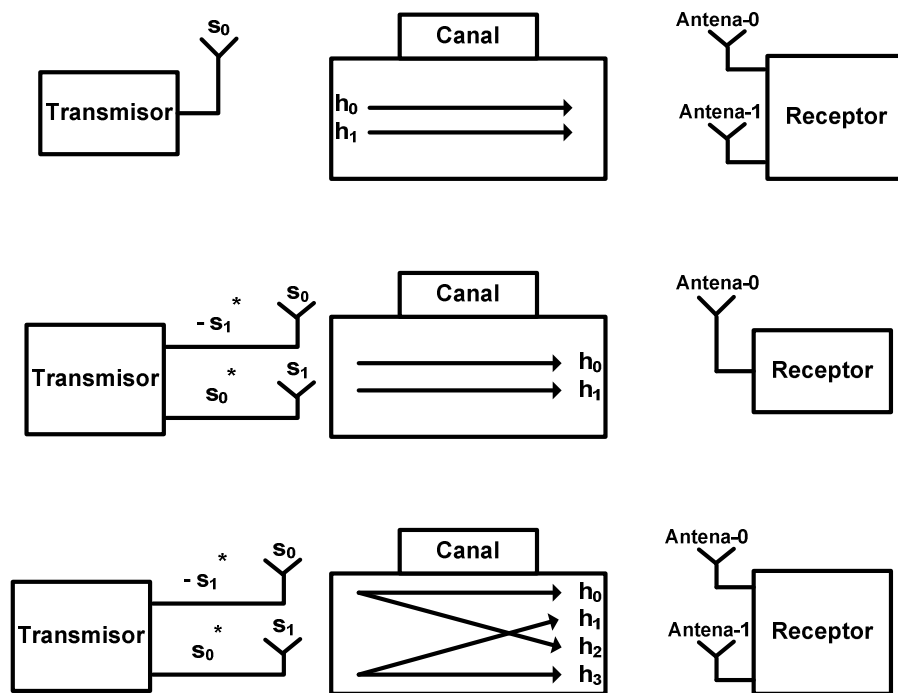
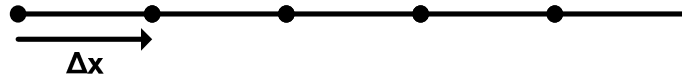


Figura 4.8 Sistemas SIMO, MISO y MIMO.

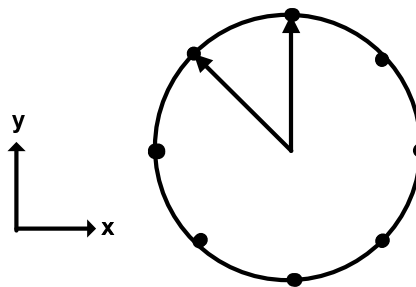
Una versión anterior del uso de múltiples antenas consistía de dos antenas idénticas separadas por varias longitudes de ondas y conectadas a un receptor convencional. Asumiendo que existe la suficiente dispersión de la energía para decorrelar las señales recibidas, la probabilidad que ambas señales recibidas sean extremadamente pequeñas al mismo tiempo es mínima. Por lo tanto para proveer de calidad a la recepción, se selecciona a la señal más fuerte a la entrada del receptor. Esta técnica es llamada Diversidad de Conmutación (Switch Diversity). Claramente un esquema tan simple mejora el rendimiento del sistema evitando el desvanecimiento de la señal, pero no eliminará o reducirá la interferencia intersímbolo, o la interferencia cocanal, así que es incapaz de incrementar la capacidad del canal o cobertura.

La geometría de un sistema de múltiples antenas puede tomar varias formas, por ejemplo, las antenas pueden ser ubicadas en una dimensión con un espaciamiento uniforme de  $\Delta x$  como se ve en la figura:



**Figura 4.9** Arreglo de Antenas Inteligentes ubicadas en una sola dimensión.

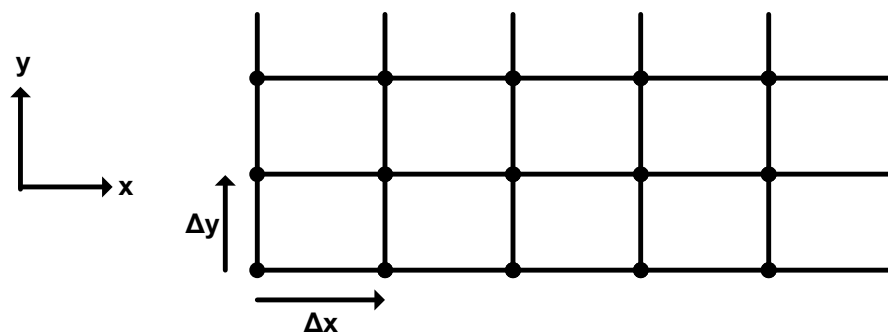
Las antenas pueden ser ubicadas en un arreglo circular de dos dimensiones con un espaciado angular uniforme de  $2\pi / N$ , donde  $N$  es el número de elementos del sistema como se ve en la figura:



**Figura 4.10** Arreglo Circular de Antenas en dos dimensiones.

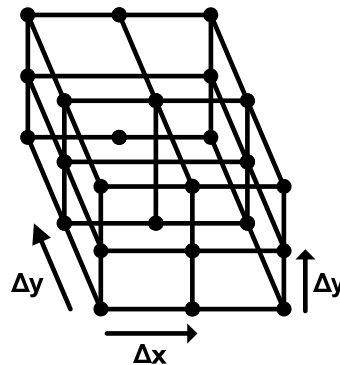
Esta estructura es comúnmente usada para la conformación del haz en dos dimensiones en el plano horizontal para ambientes externos en celdas grandes.

La tercera posible posición de los arreglos de antenas consiste en colocar los elementos en un plano de dos dimensiones donde están separados en el eje horizontal por un  $\Delta x$  y en el eje vertical por un  $\Delta y$ . Esta estructura puede realizar la conformación del haz dentro de un ángulo sólido.



**Figura 4.11** Arreglo de Antenas en un plano de dos dimensiones separados por  $\Delta x$  y  $\Delta y$ .

En el último arreglo de elementos, estos son puestos en un plano de tres dimensiones para formar una estructura cúbica o esférica donde los elementos son espaciados a  $\Delta x$ ,  $\Delta y$ ,  $\Delta z$  como se muestra en la figura:



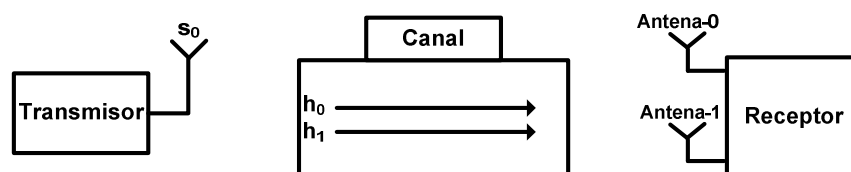
**Figura 4.12** Arreglo de antenas en tres dimensiones.

Este arreglo puede realizar la conformación del haz en el espacio entero dentro de todos los ángulos sólidos. Los elementos pueden tomar cualquier forma geométrica pero formas regulares simplifican el análisis matemático de un sistema de antenas. La potencia del transmisor es comúnmente dividida para la cantidad de antenas, es decir, las salidas de todas las antenas en el receptor son combinadas en una técnica llamada diversidad de combinación (diversity combining) para maximizar la relación de señal a ruido.

#### 4.1.3.2 Algoritmos de diversidad espacial (Alamouti)

##### Una antena en el transmisor y dos antenas en el receptor

Considerando el sistema que se muestra en la figura:



**Figura 4.13** Sistema con una antena en el Transmisor y dos antenas en el Receptor.

A determinado instante de tiempo, sea la señal enviada en banda base, desde una antena transmisora  $s_0$ . Sea  $h_0$  la respuesta impulsiva del canal de banda base complejo entre la antena transmisora y la antena receptora-0, y sea  $h_1$  la respuesta impulsiva del canal de banda base complejo entre la antena transmisora y la antena receptora-1, donde:

$$h_0 = \alpha_0 e^{j\beta_0} \quad (\text{Ec 4.45})$$

$$h_1 = \alpha_1 e^{j\beta_1} \quad (\text{Ec 4.46})$$

**Donde**  $\alpha_0$  y  $\alpha_1$  son las magnitudes de la respuesta impulsiva del canal y  $\beta_0$  y  $\beta_1$  son las fases de la respuesta impulsiva del canal.

Asumiendo que el canal presenta pequeños cambios durante el intervalo de la señal, las señales recibidas en banda base  $r_0$  y  $r_1$  están dadas por:

$$r_0 = h_0 r_0 + n_0 \quad (\text{Ec 4.47})$$

$$r_1 = h_1 r_1 + n_1 \quad (\text{Ec 4.48})$$

Donde  $n_0$  y  $n_1$  son muestras complejas de ruido gaussiano más interferencia. La distancia Euclidiana cuadrada  $d^2(x,y)$  entre dos puntos de señal compleja  $\mathbf{x}$  e  $\mathbf{y}$  está dada por:

$$d^2(x,y) = (x - y)(x^* - y^*) \quad (\text{Ec 4.49})$$

Donde “\*” denota la conjugada compleja.

Suponga que la variable  $\mathbf{y}$  toma dos valores  $\mathbf{y}_1$  y  $\mathbf{y}_2$ , luego la regla de la decisión de máxima probabilidad escoge  $\mathbf{y}_1$  si:

$$d^2(x, y_1) \leq d^2(x, y_2) \quad (\text{Ec 4.50})$$

Ahora se combinan las señales  $\mathbf{r}_0$  y  $\mathbf{r}_1$  para que aplicando la regla y asumiendo la **Ecuación 4.49** y asumiendo que la señal transmitida  $\mathbf{s}_0$  puede tomar dos posibles valores,  $\mathbf{s}_i$  y  $\mathbf{s}_k$ . Se escoge  $\mathbf{s}_k$  si y solo si:

$$d^2(r_0, h_0 s_k) + d^2(r_1, h_1 s_k) \leq d^2(r_0, h_0 s_i) + d^2(r_1, h_1 s_i) \quad (\text{Ec 4.51})$$

Sustituyendo **Ecuación 4.49** en **Ecuación 4.51**, el decodificador de máxima probabilidad estima a  $\mathbf{s}_0$  escogiendo a  $\mathbf{s}_k$  si y solo si:

$$A|s_k|^2 + d^2(\tilde{s}_0, s_k) \leq A|s_i|^2 + d^2(\tilde{s}_0, s_i) \quad (\text{Ec 4.52})$$

Donde  $A = \alpha_0^2 + \alpha_1^2 + 1$

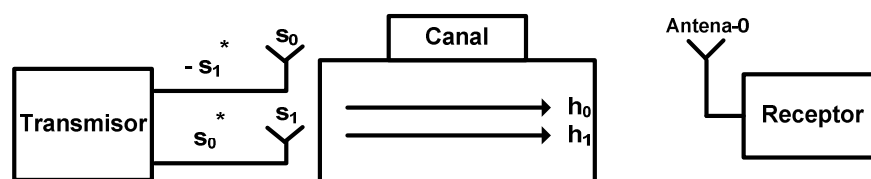
La salida del Combinador es:

$$\begin{aligned} \tilde{s}_0 &= (\alpha_0^2 + \alpha_1^2)s_0 + h_0^* n_0 + h_1^* n_1 \\ \alpha_0^2 &= |h_0|^2 \\ \alpha_1^2 &= |h_1|^2 \end{aligned} \quad (\text{Ec 4.53})$$

Claramente el algoritmo de combinación requiere que los canales  $h_0$  y  $h_1$  deben ser estimados por el receptor.

### Algoritmo de Alamouti para dos antenas en el transmisor y una antena en el receptor

Se extiende el análisis de una simple antena a dos antenas en el transmisor y una simple antena en el receptor como se muestra en la figura:



**Figura 4.14** Algoritmo de Alamouti dos antenas en el transmisor y una antena en el receptor.

Como antes, el objetivo es derivar el algoritmo óptimo combinando. A un instante de tiempo dado, el símbolo  $s_0$  y el símbolo  $s_1$  son transmitidos simultáneamente desde la antena-0 y la antena-1 respectivamente. Durante el intervalo del siguiente símbolo, el símbolo  $-s_1^*$  y el símbolo  $s_0^*$  son transmitidos



simultáneamente desde la antena-0 y la antena-1, respectivamente. Por tanto el código de bloque espacio-tiempo de Alamouti  $\mathbf{S}$  está dado por:

$$\mathbf{S} = \begin{matrix} s_0 & & s_1 \\ & \downarrow & \\ -s_1^* & \text{Tiempo} & s_0^* \end{matrix} \rightarrow \begin{matrix} \text{Espacio} \\ \\ \end{matrix} \quad (\text{Ec 4.54})$$

La matriz  $\mathbf{S}$  de transmisión Alamouti es una matriz compleja-ortogonal tanto en tiempo como en espacio. Para autenticar la ortogonalidad espacial se formula el producto:

$$\begin{aligned} \mathbf{S}\mathbf{S}^T &= \begin{matrix} s_0 & s_1 \\ -s_1^* & s_0^* \end{matrix} \times \begin{matrix} s_0^* & s_1 \\ s_1^* & s_0 \end{matrix} \\ \mathbf{S}\mathbf{S}^T &= [ |s_0|^2 + |s_1|^2 ] \times \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \end{aligned} \quad (\text{Ec 4.55})$$

$$\mathbf{S}\mathbf{S}^T = |s_0|^2 + |s_1|^2 \times \mathbf{I}$$

El producto  $\mathbf{S}\mathbf{S}^T$  en **ecuación 4.55** es igual a la matriz de identidad  $\mathbf{I}$  multiplicada por una constante  $|s_0|^2 + |s_1|^2$ .

La ortogonalidad temporal de  $\mathbf{S}$  se demuestra formulando el producto:

$$\begin{aligned} \mathbf{S}^T\mathbf{S} &= \begin{matrix} s_0^* & -s_1 \\ s_1^* & s_0 \end{matrix} \times \begin{matrix} s_0 & s_1 \\ -s_1^* & s_0^* \end{matrix} \\ \mathbf{S}^T\mathbf{S} &= \mathbf{S}\mathbf{S}^T \end{aligned} \quad (\text{Ec 4.56})$$

Donde  $\mathbf{T}$  es la traspuesta de la matriz. La señal recibida al tiempo  $\mathbf{t}$  es:

$$r_0 = h_0 s_0 + h_1 s_1 + n_0 \quad (\text{Ec 4.57})$$

La señal recibida al tiempo  $\mathbf{t} + \mathbf{T}$ , donde  $\mathbf{T}$  es la duración de símbolo, es:

$$r_1 = -h_0 s_1^* + h_1 s_0^* + n_1 \quad (\text{Ec 4.58})$$

La distancia Euclidiana cuadrada al tiempo  $\mathbf{t}$  es:

$$\begin{aligned}
d^2(r_0, h_0 s_0 + h_1 s_1) &= [r_0 - (h_0 s_0 + h_1 s_1)][r_0^* - (h_0^* s_0^* + h_1^* s_1^*)] \\
&= |r_0|^2 - r_0 h_0^* s_0^* - r_0 h_1^* s_1^* - r_0^* h_0 s_0 - r_0^* h_1 s_1 + \alpha_0^2 |s_0|^2 + \alpha_1^2 |s_1|^2
\end{aligned} \tag{Ec 4.59}$$

La distancia Euclidiana cuadrada al tiempo  $\mathbf{T} + \mathbf{t}$  es:

$$\begin{aligned}
d^2(r_1, -h_0 s_1^* + h_1 s_0^*) &= [r_1 - (-h_0 s_1^* + h_1 s_0^*)][r_1^* - (-h_0^* s_1 + h_1^* s_0)] \\
&= |r_1|^2 + r_1 h_0^* s_1 - r_1 h_1^* s_0 + r_1^* h_1 s_1^* - r_1^* h_0 s_0 + \alpha_0^2 |s_1|^2 + \alpha_1^2 |s_0|^2
\end{aligned} \tag{Ec 4.60}$$

Se agrupan los términos en **Ecuación 4.59** y en **Ecuación 4.60** asociados con  $s_0$  y se obtiene:

$$\begin{aligned}
&-r_0 h_0^* s_0^* - r_0^* h_0 s_0 + \alpha_0^2 |s_0|^2 - r_1 h_1^* s_0 - r_1^* h_1 s_0^* + \alpha_1^2 |s_0|^2 \\
&= -(r_0 h_0^* + r_1^* h_1) s_0^* - r_0^* h_0 s_0 + \alpha_0^2 |s_0|^2 - r_1 h_1^* s_0 + \alpha_1^2 |s_0|^2 \\
&= -\tilde{s}_0 s_0^* - \tilde{s}_0^* s_0 + (\alpha_0^2 + \alpha_1^2) |s_0|^2
\end{aligned} \tag{Ec 4.61}$$

Donde:

$$\tilde{s}_0 = r_0 h_0^* + r_1^* h_1 \tag{Ec 4.62}$$

Sustituyendo por  $r_0$  y  $r_1$ , la señal de salida del combinador  $\tilde{s}_0$  se simplifica a:

$$\tilde{s}_0 = (\alpha_0^2 + \alpha_1^2) s_0 + h_0^* n_0 + h_1 n_1^* \tag{Ec 4.63}$$

Similarmenete la señal de salida del Combinador  $\tilde{s}_1$  está dada por:

$$\begin{aligned}
\tilde{s}_1 &= r_0 h_1^* - r_1^* h_0 \\
\tilde{s}_1 &= (\alpha_0^2 + \alpha_1^2) s_1 - h_0^* n_1 + h_1^* n_0
\end{aligned} \tag{Ec 4.64}$$

Ahora

$$\begin{aligned}
d^2(\tilde{s}_0, s_0) &= (\tilde{s}_0 - s_0)(\tilde{s}_0^* - s_0^*) \\
d^2(\tilde{s}_0, s_0) &= |\tilde{s}_0|^2 - \tilde{s}_0 s_0 - \tilde{s}_0^* s_0^* + |s_0|^2
\end{aligned} \tag{Ec 4.65}$$

Se utiliza la **Ecuación 4.63** para simplificar la **Ecuación 4.61**:

$$(\alpha_0^2 + \alpha_1^2 - 1)|s_0|^2 + d^2(\tilde{s}_0, s_0) - |\tilde{s}_0|^2 \quad (\text{Ec 4.66})$$

El decodificador ML estima la señal  $\mathbf{s}_0$  escogiendo  $\mathbf{s}_k$  si y solo si:

$$(\alpha_0^2 + \alpha_1^2 - 1)|s_k|^2 + d^2(\tilde{s}_0, s_k) \leq (\alpha_0^2 + \alpha_1^2 - 1)|s_i|^2 + d^2(\tilde{s}_0, s_i) \quad (\text{Ec 4.67})$$

Un método similar se puede usar para estimar  $\mathbf{s}_1$ . Comparando la **Ecuación 4.63** y la **Ecuación 4.64** con **Ecuación 4.53** hay rotaciones de fase en las componentes de ruido en  $\tilde{s}_1$  las cuales no tienen efectos en el SNR combinado. Por tanto se concluye que la diversidad de transmisión de dos ramas es igual a la diversidad de recepción de dos ramas.

### Algoritmo de Alamouti para dos antenas en el Transmisor y dos antenas en el Receptor

Este esquema de arreglo de antenas es el más común. A un tiempo dado  $t$ , señales  $s_0$  y  $s_1$  son enviadas desde la antena transmisora-0 y la antena transmisora-1 simultáneamente, y las señales  $r_0$  y  $r_2$  son recibidas por las antenas receptoras Antena-0 y antena-1. Al tiempo  $T+t$ , las señales  $-s_1^*$  y  $s_0^*$  son enviadas por la antena-0 y la antena-1 simultáneamente, y la señal  $r_1$  es recibida por la antena-0 y  $r_3$  es recibida por la antena-1.

El código de transmisión de Alamuti, los canales, y las señales recibidas están dadas en las siguientes tablas.

#### Código de transmisión:

	Antena Tx-0	Antena Tx-1
Tiempo $t$	$s_0$	$s_0$
Tiempo $T+t$	$-s_1^*$	$s_0^*$

### Canales entre las antenas Tx y Rx

	Antena Rx-0	Antena Rx-1
Antena Tx-0	$h_0$	$h_2$
Antena Tx-1	$h_1$	$h_3$

### Señales Recibidas

	Rx Antena Tx-0	Rx Antena Tx-1
Antena Tx-0	$r_0$	$r_2$
Antena Tx-1	$r_1$	$r_3$

Las señales recibidas se definen como:

$$r_0 = h_0 s_0 + h_1 s_1 + n_0$$

$$r_2 = h_2 s_0 + h_3 s_1 + n_2$$

(Ec 4.68)

$$r_1 = -h_0 s_1^* + h_1 s_0^* + n_1$$

$$r_3 = -h_2 s_1^* + h_3 s_0^* + n_3$$

Formulando la distancia Euclidiana al tiempo t:

$$d^2[r_0, (h_0 s_0 + h_1 s_1)] + d^2[r_2, (h_2 s_0 + h_3 s_1)] \quad (\text{Ec 4.69})$$

Al tiempo t + T:

$$d^2[r_0, (h_0 s_0 + h_1 s_1)] + d^2[r_2, (h_2 s_0 + h_3 s_1)] \quad (\text{Ec 4.70})$$

Separando las señales  $s_0$  y  $s_1$  de las ecuaciones **4.69** y **4.70**, se puede ver la señal a la salida del Combinador es:

$$\widetilde{s}_0 = s_0 \sum_{i=0}^3 \alpha_i^2 + h_0^* n_0 + h_1 n_1^* + h_2^* n_2 + h_3 n_3^* \quad (\text{Ec 4.71})$$

$$\tilde{s}_1 = s_1 \sum_{i=0}^3 \alpha_i^2 + h_0 n_1^* + h_1^* n_0 + h_2 n_3^* + h_3^* n_2 \quad (\text{Ec 4.72})$$

El decodificador de máxima probabilidad estima a  $s_0$  y  $s_1$ , escogiendo  $s_k$  si y solo si:

$$\left[ \sum_{i=0}^3 \alpha_i^2 - 1 \right] |s_k|^2 + d^2(\tilde{s}_0, s_k) \leq (\alpha_0^2 + \alpha_1^2 - 1) |s_i|^2 + d^2(\tilde{s}_0, s_i) \quad (\text{Ec 4.73})$$

$$\left[ \sum_{i=0}^3 \alpha_i^2 - 1 \right] |s_k|^2 + d^2(\tilde{s}_1, s_k) \leq \left[ \sum_{i=0}^3 \alpha_i^2 - 1 \right] |s_i|^2 + d^2(\tilde{s}_1, s_i) \quad (\text{Ec 4.74})$$

### Análisis en banda base de un canal MIMO

Considerando un canal Múltiple Transmisor- Múltiple Receptor (MIMO) con  $n_t$  antenas transmisoras y  $n_r$  antenas receptoras. La señal compleja transmitida  $s$  a un determinado tiempo discreto, es representada por un vector de  $n_t$  elementos:

$$s = [s_1, s_2, \dots, s_{n_t}]^T \quad (\text{Ec 4.75})$$

Los elementos de  $s$  tienen una media cero y una varianza  $\sigma_s^2$ , así que la potencia transmitida  $P$  está dada por:

$$P = n_t \sigma_s^2 \quad (\text{Ec 4.76})$$

La matriz de correlación  $\mathbf{R}_s$  del vector transmitido  $s$  es una matriz  $n_t$  por  $n_t$  dada por:

$$R_s = E[s s^T] = \sigma_s^2 I_{n_t} \quad (\text{Ec 4.77})$$

Donde  $I_{n_t}$  es una matriz identidad de  $n_t$  por  $n_t$ .

La señal compleja recibida  $r$  en un tiempo dado es expresado por el vector dado por:

$$r = [r_1, r_2, \dots, r_{n_t}]^T \quad (\text{Ec 4.78})$$

Denota el desvanecimiento plano del canal Gaussiano entre la antena transmisora  $i$  y la antena receptora  $k$  en un tiempo discreto dado  $h_{ki}$  donde:  $i = 1, 2, \dots, n_t$  y  $k = 1, 2, \dots, n_r$ . El canal complejo  $n_t$  por  $n_r$  puede ser representada por una matriz **H**:

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,n_t} \\ h_{2,1} & h_{2,2} & \dots & h_{2,n_t} \\ \dots & \dots & \ddots & \dots \\ h_{n_r,1} & h_{n_r,2} & \dots & h_{n_r,n_t} \end{bmatrix} \quad (\text{Ec 4.79})$$

Cada elemento complejo **H** está expresado por dos componentes reales (uno para la parte real y otro para la parte imaginaria), cada uno con una media cero y una derivación estándar  $\frac{1}{\sqrt{2}}$  como se puede ver:

$$h_{ik} = n \left( 0, \frac{1}{\sqrt{2}} \right) + jn \left( 0, \frac{1}{\sqrt{2}} \right) \quad (\text{Ec 4.80})$$

Donde  $n(x,x)$  representa una distribución Gaussiana real, así que la amplitud de cada elemento es Rayleigh distribuido. Los elementos de **H** tienen un media cero y una varianza unitaria. El canal complejo de ruido puede ser expresado por un vector  $n_r$  por 1:

$$n = [n_1, n_2, \dots, n_{n_r}]^T \quad (\text{Ec 4.81})$$

Donde los elementos de **n** son variables Gaussianas complejas con media cero y varianza  $\sigma_n^2$ . La matriz de correlación **R<sub>n</sub>** de un canal ruidoso está dado por:

$$R_n = E[n n^T] = \sigma_n^2 I_{n_r} \quad (\text{Ec 4.82})$$

Donde  $I_{n_r}$  es una matriz identidad de  $n_r$  por  $n_r$ .

La señal compleja recibida  $r_i$  de la  $i^{\text{th}}$  antena debido al  $s_k$  símbolo transmitido por la  $k^{\text{th}}$  antena:

$$r_l = \sum_{m=1}^{n_t} h_{lm} s_m + n_l \quad (\text{Ec 4.83})$$

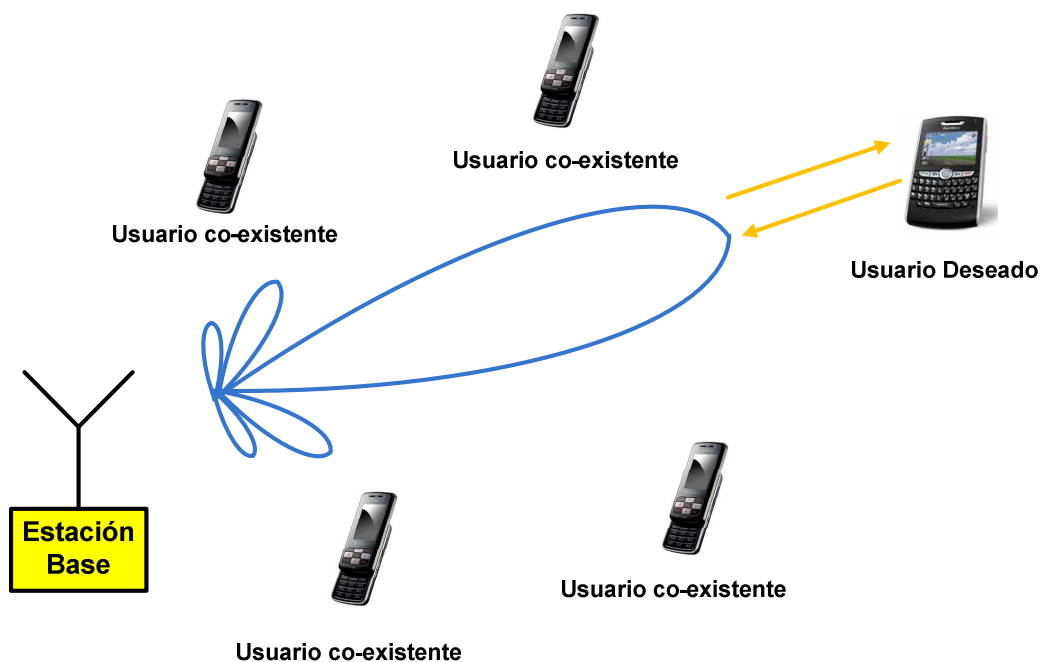
Por lo tanto, el modelo del canal complejo MIMO puede ser expresado en forma de matriz:

$$r = Hs + n \quad (\text{Ec 4.84})$$

#### 4.1.3.3 Técnicas de conformación del Haz

La tecnología de conformación del Haz ofrece una solución significativa para reducir los niveles de interferencia. El principio de esta tecnología es transmitido o recibido por la estación base solo en la dirección del usuario deseado. Por lo tanto, la interferencia conjunta es reducida en el sistema.

En la figura se muestra un sistema de conformación de haz, donde, cada señal de usuario es multiplicada por los pesos complejos los cuales ajustan la magnitud y la fase de la señal desde y hasta cada antena, así que la salida desde las antenas del transmisor/receptor forman el haz en una dirección requerida y reduce la interferencia en las otras direcciones.



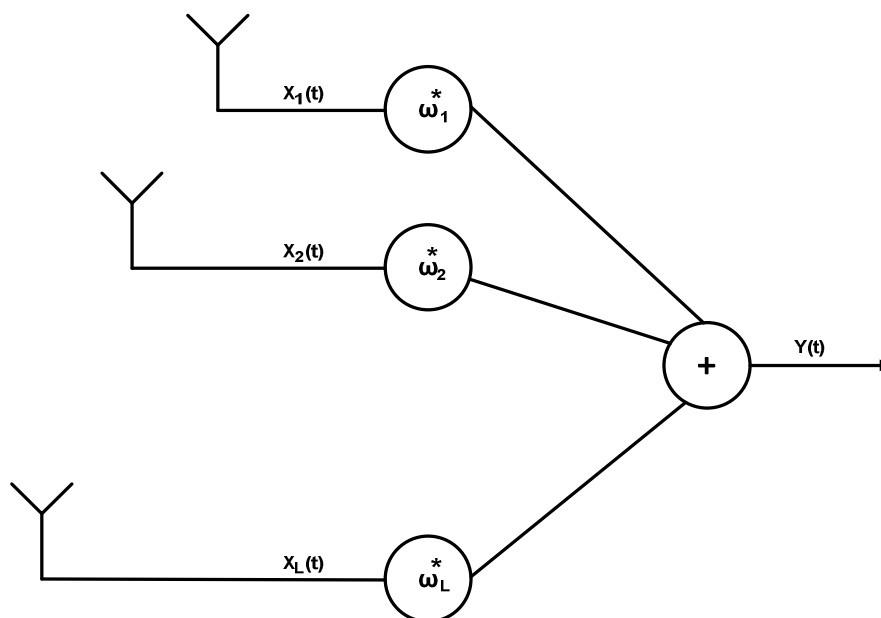
**Figura 4.15** Sistema de conformación de haz.

### Sistema básico de conformación de Haz

Se considera un arreglo de  $L$  receptores omnidireccionales y  $M$  antenas transmisoras direccionales. Sea  $x_l(t)$  el cual consiste de la señal recibida  $m(t)$  más ruido de fondo e interferencia  $n_l(t)$ .

$$x_l(t) = m(t) + n_l(t) \quad (\text{Ec 4.85})$$

La siguiente figura se muestra un formador de haz simple, donde la señal recibida de cada antena es multiplicada por un peso complejo  $\omega_l$  y sumada para formar el arreglo de salida  $y(t)$ .



**Figura 4.16** Diagrama de una estructura simple de conformador de Haz.

$$y(t) = \sum_{l=1}^L \omega_l^* x_l(t) \quad (\text{Ec 4.86})$$

Donde  $**$  denota la conjugada compleja.

El vector de pesos de un conformador de haz puede ser expresado como:



$$W = [\omega_1, \omega_2, \dots, \omega_L(t)]^T \quad (\text{Ec 4.88})$$

Las señales complejas a la salida de las antenas receptoras  $\mathbf{x}(t)$  pueden ser como:

$$X = [x_1(t), x_2(t), \dots, x_L(t)]^T \quad (\text{Ec 4.88})$$

Por lo tanto puede ser expresado en forma vectorial como:

$$y = W^H X \quad (\text{Ec 4.89})$$

Para un vector de pesos dado  $\mathbf{W}$ , la potencia recibida  $P(\mathbf{W})$  está dada por:

$$P(W) = E[y(t)y^*(t)] = W^H R W \quad (\text{Ec 4.90})$$

Donde  $\mathbf{R}$  es una matriz de correlación de las señales recibidas de orden L:

$$R = E[XX^H] \quad (\text{Ec 4.91})$$

Los elementos de la matriz R denotan la correlación entre varias señales recibidas  $\mathbf{X}$ . Por ejemplo,  $R_{ij}$  denota la correlación entre las señales recibidas de las antenas  $i^{\text{th}}$  y  $j^{\text{th}}$ .

### El conformador de haz de retardo y suma

En este esquema, los elementos de  $\mathbf{W}$  son de igual magnitud y las fases son seleccionadas para dirigir el arreglo recibido en la dirección de la llegada de la señal de las antenas transmisoras. Un conformador de haz de retraso y suma de dos antenas se muestra a continuación:

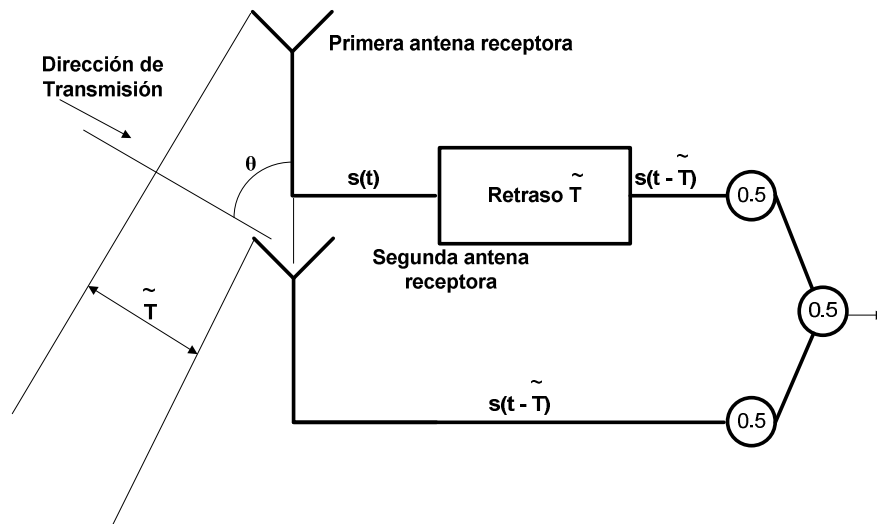


Figura 4.17 Conformador de haz de retardo y suma

Cada señal recibida es multiplicada por un factor 0.5 de manera que la ganancia total en la dirección de la señal  $\theta$  es igual a la unidad.

Las dos antenas están separadas por una distancia  $d$ . Se asume que la señal transmitida está llegando de la dirección  $\theta$  y que la señal recibida a la salida de la primera antena es  $s(t)$ . La señal en la segunda antena llega después de  $\tilde{T}$  segundos donde  $\tilde{T}$  está dada por:

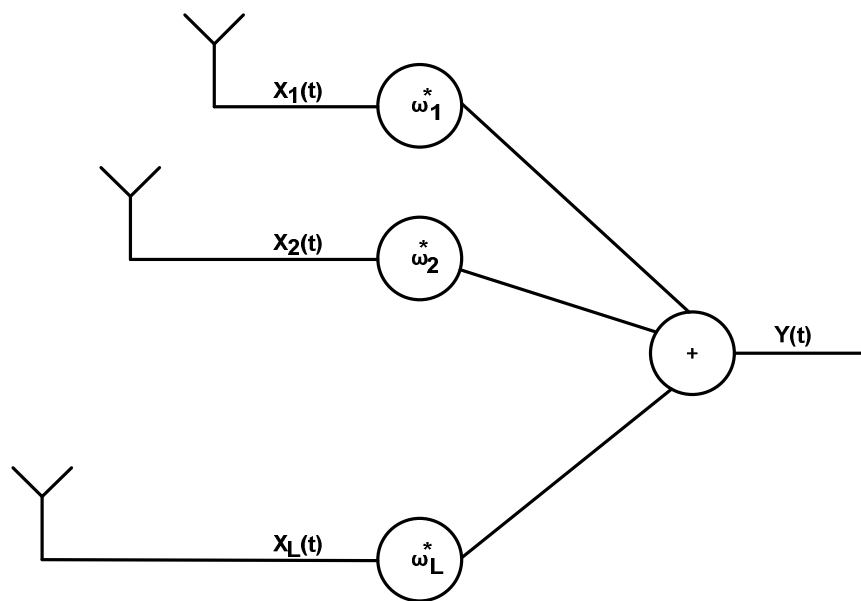
$$\tilde{T} = \frac{d}{c} \cos\theta \quad (\text{Ec 4.92})$$

Donde  $c$  es la velocidad de la luz. Por lo tanto, la señal a la salida de la segunda antena es  $s(t - \tilde{T})$ . Se retrasa la señal de la primera antena para producir que  $s(t - \tilde{T})$  esté en fase con la señal de la segunda antena, se multiplican ambas señales con un factor de ajuste 0.5 y se suman para producir una señal de salida  $s(t - \tilde{T})$ . Este conformador de haz reduce la potencia del ruido no correlacionado en el arreglo de salida y produce una ganancia en SNR igual a  $L$ . Además provee un máximo SNR de salida cuando no hay interferencias direccionales operando a la misma frecuencia. Sin embargo no es efectivo en la presencia de interferencias direccionales.

### Conformador de haz direccionable nulo (Null steering beam former)

El conformador de haz produce nulo en el patrón de respuesta en la dirección de llegada de la señal interferente, cancelando la interferencia que llega de esa dirección. Una versión anterior de este esquema sustrae la interferencia estimada usando un conformador de retraso y suma, el cual es efectivo para cancelar interferencias fuertes. Sin embargo se vuelve pesado mientras aumenta el número de señales interferentes en aplicaciones como comunicaciones móviles celulares.

Un haz en la dirección deseada y nulo en las direcciones de interferencia puede ser formado por elecciones apropiadas de los pesos en la conformación de haz mostrada en la figura:



**Figura 4.18** Diagrama de una estructura simple de conformador de Haz.

Define los vectores de direccionamiento complejos de tal manera que asociadas con la  $i$ -ésima dirección de la antena de transmisión por un vector complejo  $s_i$ :

$$s_i = [e^{j\omega_0\tau_1}, e^{j\omega_0\tau_2}, \dots, e^{j\omega_0\tau_L}] \quad (\text{Ec 4.93})$$

Donde  $\omega_0$  es la frecuencia en radianes de la señal deseada, y  $\tau_i$  es el retraso de tiempo medido de la antena receptora de referencia al origen para señal que llega de la  $i$ -ésima antena de transmisión. El principio detrás del **null beam former** es escoger los pesos que hacen a la respuesta del receptor en la dirección deseada (vector direccionado  $s_1$ ), unidad, y la respuesta en las otras ( $L - 1$ ) direcciones (vectores direccionados  $s_2, s_3, \dots, s_{L-1}$ ), nula. Se pueden expresar estas limitaciones por las siguientes expresiones:

$$W^H s_1 = 1 \quad (\text{Ec 4.94})$$

$$W^H s_i = 0 \quad i = 2, 3, \dots, L - 1 \quad (\text{Ec 4.95})$$

Combinando **Ecuación 4.94** y **Ecuación 4.95** en una matriz expresada:

$$W^H A = \zeta^T \quad (\text{Ec 4.96})$$

Donde:

$$\zeta^T = [1, 0, 0, \dots, \dots, \dots]^T \quad (\text{Ec 4.97})$$

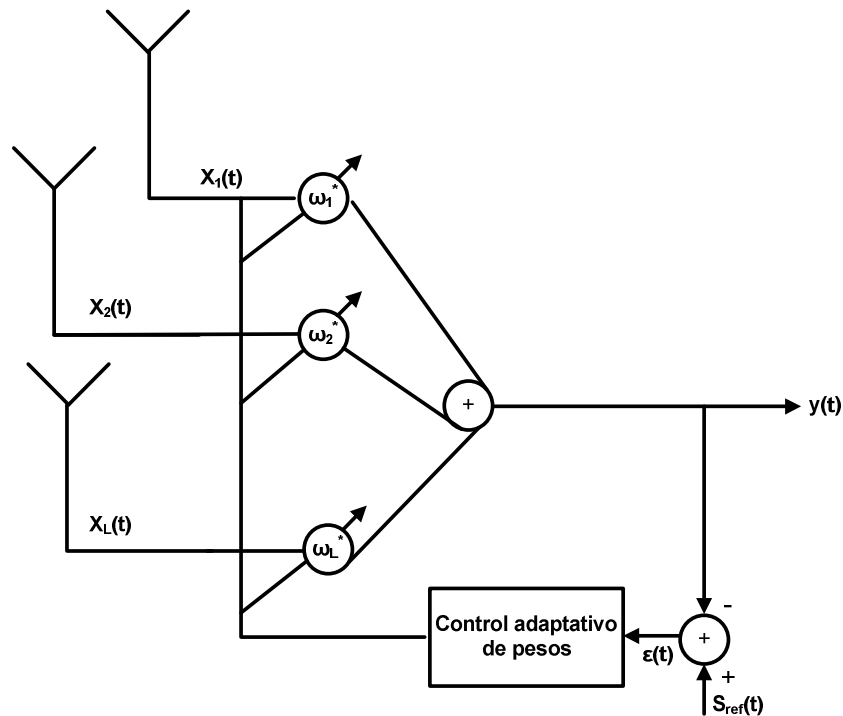
$$A = [s_0, s_1, s_2, s_3, \dots, \dots, s_{L-1}] \quad (\text{Ec 4.98})$$

Consecuentemente:

$$W^H = \zeta^T A^{-1} \quad (\text{Ec 4.99})$$

### Optimización del conformador de haz usando una señal de referencia

Un simple esquema para optimizar los pesos de las antenas se muestra a continuación el cual se basa en el conformador de haz básico con un lazo cerrado añadido para optimización adaptativa de los pesos usando la señal de referencia  $s_{ref}(t)$ .



**Figura 4.19** Conformador de haz adaptativo.

La señal de error  $\varepsilon(t)$  es:

$$\varepsilon(t) = s_{ref}(t) - W^H X \quad (\text{Ec 4.100})$$

El error cuadrado medio (mean square error, MSE) usado para ajustar los pesos está dado por:

$$MSE = E[|\varepsilon(t)|^2]$$

$$MSE = E[|s_{ref}(t)|^2] - 2W^H E[X(t)s_{ref}(t)] + W^H X X^H W$$

$$MSE = E[|s_{ref}(t)|^2] - 2W^H z + W^H R W \quad (\text{Ec 4.101})$$

Donde:

$$z = E[X(t)s_{ref}(t)]$$

$R = [XX^H]$  se explicó anteriormente y  $\mathbf{z}$  es la correlación entre  $\mathbf{X}$  y la señal de referencia  $\mathbf{s}_{ref}(t)$ . Minimización del MSE es dado por la ecuación para el vector de peso óptimo:

$$\tilde{\mathbf{W}} = R^{-1}\mathbf{z} \quad (\text{Ec 4.102})$$

El MSE mínimo (MMSE) entre el arreglo de salida y la señal de referencia utilizando los pesos óptimos es:

$$MMSE = E \left[ |s_{ref}(t)|^2 \right] - \mathbf{z}^H R^{-1}\mathbf{z} \quad (\text{Ec 4.103})$$

Un parámetro importante que contribuye al diseño del conformador de haz es el estimado de la dirección de Llegada (Direction of arrival, DOA)  $\theta$  de la señal deseada. Uno de los esquemas más simples para la estimación del DOA es el método de estimación espectral. El proceso es similar a dirigir mecánicamente las antenas receptoras en dirección ( $\theta$ ) y medir la potencia de salida. Sin embargo debido a lóbulos laterales, la potencia medida no solo es formada por la dirección en la cual las antenas son direccionadas, pero también de direcciones donde los lóbulos laterales están apuntando. La cantidad de potencia interferente recibida de los lóbulos laterales depende en la apertura del arreglo y el ancho del lóbulo principal.

#### 4.1.4 TÉCNICAS DE CANCELACIÓN DE INTERFERENCIA (IC)<sup>90 91</sup>

Las técnicas de cancelación están basadas en el principio de que es posible remover la interferencia de múltiple acceso de cada señal recibida de los usuarios antes de tomar decisiones de datos. Las técnicas IC pueden ser agrupadas en dos categorías:

---

<sup>90</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 453-458.

<sup>91</sup> Multiuser Detection for CDMA Systems, Duel-Hallen, Alexandra; Holtzman, Jack; Zvonar, Zoran.

- **Cancelación Sucesiva de Interferencia** donde la interferencia es cancelada serialmente y en etapas comenzando con la interferencia más fuerte.
- **Cancelación de Interferencia Paralela** la cual se logra cancelando la interferencia de todos los usuarios simultáneamente y puede ser llevada a cabo en múltiples etapas. La principal etapa que involucran los esquemas de Cancelación de Interferencia son la estimación de las amplitudes de las señales recibidas (Energías) del usuario activo, la regeneración de las señales adecuadas de interferencia y la sustracción de la interferencia de la señal recibida.

Ambos esquemas de Cancelación de Interferencia utilizan el filtro adaptado convencional como su primera etapa de detección. Hay que destacar que las técnicas de IC tratan con eficacia a la interferencia dentro de las celdas y no tienen ningún control sobre las interferencias procedentes de otras celdas (interferencia entra celdas) ya que estas fuentes de interferencia no pueden ser controladas por la estación base y en consecuencia no se pueden cancelar.

#### 4.1.4.1 Cancelación Sucesiva de Interferencia

El esquema de cancelación sucesiva es más simple de implementar en Hardware que el esquema de cancelación paralela, pero es más robusta en cancelación de interferencia ya que resta sucesivamente la interferencia de la señal recibida. La interferencia más fuerte puede ser borrada en presencia de la interferencia más débil, y es removida de la señal recibida mejorando la detección de las señales más débiles.

Esto implica que primeramente se debe clasificar las señales de acuerdo a su potencia:

$$\sqrt{E_1} > \sqrt{E_2} > \sqrt{E_3} \dots \dots \dots > \sqrt{E_K} \quad (\text{Ec 4.104})$$

Donde  $E_K$  es la energía del  $k^{\text{th}}$  usuario.

Una vez que los rangos de las señales recibidas son alcanzados, se puede lograr la detección de la señal del usuario 1 usando un receptor de filtro adaptado convencional.

Sea el símbolo detectado del usuario 1,  $\widehat{b}_1(t)$  el cual será utilizado para generar la señal de usuario 1  $\widehat{x}_1$  como:

$$\widehat{x}_1(t) = \sqrt{E_1} \widehat{b}_1(t) C_1(t) \quad (\text{Ec 4.105})$$

Donde  $C_1(t)$  la forma de onda forma de onda del usuario.

Sustraendo  $\widehat{x}_1(t)$  de la señal recibida  $r(t)$ , se obtiene la entrada para la siguiente etapa para la Cancelación de Interferencia:

$$r(t) - \widehat{x}_1(t)$$

Repetiendo este procedimiento con la segunda señal de usuario más fuerte se permite la detección de  $\widehat{b}_2(t)$  y la generación de la señal realimentada  $\widehat{x}_2(t)$ . La entrada de la siguiente etapa de la cancelación de interferencia será:

$$r(t) - \widehat{x}_1(t) - \widehat{x}_2(t) \quad (\text{Ec 4.106})$$

El procedimiento puede ser repetido hasta el usuario con la señal de usuario más débil  $\widehat{b}_K(t)$  sea detectada. Así la entrada a la etapa final es:

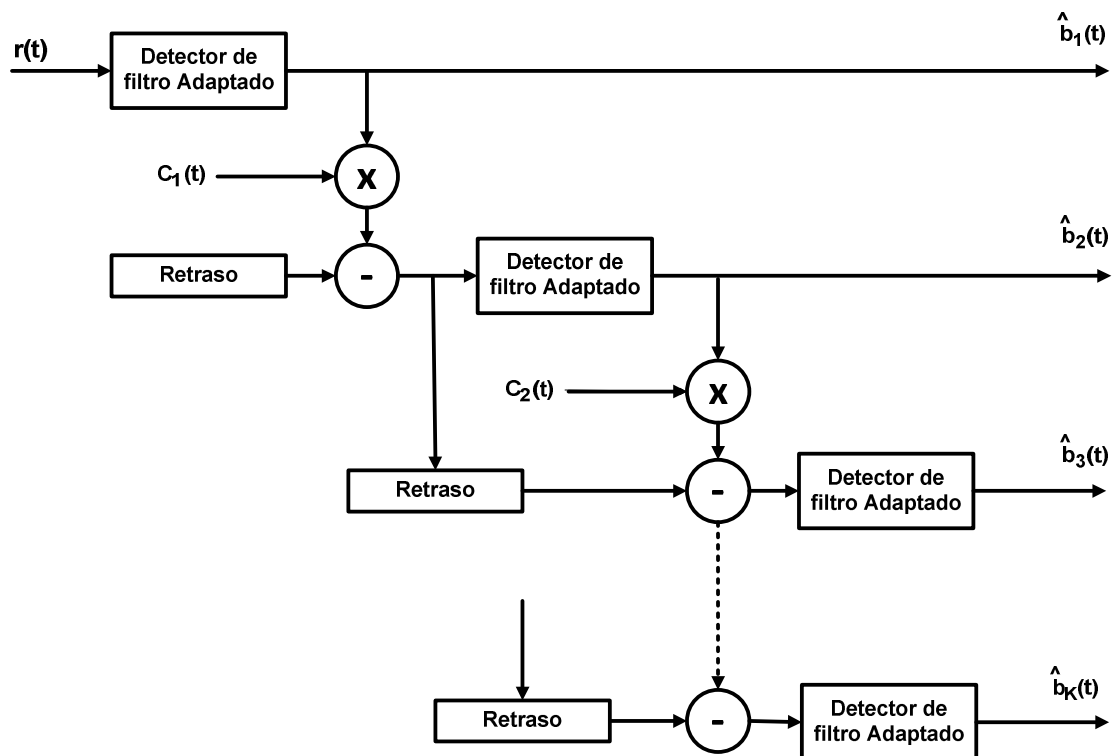
$$r(t) - \sum_{i=1}^{K-1} \widehat{x}_i(t) \quad (\text{Ec 4.107})$$

La complejidad en el cálculo del algoritmo de la Cancelación sucesiva de Interferencia es lineal en el número de usuarios. Este esquema, aunque simple, tiene desventajas, la más destacada de ellas es que, debido al proceso serial de la cancelación de interferencia, se requiere de un retraso en el cálculo de las  $K$  etapas requeridas para completar la detección multiusuario.



Por lo tanto, una detección errónea de cualquier etapa se realimentará causando que el error se propague en las etapas siguientes, haciendo que el nivel de interferencia se incremente en lugar de disminuir.

Otra desventaja es que el primer usuario a ser procesado, ve la interferencia de los  $K-1$  usuarios mientras que cada usuario de las etapas inferiores ve menos interferencias, por lo que la calidad de servicio en el sistema no es el mismo.



**Figura 4.20** Vista esquemática de la Cancelación Sucesiva de Interferencia.

#### 4.1.4.2 Cancelación de Interferencia Paralela (Parallel Interference Cancellation, PIC)

Este esquema logra procesamiento paralelo de la interferencia de acceso y remueve la interferencia de todos los usuarios simultáneamente. Ya que la cancelación de interferencia se realiza en paralelo, el retardo requerido para la eliminación de interferencia es, a lo sumo, de la duración de unos pocos bits. Para cancelar la interferencia, un estimado de la interferencia es requerido.

Sin embargo este estimado es pobre en las primeras etapas del proceso PIC multi etapa. Por lo tanto es preferible utilizar Cancelación de interferencia Parcial e incrementar la porción de la cancelación de interferencia como la estimación de la interferencia mejora en las próximas etapas. En el esquema iterativo paralelo, cada etapa de la iteración produce un estimado mejor y nuevo de los bits del usuario basado en aquellos obtenidos en las etapas previas, lo que mejora los estimados de interferencia.

La cancelación de interferencia paralela multi etapa se muestra a continuación:

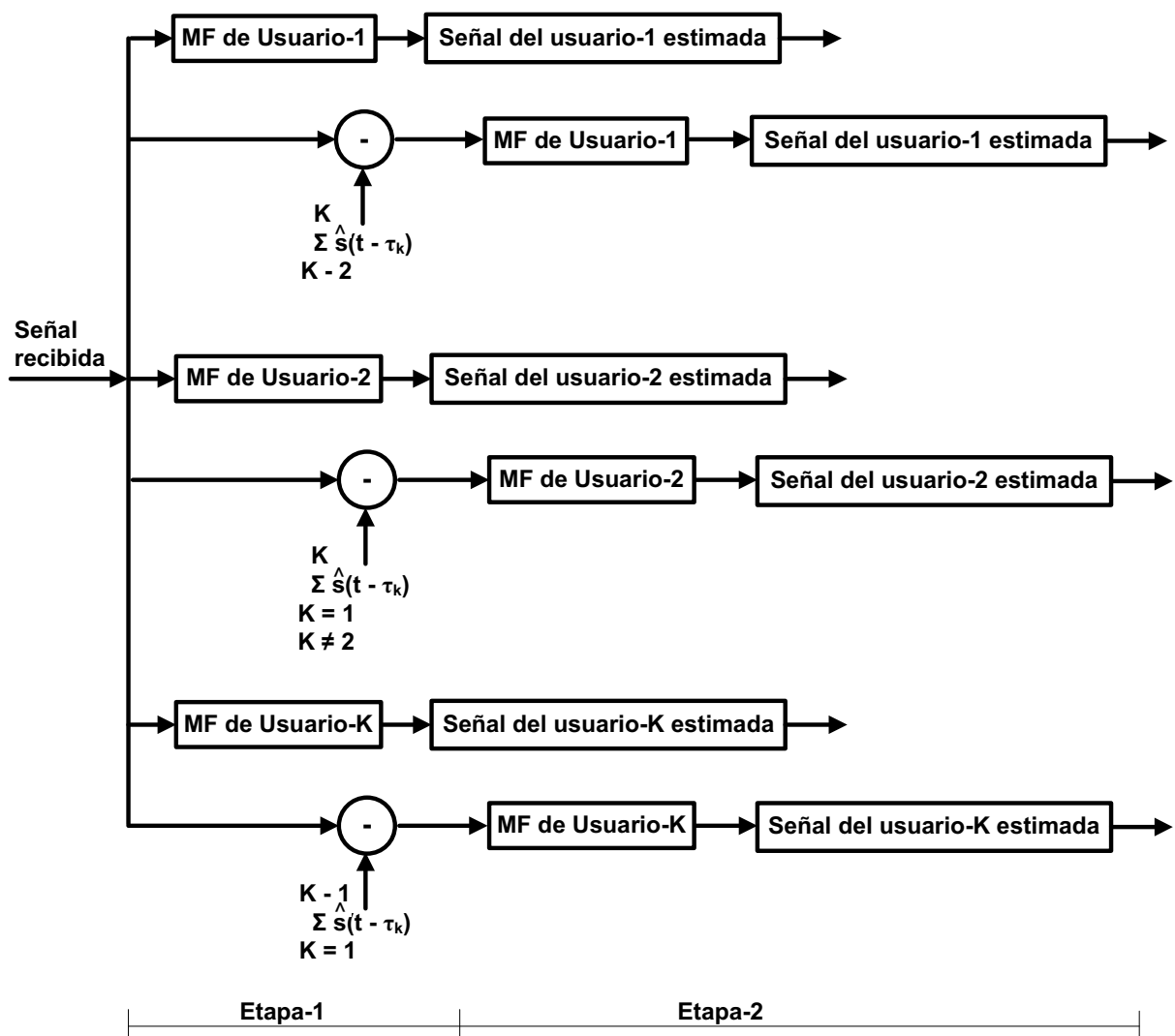


Figura 4.21 Cancelación de Interferencia Paralela.

## 4.2 ASPECTOS BÁSICOS DEL ESTÁNDAR IS-95A<sup>92</sup>

El estándar IS-95 es uno de los mayores logros de la industria celular, ya que es la primera aplicación de la tecnología CDMA en sistemas de comunicación para civiles. La empresa Qualcomm ha realizado grandes contribuciones al desarrollo de los sistemas de IS-95.

El estándar IS-95 fue primeramente publicado en Julio de 1993. La revisión IS-95A fue publicada en Mayo de 1995 y es la base para muchos de los sistemas comerciales CDMA 2G.

IS-95A describe la estructura de canales CDMA de banda ancha de 1.25 MHz, control de potencia, procesado de llamadas, hand-offs, servicios de voz, y técnicas de registro para operación del sistema.

La revisión IS-95B combina IS-95A, ANSI JSTD-008 y TSB-74 en un mismo estándar. Generalmente, con IS-95B se ofrece datos a 64 Kbps a través de conmutación de paquetes, además de servicios de voz. En 1999 fue instalada una red celular basada en IS-95B en Corea.

La especificación ANSI JSTD-008 publicada en 1995, define un estándar de compatibilidad para sistemas PCS CDMA de 1.8-2 GHz. TSB-74 describe la interacción entre IS-95A y sistemas PCS CDMA que conforman la especificación ANSI JSTD-008.

CdmaOne se utiliza como nombre colectivo dado a IS-95A, ANSI JSTD-008 y IS-95B.

IS-95A y ANSI JSTD-008, son muy parecidos excepto por la banda de operación y unos pocos protocolos de mensaje.

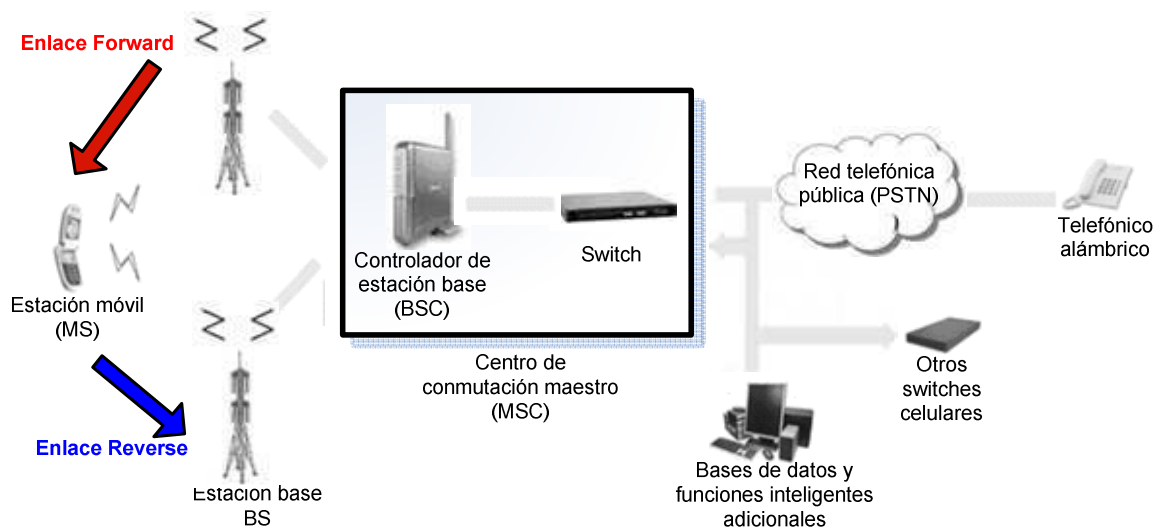
---

<sup>92</sup> The Next Generation CDMA Technologies, Chen, Hsiao-Hwa, p. 76-80.

IS-95A opera en la banda celular (824-894 MHz) y ANSI JSTD-008 opera en la banda PCS (1850-1990 MHz). Existen otras diferencias en las especificaciones técnicas.

IS-95A utiliza un espaciado de 45 MHz para los canales forward y reverse. ANSI JSTD-008 utiliza un espaciado de 80 MHz.

#### 4.2.1 CONFIGURACIÓN DE UNA RED IS-95A



**Figura 4.22** Sistema de Red Celular IS-95A.

Un sistema IS-95 consiste de los siguientes bloques: estación móvil, estaciones base, controlador de estaciones base, switch, etc.

