

**ESCUELA POLITÉCNICA DEL EJÉRCITO**

**FACULTAD DE INGENIERÍA ELECTRÓNICA**

**PROYECTO DE GRADO PARA LA OBTENCIÓN DEL  
TÍTULO EN INGENIERÍA ELECTRÓNICA**

**“DESARROLLO DE GUÍAS DE LABORATORIO DE  
COMUNICACIONES DIGITALES, PARA LA FACULTAD DE  
INGENIERÍA ELECTRÓNICA DE LA E.S.P.E., UTILIZANDO  
MATLAB”**

**ALEX PAÚL ESPINOZA GUTIÉRREZ**

**QUITO – ECUADOR**

**NOVIEMBRE 2005.**

## **CERTIFICACIÓN**

Certificamos que la presente Tesis de Grado fue realizada en su totalidad por el señor **Alex Paúl Espinoza Gutiérrez** bajo nuestra dirección.

---

**Sr. Ing. Rodrigo Silva**

**DIRECTOR DE TESIS**

---

**Sr. Ing. Carlos Romero**

**CO DIRECTOR DE TESIS**

## **AGRADECIMIENTO**

*Quiero dar gracias a Dios, por todos y cada uno de los días de mi vida, porque ha sido Él quien me dado la fortaleza para seguir luchando cada instante, ha sido Él quien me ha dado mi familia, mis amigos y todo cuanto puedo conocer, y ha sido Él quien me ha ayudado a conseguir todas las metas que me he propuesto, como lo era acabar mi carrera universitaria con éxito, junto a mi familia y demás personas que quiero.*

*A mis padres, hermanos y toda mi familia por su apoyo, su paciencia, su dedicación y por todo aquello que han sabido brindarme y que hemos compartido a lo largo de mi vida; en especial en esta última etapa que está culminando, la cual ha sido difícil pero no imposible.*

*A todos mis amigos, en especial a Fredy, Esteban y Juan Carlos, por su valiosa y verdadera amistad en todo momento y en toda circunstancia, aunque salga de lo académico, porque al final de este largo y arduo caminar nos hacemos amigos y compañeros de vida más que de aulas. A todos mis compañeros y compañeras que han recorrido junto a mí en este tiempo, y que siempre serán recordados por todas las cosas que vivimos juntos.*

*A María Paola por el amor y la confianza que ha depositado en mí, por su preocupación y comprensión en todo momento, que me ha ayudado ha superar momentos difíciles.*

*A todos mis profesores, quienes con su guía y paciencia han sabido instruirme y a hacer de mí a más de un profesional una persona de provecho para saber servir y defender nuestras creencias y nuestros valores, en especial al Sr. Ing. Rodrigo Silva por todo su apoyo y colaboración en la dirección de este proyecto de grado, así como por la idea de su desarrollo; de igual manera al Sr. Ing. Carlos Romero, codirector de este proyecto, por toda su colaboración en la consecución de los objetivos del proyecto.*

*Alex Paúl Espinoza Gutiérrez.*

## **DEDICATORIA**

*A mis padres Angel Orlando y Martha Dolores, quienes han sido el mayor incentivo y apoyo para alcanzar mis metas durante toda mi existencia, y han sido Ellos quienes me han dedicado su vida y han querido para mi siempre lo mejor, y ahora este logro mío es de ellos y para ellos...*

*A mis hermanos Oswaldo Santiago y Cynthia Fernanda, quienes han compartido conmigo mis alegrías y tristezas, mis triunfos y mis fracasos y porque a pesar de las dificultades siempre han estado cerca de mi apoyándome en todo lo que he necesitado, y en quienes tengo plena confianza serán profesionales y personas de bien...*

*Finalmente, a todas las personas que luchan día a día y que son honestas en sus acciones, como una muestra de que siempre lo correcto y que el trabajo con tesón y transparencia llegan más lejos y dan mayor satisfacción que las extravagancias y sentimientos de éxitos mal ganados o inmerecidos...*

*Alex Paúl Espinoza Gutiérrez*



## **PRÓLOGO**

El desarrollo del proyecto “DESARROLLO DE GUÍAS DE LABORATORIO DE COMUNICACIONES DIGITALES, PARA LA FACULTAD DE INGENIERÍA ELECTRÓNICA DE LA E.S.P.E., UTILIZANDO MATLAB” comprende un estudio teórico y práctico de los contenidos analíticos de las asignaturas de Comunicación Digital y Codificación Digital de Señales.

El objetivo del presente proyecto es proveer una fuente de consulta y apoyo tanto a profesores como a estudiantes de Ingeniería Electrónica en el estudio de dichas asignaturas.

Para este efecto, la teoría contenida en el proyecto abarca muchos de los temas relacionados con los Sistemas de Comunicación Digital que usamos hoy en día en nuestras actividades casi ya sin darnos cuenta. Estos temas van desde la forma de convertir una señal de información análoga en una de tipo digital, hasta la multiplexión de estas señales para poder acceder al medio de forma que se optimicen los recursos y esquemas especiales como OFDM que son herramientas de actualidad con aplicaciones importantes a futuro; revisando para ello las modulaciones hechas en un transmisor, los efectos y formas de corregir los errores del ruido en el canal y la forma de recuperar la información en un receptor.

Pero el entendimiento no llega solo de la teoría, razón por la cual se incluyen algunas simulaciones de prácticas de laboratorio utilizando un software bastante potente como lo es MATLAB. La versión utilizada es la 7.0, (bastante nueva), cuyas herramientas y características facilitan el desarrollo de las simulaciones y por tanto el entendimiento sobre las características y funcionamiento de los Sistemas Digitales de Comunicación y sus partes revisadas en el compendio teórico.

## ÍNDICE

<b>CAPÍTULO I: CODIFICACIÓN DE FUENTE.....</b>	<b>1</b>
<b>1.1. INTRODUCCIÓN.....</b>	<b>1</b>
<b>1.2. MARCO TEÓRICO.....</b>	<b>4</b>
1.2.1. CONVERSIÓN ANÁLOGA-DIGITAL: MUESTREO, CUANTIZACIÓN, CODIFICACIÓN.....	4
1.2.1.1.1. Muestreo Ideal: Teorema del muestreo.....	6
1.2.1.1.2. Muestreo Natural.....	10
1.2.1.1.3. Muestreo de Señales Pasabanda.....	11
1.2.1.2. Cuantización.....	14
1.2.1.2.1. Cuantización uniforme.....	15
1.2.1.2.2. Cuantización logarítmica.....	16
1.2.1.2.3. Cuantización no uniforme.....	17
1.2.1.2.4. Cuantización vectorial.....	18
1.2.1.2.5. Error de Cuantización.....	19
1.2.1.3. Codificación.....	22
1.2.2. CODIFICADORES DE FORMA DE ONDA: PCM, DPCM, ADPCM.....	23
1.2.2.1. Codificadores en el Dominio del Tiempo.....	24
1.2.2.1.1. Modulación por Codificación de Pulsos.....	24
1.2.2.1.2. Modulación por Codificación de Pulsos Diferencial (DPCM).....	24
1.2.2.1.3. Modulación por Codificación de Pulsos Diferencial Adaptativa (ADPCM).....	26
1.2.2.2. Codificación en el Dominio de la Frecuencia.....	27
1.2.2.2.1. Codificación en Sub-Bandas.....	27
1.2.2.2.2. Codificación por Transformada.....	28
1.2.3. CODIFICADORES PARAMÉTRICOS: CELP, VSELP.....	29
1.2.3.1. Vocoder por Predicción Lineal.....	29

1.2.3.2. Vocoder LPC-10. LPC-10 (FS-1015).....	30
1.2.3.2.1. Emisor. ....	31
1.2.3.2.2. Receptor. ....	32
1.2.3.3. Codificadores Híbridos. ....	33
1.2.3.3.1. Codificación RELP. ....	33
1.2.3.3.2. Codificación Multipulso, MPC. ....	34
1.2.3.3.3. Introducción a la codificación CELP (FS-1016). ....	35
1.2.3.3.3.1. Codificador. ....	36
1.2.3.3.3.2. Decodificador. ....	37
1.2.3.3.4. Codificación VSELP. ....	37
1.2.3.3.5. RPE-LTP. ....	38
1.2.3.3.5.1. Codificador RPE-LTP. ....	40
1.2.3.3.5.2. Análisis LPC. ....	40
1.2.3.3.5.3. Filtro de predicción de retardo largo (LTP). ....	41
1.2.3.3.5.4. Codificación RPE. ....	41
1.2.3.3.5.5. Decodificador RPE-LTP. ....	43
<b>1.3. PRÁCTICAS CON MATLAB .....</b>	<b>44</b>
1.3.1. AUTORRELACIÓN DE SEÑAL DE VOZ .....	44
1.3.1.1. Objetivos .....	44
1.3.1.2. Procedimiento .....	44
1.3.1.3. Conclusiones .....	52
1.3.2. ANÁLISIS DE LA SEÑAL DE VOZ POR MODELAMIENTO AUTOREGRESIVO .....	53
1.3.2.1. Procedimiento .....	53
1.3.3. TASA DE CRUCE POR CERO .....	60
1.3.3.1. Procedimiento .....	60
1.3.3.2. Conclusiones. ....	63
<b>1.4. REFERENCIAS .....</b>	<b>65</b>
1.4.1. LIBROS .....	65
1.4.2. INTERNET.....	65

<b>CAPÍTULO II: TRANSMISIÓN BANDA BASE .....</b>	<b>66</b>
<b>2.1. INTRODUCCIÓN.....</b>	<b>66</b>
<b>2.2. MARCO TEÓRICO.....</b>	<b>68</b>
2.2.1. CÓDIGOS DE LÍNEA .....	68
2.2.1.1. Códigos de Línea Comunes .....	69
2.2.1.1.1. NRZ Unipolar (Binary On-Off Keying - OOK).....	70
2.2.1.1.2. Unipolar RZ. ....	71
2.2.1.1.3. Polar NRZ. ....	73
2.2.1.1.4. Polar RZ -Bipolar, Alternate Mark Inversion (AMI), o Pseudoternaria. ....	75
2.2.1.1.5. Codificación Manchester (Fase Dividida o Bi-fase Digital).....	77
2.2.1.2. Tamaño de la Palabra PCM .....	79
2.2.1.3. Señalización Multinivel, de Respuesta Parcial y Codificación Duobinaria .....	80
2.2.1.3.1. Señalización Multinivel. ....	80
2.2.1.3.2. Señalización de Respuesta Parcial y Codificación Duobinaria. ....	80
2.2.1.3.3. Precodificación. ....	82
2.2.2. INTERFERENCIA INTERSIMBÓLICA.....	83
2.2.2.1. Primer Método De Nyquist – Cero ISI .....	85
2.2.2.1.1. Filtro de Coseno Levantado. ....	86
2.2.2.2. Segundo y Tercer Métodos de Nyquist .....	88
2.2.2.3. DIAGRAMA DE OJO.....	89
2.2.3. RECEPTOR ÓPTIMO:PROBABILIDAD DE ERROR.....	90
<b>2.3. PRÁCTICAS CON MATLAB .....</b>	<b>93</b>
2.3.1. CÓDIGOS DE LÍNEA .....	93
2.3.1.1. Objetivos.....	93
2.3.1.2. Procedimiento .....	93
2.3.1.3. Conclusiones .....	97
2.3.2. ERRORES EN MODULACION BANDABASE.....	99
2.3.2.1. Objetivos.....	99
2.3.2.2. Procedimiento .....	99
2.3.2.3. Conclusiones .....	107

<b>2.4. REFERENCIAS .....</b>	<b>109</b>
2.4.1. LIBROS .....	109
2.4.2. INTERNET.....	109
<b>CAPÍTULO III: TRANSMISIÓN PASA BANDA M-ARIA.....</b>	<b>110</b>
<b>3.1. INTRODUCCIÓN.....</b>	<b>110</b>
<b>3.2. MARCO TEÓRICO.....</b>	<b>112</b>
3.2.1. RECEPTORES .....	112
3.2.1.1. Regiones de Decisión .....	113
3.2.1.2. Receptor de Correlación .....	113
3.2.1.3. Detección Coherente.....	115
3.2.1.3.1. Detección Coherente de MPSK. ....	115
3.2.1.3.2. Detección Coherente de FSK.....	117
3.2.1.4. Detección No Coherente .....	118
3.2.1.4.1. Detección de PSK Diferencial – DPSK. ....	118
3.2.1.4.2. Detección de FSK. ....	120
3.2.2. MODULADORES / DEMODULADORES:MPSK, MFSK, MQAM.....	121
3.2.2.1. Limite de Shannon para Capacidad de Información. ....	122
3.2.2.2. Desplazamiento en Frecuencia FSK (Frequency Shift Keying) .....	123
3.2.2.2.1. FSK Binario. ....	124
3.2.2.2.2. FSK M – ario.....	127
3.2.2.3. Desplazamiento en Fase PSK (Phase Shift Keying).....	128
3.2.2.3.1. BPSK – PSK Binario. ....	128
3.2.2.3.2. QPSK – PSK Cuaternario. ....	130
3.2.2.3.3. OQPSK – QPSK de Compensación (Offset). ....	132
3.2.2.3.4. PSK de Ocho Fases. ....	133
3.2.2.3.5. PSK de Dieciséis Fases. ....	136
3.2.2.3.6. BPSK Diferencial.....	137
3.2.2.4. Desplazamiento en Amplitud ASK (Amplitude Shift Keying) .....	138
3.2.2.5. Desplazamiento en Amplitud y Fase APK (Amplitude Phase Keying) .....	139