Una estación móvil (movil station, MS) o teléfono celular es de dimensiones pequeñas. Utiliza una antena para transmisión y recepción. El consumo de potencia debe ser mínimo y controlado debido a que el móvil funciona con una batería.



**Figura 4.23** Estación Móvil (MS)

Una estación base (base station, BS) es el lugar donde las señales de radio desde y hacia la estación móvil son procesadas. Puede variar en sus dimensiones y ser alimentada por 100-240 volt AC. La BS se coloca en el centro de la celda para proveer cobertura a esta. Para mayor cobertura se puede dividir la celda en tres o más sectores, para lo cual se utilizan varias antenas para dar cobertura a diferentes sectores.



**Figura 4.24** Estación Base.

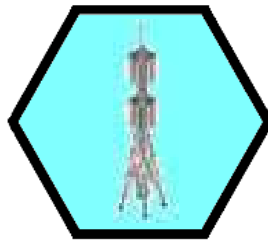
El controlador de estaciones base (base station controller, BSC), el cual trabaja en conjunto a un switch para realizar tareas como registro de ubicación del móvil, soft handoff, paging del móvil, etc. La combinación del switch y la BSC se llama Centro de Conmutación Maestro (master switching center, MSC).



**Figura 4.25** Centro de Conmutación Maestro (MSC).

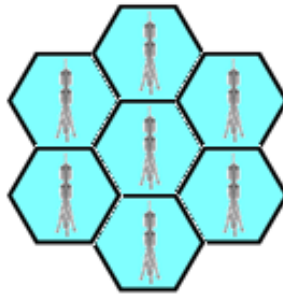
Como parte de la red se toma en cuenta al enlace entre el MSC y la red telefónica pública (PSTN), además del enlace con otros switches celulares de otros proveedores.

Toda el área geográfica de la red está dividida en espacios llamados celdas.



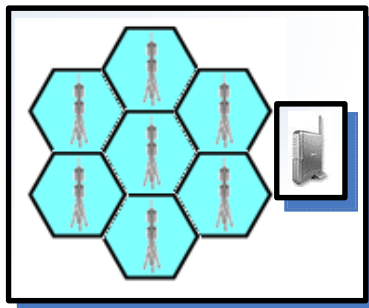
**Figura 4.26** Celda IS-95A.

Una BS da servicio a una de las celdas. Las celdas generalmente se agrupan en clusters de 3 a 7 celdas.



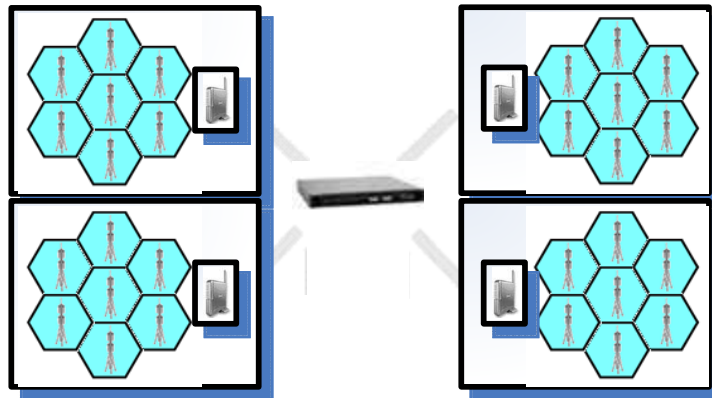
**Figura 4.27** Cluster IS-95A.

Una BSC se utiliza con un grupo de clusters.



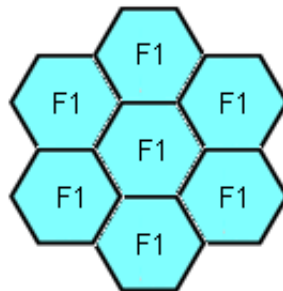
**Figura 4.28** BSC y Cluster.

Todas los BSCs de una red son controladas por una MSC.



**Figura 4.29** BSCs y Switch (MSC)

Con CDMA se puede asignar el espectro entero a las siete celdas de un cluster, gracias al uso de códigos ortogonales en la transmisión de datos. Esto permite una mejora de la calidad de transmisión. La interferencia por adyacencia de canal se controla por control de potencia.



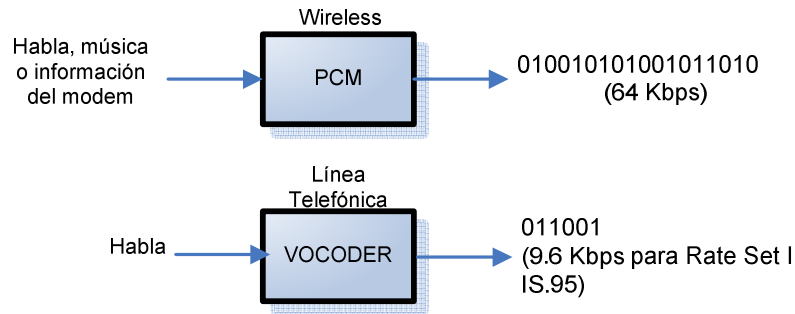
**Figura 4.30** Reuso de códigos en un Cluster.

#### 4.2.2 VOCODER <sup>93</sup>

En un sistema celular, la voz humana debe ser codificada de forma digital para ser transmitida. Técnicas de digitalización como PCM no son viables en aplicaciones inalámbricas debido a que requieren un ancho de banda constante y mayor para lo que puede ofrecer el medio inalámbrico. Técnicas de digitalización alternativas son necesarias para representar a la voz humana que utilicen menos

<sup>93</sup> CDMA RF System Engineering, Yang, Samuel C., p. 33-34.

ancho de banda. Un vocoder explota las características del habla humana y utiliza menor número de bits para representarla.



**Figura 4.31** Comparación entre PCM y Vocoder.

Es así que el vocoder es un analizador y sintetizador de voz. El espectro de la voz humana es típicamente estacionario sobre un periodo entre 20 a 40 ms. Por esto, la mayoría de vocoders producen tramas con esta duración. Los vocoders de IS-95 producen tramas con una duración de 20 ms, además que son de tasas variables: durante periodos de silencio, la tasa de bits de salida del vocoder es disminuida. El vocoder de IS-95 soporta cuatro tasas de transmisión. Esto reduce el ancho de banda efectivo para transmitir habla y una reducción de ancho de banda canales individuales, se incrementa el número de canales de voz disponibles.

### 4.2.3 CÓDIGOS UTILIZADOS EN IS95-A

En IS-95A se utilizan tres tipos de códigos: Códigos de Walsh, Códigos PN Cortos, y Códigos PN largos.

#### 4.2.3.1 Códigos de Walsh

Los códigos de Walsh, como se mencionó en capítulos anteriores, ofrecen una ortogonalidad ideal cuando se utilizan en canales sincrónicos, como el canal forward de IS-95A.

IS-95A e IS-95B utiliza códigos de Walsh de longitud 64, mientras que cdma2000 utiliza códigos de Walsh de longitud 128.



Los códigos de Walsh se utilizan en el canal forward para ensanchamiento y asignación de canal, por lo que el enlace forward puede ser dividido en 64 canales de código. En el enlace reverse, los códigos de Walsh se utilizan para realizar una modulación 64-ava, en la cual cada 6 bits entrantes de datos son mapeados en uno de los 64 códigos de Walsh, dependiendo del patrón particular de los 6 bits.

#### 4.2.3.2 Códigos PN cortos

Los códigos PN cortos se utilizan para identificar a las estaciones base (BS) y por tanto a las celdas donde las BSs están ubicadas. En el estándar IS-95 el código corto es un par de secuencias periódicas PN binarias con un período de  $2^{15}$ . Las diferentes BSs son diferenciadas asignando un Offset del código a cada BS dentro de una referencia de tiempo común en toda la red. En el enlace Reverse, los móviles utilizan el código PN corto para dar robustez a la señal sin ningún Offset. Para obtener la sincronización de los Offsets de las BSs en la red, se utilizan servicios GPS (Global Positioning System). Los polinomios generadores son, para la secuencia en fase  $P_I(x) = x^{15} + x^{13} + x^9 + x^8 + x^7 + x^5 + 1$  y para la secuencia en cuadratura  $P_Q(x) = x^{15} + x^{12} + x^{11} + x^{10} + x^6 + x^5 + x^4 + x^3 + 1$ .

Los polinomios especificados generan secuencias de longitud  $2^{15} - 1$ . Cada 14 ceros consecutivos en cada periodo de los códigos I y Q un cero es insertado después de estos para producir una secuencia de 15 ceros consecutivos, haciendo el periodo resultante de la secuencia igual a  $2^{15}$ .

#### 4.2.3.3 Códigos PN Largos

El código PN largo es una secuencia con un período de  $2^{42}-1$ . En el enlace Forward se utiliza para realizar el scrambling de los datos. En el enlace Reverse se utiliza para realizar el ensanchamiento de la señal, además como código de ensanchamiento para identificar diferentes móviles.

El polinomio generador utilizado en IS-95A es:

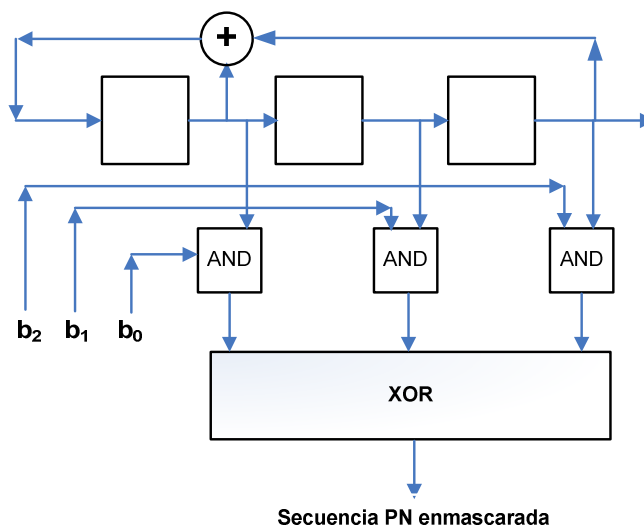
$$P(x) = x^{42} + x^{35} + x^{33} + x^{31} + x^{27} + x^{26} + x^{25} + x^{22} + x^{21} + x^{19} + x^{18} + x^{17} + x^{16} + x^{10} + x^7 + x^6 + x^5 + x^3 + x^2 + x + 1.$$

#### 4.2.3.4 Enmascaramiento de secuencias PN.<sup>94</sup>

En el estándar IS-95, las secuencias PN son enmascaradas para dar un Offset a las mismas. Para explicar el enmascaramiento y el desplazamiento que este genera a la secuencia PN se utiliza el siguiente ejemplo:

##### Ejemplo:

Considerando el polinomio generador  $P(x) = x^3 + x + 1$ , se conoce que su periodo es  $2^3 - 1 = 7$ . La secuencia PN con desplazamiento cero consiste de 7 bits. Una máscara de código de 3 bits desde 000 a 111 debe generar las 8 posibles fases de la secuencia PN. El generador de la máscara es el siguiente:



**Figura 4.32** Generador de código PN enmascarado.

Siendo el estado inicial del generador 001. Se puede mostrar el generador sin enmascarado, y luego de siete ciclos de reloj desde su estado inicial, se obtiene la secuencia 1001011. Con más ciclos de reloj, la secuencia se repetirá. Ahora,

<sup>94</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 487-489.

operando el generador con los bits de máscara y los estados mostrados. Las secuencias PN generadas son:

**Tabla 4.1** Máscara y secuencia PN generada.

<b>Máscara</b>	<b>Secuencia PN</b>
001	0101110 01. .
100	1011100 10. . .
110	1100101 11. . .
111	1001011 10. . .
011	0010111 00. . .
101	111010 11. . .
010	0111001 01. . .

De esta tabla, se puede observar que las secuencias PN enmascaradas, son versiones desplazadas de la secuencia original. Este método de enmascaramiento se utiliza tanto para las secuencias PN largas, como para las secuencias PNI y PNQ cortas para tener cierto desplazamiento (offset) utilizando una máscara de 42 y 15 bits respectivamente.

#### 4.2.4 TASAS DE TRANSMISIÓN Y RATE SETS EN IS-95A

Los sistemas IS-95 permiten transmitir datos a 4 diferentes tasas. Estas 4 tasas forman un Rate Set (conjunto de tasas). Para IS-95 existen dos Rate Sets: Rate Set I y Rate Set II. El segundo ofrece una mayor velocidad y mejor calidad de voz.

**Tabla 4.2** Tasas de Transmisión en IS-95A.

<b>Tasas de transmisión Rate Set I (bps).</b>	<b>Tasas de transmisión Rate Set II (bps).</b>
9600 (full rate)	14400 (full rate)
4800 (half rate)	7200 (half rate)
2400 (quarter rate)	3600 (quarter rate)
1200 (eighth rate)	1800 (eighth rate)

#### 4.2.5 POTENCIA TOTAL EN EL ENLACE FORWARD

La potencia de transmisión depende del fabricante y la regulación de cada país. La potencia típica total del enlace Forward es de 8 watts o 39 dBm. Esto produce una densidad espectral sobre el canal de 1,25MHz de -22 dBm/Hz. Con una atenuación por trayectoria de 130 dB la señal llega al móvil con una densidad espectral de -152 dBm/Hz, que está por arriba de la densidad espectral del ruido térmico.

$$\delta = \frac{8W}{1,25MHz} = 0,0064W / Hz$$

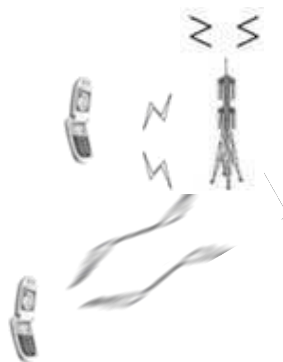
$$\delta_{dBm/Hz} = 10 \log \left( \frac{0,0064W / Hz}{0,001W} \right) = -22dBm / Hz$$

$$llega \quad MS = \delta_{dBm/Hz} - atenuación_{dB}$$

$$llega \quad MS = -22 - 130 = -152dBm / Hz$$

#### 4.2.6 CONTROL DE POTENCIA<sup>95</sup>

La interferencia en sistemas CDMA tradicionales es un gran problema. Este es el porque del control de potencia. Sin este control, un usuario cercano a una celda o a una BS saturaría el receptor de la BS e impediría que otras señales ingresen.

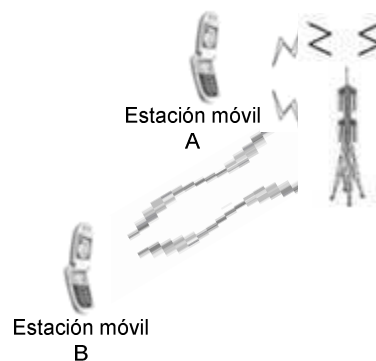


**Figura 4.33** Control de Potencia.

<sup>95</sup> CDMA RF System Engineering, Yang, Samuel C., p. 83-91.

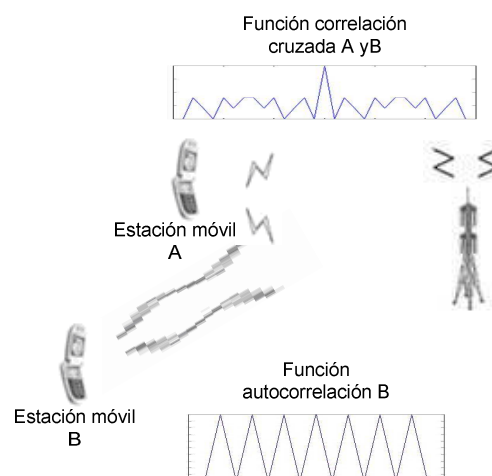
Este problema se puede explicar en términos de propiedades de auto correlación y de correlación cruzada considerando los códigos Walsh utilizados en un sistema IS-95A en el enlace Forward. La diferencia entre el pico de auto correlación y el pico de auto correlación fuera de fase debe ser lo más grande posible.

Asumiendo dos móviles dentro de una celda, si el móvil A se encuentra muy cerca de la BS, y el móvil B se encuentra mucho más alejado de la BS.



**Figura 4.34** Control de Potencia. El Móvil A se encuentra más cerca que el móvil B.

Si el receptor de la BS quiere recibir la señal del móvil B y no existe control de potencia, transmitiendo los dos móviles a la misma potencia, el receptor de la BS nunca detectará el pico de auto correlación de la señal de B porque la función de correlación cruzada generada por A será mucho mayor que el pico de auto correlación de B.



**Figura 4.35** Control de Potencia.

Por esto se debe controlar la potencia de transmisión de los móviles de una celda para que sus señales sean vistas por el receptor de la BS a un nivel similar. Ya que los códigos utilizados en CDMA poseen imperfecciones, una red IS-95A es un sistema limitado por interferencia. Cada móvil es un generador de interferencia. Los móviles poseen un receptor especial llamado RAKE que puede hacer estimaciones de desvanecimiento por multitrayectoria y puede recuperar información exitosamente.

El sistema realiza el control de potencia tanto en el canal Forward como en el canal Reverse.

Existen dos tipos de control de potencia: de lazo abierto y de lazo cerrado.

#### 4.2.6.1 Control de potencia de lazo abierto

Antes que el móvil establezca contacto con la estación Base no conoce la ubicación de esta, ni conoce con que potencia debe realizar la transmisión de información. El **estándar IS-95A** define que en los primeros intentos del móvil para ingresar al sistema transmite una serie de **Pruebas de Acceso (Access Probe)**.

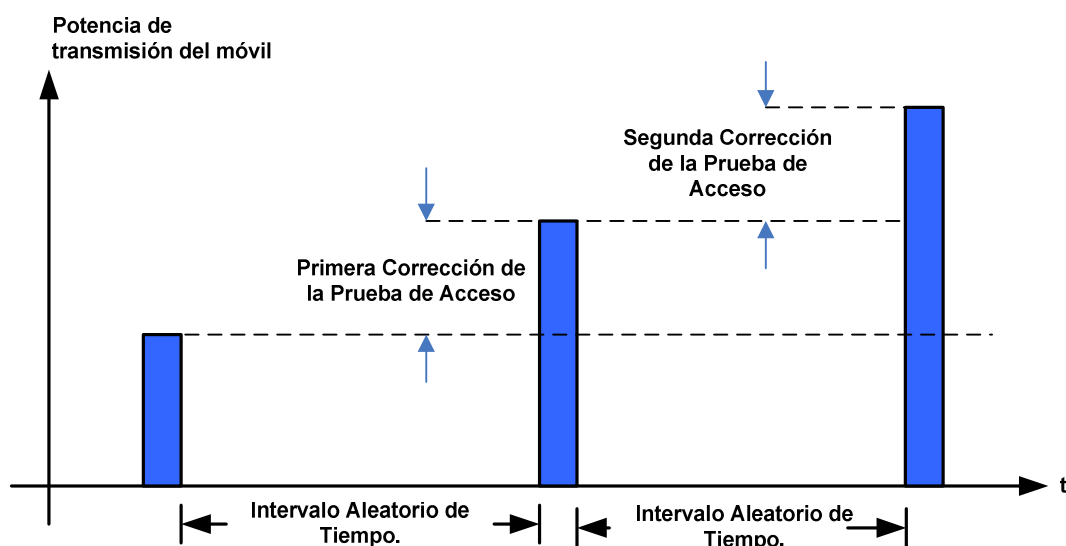
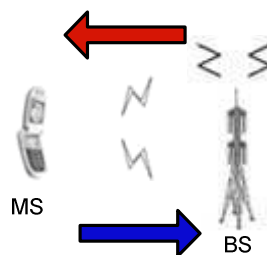


Figura 4.36 Control de Potencia de Lazo Abierto; Pruebas de Acceso.

Las **Pruebas de Acceso** son transmisiones progresivas, cada una de mayor potencia respecto a la anterior. La primera prueba de acceso se transmite a una potencia relativamente baja. Si después de un intervalo de tiempo aleatorio, el móvil no recibe un acuse de recibo **ACK** de la estación base, entonces el móvil transmite una segunda prueba de acceso a una potencia ligeramente mayor. Este proceso se repite hasta que el móvil recibe un **ACK** de la estación base.

#### 4.2.6.2 Control de potencia de lazo cerrado

Utilizado para compensar las fluctuaciones de potencia por desvanecimiento. Cuando el móvil recibe un canal de tráfico y comienza a comunicarse con la estación base, comienza el uso del lazo cerrado.



**Figura 4.37** Control de Potencia de Lazo Cerrado.

La BS monitorea continuamente el enlace reverse y mide la calidad del enlace. Si la calidad del enlace es mala, la BS comanda al móvil mediante el enlace Forward que aumente la potencia de transmisión; si la calidad del enlace es demasiado buena hay un exceso de potencia en el enlace reverse y la BS comanda al móvil que reduzca la potencia de transmisión. Eb /No es utilizado como indicador de calidad del enlace Reverse.

La estación base envía los controles de control de potencia en el enlace Forward como bits de control de potencia (PCB, power control bits). Nominalmente por cada PCB se aumenta o disminuye la potencia del móvil +1 dB y -1dB.

Los PCBs son enviados sobre el canal de tráfico.

#### 4.2.7 CANALES FORWARD Y REVERSE

La comunicación entre el móvil y la estación base toma lugar con el uso de canales específicos.

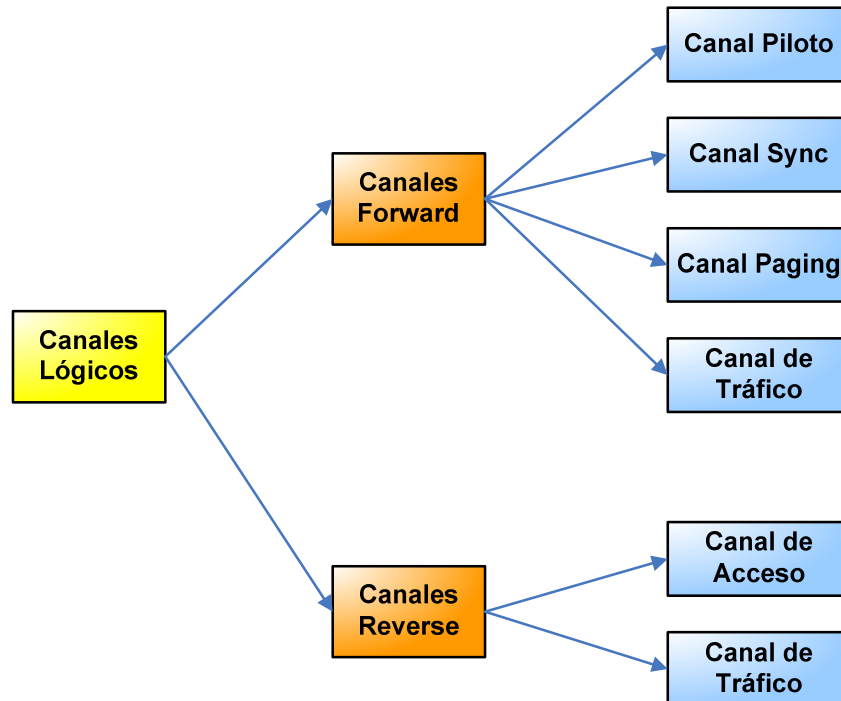
El canal forward (desde BS a MS) está formado por los siguientes canales:

- Canal Piloto (siempre utiliza el código Walsh W0). Con este canal, la estación base establece contacto con los móviles dentro de su celda.
- Canal Sync (siempre utiliza el código Walsh W32). Este canal notifica al móvil de información importante sobre la sincronización del sistema y otros parámetros como tiempo del sistema, tiempo del día, basado en el tiempo del GPS.
- Canales Paging (utiliza los códigos Walsh W1-W7). Este canal provee información e instrucciones al móvil y mensajes de reconocimiento luego de los intentos de acceso.
- Canales de Tráfico (utilizan los códigos Walsh W8-W31 y W33-W63). Este canal se utiliza para transmitir datos del usuario y voz. También son enviados mensajes de señalización.

El canal Reverse (desde MS a BS) está formado por los siguientes canales:

- Access channel. Este canal es usado por los móviles que todavía no están en una llamada, para transmitir peticiones de registro, peticiones de configuración de llamada, respuestas de paginación, respuestas de orden
- Traffic channel. Este canal se utiliza para transmitir datos del usuario y voz. También son enviados mensajes de señalización.





**Figura 4.38** Canales Lógicos de IS-95A.

#### 4.2.8 CANALES FORWARD<sup>96</sup>

Un canal Forward es un canal físico cuya dirección de transmisión es desde la BS a la MS. Estos canales pueden transportar tráfico normal, además de información de cabecera. Esta información se utiliza para establecer parámetros como sincronización del sistema, identidad de estaciones, etc. Existen tres tipos principales de información de cabecera: señales piloto, de sincronización y de paging. Tiene una señal piloto, un canal sync y 7 canales paging.

El enlace Forward consiste de 64 canales lógicos, y cada uno de estos canales utiliza uno de los 64 códigos de Walsh disponibles. Gracias a la sincronización del enlace Forward todos los códigos son ortogonales entre sí, permitiendo reconocer a cada uno en el receptor. Cada canal Forward es multiplicado por el código corto, que posee dos componentes, I y Q. Con el resultado, se realiza una modulación QPSK. En este estándar, los canales I y Q, llevan la misma información pero como se mencionó antes, utilizan dos códigos PN cortos diferentes (PNI y PNQ).

<sup>96</sup> The Next Generation CDMA Technologies, Chen, Hsiao-Hwa, p. 80-82.

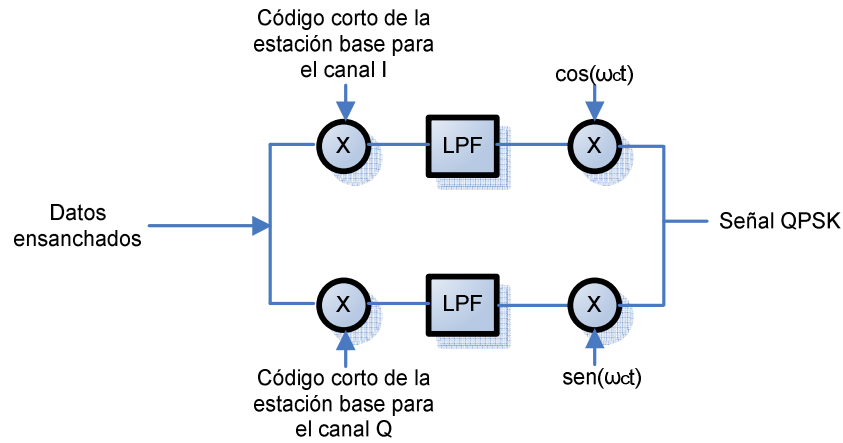


Figura 4.39 Modulación QPSK.

La potencia de transmisión se reduce por 3, 6, o 9 dB para las diferentes tasas de transmisión a 9600, 4800 o 2400 bps. Cada canal de tráfico está formado por un subcanal de datos de tráfico y por un subcanal de control de potencia del móvil.

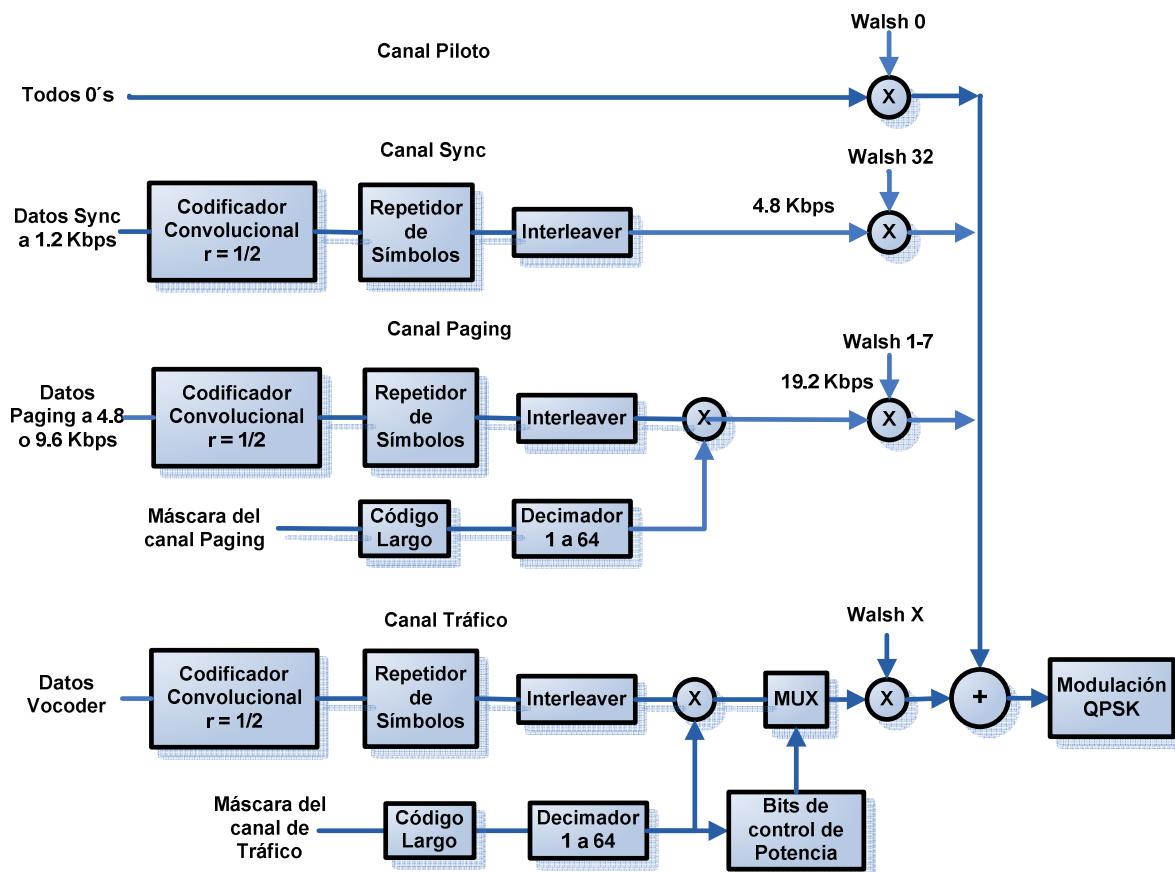


Figura 4.40 Esquema general del canal Forward

#### 4.2.8.1 Canal Piloto<sup>97</sup>



Figura 4.41 Canal Piloto.

Con este canal, la estación base establece contacto con los móviles dentro de su celda. La estación base está transmitiendo continuamente sobre el canal lógico piloto una señal formada solo por ceros, la cual está asociada al código Walsh 0, que también es un código formado por solo ceros. Entonces sirve como identificación de celda/sector, además sirve como referencia de tiempo, fase y frecuencia.

El canal Piloto tiene un buen SNR, permitiendo que los móviles lo localicen fácilmente. Esta señal es luego multiplicada y ensanchada por un par de secuencias PN cortas (ambas de longitud 32768 chips).

Las mismas secuencias PN piloto son utilizadas en todas las estaciones base, pero a cada estación base se le asigna un offset (Offset PN).

Los Offset PN Piloto son asignados a las estaciones en múltiplos de 64 chips, dando un total de 512 posibles asignaciones (de 0 a 511) y permiten aislar a las estaciones bases que transmiten información en el mismo espectro de frecuencia.

Si el índice de offset de la secuencia piloto PN es 12, el offset de fase de la secuencia será  $12 \times 64 = 768$  chips.

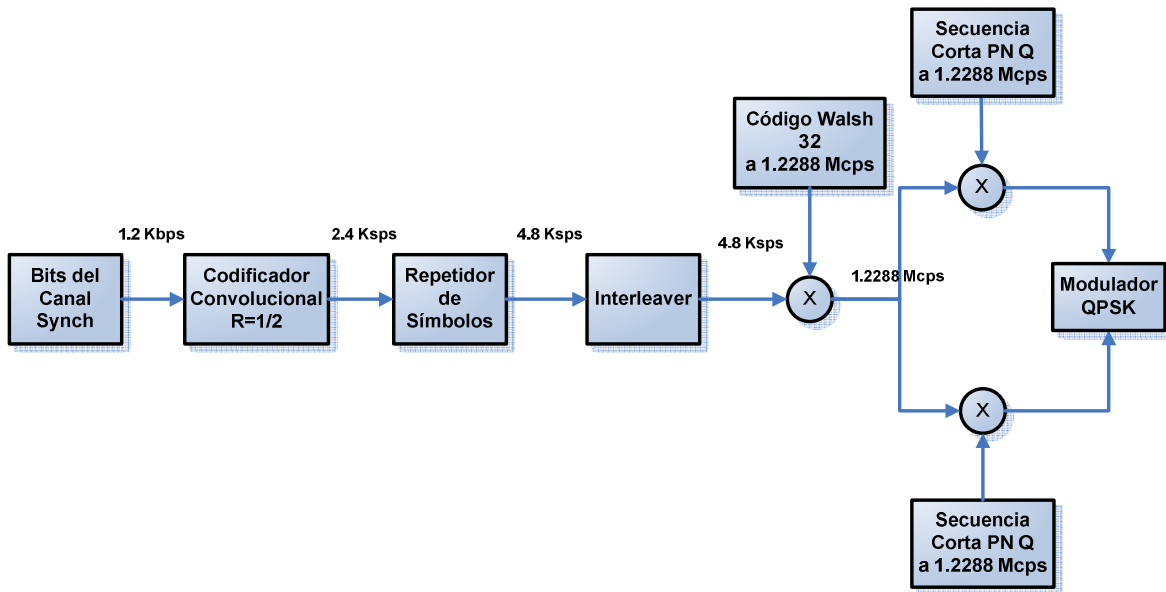
En el lado del receptor, el móvil recoge la esta señal y anota la estación base que la está transmitiendo. Para determinar todas las estaciones base posibles (512 posibles, cada una utilizando un código corto desplazado 64 chips), la red utiliza una señal GPS.

---

<sup>97</sup> CDMA RF System Engineering, Yang, Samuel C., p. 106.

Este canal provee sincronismo con el canal forward. La sincronización se da por medio del proceso de correlación cruzada entre la señal que se recibe y otra generada localmente por el móvil, así que provee la fase de referencia para el demodulador coherente.

#### 4.2.8.2 Canal Sync<sup>98</sup>



**Figura 4.42** Estructura del Canal Sync.

Cada estación base está sincronizada con un receptor GPS, y la transmisión en el canal Forward es controlada en tiempo de forma estricta. El canal Sync es utilizado por la estación móvil para adquirir sincronización de tiempo inicial.

El canal Sync lleva información en banda base. La información es contenida en el ***mensaje del canal sync***, que notifica al móvil de información importante sobre la sincronización del sistema y otros parámetros como tiempo del sistema, tiempo del día, basado en el tiempo del GPS. Los bits son ensanchados con la misma secuencia corta del canal Piloto.

<sup>98</sup> CDMA RF System Engineering, Yang, Samuel C., p. 106-109.

La transmisión de los canales Piloto y Sync es iniciada por la misma referencia GPS. Cuando el móvil adquiere el canal Piloto de una estación base en particular, automáticamente conoce la estructura básica de temporización del canal Sync, y por tanto su inicio, además del inicio de la secuencia PN corta. Cada estación base transmite continuamente un canal Sync.

La información de este canal incluye el offset de la secuencia PN corta de la BS que ha obtenido el móvil, ID del sistema(SID) y ID de la red (NID), hora del sistema y hora local, versiones del protocolo, tasa de datos del canal paging (4,8 o 9,6 Kbps)y la lista de canales paging.

El estado del generador de código largo en el tiempo de sistema es también enviado en el mensaje de Sincronización. Esto permite que el móvil inicialice y ejecute su generador de código largo muy cerca en el tiempo de sincronización con los generadores de código largo en las estaciones base.

La velocidad de transmisión de este canal es 1200 bps.

Utiliza el código de Walsh 32 (W32).

De los tres canales del enlace Forward, el canal Sync tiene asociado la menor cantidad de potencia.

Con este canal y el canal Sync, el móvil obtiene la información de temporización absoluta, necesaria para realizar la demodulación coherente de los canales paging y tráfico del enlace forward, y para establecer la temporización de su transmisor.

### **Estructura del Mensaje del Canal Sync**

El canal Sync es transmitido en grupos de supertramas, cada supertrama contiene 96 bits y tiene una duración de 80 ms, obteniendo una velocidad de transmisión:

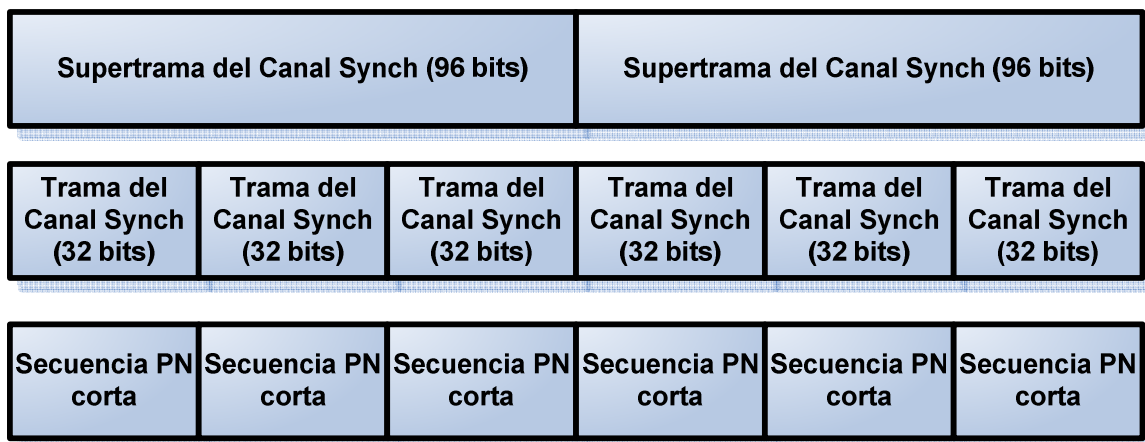
$$V_{tx} = \frac{96bits}{80ms} = 1200bps$$

Cada supertrama contiene tres tramas del canal sync de igual longitud y duración:

$$\#bits / trama = \frac{96bits}{3} = 32bits$$

$$Duración\_trama = \frac{80ms}{3} = 26.67ms$$

Cada trama del canal sync está alineada con la secuencia PN corta, asociada con el sector de transmisión.



**Figura 4.43** Estructura del mensaje del canal Sync

Cada secuencia PN corta se repite cada 26.67ms, y cada periodo de la secuencia PN corta está sincronizado con cada trama del canal Sync.

Por tanto, cuando el móvil adquiere sincronización con el canal Sync, la alineación para el canal Sync es conocida inmediatamente. Esto debido a que el canal Sync es ensanchado con la misma secuencia PN Sync, y porque la sincronización de la trama del canal Sync está alineada con la secuencia PN Sync.

Cuando el móvil alcanza alinearse con el canal Sync, el móvil puede empezar a leer el mensaje del canal Sync.

El mensaje del canal Sync es largo y puede ocupar más de una trama del canal Sync. El mensaje del canal Sync está organizado en una estructura llamada cápsula del mensaje del canal sync. Una cápsula consiste del mensaje del canal sync y relleno.

Los bits de relleno son utilizados para llenar las posiciones de bit, antes de que comience una nueva supertrama, donde el siguiente mensaje del canal sync comienza.

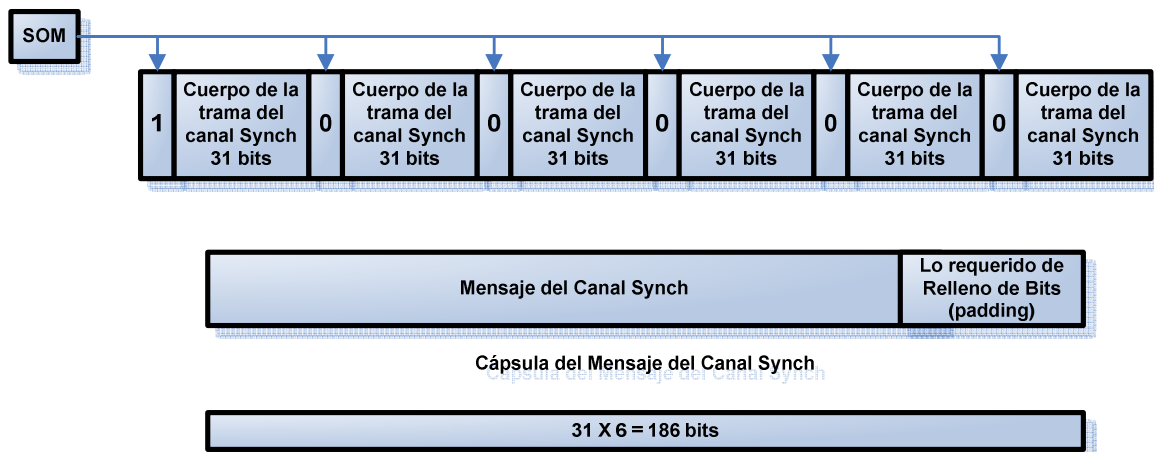
Cada trama del canal Sync comienza con el bit SOM (Start of Message). El SOM es el primer bit de la trama del canal Sync.

Un SOM de 1 indica el inicio de un mensaje. Un SOM de 0 indica que la actual trama contiene parte de un mensaje que comenzó en una trama anterior. De esta manera, la estación base puede transmitir el mensaje del canal Sync en tramas consecutivas.

Un SOM de 1 coincide también con el inicio de una supertrama.

Los 31 bits restantes se los conoce como el cuerpo de la Trama.

El gráfico siguiente muestra la estructura de un mensaje del canal Sync que ocupa dos supertramas.



**Figura 4.44** Estructura de un Mensaje del canal Sync que ocupa dos supertramas.

El mensaje del canal Sync contiene diferentes campos.

Para la sincronización con las secuencias PN cortas y largas se tienen los siguientes campos:

El mensaje contiene la información del offset de la secuencia PN corta usada por la estación base en el campo **PILOT\_PN**.

El mensaje también contiene información que le permite al móvil sincronizarse con la secuencia PN larga a través de los campos **LC\_STATE** y **SYS\_TIME**. La estación base configura al campo **LC\_STATE** con el estado del código largo a un tiempo futuro dado por el campo **SYS\_TIME**. En el tiempo preciso dado por el campo **SYS\_TIME**, el móvil comienza a correr la secuencia PN larga en el estado dado por el **LC\_STATE**.

Por lo tanto después de la recepción exitosa del mensaje del canal Sync, el móvil está sincronizado con la secuencia PN corta transmitida por la estación base y conoce su offset. Además el móvil está sincronizado con la secuencia PN larga.

#### 4.2.8.3 Canal Paging<sup>99</sup>

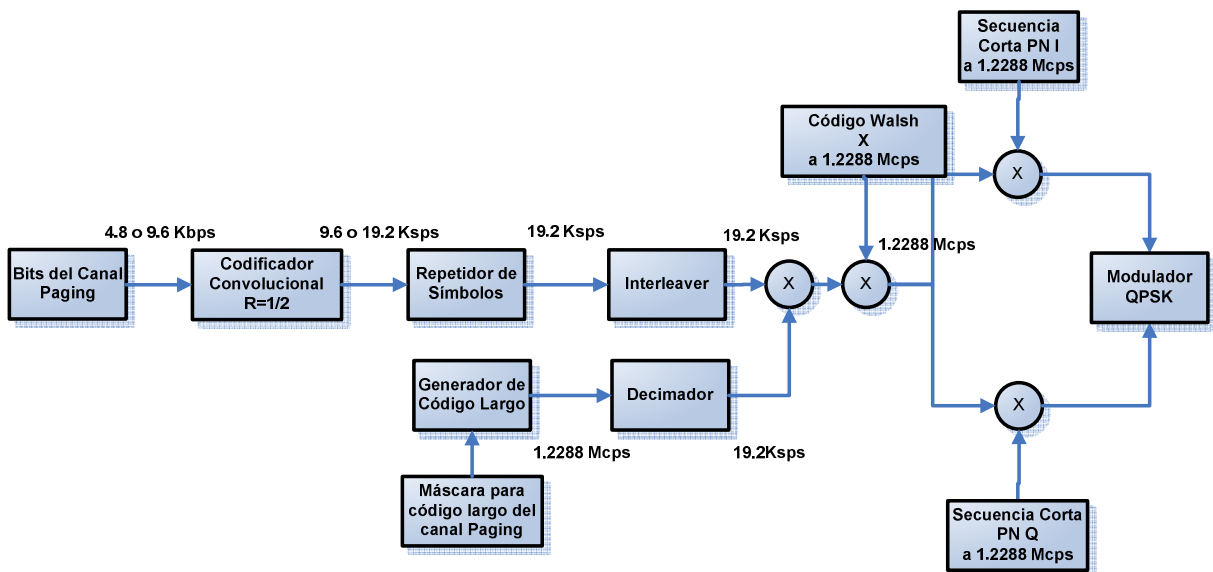


Figura 4.45 Canal Paging.

Con la información de los canales anteriores (el nombre de la red a conectarse) y una vez que adquiere la temporización y sincronismo a través del canal Sync, el

<sup>99</sup> CDMA RF System Engineering, Yang, Samuel C., p. 109-112.



móvil está preparado para recibir y realizar llamadas, y comienza a monitorear un canal paging.

El canal Paging lleva información en banda base. Son 7 canales, cada uno con un código de walsh. A pesar de que cada sector dispone de siete canales paging, el móvil solo monitorea uno.

El canal Paging puede estar lleno de datos. Puede incluir un tono de llamada o un mensaje mail de voz recibido. Provee al sistema de parámetros de voice pages, sms y otros mensajes de broadcast para los usuarios en una celda.

Los datos en el canal Paging enviados por la estación base incluyen el ESIN (Número de Identificación Serial Electrónico) del móvil, y está cubierto por un código largo.

Para el canal Paging, el sistema utiliza el código largo público, ya que no hay una conexión con un móvil específico. El canal debe llegar a todos los móviles.

Cuando el móvil correcto responde, un nuevo código largo privado será asignado al tiempo anterior a que la llamada sea conectada.

Mientras el móvil escanea el canal Paging, reconoce su número telefónico y responde con un tono.

Cuando se recibe la llamada, un mensaje de acceso vuelve a la estación base.

El móvil, que utiliza Qualcomm CDMA, genera un código de 18 bits. El móvil envía esta secuencia de autenticación a la estación base durante la parte de sincronización del protocolo de mensajería.

La estación base revisa el código de autenticación antes de permitir el establecimiento de llamada. Luego emite un número aleatorio al móvil, el cual es utilizado por el móvil en el algoritmo CAVE para generar una máscara de código largo específica. Al mismo tiempo, la estación base realiza la misma operación. Con esto, tanto el móvil como la estación base poseen el mismo código largo.

La velocidad de transmisión puede ser 4.8 o 9.6 Kbps. El campo PRAT, en el mensaje del canal Sync, informa al móvil de la velocidad de transmisión del canal paging. Cuando el móvil adquiere timing y sincronización utilizando el canal Sync, el móvil comienza a monitorear al canal paging.

Luego de realizarse el interleaving, a los datos se les realiza scrambling con una secuencia PN larga decimada, luego son ensanchados por un código de Walsh asignado a ese canal paging y luego es ensanchado por la secuencia PN corta asignada a ese sector.

La secuencia PN larga pasa por un radio de decimación de 64:1 (de 1.2288 Mcps a 19.2 Ksps). El mismo generador de código largo es enmascarado con unas máscara específica a cada número de canal de paging (entre 1 y 7)

### **Estructura del Mensaje del Canal paging**

El canal paging se divide en ranuras de 80 ms. Un grupo de 2048 ranuras es llamado un ciclo de ranuras máximo (maximum slot cycle).

$$\#bits / ranura = 80ms * 4.8kbps = 384bits$$

$$\#bits / ranura = 80ms * 9.6kbps = 768bits$$

Una ranura de 80 ms se divide en 4 tramas del canal paging, y cada trama se divide en dos medias tramas. El primer bit de cada media trama se llama el indicador de cápsula sincronizado (SCI).

$$\#bits / trama = 384bits / 4 = 96bits$$

$$\#bits / trama = 768bits / 4 = 192bits$$

$$\#bits / media\_trama = 96bits / 2 = 48bits$$

$$\#bits / media\_trama = 192bits / 2 = 96bits$$

$$Duración\_trama = 96bits / 4.8Kbps = 20ms$$

$$Duración\_trama = 192 / 9.6Kbps = 20ms$$

$$Duración\_media\_trama = 48bits / 4.8Kbps = 10ms$$

$$Duración\_media\_trama = 96bits / 9.6Kbps = 10ms$$

Un mensaje del canal paging puede ocupar más de una media trama del canal paging, y un mensaje puede terminar en la mitad de una media trama del canal paging.

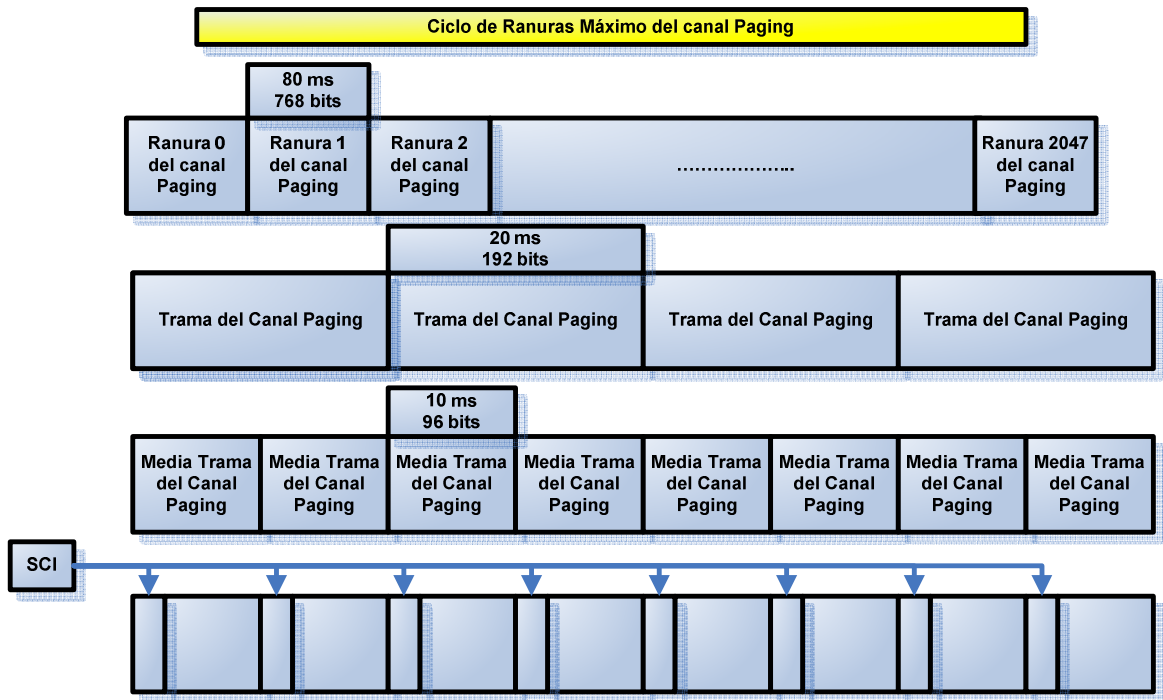


Figura 4.46 Mensaje del canal Paging

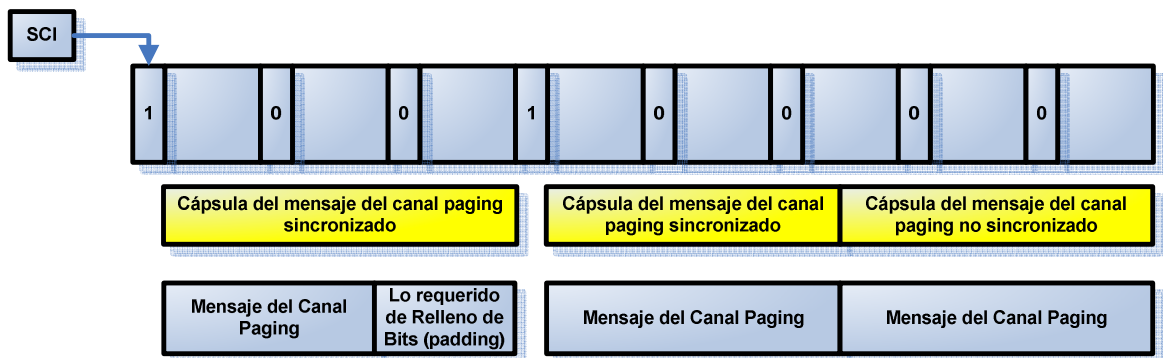


Figura 4.47 Estructura de un mensaje del canal Paging para tres mensajes que ocupan más de dos medias tramas.

El mensaje del canal paging puede ser transportado por cápsulas de mensaje del canal paging sincronizadas o por cápsulas de mensaje del canal paging no sincronizadas.

Si un mensaje termina en el medio de una media trama del canal paging, y si hay menos de 8 bits entre el final de ese mensaje y el bit SCI de la siguiente media trama, entonces la estación base incluiría suficientes bits de relleno al final de la cápsula de mensaje actual para extender la cápsula hasta el comienzo de la siguiente media trama.

Si la siguiente cápsula de mensaje a ser transmitida, es una cápsula de mensaje sincronizada, entonces la estación base incluiría suficientes bits de relleno al final del mensaje actual para extender a la cápsula hasta el comienzo de la siguiente media trama.

Si el mensaje termina en el medio de una media trama del canal paging, y si hay más de 8 bits entre el final del mensaje y el bit SCI de la siguiente media trama, entonces la estación base puede transmitir una cápsula de mensaje no sincronizada inmediatamente después del mensaje. En este caso no son añadidos bit de relleno.

El bit SCI muestra el comienzo de una nueva cápsula de mensaje en la actual media trama, si es 1, en caso contrario es 0.

Mensajes como overhead y paging son enviados por este canal. El mensaje overhead sirve para notificar al móvil de parámetros importantes de la configuración del sistema. Como por ejemplo mensajes de parámetros del sistema, mensajes de parámetros de acceso, y el mensaje de lista de vecinos.

Un mensaje de page puede contener información para un móvil en particular o puede contener información de un grupo de móviles.

### **Código PN Largo**

Luego de que se realiza el Interleaver. Los datos son multiplicados por el código largo. El código PN largo se genera con el polinomio anteriormente mencionado. Con este código PN largo, se realiza una operación de enmascarado con una máscara de 42 bits para generar diferentes fases del código largo que sirven para paginar estaciones móviles. La estructura de la máscara es la siguiente:



Figura 4.48 Máscara de canal Paging para código largo.

Donde **PCN** indica el número de canal Paging y **PN Piloto** indica el índice de Offset de secuencia PN corta para el enlace Forward.

#### 4.2.8.4 Canal de Tráfico<sup>100 101</sup>

Ahora, se explicará en detalle las etapas que conforman el canal de tráfico, ya que es necesario para analizar más adelante el demo del blockset de referencia CDMA, *IS-95A fwdchendtoend2* en MATLAB 7.0.

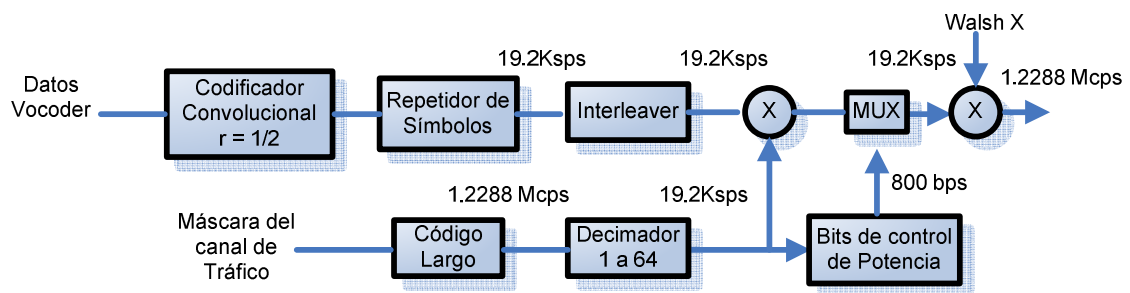
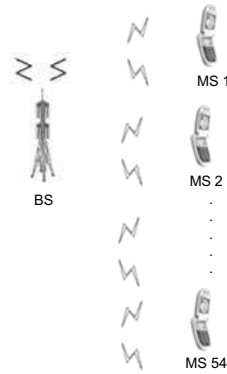


Figura 4.49 Canal Forward de Tráfico.

El canal de tráfico se utiliza para transmitir datos del usuario y voz. Los mensajes de señalización también son enviados sobre el canal de tráfico. La estructura del canal de tráfico es similar a la del canal paging con la diferencia que el canal de tráfico contiene bits de control de potencia multiplexados. La estación base puede transmitir datos de tráfico a un máximo de 54 móviles al mismo tiempo. Los canales son separados por los códigos Walsh. Estos códigos son solo utilizados para canalización por la estación base, por lo que el enlace Forward es sincrónico.

<sup>100</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 489-491.

<sup>101</sup> CDMA RF System Engineering, Yang, Samuel C., p. 114-118.



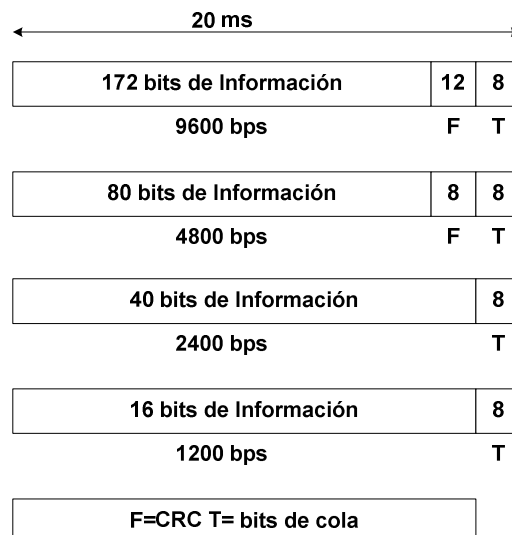
**Figura 4.50** Capacidad de BS.

Para el Rate Set I, el vocoder es capaz de variar su velocidad de transmisión de salida en respuesta a la actividad de voz. El canal soporta 4 diferentes tasas de transmisión.

A estos datos se les agrega un generador CRC para detección de errores y se obtienen tasas de: 9.6, 4.8, 2.4 y 1.2 Kbps.

### **Codificador CRC**

Existen cuatro tramas de canal de tráfico cada una con su velocidad de transmisión, tal como se muestra en la siguiente figura:

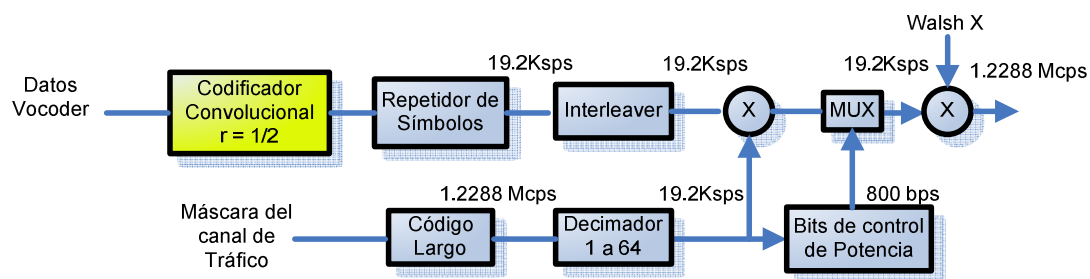


**Figura 4.51** Estructura de las tramas para el Canal de Tráfico del enlace Forward.



- Con los switches en la posición hacia la posición de arriba, y todos los elementos del generador cargados en uno lógico (**1**), el generador se desplaza 172 veces para los 9600 bps y 80 veces para los 4800 bps.
- Luego, los switches se ponen en la posición de abajo, y el generador es desplazado 12 veces más para los 9600 bps y 8 veces para los 4800 bps, para generar los bits CRC correspondientes, los cuales son transmitidos en el orden que son calculados.