3.2.2.6. Modulación de Amplitud en Cuadratura QAM. ....	140
3.2.2.6.1. 8 – QAM. ....	140
3.2.2.6.2. 16 – QAM. ....	142
3.2.2.7. Desplazamiento Mínimo De Fase MsSK (Minimun Shift Keying) .....	143
<b>3.2.3. DESEMPEÑO DE SISTEMAS .....</b>	<b>148</b>
3.2.3.1. Eficiencia del Ancho de Banda.....	148
3.2.3.2. Probabilidad de Error $P_b(E)$ .....	149
3.2.3.2.1. BPSK Detectado Coherentemente. ....	149
3.2.3.2.2. DBPSK Detectado Coherentemente. ....	150
3.2.3.2.3. FSK Binario Detectado Coherentemente.....	150
3.2.3.2.4. FSK Binario Detectado No Coherentemente.....	151
3.2.3.2.5. Señalización M-aria. ....	153
3.2.3.2.6. Probabilidad de Error de Símbolo para MPSK.....	154
3.2.3.2.7. M- QAM.....	155
3.2.3.2.8. Probabilidad de Error de Símbolo para MFSK.....	156
<b>3.3. PRÁCTICAS CON MATLAB .....</b>	<b>158</b>
3.3.1. MODULACIÓN DIGITAL .....	158
3.3.1.1. Objetivos.....	158
3.3.1.2. Procedimiento .....	158
3.3.1.3. Conclusiones.....	163
3.3.2. DETECCIÓN DE SEÑALES PASABANDA.....	165
3.3.2.1. Objetivos.....	165
3.3.2.2. Procedimiento .....	165
3.3.2.3. Conclusiones.....	172
<b>3.4. REFERENCIAS .....</b>	<b>176</b>
3.4.1. LIBROS.....	176
3.4.2. INTERNET.....	176

<b>CAPÍTULO IV: CODIFICACIÓN DEL CANAL .....</b>	<b>177</b>
<b>4.1. INTRODUCCIÓN.....</b>	<b>177</b>
<b>4.2. MARCO TEÓRICO.....</b>	<b>179</b>
4.2.1. CODIFICACIÓN DE FORMA DE ONDA .....	179
4.2.1.1. Señales Antipodales y Ortogonales .....	179
4.2.1.2. Codificación de Correlación Cruzada.....	180
4.2.1.2.1. Códigos Ortogonales.....	181
4.2.1.2.2. Códigos Bi – Ortogonales.....	182
4.2.1.2.3. Códigos Trans–Ortogonales.....	183
4.2.2. ALGORITMOS FEC Y ARQ:CÓDIGOS DE BLOQUES, CONVOLUCIONALES, ENTRELAZADOS Y CONCATENADOS.....	183
4.2.2.1. ARQ (Automatic Repeat Request) .....	184
4.2.2.1.1. ARQ de Parada y Espera (stop-wait ARQ).....	184
4.2.2.1.2. ARQ Continuo con PullBack.....	184
4.2.2.1.3. ARQ Continuo con Repetición Selectiva .....	185
4.2.2.2. Secuencias Estructuradas.....	185
4.2.2.2.1. Códigos de Paridad. ....	186
4.2.2.2.1.1. Paridad Singular. ....	186
4.2.2.2.1.2. Paridad Rectangular. ....	187
4.2.2.3. Códigos de Bloques Lineales.....	188
4.2.2.3.1. Fortaleza del Código. ....	191
4.2.2.3.2. Ganancia de Codificación. ....	192
4.2.2.4. Códigos de Hamming. ....	193
4.2.2.5. Códigos BCH (Bose-Chaudhuri-Hocquenghem) .....	195
4.2.2.6. Códigos Cíclicos .....	197
4.2.2.6.1. Códigos Cíclicos Sistemáticos.....	199
4.2.2.6.2. Códigos de Redundancia Cíclica – CRC. ....	200
4.2.2.6.3. Generación de Códigos Cíclicos.....	201
4.2.2.6.4. Decodificación de Códigos Cíclicos.....	202
4.2.2.7. Códigos Entrelazados y Concatenados .....	203
4.2.2.8. Códigos Convolutivos .....	205
4.2.2.8.1. Codificación. ....	206

4.2.2.8.1.1. Diagrama de Estados.....	209
4.2.2.8.1.2. Árbol de código.....	210
4.2.2.8.1.3. Diagrama de Trellis.....	211
4.2.2.8.2. Decodificación.....	214
4.2.2.8.2.1. Algoritmo de Viterbi.....	214
4.2.2.9. Códigos Reed – Solomon.....	220
4.2.2.9.1. Codificación.....	221
4.2.2.10. Códigode Detección y Corrección de Ráfaga de Errores.....	222
4.2.3. TURBO CÓDIGOS.....	223
4.2.3.1. Problemas Prácticos.....	225
<b>4.3. PRÁCTICAS CON MATLAB.....</b>	<b>228</b>
4.3.1. CÓDIGOS DE CONTROL DE ERRORES.....	228
4.3.1.1. Objetivos.....	228
4.3.1.2. Procedimiento.....	228
4.3.1.3. Conclusiones.....	233
<b>4.4. REFERENCIAS.....</b>	<b>235</b>
4.4.1. LIBROS.....	235
4.4.2. INTERNET.....	235
<b>CAPÍTULO V: MÚLTIPLEX.....</b>	<b>236</b>
<b>5.1. INTRODUCCIÓN.....</b>	<b>236</b>
<b>5.2. MARCO TEÓRICO.....</b>	<b>238</b>
5.2.1. SINCRONISMO.....	238
5.2.1.1. Modos de Operación.....	239
5.2.1.1.1. Plesiócromo.....	239
5.2.1.1.2. Síncrono.....	239
5.2.1.2. Métodos de Sincronismo.....	240
5.2.1.2.1. Reloj Primario de Referencia –PRC.....	240



5.2.1.2.2. Escala de Tiempo Universal Coordinado – UTC. ....	240
5.2.1.2.3. Sistema de Posicionamiento Global – GPS. ....	241
5.2.1.2.4. LORAN – C. ....	241
5.2.1.2.5. TAI. ....	241
5.2.2. PDH – PLESIOSYNCHRONOUS DIGITAL HIERARCHY .....	241
5.2.2.1. Jerarquías .....	243
5.2.2.1.1. Europea. ....	243
5.2.2.1.2. Americana. ....	245
5.2.2.1.3. Japonesa. ....	245
5.2.3. SONET (SYNCHRONOUS OPTICAL NETWORK) .....	247
5.2.4. SDH – SYNCHRONOUS DIGITAL HIERARCHY .....	249
5.2.5. COMPARACIÓN DE PDH CONTRA SDH .....	253
5.2.5.1. Falta de Flexibilidad .....	253
5.2.5.2. Sin Monitorización .....	254
5.2.5.3. Ausencia de Especificación de Líneas de Salida .....	254
5.2.6. WDM – WAVELENGTH DIVISION MULTIPLEXING .....	256
<b>5.3. REFERENCIAS .....</b>	<b>264</b>
5.3.1. LIBROS .....	264
5.3.2. INTERNET.....	264
<b>CAPÍTULO VI: SPREAD SPECTRUM .....</b>	<b>265</b>
<b>6.1. INTRODUCCIÓN.....</b>	<b>265</b>
<b>6.2. MARCO TEÓRICO.....</b>	<b>267</b>
6.2.1. DEFINICIONES.....	267
6.2.1.1. Sistemas de Espectro Disperso y Secuencias Pseudo Aleatorias .....	269
6.2.1.2. Secuencias de Máxima Longitud .....	270
6.2.1.3. Ganancia del Procesamiento (Gp) .....	272
6.2.1.4. Margen Sobre la Interferencia (J) .....	273
6.2.1.5. Secuencias Gold.....	273

6.2.1.6. Secuencias Walsh .....	274
6.2.1.7. Técnicas de Modulación de Ensanchamiento.....	275
6.2.2. SECUENCIA DIRECTA.....	276
6.2.3. SALTO DE FRECUENCIA .....	279
6.2.4. SISTEMA HIBRIDO DS/(F)FH.....	282
6.2.5. SALTO DE TIEMPO .....	283
<b>6.3. PRÁCTICAS CON MATLAB .....</b>	<b>284</b>
6.3.1. SISTEMA DE COMUNICACIÓN CDMA CON SPREAD SPECTRUM .....	284
6.3.1.1. Objetivos.....	284
6.3.1.2. Procedimiento .....	284
6.3.1.3. Conclusiones.....	291
<b>6.4. REFERENCIAS .....</b>	<b>293</b>
6.4.1. LIBROS.....	293
6.4.2. INTERNET.....	293
<b>CAPÍTULO VII: ACCESO MÚLTIPLE.....</b>	<b>294</b>
<b>7.1. INTRODUCCIÓN.....</b>	<b>294</b>
<b>7.2. MARCO TEÓRICO.....</b>	<b>296</b>
7.2.1. SISTEMAS DE ACCESO MÚLTIPLE POR BANDA.....	296
7.2.1.1. Sistemas de Banda Estrecha .....	296
7.2.1.2. Sistemas de Banda Ancha.....	296
7.2.2. DIVISIÓN DE FRECUENCIA – FDMA .....	297
7.2.3. DIVISIÓN DE TIEMPO – TDMA .....	298
7.2.4. DIVISIÓN DE CÓDIGO – CDMA .....	301
7.2.4.1. Ventajas y Beneficios .....	302
7.2.4.2. Funcionamiento .....	305
7.2.5. DIVISIÓN DE ESPACIO – SDMA.....	307
7.2.6. DIVISIÓN DE POLARIDAD – PDMA .....	308

7.2.7. COMPARACIÓN DE LOS SISTEMAS DE ACCESO MÚLTIPLE.....	308
<b>7.3. PRÁCTICAS CON MATLAB .....</b>	<b>310</b>
7.3.1. PROYECTO DE DISEÑO DE UN SISTEMA DE COMUNICACIÓN CELULAR ..	
.....	310
7.3.1.1. Diseño del Codificador de Voz con Muestreo y Cuantización.....	310
7.3.1.1.1. Observaciones:.....	311
7.3.1.1.2. Primera Parte.- Cuantización No Uniforme.....	312
7.3.1.1.3. Segunda Parte.- Analítica.....	313
7.3.1.1.4. Tercera Parte.- Diferencial.....	314
7.3.1.2. Diseño de la Modulación y Forma de Pulso .....	316
7.3.1.2.1. Requerimientos de diseño.....	316
7.3.1.2.2. Condiciones de Diseño.....	316
7.3.1.2.3. Costos.....	317
7.3.1.2.4. Herramienta de Simulación.....	318
7.3.1.2.5. Desarrollo.....	319
7.3.1.2.6. Análisis y Selección de Modulación.....	327
7.3.1.3. Diseño del Esquema de Acceso Múltiple y Presupuesto del Enlace .....	330
7.3.1.3.1. Técnica de Acceso Múltiple.....	330
7.3.1.3.2. Desarrollo.....	331
7.3.1.3.3. Elección.....	332
<b>7.4. REFERENCIAS .....</b>	<b>334</b>
7.4.1. LIBROS .....	334
7.4.2. INTERNET.....	334
<b>CAPÍTULO VIII: ESQUEMAS ESPECIALES .....</b>	<b>336</b>
<b>8.1. INTRODUCCIÓN.....</b>	<b>336</b>
<b>8.2. MARCO TEÓRICO.....</b>	<b>338</b>
8.2.1. CAP /DMT.....	338

8.2.1.1. Discrete Multitone – DMT .....	338
8.2.1.1.1. Loading. ....	341
8.2.1.1.2. Partición de Canales.....	342
8.2.1.2. Carrierless Amplitude/Phase – CAP .....	342
8.2.1.2.1. Transmisor. ....	344
8.2.1.3. DMT vs CAP .....	346
8.2.2. TCM– TRELLIS CODED MODULATION .....	346
8.2.2.1. Particionamiento de Conjuntos .....	347
8.2.2.2. Modulación Codificada .....	349
8.2.2.3. Decodificación de Códigos TCM.....	352
8.2.3. OFDM – ORTHOGONAL FRECUENCY-DIVISION MULTIPLEXING.....	353
8.2.3.1. Características de OFDM .....	354
8.2.3.1.1. Diagrama de Bloques General. ....	356
8.2.3.1.2. Transmisor. ....	358
8.2.3.1.3. Receptor. ....	359
8.2.3.2. Sistemas que Utilizan OFDM.....	360
8.2.3.3. W-OFDM - Wideband OFDM .....	360
8.2.3.4. Ventajas de Usar OFDM .....	361
<b>8.3. PRÁCTICAS CON MATLAB .....</b>	<b>363</b>
8.3.1. ORTHOGONAL FRECUENCY-DIVISION MULTIPLEXING.....	363
8.3.1.1. Objetivos.....	363
8.3.1.1.1. Descripción de Bloques. ....	363
8.3.1.1.2. Transmisión.....	366
8.3.1.1.3. Recepción.....	367
8.3.1.1.4. Simulación. ....	368
8.3.1.1.5. Resultados. ....	370
8.3.1.2. Conclusiones.....	383
<b>8.4. REFERENCIAS .....</b>	<b>384</b>
8.4.1. LIBROS .....	384
8.4.2. INTERNET.....	384