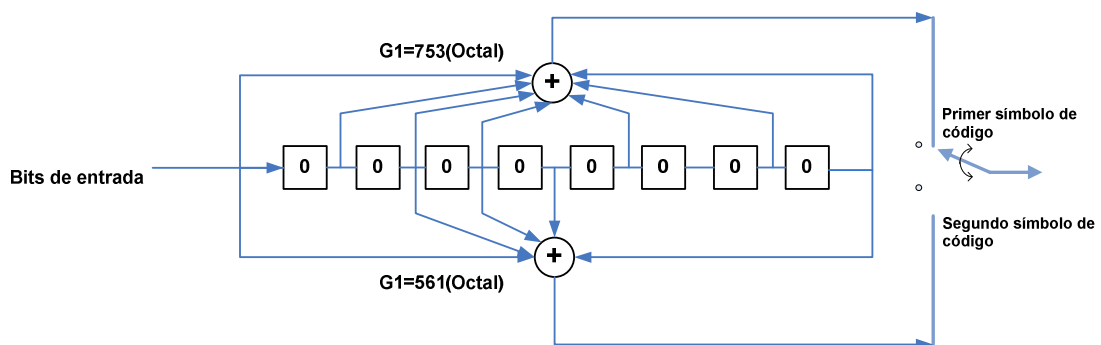
### Codificador Convolutacional



**Figura 4.54** Canal de Tráfico: Codificador Convolutacional.

La información en banda base es codificada convolutionalmente para protección contra errores. Para el Rate Set I, se utiliza un codificador convolutacional de tasa  $\frac{1}{2}$ . La codificación duplica la velocidad de transmisión. Los polinomios generadores implementados en el codificador convolutacional para el canal de tráfico Rate Set I, en forma de polinomio son:

$$1 + x + x^2 + x^3 + x^5 + x^7 + x^8 ; \text{ y } 1 + x^2 + x^3 + x^4 + x^8$$

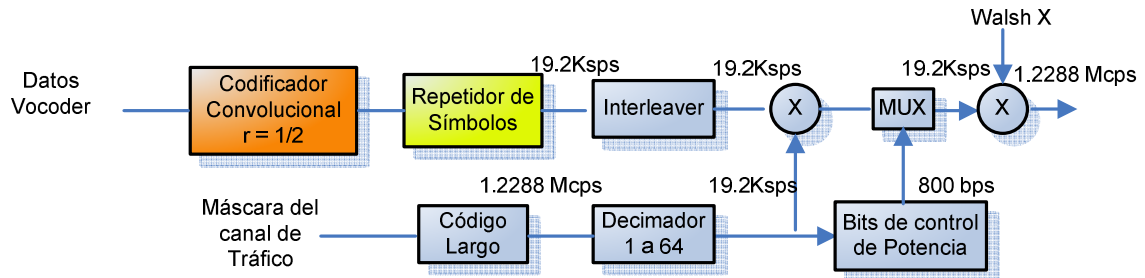


**Figura 4.55** Codificador Convolutacional para canal Forward de Tráfico.



Después de la codificación, la información pasa por un repetidor de símbolos, que repite los símbolos cuando el vocoder produce velocidades de transmisión bajas.

### Repetidor de Símbolos



**Figura 4.56** Canal de Tráfico: Repetidor de símbolos.

Cuando la tasa de transmisión es de 9.6 Kbps, la tasa de transmisión de símbolo después del codificador convolutivo es de 19.2 Ksps, por lo cual no es necesario el repetidor de símbolos.

Cuando la tasa de transmisión es de 4.8 Kbps, la tasa de transmisión de símbolo después del codificador convolutivo es de 9.6 Ksps, por lo cual cada símbolo es repetido una vez para obtener una tasa de transmisión de símbolo de 19.2 Ksps.

Cuando la tasa de transmisión es de 2.4 Kbps, la tasa de transmisión de símbolo después del codificador convolutivo es de 4.8 Ksps, por lo cual cada símbolo es repetido tres veces para obtener una tasa de transmisión de símbolo de 19.2 Ksps.

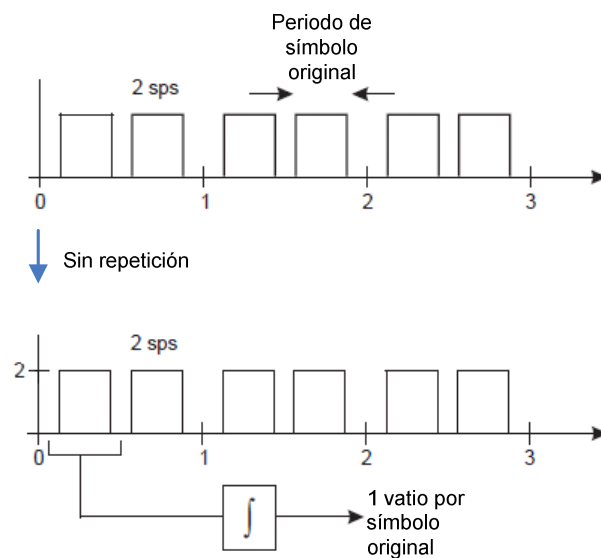
Cuando la tasa de transmisión es de 1.2 Kbps, la tasa de transmisión de símbolo después del codificador convolutivo es de 2.4 Ksps, por lo cual cada símbolo es repetido una vez para obtener una tasa de transmisión de símbolo de 19.2 Ksps.

La repetición de símbolos se utiliza para reducir la potencia de interferencia promedio a un tiempo dado cuando la tasa de transmisión más baja es transmitida.

### Ejemplo de la función del Repetidor

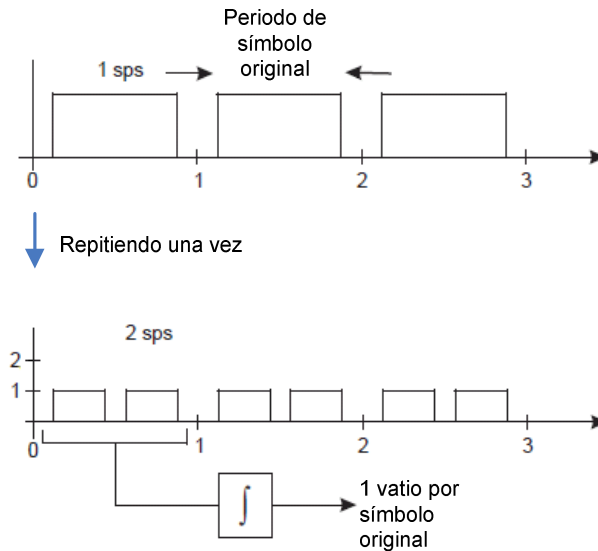
Asumiendo que un sistema quiere transmitir a una tasa de 2 símbolos por segundo y que luego el mismo sistema quiere transmitir a una tasa de 1 símbolo por segundo, se asume que el integrador necesita acumular 1 vatio de potencia para demodular un símbolo.

En el primer caso los símbolos originales no son repetidos. Si cada símbolo es enviado a un nivel de potencia de 2, entonces el integrador en el receptor es capaz de acumular 1W durante cada periodo de símbolo original. El periodo del símbolo original es 0.5 segundos.



**Figura 4.57** Caso 1: Transmisión sin repetición.

En el segundo caso, los símbolos originales son repetidos una vez, con una tasa de 2 símbolos por segundo; como la tasa de símbolos originales es 1 símbolo por segundo, se puede reducir la potencia de cada símbolo repetido a la mitad. Cuando el integrador acumule potencia sobre el periodo de símbolo original, asumiendo este 1 segundo, el integrador acumula el 1W necesario por cada periodo de símbolo original. El periodo del símbolo original es de 0.5 segundos.

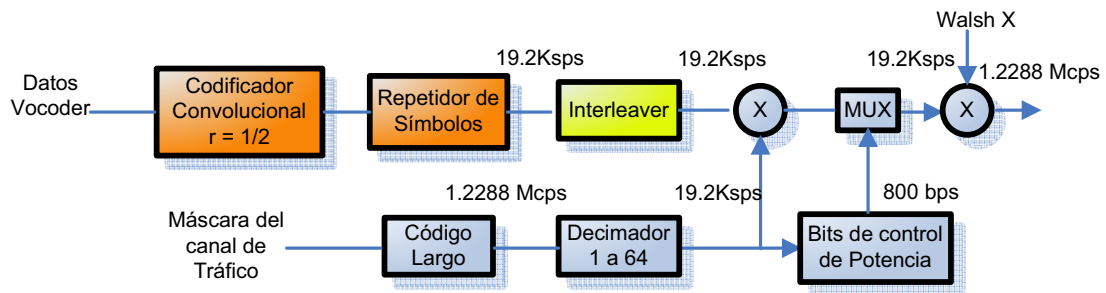


**Figura 4.58** Caso 2: Transmisión con repetición de símbolos.

La repetición de símbolos ayuda a disminuir la potencia por símbolo repetido cuando el vocoder está corriendo a una menor tasa. Esta es una forma efectiva de tomar ventaja del factor de actividad de voz en implementaciones de hardware. Cuando el vocoder está transmitiendo a menores tasas, la potencia de transmisión del canal Forward es reducida:

Cuando el vocoder está transmitiendo a 4.8 Kbps, la potencia por símbolo transmitida es la mitad que la de 9,6 Kbps. Cuando el vocoder está transmitiendo a 2.4 Kbps, la potencia por símbolo transmitida es un cuarto que la de 9.6 Kbps. Cuando el vocoder está transmitiendo a 1.2 Kbps, la potencia por símbolo transmitida es un octavo que la de 9.6 Kbps.

Después del repetidor, los datos son pasados por un Interleaver para combatir desvanecimiento de la señal.

**Interleaver**<sup>102 103</sup>

**Figura 4.59** Canal de Tráfico: Interleaver.

El interleaving es importante para comunicaciones sobre canales con desvanecimiento. Los datos son divididos en bloques o tramas. Con esto se dispersa las ráfagas de errores en el tiempo (diversidad temporal).

La técnica de Interleaving utilizada en IS-95A se basa en bits inversos que aleatorizan los errores de ráfaga e identifica la posición de cada símbolo de entrada en un arreglo de salida.

Para los canales Paging y de Tráfico se utiliza la misma estructura de Interleaving, ya que sus tramas poseen 384 símbolos con una duración de 20 ms. Como con el canal Sync, el Interleaver utiliza arreglos de entrada y salida. El canal de Tráfico puede generar 4 tasas diferentes. Este cambio de tasas solo se aprecia como una repetición de símbolos en el Interleaver. Para explicar el bloque de Interleaving de este canal, se ordenan los 384 símbolos de entrada en una matriz de 64 filas y 6 columnas para aplicar el algoritmo de bit inverso y obtener el orden de las salidas del Interleaver.

<sup>102</sup> CDMA Systems Engineering Handbook, Lee, Jhong S.; Miller, Leonard E.p.379-388.

<sup>103</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 484-485.

### Explicación del algoritmo de bit inverso para el Interleaver de canal Forward de Tráfico

Como se mencionó en el apartado anterior, las entradas del Interleaver se escriben columna por columna en una matriz de 64 X 6.

De esta matriz se identifica la posición de cada símbolo de entrada, y cada fila se indexa con un número decimal de 0 a 63. El número correspondiente a la fila se convierte a un número binario de 6 bits ( $2^6 = 64$ ).

El orden de los bits de las filas es invertido y convertido nuevamente a decimal para definir la fila que será leída a la salida del Interleaver.

Por ejemplo: considerando las filas de entrada 0, 1, 2, 3 y 4. Se calcula que filas van a ser leídas a la salida del Interleaver de la siguiente forma:

**Tabla 4.3** Algoritmo de bit inverso

Decimal	0	1	2	3	4
Binario	000000	000001	000010	000011	000100
Bit inverso	000000	100000	010000	110000	001000
Decimal	0	32	16	48	8

Como se puede observar en el ejemplo, después de realizar el algoritmo del bit inverso, las filas adquieren una nueva indexación y son reordenadas según este.

La siguiente tabla muestra el arreglo de entrada del Interleaver para el canal Forward de Tráfico de IS-95A:

Tabla 4.4 Arreglo 64X6 de Interleaving de canal de Tráfico.

Número de fila	Símbolos					
0	1	65	129	193	257	321
1	2	66	130	194	258	322
2	3	67	131	195	259	323
3	4	68	132	196	260	324
4	5	69	133	197	261	325
5	6	70	134	198	262	326
6	7	71	135	199	263	327
7	8	72	136	200	264	328
8	9	73	137	201	265	329
9	10	74	138	202	266	330
10	11	75	139	203	267	331
11	12	76	140	204	268	332
12	13	77	141	205	269	333
13	14	78	142	206	270	334
14	15	79	143	207	271	335
15	16	80	144	208	272	336
16	17	81	145	209	273	337
17	18	82	146	210	274	338
18	19	83	147	211	275	339
19	20	84	148	212	276	340
20	21	85	149	213	277	341
21	22	86	150	214	278	342
22	23	87	151	215	279	343
23	24	88	152	216	280	344
24	25	89	153	217	281	345
25	26	90	154	218	282	346
26	27	91	155	219	283	347
27	28	92	156	220	284	348
28	29	93	157	221	285	349
29	30	94	158	222	286	350
30	31	95	159	223	287	351
31	32	96	160	224	288	352

Número de fila	Símbolos					
32	33	97	161	225	289	353
33	34	98	162	226	290	354
34	35	99	163	227	291	355
35	36	100	164	228	292	356
36	37	101	165	229	293	357
37	38	102	166	230	294	358
38	39	103	167	231	295	359
39	40	104	168	232	296	360
40	41	105	169	233	297	361
41	42	106	170	234	298	362
42	43	107	171	235	299	363
43	44	108	172	236	300	364
44	45	109	173	237	301	365
45	46	110	174	238	302	366
46	47	111	175	239	303	367
47	48	112	176	240	304	368
48	49	113	177	241	305	369
49	50	114	178	242	306	370
50	51	115	179	243	307	371
51	52	116	180	244	308	372
52	53	117	181	245	309	373
53	54	118	182	246	310	374
54	55	119	183	247	311	375
55	56	120	184	248	312	376
56	57	121	185	249	313	377
57	58	122	186	250	314	378
58	59	123	187	251	315	379
59	60	124	188	252	316	380
60	61	125	189	253	317	381
61	62	126	190	254	318	382
62	63	127	191	255	319	383
63	64	128	192	256	320	384

A continuación se muestra la matriz resultante, donde se observa tanto la indexación anterior al algoritmo del bit inverso (en rojo), como la resultante (en azul):

Tabla 4.5 Arreglo de salida de Interleaver

Número de fila.		Símbolos					
0	0	1	65	129	193	257	321
32	1	33	97	161	225	289	353
16	2	17	81	145	209	273	337
48	3	49	113	177	241	305	369
8	4	9	73	137	201	265	329
40	5	41	105	169	233	297	361
24	6	25	89	153	217	281	345
56	7	57	121	185	249	313	377
4	8	5	69	133	197	261	325
36	9	37	101	165	229	293	357
20	10	21	85	149	213	277	341
52	11	53	117	181	245	309	373
12	12	13	77	141	205	269	333
44	13	45	109	173	237	301	365
28	14	29	93	157	221	285	349
60	15	61	125	189	253	317	381
2	16	3	67	131	195	259	323
34	17	35	99	163	227	291	355
18	18	19	83	147	211	275	339
50	19	51	115	179	243	307	371
10	20	11	75	139	203	267	331
42	21	43	107	171	235	299	363
26	22	27	91	155	219	283	347
58	23	59	123	187	251	315	379
6	24	7	71	135	199	263	327
38	25	39	103	167	231	295	359
22	26	23	87	151	215	279	343
54	27	55	119	183	247	311	375
14	28	15	79	143	207	271	335
46	29	47	111	175	239	303	367
30	30	31	95	159	223	287	351
62	31	63	127	191	255	319	383

Número de fila.		Símbolos					
1	32	2	66	130	194	258	322
33	33	34	98	162	226	290	354
17	34	18	82	146	210	274	338
49	35	50	114	178	242	306	370
9	36	10	74	138	202	266	330
41	37	42	106	170	234	298	362
25	38	26	90	154	218	282	346
57	39	58	122	186	250	314	378
5	40	6	70	134	198	262	326
37	41	38	102	166	230	294	358
21	42	22	86	150	214	278	342
53	43	54	118	182	246	310	374
13	44	14	78	142	206	270	334
45	45	46	110	174	238	302	366
29	46	30	94	158	222	286	350
61	47	62	126	190	254	318	382
3	48	4	68	132	196	260	324
35	49	36	100	164	228	292	356
19	50	20	84	148	212	276	340
51	51	52	116	180	244	308	372
11	52	12	76	140	204	268	332
43	53	44	108	172	236	300	364
27	54	28	92	156	220	284	348
59	55	60	124	188	252	316	380
7	56	8	72	136	200	264	328
39	57	40	104	168	232	296	360
23	58	24	88	152	216	280	344
55	59	56	120	184	248	312	376
15	60	16	80	144	208	272	336
47	61	48	112	176	240	304	368
31	62	32	96	160	224	288	352
63	63	64	128	192	256	320	384

La matriz resultante es leída fila por fila a la salida del Interleaver, por ejemplo:

Primero es leída la fila 0 que contiene los símbolos: 1, 65, 129, 193, 257 y 321, luego la fila 1, que contiene los símbolos: 33, 97, 161, 225, 289 y 353, y así con el

resto de filas. Este resultado es multiplicado por una secuencia PN larga decimada (scrambling).

### Scrambling

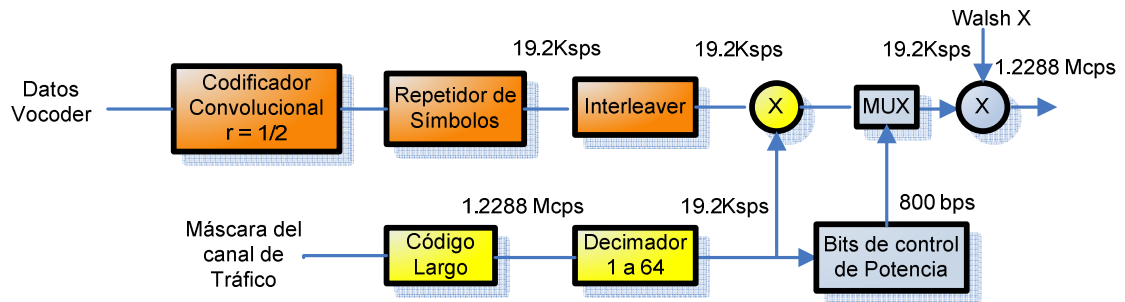


Figura 4.60 Scrambling

El generador de la secuencia PN larga produce la secuencia a una tasa de 1.2288 Mcps. Como la tasa a la salida del Interleaver es de 19.2 Ksps, la secuencia PN es decimada a un ratio de 64:1 para obtener una tasa de 19.2 Kcps. El generador de la secuencia PN larga utiliza una máscara específica del móvil para producir la secuencia; esta máscara es una función del número serial electrónico (ESN, Electronic Serial Number) del móvil. Hay dos tipos de máscaras: máscara de código largo público, utilizado para identificar la estación móvil; y la máscara de código largo privado de suscriptor, para identificar al suscriptor.



Figura 4.61 Máscara de código largo público

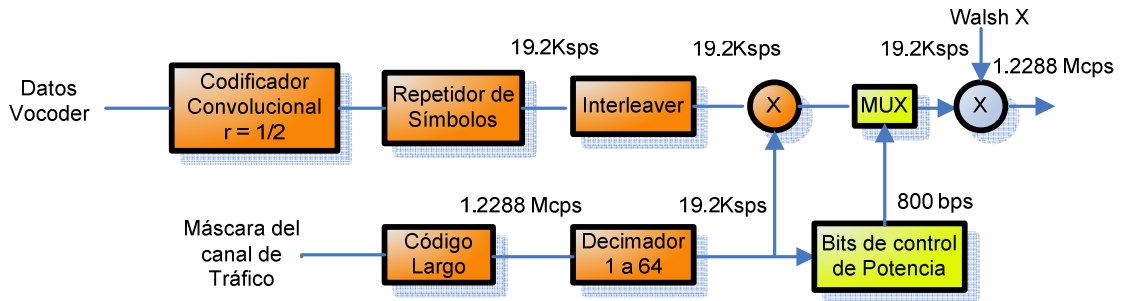


Figura 4.62 Máscara de código largo privado

### Inserción de bits de potencia



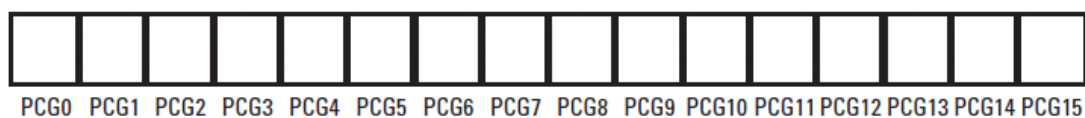
Los PCBs con una tasa de 800 bps son entonces multiplexados con los datos obtenidos de la multiplicación anterior a 19.2Ksps.



**Figura 4.63** PCBs en canal de tráfico

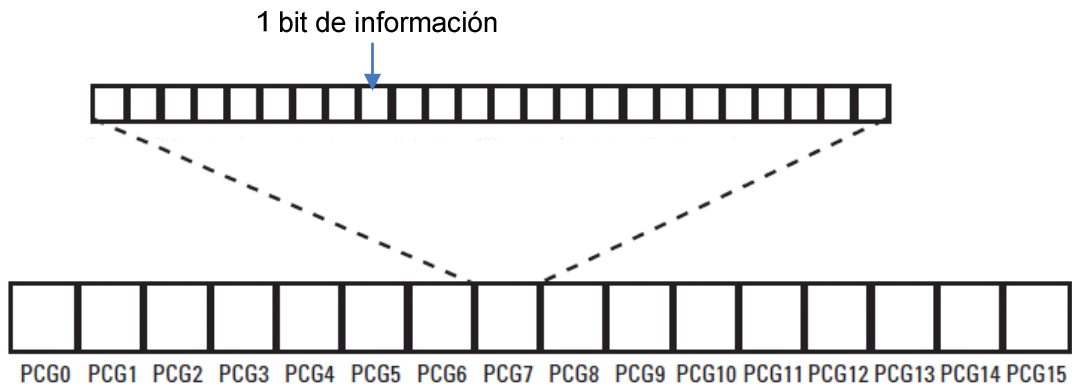
Es así que existe un subcanal para control de potencia (power control subchannel, PCS). Estos PCBs son transmitidos continuamente por la estación base. Un PCB es enviado cada  $1/800$  segundos o 1.25 ms

Las tramas de los enlaces Forward y Reverse tienen una duración de 20 ms. Como un PCB se envía cada 1.25 ms, cada trama del canal de tráfico puede ser dividida en  $20\text{ms}/1.25\text{ms} = 16$  segmentos. Estos segmentos se llaman grupos de control de potencia (power control groups, PCG).



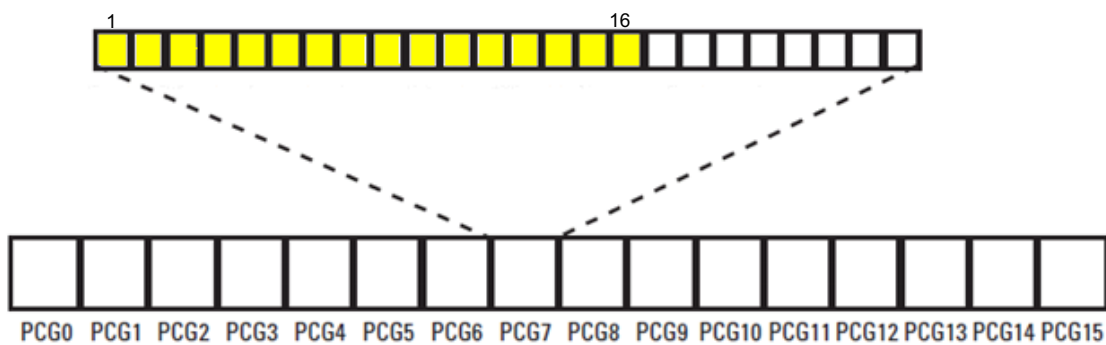
**Figura 4.64** Trama del canal Forward o Reverse dividida en PCGs.

Como cada PCG dura 1.25 ms y los datos tienen una tasa de 19.2 Ksps, entonces cada PCG contiene  $19.2 \times 10^3 \times 1.25 \times 10^{-3} = 24$  bits.



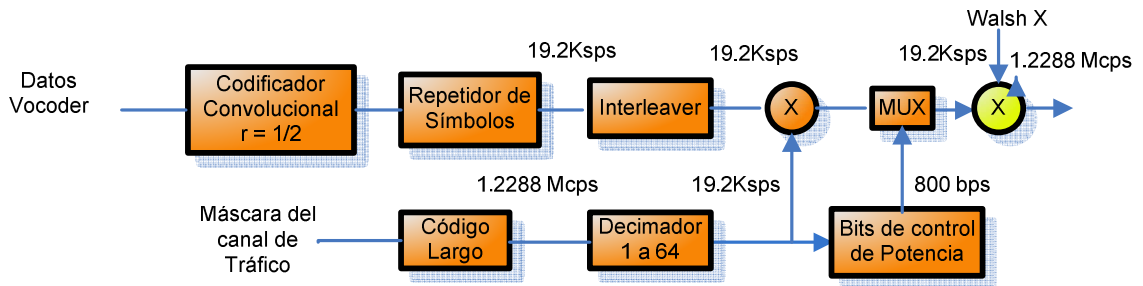
**Figura 4.65** Trama dividida en PCGs y en bits (1).

Como cada PCG contiene 24 bits, el PCB puede ser insertado en cualquiera de las 16 primeras posiciones de bits.



**Figura 4.66** Posibles posiciones de un PCB en un PCG del canal de tráfico

La ubicación exacta del PCB en el PCG es determinada de manera pseudoaleatoria. La posición del siguiente PCB es determinada por el valor decimal de los cuatro bits más significantes de la salida del decimador, dependiendo entonces de la secuencia PN larga. En el canal de tráfico Full rate, Rate Set I, se coloca el mismo PCB en dos posiciones contiguas. Los datos multiplexados (a 19.2 Ksps) son multiplicados por un código de Walsh. Cada canal de tráfico Forward se identifica por su código de Walsh asignado. Con esto se logra ensanchar cada símbolo por un factor de 64 y se obtiene una tasa de 1.2288 Mcps.



**Figura 4.67** Multiplicación del canal de tráfico por código Walsh.

Luego el resultado es multiplicado por el código corto correspondiente al sector de transmisión. Gracias a esto los 64 códigos de Walsh pueden ser reutilizados en cada sector.

Con el Rate Set II, la estructura del canal no cambia mucho. El vocoder de Rate Set II codifica la voz a mayores velocidades y mejora la calidad de voz. El Rate Set II soporta cuatro tasas de transmisión: 14.4, 7.2, 3.6, y 1.8 Kbps. Para mantener la tasa de transmisión a la salida del Interleaver a 19.2Ksps, la tasa del codificador convolutivo es de  $R = 3/4$ .

Con la información de los canales anteriores (el nombre de la red a conectarse) y una vez que adquiere la temporización y sincronismo a través del canal Sync, el móvil está preparado para recibir y realizar llamadas, y comienza a monitorear un canal Paging.

#### 4.2.8.5 Análisis del Demo IS-95A Forward Traffic Channel End-To-End Model

Este demo simula el canal de tráfico del enlace Forward para IS-95A con la mayor tasa de transmisión de 9600 bps para Rate Set I. El modelo en Simulink es el siguiente:

### IS-95A Forward Traffic Channel End-to-End Model

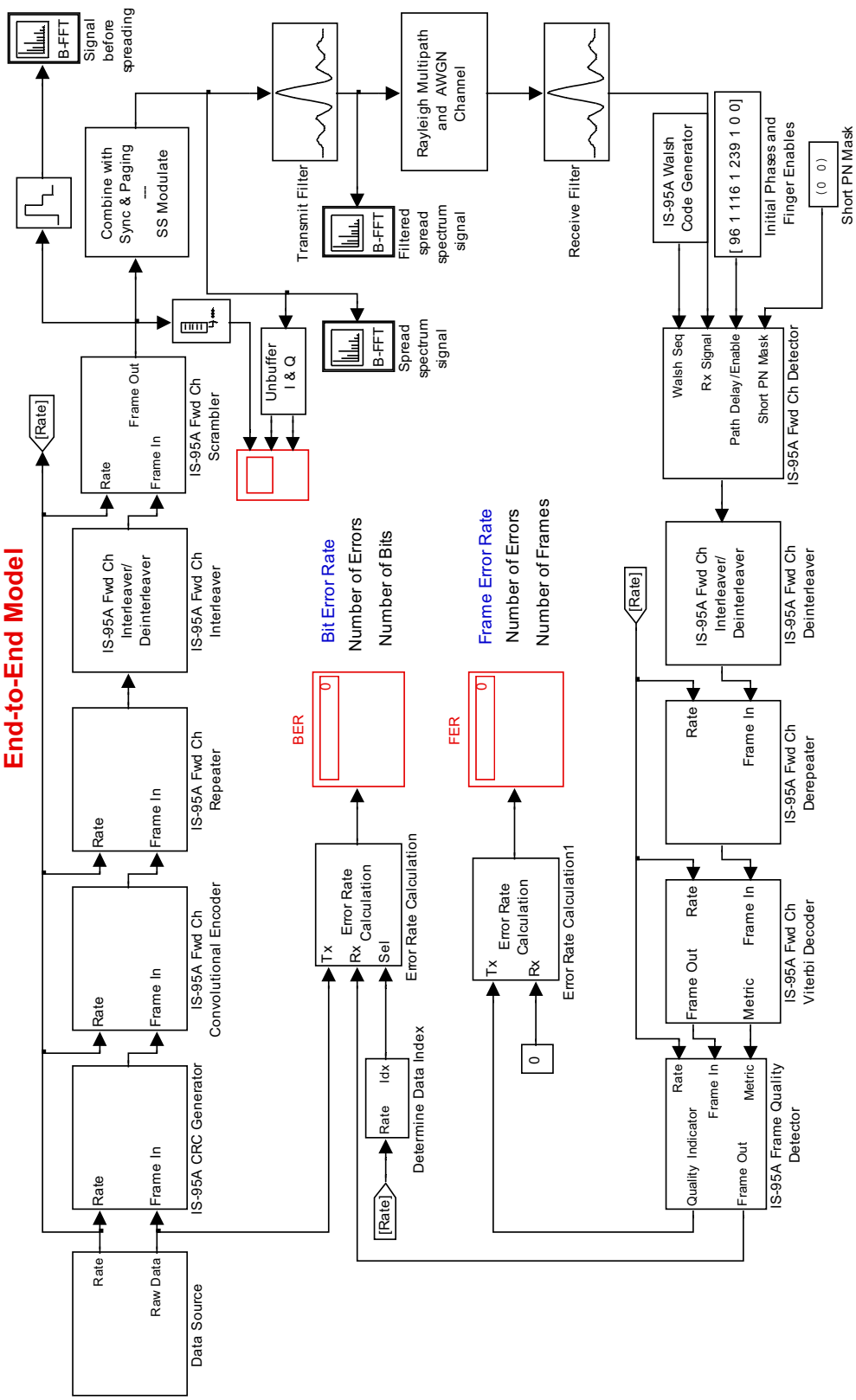


Figura 4.68 Sistemas de Bloques en Simulink.

El análisis del demo se realiza con un tiempo de simulación de 40 milisegundos.

Para hacer la captura de las señales después de cada bloque, se utiliza el bloque **To Workspace**.

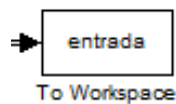
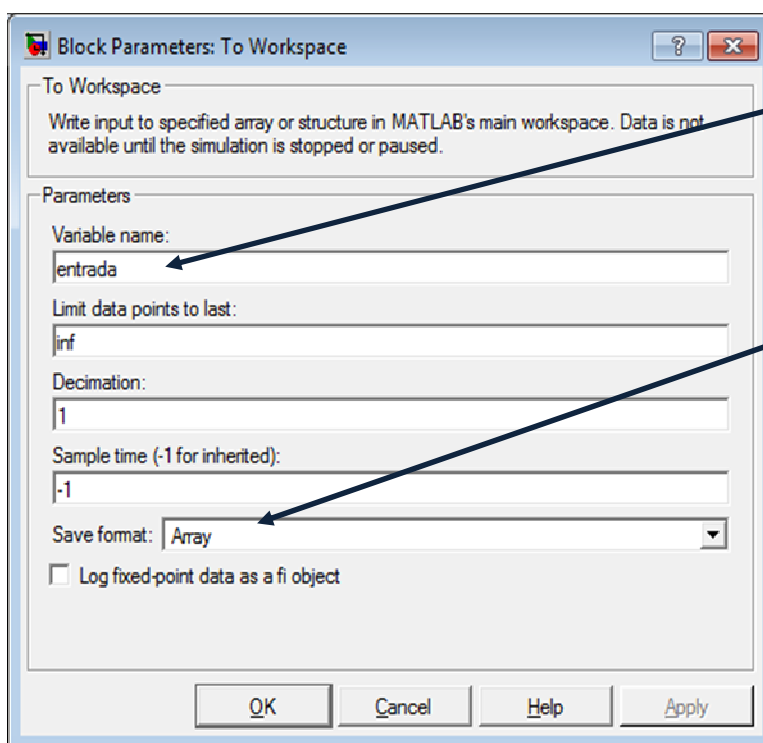


Figura 4.69 Bloque To Workspace



Se ingresa el nombre de la variable que se almacenará en el **Workspace**.

Se define que la variable almacena en el Workspace es un vector columna.

Figura 4.70 Configuración del bloque to Workspace.

### Generación de Datos

Para la generación de datos en el demo, se utiliza el bloque **Data Source**:

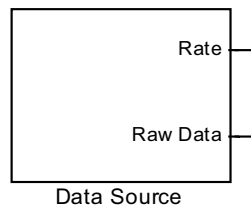


Figura 4.71 Bloque Data Source para demo IS-95A.

Este bloque está conformado por el bloque generador de tramas binarias aleatorio **Random Binary Frame Generator** y el bloque **Base Station Transmitter Data Rate**.

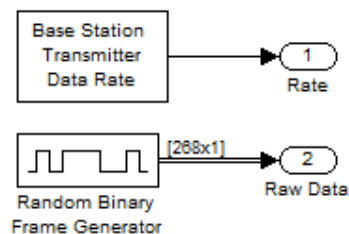


Figura 4.72 Bloque Data Source expandido.

El bloque **Random Binary Frame Generator** genera tramas de forma aleatoria con un tiempo de muestreo o tiempo de bit, y el número de bits por trama

Block Parameters: Random Binary Frame Generator

Random Binary Frame Generator (mask) (link)

Generate a random frame of binary integers. The output frame period is (Sample time) \* (Output frame size).

Parameters:

Initial seed:  
1234

Sample time (sec):  
1/(50\*268)

Output frame size:  
268

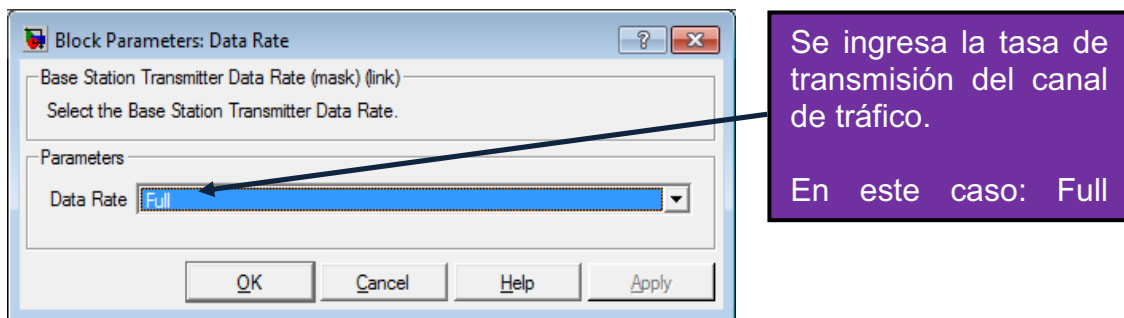
OK Cancel Help Apply

Se ingresa el tiempo de bit.  
En este caso:  
1/(50\*268) segundos

Se ingresa el número de bits por trama.

Figura 4.73 Configuración del bloque Random Binary Frame Generator.

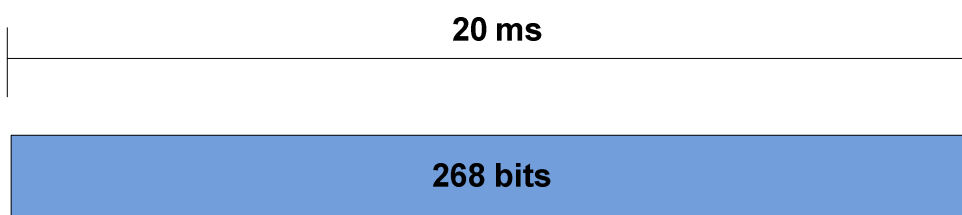
El bloque **Base Station Transmitter Data Rate** permite establecer la tasa de transmisión del canal de tráfico. Se puede escoger entre 4 tasas de transmisión (Full, Half, Quarter y Eighth rate), las cuales dependen si se implementa Rate Set I o Rate Set II. En el demo se implementa Rate Set I con la mayor tasa de transmisión (9600 bps, **Full rate**).



**Figura 4.74** Configuración del bloque Data Rate.

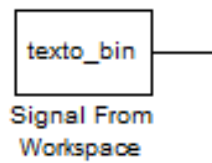
Todos los bloques del CDMA Reference Blockset trabajan con de entrada y salida, referidas a las tramas a la mayor tasa de transmisión que puede soportar un sistema IS-95A. Es decir la trama de un canal de tráfico del enlace Forward IS-95A para el Rate Set II. Las tramas están formadas por 268 elementos cada 0.02 segundos:

$$268\text{bits}/20\text{ms} = 13400\text{bps}$$



**Figura 4.75** Trama de Canal de Tráfico Forward Rate Set II.

Para propósitos de análisis del demo se ha cambiado el bloque **Random Integer Binary Generator** por el bloque **Signal From Workspace**, el cual importa una señal del **Workspace** a Simulink con una tasa de transmisión y tamaño de trama especificadas por el usuario.



Bloque Signal From Workspace

**Block Parameters: Signal From Workspace**

Signal From Workspace (mask) (link)

Output signal samples obtained from the MATLAB workspace at successive sample times. A signal matrix is interpreted as having one channel per column. Signal columns may be buffered into frames by specifying a number of samples per frame greater than 1.

An  $M \times N \times P$  signal array outputs  $M \times N$  matrices at successive sample times. The samples per frame must be equal to 1 for 3 dimensional signal arrays.

Parameters

Signal:

Sample time:

Samples per frame:

Form output after final data value by:

OK Cancel Help Apply

Se ingresa el nombre de la señal proveniente del **Workspace**.  
En este caso: texto\_bin

Se ingresa el tiempo de bit con el que se produce la señal en Simulink.  
En este caso:  $1/(268 \times 50)$

Se ingresa el número de bits por trama.  
En este caso: 268

Se especifica que la señal se repite cíclicamente.

Figura 4.76 Configuración del bloque Signal From Worspace

Para el análisis del canal Forward de Tráfico Rate Set I (9600 bps), se utiliza el siguiente formato de trama:

Se genera una trama de 268 bits, en la cual se encuentra un texto de 21 caracteres en formato ASCII, 4 bits de cola y 96 bits aleatorios. Se importa la señal **texto\_bin** guardada en el archivo **prueba.mat**. Este archivo se generó con el programa **texto\_a\_bits.m** en el cual se hace la conversión del texto **Politecnica Nacional\*** a bits. El código es:



```

clear all; clc
mensaje=('Politecnica Nacional*'); % texto que se transforma a
bits
texto_original= mensaje;
disp(texto_original) % Muestra la variable
size(texto_original);
% Transformación a caracteres ASCII
texto_ascii=abs(texto_original);
texto_bin=[];
% Transformación de caracteres ASCII a valores binarios (cada
carácter ASCII se transforma a un valor de 8 bits)
for i=1:length(texto_ascii)
    texto_bin=cat(2,texto_bin,dec2binvec(texto_ascii(i),8));
end
texto_bin=[texto_bin 0 1 0 1 randint(1,96)]; %Tenemos una trama
268 % bits

```

La trama generada por el código es la siguiente:

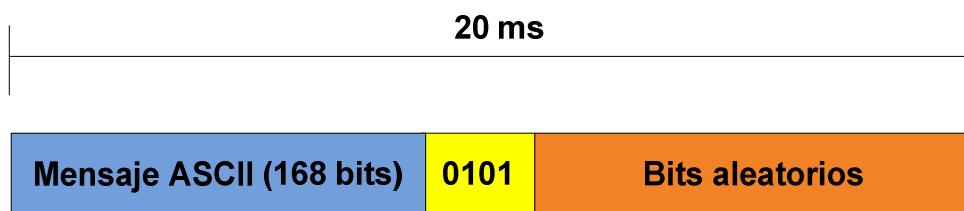


Figura 4.77 Trama generada con texto\_a\_bits.m

El bloque **Signal From Workspace**, repite la trama cíclicamente.

Se debe correr el programa **texto\_a\_bits.m** para que se almacene el vector **texto\_bin** en el **Workspace** y pueda ser importado por el bloque.

### Codificación CRC

Para realizar la codificación CRC en el demo, se utiliza el bloque **IS-95A CRC Generator**.

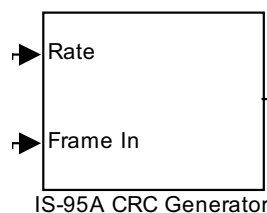


Figura 4.78 Bloque IS-95A CRC Generator.

- Este bloque genera bits de comprobación de redundancia cíclica (CRC) de acuerdo al estándar.
- El bloque anexa los bits CRC y los bits de cola (tail bits) a los bits de información de entrada del canal de tráfico para generar la trama de salida.
- El polinomio generador para los bits CRC está especificado en el estándar.
- El número de bits de información relevantes de la trama depende del tipo de canal y la tasa de datos.
- Los bits CRC y de cola siguen inmediatamente a los bits de información relevantes.

Los parámetros que se deben definir son: el Rate Set I y el canal de tráfico.

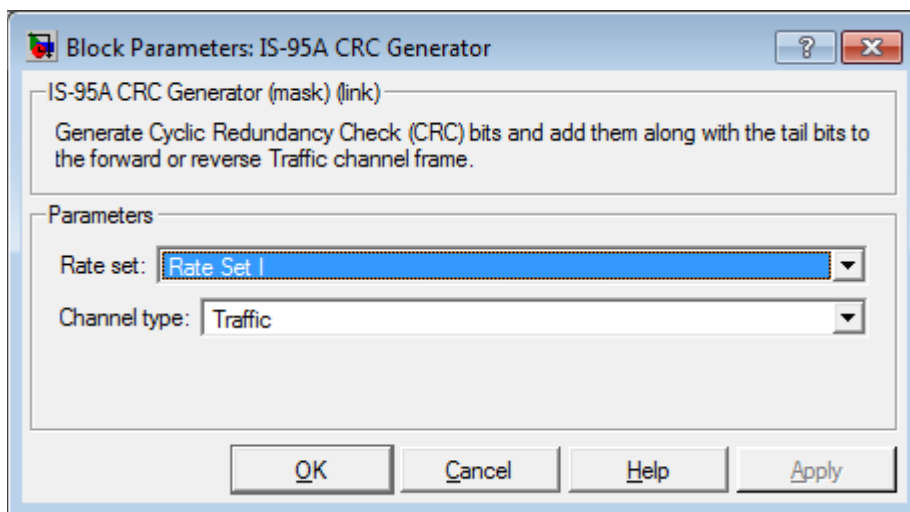


Figura 4.79 Configuración del bloque IS-95A CRC Generator.

### Número de bits de entrada y salida relevantes.

El número de bits de información relevantes de la simulación varía dependiendo del canal y su tasa de transmisión.

El tamaño del puerto de entrada **Frame In** debe ser obligatoriamente **268 elementos**, siendo este el número de bits producidos en una trama de Tráfico a Full Rate y Rate Set II:

$$13400bps \times 20ms = 268bits$$

El número de bits añadidos a la trama depende del canal y su tasa de transmisión. Por lo tanto **el tamaño del puerto de salida es de 288 elementos**, siendo este el número de bits producidos en una trama de Tráfico a Full Rate y Rate Set II luego del codificador CRC y con bits de cola:

$$268 \text{ bits de datos} + 12 \text{ bits CRC} + 8 \text{ bits de cola} = 288 \text{ bits.}$$

En la siguiente tabla se muestra el número de bits relevantes de entrada y salida para los distintos canales de IS-95A.

**Tabla 4.6** Bits relevantes de entrada y salida del bloque IS-95A CRC Generator del enlace Forward.

Tipo de Canal	Bits de entrada relevantes	Bits CRC	Bits de cola	Bits de salida relevantes.
Sync, Eighth Rate	32	0	0	32
Paging, Full Rate	192	0	0	192
Paging, Half Rate	96	0	0	96
Access	88	0	8	96
<b>Tráfico, Rate Set I, Full Rate</b>	<b>172</b>	<b>12</b>	<b>8</b>	<b>192</b>
Tráfico, Rate Set I, Half Rate	80	8	8	96
Tráfico, Rate Set I, Quarter Rate	40	0	8	48
Tráfico, Rate Set I, Eighth Rate	16	0	8	24
<b>Tráfico, Rate Set II, Full Rate</b>	<b>268</b>	<b>12</b>	<b>8</b>	<b>288</b>
Tráfico, Rate Set II, Half Rate	126	10	8	144
Tráfico, Rate Set II, Quarter Rate	56	8	8	72
Tráfico, Rate Set II, Eighth Rate	22	6	8	36

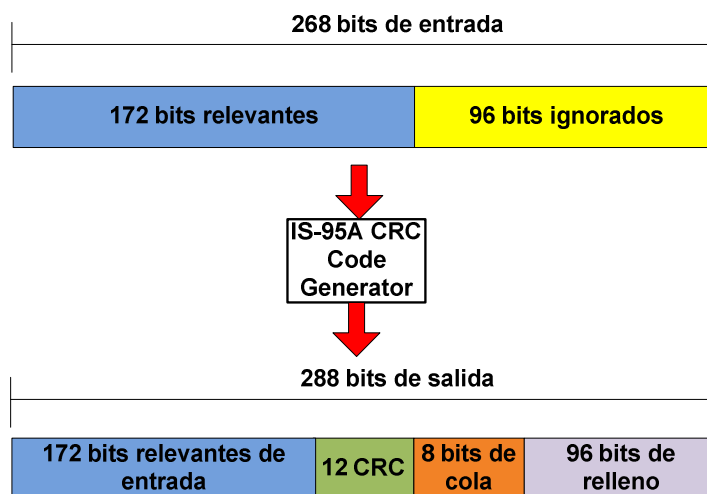
Para el análisis del canal de tráfico a 9600 bps se tiene:

**Tabla 4.7** Bits relevantes de entrada y salida del bloque IS-95A CRC Generator del canal de Trafico a 9600 bps.

Tipo de Canal	Bits de entrada relevantes	Bits CRC	Bits de cola	Bits de salida relevantes.
<i>Tráfico, Rate Set I, Full Rate</i>	172	12	8	192

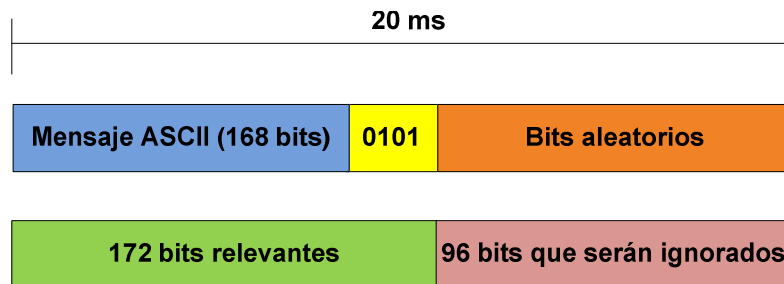
El número de bits de entrada y salida relevantes para el canal de Tráfico con Rate Set I, Full Rate es: para la entrada 172 bits, para la salida 192 bits (12 bits CRC *F* y 8 bits de cola *T*).

Como la entrada y la salida de este generador es fija (**268 y 288 elementos respectivamente**), 96 bits de la entrada serán ignorados, así como 96 bits en la trama de salida serán puestos a cero (bits de relleno) por el generador **IS-95A CRC Generator**.



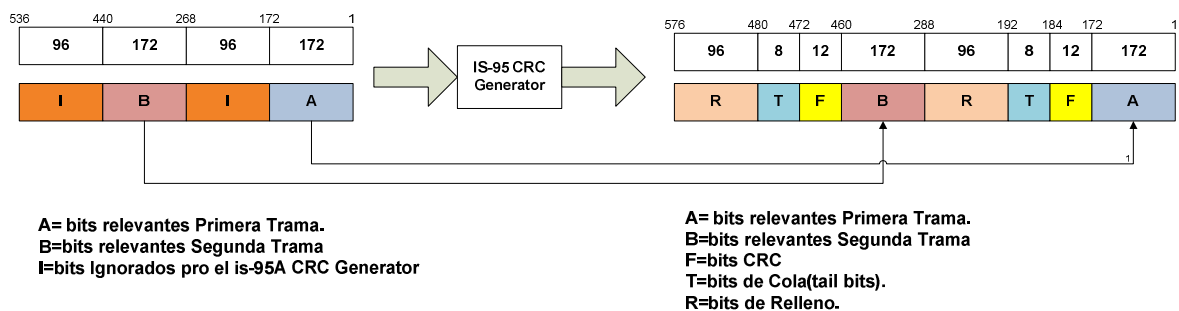
**Figura 4.80** Tramas de entrada y salida para el IS-95A Code Generator.

Como se puede ver en la figura, 96 bits producidos por el generador de datos, no intervienen en la codificación de redundancia cíclica. Por lo tanto, se pierden. Debido a esta consideración, con el objetivo de verificar cada etapa de la simulación, se generó la trama descrita en el apartado anterior.



**Figura 4.81** Trama de datos de entrada generada para el IS-95A Code Generator.

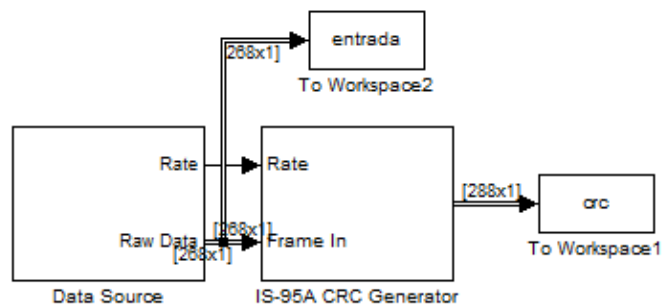
Para la verificación del funcionamiento de este bloque se va a realizar el análisis de los primeros 536 bits (172 bits relevantes de la primera trama + 96 bits ignorados + 172 bits relevantes de segunda trama + 96 bits ignorados) producidos por el generador de datos.



**Figura 4.82** Estructuras de las tramas de entrada y salida del bloque IS-95A CRC Generator.

Primero se observan los primeros 172 bits de datos generados y los 288 bits generados después del bloque **IS-95A CRC Generator**.

Para realizar la captura de los datos obtenidos antes y después del bloque **IS-95A CRC Generator** se utiliza el bloque **To Workspace** con la variable **crc** y la variable **entrada**.



**Figura 4.83** Verificación de la operación del bloque IS-95A CRC Generator.

El vector columna **entrada** en el Workspace se transforma a un vector fila con el comando:

```
 analisis_entrada=entrada';
```

Se almacenan los 172 bits en la variable **texto\_analisis** con el siguiente comando:

```
 texto_analisis=( analisis_entrada(1:172)); % trama A de entrada
```

El vector columna **crc** en el Workspace se transforma a un vector fila con el comando:

```
 salida_crc=crc';
```

Se almacena en la variable **crc\_comparacion** los primeros 172 bits del vector **salida\_crc** y se almacenan los 20 últimos en la variable **F\_T** (12 bits CRC **F** + 8 bits de cola **T**) con los comandos:

```
 crc_comparacion=salida_crc(1:172); %trama A de salida
 F_T=salida_crc(173:192); %bits F y T de primera trama de salida
```

Se almacena en la variable **bit\_relleno** los 96 bits del vector **salida\_crc(193:288)** que representan los bits de relleno que genera el bloque para completar una trama de 288 elementos. Se utilizan los comandos:

```
 bit_relleno=salida_crc(193:288); %bits R de primera trama de salida
```

Se realiza una comparación del vector **texto\_analisis** con el vector **crc\_comparacion** para verificar que no haya ningún cambio de los datos originales. Esto se realiza mediante una operación XOR y la suma del resultado:

$$\text{comparacion} = \text{suma}(\text{texto\_analisis XOR crc\_comp})$$

$$\text{comparacion} = \text{suma}(A \text{ entrada XOR } A \text{ salida})$$

```
comparacion=sum(xor(texto_analisis,crc_comp));
```

```
>> analisis_entrada=entrada';
>> texto_analisis=(analisis_entrada(1:172));
>> salida_crc=crc';
>> crc_comparacion=salida_crc(1:172);
>> F_T=salida_crc(173:192);
>> bit_relleno=salida_crc(193:288);
>> comparacion=sum(xor(texto_analisis,crc_comp))

comparacion =

    0
```

**Figura 4.84** Comparación de los campos de la primera trama de entrada y salida del bloque IS-95A CRC Generator.

Se obtiene un valor de cero, lo que quiere decir que ambos vectores son idénticos y la información no ha cambiado, por tanto **A de entrada es igual a A de salida**. A continuación se muestra el vector **F\_T (bits F y T)**:

```
>> F_T

F_T =

Columns 1 through 12      Bits F
1     1     1     1     0     1     0     0     0     0     1     1

Columns 13 through 20    Bits T
0     0     0     0     0     0     0     0
```

**Figura 4.85** Vector F\_T muestra los valores del campo CRC y los tail bits de la primera trama de salida del bloque IS-95A CRC Generator.

Se muestra el vector *bit\_relleno (bits R)*:

```

Bit_relleno =
Columns 1 through 15
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 19 through 36
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 37 through 54
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 55 through 72
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 73 through 90
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 91 through 96
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

```

**Figura 4.86** Bits de relleno para la trama de la primera trama de salida del bloque IS-95A CRC Generator.

Se puede observar que se está cumpliendo la función del bloque **IS-95A CRC Generator** con la primera trama de datos y que se han generado 96 bits de relleno iguales a cero para completar una trama de 288 elementos.

Ahora se analiza la segunda trama de bits relevantes.

Se almacenan los 172 bits relevantes de la siguiente trama de la en la variable **texto\_analisis2** con el siguiente comando:

```
texto_analisis2= analisis_entrada(269:440); %trama B de entrada
```

Se realiza el análisis de los bits a la salida del bloque **IS-95A CRC Generator** desde el elemento **salida\_crc(289)** hasta el elemento **salida\_crc(460)**. Estos bits se almacenan en la variable **crc\_comparacion2**:

```
crc_comparacion2=salida_crc(289:460); %trama B de salida
```

y se almacenan los 20 últimos bits en la variable **F\_T2** (12 bits CRC **F** + 8 bits de cola **T**) con los comandos:



```
F_T2=salida_crc(461:480); %bits F y T de segunda trama de salida
```

Se almacena en la variable **bit\_relleno2** los 96 bits del vector **salida\_crc(481:576)** que representan los bits de relleno que genera el bloque para completar una trama de 288 elementos. Se utilizan los comandos:

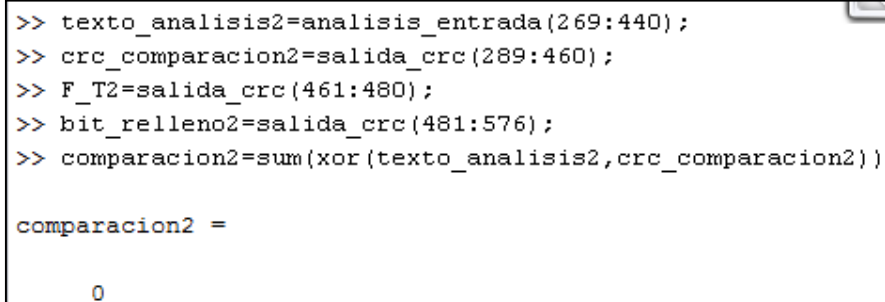
```
bit_relleno2=salida_crc(481:576); %bits R de la segunda trama de salida
```

Se realiza una comparación del vector **texto\_analisis2** con el vector **crc\_comp2** para verificar que no haya ningún cambio de los datos originales de la segunda trama:

$$\text{comparacion2} = \text{suma}(\text{texto\_analisis2} \text{ XOR } \text{crc\_comparacion2})$$

$$\text{comparacion} = \text{suma}(B \text{ entrada XOR } B \text{ salida})$$

```
comparacion2=sum(xor(texto_analisis2,crc_comparacion2));
```



```
>> texto_analisis2= analisis_entrada(269:440);
>> crc_comparacion2=salida_crc(289:460);
>> F_T2=salida_crc(461:480);
>> bit_relleno2=salida_crc(481:576);
>> comparacion2=sum(xor(texto_analisis2,crc_comparacion2))

comparacion2 =

     0
```

**Figura 4.87** Comparación de los campos de la primera trama de entrada y salida del bloque IS-95A CRC Generator.

Se obtiene un valor de cero, lo que quiere decir que ambos vectores son idénticos y la información no ha cambiado, por tanto **B de entrada es igual a B de salida..** A continuación se muestra el vector **F\_T2 (bits F y T)**:

```

>> F_T2
F_T2 =
Columns 1 through 12      Bits F
1   1   1   1   0   1   0   0   0   0   1   1
Columns 13 through 20    Bits T
0   0   0   0   0   0   0   0

```

**Figura 4.88** Vector F\_T muestra los valores del campo CRC y los tail bits de la segunda trama de salida del bloque IS-95A CRC Generator.

Se muestra el vector *bit\_relleno2* (*bits R de segunda trama de salida*):

```

bit_relleno2 =
Columns 1 through 18
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
Columns 19 through 36
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
Columns 37 through 54
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
Columns 55 through 72
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
Columns 73 through 90
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
Columns 91 through 96
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

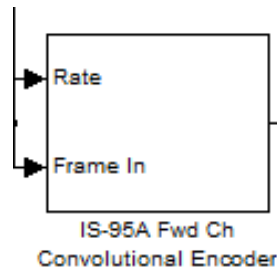
```

**Figura 4.89** Bits de relleno para la trama de la segunda trama de salida del bloque IS-95A CRC Generator.

Se puede observar que se está cumpliendo la función del bloque **IS-95A CRC Generator** con la segunda trama de datos y que se han generado 96 bits de relleno iguales a cero para completar una trama de 288 elementos como con la primera trama.

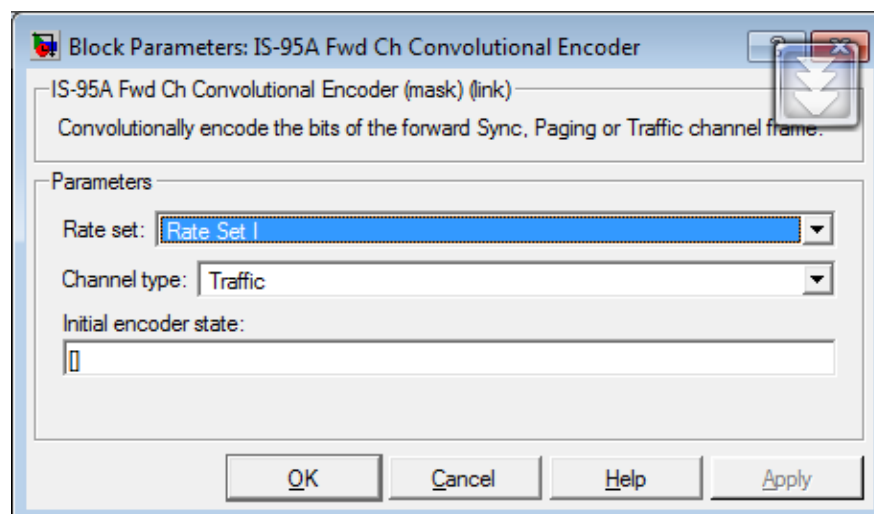
## Codificación Convolutiva

Para realizar la codificación convolutiva en el demo, se utiliza el bloque **IS-95A CRC Fwd Convolutional Encoder**.



**Figura 4.90** Bloque IS-95A Fwd Ch Convolutional Encoder

- Este bloque codifica convolutivamente la trama de datos de entrada. Para el canal de tráfico, la trama de entrada al bloque incluye datos de información, bits CRC y de cola.
- El codificador convolutivo es un codificador  $\frac{1}{2}$ , teniendo una longitud de restricción  $K = 9$ . Los polinomios generadores son 753 y 561.
- El estado del codificador al inicializar es de todos los estados como ceros.
- El tamaño del puerto **Frame In** es igual al valor máximo posible 288.
- El tamaño del puerto **Output** es igual al mayor posible 576.



**Figura 4.91** Configuración del bloque IS-95A Fwd Ch Convolutional Encoder

### Número de bits de entrada y salida relevantes.

El tamaño del puerto de entrada ***Frame In debe ser obligatoriamente 288 elementos***. El número de bits añadidos a la trama depende del canal y su tasa de transmisión. Por lo tanto ***el tamaño del puerto de salida es de 576 elementos***, siendo este el doble de número de bits ingresados (tasa 1/2). En la siguiente tabla se muestra el número de bits relevantes de entrada y salida para los distintos canales de IS-95A.

**Tabla 4.8** Bits relevantes del bloque IS-95 Fwd Ch Convolutional Encoder.

Tipo de Canal	Bits de entrada relevantes	Bits de salida relevantes
Sync, Eighth Rate	32	64
Paging, Full Rate	192	384
Paging, Half Rate	96	192
<b><i>Traffic, Rate Set I, Full Rate</i></b>	<b><i>192</i></b>	<b><i>384</i></b>
Tráfico, Rate Set I, Half Rate	96	192
Tráfico, Rate Set I, Quarter Rate	48	96
Tráfico, Rate Set I, Eighth Rate	24	48
<b><i>Tráfico, Rate Set II, Full Rate</i></b>	<b><i>288</i></b>	<b><i>576</i></b>
Tráfico, Rate Set II, Half Rate	144	288
Tráfico, Rate Set II, Quarter Rate	72	144
Tráfico, Rate Set II, Eighth Rate	36	72

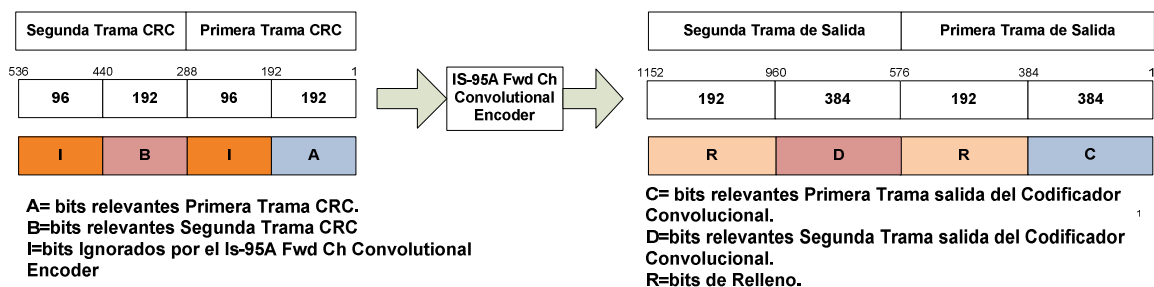
Para el análisis del canal de tráfico a 9600 bps se tiene:

**Tabla 4.9** Bits relevantes del bloque IS-95 Fwd Ch Convolutional Encoder para el canal de Tráfico a 9600 bps

Tipo de Canal	Bits de entrada relevantes	Bits de salida relevantes.
<b>Tráfico, Rate Set I, Full Rate</b>	<b>192</b>	<b>384</b>

El número de bits de entrada y salida relevantes para el canal de Tráfico con Rate Set I, Full Rate es: para la entrada 192 bits, para la salida 384 bits (192 X 2 = 384 bits). El bloque ignora bits de entrada pasados de los bits relevantes y pone a cero a los bits de salida pasados de los bits relevantes.

Como la entrada y la salida de este generador es fija (**288 y 576 elementos respectivamente**), 96 bits (bits de relleno) de la entrada serán ignorados, así como 192 bits en la trama de salida serán puestos a cero (bits de relleno) por el bloque **IS-95A Fwd Ch Convolutional Encoder**.



**Figura 4.92** tramas de entrada y salida del bloque IS-95A Fwd Ch Convolutional Encoder.

Recordando, los polinomios generadores implementados en el codificador convolucional para el canal de tráfico en forma de polinomio son:

$$1 + x + x^2 + x^3 + x^5 + x^7 + x^8$$

$$1 + x^2 + x^3 + x^4 + x^8$$

Para comprobar el funcionamiento de este bloque, se analizan los 10 primeros bits del vector **salida\_crc** que se generó en anteriormente y se calculará el código convolucional correspondiente. Primero se toman los 10 bits con el comando:

```

 analisis_convolucional=salida_crc(1:10);  %primeros 10 bits de
 trama A

```

```

>> analisis_convolucional=salida_crc(1:10)

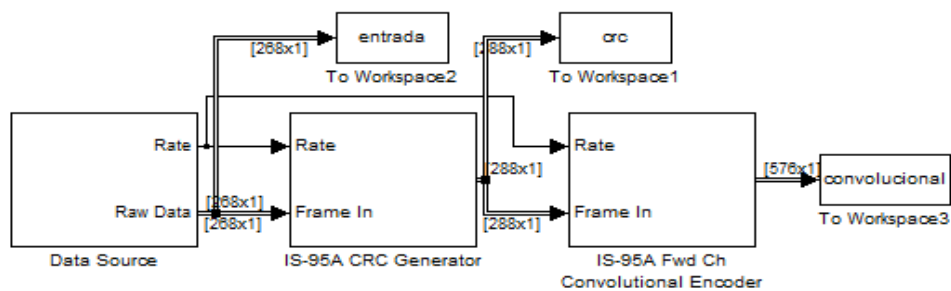
 analisis_convolucional =

      0      0      0      0      1      0      1      0      1      1

```

**Figura 4.93** Primeros 10 bits de entrada al bloque IS-95A Fwd Ch Convolutional Encoder

Para realizar la captura de los datos obtenidos después del bloque **IS-95A Fwd Ch Convolutional Encoder** se utiliza el bloque **To Workspace** con la variable **convolucional**.



**Figura 4.94** Verificación de la operación del bloque IS-95A Fwd Ch Convolutional Encoder.

El vector columna **convolucional** en el Workspace se transforma a un vector fila con el comando:

```

 salida_convolucional=convolucional';

```

Se observan los 20 bits de la salida del codificador convolucional con el comando:

```

 analisis_convolucional_comparacion=salida_convolucional(1:20);
 %primeros 20 bits de trama C

```

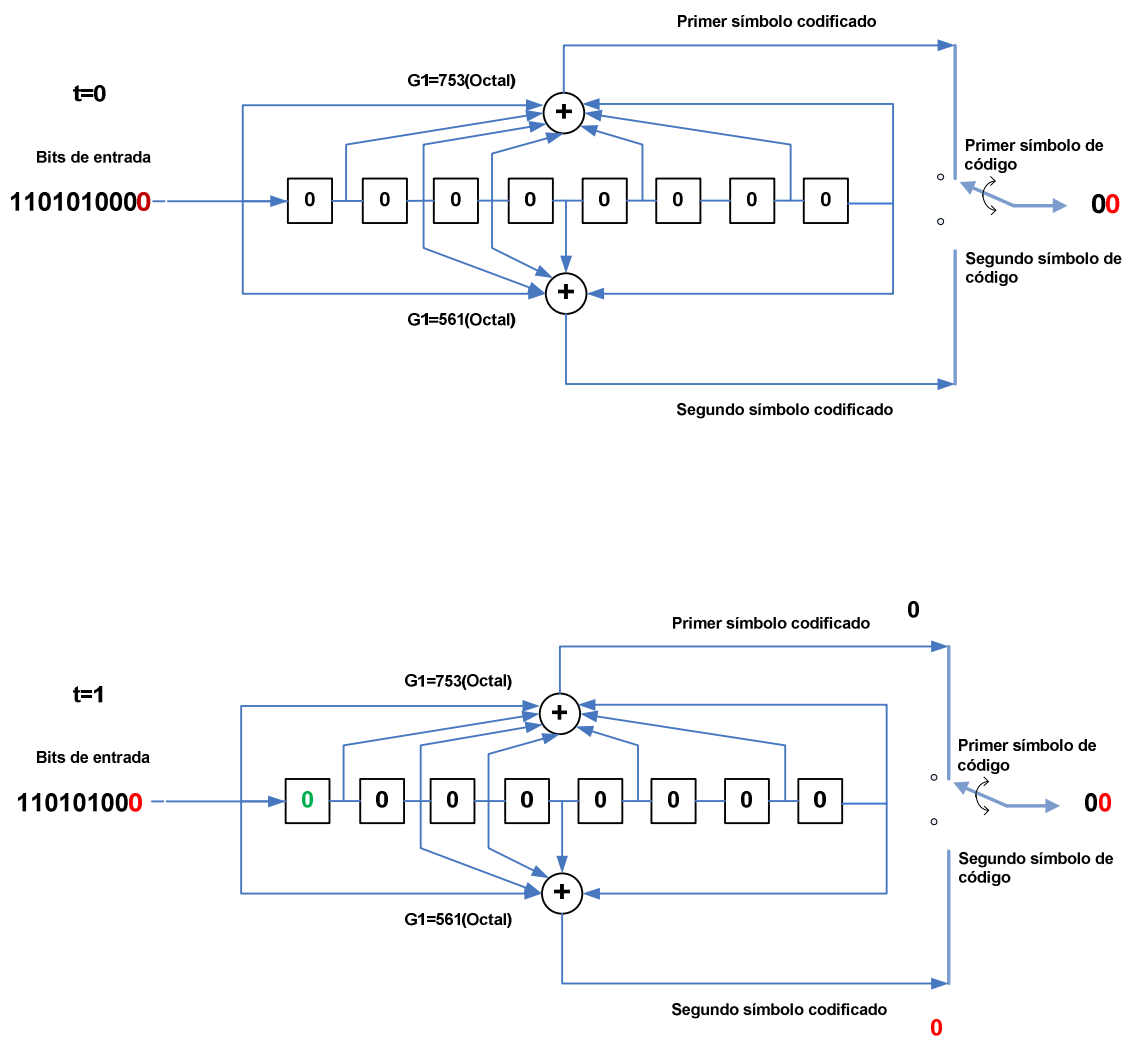
```
>> analisis_convolucional_comparacion=salida_convolucional(1:20)
analisis_convolucional_comparacion =

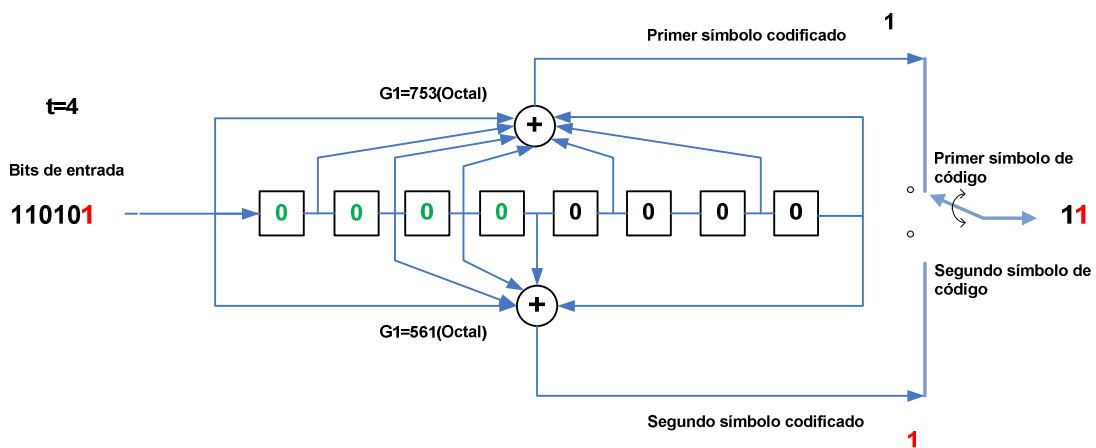
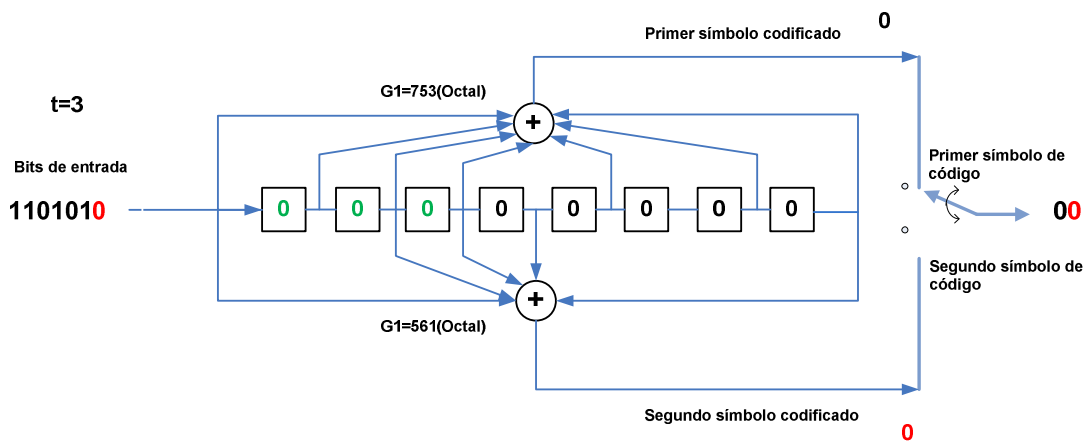
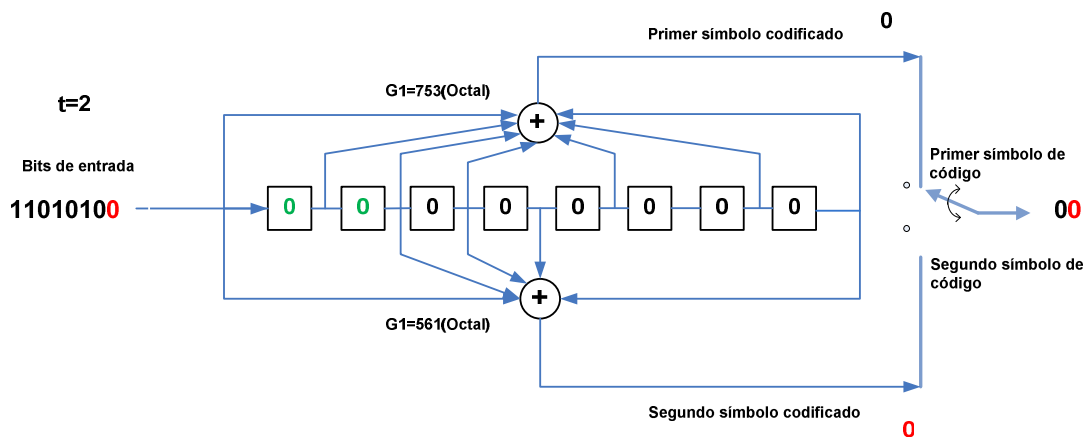
Columns 1 through 11
    0     0     0     0     0     0     0     0     1     1     1

Columns 12 through 20
    0     0     0     0     1     0     1     0     0
```

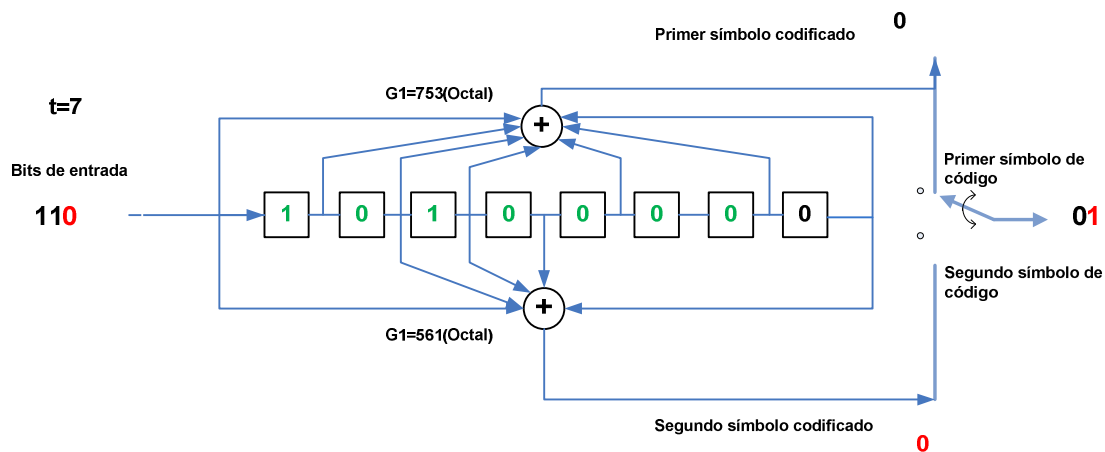
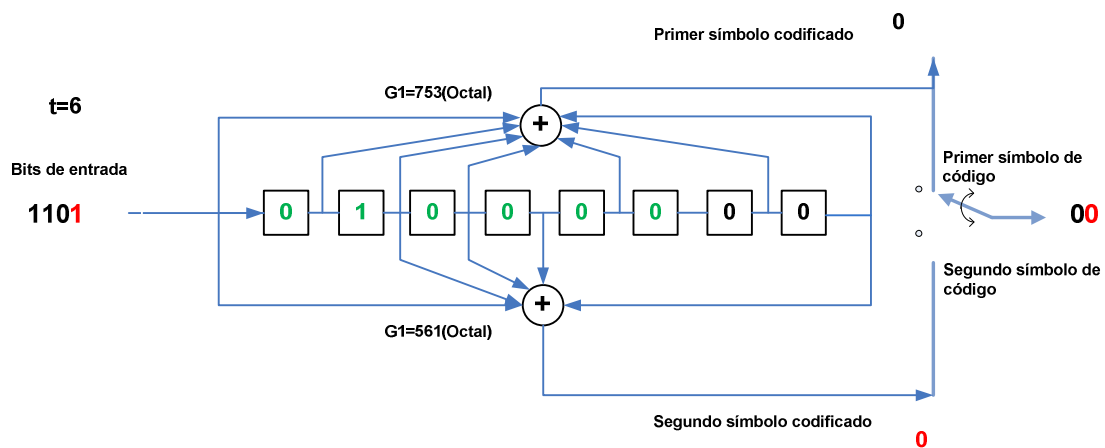
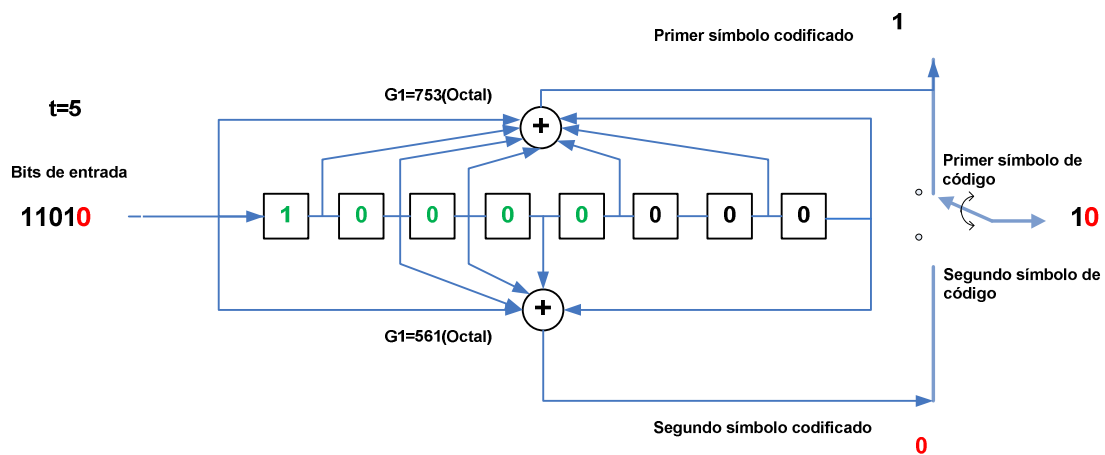
Figura 4.95 20 Primeros bits de salida del bloque IS-95A Fwd Ch Convolutional Encoder.

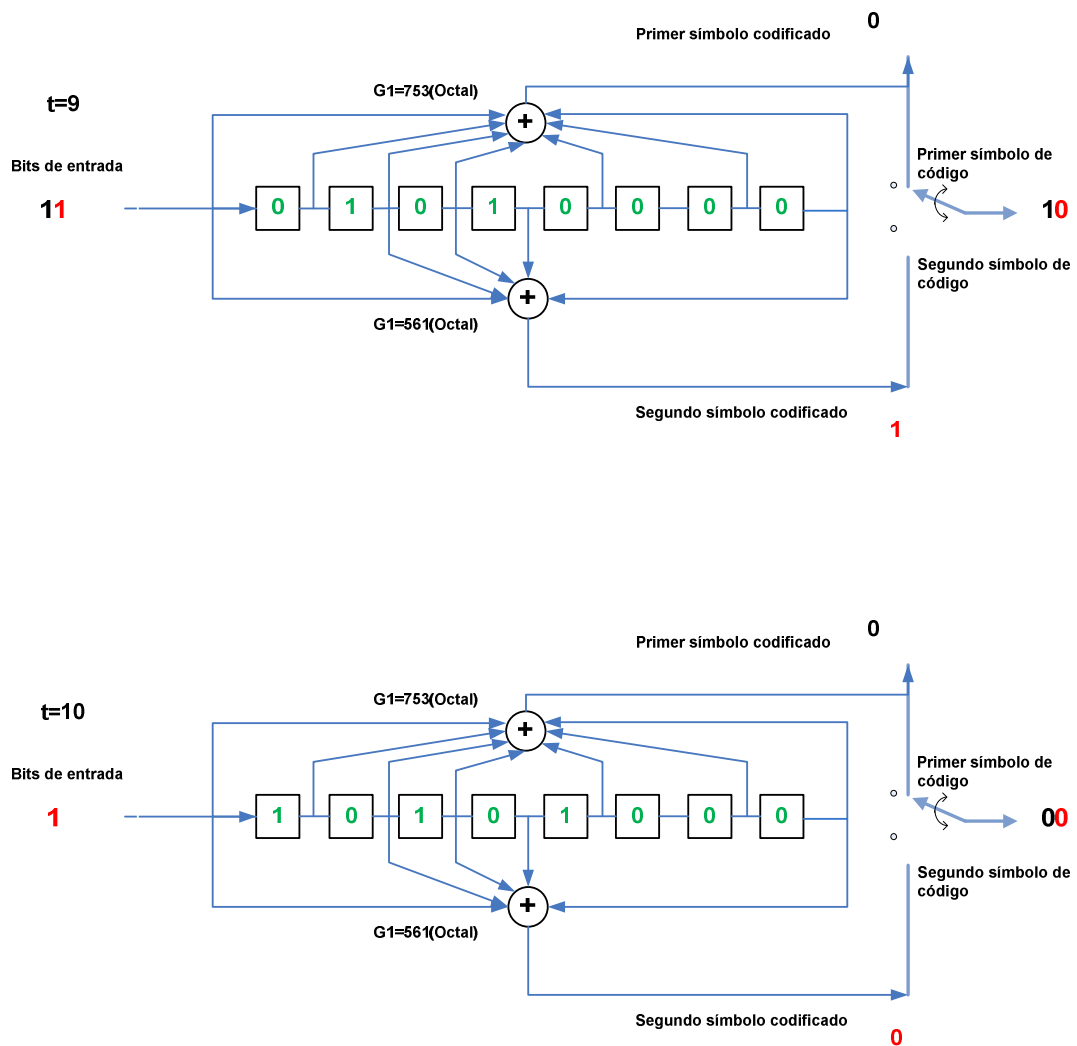
Ahora para verificar que se generó el código correspondiente, se procede a realizar la generación de los bits codificados:











**Figura 4.96** Codificación de los primeros 10 bits de entrada al bloque IS-95A Fwd Ch Convolutional Encoder.

Para los primeros 10 valores [0000101011], la secuencia codificada es: [00000000111000010100]. Con esto se verifica que el bloque ha codificado correctamente a los bits de entrada.

Se observan los 192 bits del vector **salida\_convolutional(385:576)** para comprobar que sean solo bits de relleno con el siguiente comando:

```
bit_relleno_convolutional=sum(salida_convolutional(385:576));
%bits R de primera trama de salida
```

```

>> bit_relleno_convolutacional= sum(salida_convolutacional(385:576))

bit_relleno_convolutacional =

    0

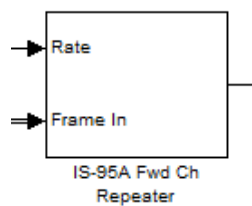
```

**Figura 4.97** Bits de relleno de la primera trama de salida del bloque IS-95A Fwd Ch Convolutional Encoder.

Se puede observar que la suma de los bits de relleno es igual a cero y por tanto todos estos bits son cero.

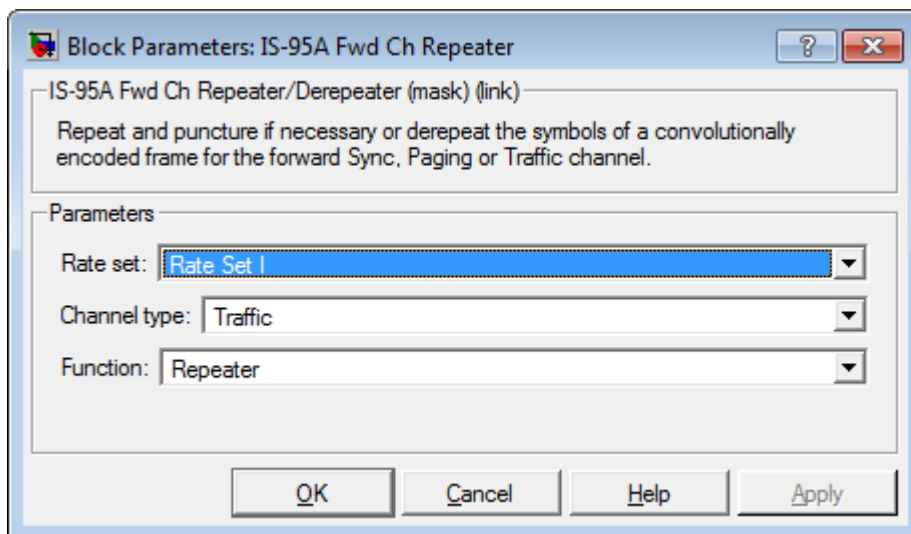
### **Repetidor**

En el canal de tráfico con Full Rate no se realiza repetición de símbolos. El bloque **IS-95A Fwd Ch Repeater** repite o quita los símbolos de trama codificados convolutionalmente para el canal Forward.



**Figura 4.98** Bloque IS-95A Fwd Ch Repeater

- En el Rate Set I, las operaciones de repetición resultan en un tamaño de trama de 384 símbolos para todas las tasas de datos.
- En el Rate Set II, las operaciones de repetición resultan en un tamaño de trama de 576 símbolos para todas las tasas de datos, pero a los datos repetidos se realiza puncturing antes de partir del puerto de salida.
- El tamaño de la trama de la entrada y salida es 576 y 384 respectivamente, en canales de Tráfico y Paging.



**Figura 4.99** Configuración del bloque IS-95A Fwd Ch Repeater.

Se puede notar que está configurado para operar en modo Rate Set I, para el canal de tráfico.

### Número de bits de entrada y salida relevantes

El tamaño del puerto de salida para el canal de Tráfico, Full Rate, y Rate Set I es de 384 símbolos.

**Tabla 4.10** Bits relevantes del bloque IS-95A Fwd Ch Repeater.

Tipo de Canal	Bits relevantes para la entrada del Repetidor y la salida del Derepetidor
Sync, Eighth Rate	64
Paging, Full Rate	384
Paging, Half Rate	192
<b>Traffic, Rate Set I, Full Rate</b>	<b>384</b>
Tráfico, Rate Set I, Half Rate	192
Tráfico, Rate Set I, Quarter Rate	96
Tráfico, Rate Set I, Eighth Rate	48

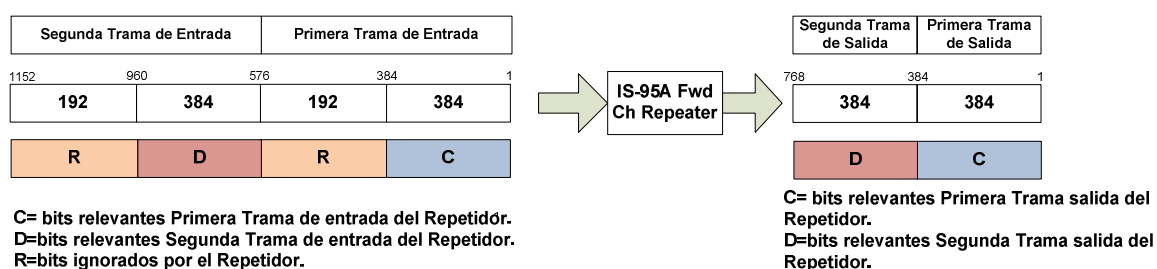
Tráfico, Rate Set II, Full Rate	576
Tráfico, Rate Set II, Half Rate	288
Tráfico, Rate Set II, Quarter Rate	144
Tráfico, Rate Set II, Eighth Rate	72

Para el análisis del canal de tráfico a 9600 bps se tiene:

**Tabla 4.11** Bits relevantes del bloque IS-95A Fwd Ch Repeater para el canal de tráfico a 9600 bps.

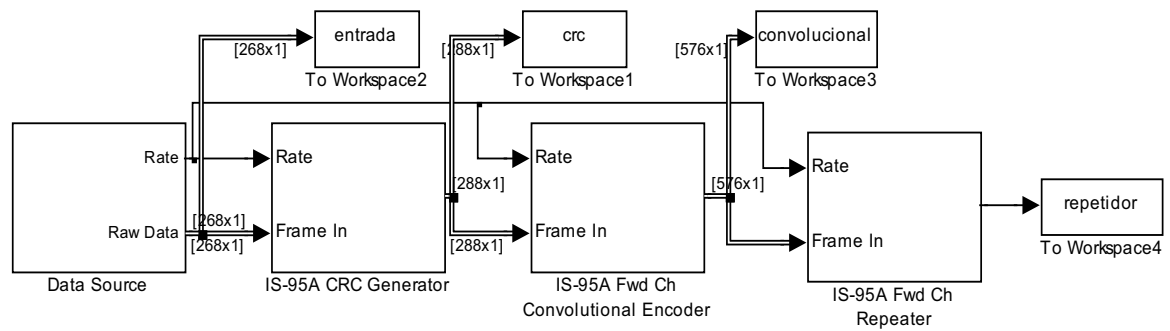
Tipo de Canal	Bits de entrada relevantes	Bits de salida relevantes.
<i>Tráfico, Rate Set I, Full Rate</i>	384	384

El número de bits de salida relevantes para el canal de Tráfico con Rate Set I, Full Rate es: 384 bits. El bloque ignora bits de entrada pasados de los bits relevantes y pone a cero a los bits de salida pasados de los bits relevantes.



**Figura 4.100** Tramas de entrada y salida para el bloque IS-95A Fwd Ch Repeater.

Para realizar la captura de los datos obtenidos después del bloque *IS-95A Fwd Ch Repeater* se utiliza el bloque *To Workspace* con la variable *repetidor*.



**Figura 4.101** Verificación de la operación del bloque IS-95A Fwd Ch Repeater.

El vector columna **repetidor** en el Workspace se transforma a un vector fila con el comando:

```
salida_repetidor=repetidor';
```

Se obtienen los 384 bits de entrada al repetidor con el comando:

```
 analisis_repetidor=salida_convolucional(1:384); %trama C de entrada
```

Se obtienen los 384 bits de salida del repetidor con el comando:

```
 analisis_repetidor_comparacion=salida_repetidor(1:384); %trama C de salida
```

Se compara ambos para comprobar que son iguales:

```
 comparacion = suma( analisis_repetidor XOR analisis_repetidor_comparacion)
```

```
 comparacion = suma(C entrada XOR C salida)
```

```
 comparacion_repetidor=sum(xor( analisis_repetidor, analisis_repetidor_comparacion))
```

```

>> salida_repetidor=repetidor';
>> analisis_repetidor=salida_convolutacional(1:384);
>> analisis_repetidor_comparacion=salida_repetidor(1:384);
>> comparacion_repetidor=sum(xor(analisis_repetidor, analisis_repetidor_comparacion))

comparacion_repetidor =

    0

```

**Figura 4.102** Comparación de los campos de la primera trama de entrada y salida del bloque IS-95A Fwd Ch Repeater.

Se puede observar que la comparación resulta cero y por tanto los vectores son iguales. **Los primeros 384 bits de la trama de entrada corresponden a los 384 bits de la trama de salida del Repetidor.**

Ahora se compara la segunda trama de entrada con la segunda trama de salida. No se toman en cuenta los valores no relevantes del vector **salida\_convolutacional**:

```

analisis_repetidor =salida_convolutacional(577:960); %trama D de
entrada

```

Del vector **salida\_repetidor** se toman los valores correspondientes a la segunda trama de salida:

```

analisis_repetidor_comparacion=salida_repetidor(385:768); %trama D
de salida

```

Se compara ambos vectores para comprobar que los vectores son iguales:

```

comparacion = suma(analisis_repetidor XOR analisis_repetidor_comparacion)

```

$$\text{comparacion} = \text{suma}(\text{D entrada XOR D salida})$$

```

comparacion_repetidor=sum(xor(analisis_repetidor, analisis_repetido
r_comparacion))

```

```

>> analisis_repetidor=salida_convolutacional(577:960);
>> analisis_repetidor_comparacion=salida_repetidor(385:768);
>> comparacion_repetidor=sum(xor(analisis_repetidor, analisis_repetidor_comparacion))

comparacion_repetidor =

    0

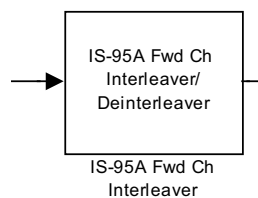
```

**Figura 4.103** Comparación de los campos de la segunda trama de entrada y salida del bloque IS-95A Fwd Ch Repeater.

Se puede observar que la comparación resulta cero y por tanto los vectores son iguales. **Los primeros 384 bits de la segunda trama de entrada corresponden a los 384 bits de la segunda trama de salida del Repetidor.** Por lo tanto en este bloque se eliminan los bits no relevantes, ignorados en la salida del codificador convolutacional.

### Interleaver

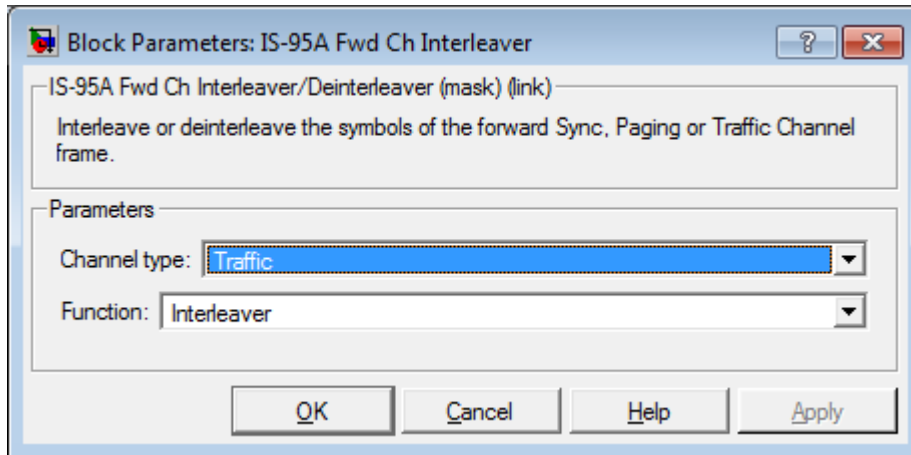
Para realizar la operación de Interleaving en el demo se utiliza el bloque **IS-95A Fwd Ch Interleaver**.



**Figura 4.104** Bloque IS-95A Fwd Ch Interleaver

- Este bloque entrelaza (interleaves) o desentrelaza (deinterleaves) los símbolos de la trama entrante. Los lapsos de tiempo de los interleaver de los canales de Paging y Tráfico son de 20 ms, equivalentes a 384 símbolos.
- El interleaver escribe los símbolos de códigos en la matriz columna por columna para todos los datos de entrada. El deinterleaver restaura a los símbolos en su orden original.



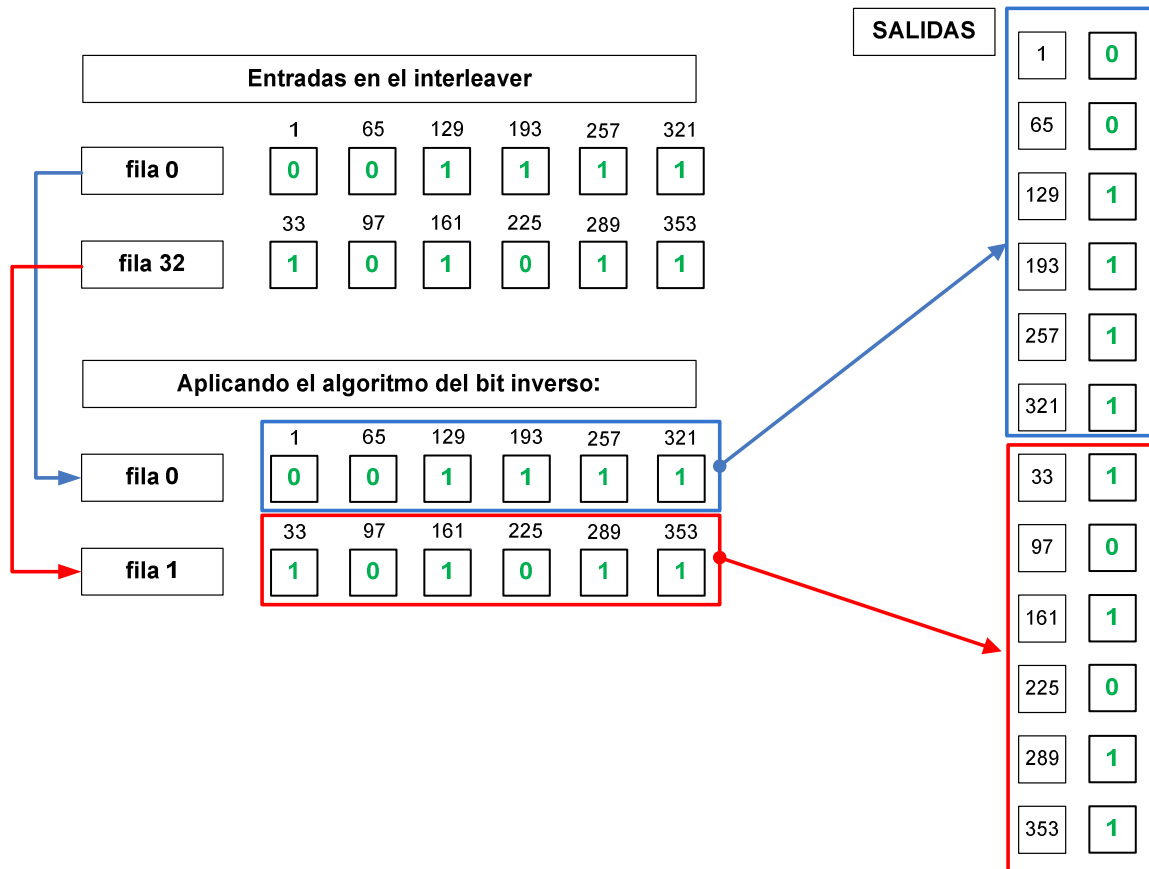


**Figura 4.105** Configuración del bloque IS-95A Fwd Ch Interleaver

Para comprobar el funcionamiento de este bloque. Se debe conocer el proceso del interleaver en canal Forward de Tráfico.

Como se explicó antes, los 384 símbolos de la trama entrada del Interleaver forman una matriz de entrada equivalente de 64 X 6, donde los símbolos se escriben por columnas. Luego cada fila es indexada para realizar el algoritmo del **bit inverso**. El Interleaver reordena las filas de la matriz de acuerdo al resultado del algoritmo. Por último, a la salida el interleaver se leen los elementos por filas de la matriz equivalente resultante.

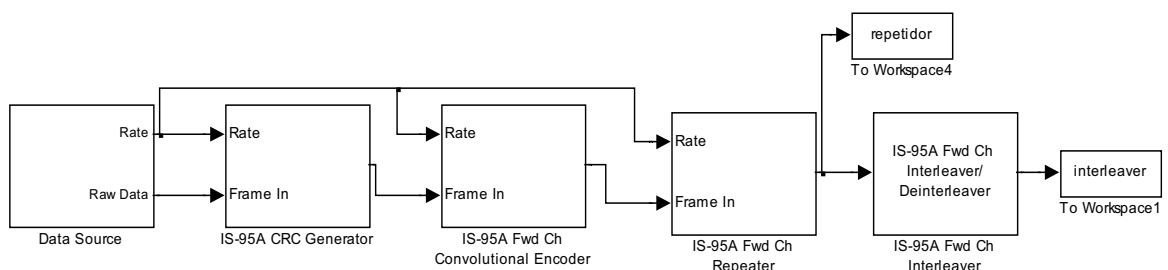
Por ejemplo tomando las filas 0 y 32 de la matriz de entrada equivalente, el proceso del bit inverso hace que estas filas se reordenen, tal como se muestra en la figura:



**Figura 4.106** Operación del Interleaver en el canal Forward de Tráfico IS-95A

Como se puede ver, las salidas se leen por filas, de acuerdo a su indexación final.

Para el análisis del correcto funcionamiento del bloque **Interleaver** se toman las siguientes consideraciones:



**Figura 4.107** Verificación de la operación del bloque IS-95A Fwd Ch Interleaver.

Se almacenan los datos en los bloques to Workspace **repetidor e interleaver**.

El vector columna **repetidor** en el Workspace se transforma a un vector fila con el comando:

```
a=repetidor';
```

Para realizar la captura de los datos obtenidos después del bloque **IS-95A Fwd Ch Interleaver** se utiliza el bloque **To Workspace** con la variable **interleaver**.

El vector columna **interleaver** en el Workspace se transforma a un vector fila con el comando:

```
b=interleaver';
```

En las variables **fila\_0** y **fila\_32** se almacenan los elementos correspondientes a la fila 0 y a la fila 32 de la matriz equivalente de entrada, respectivamente, como se observó en la figura:

```
fila_0=[a(1) a(65) a(129) a(193) a(257) a(321)];
fila_32=[a(33) a(97) a(161) a(225) a(289) a(353)];
```

Ahora se quiere verificar que los seis primeros y los siguientes seis elementos de la salida del **Interleaver** corresponden a los elementos de los vectores **fila\_0** y **fila\_32** de la matriz de entrada, respectivamente. **Con esto, se comprueba que se realiza el proceso de Interleaving definido por estándar.**

Se definen las salidas de la siguiente forma:

```
c=b(1:6);
d=b(7:12);
```

Donde **c** y **d** almacenan los primeros doce elementos de la salida del interleaver y **b** es el vector de salida del interleaver.

Comparando c con **fila\_0** y d con **fila\_32** se tiene:

```

>> compara=xor(filas_0,c)

compara =

     0     0     0     0     0     0

>> compara=xor(filas_32,d)

compara =

     0     0     0     0     0     0

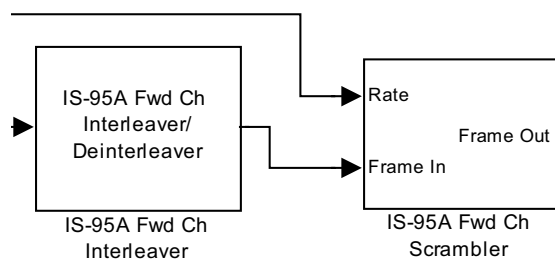
```

**Figura 4.108** Comparación de los elementos de la fila 0 y fila 32 con la salida del interleaver.

Se observa que en los dos casos, se obtienen vectores nulos. Por lo tanto, los vectores comparados son iguales y se ha comprobado el funcionamiento correcto del bloque.

### Scrambler

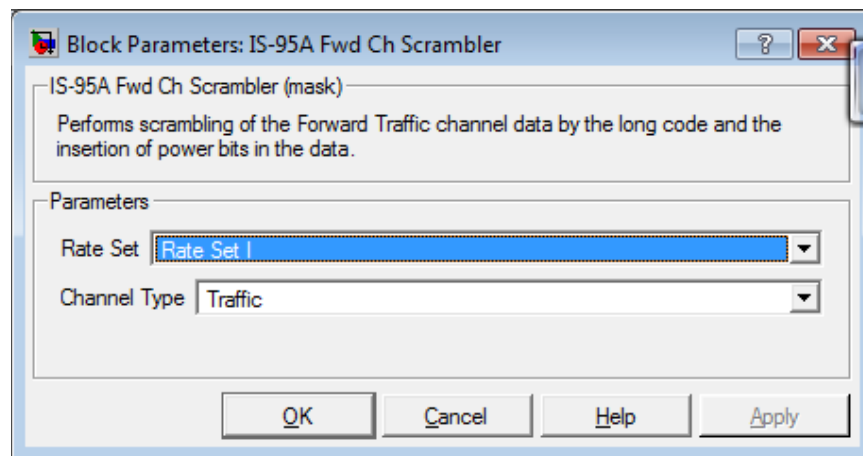
Se utiliza el bloque **IS-95A Fwd Ch Scrambler**:



**Figura 4.109** Bloque IS-95A Fwd Ch Scrambler

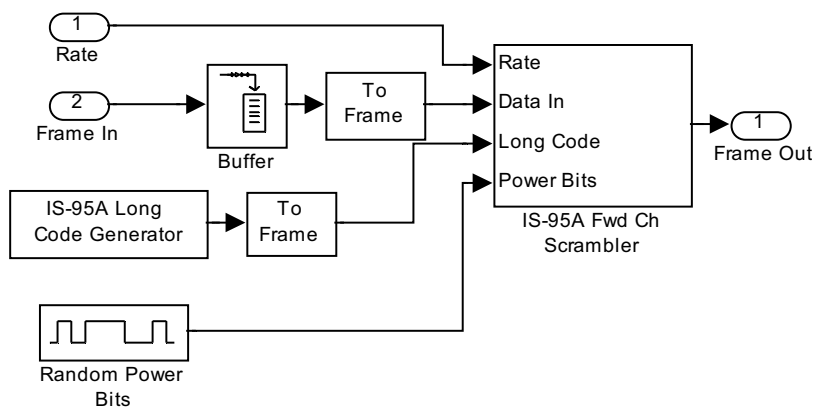
- Este bloque realiza la acción de **scrambling e inserta bits de control de potencia** dentro de la corriente de datos de entrada.
- La función scrambling se aplica a canales Forward Tráfico y Paging. La inserción de bits de potencia, que se aplican solo a los canales de Tráfico, ocurre a una tasa de datos con 800 bps.
- Para el scrambling de los datos **Frame In**, el bloque combina los datos con los datos de entrada del código largo decimado mediante la suma módulo 2 y luego reemplaza algunos bits con los datos de entrada de bits de potencia.

- El Rate Set I y Rate Set II del canal de tráfico incluye 2 y 1 bits de control de potencia, por grupo de potencia respectivamente. Para el **Rate Set I**, los bits de potencia son insertados en dos localidades de símbolos consecutivos.
- Los últimos cuatro bits de código largo decimado en un grupo de potencia, determinan la posición de inserción del bit de potencia en el siguiente grupo.



**Figura 4.110** Configuración del bloque IS-95A Fwd Ch Scrambler.

Expandiendo el bloque se tiene:



**Figura 4.111** Bloque IS-95A Fwd Ch Scrambler Expandido.

El generador de código largo produce una secuencia PN larga decimada de acuerdo al estándar.

El proceso de decimación de cada 64<sup>th</sup> símbolo de la secuencia, hace que la tasa de chip se establezca en 19200 bps.

Luego será multiplicado por la salida del *interleaver* para aumentar la complejidad de la señal del usuario para que no pueda ser explotada o reconocida por un usuario no autorizado o atacante.

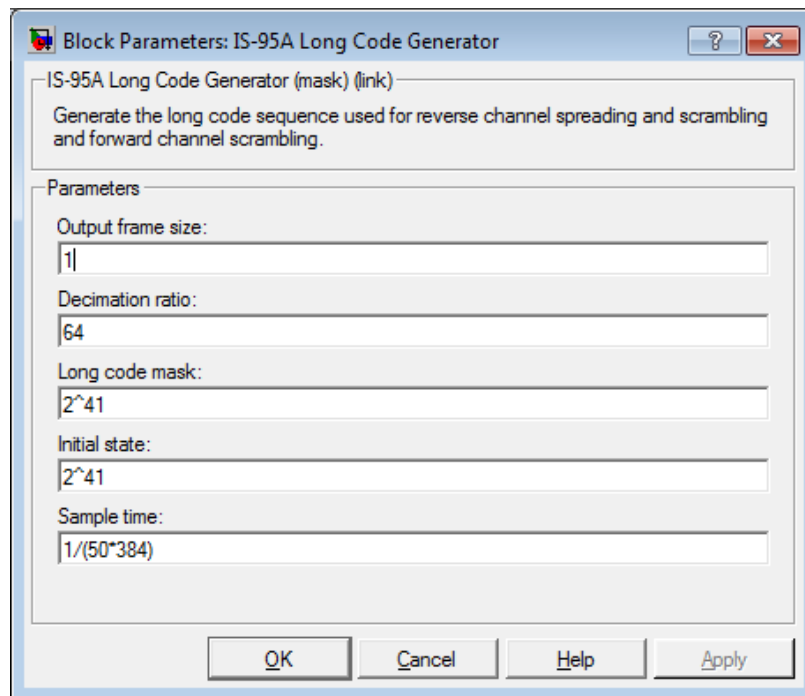


Figura 4.112 Configuración del bloque IS-95A Long Code Generator.

Como se puede ver, el bloque se encuentra configurado para que produzca la secuencia Larga decimada (1:64), a una tasa de 19200 Kbps.

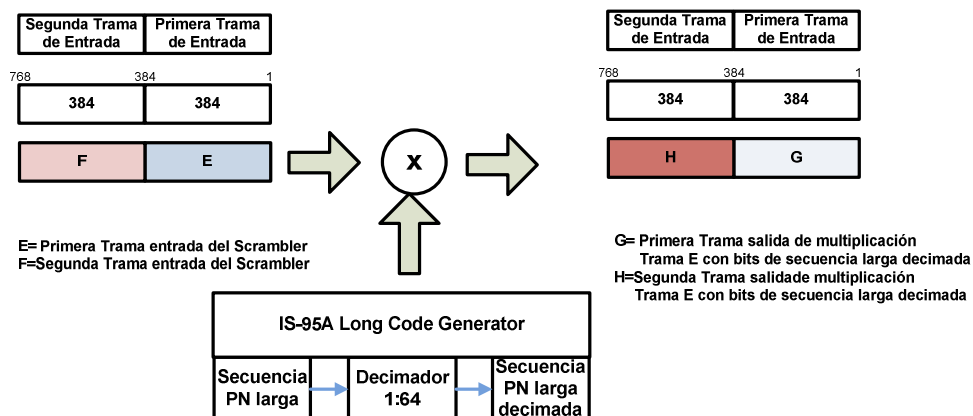
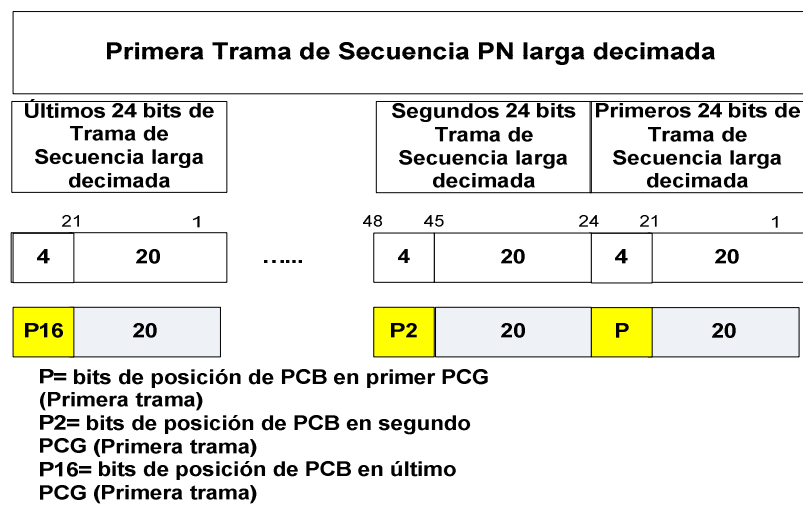


Figura 4.113 Multiplicación de la entrada con el código PN largo decimado.

En cuanto a la inserción de bits de potencia (PCB) el bloque realiza lo siguiente:

De la figura, las tramas de salida H y G de 384 bits se dividen en 16 grupos de control de potencia (**PCG**) de 24 bits cada una. El bit de potencia (**PCB**) tomará dos lugares de forma consecutiva dentro de los primeros 16 bits (el **PCB** se repite). La posición del PCB está dada por la secuencia larga decimada. Como se muestra en la figura:



**Figura 4.114** Bits que definen la posición de los bits de control de potencia en la salida del bloque IS-95A Fwd Ch Scrambler.

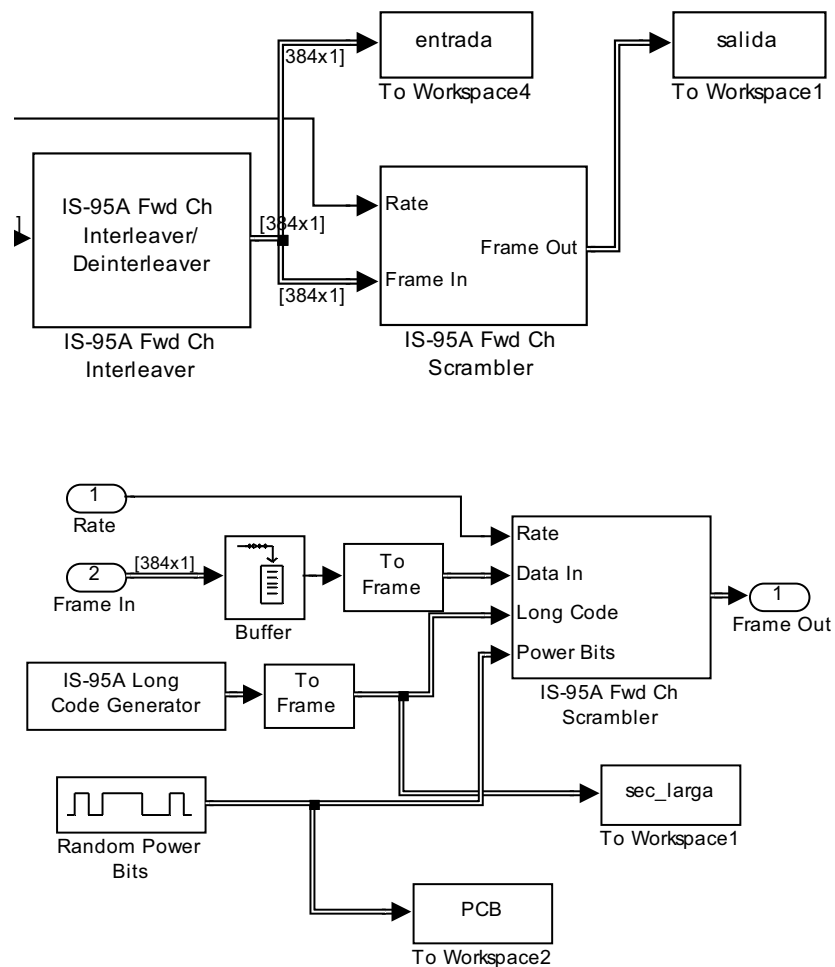
De cada 24 bits de la secuencia decimada, se toman en cuenta los cuatro últimos bits.

Realizado la conversión a decimal se define la posición que ocupara un PCB dentro de un PCG en la salida del bloque **IS-95A Fwd Ch Scrambler**.

Por lo tanto la salida constará de bits de información multiplicados por una secuencia PN larga decimada, más bits de control de potencia, en formato bipolar con lógica negativa.

La salida del **IS-95A Fwd Ch Scrambler** se da bit a bit para realizar el ensanchamiento y canalización de los datos, tal como se explicará en el siguiente apartado.

Para realizar la comprobación de este bloque, se realiza la captura de los datos obtenidos después a la entrada y salida del mismo, a la salida del generador de los PCBs (**Random Power Bits**) y a la salida del generador de la secuencia PN larga (**IS-95A Long Code Generator**) con el bloque **To Workspace** generando las variables **entrada**, **salida**, **PCB** y **sec\_larga** respectivamente:



**Figura 4.115** Bloques to Workspace para realizar el análisis del bloque IS-95A Fwd Ch Scrambler.

Para realizar el análisis de los grupos de control de potencia (**Power Control Groups**) se ha implementado la función **pcb.m**, que analiza las variables de entrada de 24 en 24 bits. Con esta función se analiza un **PCG** (24 bits) designado en la entrada de la función:



```

function [posicion_pcb_sig
matriz]=pcb(entrada,sec_larga,salida,PCG)
x=24*PCG-23;
y=24*PCG;
a=entrada(x:y); %datos de entrada de Scrambler en grupos de 24
bits
b=sec_larga(x:y); %secuencia larga decimada en grupos de 24 bits
c=xor(a,b); %multiplicación de entre datos de entrada y secuencia
PN larga
d=bipolar(c)'; %transformación del vector c unipolar, a un vector
bipolar
e=salida(x:y); %datos de salida de Scrambler en grupos de 24 bits
posicion_pcb_sig=binvec2dec(b(21:24)')+1; %conversión a decimal de
últimos 4 bits del grupo de 24 bits de la secuencia PN larga
decimada y obtención de la posición de PCB en siguiente PCG
matriz=[a b c d e]; %matriz generada para comparación entre las
variables de entrada y salida

```

Donde las entradas: **entrada,sec\_larga,salida**, corresponderán a las variables **entrada,sec\_larga,salida** exportadas al **Workspace** desde Simulink. La entrada **PCG** define el grupo de control de potencia (**Power Control Group**, 24 bits) que se desea analizar.

La función produce las salidas **posicion\_pcb\_sig** y **matriz**. La primera salida corresponde a la posición que tendrá el **PCB** en el siguiente **PCG** y que se calcula a partir de los últimos 4 bits del grupo de 24 bits de la secuencia larga decimada. Estos 4 bits se transforman a decimal y se les suma 1 para obtener un valor decimal entre 1 y 16.

La segunda salida corresponde a la matriz, cuyas columnas corresponden a: entrada del bloque (columna **a**), secuencia larga decimada (columna **b**), operación XOR entre la entre **a** y **b** (columna **c**), transformación de unipolar a bipolar de **c** (columna **d**), y los datos de salida del bloque.

### Resultados:

Se realiza el análisis de los primeros 4 **PCGs** (24 bits cada uno).

Se observan los primeros 4 bits de la variable **PCB** en el **Workspace** que corresponden a los 4 bits que serán ingresados, cada uno, en los primeros 4 **PCGs** a analizar:

```
>> PCB(1:4)
ans =
 1
 1
 0
 1
```

Figura 4.116 PCBs generados para ser insertados.

Se realiza el análisis del primer *PCG*:

```
>> [a b]=pcb(entrada,sec_larga,salida,1)
a =
 13
b =
 0 1 1 -1 -1
 0 0 0 1 -1
 1 0 1 -1 -1
 1 1 0 1 1
 1 0 1 -1 -1
 1 0 1 -1 -1
 1 1 0 1 1
 0 1 1 -1 -1
 1 1 0 1 1
 0 0 0 1 1
 1 0 1 -1 -1
 1 0 1 -1 -1
 0 1 1 -1 -1
 1 0 1 -1 -1
 0 0 0 1 1
 1 1 0 1 1
 1 1 0 1 1
 0 0 0 1 1
 0 0 0 1 1
 1 0 1 -1 -1
 1 0 1 -1 -1
 1 1 0 1 1
 1 1 0 1 1
```

Bits para calcular la posición de  $PCB_2$  en  $PCG_2$

Figura 4.117 Análisis del Primer *PCG*.