# CAPÍTULO I

## CODIFICACIÓN DE FUENTE

### 1.1. INTRODUCCIÓN

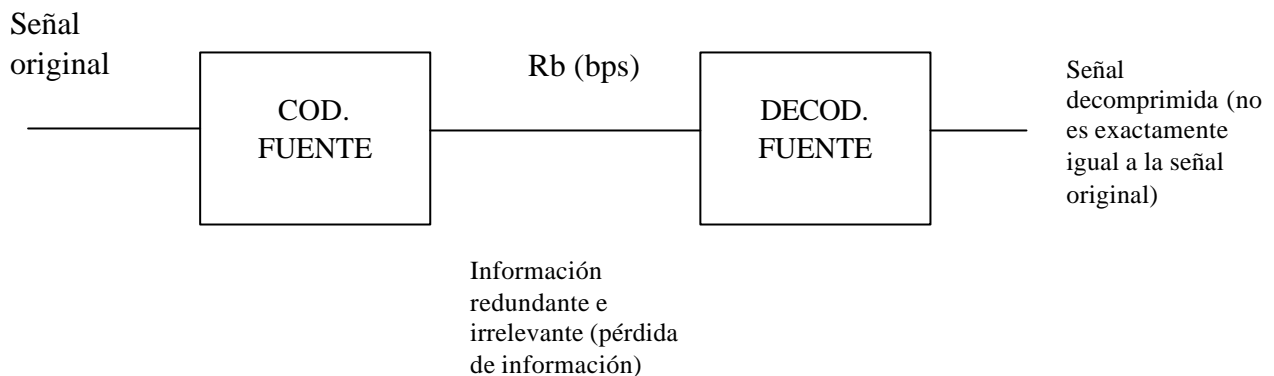
Este proceso se lleva a cabo con el uso del codificador y decodificador de la fuente. En el extremo transmisor el codificador de la fuente elimina la mayor cantidad posible de información redundante e irrelevante de la fuente original (audio, video, datos). El resultado es una disminución en la razón de bits ( $R_b$ , bps) y una reducción en los requerimientos de ancho de banda de transmisión. En el extremo receptor se reinserta (aproximadamente) la información eliminada y se recupera la información original (mensaje fuente).

Los algoritmos que llevan a cabo los procesos anteriores se denominan técnicas de compresión y sus objetivos fundamentales son:

1. Optimizar la cantidad de información a eliminar de tal forma que el receptor no pueda identificar las diferencias entre la señal decomprimida y la de la fuente original.
2. Reducir la razón de bits de información para reducir los requerimientos de ancho de banda de transmisión.

La familia de normas MPEG (Motion Picture Expert Group) se han convertido en estándares mundiales para la compresión-decompresión de señales audiovisuales. La norma MPEG-1 es utilizada para el mantenimiento y transmisión de imágenes fijas en aplicaciones de multimedia con velocidades de salida de 1.5 Mbps. La norma MPEG-2 es utilizada para el almacenamiento y transmisión de imágenes en movimiento, fundamentalmente en radiodifusión, con velocidades de salida que van desde 2 hasta 15

Mbps. En la figura 1.1 se muestra el proceso integral de compresión-decompresión de la información digital.



**Figura. 1.1. Proceso de Compresión-Decompresión de la Información Digital**

En cuanto a los protocolos de audio soportados por H.320 (G.711, G.722 y G.728) diseñados para distintas necesidades de audio: G.711 utiliza la codificación PCM proporcionando calidad de audio a 64 Kbits (en el tramo de 3 KHz).

La recomendación G.722 describe el uso de la modulación adaptativa diferencial de pulsos para transmitir audio de alta calidad 7 khz en 48, 56 o 64 kbps. Esta recomendación también permite la transmisión de datos a 16 kbps sobre un canal de 64 kbps, con los 48 kbps restantes para audio.

El algoritmo G.728 usa sólo 16 kbps para compresión de audio, lo cual da mayor espacio para el vídeo y opcionalmente para los datos. El resultado es una significativa mejor calidad de vídeo que cuando se utilizan algoritmos de audio convencionales. Es especialmente recomendable cuando se trabaje sobre líneas de 128 kbps.

ADPCM es muy útil para codificar voz a bit rates medios. La CCITT propone un estándar de codificación de voz telefónica a una velocidad de 32 kb/s. Es el estándar G.721.

El G.729 (CS-ACELP) de la ITU-T es un estándar internacional que se adapta muy bien a la compresión de los flujos de estándar de 64 Kbps de los canales PCM usados típicamente para la transmisión de voz, consiguiendo un resultado tan bajo como 8 Kbps.

Se dice que los símbolos son generados por una fuente y a la conversión a dígitos binarios se le llama codificación de fuente (por ejemplo PCM o DPCM). El codificador de fuente también se puede encargar de codificar eficientemente los datos binarios, asignando, por ejemplo, más bits a los símbolos menos probables y menos bits a los más probables. Por su parte, el codificador de canal introducirá bits redundantes de una manera controlada, con el fin de fortalecer la información frente al ruido o a la interferencia que pueda conseguir en el canal. Luego de pasar por un canal que, entre otras cosas, agrega ruido, el receptor se encarga de realizar todas las operaciones inversas a fin de rescatar el mensaje original.

El presente capítulo presenta una descripción teórica del procedimiento seguido para la conversión analógica a digital, que consta de procesos de muestreo (teorema de Nyquist), de cuantización (clases) y de codificación binaria de las señales. A continuación los diferentes codificadores de forma de onda PCM y sus principales variaciones Diferenciales y características. Se presenta también los codificadores paramétricos más conocidos en los dominios tiempo y frecuencia, la estructura de los codificadores y principales características.

Al final se incluyen las prácticas de laboratorio relativas a los temas estudiados con sus respectivos desarrollo y análisis de resultados.

## 1.2. MARCO TEÓRICO

### 1.2.1. CONVERSIÓN ANÁLOGA-DIGITAL: MUESTREO, CUANTIZACIÓN, CODIFICACIÓN

La conversión Analógico-Digital consta de varios procesos:

- Muestreo
- Cuantización
- Codificación

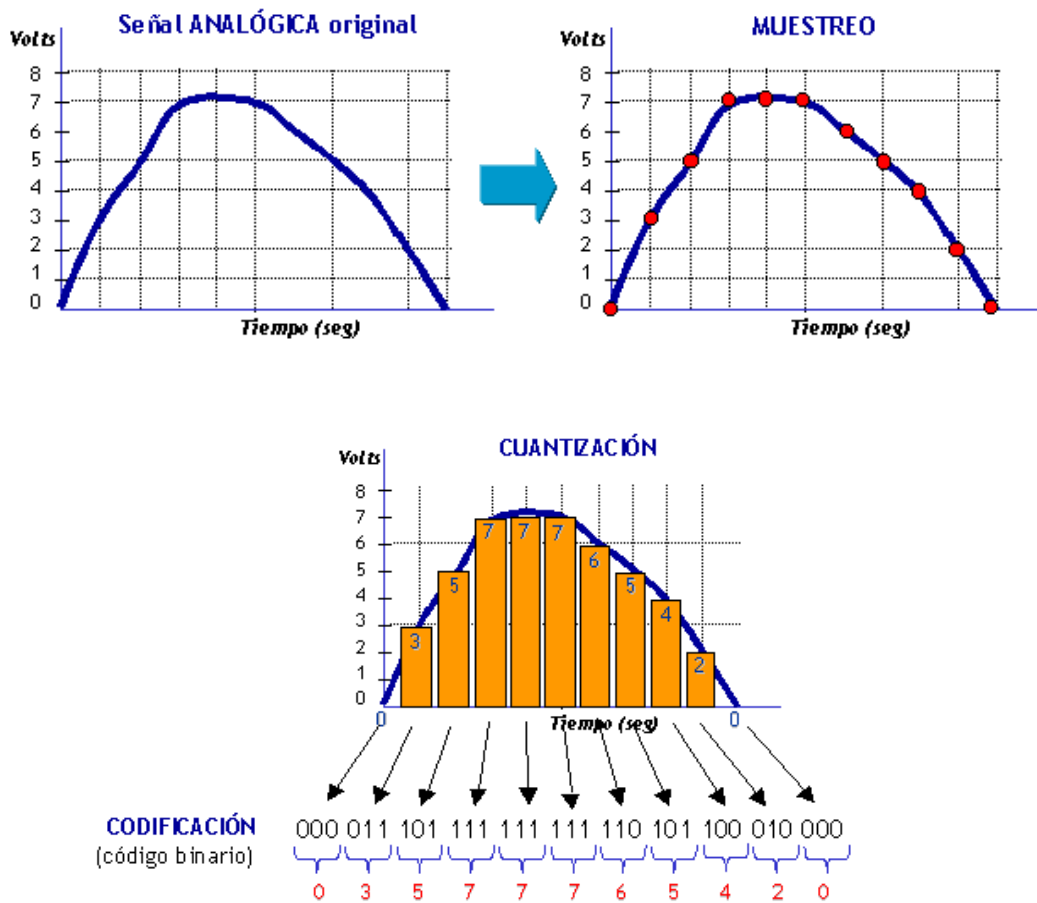


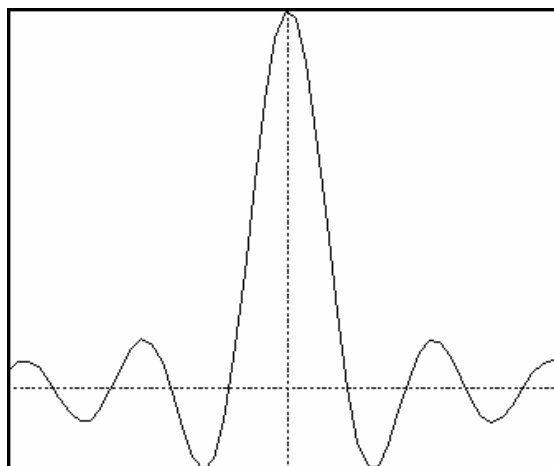
Figura. 1.2. Pasos del Proceso de Conversión Análoga- Digital



### 8.1.1.1. Muestreo

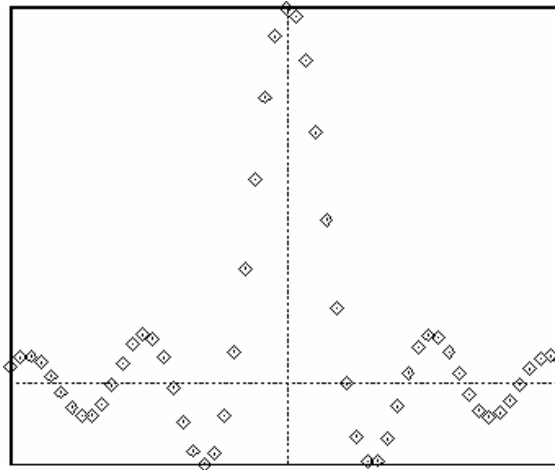
Las señales digitales presentan grandes ventajas a la hora de ser transmitidas y/o procesadas, principalmente: mayor inmunidad al ruido, mayor facilidad de procesamiento y facilidad de multiplexaje. Es por esto que existe interés en convertir señales analógicas (tiempo y amplitud continuos) en señales digitales (tiempo y amplitud discretos). Todas las señales en la naturaleza son de analógicas, como por ejemplo la voz. Así, para que la voz pueda ser procesada por hardware (y software) digital es necesario convertirla a una señal que sea discreta tanto en el tiempo como en amplitud.

El muestreo consiste en el proceso de conversión de señales continuas a señales discretas en el tiempo. Este proceso se realiza midiendo la señal en momentos periódicos del tiempo. Por ejemplo, dada la siguiente señal continua:



**Figura. 1.3. Señal Sinusoidal Continua**

Tras muestrearla, obtenemos la siguiente señal discreta:



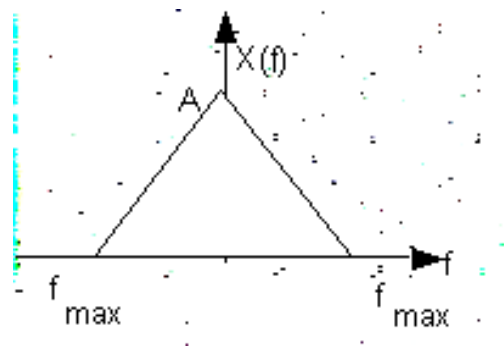
**Figura. 1.4. Señal Sinusoidal Discreta**

Si se aumenta el número de muestras por unidad de tiempo, la señal muestreada se parecerá más a la señal continua. El número de muestras por segundo se conoce en inglés como el **bit-rate**. Si el bit-rate es lo suficientemente alto, la señal muestreada contendrá la misma información que la señal original.

Existen diferentes formas para realizar el proceso de muestreo:

#### 1.2.1.1.1. Muestreo Ideal: Teorema del muestreo.

Suponemos una señal  $x(t)$  cuya transformada (Fourier)  $X(f)$  tiene la siguiente forma:



**Figura. 1.5. Señal Base  $X(f)$**

Por ejemplo las señales de voz para telefonía básica tienen  $f_{\max}=4$  KHz, y la voz en general puede alcanzar  $f_{\max}=20$  KHz, por lo que para señales de audio se toma  $f_{\max}=30$  KHz y para señales de video  $f_{\max}=6$  MHz.

Se pueden tomar muestras de la señal multiplicándola por un tren de impulsos periódicos de periodo  $t_s$ , tal y como se muestra en la figura, lo que es conocido como *muestreo ideal*.