Como se puede observar el  $PCB_1$  se encuentra en la posición 1 de la columna  $e$  por defecto y se repite en la siguiente posición. La posición del  $PCB_2$  en el  $PCG_2$  se calcula en base a los 4 últimos bits de la columna  $b$  de la matriz y es igual a 13.

Se realiza el análisis del segundo  $PCG$ :

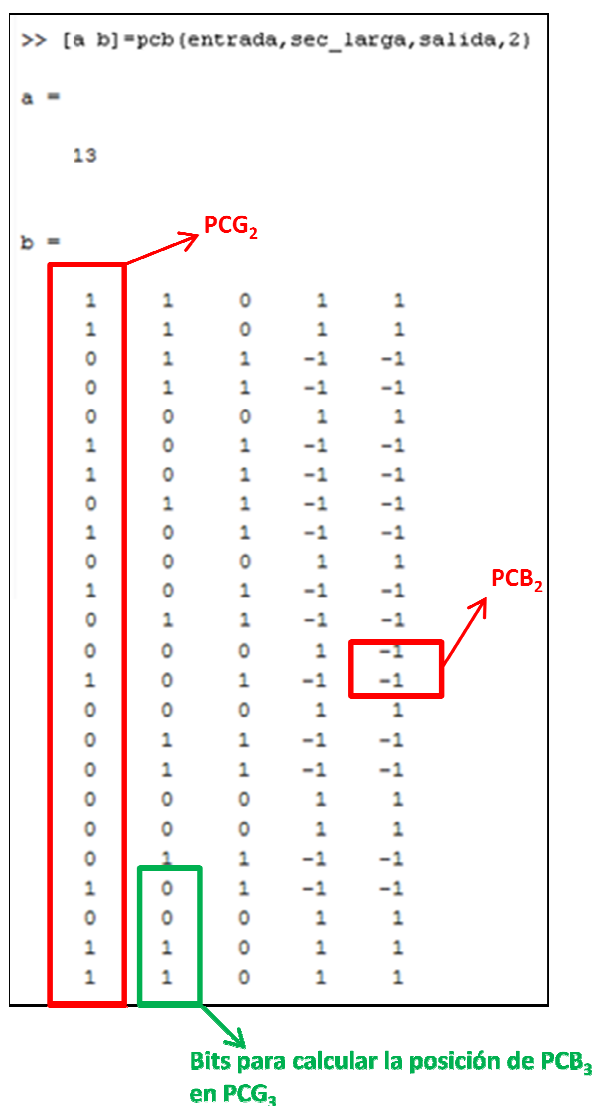
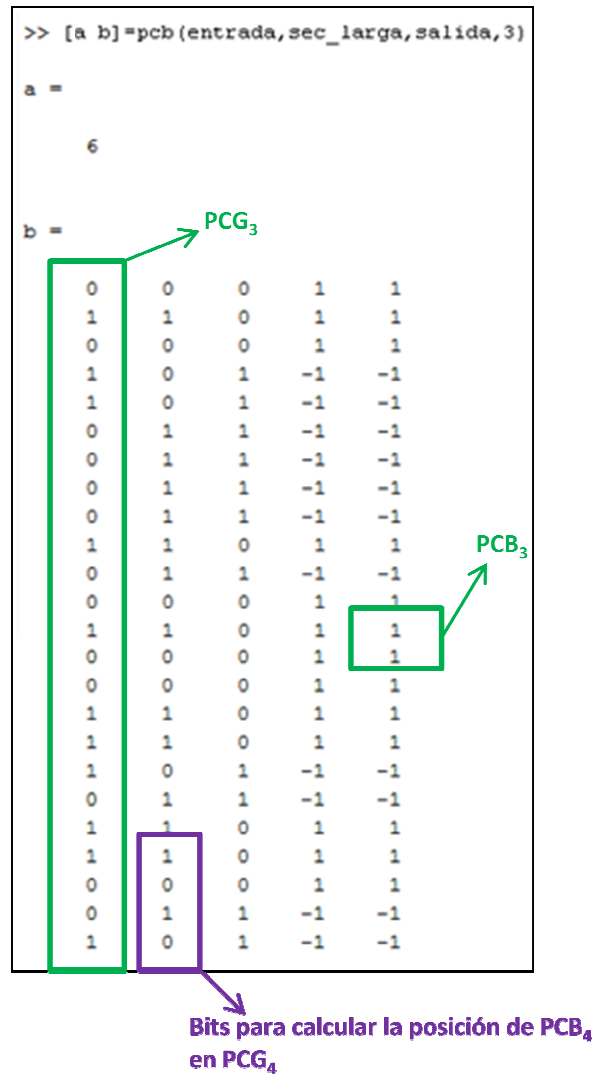


Figura 4.118 Análisis del Segundo  $PCG$ .

Como se puede observar el  $PCB_2$  se encuentra en la posición 13 (calculada anteriormente) de la columna  $e$  y se repite en la siguiente posición. La posición del  $PCB_3$  en el  $PCG_3$  es igual a 13.

Se realiza el análisis del tercer **PCG**:



**Figura 4.119** Análisis del Tercer PCG.

Como se puede observar el **PCB<sub>3</sub>** se encuentra en la posición 13 (calculada anteriormente) de la columna **e** y se repite en la siguiente posición. La posición del **PCB<sub>4</sub>** en el **PCG<sub>4</sub>** es igual a 6.

Se realiza el análisis del cuarto **PCG**:

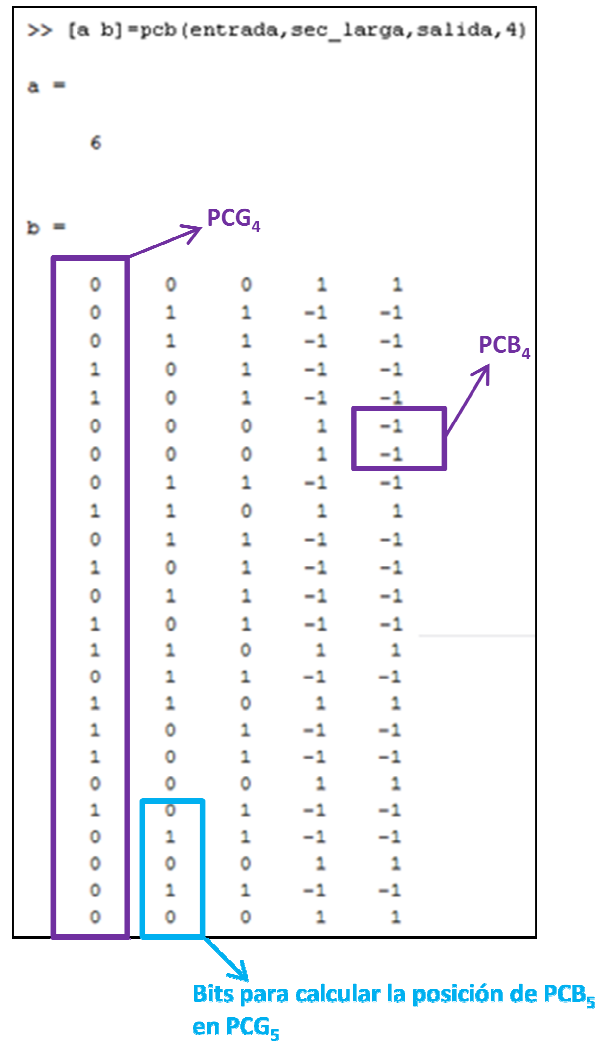


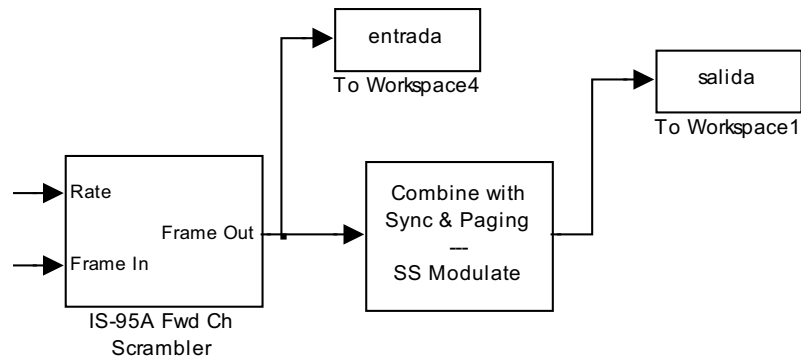
Figura 4.120 Análisis del Cuarto PCG.

Como se puede observar el  $PCB_4$  se encuentra en la posición 6 (calculada anteriormente) de la columna  $e$  y se repite en la siguiente posición. La posición del  $PCB_5$  en el  $PCG_5$  es igual a 6.

Se puede comprobar que los  $PCBs$  generados con el bloque *Random Power Bits* han ocupado los lugares correspondiente dentro los  $PCGs$ . En el primer  $PCG$ , por defecto, el  $PCB$  ocupa la posición 2.

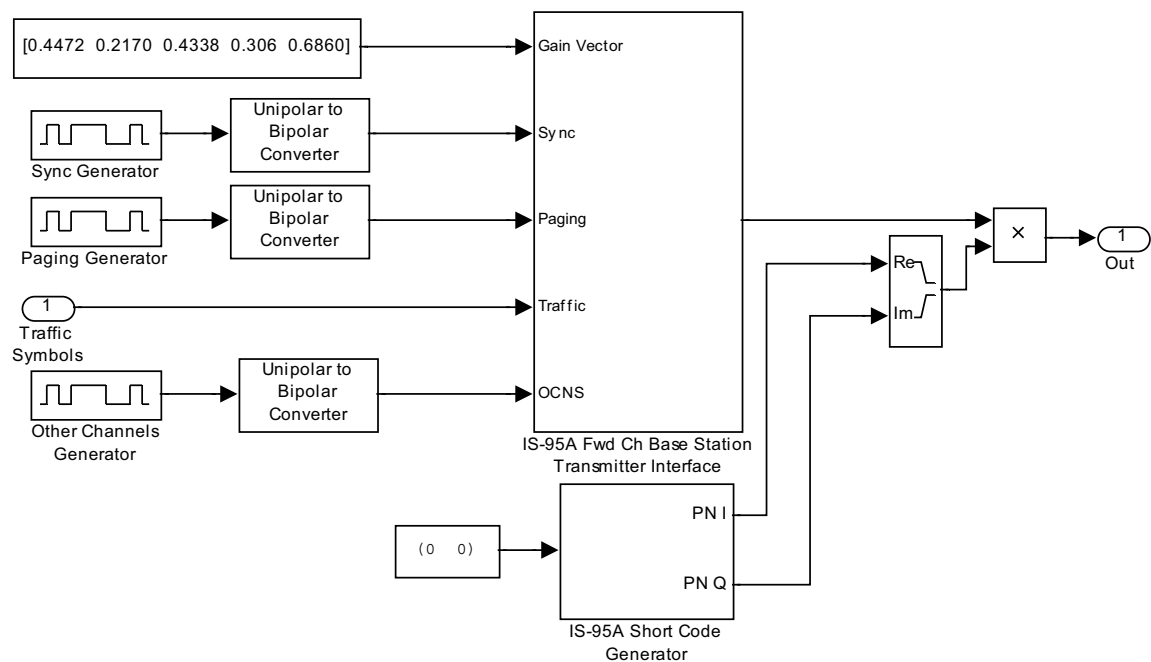
### Combinación con los canales Sync, Paging, Piloto

Este subsistema combina los canales del enlace Forward para formar la señal DSSS y transmitirla en cuadratura.



**Figura 4.121** Subsistema para realizar la combinación con los canales Piloto, Sync y Paging.

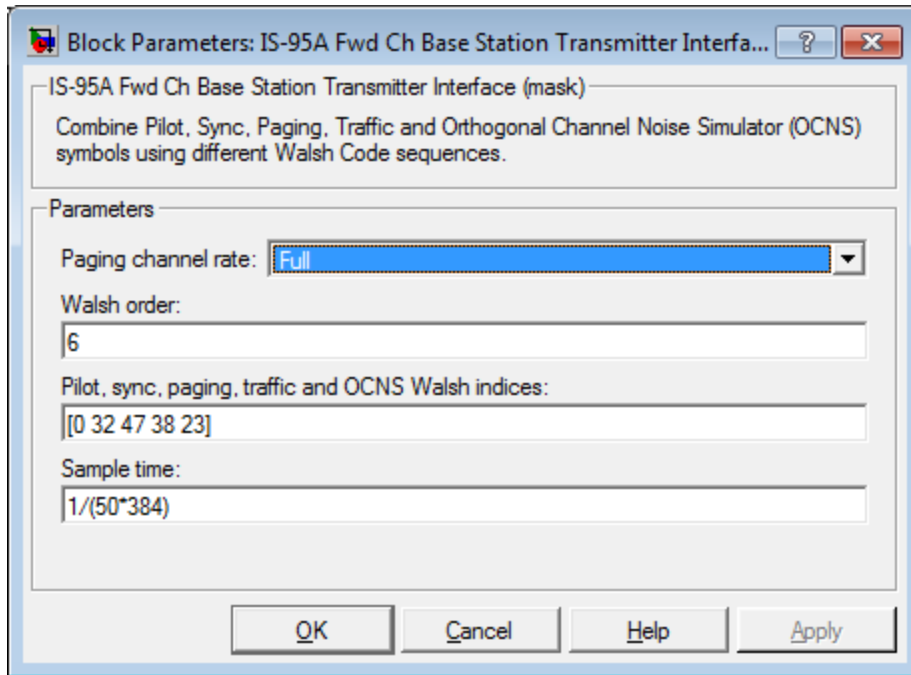
Expandiendo el subsistema se tiene:



**Figura 4.122** Expansión del Subsistema Combinador de canales y Modulación de Espectro Ensanchado.

Para reproducir los canales Sync, Paging se utilizan generadores aleatorios de datos con las tasas correspondientes a cada canal. Además se implementa un generador de ruido ortogonal **OCNS** para simular a un usuario interferente.

El bloque que realiza el ensanchamiento de los canales por el código Walsh respectivo es el **IS-95A Fwd Ch Base Station Transmitter Interface**. En el cual se deben ingresar los datos de los diferentes canales, así como la ganancia correspondiente a cada uno de ellos.

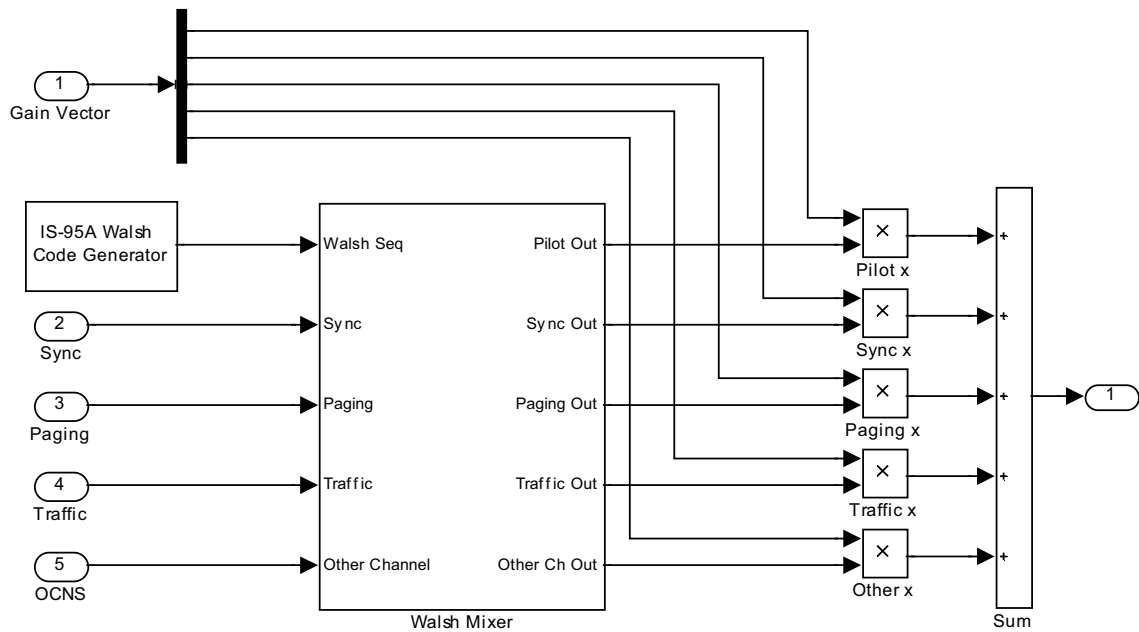


**Figura 4.123** Configuración del bloque IS-95A Fwd Ch Base Station Transmitter Interface.

Se puede ver, que el canal paging va a tener una tasa de datos **full** (9600 bps). También se configura que se utilizan códigos Walsh de longitud  $2^6=64$ . Los códigos Walsh asignados a los canales piloto, Sync, paging, tráfico, y ocns son 0, 32, 47, 38, 23 respectivamente. La tasa de salida se define como 1/19200 segundos, considerando tramas de 64 bits. Por lo tanto se define una tasa de:

$$V_{tx} = \frac{64 \text{ bits}}{\frac{1}{19200} \text{ segundos}} = 1,2288 \text{ Mcps}$$

Expandiendo el bloque **IS-95A Fwd Ch Base Station Transmitter Interface** se tiene:



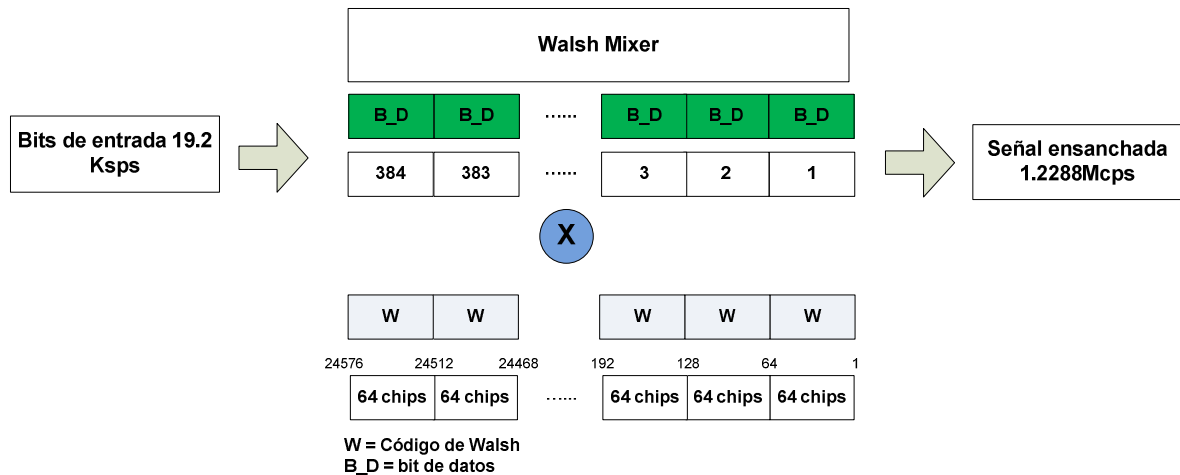
**Figura 4.124** Expansión del bloque IS-95A Fwd Ch Base Station Transmitter Interface.

Se puede observar que existe un subsistema llamado **Walsh Mixer**, en el cual ingresan los datos de los canales con el respectivo código de Walsh.

Los códigos Walsh se forman con el bloque **IS-95A Walsh Generator** configurado de forma que genere 4 códigos a una tasa de transmisión de 1.2288 Mcps concatenados para realizar el ensanchamiento de cada canal dentro del subsistema **Walsh Mixer**.

La siguiente figura muestra el proceso de ensanchamiento de los datos del canal de tráfico por el código de Walsh respectivo. En este caso cada bit de datos de tráfico es multiplicado por un código de Walsh ( **$G_p = 64$** ).

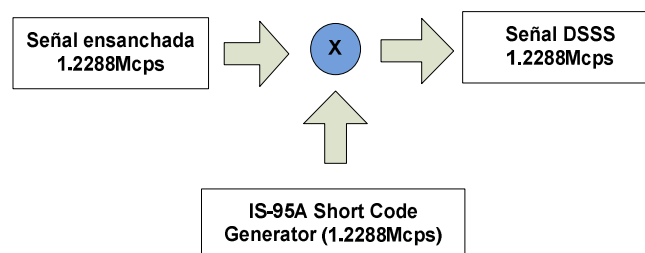




**Figura 4.125** Proceso de ensanchamiento de los datos del canal de tráfico por el código de Walsh respectivo

Luego de que los datos de cada canal son ensanchados, se multiplican por su respectiva ganancia, y son sumados. El vector resultante es la salida del subsistema **IS-95A Fwd Ch Base Station Transmitter Interface**.

Seguidamente, la salida del **IS-95A Fwd Ch Base Station Transmitter Interface** se la multiplica por las secuencias cortas PNI y PNQ, definidas en el estándar de **IS-95A**, para formar la señal DSSS del enlace Forward a 1.2288 Mcps y poder realizar la Modulación QPSK.



**Figura 4.126** Multiplicación por la secuencia PN corta.

Como resultado de este bloque, a la salida se generan tramas de 64 elementos.

Hay que recalcar que el demo no realiza la modulación ni demodulación QPSK. En lugar de la modulación QPSK se combinan los canales I y Q, a través del bloque **Real-Imag to Complex**, formando una señal compleja que simulará la modulación en cuadratura.

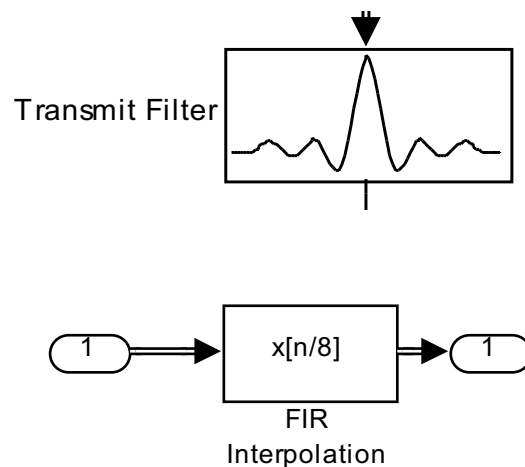
### Filtrado de la Señal

Para el filtrado de la señal, se utiliza un filtro FIR. Este es un filtro pasabajos con frecuencia de corte:

$$W_c = W_s/2$$

Siendo  $W_s$  la frecuencia de muestreo.

Se define un factor de interpolación que consiste en aumentar la frecuencia de muestreo, obteniendo muestras de mayor frecuencia a partir de datos muestreados a menor frecuencia. En el demo se utiliza **un índice de interpolación igual a 8**. Esto quiere decir que por cada símbolo de entrada se obtienen 8 símbolos. Entonces, después del filtro, se obtienen  $8 \times 64 = 512$  muestras.



**Figura 4.127** Filtro de Transmisor

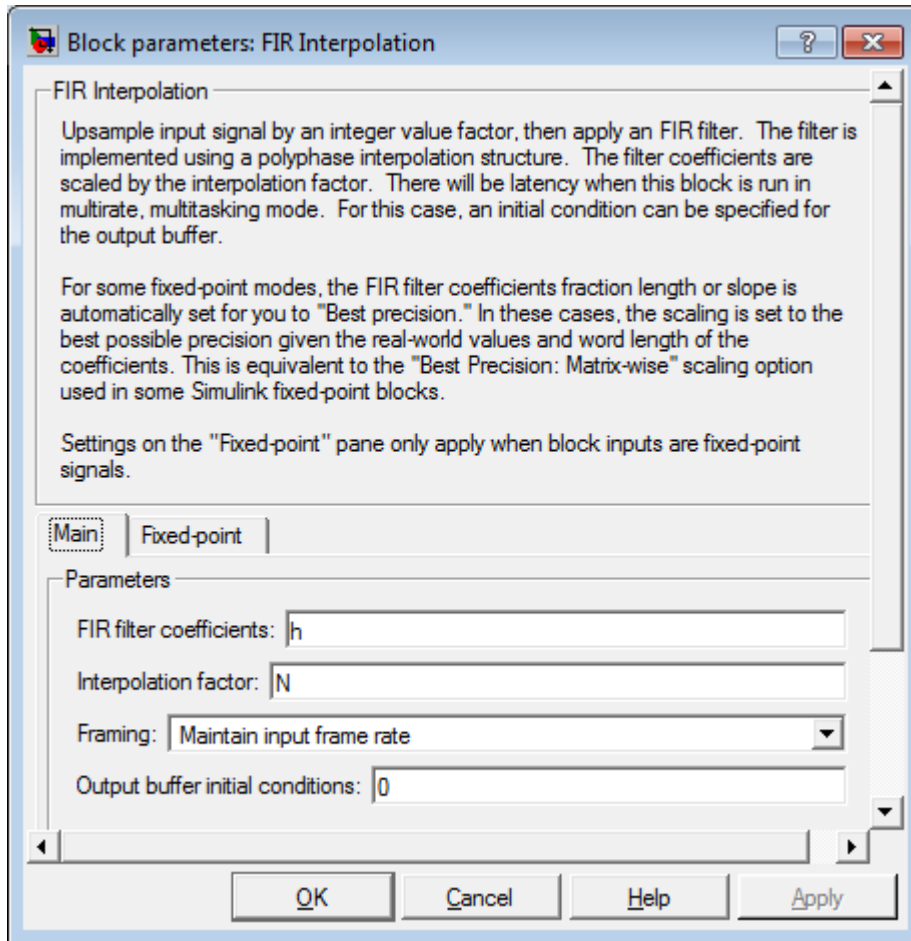
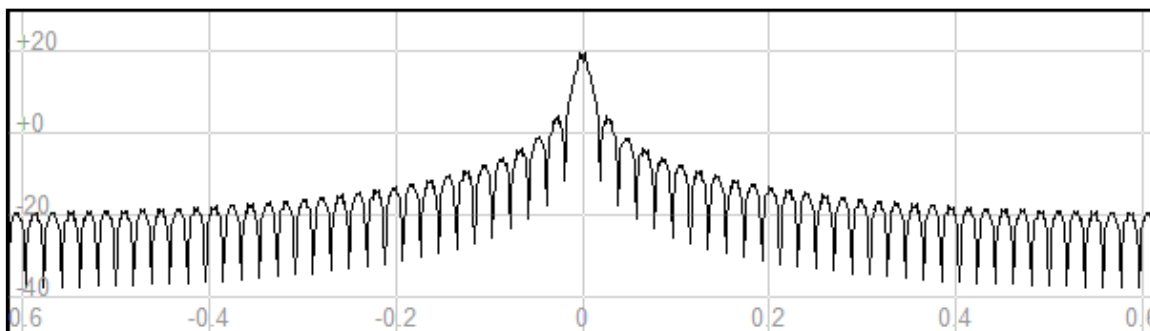
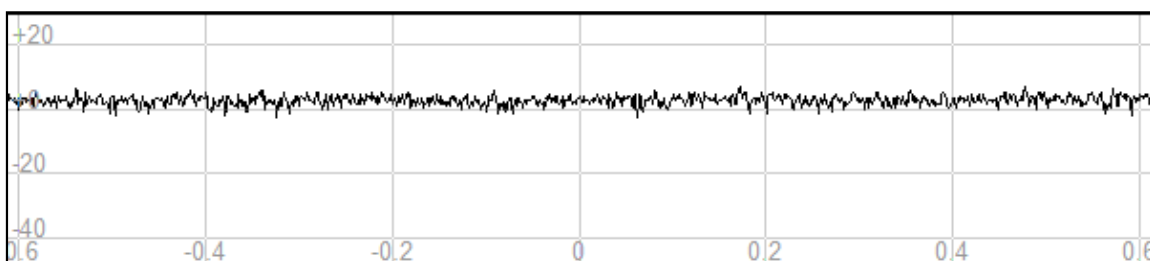
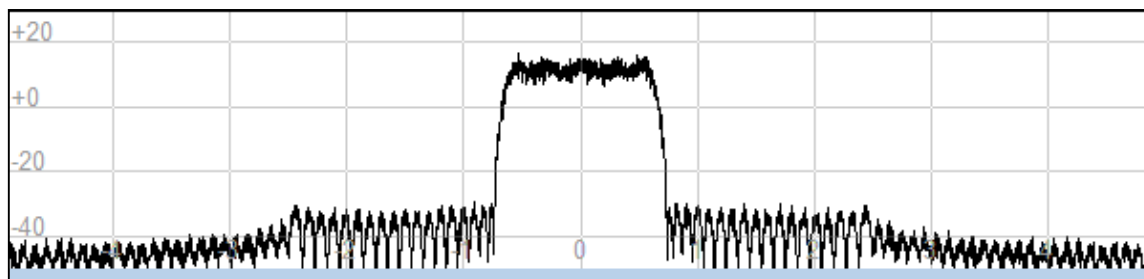


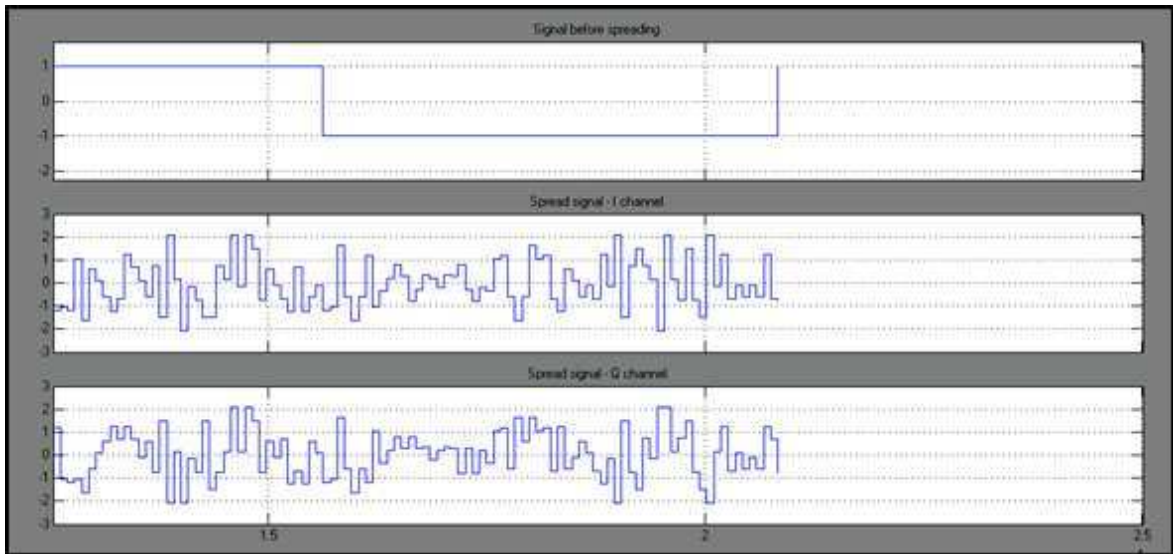
Figura 4.128 Parámetros del Bloque de Interpolación FIR.

### Visualización de las Señales del Transmisor

Para la visualización de las señales, se utilizan los bloques **B-FFT** junto al bloque **Zero-Order Hold** para obtener los espectros de las señales. Para la visualización de las señales en el dominio del tiempo, se utiliza el bloque **Scope** junto con los bloques **Unbuffer** para transformar las tramas a muestras. También se utiliza el bloque **Complex to Real-Imag** para descomponer la señal compleja DSSS en dos señales y puedan ser visualizadas en el **Scope**.

**Resultados:****Figura 4.129** Señal antes de ser ensanchada**Figura 4.130** Señal DSSS**Figura 4.131** Señal DSSS filtrada

Como se puede observar, la señal DSSS toma una apariencia de ruido. Para que la señal sea enviada por un medio inalámbrico con un ancho de banda limitado, se necesita primero pasarla por un filtro. La señal filtrada ocupa mayor ancho de banda que la señal en banda base (antes de ser ensanchada).

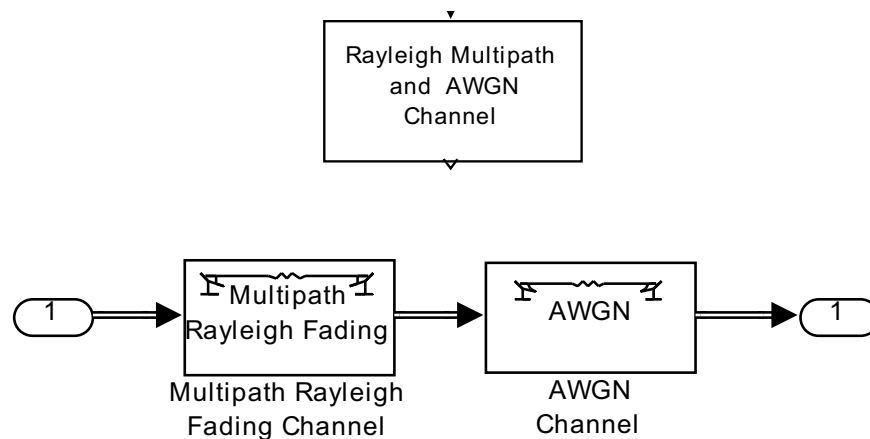


**Figura 4.132** Señales del transmisor en el dominio del tiempo.

Se puede observar que las señales del canal I y Q son la combinación de los canales del enlace Forward, pues posee varias amplitudes en un tiempo de chip y la máxima amplitud es el valor absoluto de 2 (potencia total del enlace).

### Canal

Para simular un canal inalámbrico, se implementa dentro del subsistema Canal multicanal Rayleigh y AWGN.



**Figura 4.133** Simulación de un canal inalámbrico.

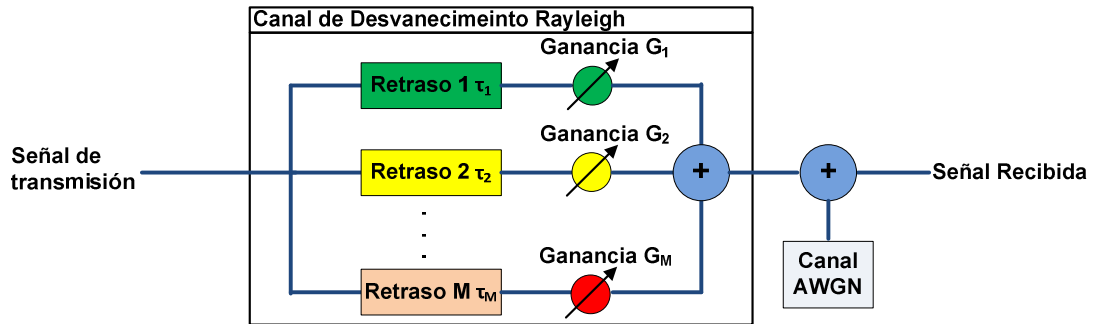


Figura 4.134 Estructura de un canal inalámbrico.

Para la simulación del canal, en el demo se ha utilizado tanto canal multiplicativo, como aditivo.

Para el canal multiplicativo se ha implementado el bloque **Multipath Rayleigh Fading**:

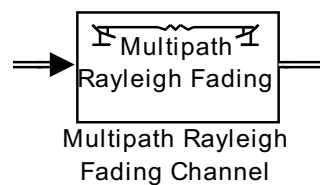
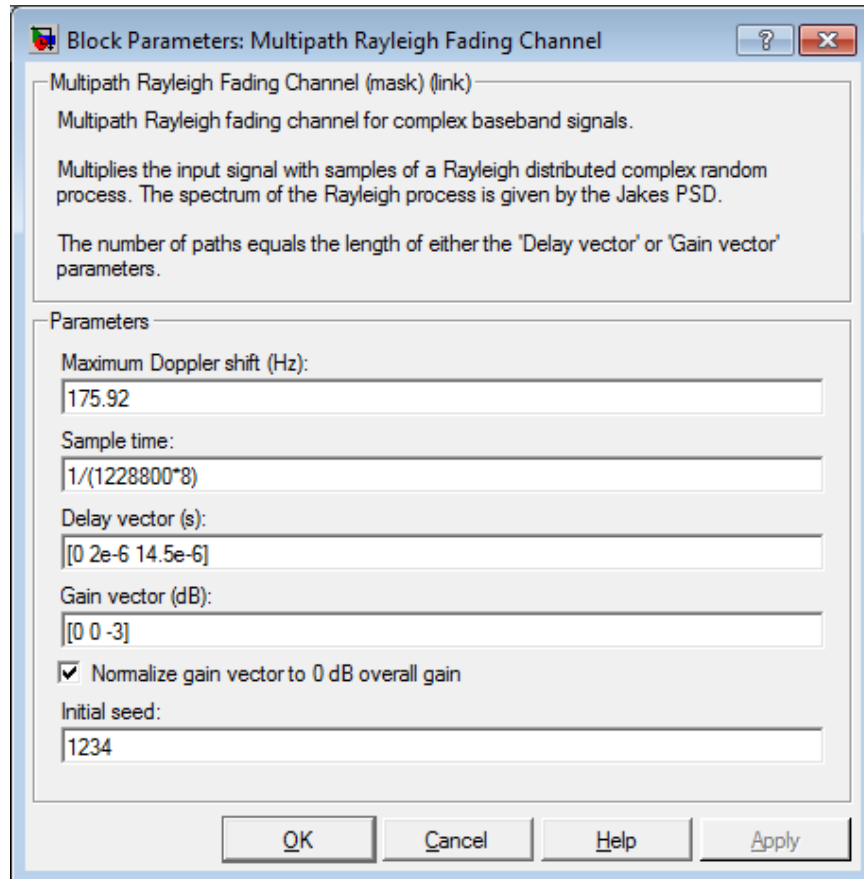


Figura 4.135 Bloque de Desvanecimiento por Multitrayectoria Rayleigh.

Este bloque multiplica la señal de entrada con muestras un proceso aleatorio complejo con distribución Rayleigh.



**Figura 4.136** Parámetros del Bloque Multipath Rayleigh Fading Channel.

Se ingresa un desplazamiento de Doppler Máximo de frecuencia de 175.92 Hz. Se considera el tiempo de muestreo de  $1/1.2288 \times 8$  segundos debido a la tasa de transmisión de 1.2288 Mcps por el factor de interpolación igual a 8. Con este bloque se generan replicas de la señal de transmisión, retrasadas y con cierta ganancia. Estas replicas son sumadas para simular un camino multitrayectoria.

En el parámetro **Delay vector(s)** se define cuantas replicas se producirán y el retardo de cada una. En el demo se especifican tres replicas: una con retraso igual a cero segundos, otra con retraso igual a  $2 \times 10^{-6}$  segundos y otra con retraso igual a  $5 \times 10^{-6}$  segundos.

Para realizar la conversión de los retardos a símbolos, se multiplica el retardo por la tasa de transmisión de 1.2288 Mcps y por el factor de interpolación del filtro **FIR** igual a 8. Esto es igual a:

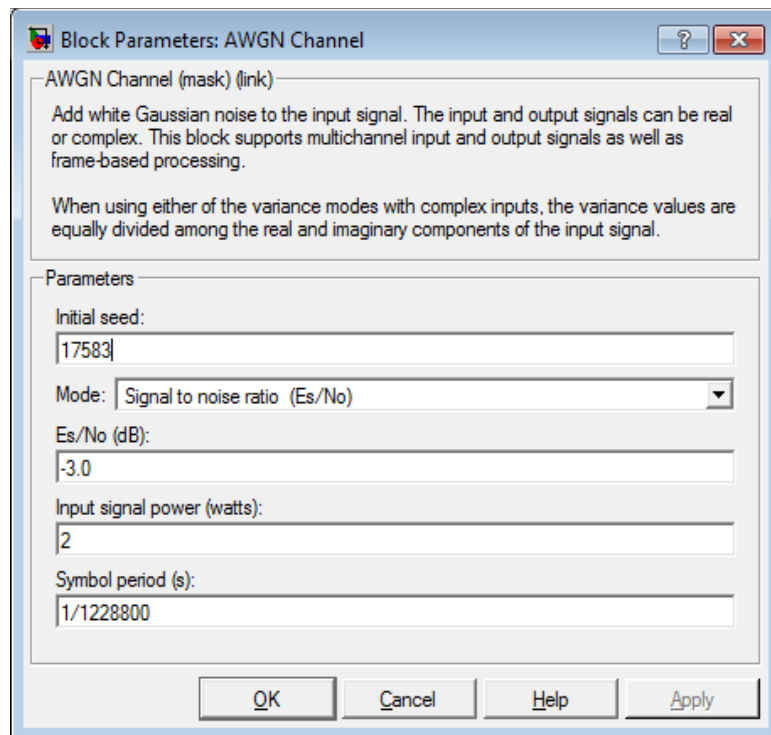
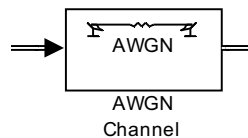
$$\text{Retardo de 0 segundos} = 0 \times 1.2288 \times 10^6 \times 8 = 0 \text{ símbolos de retardo}$$

*Retardo de 2  $\mu$ segundos =  $2 \times 10^{-6} \times 1.2288 \times 10^6 \times 8 = 20$  símbolos de retardo*

$$\begin{aligned} \text{Retardo de 14.5 } \mu\text{segundos} &= 14.5 \times 10^{-6} \times 1.2288 \times 10^6 \times 8 \\ &= 143 \text{ símbolos de retardo} \end{aligned}$$

Con el parámetro **Gain vector(dB)** se definen las ganancias de las réplicas. En el demo se especifican las ganancias para las tres réplicas: 0 dB para la primera, 0 dB para la segunda y -3 dB para la tercera.

Para el canal aditivo se ha implementado ruido Gaussiano con el bloque **AWGN Channel**:



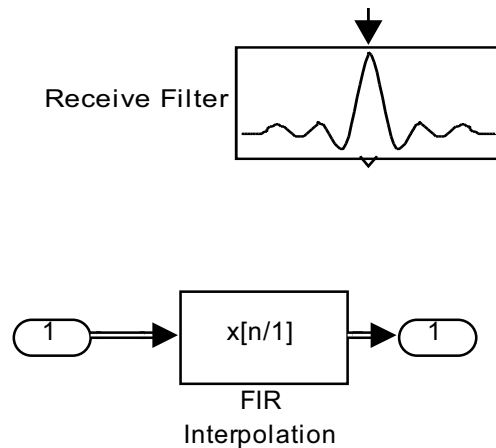
**Figura 4.137** Parámetros del bloque AWGN channel.

Este bloque añade ruido blanco gaussiano a la señal de entrada.



Se puede observar que la relación **Energía de la señal/ Ruido** es negativa, y por lo tanto, la potencia del ruido es mayor a la señal DSSS filtrada. Se considera una potencia total de transmisión de **2 Vatios**. Se especifica que el periodo de cada símbolo modulado es  $1/1.228800$  segundos.

### Filtrado de Recepción



**Figura 4.138** Filtro Receptor.

En la recepción se utiliza el mismo filtro FIR implementado en el transmisor para recuperar la señal del Canal, con la diferencia de que tiene un factor de Interpolación igual a 1.

Cada filtro **FIR** ingresa un retardo 48 símbolos. Por lo tanto se genera un retardo por filtros igual a 96 símbolos. (5)

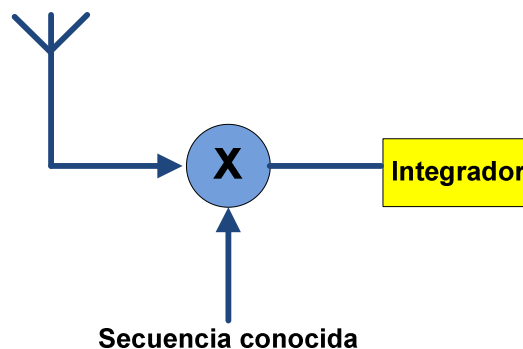
### Detección de la Señal

Para comprender la detección, primero se explican los aspectos básicos del receptor Rake:

El receptor Rake consiste de múltiples correladores, en los que la señal recibida es multiplicada por versiones desplazadas en el tiempo de una secuencia generada localmente. Con esto se intenta separar señales, tal que cada dedo solo

vea señales que vienen de un solo camino (resoluble). El código ensanchador es escogido para tener una muy baja autocorrelación para cualquier desplazamiento de tiempo diferente de cero. Esto evita diafonía entre los dedos. En la práctica, la diafonía entre señales de diferentes dedos no se define por la autocorrelación periódica, sino se define por dos correlaciones parciales con contribuciones de 2 bits o símbolos consecutivos.

El receptor Rake es diseñado para detectar de forma óptima una señal DSSS CDMA transmitida sobre un canal multitrayectoria. Este receptor es una extensión del concepto del filtro adaptado.



**Figura 4.139** Receptor de filtro adaptado

En el receptor de filtro adaptado, la señal es correlacionada con una copia de la señal generada localmente. Si la señal está distorsionada por el canal, el receptor debe correlacionar la señal de entrada con una copia de la señal esperada, en vez que una copia de la señal transmitida. El receptor debe estimar los retrasos producidos por el canal y adaptar su copia generada localmente de acuerdo a las estimaciones.

En un canal multitrayectoria, reflexiones retrasadas interfieren con la señal directa. Sin embargo, una señal DSSS CDMA que sufre de dispersión multitrayectoria puede ser detectada por un receptor Rake. Este receptor combina de forma óptima, señales recibidas sobre múltiples trayectorias.

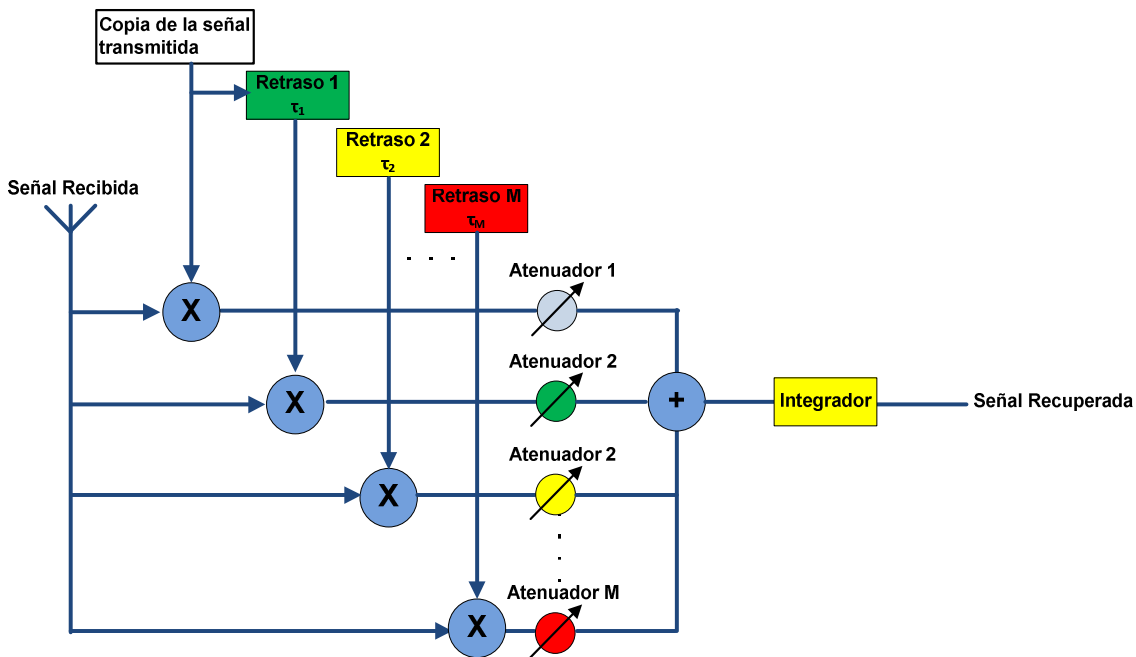


Figura 4.140 Receptor Rake de M-dedos

El receptor Rake reúne la energía recibida sobre los diferentes caminos de propagación con retrasos. El SNR a la salida del receptor es la suma de los SNRs en las ramas individuales.

Para realizar la detección de la señal se utiliza el bloque **IS-95A Fwd Ch Detector**.

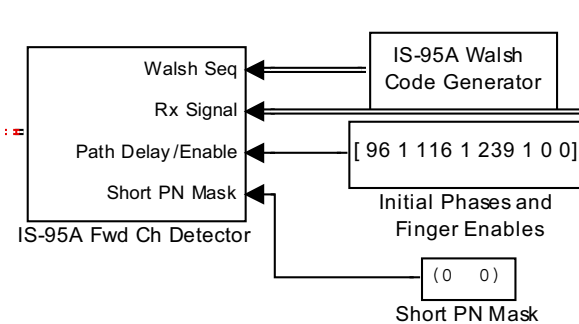
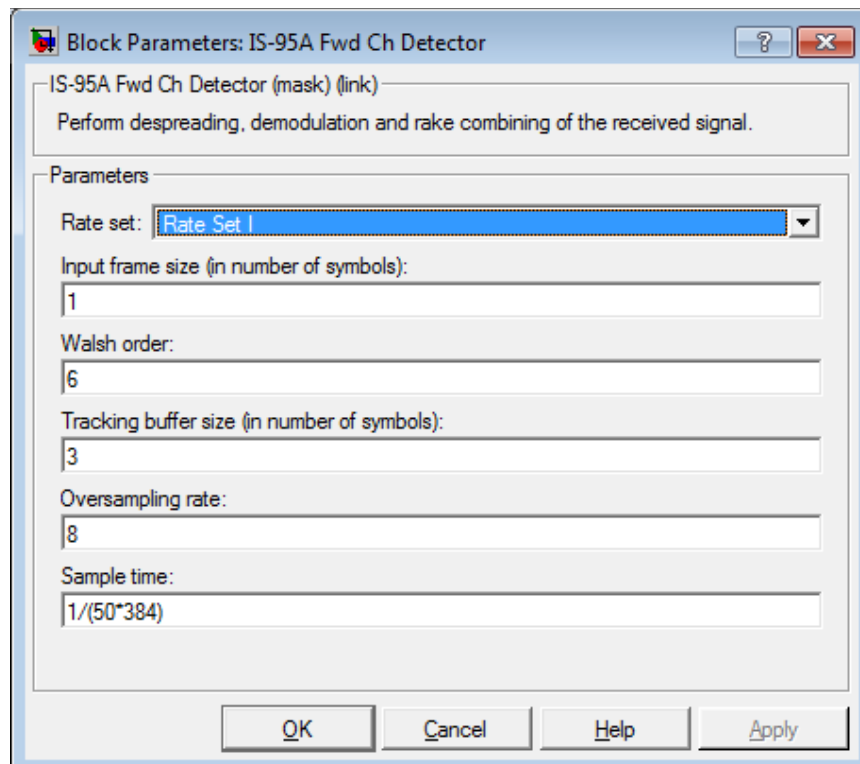


Figura 4.141 Esquema para la detección de la señal a través del IS-95A Fwd Ch Detector.

- Este bloque consiste de 4 dedos Rake, un demodulador Rake coherente, un Descrambler, y generadores de secuencias PN larga y cortas.
- Este bloque realiza las funciones de recepción del enlace Forward para detectar la señal recibida.
- Por cada dedo, este bloque decima la señal recibida, la desensancha con el código de Walsh, la correlaciona con la secuencia PN corta y estima la potencia del canal piloto en la fase asignada al dedo.
- El bloque utiliza las estimaciones de canal para cada dedo, demodula la señal recibida, extrae los bits de potencia (PCB), y combina las señales de todos los dedos Rake para representar los símbolos detectados con decisión suave, cuyo valor es codificado (scrambled) con la secuencia PN larga decimada.
- Finalmente el bloque almacena y saca una trama de los símbolos detectados con decisión suave después de decodificarlos (descrambling) con la secuencia PN larga decimada.

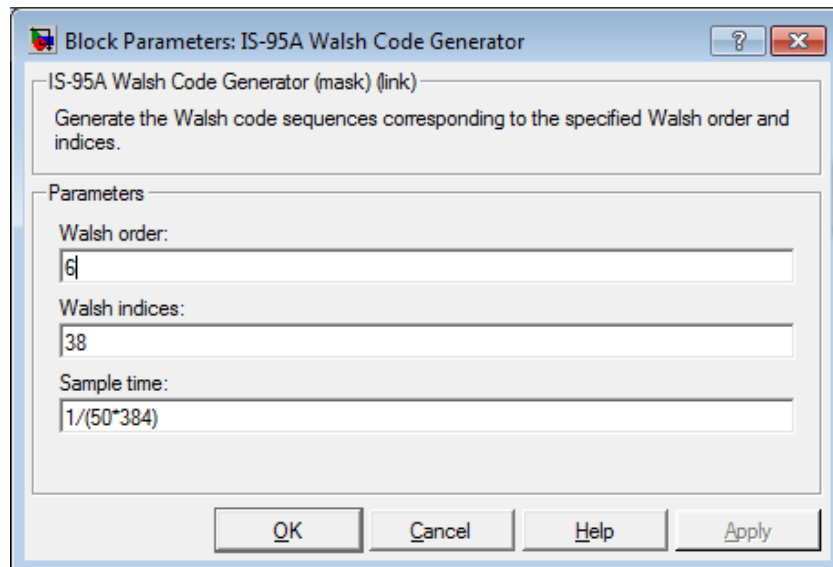


**Figura 4.142** Configuración del Bloque IS-95A Fwd Ch Detector.

Como se puede ver, el receptor Rake procesa la información recibida bit a bit. Se define también la longitud del código de Walsh y su tasa de transmisión, así como el factor de oversampling se fue aplicado en el filtro de salida de la señal DSSS y tracking buffer size que debe ser igual a la cantidad de los dedos Rake para la corrección de errores en los retardos de las trayectorias de los rayos multitrayectoria.

Este bloque necesita como entradas, los bloques ***IS-95A Walsh Code Generator***, ***Initial Phases and Finger Enables*** y ***Short PN Mask***.

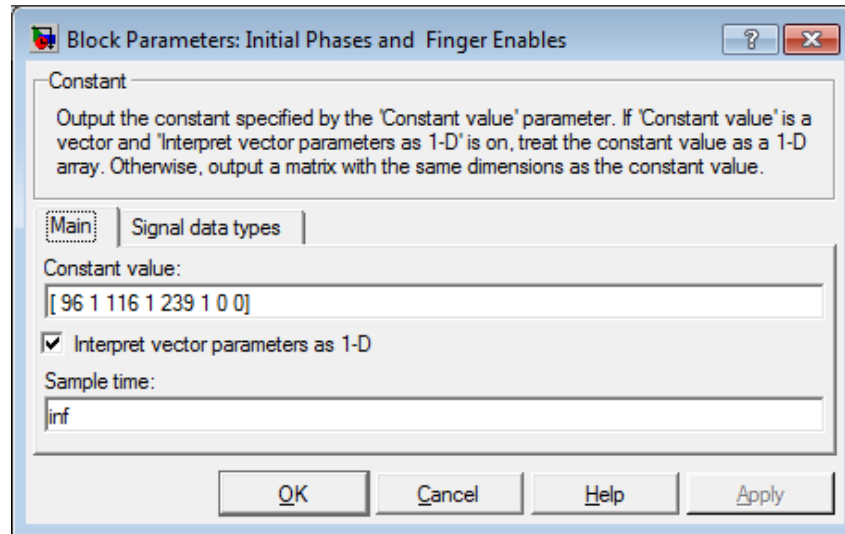
El bloque ***IS-95A Walsh Code Generator*** genera el código de Walsh de orden 6 y con el índice especificado para el canal de tráfico deseado (índice 38).



**Figura 4.143** Configuración del Bloque IS-95A Walsh Code Generator.

El bloque ***Initial Phases and Finger Enables*** es un vector constante de tamaño 8, que contiene un par de elementos para cada dedo del receptor Rake que se encuentra dentro del bloque ***IS-95A Fwd Ch Detector***. El primer elemento en cada par representa el offset PN inicial que es aplicado a cada dedo.

El segundo elemento en cada par sirve como señal de enable para el dedo Rake. Un valor de 0 deshabilita el dedo y un valor de 1 lo habilita. Para la generación de este vector se utiliza el bloque ***Constant***:



**Figura 4.144** Configuración del Bloque Initial Phases and Fingers Enables.

Como se define en el estándar, en el enlace Forward, el receptor Rake tiene habilitado solo 3 dedos. Por esto, el segundo elemento del cuarto par tiene un valor de cero. El offset PN inicial es igual al retardo introducido por el canal multitrayectoria más el retardo introducido por los filtros de transmisión y recepción en símbolos.

$$\text{Offset dedo} = \text{retardo filtros} + \text{retardo canal multitrayectoria}$$

El offset para el primer dedo se calcula:

$$\text{Offset primer dedo} = 96 \text{ símbolos} + 0 \text{ símbolos} = 96 \text{ símbolos}$$

El offset para el segundo dedo se calcula:

$$\text{Offset segundo dedo} = 96 \text{ símbolos} + 20 \text{ símbolos} = 116 \text{ símbolos}$$

El offset para el tercer dedo se calcula:

$$\text{Offset tercer dedo} = 96 \text{ símbolos} + 143 \text{ símbolos} = 239 \text{ símbolos}$$

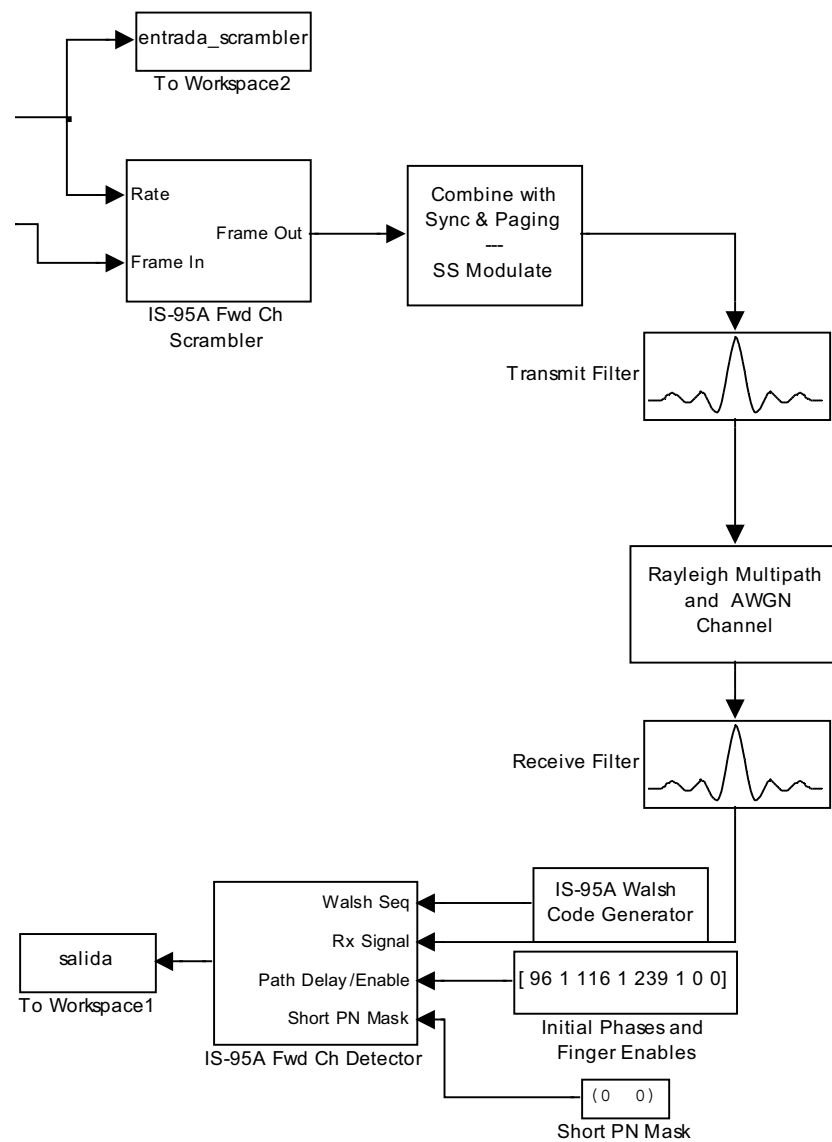
El bloque **Short PN Mask** es un vector constante de tamaño 2 que representa a las máscaras de las componentes de cuadratura y fase de la secuencia PN corta que sirven para especificar con que fase se va a generar. En este caso se utiliza un offset igual a 0.

La decisión suave (soft-decision) en decodificadores es una clase de algoritmo utilizada para decodificar datos. Un decodificador de decisión dura (hard-decision) opera en datos que toman un conjunto fijo de posibles valores (generalmente 0 o 1 en un código binario) mientras que las entradas de un decodificador de decisión suave pueden tomar un rango de valores intermedios.

Esta información extra indica la confiabilidad de cada dato de entrada y se utiliza para formar mejores estimaciones de los datos originales.

El bloque **IS-95A Detector** actúa como un decodificador de decisión suave y entrega valores estimados de la señal transmitida. Estos valores están en el rango de  $]1, 1[$ . Un valor entre  $] -1, 0[$  corresponde a un 1 lógico (-1 bipolar), mientras que un valor entre  $[0, 1[$  corresponde a 0 lógico (1 bipolar). Más adelante se toma la decisión para formar una señal unipolar.

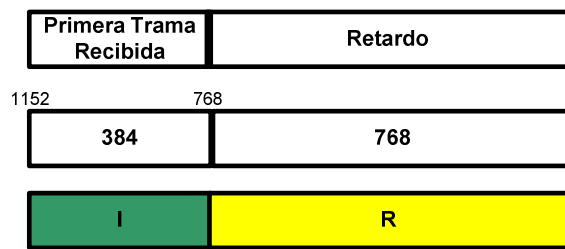
Cuando se realiza la extracción de los PCBs, estos bits se ponen a cero en la señal de salida del bloque. Para comprobar el funcionamiento de este bloque se realiza la comparación de la salida del mismo, con la entrada del bloque **IS-95A Fwd Ch Scrambler**. Se almacenan estas salidas en las variables **salida** y **entrada\_scrambler** respectivamente.



**Figura 4.145** Comprobación del funcionamiento del bloque IS-95A Fwd Ch Detector.

El bloque genera un retardo de 768 elementos correspondientes a dos tramas. Por lo tanto se debe analizar los últimos 384 de 1152 elementos del vector **salida** obtenidos en un tiempo de simulación de 0.04 segundos:





R = Retardo de 768 elementos  
 I = Primera trama recibida y sin PCBs

**Figura 4.146** Salida del bloque IS-95A Fwd Ch Detector en 0.04s.

```
a=salida(769:1152);
```

Del vector **entrada\_scrambler** se almacenan los primeros 384 elementos generados que entran al bloque:

```
b=entrada_scrambler(1:384);
```

Ahora realizará el análisis de los dos primeros **PCGs** (24 bits) de los vectores. Para el primer **PCG**:

```
c=a(1:24);
d=b(1:24);
comparacion=[c d];
```

donde **c** es el primer **PCG** en la salida del bloque **IS-95A Fwd Ch Detector**, **d** es el primer **PCG** en la entrada del Scrambler, y **comparacion** es la matriz donde se comparan ambos vectores:

comparacion =	
0	0
0	0
-0.4847	1.0000
-0.2097	1.0000
-0.1010	1.0000
-0.2643	1.0000
-0.1919	1.0000
0.1980	0
-0.2297	1.0000
0.2340	0
-0.1500	1.0000
-0.1283	1.0000
0.1386	0
-0.0872	1.0000
-0.0172	1.0000
0.2045	0
-0.0860	1.0000
-0.0914	1.0000
0.0777	0
0.1249	0
-0.2614	1.0000
-0.1583	1.0000
-0.1036	1.0000
-0.1845	1.0000

→ Posición PCB<sub>1</sub>

**Figura 4.147** Análisis del primer PCG de salida del bloque IS-95A Fwd Ch Detector.

Como se puede observar para este caso, según las consideraciones hechas anteriormente, los datos de ambos vectores son iguales a excepción de las posiciones de los **PCBs** (1 y 2) donde se observa un dos ceros consecutivos en la columna del detector.

Para el segundo **PCG** se tiene:

```
e=a(25:48);
f=b(25:48);
comparacion2=[e f];
```

donde **e** es el segundo **PCG** en la salida del bloque **IS-95A Fwd Ch Detector**, **f** es el segundo **PCG** en la entrada del Scrambler, y **comparacion2** es la matriz donde se comparan ambos vectores:

comparacion2 =	
-0.1551	1.0000
-0.0842	1.0000
0.2515	0
0.1026	0
0.2125	0
-0.0078	1.0000
-0.1473	1.0000
0.0561	0
-0.1112	1.0000
0.0747	0
-0.1452	1.0000
0.0273	0
0	0
0	1.0000
0.1041	0
-0.0367	0
0.0444	0
0.0093	0
0.1014	0
0.1580	0
-0.0303	1.0000
0.0910	0
-0.0694	1.0000
-0.0441	1.0000

→ Posición PCB<sub>2</sub>

→ Error

**Figura 4.148** Análisis del segundo PCG de salida del bloque IS-95A Fwd Ch Detector.

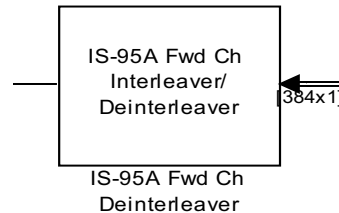
Como se puede observar, en las posiciones 13 y 14 se encuentran dos ceros consecutivos correspondientes a la ubicación que tomaron los **PCBs**. Además, se puede notar que el receptor Rake ha producido errores debido al canal AWGN.

El receptor Rake **no corrige errores** producidos por el canal AWGN, por lo tanto estos errores se generarán de forma aleatoria y serán corregidos en etapas posteriores.

Por la generación del retardo del bloque solo se puede recuperar una trama de transmisión (384 símbolos) en un tiempo de simulación de 0.04 segundos.

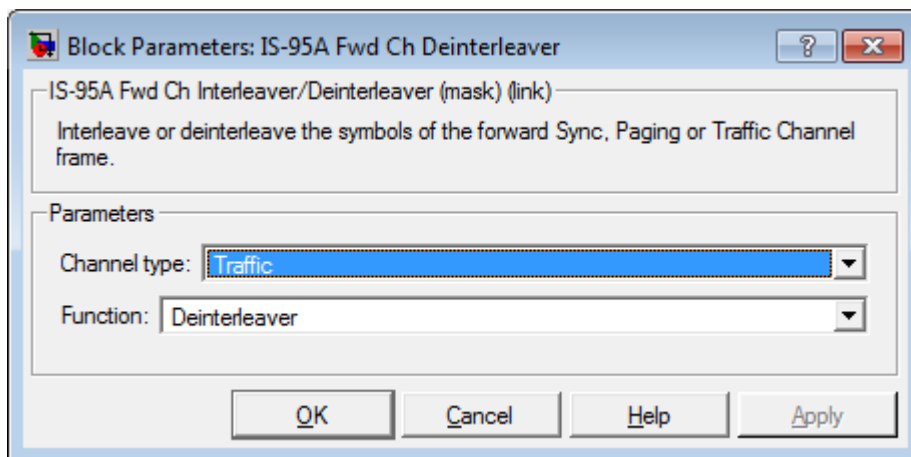
## Deinterleaving

Para realizar la operación de Deinterleaving se utiliza el bloque **IS-95A Fwd Ch Deinterleaver**



**Figura 4.149** Bloque IS-95A Fwd Ch Deinterleaver.

- Este bloque desintercala una trama de información para el canal de Acceso o de Tráfico. El espacio de tiempo de intercalado es 20 ms.
- El desintercalador devuelve los símbolos intercalados a su orden original.
- Solo se diferencia del bloque Interleaver del transmisor en su configuración, en su parámetro **Function: Deinterleaver**.

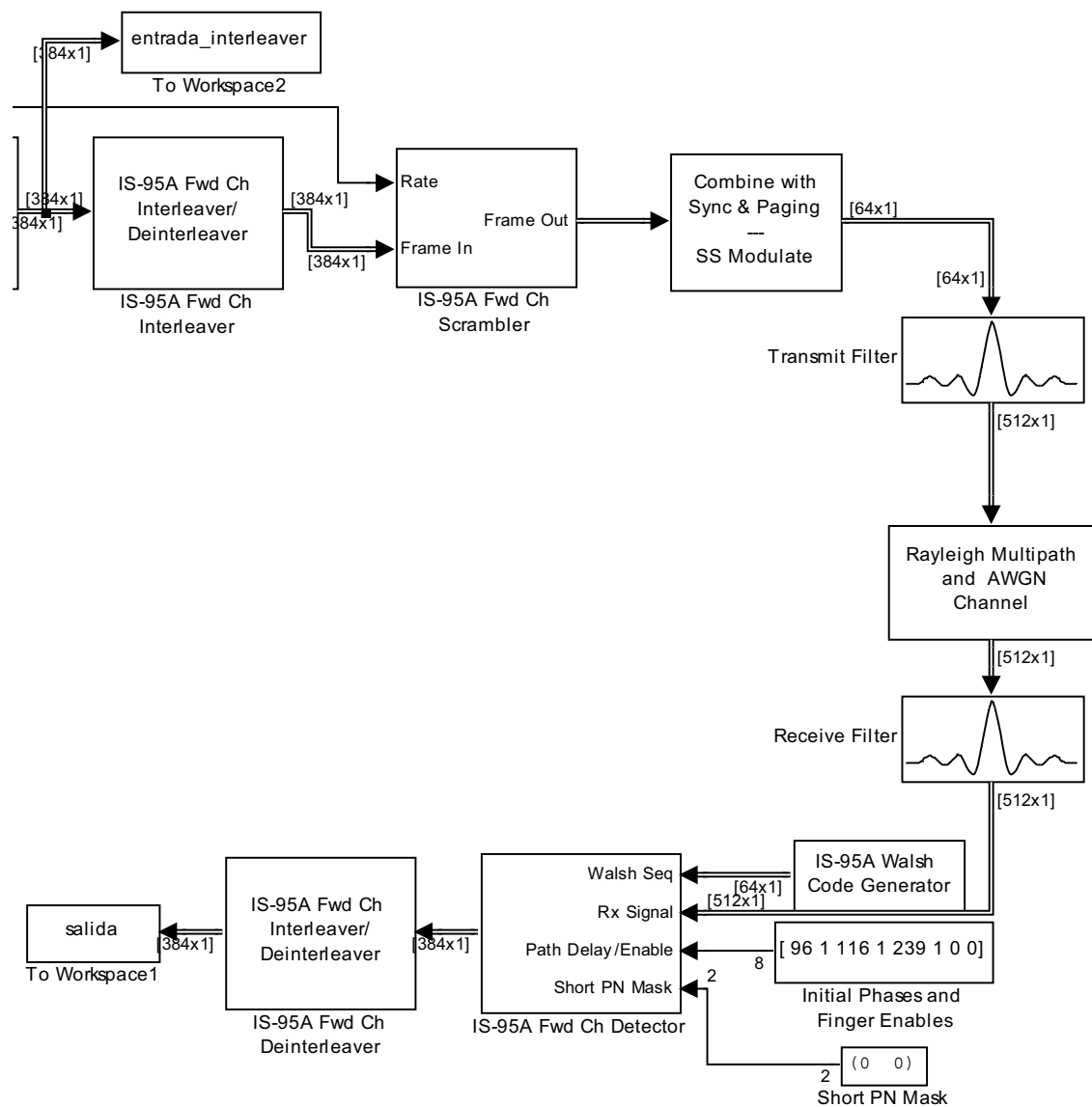


**Figura 4.150** Configuración del Bloque IS-95A Fwd Ch Deinterleaver.

Se puede ver que el bloque se ha configurado para trabajar como Deinterleaver para el canal de Forward de Tráfico.

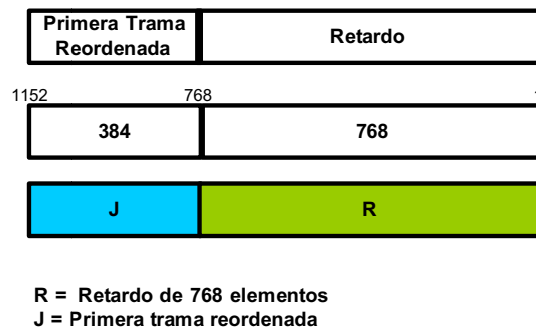
Este bloque realiza el mismo proceso de ingresar los datos de entrada dentro de una matriz equivalente con dimensión 64 X 6 y se realiza el algoritmo del bit inverso por filas.

Para comprobar el funcionamiento de este bloque se realiza la comparación de la salida del mismo, con la entrada del bloque *IS-95A Fwd Ch Interleaver*. Se almacenan estas salidas en las variables *salida* y *entrada\_interleaver* respectivamente.



**Figura 4.151** Comprobación del funcionamiento del Bloque IS-95A Fwd Ch Deinterleaver.

Se almacenan los últimos 384 de 1152 elementos del vector *salida*, puesto que el resto de bits son bits de retardo puestos por el bloque IS-95A Fwd Ch Detector.



**Figura 4.152** Salida del bloque IS-95A Fwd Ch Deinterleaver en 0.04s.

```
a=salida(769:1152);
```

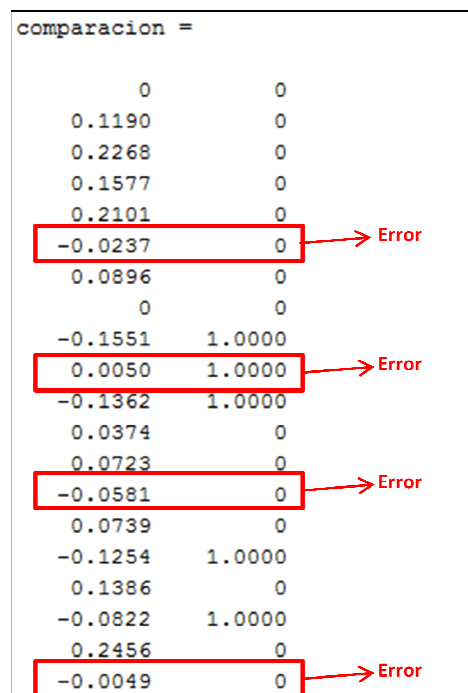
Del vector *entrada\_interleaver* se almacenan los primeros 384 elementos generados que entran al bloque:

```
b=entrada_interleaver(1:384);
```

Se realiza la comparación entre los dos vectores:

```
comparacion=[a b];
```

En el Workspace se tiene lo siguiente:

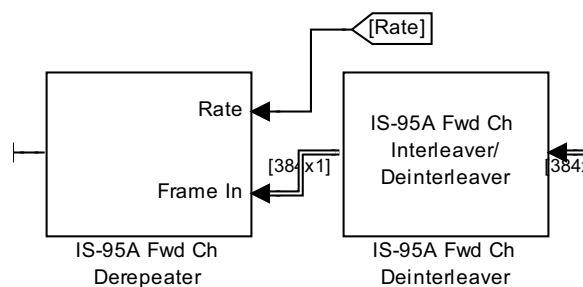


**Figura 4.153** Comparación de la salida del Deinterleaver con los datos originales.

Se puede observar que se cumple la función del Deinterleaver. Además que se existen errores, debido a la presencia del ruido, que han sido reordenados. Esto es lo que se conoce como diversidad temporal, que se la utiliza para prevenir grandes ráfagas de errores.

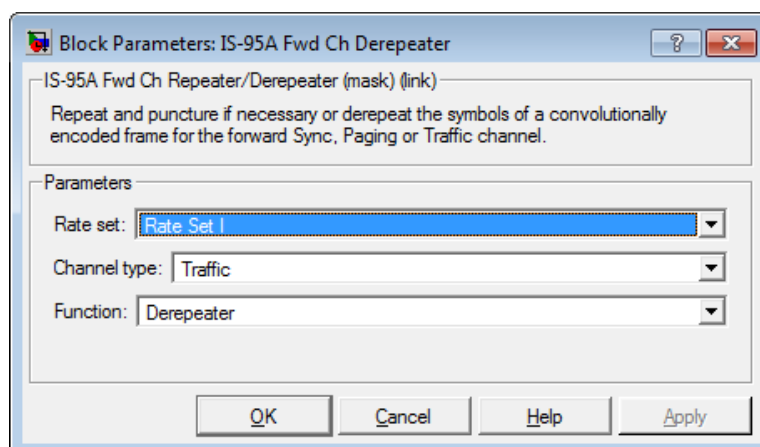
### Eliminación de Símbolos Repetidos

La eliminación de símbolos repetidos se realiza con el bloque **IS.-95A Fwd Ch Derepeater**.



**Figura 4.154** Bloque IS-95A Fwd Ch Derepeater.

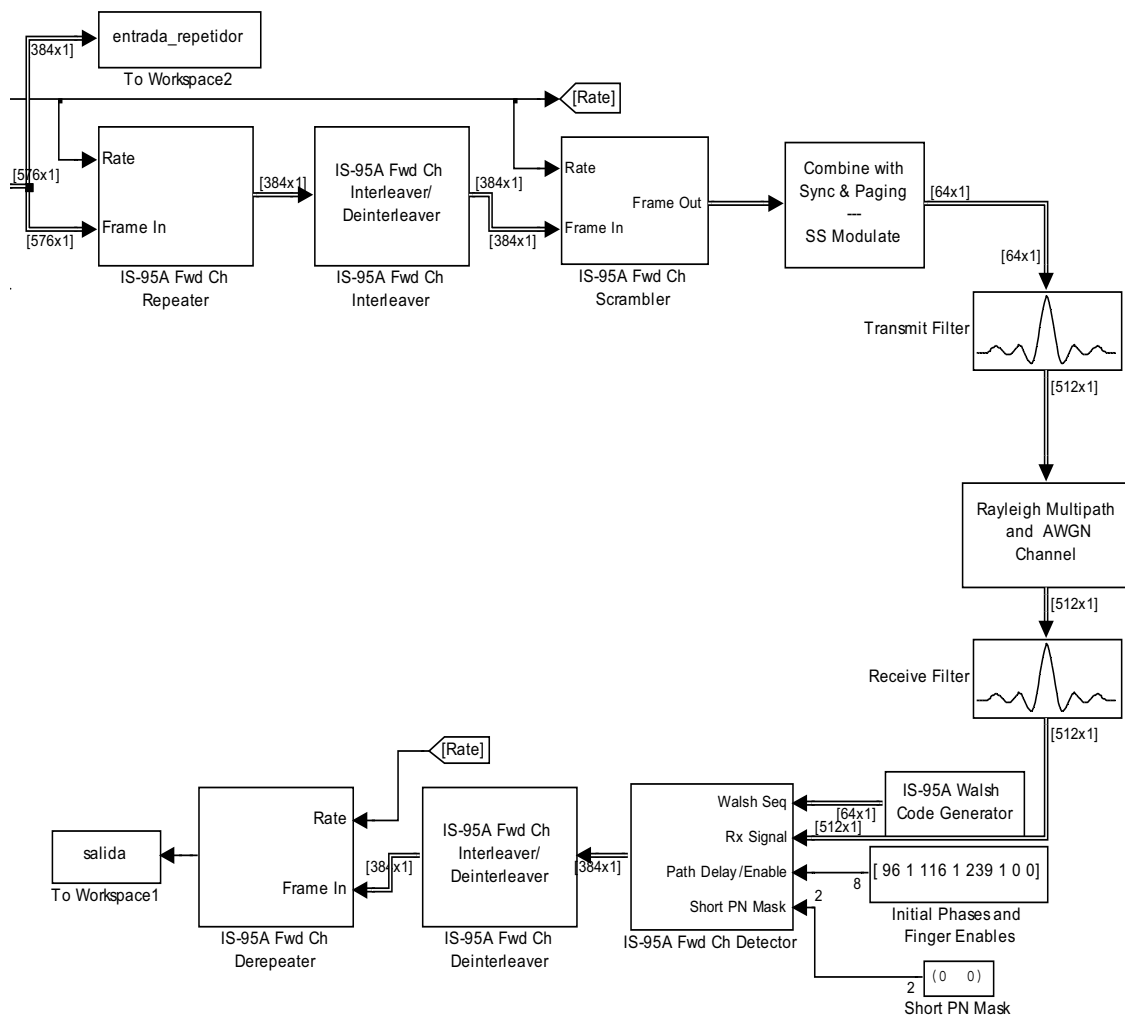
- Para el canal de tráfico Full Rate y Rate Set I no se repiten los bits relevantes, pero se ingresan 192 símbolos iguales a cero para obtener un total de 576 símbolos que ingresarán al decodificador de Viterbi.
- Solo se diferencia del bloque Repeater del transmisor en su configuración, en su parámetro **Function: Derepeater**.



**Figura 4.155** Configuración del Bloque IS-95A Fwd Ch Derepeater.

El bloque se encuentra configurado para realizar la función de Derepeater del Canal Forward de Tráfico del Rate Set I.

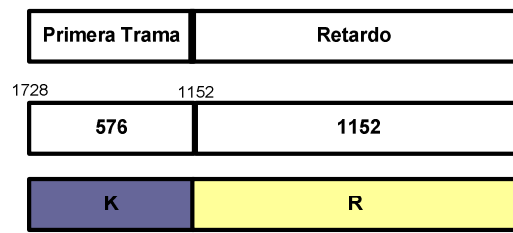
Para comprobar el funcionamiento de este bloque se realiza la comparación de la salida del mismo, con la entrada del bloque *IS-95A Fwd Ch Repeater*. Se almacenan estas salidas en las variables *salida* y *entrada\_repetidor* respectivamente.



**Figura 4.156** Comprobación del funcionamiento del Bloque IS-95A Fwd Ch Derepeater.

Se almacenan los últimos 576 de 1728 elementos del vector *salida* producidos en un tiempo de simulación de 0.04 segundos:





R = Retardo de 1152 elementos  
K = Primera trama con bits de relleno

Figura 4.157 Salida del bloque IS-95A Fwd Ch Deinterleaver en 0.04s.

```
a=salida(1153:1728);
```

Del vector **entrada\_repetidor** se almacenan los primeros 576 elementos generados que entran al bloque:

```
b=entrada_repetidor(1:576);
```

Se realiza la comparación entre los dos vectores:

```
comparacion=[a b];
```

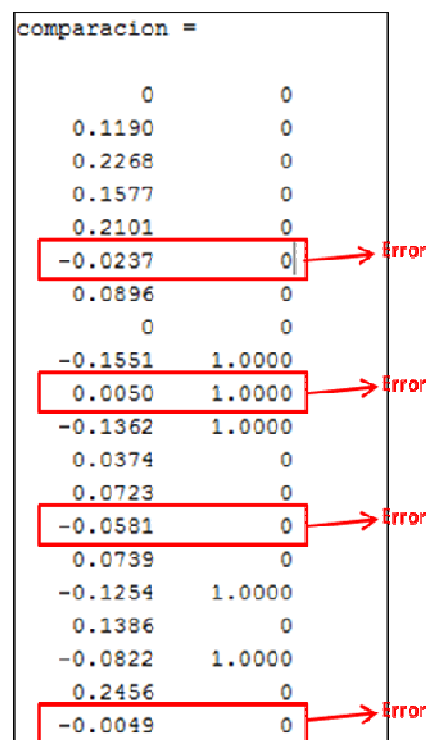


Figura 4.158 Comparación entre la salida del Derepeater con los datos reales.

Como se puede observar, no ha habido ningún cambio con respecto a la salida del Deinterleaver. Para comprobar que los 192 últimos elementos del vector **a** son solo bits de relleno, la suma de los bits debe dar cero:

```
relleno=sum(a(385:576));
```

```
>> relleno=sum(a(385:576))

relleno =

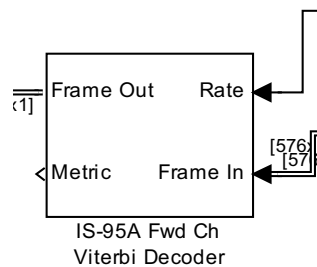
    0
```

**Figura 4.159** Comprobación de los bits de relleno.

Se puede observar que la suma de estos bits, es cero y por tanto solo son de relleno.

### Decodificador de Viterbi

Para realizar la decodificación de los datos codificados convolucionalmente, se utiliza el bloque **IS-95A Fwd Ch Viterbi Decoder**.

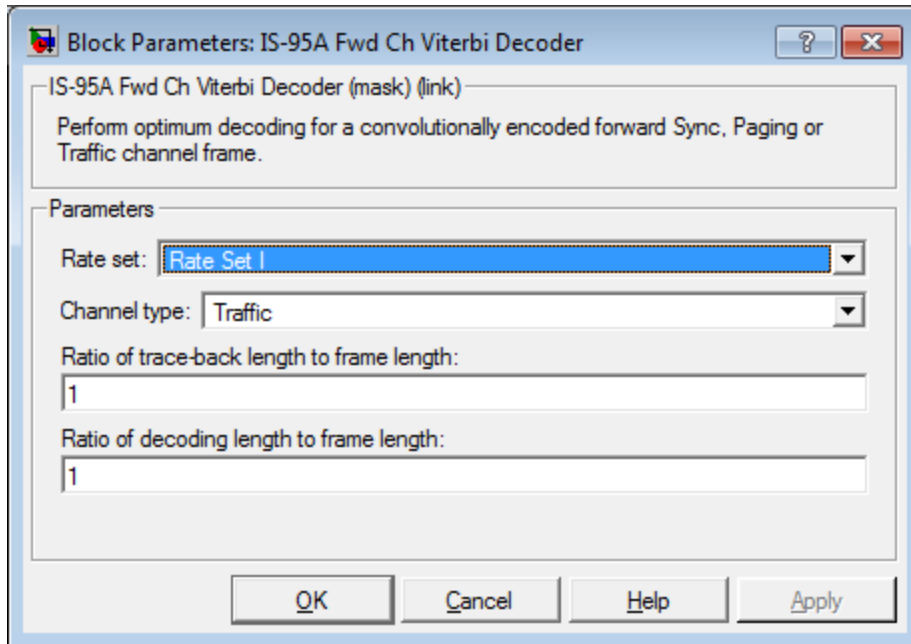


**Figura 4.160** Bloque IS-95A Fwd Ch Viterbi Decoder.

- Este bloque utiliza el algoritmo de Viterbi para decodificar de forma óptima una trama de la información codificada convolucionalmente.
- Para el canal de tráfico el decodificador inicializa en un estado todo cero antes de procesar cada entrada.
- El tamaño del puerto de entrada de la trama es igual al máximo posible de 576 (canal de tráfico Full Rate, Rate Set II). El tamaño del puerto de salida

de la trama es igual al máximo posible de 288 (canal de tráfico Full Rate, Rate Set II).

- El bloque ignora bits de entrada pasados de los bits relevantes y pone a cero a los bits de salida pasados de los bits relevantes.



**Figura 4.161** Configuración del bloque IS-95A Fwd Ch Viterbi Decoder.

Este bloque toma la decisión de datos y convierte los valores entre  $]-1, 0[$  a 1 lógico y los valores entre  $[0, 1[$  a 0 lógico. Para comprobar el funcionamiento de este bloque se realiza la comparación de la salida del mismo, con la entrada del bloque *IS-95A Fwd Ch Convolutional Encoder*. Se almacenan estas salidas en las variables *salida* y *entrada\_convolutcional* respectivamente.

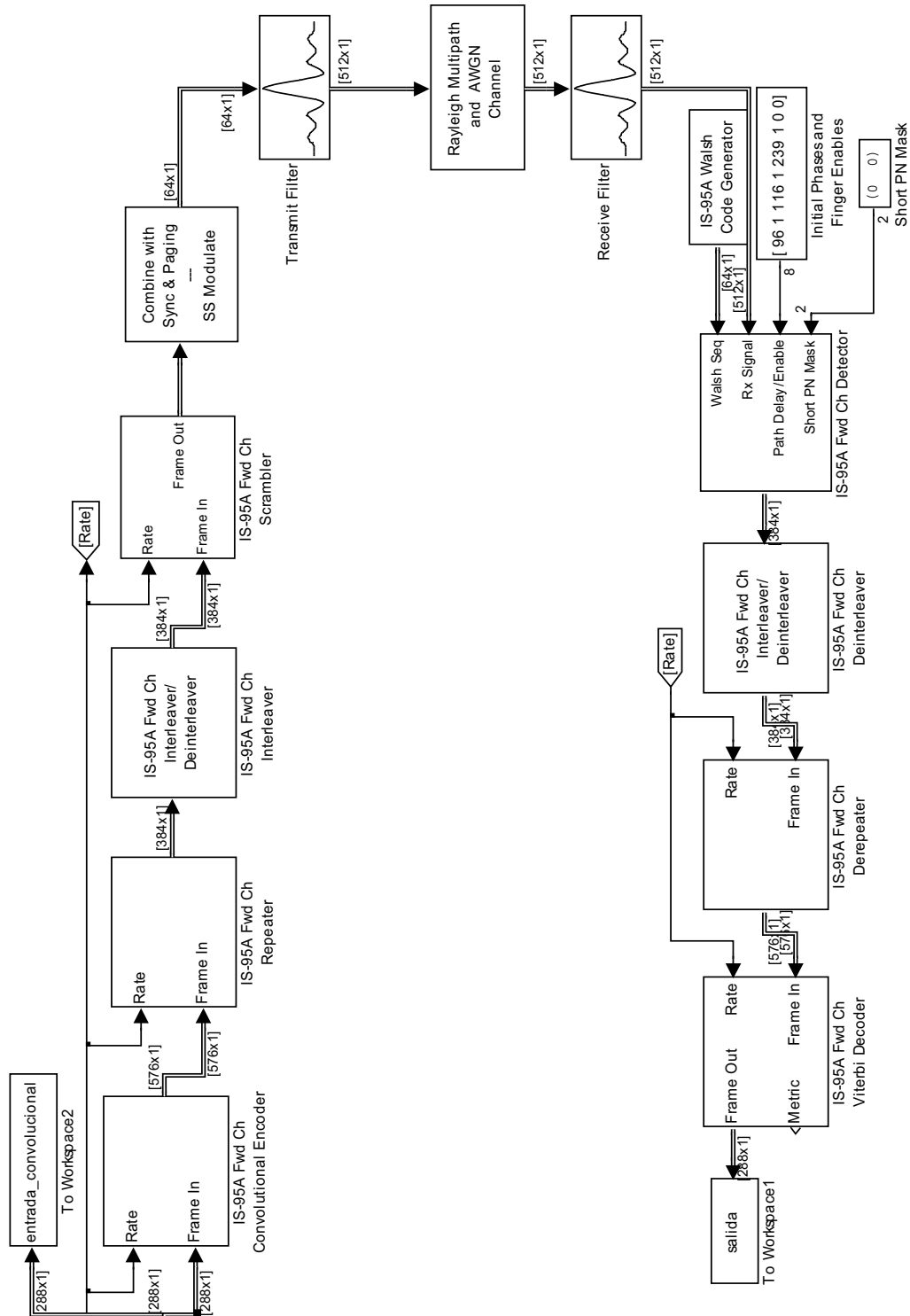
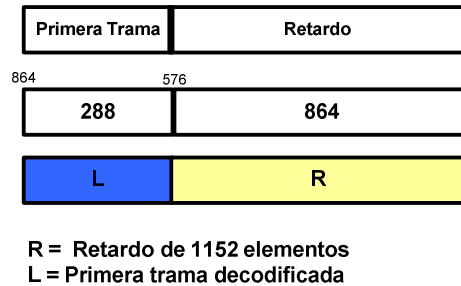


Figura 4.162 Verificación de Decodificador de Viterbi

Se almacenan los últimos 288 de 864 elementos del vector **salida** producidos en un tiempo de simulación de 0.04 segundos:



**Figura 4.163** Salida del Bloque IS-95A Viterbi Decoder para t=0.04s

```
a=salida(577:864);
```

Del vector **entrada\_convolutiva** se almacenan los primeros 288 elementos generados que entran al bloque:

```
b=entrada_convolutiva(1:288);
```

Se realiza la comparación entre los dos vectores binarios:

```
comparacion= sum(xor(a, b));
```

```
>> comparacion= sum(xor(a, b))

comparacion =

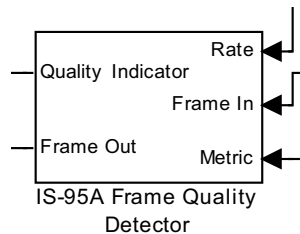
    0
```

**Figura 4.164** Comparación de los datos reales con la salida del IS-95A Fwd Ch Viterbi Decoder

Se puede observar la comparación da un resultado igual a cero, por lo tanto ambos vectores son iguales. Como se sabe, de estos 288 bits, solo 192 bits son relevantes.

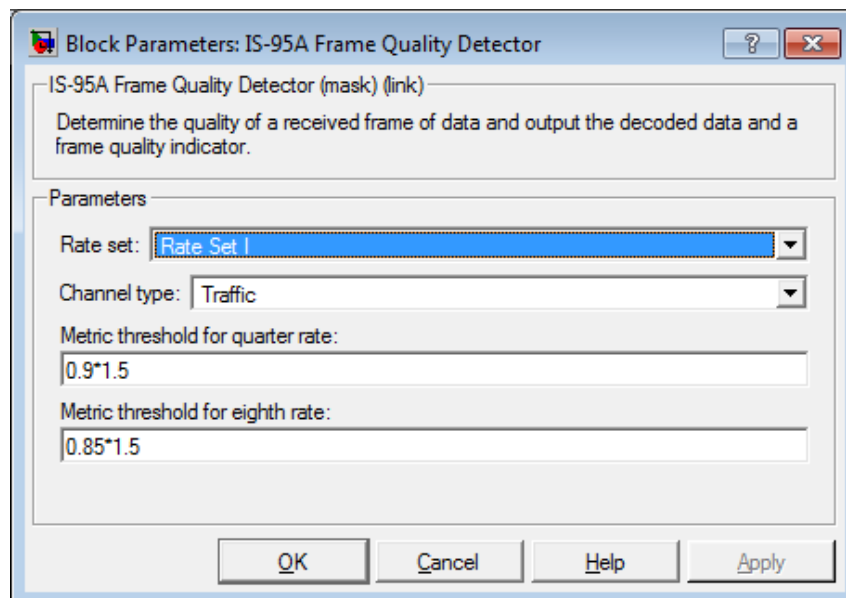
### Detector de Calidad de Trama

Para verificar si la trama tiene errores se utiliza el bloque **IS-95A Frame Quality Detector**.



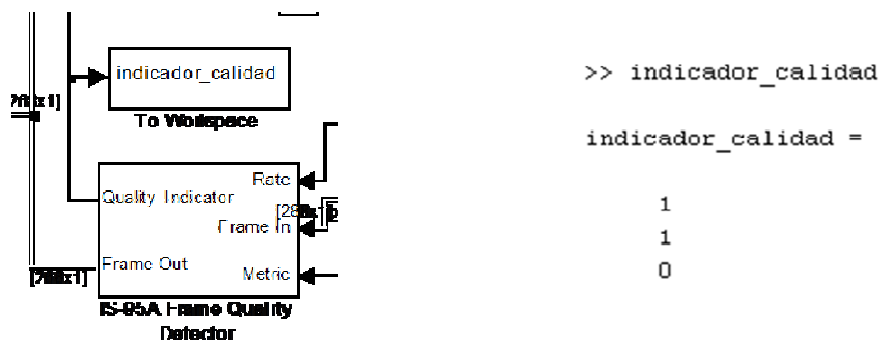
**Figura 4.165** Bloque IS-95A Frame Quality Detector.

- Este bloque determina si hay errores de trama utilizando un procedimiento de detección de síndrome para el canal de Tráfico.
- El procedimiento de detección de síndrome utiliza el bloque **IS-95A Syndrome Detector** para calcular un síndrome para la trama de datos codificada con CRC. Un valor de cero para el síndrome indica que una trama correcta ha sido recibida (**Quality Indicator**).
- El bloque remueve los bits de cola y CRC.



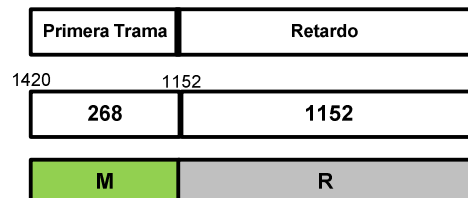
**Figura 4.166** Configuración del bloque IS-95A Frame Quality Detector.

El bloque está configurado para trabajar en el canal forward de tráfico del Rate Set I. Las métricas que están configuradas corresponden a un umbral de comparación con la métrica del decodificador Viterbi para las tramas que no poseen un campo de CRC como son las de un octavo y un cuarto de velocidad (en Rate Set I para las velocidades de 2400 y 1200 respectivamente). Para el caso de Full Rate estos valores no son tomados en cuenta. Para verificar el estado de las tramas de llegada, se observa la salida del parámetro indicador de calidad (**Quality Indicator**) en un tiempo de 0.04 segundos. Los valores que este bloque entregue se almacenan en *indicador\_calidad* tal como se muestra en la figura:



**Figura 4.167** Obtención de datos del bloque IS-95A Frame Quality Detector y muestra de los datos obtenidos para  $t=0.04$  s.

Se puede concluir que las dos primeras tramas poseen un error debido a que corresponden a un retardo agregado por el bloque **IS-95A Fwd Ch Detector**. La tercera trama de llegada corresponde a la primera trama de datos transmitida. El valor de cero indica que esta trama ha llegado sin errores.

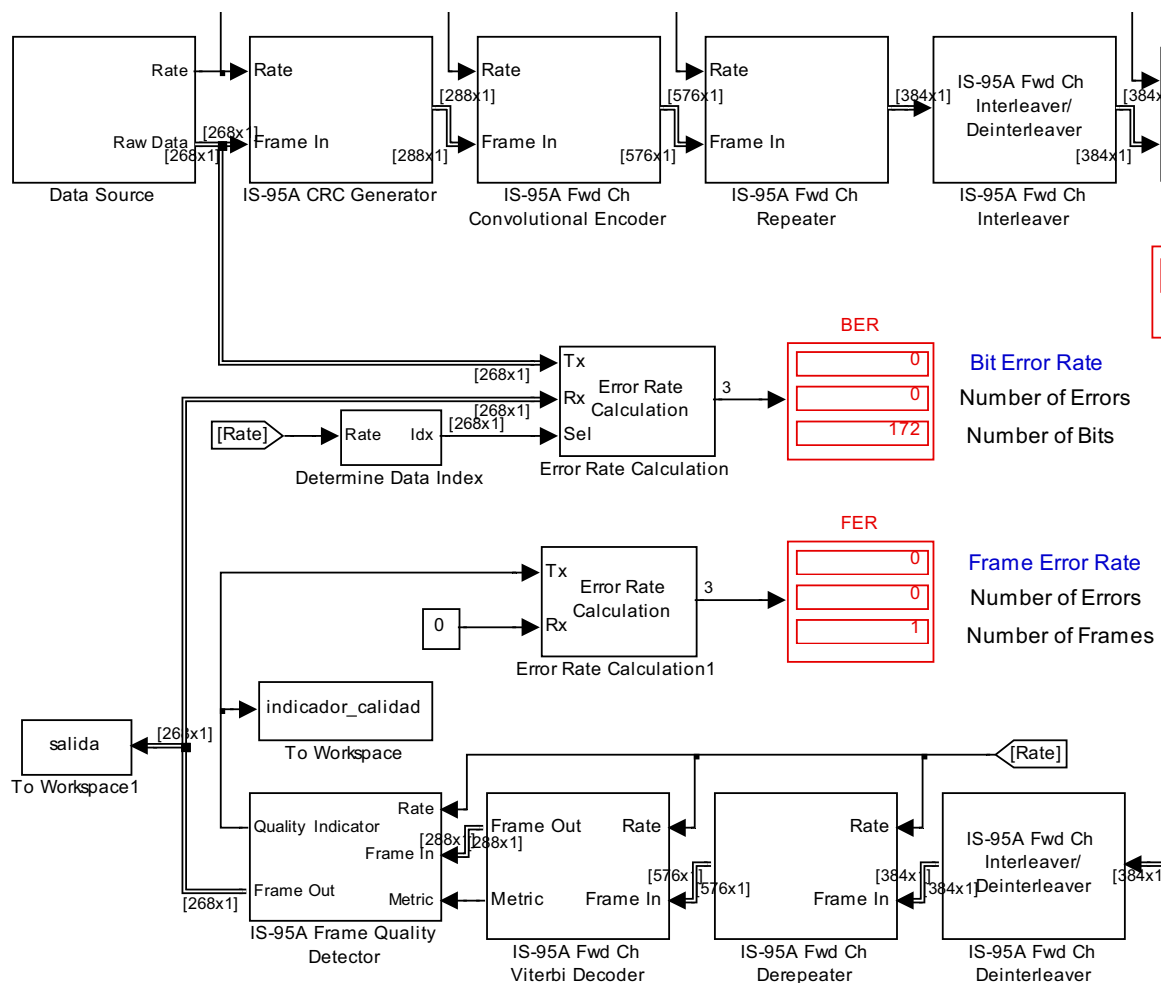


R = Retardo de 1152 elementos  
L = Primera trama sin bits de cola ni CRC

**Figura 4.168** Trama de salida del bloque IS-95A Fwd Ch Detector

## Detección de Errores

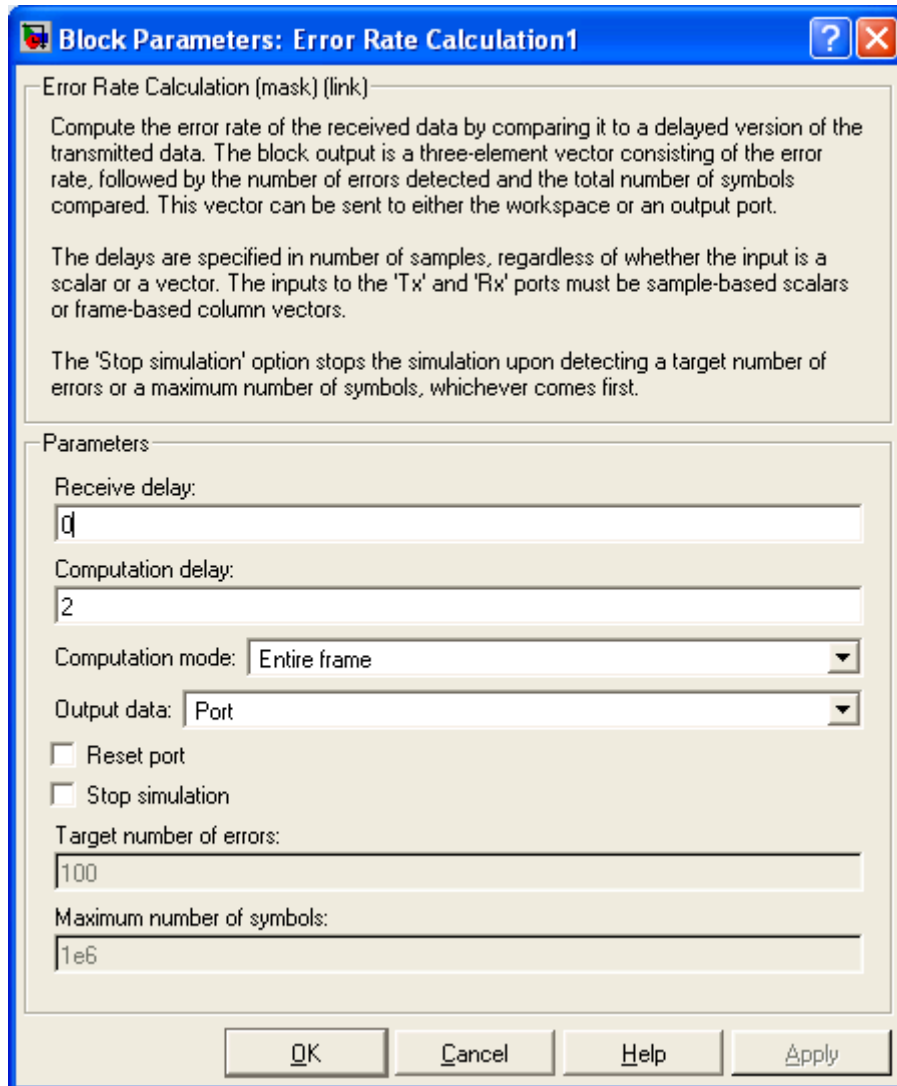
Dentro del demo están implementados los bloques **Error Rate Calculation** y **Error Rate Calculation1**. Para comprobar el funcionamiento del bloque.



**Figura 4.169** Cálculo del BER y FER.

El bloque **Error Rate Calculation1** calcula el FER (Frame Error Rate) o tasa de tramas erradas. El bloque compara una constante igual a cero con la salida **Quality Indicator** del bloque Frame Quality Detector. Un valor de cero indica que una trama ha llegado sin errores. Con estas dos entradas, el bloque sacará en un Display el número de tramas erradas y generadas en el tiempo de simulación.



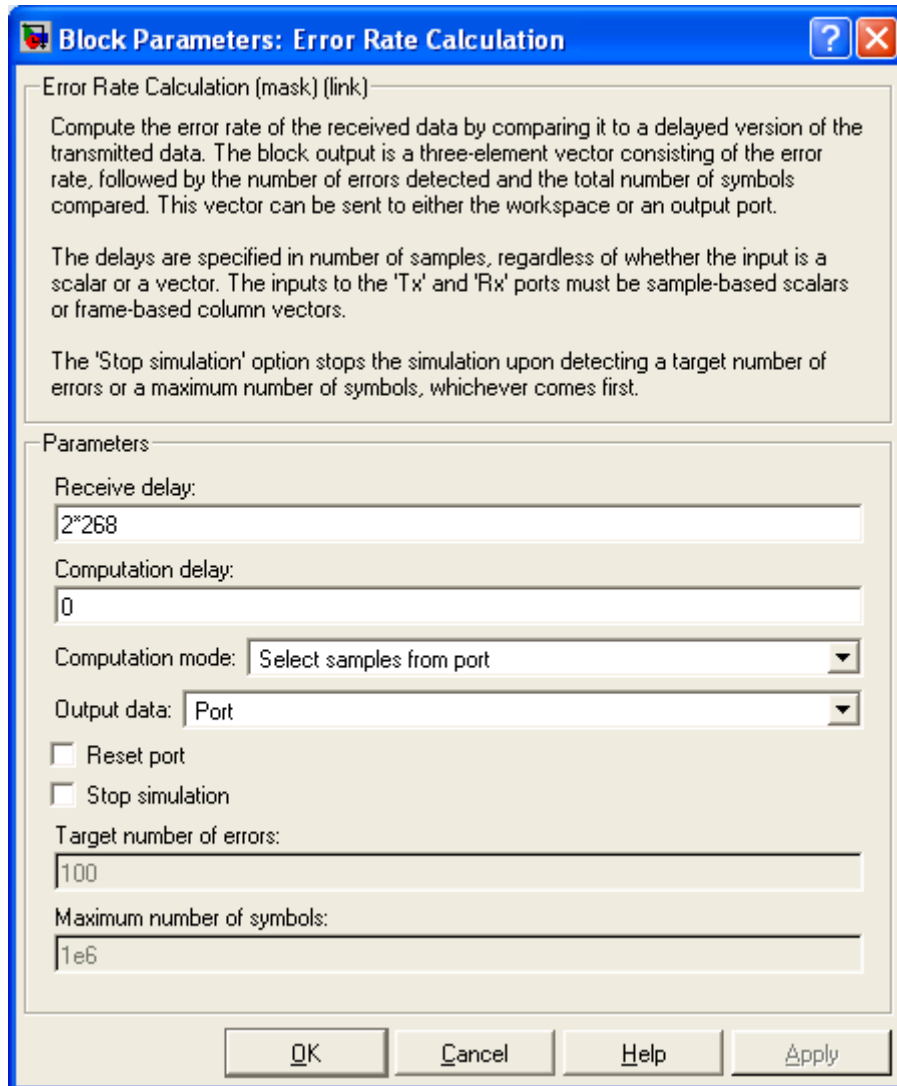


**Figura 4.170** Configuración del Bloque Error Rate Calculation1 para calcular el FER.

El bloque se configura para que no considere retrasos entre los datos recibidos y lo datos transmitidos. Además que no considere para el cálculo los valores correspondientes a las 2 primeras tramas recibidas, para realizar las comparaciones solo con los datos relevantes.

Para el ejemplo del demo, el display muestra que ha llegado una trama sin errores en  $t = 0.04s$ .

El bloque **Error Rate Calculation** calcula el BER (Bit Error Rate) o tasa de errados. El bloque compara los bits relevantes de la señal del transmisor (172 bits) con los bits relevantes de la tercera trama recibida (las dos primeras tramas corresponden a un retardo).



**Figura 4.171** Configuración del Bloque Error Rate Calculation para calcular el BER.

De la misma forma que en el apartado anterior, se configura este bloque para que solo realice las comparaciones de los datos recibidos a partir de la tercera trama con los datos transmitidos. Por lo tanto el valor del parámetro **recibe delay** corresponde a  $2 \times 268$  símbolos.

En el display se puede observar que se han comparado 172 bits relevantes del canal de tráfico y que no ha ocurrido ningún error.

Para visualizar el texto recuperado en cada trama recibida se implementó la función **recupera\_text.m**. Esta función divide el vector recibido en tramas de 268

elementos para luego decodificarlos en ASCII. La decodificación a ASCII se la realiza mediante la función *recupera\_texto.m*.

### Función recupera\_text.m

```
function texto=recupera_text(datos_recibidos)
%guardar en una matriz de tramas
filas=floor(length(datos_recibidos)/268);
matriz=[];
for i=0:filas-1
    matriz(i+1,:)=datos_recibidos(i*268+1:268*(i+1));
end
%matriz de texto enviado
matriz=matriz(:,1:168);

texto_recuperado=[];
for i=1:filas;
    aux=matriz(i,:);
    recupera=recupera_texto(aux);
    texto=[texto_recuperado; recupera];
end
```

### Función recupera\_texto.m

```
function texto_recuperado=recupera_texto(datos_recibidos)
%guardar en una matriz
filas=floor(length(datos_recibidos)/8);
matriz=[];
for i=0:filas-1
    matriz(i+1,:)=datos_recibidos(i*8+1:i*8+7);
end
texto_recuperado=[];
for i=1:filas;
    aux=matriz(i,:);
    recupera=char(binvec2dec(aux));
    texto_recuperado=[texto_recuperado recupera];
end
```

Para recuperar los datos se utiliza el bloque **To Workspace** con la variable **salida**:

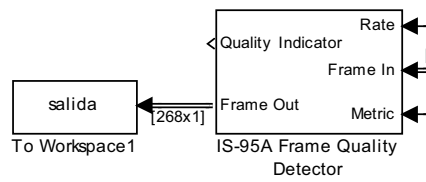


Figura 4.172 Captura de datos

Para recuperar el texto se ingresa los siguientes comandos:

```
>> a=salida';
>> d=recupera_text(a)

d =

[]
[]
Politecnica Nacional*
Politecnica Nacional*
Politecnica Nacional*
Politecnica Nacional*
Politecnica Nacional*
Politecnica Nacional*
Politecnica Nacional*
Politecnica Nacional*
Politecnica Nacional*
Politecnica Nacional*
Politecnica Nacional*
Politecnic[]%&oH!Poal*
Politecnica Nacional*
Politecnica Nacional*
Politecnica Nacional*
```

Trama recibida sin errores

Trama recibida con errores

Figura 4.173 Recuperación de texto

Se puede observar que los primeros caracteres no corresponden al texto, si no que corresponde al retardo de dos tramas propio del procesamiento en simulink. Luego de estos símbolos se recupera el texto **Politecnica Nacional\*** previamente codificado. Observando mayor cantidad de tramas se puede constatar la presencia de errores en las tramas recibidas, debido a los efectos del canal inalámbrico.

## **Conclusión**

Los resultados obtenidos en el demo demuestran que los bloques para CDMA de MATLAB 7 (***CDMA Reference Blockset***) funcionan de manera correcta según el estándar IS-95A. Se debe tener muy en cuenta la configuración de los bloques, ya que todos estos funcionan con tasas referenciadas al canal de Tráfico del enlace Forward Full rate, Rate Set II (13400 bps).

Cada bloque se encarga de reconocer los bits relevantes para cada tipo de canal y su tasa de transmisión.

Es por esto, que a pesar de trabajar en demo del canal Forward, con un canal de tráfico Full rate, Rate Set I, se generan datos con una tasa de 13400 bps.

Como problema se presentó la eliminación de datos ( $268 - 172 = 96$  bits) generados para el canal Forward, debido a que solo se consideran relevantes a los primeros 172 bits generados de la trama de 268 bits. Para evitar este problema con la simulación, se realizó el programa ***texto\_a\_bits*** que genera 172 bits relevantes más 96 bits de relleno.

En el demo se presenta en el receptor un retardo correspondiente a 2 tramas, esto debido al procesamiento del bloque detector.

Como se pudo observar, en un tiempo de simulación de 4 milisegundos se generan tres tramas, esto debido al comportamiento que Simulink presenta al comienzo de la simulación con datos generados en tramas y por tanto se genera una trama de más. Debido al retardo mencionado antes, solo se puede recuperar la primera trama enviada por el transmisor.

#### 4.2.9 CANALES REVERSE<sup>78 79 80</sup>

Un canal reverso en un sistema móvil celular IS-95A está definido como un canal físico cuya dirección de transmisión es desde la MS a la BS. Los canales reversos llevan información tanto de tráfico como de señalización.

En IS-95, hay solamente dos canales por los cuales el móvil transmite, pero no simultáneamente. Estos son el canal de Acceso o de Tráfico. La estructura del canal es similar pero más simplificada que el canal Forward, con la adición de la 64-ava modulación.

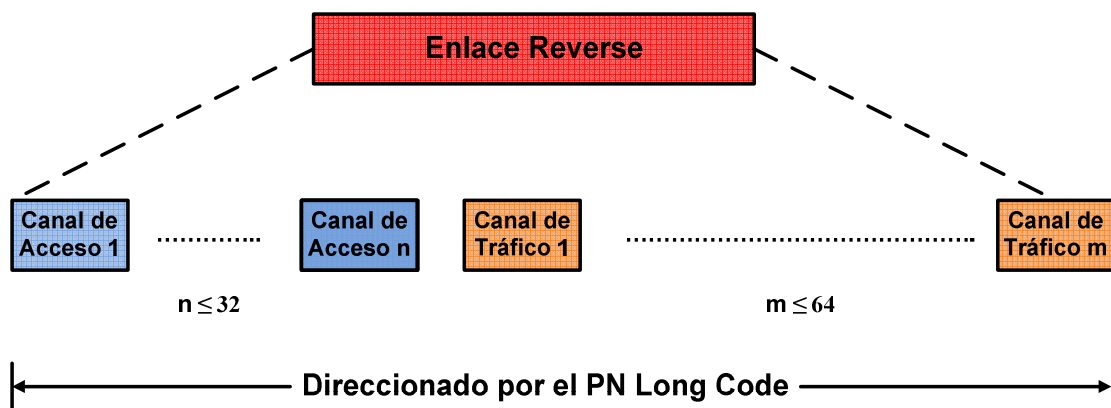


Figura 4.174 Canales del Enlace Reverse

##### 4.2.9.1 Canal de Acceso

Los canales de acceso en el enlace Reverse son usados por los móviles que todavía no están en una llamada, para transmitir peticiones de registro, peticiones de configuración de llamada, respuestas de paginación, respuestas de orden, y otras ráfagas de datos, como el proceso de mensajes entre la MS y la BS.

Un canal de acceso realmente es solo un desplazamiento de código largo público único al sector de la BS. Hay que notar que los canales de acceso en el enlace

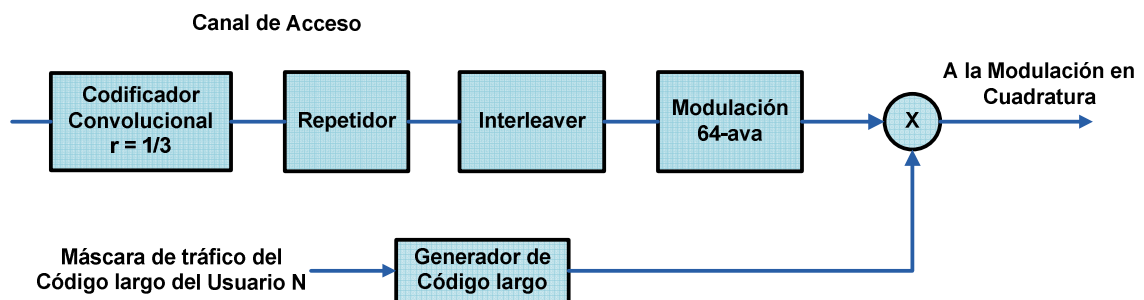
<sup>78</sup> Introduction to CDMA Wireless Communications, Abu-Rgheff, Mosa Ali, p. 496-503.

<sup>79</sup> The Next Generation CDMA Technologies, Chen, Hsiao-Hwa, p. 86-92.

<sup>80</sup> CDMA RF System Engineering, Yang, Samuel C., p. 118-125.

Reverse están siempre pareados a los canales paging en el enlace Forward. Cada canal Paging puede tener hasta 32 canales de acceso. Todos estos canales de acceso operan como canales de acceso aleatorio con una tasa de datos de 4800 bps (o una duración de trama de 20 ms).

La siguiente figura muestra el diagrama de bloques para el proceso de generación de los canales de acceso, los cuales operan en modo de conmutación de paquetes.



**Figura 4.175** Canal Reverse de Acceso

Esta información se codifica con un codificador convolutivo de tasa 1/3. Luego de esto, se utiliza un repetidor de símbolos para repetir una vez cada símbolo, dando una tasa de símbolos de 28.8 Ksps. Luego, los datos pasan a través de un interleaver para combatir el efecto del desvanecimiento.

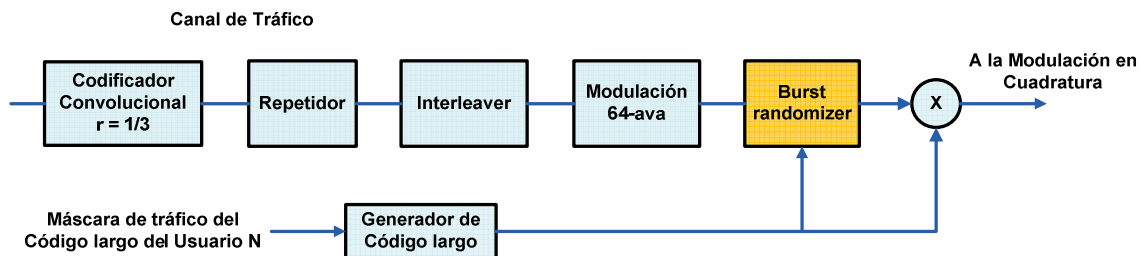
Estos datos intercalados son codificados por el Modulador 64-avo Ortogonal, donde se utilizan los códigos de Walsh para modular o representar grupos de 6 símbolos.

Los datos son ensanchados con la secuencia PN larga a una tasa de 1.2288 Mcps, para luego ser modulados en cuadratura.

#### 4.2.9.2 Canal de Tráfico Reverse

El canal de tráfico Reverse es usado para transmitir datos del usuario, voz, y también mensajes de señalización. La estructura del canal de tráfico es similar al canal de Acceso. La mayor diferencia es que el canal de Tráfico contiene un

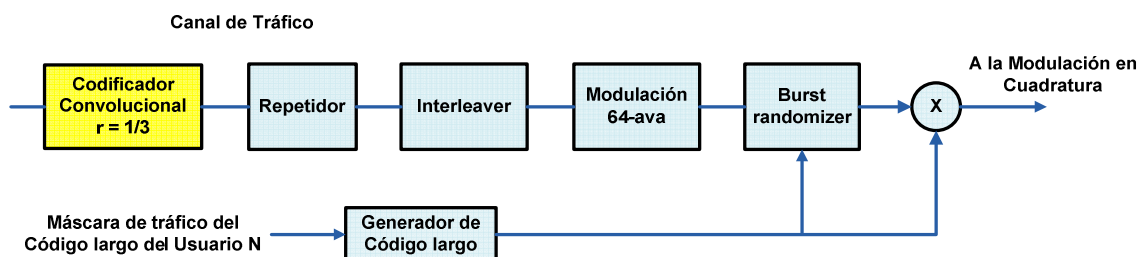
Aleatorizador de ráfagas de datos (Data Burst Randomizer), tal como se muestra en la figura:



**Figura 4.176** Canal Reverse de tráfico

El canal reverse de tráfico en IS-95A utiliza el esquema de codificación convolutiva con tasa de 1/3.