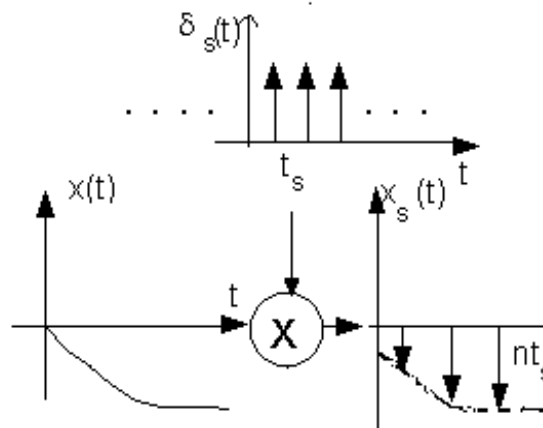


Figura. 1.6. Esquema de Muestreo Ideal

Es decir:

$$x_s(t) = x(t) \cdot d_s(t)$$

$$x_s(t) = \sum_{n=-\infty}^{n=\infty} x(nt_s) \mathbf{d}(t - nt_s)$$

En el dominio de la frecuencia se tendrá que:

$$F \left[ \sum_{n=-\infty}^{n=\infty} \mathbf{d}(t - nt_s) \right] = \frac{1}{t_s} \sum_{n=-\infty}^{n=\infty} \mathbf{d}(f - nf_s)$$

Por lo tanto, llamando  $f_s$  al inverso de  $t_s$ , se tendrá que:

$$X_s(f) = f_s \sum_{n=-\infty}^{n=\infty} X(f - f_s)$$

La convolución de una función cualquiera con una delta reproduce a la función en el punto donde ocurre la delta y así

$$X_s(f) = f_s X(f) * \sum_{n=-\infty}^{n=\infty} \mathbf{d}(f - f_s)$$

De manera que el espectro de la señal muestreada será el siguiente:

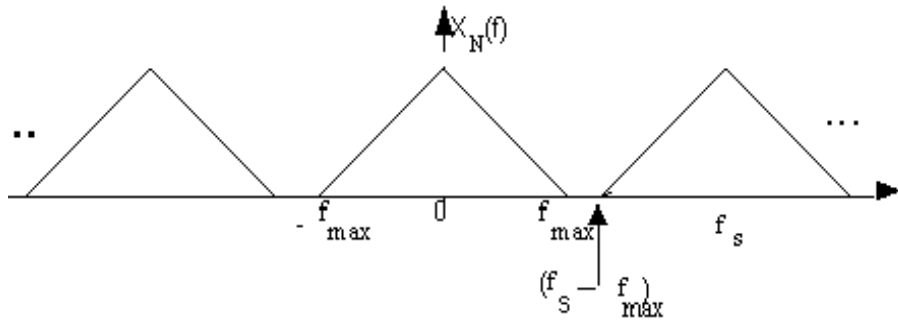


Figura. 1.7. Señal Muestreada en el Dominio de Frecuencia

El espectro de la señal original se repite cada  $f_s$ . Si quisiéramos rescatar la señal original, bastaría utilizar un filtro pasabajo (LPF) ideal cuya frecuencia de corte sea  $f_{\max}$ .

La frecuencia mínima de muestreo sería  $f_s = 2 f_{\max}$  muestras por segundo, y se conoce como la frecuencia de Nyquist. Si se muestrea a una frecuencia inferior a la de Nyquist los espectros de la señal muestreada se solaparán y no se podrá recuperar el mensaje original, produciendo se el efecto conocido como "aliasing".

La salida del filtro, en el dominio de la frecuencia, será:

$$Y(f) = X_s(f) \cdot KP(f/2f_{\max}) = \frac{K}{t_s} X(f)$$

En el dominio del tiempo el producto se cambia por la convolución:

$$y(t) = h(t) \sum_{n=-\infty}^{n=\infty} x(nt_s) \mathbf{d}(t - nt_s) = \sum_{n=-\infty}^{n=\infty} x(nt_s) h(t - nt_s)$$

La respuesta al impulso de un filtro pasabajo ideal como el descrito sería

$$h(t) = 2Kf_{\max} \text{Sinc}(2f_{\max}t)$$

Por lo tanto

$$y(t) = 2Kf_{\max} \sum_{n=-\infty}^{n=\infty} x(nt_s) \text{Sinc}[2f_{\max}(t - nt_s)] = Kf_s x(t)$$

$$y(t) = 2f_{\max} t_s \sum_{n=-\infty}^{n=\infty} x(nt_s) \text{Sinc}[2f_{\max}t - 2nf_{\max}t_s]$$

Cuando  $f_s = 2f_{\max}$

$$x(t) = \sum_{n=-\infty}^{n=\infty} x(n/2f_{\max}) \text{Sinc}[2f_{\max}t - n]$$

Es decir, se suman infinitos Sincs con pesos iguales a cada muestra  $x(nt_s)$  y esto reproduce a la señal  $x(t)$  (fórmula de interpolación). Se necesitan todas las muestras para obtener  $x(t)$ . Como en la práctica solo se tendrá un número finito de muestras, existirá un error llamado de truncamiento. En la práctica no se puede realizar este tipo de muestreo ideal ya que es imposible "fabricar" un tren de impulsos periódicos, una solución sería usar cualquier señal periódica de forma que la señal muestreada  $x_M(t)$  vendría dada por:

$$x_M(t) = \sum_{n=-\infty}^{n=\infty} x(t) C_n e^{jn2\pi f_s t}$$

En donde se ha sustituido la expresión de la señal periódica por la de su serie de Fourier. En el dominio de la frecuencia, la señal periódica se representa por un tren de impulsos con peso  $C_n$ , de manera que el espectro de la señal muestreada será la repetición del espectro de  $X(f)$  cada  $f_s$  multiplicado por  $C_n$ .

$$X_M(f) = \sum_{n=-\infty}^{n=\infty} C_n X(f - nf_s)$$

$$X_M(f) = X(f) * \sum_{n=-\infty}^{n=\infty} C_n \delta(f - nf_s)$$

### 1.2.1.1.2. Muestreo Natural.

Cuando la señal periódica es un tren de pulsos el muestreo se le llama muestreo natural. La señal en tiempo luciría como sigue:

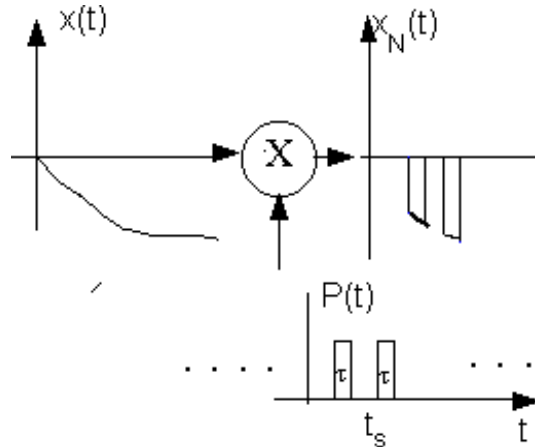


Figura. 1.8. Muestreo Natural con Pulsos

El espectro en frecuencia por otro lado, sería:

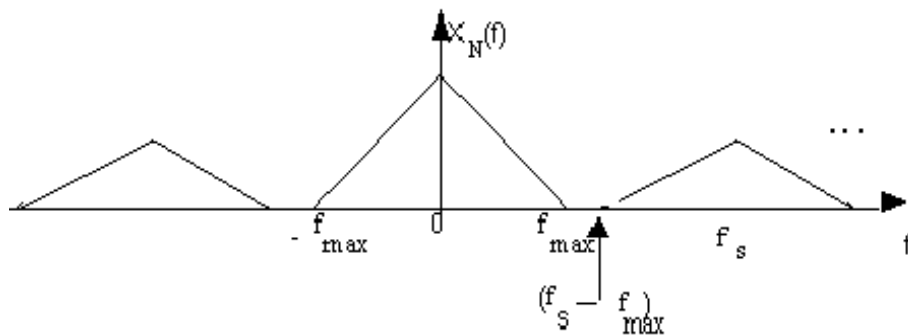
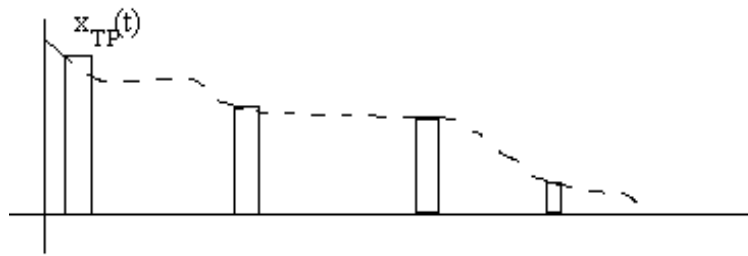


Figura. 1.9. Señal Muestreada en el Dominio de Frecuencia

En este caso la señal  $x(t)$ , al igual que en muestreo ideal, se puede recuperar con un filtro pasabajo. Aunque, en la práctica se prefiere otro tipo de muestreo, el llamado **tope plano**, en el cual se toma una muestra de la señal cada  $t_s$  y se mantiene durante un tiempo  $t$ . La señal luciría como sigue:



**Figura. 1.10. Muestreo de Tope Plano**

Las ventajas de este tipo de muestreo, entre otras, son las siguientes:

- Es más fácil de realizar con circuitos llamados Sample&Hold.
- Es más inmune al ruido.
- No importa la forma de los pulsos.
- La desventaja mas resaltante es que el espectro de la señal muestreada esta conformado por repeticiones distorsionadas del espectro de la señal original, tal y como demostraremos a continuación.

La señal muestreada tope-plano puede expresarse de la siguiente forma:

$$x_{TP}(t) = x_s(t) * \Pi\left(\frac{t}{\tau}\right)$$

$$X_{TP}(f) = X_s(f) \cdot \tau \text{Sincf } \tau$$

Con ello se introduce distorsión, por lo que para disminuirla se debería disminuir el valor de  $\tau$ , pero esto también disminuye la amplitud de la señal. Lo que se acostumbrar hacer en estos casos es, usar un  $\tau$  intermedio y luego en el receptor se compensa con una red de respuesta en frecuencia inversa en la banda de la señal original.

### 1.2.1.1.3. Muestreo de Señales Pasabanda.

Suponga  $x(t)$  tal que  $X(f)$  sea:

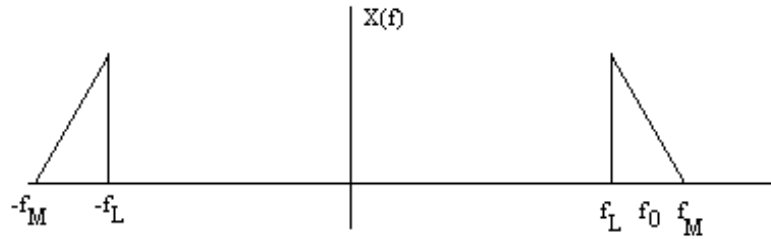


Figura. 1.11. Señal Pasabanda

Además si contamos con los parámetros:

$$B = f_M - f_L$$

$$k = \frac{f_M}{B}$$

$$k - 1 = \frac{f_L}{B}$$

Siempre que  $B$  (ancho de banda de la señal) sea menor que  $f_L$ , es posible muestrear a una frecuencia menor que la que impondría Nyquist siempre que se garantice que las repeticiones espectrales no se superponen con el espectro  $X(f)$ , es decir:

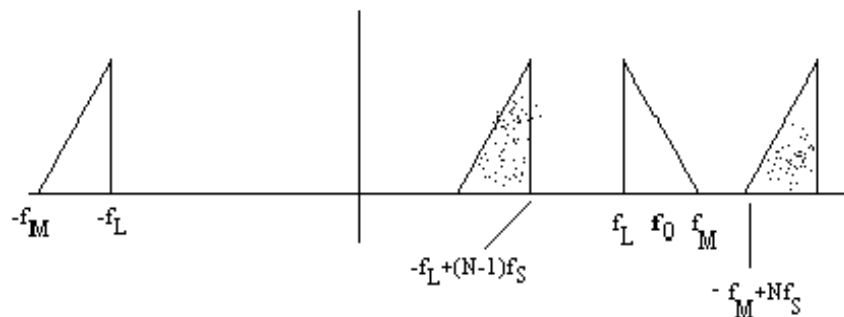


Figura. 1.12. Señal Muestreada en el Dominio de Frecuencia

Para que no exista solapamiento, de forma que  $x(t)$  pueda ser rescatado con un filtro pasabanda, se debe cumplir que:

$$-f_L + (N-1)f_s \leq f_L$$

$$-f_M + Nf_s \geq f_M$$

$$f_s \leq \frac{2f_L}{N-1}$$



$$f_M \geq \frac{2f_M}{N}$$

$$\frac{2kB}{N} \leq f_s \leq \frac{2(k-1)B}{N-1}$$

$$k \geq N$$

De forma que dado el espectro  $X(f)$ , se obtiene  $f_L$  y  $f_M$ , y por ende  $B$  y  $k$ . Con esto se definen los posibles valores de  $N$  y finalmente esto delimita los posibles valores de la frecuencia de muestreo.

### Ejemplo:

**Suponga que  $X(f)$  es tiene la forma pasabanda ilustrada antes con  $f_L=3000$  Hz,  $f_M=4000$  Hz. Por lo tanto  $B=1000$  Hz.**

De esta forma  $k=4$  y  $N$  puede tomar los siguientes valores 1, 2,3 y 4, y al aplicar la ecuación que define el rango de valores de  $f_s$  tenemos:

$$\frac{2.4B}{N} \leq f_s \leq \frac{2.3B}{N-1}$$

$$\frac{8B}{N} \leq f_s \leq \frac{6B}{N-1}$$

$$N = 1$$

$$8B \leq f_s \leq \infty$$

$$N = 2$$

$$4B \leq f_s \leq 6B$$

$$N = 3$$

$$\frac{8B}{3} \leq f_s \leq 3B$$

$$N = 4$$

$$2B \leq f_s \leq 2B$$

Por lo tanto en este caso se puede muestrear a 2KHz (que es menor que Nyquist que es 8 KHz)) y recuperar la señal con un filtro pasabanda. Se puede observar que mientras quepan mas repeticiones entre 0 y  $f_L$  es posible bajar cada vez más la frecuencia de muestreo.

### 1.2.1.2. Cuantización

La cuantificación es la conversión de una señal discreta en el tiempo evaluada de forma continua a una señal discreta en el tiempo discretamente evaluada. El valor de cada muestra de la señal se representa como un valor elegido de entre un conjunto finito de posibles valores.

Para hacer esto, la amplitud de la señal de audio es representada en una serie de pasos discretos. Cada paso está dado entonces por un número en código binario que digitalmente codifica el nivel de la señal. La longitud de la palabra determina la calidad de la representación. Una vez más, una palabra más larga, mejor la calidad de un sistema de audio (comparando una palabra de 8 bits con una de 16 bits o 32 bits).

Se divide el rango total de la señal en **M** franjas de tamaño **a**. **M es el número de niveles de cuantización y a es llamado el paso del cuantificador.** En cada intervalo de tiempo se observa en que rango de voltaje se encuentra la señal y en función de esto se le asigna un nivel de voltaje a la salida tal y como se ilustra a continuación:

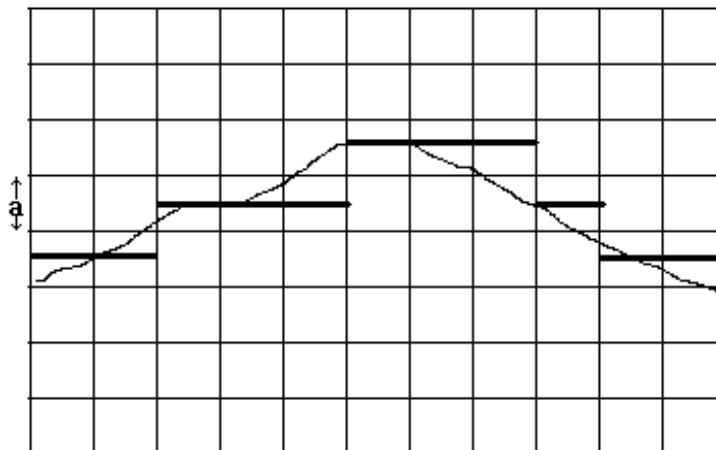


Figura. 1.13. Paso de Cuantización “a”

Los niveles de cuantización se eligen en función de la aplicación y del receptor. Si por ejemplo la señal es de voz, para lograr inteligibilidad basta usar 256 niveles de cuantización. El bit de resolución de un sistema define el rango dinámico del sistema.

Aproximadamente cada bit da una ganancia de 6 dB. Por ejemplo:

- **8 bits** equivale a **256 estados** = 48 dB (decibeles)
- **16 bits** equivalen a **65,536 estados** = 96 dB.

Entonces, se debe de tomar muestras a tiempos menores y se debe de cuantizar a mayores niveles (bits), para minimizar los **errores de cuantización**.



**Figura. 1.14. Error de Cuantización**

Se conoce como error de cuantificación (o **ruido**), a la diferencia entre la señal de entrada (sin cuantificar) y la señal de salida (ya cuantificada), interesa que el ruido sea lo más bajo posible. Para conseguir esto, se pueden usar distintas técnicas de cuantificación:

#### **1.2.1.2.1. Cuantización uniforme.**

En los cuantificadores uniformes la distancia entre los niveles de reconstrucción es siempre la misma. No se hace ninguna suposición acerca de la naturaleza de la señal a cuantizar, de ahí que no proporcionen los mejores resultados. Sin embargo, tienen como ventaja que son los más fáciles y menos costosos de implementar.

En la siguiente figura se ve un ejemplo de cuantización uniforme:

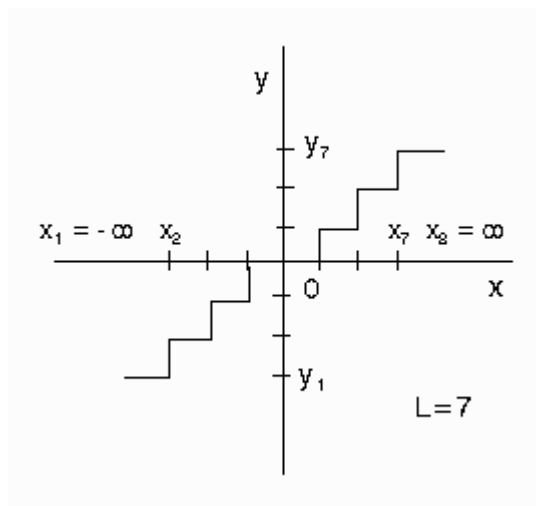


Figura. 1.15. Cuantización Uniforme

### 1.2.1.2.2. Cuantización logarítmica.

Las señales de voz pueden tener un rango dinámico superior a los 60 dB, por lo que para conseguir una alta calidad de voz se deben usar un elevado número de niveles de reconstrucción. Sin embargo, interesa que la resolución del cuantificador sea mayor en las partes de la señal de menor amplitud que en las de mayor amplitud. Por tanto, en la cuantización lineal se desperdician niveles de reconstrucción y, consecuentemente, ancho de banda. Esto se puede mejorar incrementando la distancia entre los niveles de reconstrucción conforme aumenta la amplitud de la señal.

Un método sencillo para conseguir esto es haciendo pasar la señal por un compresor logarítmico antes de la cuantización. Esta señal comprimida puede ser cuantizada uniformemente. A la salida del sistema, la señal pasa por un expansor, que realiza la función inversa al compresor. A esta técnica se le llama **compresión**. Su principal ventaja es que es muy fácil de implementar y funciona razonablemente bien con señales distintas a la de la voz.

Para llevar a cabo la compresión existen dos funciones muy utilizadas: Ley-A (utilizada principalmente en Europa) y ley- $\mu$  (utilizada en EEUU).

**Ley-A :**

$$c(x) = \operatorname{sgn}(x) \frac{\ln A|x|}{1 + \ln A|x|} \quad \text{para } \frac{1}{A} \leq |x| \leq 1$$

$$c(x) = \operatorname{sgn}(x) \frac{A|x|}{1 + \ln A|x|} \quad \text{para } 0 \leq |x| \leq \frac{1}{A}$$

$$A = 87.6$$

**Ley- $\mu$  :**

$$c(x) = \operatorname{sgn}(x) \frac{\ln(1 + m|x|)}{\ln(1 + m)} \quad \text{para } -1 \leq |x| \leq 1$$

$$m = 255$$

*En la mayoría de los sistemas telefónicos, A se fija a 87.56 y  $\mu$  a 255.*

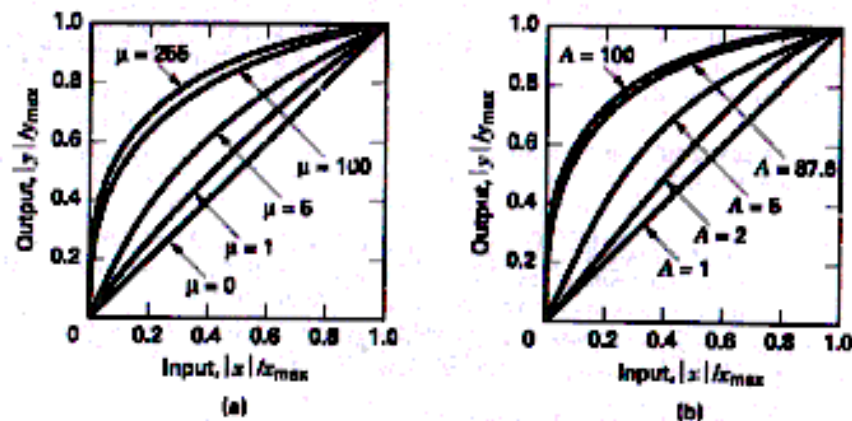
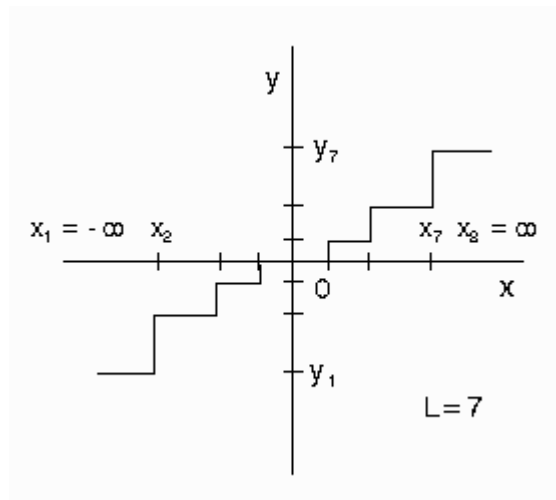


Figura. 1.16. Características de Compresión, (a) Ley U, (b) Ley A

### 1.2.1.2.3. Cuantización no uniforme.

El problema de la cuantización uniforme es que conforme aumenta la amplitud de la señal, también aumenta el error. Este problema lo resuelve el cuantizador logarítmico de forma parcial. Sin embargo, si conocemos la función de la distribución de probabilidad, podemos ajustar los niveles de reconstrucción a la distribución de forma que se minimice el error cuadrático medio. Esto significa que la mayoría de los niveles de reconstrucción se

den en la vecindad de las entradas más frecuentes  $y$ , consecuentemente, se minimice el error (ruido).

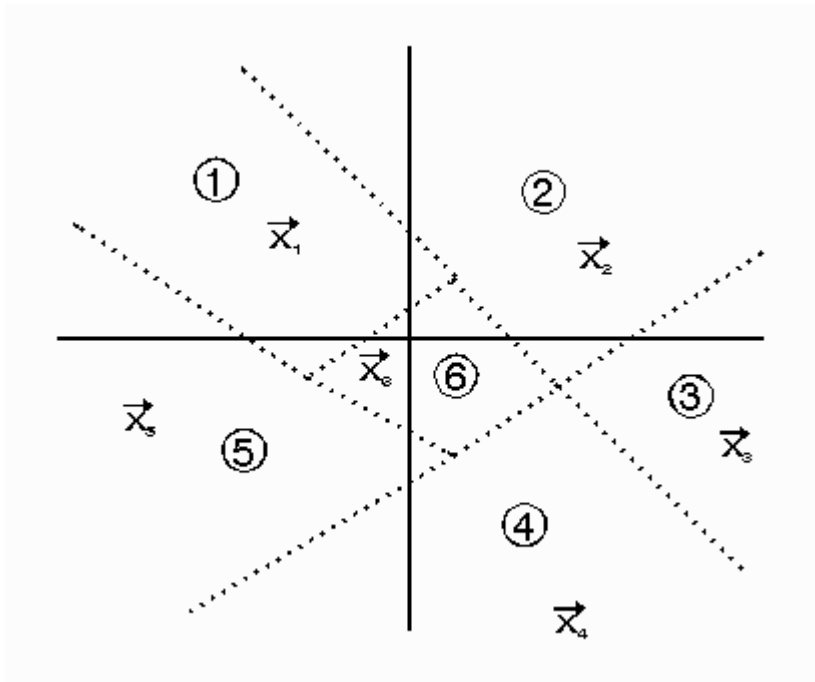


**Figura. 1.17. Cuantización No Uniforme**

En la práctica, se puede usar una estimación de la distribución para diseñar los cuantizadores. Esta estimación se puede obtener a partir de los datos a cuantizar de forma iterativa.

#### 1.2.1.2.4. Cuantización vectorial.

En los métodos anteriores, cada muestra se cuantizaba independientemente a las muestras vecinas. Sin embargo, la teoría demuestra que ésta no es la mejor forma de cuantizar los datos de entrada. Resulta más eficiente cuantizar los datos en bloques de  $N$  muestras. El proceso es sencillamente una extensión de los anteriores métodos escalares descritos anteriormente. En este tipo de cuantización, el bloque de  $N$  muestras se trata como un vector  $N$ -dimensional.



**Figura. 1.18. Cuantización Vectorial en Dos Dimensiones**

El plano XY está dividido en seis regiones distintas. El vector de entrada (con dos componentes) se reemplaza por el centroide  $i$  (representa todos los vectores de una determinada región  $i$ ) de la región a la que pertenece.

La cuantización vectorial ofrece mejores resultados que la cuantización escalar, sin embargo, es más sensible a los errores de transmisión y lleva consigo una mayor complejidad computacional.

#### 1.2.1.2.5. Error de Cuantización.

El proceso de cuantización genera una diferencia entre la señal original  $x(nts)$  y la cuantizada  $x_q(nts)$ . La medida de esta diferencia se llama error o ruido de cuantización

$$\mathbf{e} = x_q(nts) - x(nts)$$

Por eso se acostumbra calcular una relación señal a ruido de cuantización dada por:

$$\left(\frac{S}{N}\right)_Q = \frac{E[x^2]}{E[e^2]}$$

Donde

$E[x^2]$ =Potencia de Señal

$E[e^2]$ =Potencia del error o potencia de ruido de cuantización.