### **Codificación Convolutiva**



**Figura 4.177** Codificación convolutiva

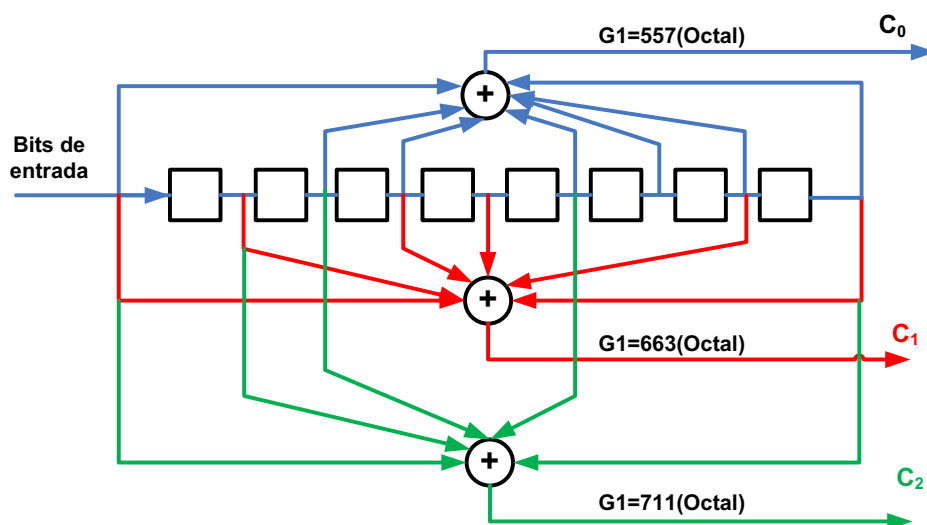
El codificador convolutivo utilizado en el canal de Tráfico Reverse se define por los siguientes polinomios:

$$x^8 + x^6 + x^5 + x^3 + x^2 + x + 1$$

$$x^8 + x^7 + x^5 + x^4 + x + 1$$

$$x^8 + x^7 + x^6 + x^3 + 1$$

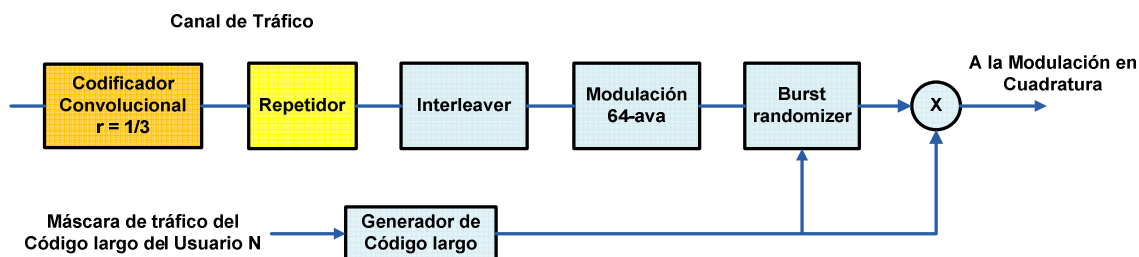




**Figura 4.178** Generador convolucional para el canal de Tráfico Reverse.

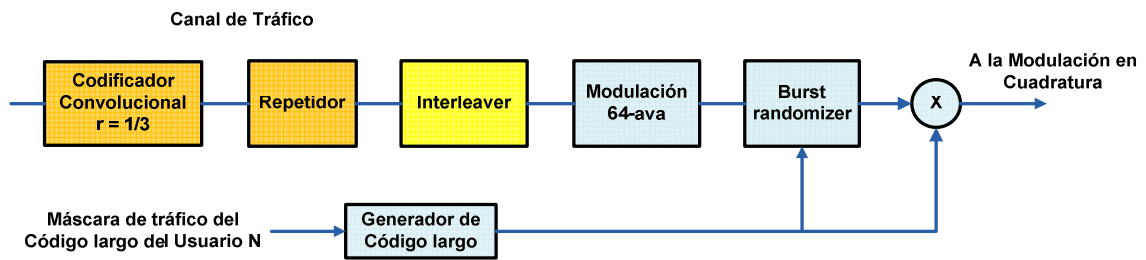
La velocidad de símbolos de código resultante es de 28.8 Ksps para Full Rate, Rate Set I

### Repetidor



**Figura 4.179** Repetidor

En caso de trabajar con canales de tráfico con tasas menores 9600 bps, se realiza la repetición de símbolos para obtener una tasa de transmisión de 28.8 Ksps. Para el canal de tráfico Full Rate, Rate Set I Reverse no es necesaria la repetición de los símbolos.

**Interleaver**<sup>81</sup>**Figura 4.180** Interleaver

El canal de tráfico reverse realiza interleaving sobre una trama de 20 ms. La tasa de entrada del Interleaver es 28.8 Ksps y por tanto el tamaño del bloque es de 576 símbolos. El enlace reverse realiza una eliminación de símbolos repetidos del canal de tráfico. Cada trama de 20 ms se divide en 16 segmentos de igual longitud (**PCGs**). Cada segmento tiene una duración de  $20 \text{ ms}/16 = 1.25 \text{ ms}$  o  $576/16 = 36$ .

A la mayor tasa de transmisión (Full rate, 9600 bps), todos los segmentos contienen símbolos no repetidos. A 4800 bps (Half rate), la mitad de los símbolos serán eliminados, de los cuales podrían estar en la posición 0 o 1, en la posición 2 o 3 y así con todos los símbolos. A 2400 (Quarter rate),  $1/4$  de símbolos no son eliminados y a 1200 bps (One-eight rate),  $1/8$  no son eliminados.

Para todas las tasas del canal de tráfico reverse es un arreglo de  $32 \times 18$ , por lo que se necesita 2 filas para llenar un **PCG**. **Las filas de salida del Interleaver son leídas en un orden que asegure que grupos de 36 símbolos (dos filas) no contienen símbolos duplicados.** A continuación se muestra el arreglo de entrada para 9600 bps:

<sup>81</sup> CDMA Systems Engineering Handbook, Lee, Jhong S.; Miller, Leonard E., p.386-388.

1	33	65	97	129	161	193	225	257	289	321	353	385	417	449	481	513	545
2	34	66	98	130	162	194	226	258	290	322	354	386	418	450	482	514	546
3	35	67	99	131	163	195	227	259	291	323	355	387	419	451	483	515	547
4	36	68	100	132	164	196	228	260	292	324	356	388	420	452	484	516	548
5	37	69	101	133	165	197	229	261	293	325	357	389	421	453	485	517	549
6	38	70	102	134	166	198	230	262	294	326	358	390	422	454	486	518	550
7	39	71	103	135	167	199	231	263	295	327	359	391	423	455	487	519	551
8	40	72	104	136	168	200	232	264	296	328	360	392	424	456	488	520	552
9	41	73	105	137	169	201	233	265	297	329	361	393	425	457	489	521	553
10	42	74	106	138	170	202	234	266	298	330	362	394	426	458	490	522	554
11	43	75	107	139	171	203	235	267	299	331	363	395	427	459	491	523	555
12	44	76	108	140	172	204	236	268	300	332	364	396	428	460	492	524	556
13	45	77	109	141	173	205	237	269	301	333	365	397	429	461	493	525	557
14	46	78	110	142	174	206	238	270	302	334	366	398	430	462	494	526	558
15	47	79	111	143	175	207	239	271	303	335	367	399	431	463	495	527	559
16	48	80	112	144	176	208	240	272	304	336	368	400	432	464	496	528	560
17	49	81	113	145	177	209	241	273	305	337	369	401	433	465	497	529	561
18	50	82	114	146	178	210	242	274	306	338	370	402	434	466	498	530	562
19	51	83	115	147	179	211	243	275	307	339	371	403	435	467	499	531	563
20	52	84	116	148	180	212	244	276	308	340	372	404	436	468	500	532	564
21	53	85	117	149	181	213	245	277	309	341	373	405	437	469	501	533	565
22	54	86	118	150	182	214	246	278	310	342	374	406	438	470	502	534	566
23	55	87	119	151	183	215	247	279	311	343	375	407	439	471	503	535	567
24	56	88	120	152	184	216	248	280	312	344	376	408	440	472	504	536	568
25	57	89	121	153	185	217	249	281	313	345	377	409	441	473	505	537	569
26	58	90	122	154	186	218	250	282	314	346	378	410	442	474	506	538	570
27	59	91	123	155	187	219	251	283	315	347	379	411	443	475	507	539	571
28	60	92	124	156	188	220	252	284	316	348	380	412	444	476	508	540	572
29	61	93	125	157	189	221	253	285	317	349	381	413	445	477	509	541	573
30	62	94	126	158	190	222	254	286	318	350	382	414	446	478	510	542	574
31	63	95	127	159	191	223	255	287	319	351	383	415	447	479	511	543	575
32	64	96	128	160	192	224	256	288	320	352	384	416	448	480	512	544	576

**Figura 4.181** Arreglo de Interleaver para canal de tráfico a 9600 bps

En el arreglo para la tasa de 4800 bps se puede observar la repetición de símbolos:

1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241	257	273
1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241	257	273
2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242	258	274
2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242	258	274
3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243	259	275
3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243	259	275
4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244	260	276
4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244	260	276
5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245	261	277
5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245	261	277
6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246	262	278
6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246	262	278
7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247	263	279
7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247	263	279
8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248	264	280
8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248	264	280
9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249	265	281
9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249	265	281
10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250	266	282
10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250	266	282
11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251	267	283
11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251	267	283
12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252	268	284
12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252	268	284
13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253	269	285
13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253	269	285
14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254	270	286
14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254	270	286
15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255	271	287
15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255	271	287
16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	256	272	288
16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	256	272	288

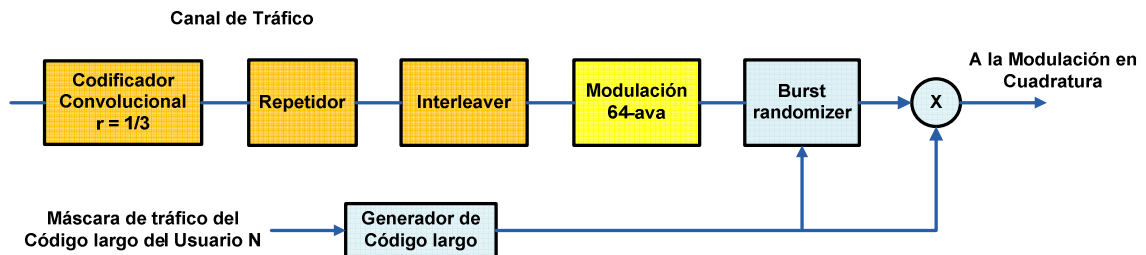
**Figura 4.182** Arreglo de Interleaver para canal de tráfico a 4800 bps

Los símbolos son leídos a la salida del Interleaver por filas en el orden que se muestra a continuación:

**Tabla 4.12** Orden de lectura de filas de Interleaver para canales de Tráfico reversos

Tasa	Número de Fila
9600 bps	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,25,26,27, 28,29, 30, 31
4800 bps	0, 2, 1, 3,4, 6, 5,7, 8, 10,9, 11, 12, 14, 13, 15, 16, 18, 17, 19,20, 22, 21, 23, 24, 26, 25, 27, 28, 30, 29, 31
2400 bps	0,4, 1, 5,2, 6,3,7, 8, 12,9, 13, 10, 14, 11, 15, 16, 20, 17, 21, 18, 22, 19, 23, 24, 28, 25, 29, 26, 30, 27, 31
1200 bps	0, 8, 1, 9,2, 10, 3, 11,4, 12, 5, 13, 6, 14,7, 15, 16, 24, 17, 25, 18, 26, 19,27,20, 28, 21,29, 22, 30, 23,31

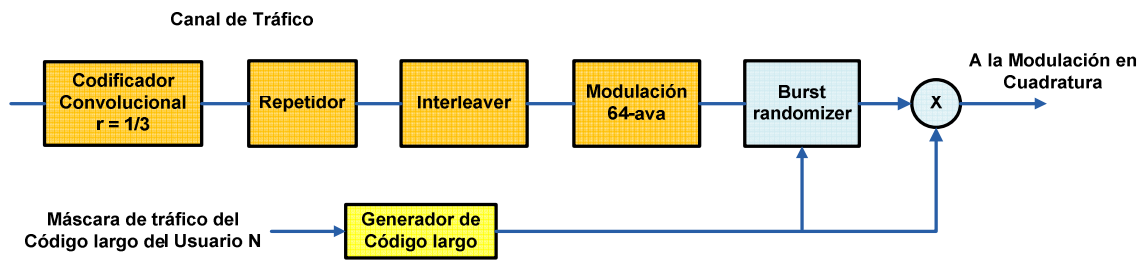
### Modulación 64-ava

**Figura 4.183** Modulación 64-ava.

La modulación 64-ava Ortogonal de Walsh, se logra de la siguiente manera: La salida del Interleaver es particionada en bloques de 6 símbolos usando para seleccionar una de las 64 códigos de Walsh. El bloque de 6 símbolos se convierte a un número decimal  $k$  entre  $0 \leq k \leq 63$  luego la función Walsh con el índice  $k$  es seleccionado para transmitir. Ya que seis símbolos intercalados son modulados como una secuencia Walsh de 64 chips, existen 4800 secuencias Walsh por segundo, lo cual implica una tasa de chip de  $4800 \times 64 = 307.2$  Kcps. Por ejemplo, si el bloque de símbolos es [110101] luego se envía la función Walsh con índice 53.

A diferencia del Enlace Forward, el enlace reverse utiliza la Modulación 64-ava Ortogonal de Walsh.

### Direccionamiento del Canal



**Figura 4.184** Direccionamiento del canal.

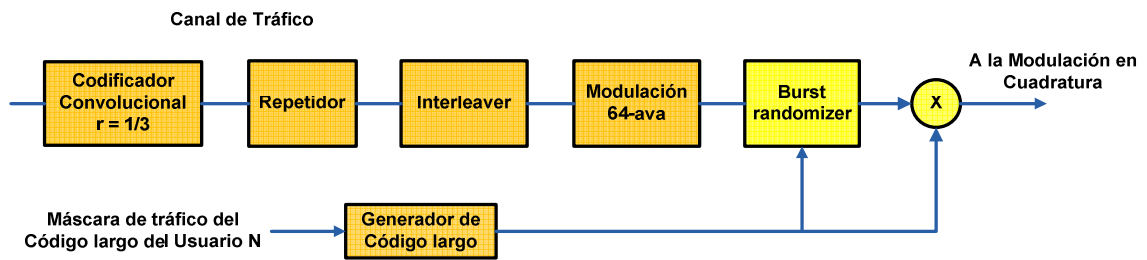
A diferencia del canal Forward en IS-95A, el cual necesita los códigos de Walsh para diferenciar las MSs en el sector o celda, los canales Reverse CDMA consisten de  $2^{42}-1$  canales lógicos, los cuales pueden ser utilizados para diferenciar casi todas las MSs dentro de una red celular de un proveedor de un servicio en particular. Cada uno de los canales lógicos está permanentemente y únicamente asociados con una MS particular. El canal asignado no cambia incluso en los handoffs, en el cual las MS viajan a través de los bordes de las celdas o sectores.

El direccionamiento del canal se logra con la manipulación de la fase inicial del código PN largo (el cual tiene un periodo de  $2^{42}-1$ ).

Con esto se debe señalar que el canal Reverse CDMA no se cumple de ninguna manera una regla ortogonal, en cambio utiliza un código de ensanchamiento de periodo muy largo o un código PN largo con diferentes fases.

La función de correlación cruzada entre diferentes MSs no es necesariamente cero, pero es lo suficientemente pequeña para controlar la interferencia.

### Aleatorizador de Ráfagas de Datos<sup>82</sup>



**Figura 4.185** Aleatorizador de Ráfagas de Datos.

Los datos modulados ortogonalmente son alimentados al Aleatorizador de datos. El Aleatorizador de datos genera un patrón de enmascaramiento de 0s y 1s que enmascara de forma aleatoria los bits redundantes. El patrón de enmascaramiento es determinado parcialmente por la tasa del vocoder. Si el vocoder está operando a 9.6 Kbps, los datos no son enmascarados. Si el vocoder opera a 1.2 Kbps, los símbolos son repetidos siete veces, y el aleatorizador de datos enmascara en promedio, siete de ocho grupos de símbolos.

Como se mencionó anteriormente, cada canal de tráfico se divide en 16 PCGs, cada uno con una duración de 1.25 ms. El Aleatorizador de datos enmascara pseudoaleatoriamente PCGs individuales. Cuando el vocoder opera a 9.6 Kbps, ningún PCG es enmascarado; cuando el vocoder opera a 4.8 Kbps, un promedio de 8 PCGs son enmascarados en una trama; cuando el vocoder opera a 2.4 Kbps, un promedio de 12 PCGs son enmascarados y cuando el vocoder opera a 1.2 Kbps, un promedio de 12 PCGs son enmascarados. Además de depender de la velocidad del vocoder, el patrón de enmascaramiento depende de la secuencia larga PN usada para ensanchar previamente la trama.

<sup>82</sup> CDMA RF System Engineering, Yang, Samuel C., p. 122-126.

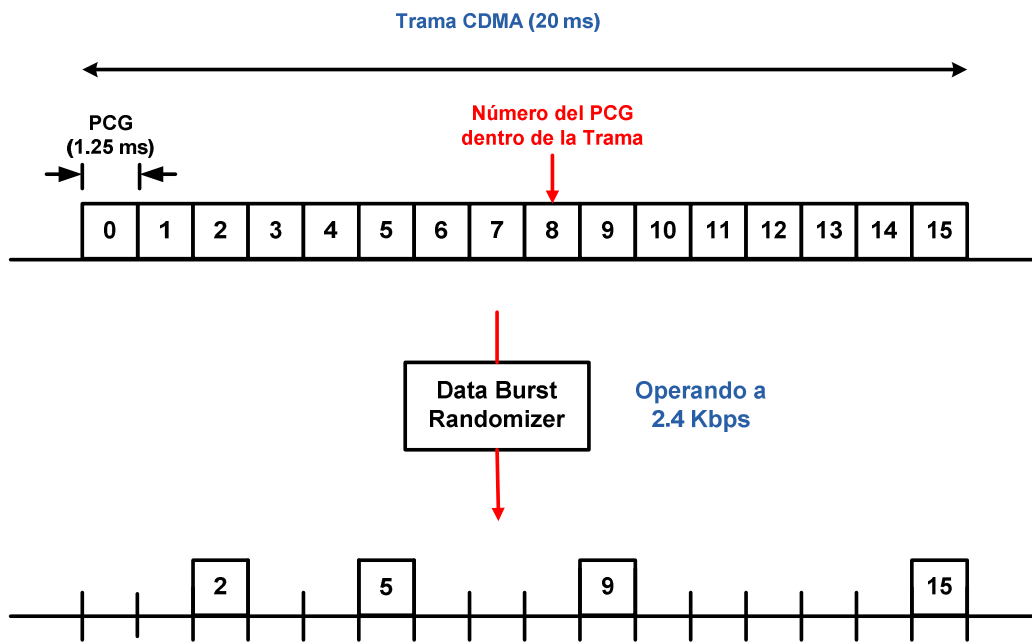


Figura 4.186 Operación del Burst Randomizer.

La estructura del canal de Tráfico Reverse para el Rate Set 2 es similar. Difiere en que para mantener los 28.8 Kbps a la salida del interleaver, la tasa del codificador convolucional cambia a  $r=1/2$ .

### Modulación OQPSK

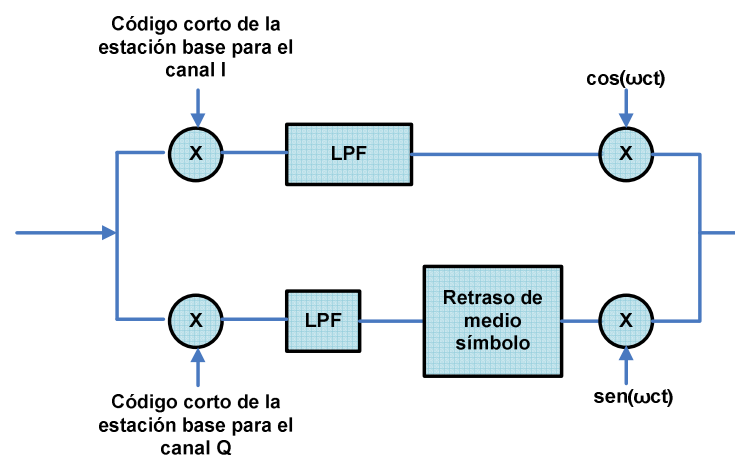


Figura 4.187 Modulación OQPSK

Mientras la modulación de portadora en los canales del enlace Forward son QPSK, la modulación de portadora en los canales del enlace Reverse es O-QPSK. La diferencia entre QPSK y OQPSK se encuentra en el hecho de que se



introduce un retraso de  $\frac{1}{2}$  de chip en el modulador OQPSK. Los beneficios de utilizar OQPSK en lugar de QPSK incluyen una importante reducción en la cantidad de cambios de fases de portadora posibles como cambios del valor del chip en el tiempo.

Se utilizan tres esquemas diferentes de diversidad en el enlace que sirven como una protección adicional contra los efectos nocivos presentes en un ambiente hostil de transmisión asincrónica: diversidad de tiempo a través de la codificación e interleaving, diversidad de trayectoria a través del receptor Rake, y diversidad espacial utilizando dos antenas receptoras en cada Estación Base.

Resumiendo las principales características del enlace reverse son:

- Los códigos de Walsh son utilizados para la modulación de símbolo a través del mapeo de cada 6 bits del bloque de datos.
- El código PN largo es usado para la identificación del canal y es concatenado con el código PN corto.
- La demodulación no coherente OQPSK es utilizada en el receptor móvil para la modulación de portadora.
- Uso de diversidad de tiempo, de trayectoria, y de espacio para mejorar la calidad del enlace.

#### **4.2.9.3 Análisis del Demo IS-95A Reverse Traffic Channel Detection Model**

Este demo simula la modulación y ensanchamiento de los símbolos de datos del transmisor de la Estación Móvil, así como el desensanchamiento y demodulación no coherente en el receptor de la estación base. Es así que se simula el canal de tráfico del enlace Reverse a la mayor tasa de transmisión (Full Rate). No se considera la codificación del canal, ya que es similar a la del enlace Forward, con ciertas diferencias como la tasa del codificador convolucional o el interleaver de los datos.

### IS-95A Reverse Traffic Channel Detection

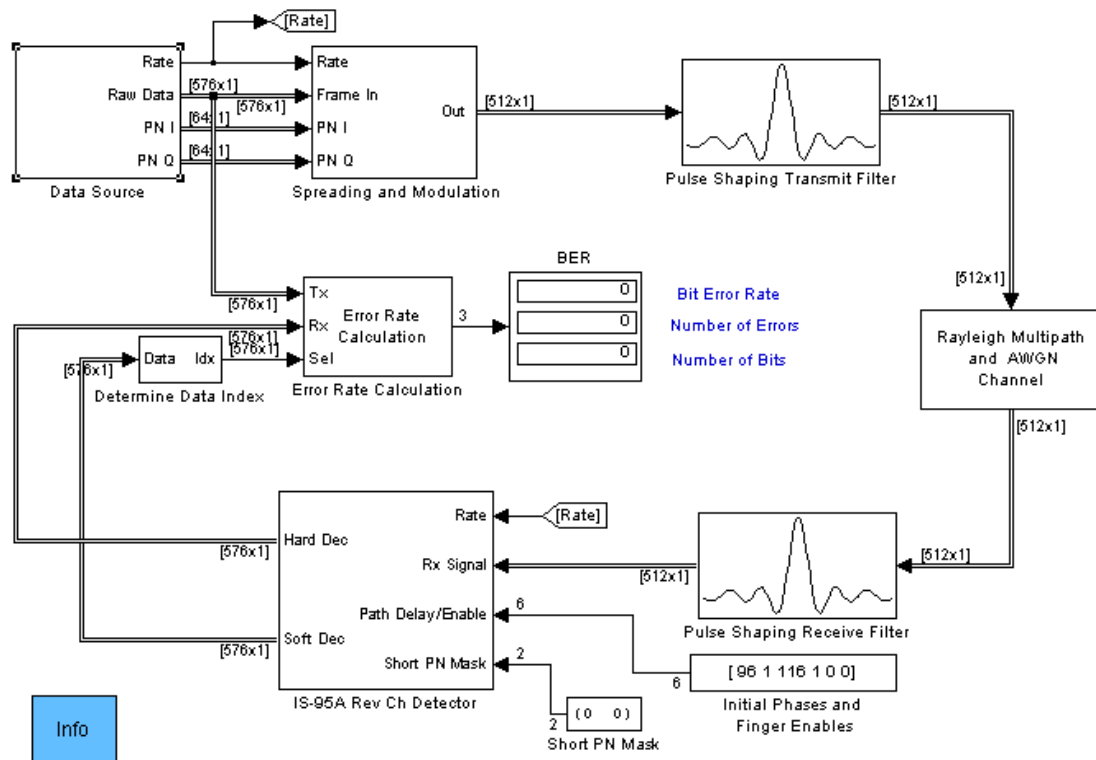


Figura 4.188 Sistema de bloques en Simulink.

El análisis del demo se realiza con un tiempo de simulación de 20 milisegundos.

Para realizar la captura de las señales después de cada bloque, se utiliza el bloque **To Workspace** con las mismas configuraciones aplicadas en el demo del enlace Forward.

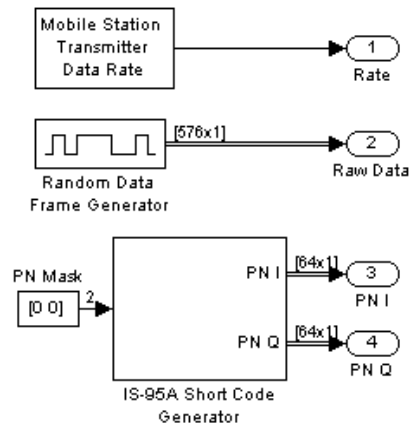
### Generación de Datos

Para la generación de datos en el demo, se utiliza el bloque **Data Source**:



Figura 4.189 Bloque Data Source para demo IS-95A

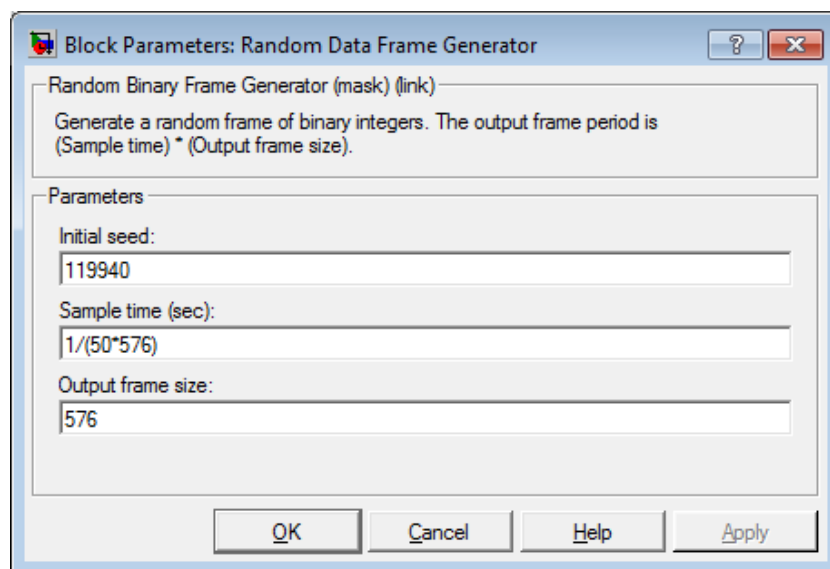
Este bloque está conformado por el bloque generador de tramas binarias aleatorio **Random Data Frame Generator**, el bloque **Mobile Station Transmitter Data Rate** y el bloque **IS-95A Short Code Generator** con su respectiva máscara.



**Figura 4.190** Bloque Data Source expandido

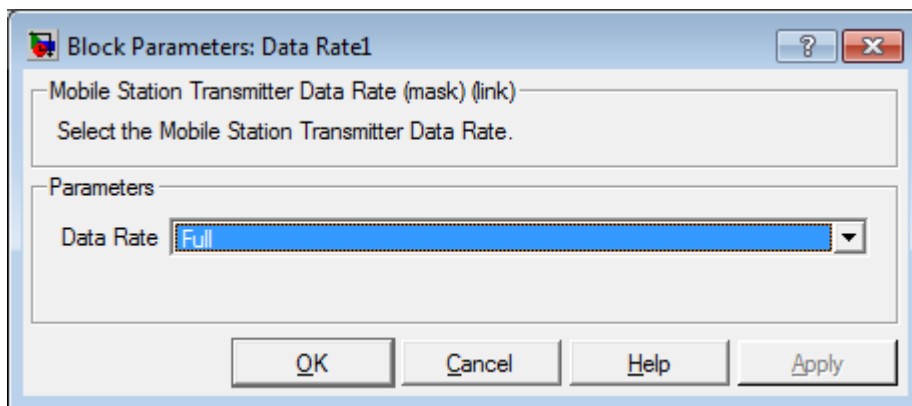
El bloque **Random Data Frame Generator** genera tramas de forma aleatoria con, de igual manera que el bloque **Random Binary Frame Generator** utilizado en el demo del enlace Forward, con la diferencia que para este demo se generan tramas de 576 elementos cada 20 milisegundos.

Esta tasa se considera luego que los datos han pasado por la etapa de codificación.



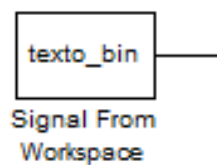
**Figura 4.191** Configuración del bloque Random Binary Frame Generator

El bloque **Mobile Station Transmitter Data Rate** permite establecer la tasa de transmisión del canal de tráfico, como en el demo analizado anteriormente. En el demo se implementa Rate Set I con la mayor tasa de transmisión **Full rate**.



**Figura 4.192** Configuración del bloque Data Rate

Para propósitos de análisis del demo se ha cambiado el bloque **Random Data Frame Generator** por el bloque **Signal From Workspace**.



**Figura 4.193** Bloque Signal From Workspace

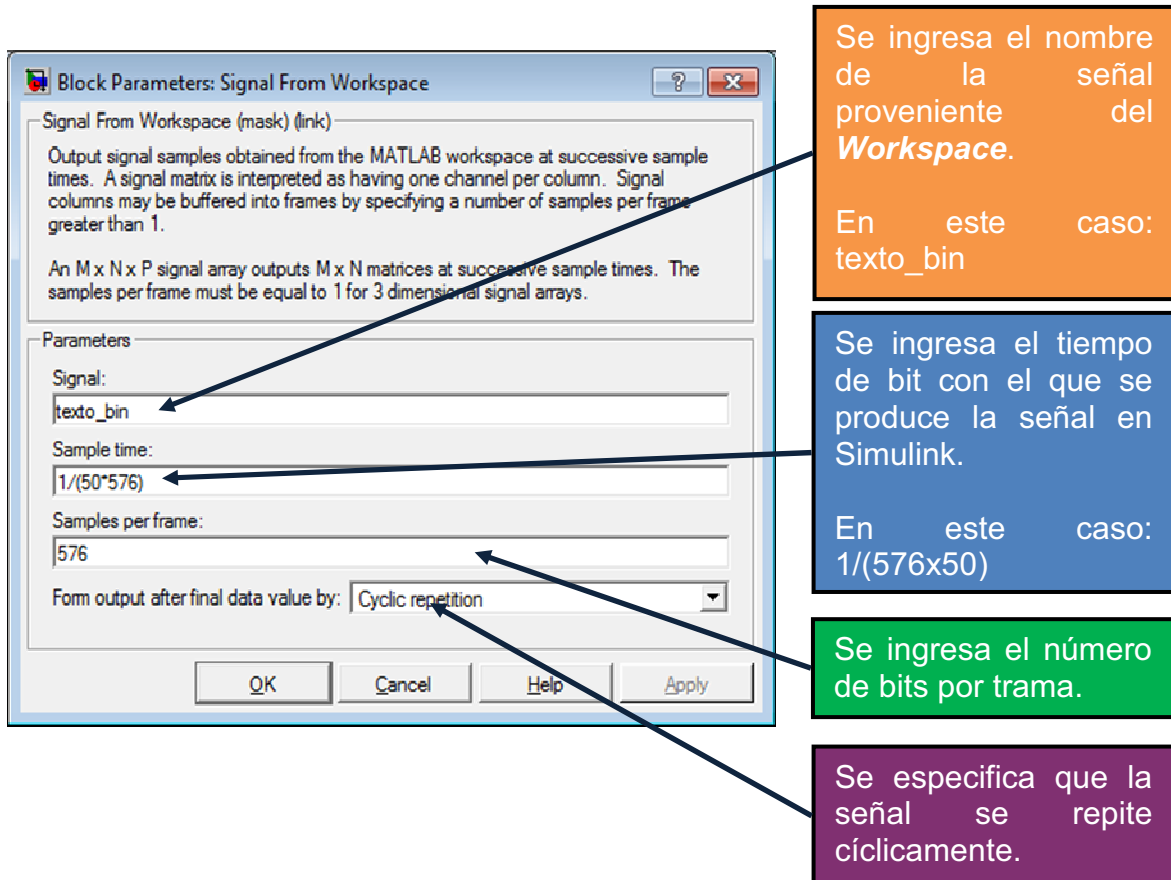


Figura 4.194 Configuración del bloque Signal From Worspace

Para el análisis del canal Forward de Tráfico Rate Set I (9600 bps), se utiliza el siguiente formato de trama:

Se genera una trama de 576 bits, en la cual se encuentra un texto de caracteres en formato ASCII. Se importa la señal **texto\_bin** guardada en el archivo **prueba.mat**. Este archivo se generó con el programa **texto\_a\_bits\_reverse.m** en el cual se hace la conversión del texto **\*\*\*\*\*Escuela Politecnica Nacional Analisis del Canal Reverse CDMA\*\*\*\*\*** a bits. El código es:

```
clear all; clc
mensaje=('*****Escuela Politecnica Nacional Analisis del Canal
Reverse CDMA*****');
texto_original= mensaje;
texto_ascii=abs(texto_original);
texto_bin=[];
for i=1:length(texto_ascii)
    texto_bin=cat(2,texto_bin,dec2binvec(texto_ascii(i),8));
end; %Se tiene una trama 576 bits
```

El generador de código PN corto se utiliza de igual manera que en el enlace Forward. Se generan dos secuencias cortas **PN I** y **PN Q** a una tasa de 1.2288 Mcps, en tramas de 64 elementos y con una máscara igual a 0 para ambos códigos.

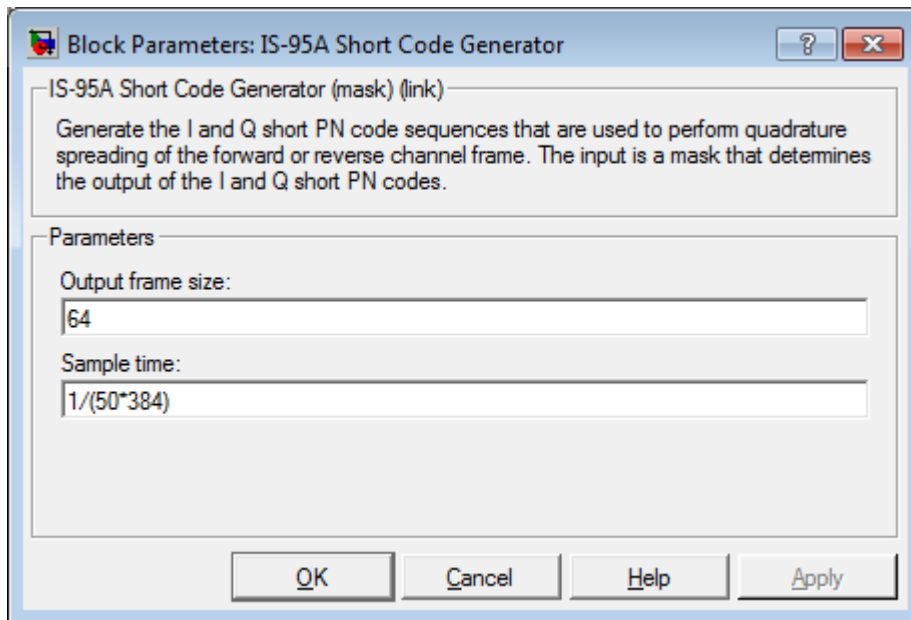


Figura 4.195 Parámetros del Bloque IS-95A Short Code Generator.

### **Ensanchamiento y Modulación**

Para realizar el ensanchamiento y modulación de la señal de tráfico, se utiliza el bloque **Spreading and Modulation**.

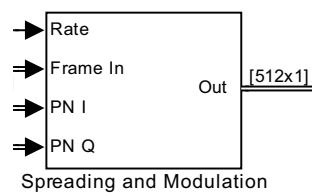
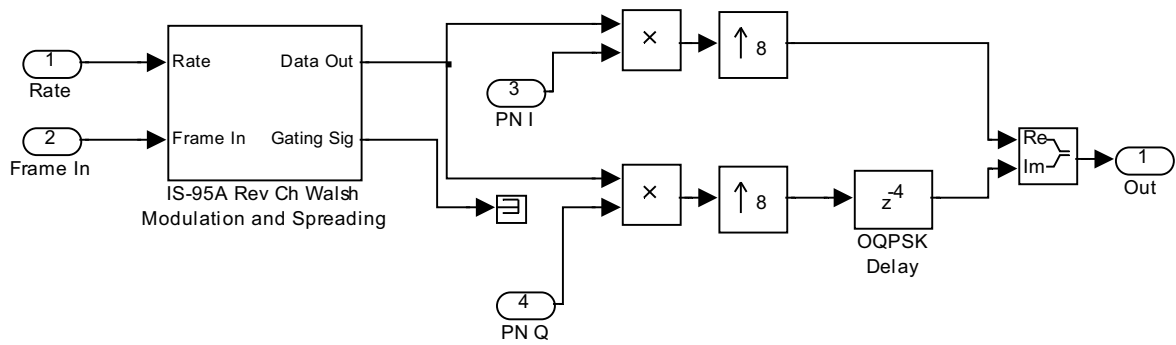


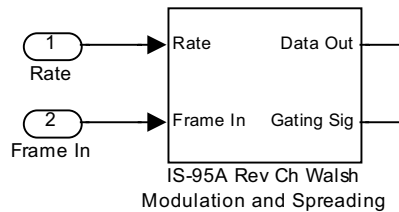
Figura 4.196 Bloque Spreading and Modulation

Expandiendo el bloque se tiene el siguiente subsistema:



**Figura 4.197** Bloque Spreading and Modulation expandido

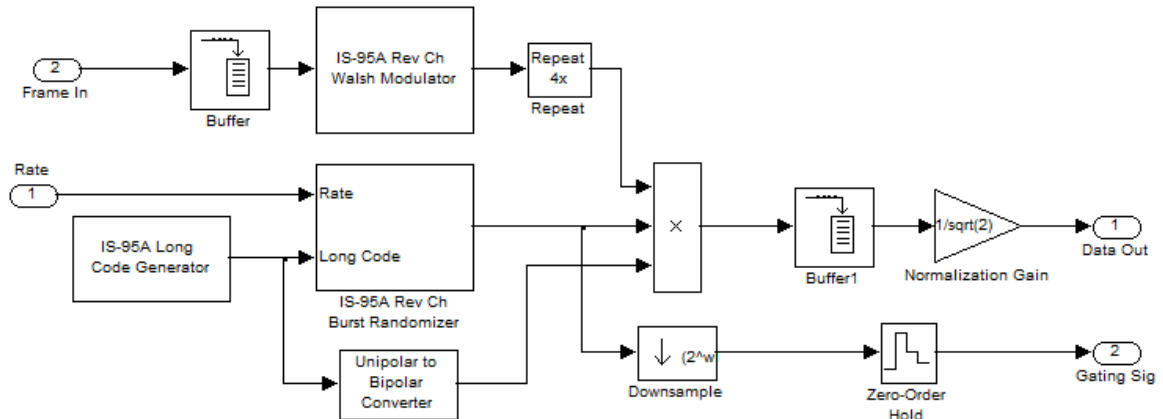
Para realizar la modulación 64-ava y para realizar la eliminación de símbolos repetidos se utiliza el bloque **IS-95A Rev Ch Walsh Modulation and Spreading**:



**Figura 4.198** Bloque IS-95A Rev Ch Walsh Modulation and Spreading.

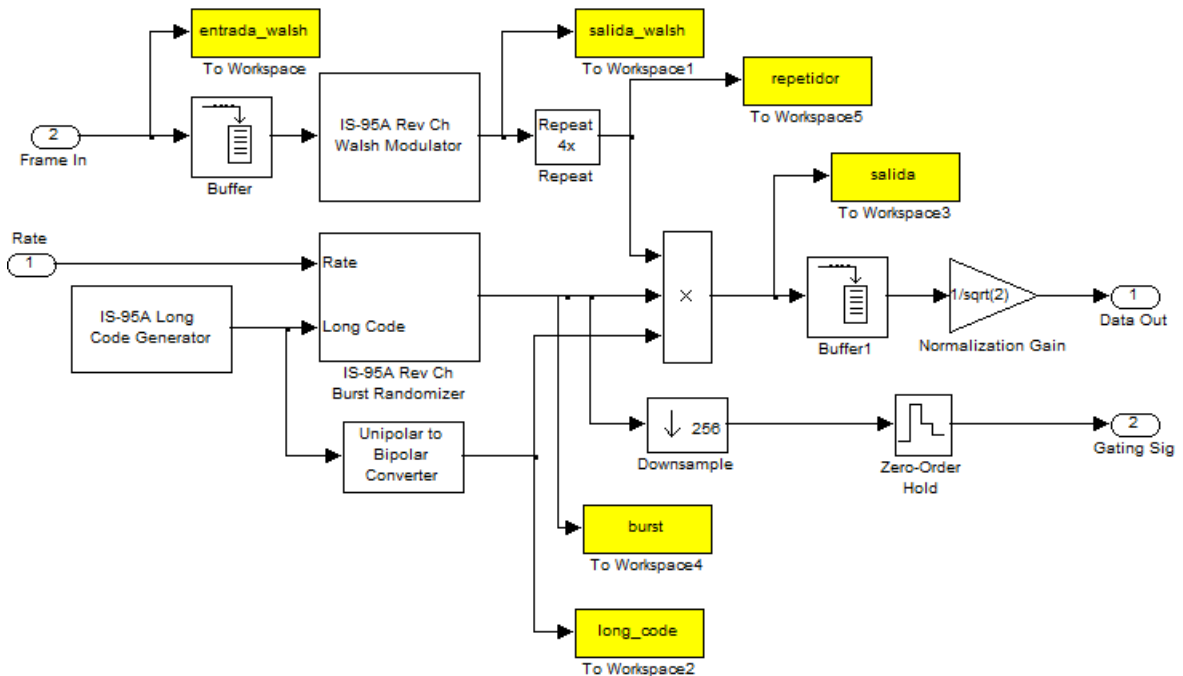
**Figura 4.199** Parámetros del bloque IS-95A Rev Ch Walsh Modulation and Spreading.

Expandiendo el subsistema IS-95A Rev Ch Walsh Modulation and Spreading se tiene:



**Figura 4.200** Bloque IS-95A Rev Ch Walsh Modulation and Spreading expandido.

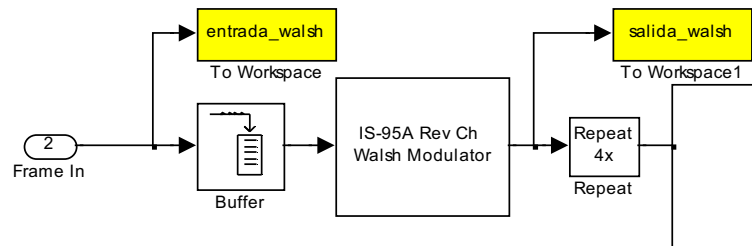
Para realizar el análisis de las etapas de la modulación y spreading, se añade bloques to workspace a la entrada y salida de cada etapa, tal como se muestra en la siguiente figura:



**Figura 4.201** Sistema del demo con bloques To Workspace

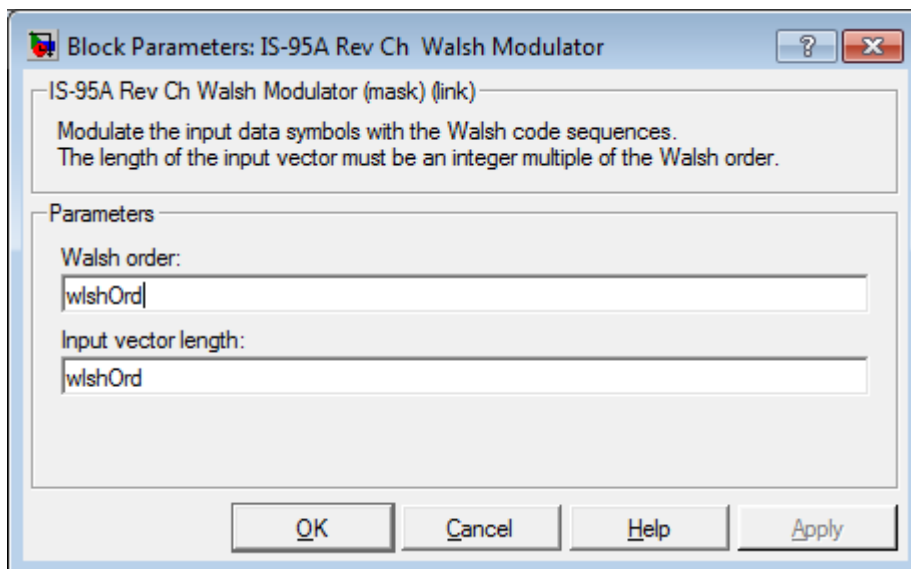


## Modulación Walsh



**Figura 4.202** Modulación Walsh

Este bloque primero realiza una 64-ava modulación Walsh, donde  $64 = 2^6$  y 6 es el orden del orden de los códigos de Walsh (parámetro **Walsh order**).



**Figura 4.203** Configuración de los parámetros del bloque IS-95A Rev Ch Walsh Modulator.

El buffer divide a los datos en grupos de 6 bits para la entrada del modulador. Luego realiza la conversión de los 6 bits de entrada a su valor decimal correspondiente, para conseguir el índice del código Walsh que se debe generar.

En **entrada\_walsh** se guardarán los bits de entrada al modulador de Walsh y en **salida\_walsh** se almacenarán los bits de salida del generador Walsh. Para el análisis se escogen los primeros seis bits de la entrada y se los convierte a su equivalente decimal para conocer el código Walsh que se debe generar a la salida. Para esto, se ingresa los siguientes comandos en Matlab:

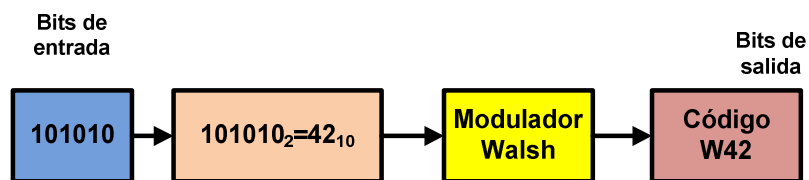
```

>> entrada_walsh(1:6) '
ans =
     0     1     0     1     0     1
>> binvec2dec(ans)
ans =
    42

```

**Figura 4.204** Datos transformados a índice de Walsh

Con el comando ***binvec2dec*** se consigue transformar un vector binario a su correspondiente número decimal. Note, que el primer elemento del vector es el menor bit significativo.



**Figura 4.205** Modulación 64-ava.

Por lo tanto, la salida del modulador debe corresponder al código Walsh 42.

Para comprobar la funcionalidad del bloque, se ingresa los siguientes comandos en el ***workspace*** de Matlab:

```

>> walsh=hadamard(64);
>> walsh42=walsh(43,:);
>> b=salida_walsh(1:64)';
>> sum(xor(walsh42,b))

ans =
     0

```

**Figura 4.206** Comparación de código de Walsh con salida de modulación 64-ava

Donde, primero se genera una matriz que contiene todos los códigos Walsh de tamaño 64. Luego se escoge el código Walsh 42, el cual corresponde al código de la fila 43 de la matriz de Hadamard generada en Matlab.

Para comprobar que se ha generado el mismo código, se realiza la XOR entre los 64 primeros bits de la salida del modulador Walsh y el código Walsh calculado para el ejemplo. Cuando los elementos de un vector son iguales, la función XOR genera un vector nulo. Por lo tanto al realizar la suma de todos los elementos del vector, el resultado debe ser cero. Como se pudo ver, el resultado que se obtiene es cero, por lo tanto el código de salida del modulador Walsh corresponde a  $W_{42}$ .

Luego el bloque genera internamente la secuencia PN larga con la cual ensancha los símbolos Walsh que salen a una tasa de 307,2 Kbps. El generador de secuencia PN larga produce la secuencia con una tasa de 1.2288 Mcps.

Para poder realizar la multiplicación entre esta secuencia y los datos producidos por la 64-ava modulación a 307.2 Kcps, se debe repetir cada muestra de la señal modulada por el factor de 4.

Tomando 16 de los valores almacenados en *repetidor* y comparándolos con los primeros 4 bits del código Walsh, se tiene:

```
>> a=repetidor(1:16)';
>> disp(a)
Columns 1 through 11
 1   1   1   1   1   1   1   1   1   1   1
Columns 12 through 16
-1  -1  -1  -1  -1
>> walsh42(1:4)
ans =
 1   1  -1  -1
```

**Figura 4.207** Verificación de repetición de muestras

En la figura se puede ver que los símbolos correspondientes al código de Walsh son repetidos 4 veces, antes de realizar el ensanchamiento por la secuencia PN larga, ya que 1 bit del código Walsh debe multiplicarse por 4 chips de la secuencia PN larga. Es decir que para este caso se tiene  $G_p=4$ .

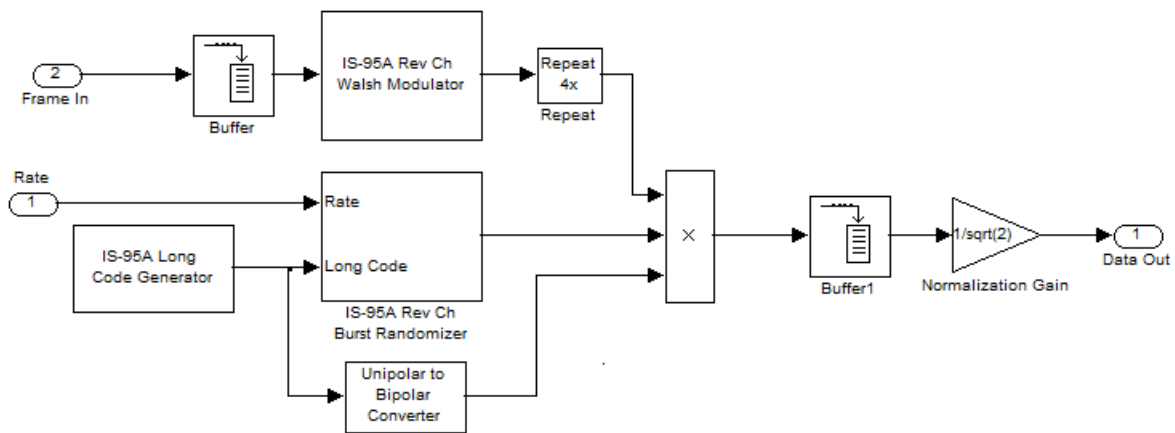


Figura 4.208 Modulación y pseudoaleatorización de datos

La configuración del bloque **IS-95A Long Code Generator** es el siguiente:

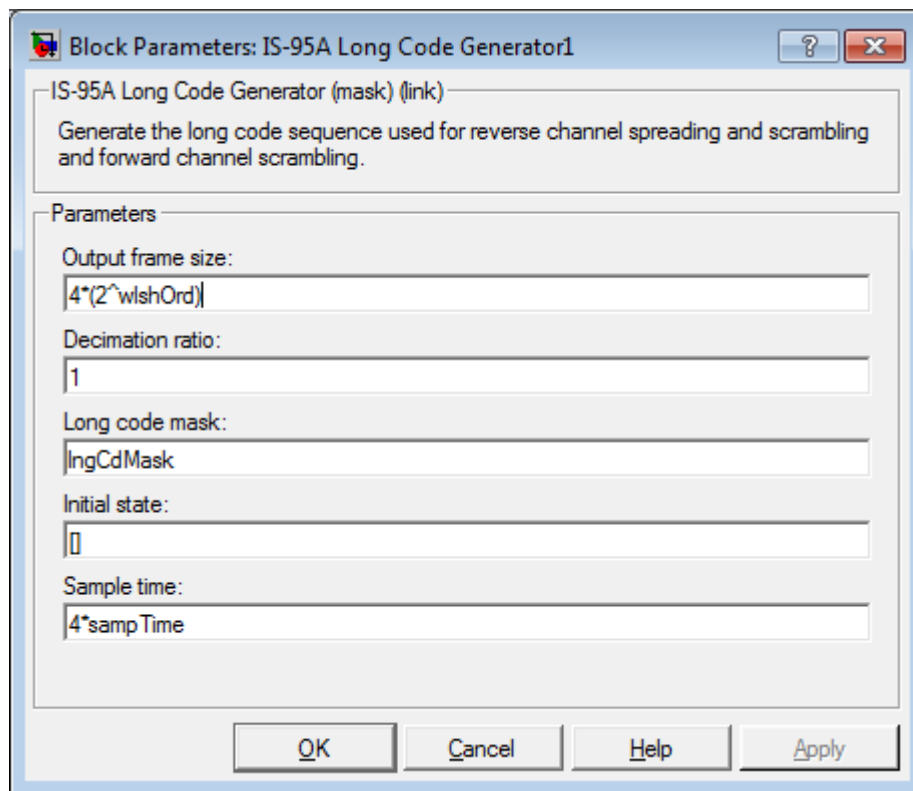


Figura 4.209 Parámetros del bloque IS-95A Long Code Generator.

Donde se define que salgan los datos en tramas de  $4 \times 64 = 256$  símbolos en un tiempo de  $4/1.2288 \times 10^6 = 1/307,2 \times 10^3$  segundos, lo cual asegura la tasa de transmisión de 1,2288 Mcps.

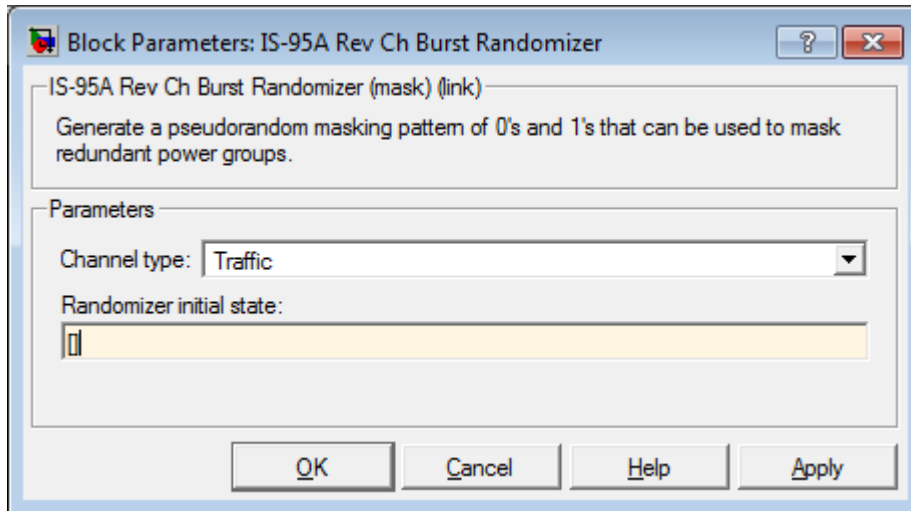


Figura 4.210 Configuración del bloque IS-95A Ch Burst Randomizer

### **IS-95A Rev Ch Burst Randomizer**

La transmisión del canal Reverse es cerrada, de manera que la transmisión en los casos de nonfull-rate es solo realizada para una fracción del tiempo: la mitad del tiempo para el canal half-rate, un cuarto para el quater-rate y un octavo del tiempo para el eighth-rate. Esta sincronización es hecha en intervalos de 1.25 ms conocidos como grupos de potencia (**PCGs**). Usando el código largo la ráfaga aleatorizada de datos se implementa un switch para cerrar la puerta a los grupos de potencia redundantes después del ensanchamiento de código largo.

En el caso del demo, como se utiliza un canal de tráfico Full Rate, Rate Set I, no se repiten símbolos, y por lo tanto no se necesita realizar eliminación de símbolos. Por lo tanto la salida del **IS-95A Rev Ch Data Burst Randomizer** debe ser solamente 1, para permitir el paso de todos los **PCGs**.

La salida del **IS-95A Rev Ch Data Burst Randomizer** se almacena en **burst**. Para 0.04 segundos se obtienen 49408 símbolos. Para saber si todos los símbolos son 1, se realiza la suma de todos los elementos almacenados en **burst**, tal como se muestra a continuación:

```

>> sum(burst)

ans =

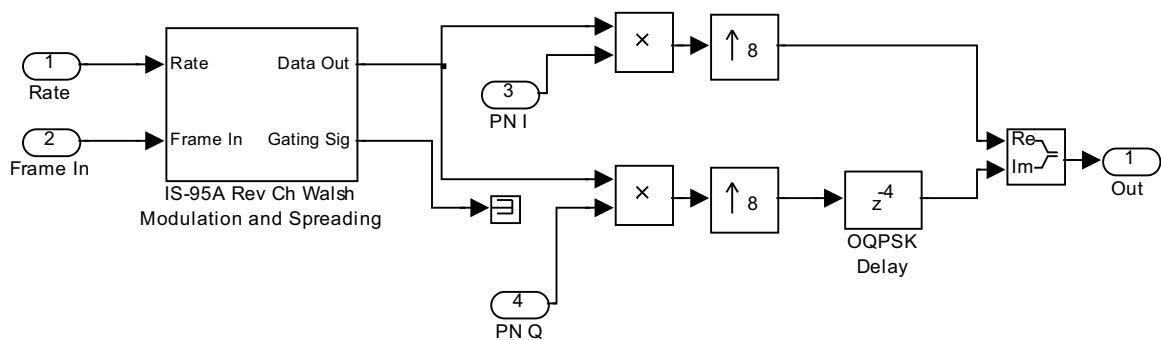
    49408

```

**Figura 4.211** Verificación de funcionamiento del bloque IS-95A Rev Ch Data Burst Randomizer

El resultado de la suma de los elementos de *burst* que se muestra en la figura es igual al número de elementos, lo que indica que efectivamente, todos los elementos son 1 y por lo tanto todos los *PCGs* están siendo transmitidos.

Luego de realizarse la modulación 64-ava, aleatorización de datos y el ensanchamiento por la secuencia PN larga, se realiza la multiplicación por el código PN corto:



**Figura 4.212** Multiplicación de datos modulados y ensanchados por código PN corto.

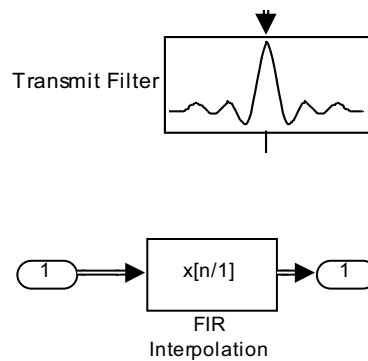
Donde se puede observar que se realiza la multiplicación se realiza entre tramas de 64 símbolos. El resultado de la multiplicación se interpola, para ser ingresado a un filtro FIR. También se puede observar que en la Secuencia PN corta Q, se ingresa un retardo de 4 elementos, que se traduce a un retardo de  $\frac{1}{2}$  chip para simular la modulación OQPSK.

Los datos son combinados en una señal imaginaria con el bloque *Real-Imag to Complex* para simular la modulación en cuadratura, ya que no se implementa moduladores ni demoduladores QPSK en los demos.