La potencia de ruido de cuantización se podría calcular como:

$$E[e^2] = \int_{-\infty}^{\infty} e^2 p_e(e) de$$

Sin embargo la fdp del ruido no se conoce, por lo que podemos referirla a la distribución probabilística de la señal o mensaje  $x(t)$ . Para facilitar el cálculo se dividirá la integral en  $M$  intervalos resultando:

$$E[e^2] = \sum_{k=1}^M \int_{x_k - 0.5a}^{x_k + 0.5a} (x(nt_s) - x_k)^2 p_x(x) dx =$$

$$\int_{x_1 - 0.5a}^{x_1 + 0.5a} (x(nt_s) - x_1)^2 p_x(x) dx + \dots + \int_{x_M - 0.5a}^{x_M + 0.5a} (x(nt_s) - x_M)^2 p_x(x) dx$$

Si se ha elegido  $M$  suficientemente grande, se puede considerar que  $p_x(x)$  es constante dentro de cada subintervalo y por ende se puede sacar fuera de la integral. Además se hace en cada una el siguiente cambio de variable  $x(nt_s) - x_k = y$

$$E[e^2] = \sum_{k=1}^M p_x(x) \int_{-0.5a}^{0.5a} y^2 dy = \sum_{k=1}^M p_x(x) \frac{a^3}{12} = \frac{a^2}{12} \sum_{k=1}^M a p_x(x)$$

$$\text{pero } \sum_{k=1}^M a p_x(x) = 1$$

$$E[e^2] = \frac{a^2}{12}$$

Por lo tanto la relación señal a ruido de cuantización resulta:

$$\left(\frac{S}{N}\right)_Q = \frac{E[x^2]}{\frac{a^2}{12}} = 12 \frac{E[x^2]}{a^2}$$



Ahora, como el pico de potencia de una señal analógica (normalizada a 1?) se puede expresar como

$$V_p^2 = \left( \frac{V_{pp}^2}{2} \right) = \left( \frac{M \cdot a}{2} \right)^2 = \frac{M^2 \cdot a^2}{4}$$

Y esto a su vez combinado con la ecuación anterior en

$$\left( \frac{S}{N} \right)_q = \frac{M^2 \cdot a^2 / 4}{a^2 / 12} = 3M^2$$

Así, la relación Señal a Ruido de Cuantización mejora en función del número de niveles empleados. Es decir, que si **M** tiende a infinito, el error de cuantización tiende a cero.

### Ejemplo:

**Determine la relación señal a ruido de cuantización para un mensaje uniformemente distribuido entre  $-A$  y  $A$ .**

En primer lugar un mensaje uniformemente distribuido entre  $-A$  y  $A$ , tiene  $p_x(x) = 1/2A$

Por otra parte para un cuantizador uniforme de  $M$  niveles se cumpliría que  $2A = Ma$ .

$$E[x^2] = 2 \int_{-A}^A \frac{1}{2A} x^2 dx = \frac{1}{A} \frac{A^3}{3} = \frac{A^2}{3} = \frac{a^2 M^2}{12}$$

$$\left( \frac{S}{N} \right)_q = M^2$$

Si se codifica en binario  $M=2^n$

$$10 \log \left( \frac{S}{N} \right)_q = 10 \log M^2 = 10 \log 2^{2n} = 20n \log 2 = 6n$$

Por ejemplo cada bit adicional que se use producirá 6 dB de ganancia en relación señal a ruido.

### 1.2.1.3. Codificación

Después de realizada la cuantización, los valores de las muestras de voltajes se representan numéricamente por medio de códigos y estándares previamente establecidos. Lo más común es codificar la señal digital en código numérico binario, pero también existen otros tipos de códigos que son empleados. En este paso  $M=2^n$ , donde  $n$  es el número de bits y  $M$  el número de niveles de cuantización. Es decir por cada muestra que se toma cada  $t_s$ , se deben transmitir  $n$  bits; por lo tanto esto equivale a tener una frecuencia de muestreo más rápida ( $n f_s$ ) y por consiguiente u mayor ancho de banda.

**La codificación permite asignarle valores numéricos binarios equivalentes a los valores de tensiones o voltajes que conforman la señal eléctrica analógica original.**

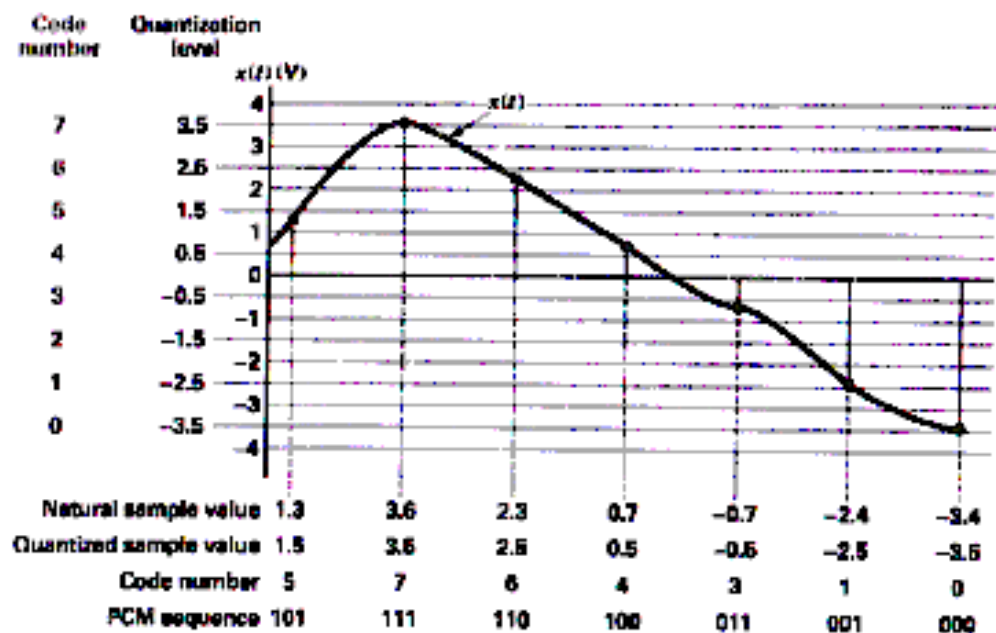


Figura. 1.19. Muestras Naturales, Muestras Cuantizadas y PCM (Codificadas)

A continuación se presenta una tabla donde se representan los números del 0 al 7 con su respectivo código binario. Como se ve, con 3 bits, podemos representar ocho estados o niveles de cuantización.

Número	Código binario
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

**Tabla. 1.1. Códigos Formados con Tres Bits**

### 1.2.2. CODIFICADORES DE FORMA DE ONDA: PCM, DPCM, ADPCM

Los codificadores de voz se clasifican en tres grandes grupos, a saber:

- Codificadores de forma de onda.
- Vocoders.
- Codificadores híbridos.

Los primeros de esta clasificación son aquellos codificadores que intentan reproducir la forma de la onda de la señal de entrada sin tener en cuenta la naturaleza de la misma. Estos, en función del dominio en el cual operan, se dividen en: **codificadores en el dominio del tiempo y codificadores en el dominio de la frecuencia.**

Este tipo de codificadores proporcionan una alta calidad de voz a bit rates medios, del orden de 32 kb/s. Sin embargo, no son útiles cuando se quiere codificar a bajos bit rates. En el grupo de vocoders están aquellos codificadores que sí tienen en cuenta la naturaleza de la señal a codificar, en este caso la voz, y aprovechan las características de la misma para ganar en eficiencia. Permiten trabajar con muy bajos bit rates, pero la señal de voz que producen suena demasiado sintética.

En el tercer grupo, los codificadores híbridos, encontramos aquellos que combinando técnicas de los vocoders y de los codificadores de la forma de la onda aúnan las ventajas de ambos, permitiendo una alta calidad de voz a bajos bit rates.

Los codificadores de la forma de onda intentan reproducir la forma de la onda de la señal de entrada. Generalmente se diseñan para ser independientes a la señal, de tal forma que pueden ser usados para codificar una gran variedad de señales. Presentan una degradación aceptable en presencia de ruido y errores de transmisión. Sin embargo, para que sean efectivos, sólo se deben usar a bit-rates medios.

### **1.2.2.1. Codificadores en el Dominio del Tiempo**

Dentro de este grupo tenemos los siguientes codificadores: PCM, DPCM, ADPCM

#### **1.2.2.1.1. Modulación por Codificación de Pulsos.**

La modulación por codificación de impulsos es la codificación de forma de onda más sencilla. Básicamente, consiste en el proceso de cuantificación. Cada muestra que entra al codificador se cuantifica en un determinado nivel de entre un conjunto finito de niveles de reconstrucción. Cada uno de estos niveles se hace corresponder con una secuencia de dígitos binarios, y esto es lo que se envía al receptor. Se pueden usar distintos criterios para llevar a cabo la cuantificación, siendo el más usado el de la cuantificación logarítmica.

#### **1.2.2.1.2. Modulación por Codificación de Pulsos Diferencial (DPCM).**

Puesto que PCM no tiene en cuenta la forma de la onda de la señal a codificar, funciona muy bien con señales que no sean las de la voz, sin embargo, cuando se codifica voz hay una gran correlación entre las muestras adyacentes.

Esta correlación puede aprovecharse para reducir el bit-rate y una forma sencilla de hacerlo es transmitir solamente las diferencias entre las muestras. Esta señal de diferencia

tiene un rango dinámico mucho menor que el de la voz original, por lo que podrá ser cuantificada con un número menor de niveles de reconstrucción. En la figura siguiente se muestra el funcionamiento de DPCM, donde la muestra anterior se usa para predecir el valor de la muestra actual:

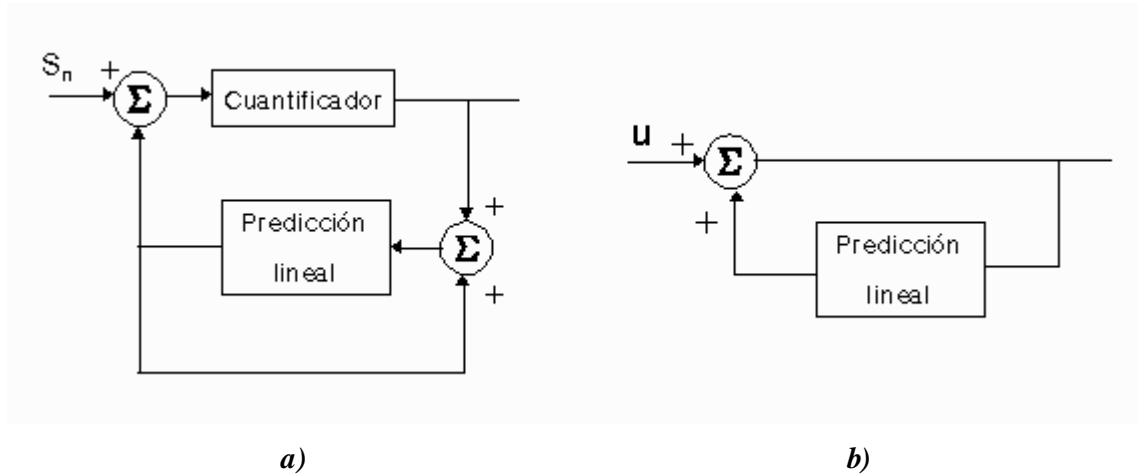


Figura. 1.20. Sistema DPCM a) Codificador, b) Decodificador

Normalmente, el valor predicho,  $s'_n$ , es una combinación lineal de un número finito de muestras anteriores,  $s_n$ :

$$s'_n = \sum_{k=1}^p a_k s_{n-k}$$

y

$$d_n = s_n - s'_n$$

La señal de diferencia,  $d_n$ , se denomina *residuo* y es el residuo lo que se cuantifica y se envía al receptor. Los coeficientes de predicción,  $\{a_k\}$ , se eligen para minimizar el error cuadrático medio,  $E$ :

$$E = \sum_n (s_n - s'_n)^2$$

### 1.2.2.1.3. Modulación por Codificación de Pulsos Diferencial Adaptativa (ADPCM).

En DPCM tanto el predictor como el cuantificador permanecen fijos en el tiempo. Se podría conseguir una mayor eficiencia si el cuantificador se adaptase a los cambios del residuo de predicción. Además, también se podría hacer que la predicción se adaptase a la señal de la voz. Esto aseguraría que la raíz cuadrada del error de predicción se minimice continuamente, con independencia de la señal de voz y de quién la emita.

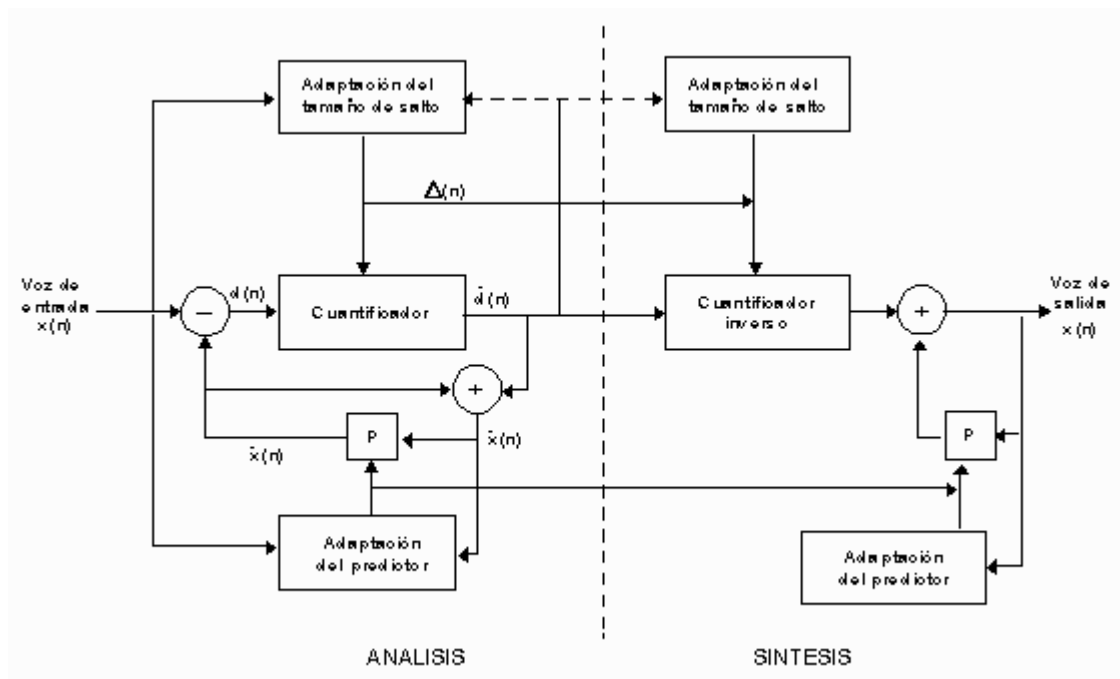


Figura. 1.21. Codificador / Decodificador ADPCM

Hay dos métodos para adaptar los cuantificadores y los predictores, llamados adaptación en **feedforward** y adaptación en **feedbackward**. En la adaptación feedforward los niveles de reconstrucción y los coeficientes de predicción se calculan en el emisor, usando un bloque de voz. Después son cuantificados y transmitidos al receptor como información lateral. Tanto el emisor como el receptor usan estos valores cuantificados para hacer las predicciones y cuantificar el residuo. En la adaptación feedbackward los niveles de reconstrucción y los coeficientes de predicción se calculan a partir de la señal codificada. Puesto que la señal es conocida tanto por el emisor como por el receptor, no

hay necesidad de transmitir información lateral, así el predictor y el cuantificador pueden actualizarse para cada muestra. La adaptación feedbackward puede dar menores bit rates, pero es más sensible a los errores de transmisión que la adaptación feedforward.

ADPCM es muy útil para codificar voz a bit rates medios. La CCITT propone un estándar de codificación de voz telefónica a una velocidad de 32 kb/s. Es el estándar G.721. Usa un esquema de adaptación feedbackward tanto para el cuantificador como para el predictor. El predictor tiene dos polos y seis ceros, por lo que produce una calidad de salida aceptable para señales que no son de voz.

### 1.2.2.2. Codificación en el Dominio de la Frecuencia

Este tipo de codificadores dividen la señal en distintas componentes en frecuencia y codifican cada una de éstas de forma independiente. El número de bits usados para codificar cada componente en frecuencia puede variar dinámicamente.

La codificación en este dominio puede ser: en sub-bandas o, por transformada.

#### 1.2.2.2.1. Codificación en Sub-Bandas.

Es el más sencillo de los métodos en el dominio de la frecuencia.

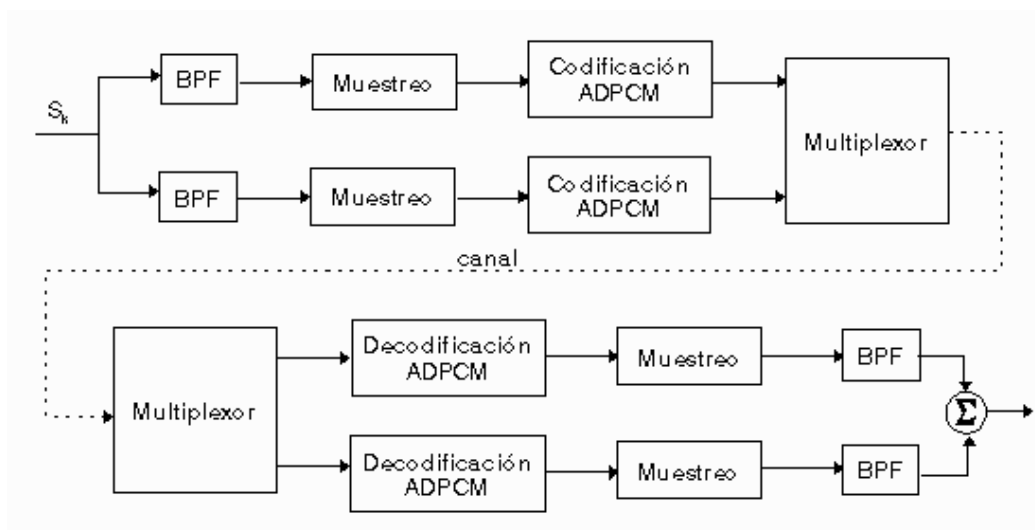


Figura. 1.22. Codificador / Decodificador Sub Bandas

La señal atraviesa un conjunto de filtros paso-banda (BPF). Después, cada sub-banda se pasa a banda baja y se realiza un proceso de decimación, es decir, se quitan muestras. Las sub-bandas se codifican usando algún método basado en el dominio del tiempo. El número de bits asignados a cada banda pueden variar en función de la importancia de dicha banda. En el receptor, se añaden muestras y se vuelven a modular las bandas a sus posiciones originales. Al final, se suman para obtener la señal de voz de salida.

La principal ventaja de la codificación en sub-bandas es que el ruido de cuantificación que se produce en cada banda queda confinado a la misma. La codificación en sub-bandas se usa mucho en señales de un gran ancho de banda, como puede ser en teleconferencia. Mayor información sobre esta técnica se encuentra referida en el estándar G.722 de la CCITT.

#### 1.2.2.2. Codificación por Transformada.

Consiste en una codificación por bloques en donde la señal de entrada se transforma en un dominio diferente y se codifican los coeficientes de la transformación. En el receptor, el decodificador calcula la transformada inversa para obtener la señal original reconstruida.

La transformación más usada es la Transformada Discreta del Coseno, DCT, cuya representación es:

$$x_c(k) = \sum_{n=0}^{N-1} x(n)g(k)\cos\left[\frac{(2n+1)kp}{2N}\right] \quad k = 0, 1, \dots, N-1$$

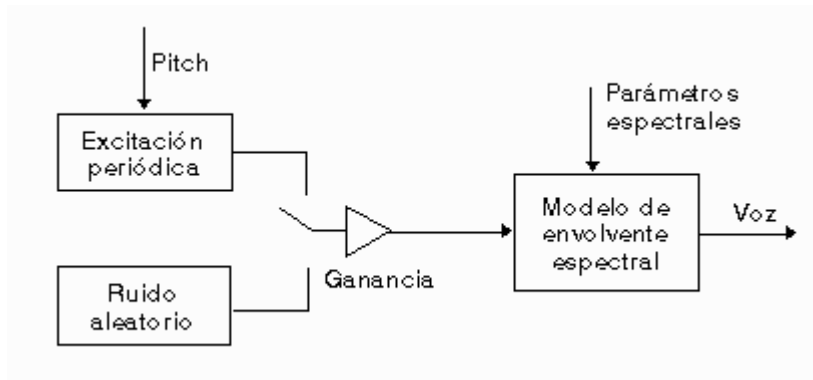
$$x(n) = (1/N) \sum_{k=0}^{N-1} x_c(k)g(k)\cos\left[\frac{(2n+1)kp}{2N}\right] \quad n = 0, 1, \dots, N-1$$

La codificación por transformada se utiliza en la codificación de señales de banda ancha de imagen y sonido. Sin embargo, no se usa mucho en codificación de voz debido a su complejidad.



### 1.2.3. CODIFICADORES PARAMÉTRICOS: CELP, VSELP

Los codificadores de la forma de la onda no tienen en cuenta la naturaleza de la señal a codificar. Sin embargo, si codificamos una señal de voz, podemos aprovechar sus características intrínsecas para que la codificación se realice de forma más eficiente. Así, los vocoders (VOICE CODERS) suponen el siguiente modelo de producción de voz:



**Figura. 1.23. Modelo de Producción de Voz**

Los vocoders intentan producir una señal que suene como la voz original, independientemente de si la forma de onda se parece o no. En el transmisor se analiza la voz y se extraen los parámetros del modelo y la excitación. Esta información se envía al receptor donde se sintetiza la voz. El resultado es que se produce voz inteligible a muy bajo bit-rate, pero tiene el problema de que no suena natural.

#### 1.2.3.1. Vocoder por Predicción Lineal

Es el tipo de vocoder más utilizado, utiliza el mismo modelo de producción que otros vocoders pero difiere en la determinación del modelo del tracto vocal. Supone que el tracto vocal se puede describir por un filtro todo polos de respuesta impulsiva infinita (filtro IIR),  $H(z)$ .

$$H(z) = \frac{G}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_p z^{-p}}$$

A este filtro se le conoce también como filtro LPC (Lineal Predictive Coding). Es decir, se supone que cada muestra es una combinación lineal de las muestras anteriores. Los coeficientes del filtro se calculan para minimizar el error entre la muestra actual y su predicción.

En este vocoder, se trabaja sobre bloques de 20 ms de voz. Se trabaja sobre lo que se conoce como modelo corto, las características de la voz se suponen que no varían con el tiempo en intervalos pequeños. Esos bloques se analizan para determinar los coeficientes de predicción. Estos se cuantifican y se envían al receptor junto a otros parámetros. El efecto de la predicción es similar a la correlación entre muestras adyacentes.

La popularidad de este vocoder viene de que el modelo todo polos del tracto vocal funciona muy bien y es posible alcanzar una señal de voz sintetizada muy inteligible a bit-rates del orden de 2.4 Kbps.

### 1.2.3.2. Vocoder LPC-10. LPC-10 (FS-1015)

Este algoritmo de codificación de voz fue desarrollado por el DOD (Department Of Defense), el Departamento de Defensa Estadounidense. Permite la codificación de la señal de la voz a una velocidad de 2400 bps.

La siguiente figura es un mapa sensitivo, seleccione qué parte de la codificación desea estudiar:

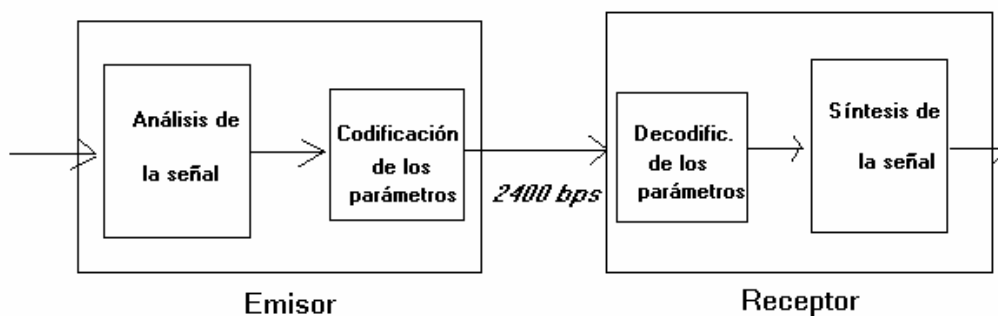


Figura. 1.24. Vocoder LPC – 10

### 1.2.3.2.1. Emisor.

El emisor se divide en dos bloques, que tienen las siguientes funciones: fase de análisis y codificación de parámetros.

La fase de análisis es la encargada de la extracción del "pitch" y del tipo de sonido (sonoro o sordo), así como de la extracción de los coeficientes de reflexión y la energía de la señal para cada una de las tramas en que dicha señal es dividida. Como coeficientes del filtro LPC se utilizan los coeficientes de reflexión. Los coeficientes  $\{a_k\}$   $k=1, \dots, 10$  presentan unas características que los hacen inadecuados para su transmisión.

La fase de codificación es la encargada de codificar estos parámetros y convertirlos en la secuencia de bits a transmitir.