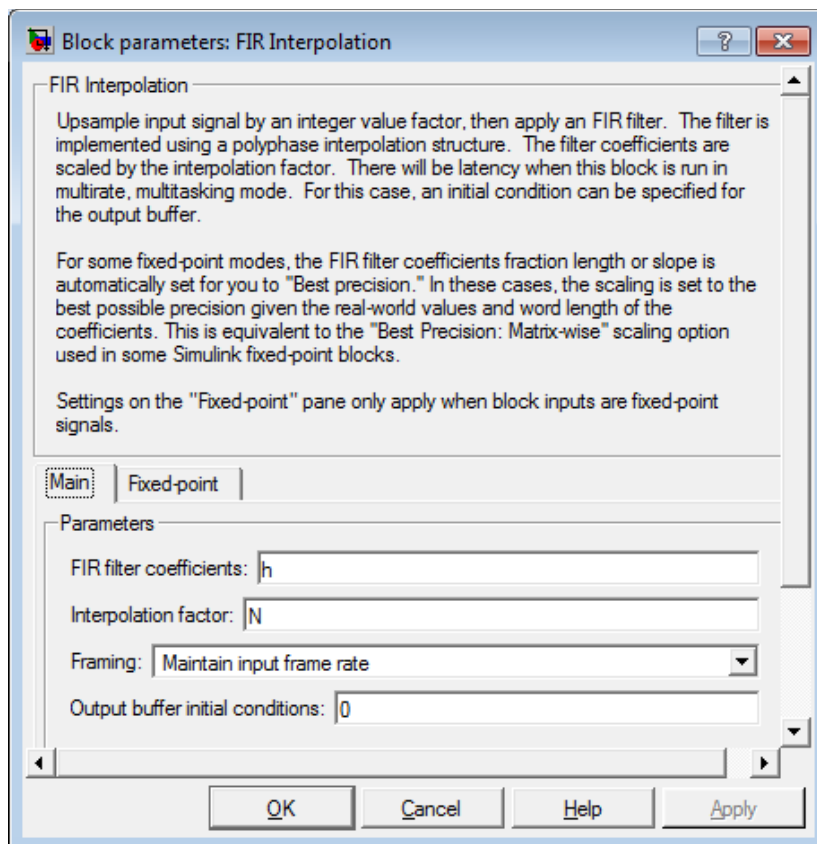
### Filtrado de la Señal

Para el filtrado de la señal, se utiliza un filtro FIR.

En el demo se define un factor de interpolación igual a 1, ya que se realizó previamente el sobremuestreo de las señales para simular el retraso de  $\frac{1}{2}$  chip de la señal PNQ (512 muestras para señal PNI y PNQ).

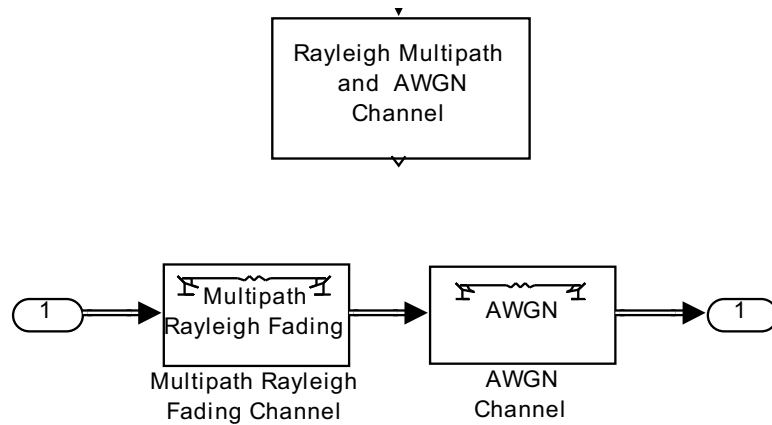


**Figura 4.213** Filtro de Transmisión



**Figura 4.214** Configuración del filtro

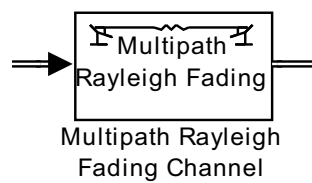
## Canal



**Figura 4.215** Simulación de un canal inalámbrico

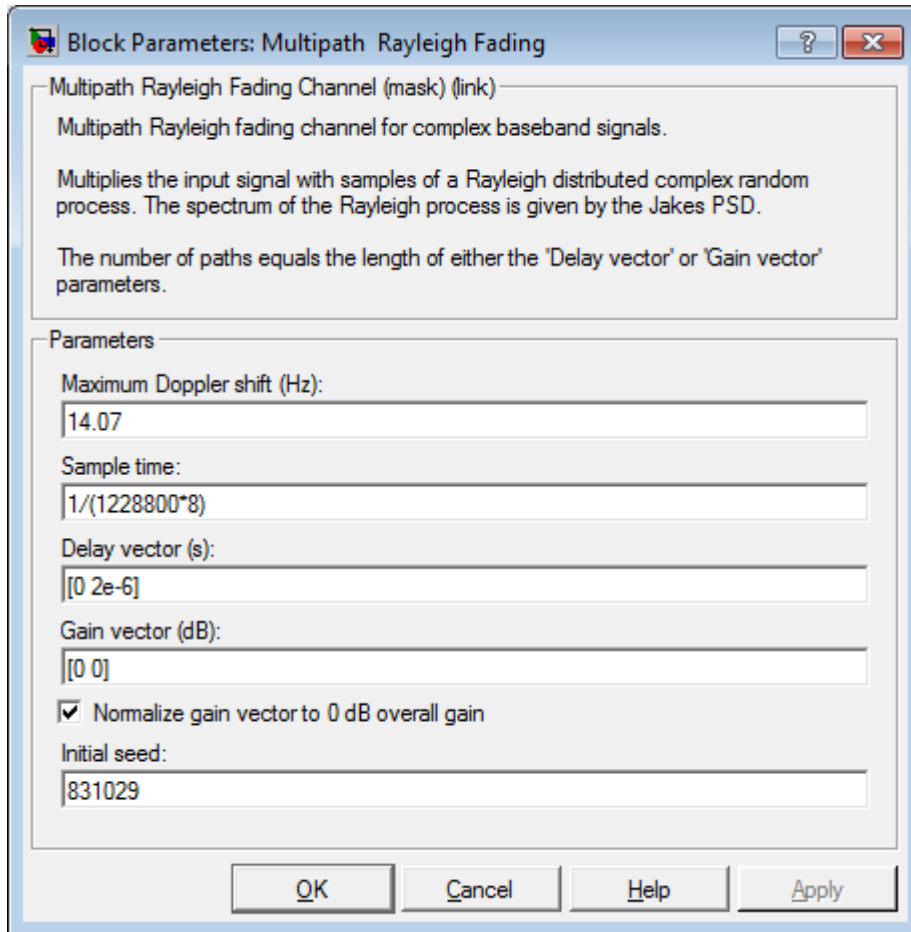
Para la simulación del canal, en el demo se ha utilizado tanto canal multiplicativo, como aditivo.

Para el canal multiplicativo se ha implementado el bloque **Multipath Rayleigh Fading**:



**Figura 4.216** Bloque Multipath Rayleigh Fading Channel





**Figura 4.217** Configuración del bloque Multipath Rayleigh Fading Channel

Se ingresa un desplazamiento de Doppler Máximo de frecuencia de 14.07 Hz. Se considera el tiempo de muestreo de  $1/1.2288 \times 8$  segundos debido a la tasa de transmisión de 1.2288 Mcps y el sobremuestreo realizado para cada señal. Con este bloque se generan replicas de la señal de transmisión, retrasadas y con cierta ganancia. Estas replicas son sumadas para simular un camino multitrayectoria.

En el parámetro **Delay vector(s)** se definen dos replicas: una con retraso igual a cero segundos y otra con retraso igual a  $2 \times 10^{-6}$  segundos.

Para realizar la conversión de los retardos a símbolos, se multiplica el retardo por la tasa de transmisión de 1.2288 Mcps y por el factor de sobremuestreo igual a 8 realizado en la modulación y ensanchamiento. Esto es igual a:

$$\text{Retardo de 0 segundos} = 0 \times 1.2288 \times 10^6 \times 8 = 0 \text{ símbolos de retardo}$$

Retardo de 2  $\mu$ segundos =  $2 \times 10^{-6} \times 1.2288 \times 10^6 \times 8 = 20$  *símbolos de retardo*

Con el parámetro **Gain vector(dB)** se definen las ganancias de las réplicas. En el demo se especifican las ganancias para las dos réplicas: 0 dB.

Para el canal aditivo se ha implementado ruido Gaussiano con el bloque **AWGN Channel**:

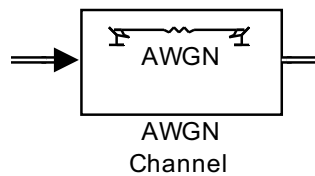


Figura 4.218 Bloque AWGN Channel

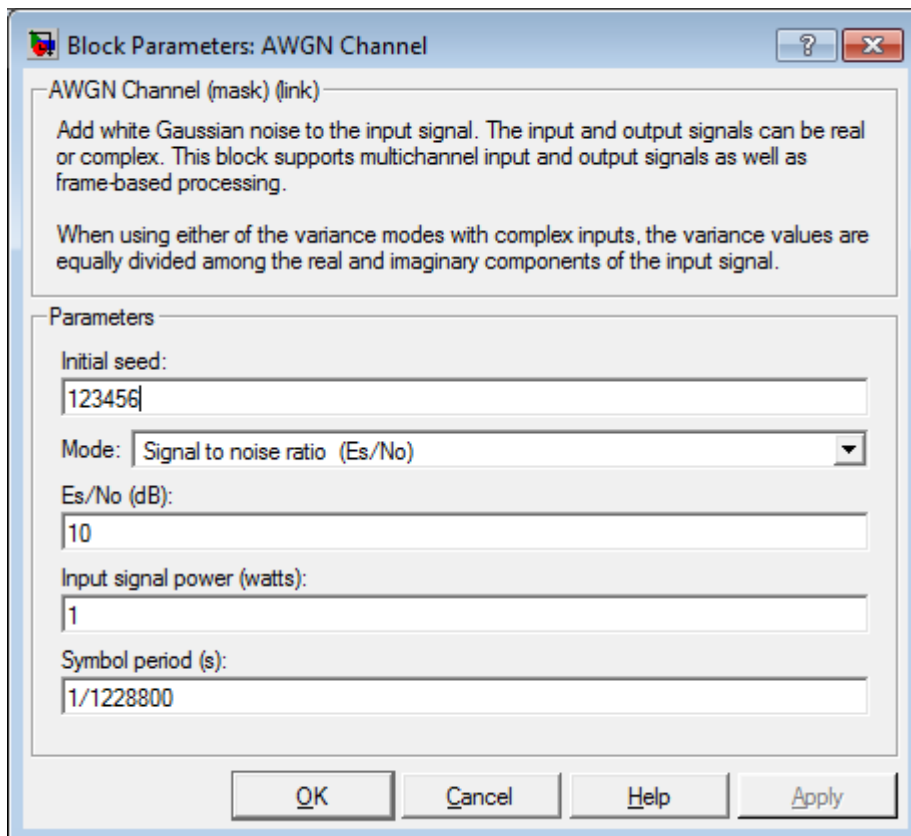
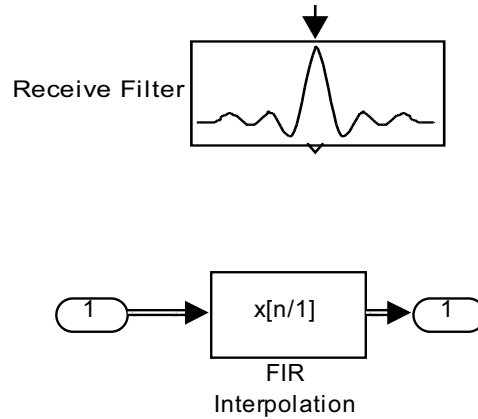


Figura 4.219 Configuración del bloque AWGN Channel

Se puede observar que la relación **Energía de la señal/ Ruido** es positiva, y por lo tanto, la potencia del ruido es menor a la señal DSSS filtrada. Se considera una

potencia total de transmisión de **1 Vatio**. Se especifica que el periodo de cada símbolo modulado es 1/1.228800 segundos de acuerdo al estándar **IS-95A**.

### Filtrado de Recepción



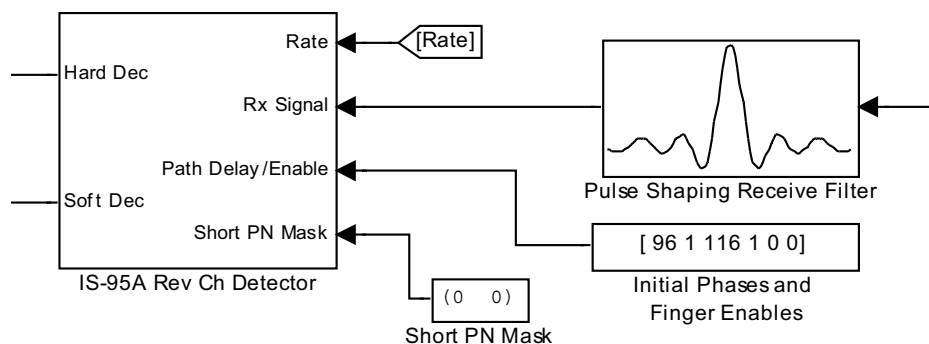
**Figura 4.220** Filtro de Recepción

En la recepción se utiliza el mismo filtro FIR implementado en el transmisor para recuperar la señal del Canal.

Cada filtro **FIR** ingresa un retardo 48 símbolos. Por lo tanto se genera un retardo por filtros igual a 96 símbolos.

### Detección de la Señal

Para realizar la detección de la señal se utiliza el bloque **IS-95A Rev Ch Detector**.



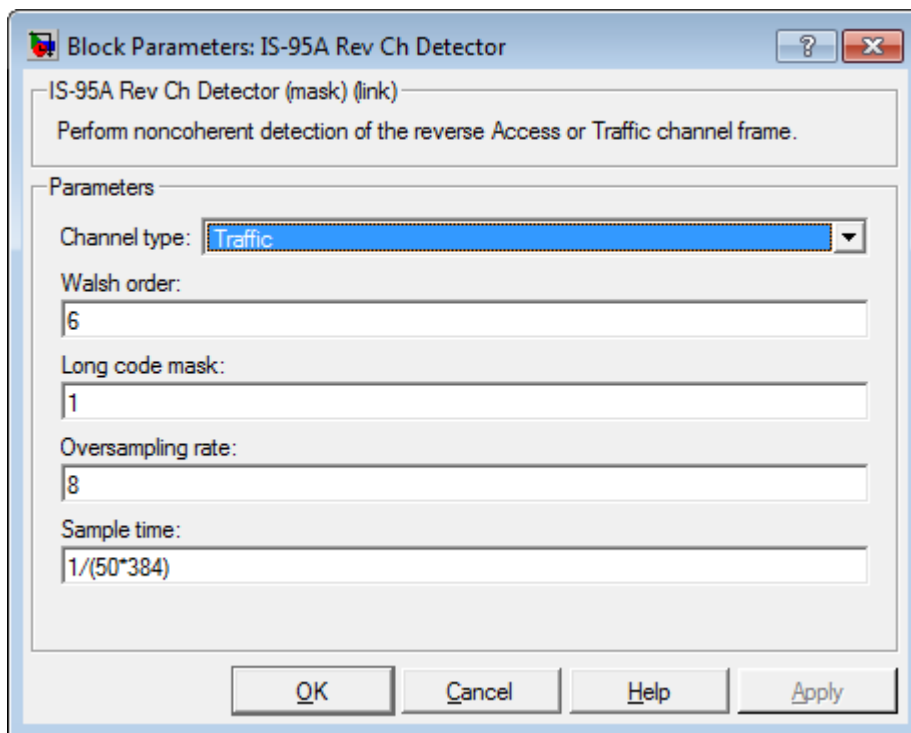
**Figura 4.221** Detección de la señal

Este bloque realiza la detección basado en el desensanchamiento y la detección no coherente de los datos de entrada durante el periodo de tiempo que la compuerta está abierta (gate-on).

Los datos en la salida del dedo Rake no coherente son alineados en el tiempo, como resultado, el receptor combina los datos de salida del dedo Rake antes del procedimiento de detección.

El receptor Rake soporta tres dedos y proporciona tanto decisiones duras como flexibles.

En los casos de tasas variables de tráfico, donde los datos son transmitidos en ráfagas, el bloque envía ceros en las ranuras de tiempo de compuerta cerrada (gate-off).

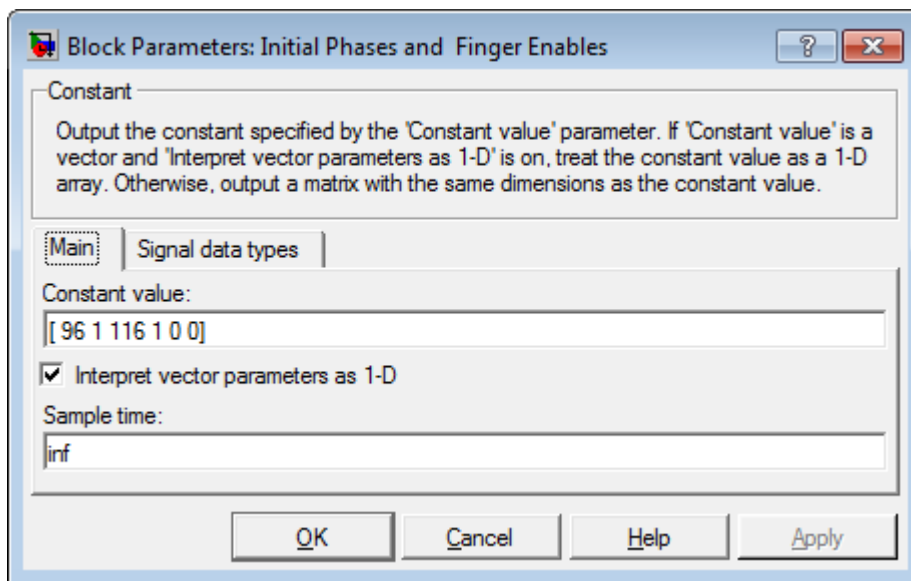


**Figura 4.222** Configuración del Bloque IS-95A Rev Ch Detector

Este bloque necesita como entradas, los bloques ***Initial Phases and Finger Enables*** y ***Short PN Mask***.

El bloque **Initial Phases and Finger Enables** es un vector constante de tamaño 6, que contiene un par de elementos para cada dedo del receptor Rake que se encuentra dentro del bloque **IS-95A Rev Ch Detector**. El primer elemento en cada par representa el offset PN inicial que es aplicado a cada dedo.

El segundo elemento en cada par sirve como señal de enable para el dedo Rake. Un valor de 0 deshabilita el dedo y un valor de 1 lo habilita. Para la generación de este vector se utiliza el bloque **Constant**:



**Figura 4.223** Configuración del bloque Initial Phases and Finger Enables

En el demo, el receptor Rake tiene habilitado solo 2 dedos, debido a que se producen dos componentes multitrayectoria. Por esto, el quinto y sexto elemento, los cuales definen el retraso y habilitación del tercer dedo Rake, tienen un valor de cero. El offset PN inicial es igual al retardo introducido por el canal multitrayectoria más el retardo introducido por los filtros de transmisión y recepción en símbolos.

$$\text{Offset dedo} = \text{retardo filtros} + \text{retardo canal multitrayectoria}$$

#### Ecuación 4.1

El offset para el primer dedo se calcula:

$$\text{Offset primer dedo} = 96 \text{ símbolos} + 0 \text{ símbolos} = 96 \text{ símbolos}$$

El offset para el segundo dedo se calcula:

$$\text{Offset segundo dedo} = 96 \text{ símbolos} + 20 \text{ símbolos} = 116 \text{ símbolos}$$

El bloque **Short PN Mask** es un vector constante de tamaño 2 que representa a las máscaras de las componentes de cuadratura y fase de la secuencia PN corta que sirven para especificar con qué fase se va a generar la secuencia. En **IS-95A** el offset se utiliza como un identificador de la estación base. En este caso se utiliza un offset igual a 0.

### Detección de Errores

Dentro del demo están implementados los bloques **Error Rate Calculation**. Para comprobar el funcionamiento del bloque.

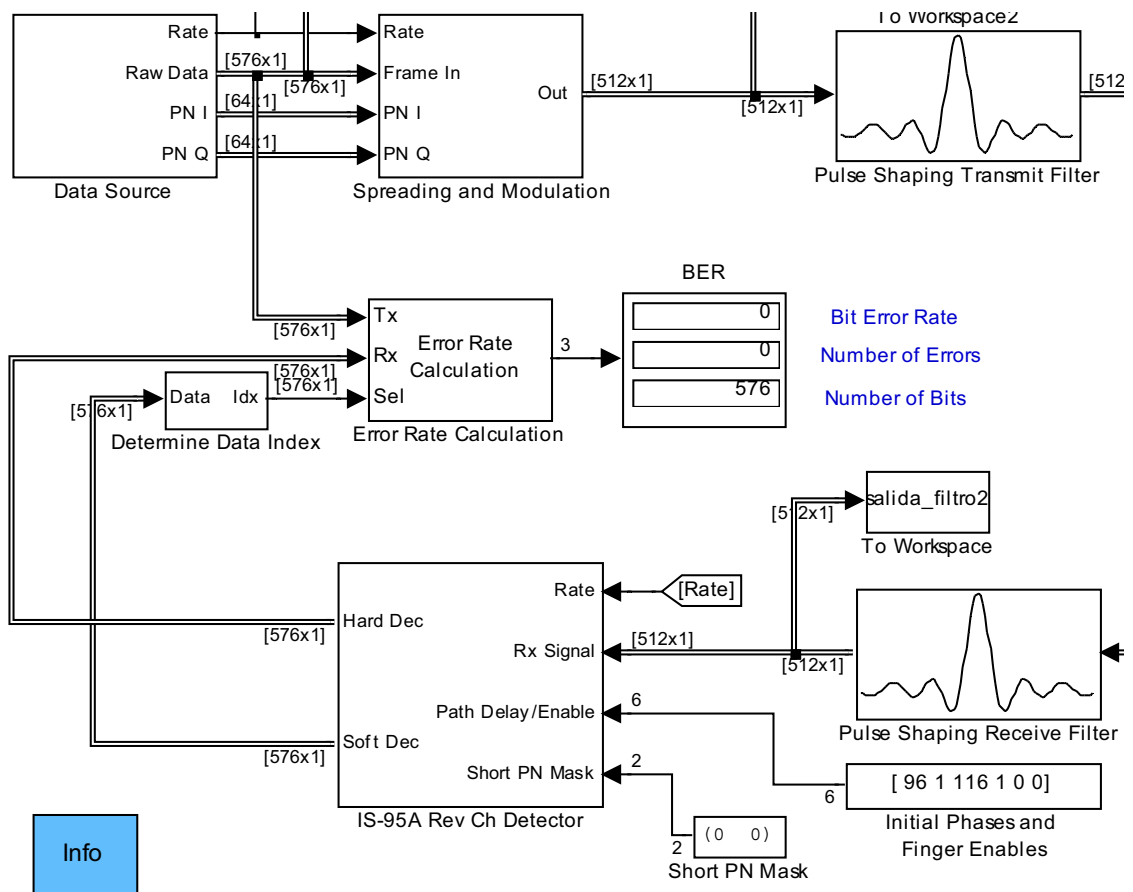
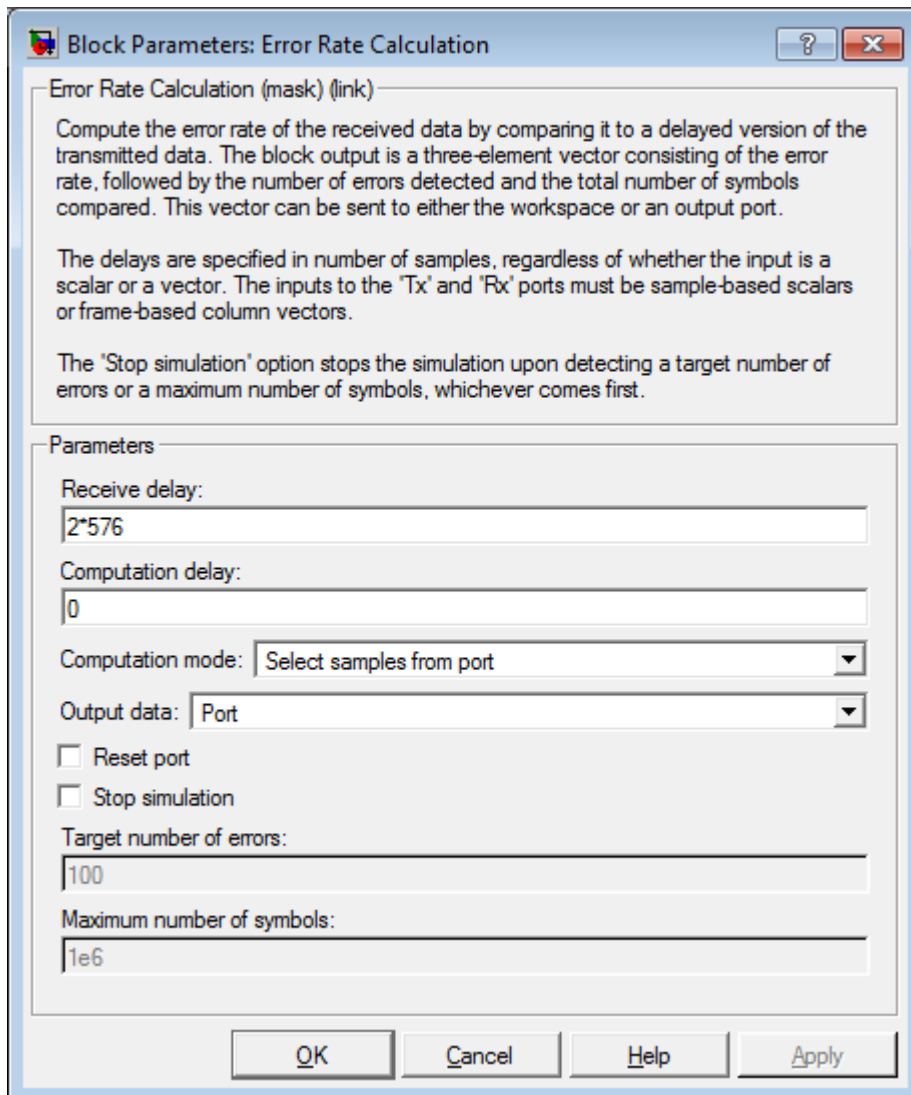


Figura 4.224 Detección de errores

El bloque **Error Rate Calculation** calcula el BER (Bit Error Rate) o tasa de errados. El bloque compara los bits relevantes de la señal del transmisor (576 bits) con los bits relevantes de la tercera trama recibida (las dos primeras tramas corresponden a un retardo).



**Figura 4.225** Configuración del bloque Error Rate Calculation

En el display se puede observar que se han comparado 576 bits relevantes del canal de tráfico y que no ha ocurrido ningún error.

Para visualizar la recepción de la primera trama sin errores, se utiliza la función **recupera\_texto**.

Para recuperar los datos se utiliza el bloque **To Workspace** con la variable **salida**:

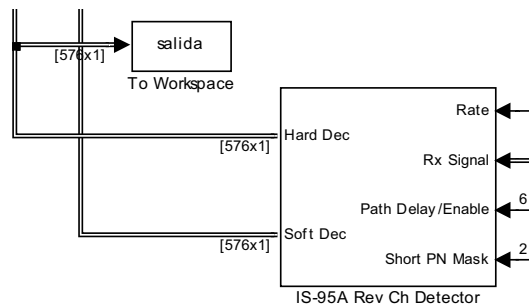


Figura 4.226 Captura de datos recibidos

Para recuperar el texto se analizan los últimos 576 elementos del vector **salida**:

```
>> a=salida(1153:1728)';
>> d=recupera_texto(a)

d =

*****Escuela Politecnica Nacional Analisis del Canal Reverse CDMA*****
```

Figura 4.227 Recuperación de texto

### Conclusión

En el demo del canal de tráfico reverso se comprueba el correcto funcionamiento según el estándar IS-95A. Con este demo se puede observar que el canal Reverse presenta algunas diferencias con respecto al canal Forward, por ejemplo los datos son modulados por la 64-ava modulación, la canalización y scrambling se realizan con la secuencia PN larga.

Como problema se presentó la eliminación de datos ( $268 - 172 = 96$  bits) generados para el canal Forward, debido a que solo se consideran relevantes a los primeros 172 bits generados de la trama de 268 bits. Para evitar este problema con la simulación, se realizó el programa **texto\_a\_bits** que genera 172 bits relevantes más 96 bits de relleno.



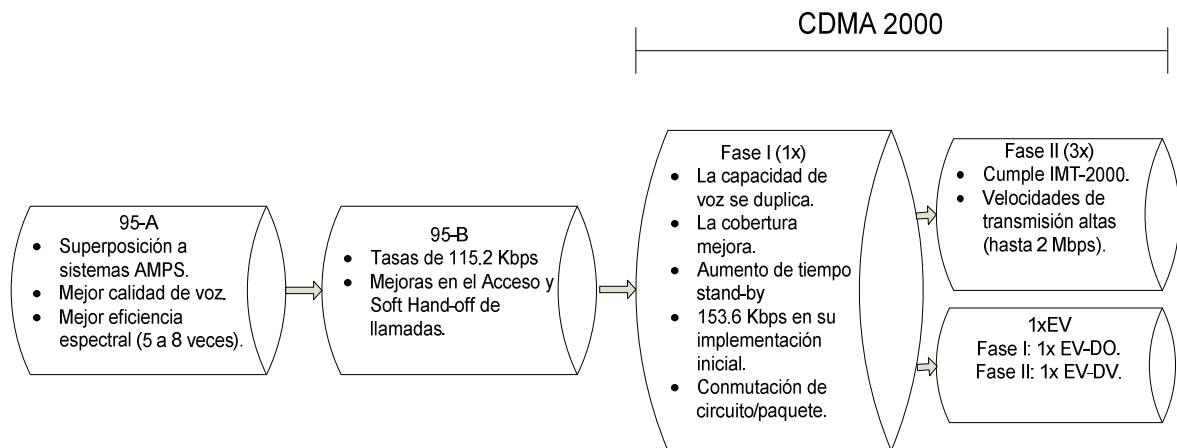
En ambos demos se presenta en el receptor un retardo correspondiente a 2 tramas, esto debido al procesamiento de los bloques detectores en los demos. Como se pudo observar, en un tiempo de simulación de 4 milisegundos se generan tres tramas, esto debido al comportamiento que Simulink presenta al comienzo de la simulación con datos generados en tramas y por tanto se genera una trama de más. Debido al retardo mencionado antes, solo se puede recuperar la primera trama enviada por el transmisor en ambos demos.

## **4.3 EVOLUCIÓN DE CDMA EN LAS COMUNICACIONES MÓVILES**

### **4.3.1 CDMA/CDMAONE**

El camino evolutivo de esta tecnología empieza con la propuesta de Qualcomm de un nuevo sistema basado en técnicas de espectro ensanchado. Esta propuesta, que luego fue estandarizada como IS-95, es el primer sistema móvil en desarrollo comercial. CdmaOne es un nombre comercial de marca registrada, reservado para uso exclusivo de las empresas que son miembros de CDG (CDMA Development Group). El mismo describe un sistema inalámbrico completo que incorpora la interfaz aérea IS-95 CDMA y la norma de la red ANSI-41 para la interconexión por conmutación, además de muchas otras normas que integran el sistema inalámbrico completo. A este sistema se le caracteriza por su alta capacidad y celdas de radio pequeño, que emplea espectro extendido y un esquema de codificación especial y, lo mejor de todo es muy eficiente en potencia. El acceso de multiplexación por división de códigos de banda estrecha IS-95 estipula un espaciamiento de portadora de 1.25MHz para servicios de telefonía. La Telecommunications Industry Association (TIA) empezó a definir esta especificación en 1991. En julio de 1993, la TIA aprobó el estándar CDMA IS-95.

En la figura se muestra el camino evolutivo que tiene que seguir CDMA.



**Figura 4.228** Evolución de CDMA.

#### 4.3.1.1 CdmaOne/IS-95-A

Los sistemas IS-95A, introducidos en un principio en 1995, proporcionaban datos con conmutación de circuitos a 9,6 Kbps, así como servicios vocales más avanzados que los prestados por los sistemas de primera generación existentes. Posteriormente la tecnología CdmaOne / IS-95-A ofrece soporte a señales de voz conmutados por circuitos y datos (conmutados por circuitos o datos), con velocidades de hasta 14,4kbps. Debido al enfoque inicial de proveedores y operadoras en señales de voz. Históricamente la CdmaOne/IS-95-A ha sido utilizada sólo para voz conmutada por circuitos y para un pequeño volumen de datos conmutados por circuitos.

#### 4.3.1.2 cdmaOne/IS-95-B

Con el advenimiento de los sistemas ANSI-95 B en 1998, los operadores fueron entonces capaces de proporcionar datos con conmutación de circuitos a 64 Kbps. La tecnología CdmaOne/IS-95-B ofrece soporte a señales de voz conmutados por circuitos y datos, conmutados por paquetes. En teoría, ella provee tasas de datos de hasta 115kbps, y alcanza, generalmente, valores prácticos de 64kbps.

Aunque los sistemas ANSI-95 A/B (colectivamente conocidos como sistemas CdmaOne) proporcionan servicios de voz y datos, quedan limitados a aplicaciones con bajas velocidades binarias. Los sistemas CDMA2000 se han

desarrollado basándose en los requisitos señalados en el mandato para las IMT-2000 y son el resultado directo de la evolución de las redes IS-95 A/B.

#### **4.3.2 ESTÁNDAR CDMA2000**

El primer estándar también conocido como IS-136 de tecnología inalámbrica de tercera generación (3G) disponible comercialmente. Identifica la norma TIA para tecnología de tercera generación, que es un resultado evolutivo de CdmaOne, el cual ofrece a los operadores que han desplegado un sistema CdmaOne de segunda generación, una migración transparente que respalda económicamente la actualización a las características y servicios 3G, dentro de las asignaciones del espectro actual, tanto para los operadores celulares como los de PCS. Este estándar sentó las bases para la transmisión de datos de alta velocidad disponible en la actualidad en muchos mercados del mundo y ofrece a los consumidores y profesionales conectividad inalámbrica sin interferencias.

La interfaz de red definida para cdma2000 apoya la red de segunda generación de todos los operadores actuales, independientemente de la tecnología: CdmaOne, IS-136 TDMA o GSM). La TIA ha presentado esta norma ante la ITU como parte del proceso IMT-2000 3G. CDMA2000 permite también desplegar una red IMT-2000 en fases sucesivas, dependiendo de la banda de frecuencias de la que disponga el operador y de la evolución de red que haga necesaria la demanda de servicios de datos a gran velocidad.

A fin de facilitar la migración de CdmaOne a las capacidades de cdma2000, ofreciendo características avanzadas en el mercado de una manera flexible y oportuna, su implementación se ha dividido en dos fases evolutivas.

##### **4.3.2.1 CDMA2000 Fase I: CDMA 1x**

Las capacidades de la primera fase se han definido en una norma conocida como 1XRTT. La publicación de la 1XRTT se hizo en el primer trimestre de 1999. Esta norma introduce datos en paquetes a 144 Kbps en un entorno móvil y a mayor velocidad en un entorno fijo. Las características disponibles con 1XRTT representan un incremento doble, tanto en la capacidad para voz como en el

tiempo de operación en espera, así como una capacidad de datos de más de 300 Kbps y servicios avanzados de datos en paquetes.

Adicionalmente extiende considerablemente la duración de la pila y contiene una tecnología mejorada en el modo inactivo. Se ofrecerán todas estas capacidades en un canal existente de 1.25 MHz de CdmaOne.

#### **4.3.2.2 CDMA2000 Fase II: CDMA 3x**

La evolución de CdmaOne, hasta llegar a las capacidades completas de cdma2000, continuará en la segunda fase e incorporará las capacidades de 1XRTT, usará tres portadoras de 1,25 MHz en un sistema multiportadora para prestar servicios de banda ancha de 3G. Cdma 3XRTT utiliza un par de canales de 3,75-MHz (p.ej., 3 X 1,25 MHz) para alcanzar mayores velocidades de datos. La versión 3x de CDMA2000 es algunas veces referidas como Multi-Carrier o MC-te.

Cdma 3XRTT proporcionará velocidad de circuitos y datos en paquete de hasta 2 Mbps, incorporará capacidades avanzadas de multimedia e incluirá una estructura para los servicios de voz y codificadores de voz 3G, entre los que figuran los datos de paquetes de "voice over" y de circuitos.

#### **4.3.3 CDMA2000 1xEV**

Es la evolución directa del estándar de tecnología inalámbrica CDMA2000 de tercera generación (3G), y ofrece una conectividad inalámbrica de alta velocidad similar a la de la banda ancha por cable.

La evolución de la CDMA2000 1x tiene la denominación de CDMA2000 1xEV. La 1xEV se implementará en etapas: 1xEV-DO y 1xEV-DV. 1xEV-DO significa "1x Evolution Data Only" (1x datos de evolución solamente). 1xEV-DV significa "1x Evolution Data and Voice" (1x datos y voz de evolución). Ambas etapas de evolución 1xEV CDMA2000 emplearán una portadora normal de 1,25 MHz carrier. El sistema 1xEV mejora la velocidad de procesamiento de datos, obteniendo

velocidades máximas de 2Mbps, sin tener que utilizar más de 1,25 MHz del espectro.

Los requisitos para los operadores recién establecidos con respecto a 1xEV establecen dos fases.

#### **4.3.3.1 CDMA2000 1xEV-DO**

En la primera fase *Cdma2000 1xEV-DO* es un mejoramiento de la tecnología CDMA2000 que se ha optimizado esencialmente para ofrecer servicios de datos y permite la transmisión de datos a velocidades más elevadas. La interfaz inalámbrica CDMA2000 1xEV-DO tiene por objeto proporcionar una plena interoperabilidad con la red CDMA2000 1x. En la revisión, Rev A, CDMA2000 1xEV-DO soporta una velocidad de datos en el enlace de bajada (forward link) de hasta 3,1 Mbps y una velocidad de datos en el enlace de subida (reverse link) de hasta 1,8 Mbps en un canal de radio dedicado a transportar paquetes de datos de alta velocidad.

En la revisión Rev B, soporta una velocidad de datos en el enlace de bajada de hasta 73 Mbps y una velocidad de datos en el enlace de subida (reverse link) de hasta 27 Mbps. En la revisión Rev C, soporta una velocidad de datos en el enlace de bajada de hasta 260 Mbps y una velocidad de datos en el enlace de subida (reverse link) de hasta 70 Mbps.

CDMA2000 1xEV-DO brinda a los operadores una opción económica para suministrar una amplia gama de servicios de datos IMT-2000 a precios abordables. Los sistemas 1xEV-DO, que se han desplegado ya comercialmente, implementan muchas características avanzadas del diseño de sistemas inalámbricos de telecomunicaciones. CDMA2000 1xEV-DO incorpora una velocidad variable adaptable en el enlace ascendente con multiplexación por división en el tiempo (TDM) que maximiza las velocidades de datos del usuario, atribuyendo toda la potencia de los transceptores de estación de base (BTS) a un usuario en un momento dado. La implementación eficiente de la programación sensible a los canales y la diversidad multiusuario eficaz permiten obtener las velocidades de datos más elevadas en determinados momentos.

La evolución hacia CDMA2000 1xEV-DO constituye un marco flexible para ofrecer servicios de datos de calidad, gracias a una amplia gama de velocidades de datos y tipos de paquetes. Los protocolos se han diseñado para proporcionar traspasos virtuales sin interfaz en una zona de servicio para servicios de datos en paquetes, así como interoperabilidad sin interfaces con el enlace inalámbrico de CDMA2000 1X. La provisión de un canal para efectuar una rápida radio búsqueda mejora significativamente el tiempo de espera.

Este sistema encabeza la convergencia de las tecnologías inalámbricas con los productos electrónicos de uso personal, creando una nueva generación de dispositivos inalámbricos multifunción.

#### **4.3.3.2 CDMA2000 1x EV-DV**

En la segunda fase, Cdma2000 1x EV-DV se centra en las funciones de datos y de voz en tiempo real, así como en la mejora del funcionamiento para mayor eficiencia en voz y en datos. La tecnología Cdma2000 1xEV-DV es un mejoramiento de los sistemas de multiportadora CDMA IMT-2000, pues combina las características de los sistemas CDMA2000 1X y CDMA2000 1xEV-DO. Así pues, constituye una opción por la que puede optarse para proporcionar la elevada capacidad vocal de los sistemas CDMA2000 1X o la gran capacidad de datos de los sistemas CDMA2000 1xEV-DO, o incluso una combinación equilibrada de elevada capacidad de voz y datos en una sola portadora de 1,25 MHz.

Los sistemas CDMA2000 1xEV-DV combinan óptimamente la voz y los datos, aumentando así la capacidad para incrementar aún más las velocidades de datos. Esto se logra incorporando un canal de datos en paquetes a gran velocidad que utilicen modulación y codificación dinámicas basadas en las condiciones del canal, retransmisiones rápidas y eficientes y mecanismos dinámicos de asignación de recursos.

Los teléfonos móviles con capacidad CDMA2000 1xEV-DV están en condiciones de entregar datos a los sistemas más antiguos y tener una compatibilidad con CDMA2000 1x.

## CAPÍTULO V

### 5. CONCLUSIONES Y RECOMENDACIONES.

#### 5.1 CONCLUSIONES.

- Las técnicas de Acceso al medio son importantes en las comunicaciones, especialmente en comunicaciones inalámbricas, debido a que permite establecer múltiples transmisiones de voz, datos y multimedia a través de un medio compartido.
- CDMA permite obtener características esenciales de un sistema de comunicaciones como Indetectabilidad, Seguridad, y Acceso Multiusuario.
- Indetectabilidad: CDMA se basa en la técnica de Espectro Ensanchado DSSS donde se asigna códigos a los usuarios del sistema. Para formar la señal DSSS se multiplican los datos del usuario por el código respectivo. La tasa del código asignado es mayor a la tasa de datos, produciendo una expansión del espectro de potencia.
- Lo que se busca con el ensanchamiento del espectro, es que éste se parezca al ruido (tenga un espectro plano). Para esto, es necesario que la densidad espectral de la señal DSSS disminuya, de tal forma que se confunda con el ruido.
- Los Postulados de Golomb permiten evaluar a los códigos en aspectos de aleatoriedad, para que puedan ser utilizados en un sistema CDMA.
- Primero se buscaron códigos de naturaleza Pseudoaleatoria, para de alguna forma poder reproducirlos en la recepción y recuperar la señal.

- El esquema de generación más utilizado es el de los generadores LFSRs.
- Las estructuras de LFSRs que generan secuencias que cumplen los postulados de Golomb, están asociadas a Polinomios Primitivos en  $GF(2)$ . Estas secuencias a más de cumplir los postulados de Golomb, tienen la propiedad de poseer **periodo máximo**. A estas secuencias se las conoce como Secuencias de Longitud Máxima o M-secuencias.
- La importancia de que una secuencia tenga un periodo máximo radica en que a mayor longitud del código usado en CDMA, el espectro tenderá a ser más plano y será menos probable que el código sea obtenido por un usuario no autorizado. Por lo tanto, se provee mayor **seguridad** a la comunicación; puesto que, solo aquel receptor que genere el código único de un usuario, podrá recuperar la información de este.
- Para que CDMA sea considerado como una **técnica de acceso al medio**, debe proveer una gran cantidad de códigos que generen la menor interferencia entre sí.
- La interferencia entre usuarios, se evalúa por medio de la función de correlación cruzada y Autocorrelación. Mientras menor sea la correlación cruzada entre los códigos y mayor sea la autocorrelación, la detección en el receptor se realizará con mayor facilidad y la tasa del BER disminuirá.
- Las M-secuencias no son aptas para un sistema de comunicación multiusuario, debido a que no existen en un gran número. A pesar que la Autocorrelación es alta y presentan características de ruido, la correlación cruzada entre las m-secuencias muestra grandes valores comparables a la Autocorrelación.
- Sin embargo, dentro de las m-secuencias existen pares de secuencias que presentan tres valores de Correlación Cruzada baja, que son la base para la generación de nuevas familias de secuencias con alta Autocorrelación y baja



correlación cruzada. Al par de secuencias que cumplen con este criterio se las conoce como **Secuencias Preferidas**.

- Las Secuencias Gold son una familia de secuencias que presentan una alta función de Autocorrelación y tres valores bajos de correlación cruzada. Se generan a partir de un par de secuencias preferidas.
- Otra familia de secuencias con buenas características de Autocorrelación y Correlación Cruzada son las secuencias Kasami.
- Las familias de Secuencias Kasami son más grandes que las familias de secuencias de Gold. Y forman a partir del proceso de decimación de secuencias.
- Las secuencias más utilizadas en sistemas de múltiple acceso son las secuencias de Gold, puesto que su implementación en hardware es más sencillo que en las Secuencias Kasami.
- Tanto las secuencias Kasami como las secuencias Gold son utilizados en esquemas asincrónicos.
- Para sistemas sincrónicos se utilizan secuencias ortogonales como Walsh y OVSF.
- En estricto sincronismo, la Autocorrelación de estas secuencias es máxima, mientras que la Correlación Cruzada es cero. Por lo tanto presentan características ideales para un sistema de comunicación multiusuario.
- Las secuencias Walsh son palabras de código fijo utilizado principalmente en IS-95A.

- OVSF son palabras de código de longitud variable, desarrollados para la transmisión de datos a velocidades variables. Se utilizan en estándares como CDMA 2000 y WCDMA, donde existen aplicaciones multimedia.
- Para aplicaciones donde no se pueda tener una sincronización adecuada, es importante generar secuencias que tengan una **autocorrelación maximizada**, como son las secuencias de **Barker**.
- Aunque en CDMA no se utilizan este tipo de secuencias, por su limitada longitud y número, son importantes para otro tipo de aplicaciones como: preámbulo para la sincronización de tramas en detección de datos de información, aplicaciones de radar y Wlan.
- En cuanto a los sistemas de Espectro Ensanchado, existen dos esquemas: DSSS y FHSS.
- En los sistemas DSSS, el ensanchamiento del espectro se realiza por medio de la multiplicación entre la información y una secuencia que posee una mayor tasa de transmisión.
- La relación entre la tasa de la secuencia (tiempo de chip) y la tasa de la información (tiempo de bit), se conoce como ganancia de procesado. Este indica el factor de ensanchamiento de la señal.
- Los sistemas DSSS presentan una gran resistencia frente al ruido y la interferencia debido al uso de códigos con propiedades especiales de correlación cruzada. Pero, a medida que se incrementa el número de usuarios o señales interferentes, el desempeño del sistema se degrada, debido a que el receptor considera a las señales de otros usuarios como ruido.
- Los sistemas FHSS siguen manteniendo la idea de pseudoaleatoriedad, solo que se aplica a la frecuencia portadora, donde esta va saltando entre canales de un ancho de banda expandido disponible. Por lo tanto, el ensanchamiento

espectral se produce por todos los saltos espectrales que realiza la información en comparación a su ancho de banda original.

- Los sistemas FHSS presentan mayor resistencia a la presencia de señales interferentes, debido a los saltos que se realizan y al gran ancho de banda que se utiliza.
- FHSS es menos resistente a la presencia del ruido, debido a que este está presente en todas las frecuencias.
- Los sistemas DSSS son preferidos en lugar de los FHSS, por su simplicidad de implementación y altas tasas de transmisión.
- La capacidad de los sistemas de espectro ensanchado está limitada por la presencia de señales interferentes en un ambiente multiusuario. Esta interferencia es conocida como MAI (Multiple Access Interference). Para aumentar esta capacidad se han desarrollado Esquemas de Detección Multiusuario.
- Los detectores óptimos buscan una secuencia que maximice la probabilidad condicional para estimar la secuencia transmitida. Estos detectores tienen un desempeño excelente, pero su implementación práctica es muy compleja. Debido a esto aparecen una nueva clase de detectores, llamados detectores subóptimos.
- Dentro de los detectores subóptimos se encuentran el detector correlador y el detector MMSE.
- Con el detector correlador se elimina completamente el MAI, pero incrementa el ruido ambiental. El desempeño del detector correlador depende de la correlación cruzada entre códigos. A medida que la correlación cruzada crece, el desempeño del detector se degrada.

- El detector MMSE lineal está diseñado para minimizar el efecto total del MAI y el ruido ambiental a la salida del detector.
- Los detectores óptimos y subóptimos implican mayor complejidad, tanto en cálculos como en implementación práctica. Debido a esto se han implementado esquemas de combate a interferencia. Estos esquemas se basa en la diversidad de antenas.
- Implementar los esquemas de combate a interferencia en las estaciones móviles resulta caro y que sus receptores sean más grandes. Debido a esto solo se implementan en las estaciones base, ya que estas proveen servicio a múltiples móviles.
- Las técnicas de cancelación es otro esquema de detección multiusuario y están basadas en el principio de que es posible remover la interferencia de múltiple acceso de cada señal recibida de los usuarios antes de tomar decisiones de datos.
- Existen dos técnicas de cancelación de interferencia: cancelación sucesiva y cancelación en paralelo.
- Con la técnica de cancelación sucesiva se elimina por etapas la interferencia de cada usuario, tomando como referencia a la señal del usuario con mayor potencia en cada etapa.
- Con la técnica de cancelación en paralelo, se elimina la interferencia de todos los usuarios simultáneamente.
- Dentro del desarrollo de las comunicaciones móviles, la técnica de acceso al medio CDMA ha permitido realizar muchos avances. Una de las primeras aplicaciones a comunicaciones móviles de segunda generación es el estándar IS-95A.

- El estándar utiliza tres tipos de código tales como: un código PN corto, una secuencia PN larga y los códigos Walsh, que combinados generan un código único para cada usuario.
- Las secuencias PN cortas se utilizan como identificadores de la estación Base. A cada estación base dentro de la red IS-95A se le asigna una versión desplazada del código PN corto. Cada offset es de 64 chips, lo cual garantiza el mayor número de estaciones base dentro de la red, generando la menor interferencia entre ellas. Este offset se produce a través de un enmascaramiento.
- El estándar define canales lógicos para el enlace Forward y para el enlace Reverse. También define esquemas para codificación del canal para mitigar los efectos de desvanecimiento y ruido del canal inalámbrico como: CRC, codificador convolucional, repetidor e interleaver cuyas características son únicas para el estándar
- La secuencia PN larga se utiliza en el enlace Forward para codificar los datos e insertar los bits de control de potencia (PCBs). Estos PCBs son utilizados en el control de potencia de lazo cerrado de IS-95A. mientras que en el canal Reverse la secuencia PN larga se utiliza para realizar la identificación del usuario.
- La mayor diferencia en la estructura del enlace Forward y Reverse se encuentra en la utilización de los códigos de Walsh. En el enlace Forward como identificador de los canales lógicos, mientras que en el canal Reverse son utilizados para realizar la 64-ava modulación. El enlace Forward utiliza una modulación QPSK, mientras que el enlace Reverse utiliza una modulación OQPSK, ya que el enlace Reverse necesita de mayor protección al ser asincrónico.
- Para la recepción de las señales se utiliza un receptor Rake, tanto en la estación base como en el móvil. Un receptor Rake consiste de un banco de

correladores que aprovechan las componentes Multicamino generadas en el canal inalámbrico para combinarlas y estimar la señal recibida.

### CDMA vs GSM

- Al finalizar el presente proyecto de titulación se pudo verificar que CDMAOne es un sistema robusto para comunicaciones móviles.
- A pesar de presentar mejores ventajas, GSM es la tecnología 2G que más se ha expandido en el mundo, debido principalmente a que es una tecnología libre. Todos los sistemas CDMA, al ser patentes de la compañía Qualcomm, están sujetos a licencias para su explotación y esto encarece su costo de implementación para los diferentes carriers.
- Los sistemas basados en CDMA ofrecen mayores tasas de transmisión debido al uso eficiente del espectro. Un sistema CDMA se puede implementar en una banda de 1.25 MHz, el cual es asignado a cada usuario.
- Para que redes GSM de nueva generación, alcancen altas tasas de transmisión de datos, necesitan utilizar como técnica de acceso al medio a WCDMA, la misma utiliza una banda de frecuencia de 5 MHz y se basa en CDMA.

CDMA	GSM
<ul style="list-style-type: none"> <li>• CDMA ofrece mejor codificación digital de la voz.</li> <li>• Seguridad en las comunicaciones.</li> <li>• Robusto frente a interferencias.</li> <li>• Mayores tasas de datos.</li> <li>• Requiere conexión dedicada por la transmisión de datos</li> <li>• Handoff suave (soft handoff)</li> </ul>	<ul style="list-style-type: none"> <li>• Tecnología abierta.</li> <li>• Mayor cantidad de Carriers.</li> <li>• Mayor facilidad de Roaming Internacional.</li> <li>• Utilización del móvil en cualquier país donde exista GSM mediante la SIM.</li> <li>• Transmisión de datos por paquetes.</li> </ul>

<ul style="list-style-type: none"> <li>• CDMA resulta más costoso por la cantidad de licencias que este necesita por parte de Qualcomm.</li> <li>• Los sistemas 3G están basados en CDMA.</li> <li>• Uso eficiente del espectro.</li> </ul>	<ul style="list-style-type: none"> <li>• Uso menos eficiente del espectro.</li> <li>• Para que redes GSM de nueva generación, alcancen altas tasas de transmisión de datos, necesitan utilizar como técnica de acceso al medio a WCDMA, diferenciándose de CDMA en el uso de un espectro de 5 MHz.</li> <li>• Con la tecnología EDGE como medio de acceso, no se alcanzan tan altas tasas de transmisión como con WCDMA.</li> </ul>
---	---

## 5.2 RECOMENDACIONES.

- Se recomienda el uso de la función `xcorrelacion` para realizar el análisis de autocorrelación y correlación cruzada de señales periódicas, debido a que la función `xcorr` de Matlab no entrega los resultados deseados.
- 
- Tomar en consideración los parámetros de ingresos de los programas desarrollados en el capítulo 2.
- 
- El ingreso de los polonios generadores deben ser en forma vectorial ascendente.
- 
- De igual forma, si se requiere de modificaciones a los programas desarrollados, se debe tomar en cuenta los parámetros necesarios para la utilización de funciones de Matlab, y las funciones desarrolladas en el proyecto de titulación.
- 
- Matlab 7.0 posee en Simulink una librería referente al estándar IS-95A, CDMA Reference Blockset. Si se desea realizar un modelo con los bloques de esta librería, se debe tomar en cuenta la configuración de los mismos.

- Además hay que tomar en cuenta que todos los bloques trabajan en referencia a la máxima tasa de transmisión definida para IS-95A. (Rate Set 2, Full Rate).
- Para distintos canales y velocidades, los bloques realizan un algoritmo distinto. Por lo que se deben tomar en cuenta los bits relevantes de las tramas de información.
- Para entender la simulación de los demos de IS-95A es necesario tener un conocimiento previo de los conceptos del estándar, como los esquemas de codificación y decodificación del canal, tasas de transmisión, secuencias aplicadas, modulación y receptor Rake.
- Con el Blockset de Comunicaciones de Simulink se puede simular cualquier sistema de comunicaciones.
- Para el correcto funcionamiento de las interfaces gráficas realizadas en los capítulos, deben ser tomadas en cuenta las consideradas descritas en los capítulos respectivos.



## REFERENCIAS BIBLIOGRAFICAS

- Yang, Samuel C.** *CDMA RF System Engineering*. Londres : Artech House, 1998.
- Lee, Jhong S. y Miller, Leonard E.** *CDMA Systems Engineering Handbook*. Londres : Artech House Publishers, 1998.
- Abu-Rgheff, Mosa Ali.** *Introduction to CDMA Wireless Communications*. Amsterdam, Londres : ELSEVIER, 2007. págs. 107-113.
- Chen, Hsiao-Hwa.** *The Next Generation CDMA Technologies*. Inglaterra : Jhon Wiley & Sons, Ltd, 2007.
- PhD Alvarado Moya, José Pablo.** *Procesamiento Digital de Señales*. Cartago : Tecnológico de Costa Rica, 2010.
- Bertran Albertí, Edward.** *Señales y Sistemas de Tiempo Discreto*. Barcelona : Ediciones UPC, 2003.
- Proakis, John; Salehi, Masoud.** *Contemporary Communication Systems using MATLAB*. Boston : PWS Publishing Company, 1998.
- Arrambí Díaz, Marcial.** *Modelado de un Radar Doppler de Pulsos (PDR)*. Tijuana : Instituto Politécnico Nacional, 2005.
- The Math Works, Inc.** *CDMA Reference Blockset For Use with Simulink, User's Guide*. s.l. : Algorex Inc., 2000-2001.
- Tranter, William H., y otros.** *Principles of Communication Systems Simulation with Wireless Applications*. New Jersey : PRENTICE HALL, 2004.
- Study Spread Spectrum in MATLAB.* **Xiaoxing, Wang.** Singapore : School of Electrical & Electronic Engineering, Nanyang Technological University.
- Frequency Hopping Spread Spectrum (FHSS) vs. Direct Sequence Spread Spectrum (DSSS) in Broadband Wireless Access (BWA) and Wireless LAN*

(WLAN). **Schwartz, Sorin M.** s.l.: Alvarion Professional Education Center (ALPEC), 2001, Vol. Versión 7.

*Frequency Hopping vs. Direct Sequence Spread Spectrum Techniques.* s.l.: RAYLINK AND RAYTHEON ELECTRONICS.

Sistemas de Cifra en Flujo. [En línea]  
[http://www.criptosec.unizar.es/doc/tema\\_c3\\_criptosec.pdf](http://www.criptosec.unizar.es/doc/tema_c3_criptosec.pdf).

Ruido. [En línea] Universidad Nacional de Colombia.  
[http://www.virtual.unal.edu.co/cursos/sedes/manizales/4040051/html/capitulos/cap\\_ii/ruido.html](http://www.virtual.unal.edu.co/cursos/sedes/manizales/4040051/html/capitulos/cap_ii/ruido.html).

Ruido en Sistemas de Comunicación. [En línea] Escuela Politécnica del Litoral.  
<http://www.slideshare.net/gbermeo/ruido-en-sistemas-de-comunicaciones-presentation>.

Spread Spectrum Scene. [En línea] SSS Online, Inc. <http://sss-mag.com/ebn0.html>.

**Chiu, L.** Procesamiento de Señales para Sistemas Inalámbricos. *Procesamiento Robusto ante desvanecimientos Selectivos en Frecuencia.* [En línea] Summa Sistemas C.A., 1995. [http://neutron.ing.ucv.ve/revista-e/No2/L\\_ChIU.html](http://neutron.ing.ucv.ve/revista-e/No2/L_ChIU.html).

**Játiva Espinoza, René.** Comunicaciones Móviles. *Modelos de Radio Radio-Propagaci Propagación Multitrayecto y Desvanecimientos de Peque Pequeña Escala.* [En línea] 21 de Marzo de 2007.  
[http://profesores.usfq.edu.ec/renej/Contenidos%20Comunicaciones%20Moviles/Apuntos%20Comunicaciones%20M%F3viles/M%F3viles\\_4.pdf](http://profesores.usfq.edu.ec/renej/Contenidos%20Comunicaciones%20Moviles/Apuntos%20Comunicaciones%20M%F3viles/M%F3viles_4.pdf).

**Willis, Mike.** Propagation Tutorial. *Fading Channels.* [En línea] 5 de Mayo de 2007. <http://www.mike-willis.com/Tutorial/PF15.htm>.

**Linnartz, Jean-Paul M.** Rake receiver. [En línea]  
[www.WirelessCommunication.NL](http://www.WirelessCommunication.NL).

<http://www.wireless.per.nl/reference/chaptr05/cdma/rake.htm>.

<http://www.dte.us.es/personal/sivianes/tcomu/Alteraciones.pdf>

<http://es.scribd.com/doc/4679956/Seguridad-Informatica-y-Criptografia-Sistemas-de-Cifra-en-Flujo>