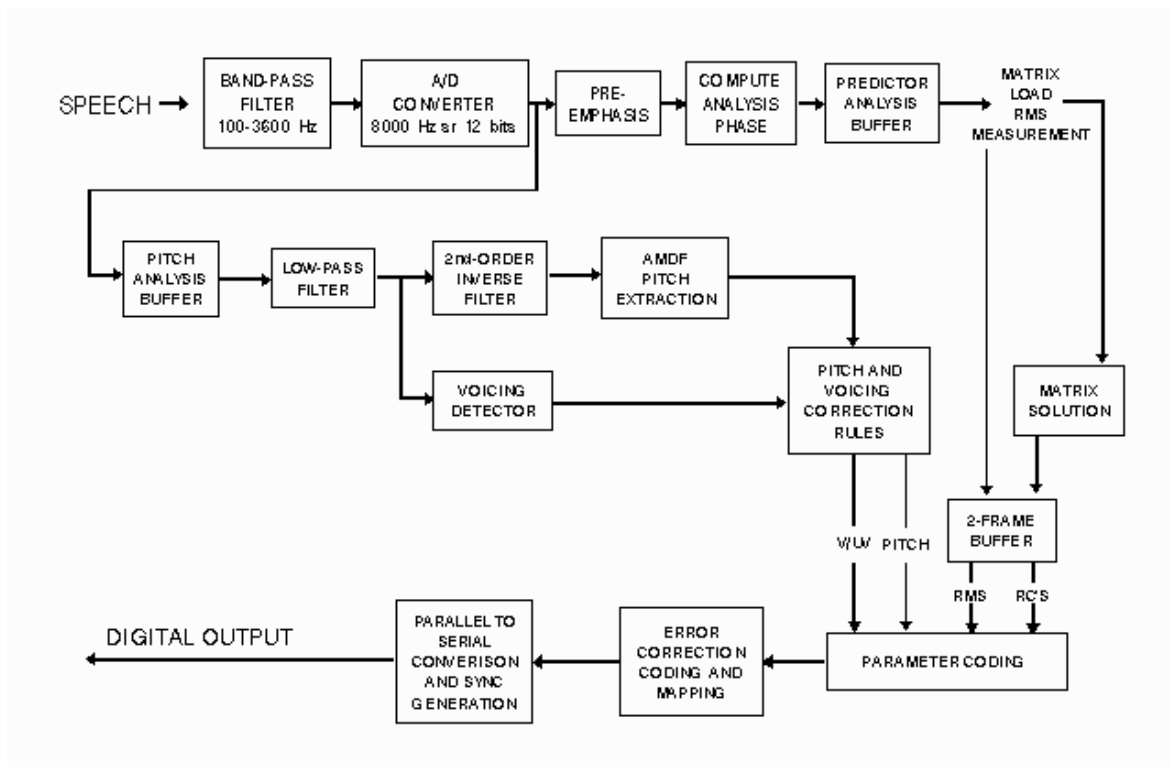


Figura. 1.25. Emisor LPC – 10

A la señal de voz se le aplica un preprocesado: atraviesa un filtro paso banda, y se realiza un pre-énfasis (el pre-énfasis reduce los requerimientos de precisión de los cálculos

posteriores). Además, con el pre-énfasis se consigue modelar mejor las amplitudes bajas de los formantes a altas frecuencias.

Después, en el análisis LPC, se calculan los coeficientes LPC con el método de la covarianza. La voz atraviesa una serie de filtros con el fin de mejorar el pitch. Después, la señal de voz se transfiere a la fase del algoritmo AMFD (Average Magnitude Difference Function) y al detector de voz (para extraer el pitch y decidir entre sonoro/sordo). Los resultados de este detector, junto con los del pitch, son corregidos y suavizados por un algoritmo de programación dinámica. Este proceso se realiza dos veces por cada trama. Los 10 coeficientes de reflexión, RMS (Root Mean Square, mide la energía del segmento, se calcula en el pitch) y el pitch se codifican y se obtiene un bit-rate de 2400 bps.

### 1.2.3.2.2. Receptor.

En el receptor, lo que se hace es reconstruir la voz simulando la excitación sonora por algún tipo de tren de impulsos repetido periódicamente a la frecuencia del "pitch" y la sorda simplemente como ruido. La señal de excitación así formada se pasa posteriormente por un filtro similar al utilizado en la fase de análisis y que es el encargado de simular el tracto vocal. La salida de dicho filtro es escalada en función de la energía RMS para obtener una reconstrucción de la señal original.

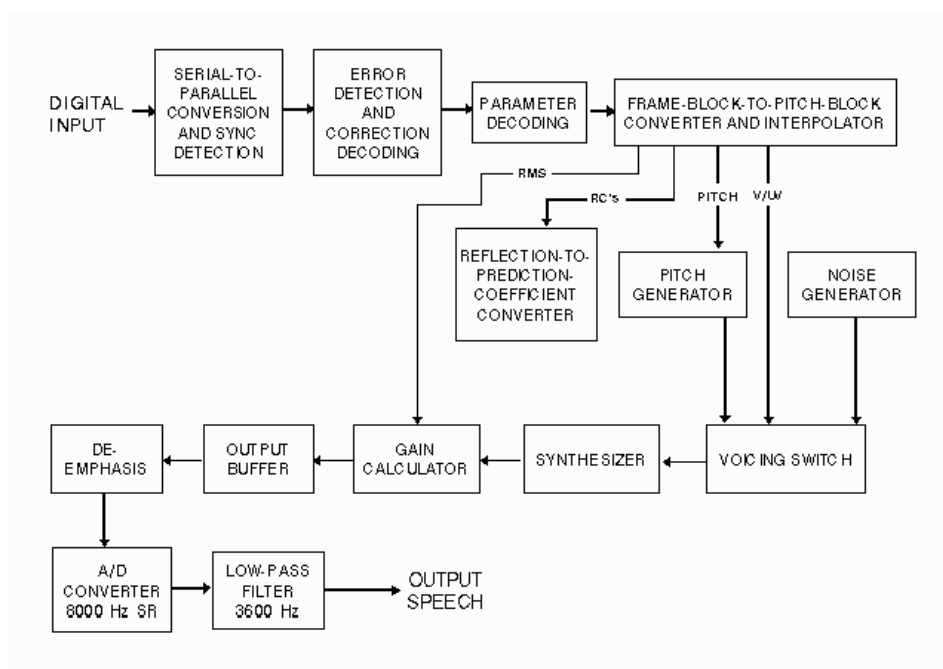


Figura. 1.26. Receptor LPC – 10

Tras la conversión de serie a paralelo, se realiza la detección y corrección de errores, así como la decodificación mediante tablas, se introduce una trama de retraso para poder suavizar la trama actual basándose en la mediana entre la trama presente, la pasada y la futura. Dicha suavización varía dependiendo de la tasa de error, a medida que ésta aumenta es necesario aumentar también la suavización.

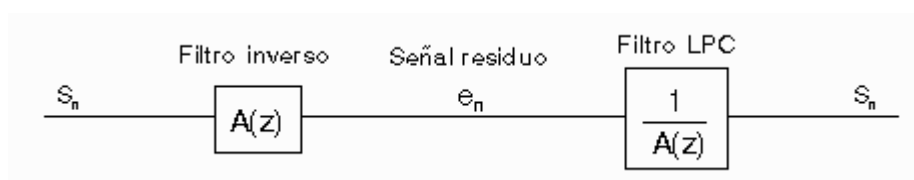
### 1.2.3.3. Codificadores Híbridos.

En la codificación híbrida se combinan las técnicas de los codificadores de la forma de la onda con las de los vocoders con el propósito de obtener una alta calidad de voz a bajos bit-rates (inferiores a 8 Kb/s). En estos codificadores, las muestras de la señal de entrada se dividen en bloques de muestras (vectores) que son procesados como si fueran uno solo. Llevan a cabo una representación paramétrica de la señal de voz para tratar que la señal sintética se parezca lo más posible a la original.

También se les conoce como codificadores de análisis-por-síntesis. En el emisor se lleva a cabo un análisis que obtiene los parámetros de la señal para luego sintetizarla y conseguir el mayor parecido a la original.

#### 1.2.3.3.1. Codificación RELP.

Cuando la señal de voz atraviesa un predictor lineal, se elimina la correlación entre tramas. Si la predicción es bastante buena, la salida del predictor será aproximadamente ruido blanco, espectralmente plano, como se muestra en la siguiente figura:



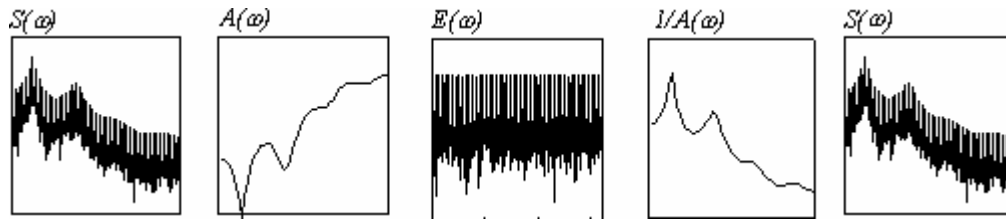


Figura. 1.27. Codificación REL

El residuo contiene toda la información de excitación y cualquier información que el filtro LPC no considera. La idea de RELP es que una pequeña parte del residuo se transmite y a partir de él reconstruir el residuo completo en el receptor. Se ve que el filtro LPC no elimina la información del pitch, y esta información no se modela correctamente por el residuo transmitido. Proporciona buena calidad para velocidad de transmisión de 9.6 kps.

#### 1.2.3.3.2. Codificación Multipulso, MPC.

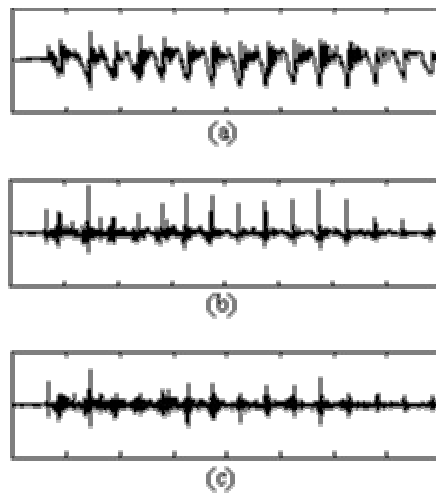
Los vocoders categorizan la voz en sonoro o no sonoro pero no consideran una mezcla de ambos sonidos y de ahí su baja calidad. El codificador multipulso trata de corregir este problema. El periodo de pitch produce una correlación de retardo largo que no se elimina con el filtro LPC.

Esta correlación se puede eliminar pasando el residuo a través de un segundo predictor lineal, llamado predictor de pitch o filtro de pitch. Este filtro es del tipo:

$$P(z) = 1 - \sum_i b_i z^{-M-i}$$

siendo  $M$  el periodo del pitch.

La salida de este predictor es aproximadamente ruido gaussiano.



**Figura. 1.28.** (a) Señal original. (b) Residuo del filtro LPC (aumentado en 10dB).  
(c) Residuo de los filtros LPC y de pitch en cascada (aumentado en 10dB)

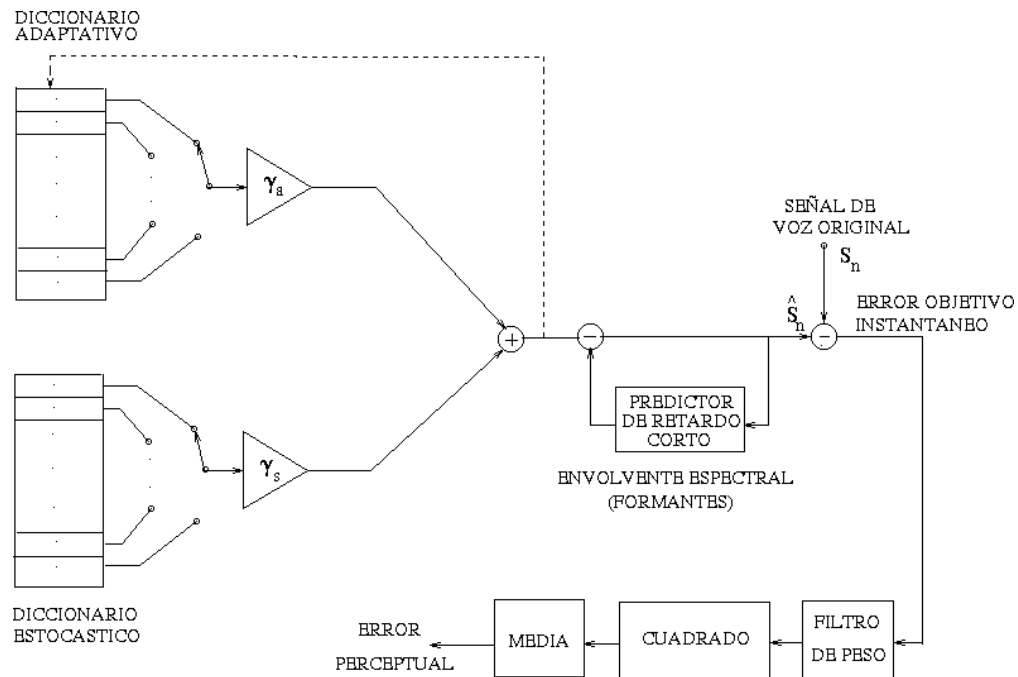
El codificador multipulso pasa como excitación a los dos filtros una serie de impulsos (sobre 4 o 6 impulsos). La posición y sus amplitudes se determinan utilizando un procedimiento de análisis-por-síntesis (se sintetiza la voz para cada uno de estos conjuntos de impulsos y el conjunto que produce el error menor entre la señal reconstruida y la original se elige como la excitación).

El codificador multipulso produce una señal de voz de muy buena calidad a bit-rates del orden de 9.6 kbps. Una variación de este codificador consiste en elegir como secuencia de excitación unos patrones de pulsos regularmente espaciados. Este codificador se conoce como RPE.

### 1.2.3.3.3. Introducción a la codificación CELP (FS-1016).

CELP se basa en procedimientos de búsqueda de análisis-por-síntesis, cuantización de vectores con pesos (VQ) y predicción lineal (LP). Se usa un filtro LPC de décimo orden para modelar las formantes de retardo corto de la señal de voz. La periodicidad de retardo largo de la señal se modela con un diccionario adaptativo VQ (también llamado *pitch*). El error de la predicción lineal de retardo corto y el *pitch* VQ se cuantifican usando un libro de secuencias estocásticas (son muestras de ruido aleatorio blanco gaussiano). Al receptor sólo se le envía como excitación el índice del diccionario y la ganancia.

### 1.2.3.3.1. Codificador.



**Figura. 1.29. Transmisor CELP**

En la figura 1.29 se muestra la etapa de análisis de un transmisor CELP en detalle. Este contiene una réplica del sintetizador del receptor (excepto el post-filtro). A la señal de entrada, se le resta la salida del sintetizador CELP, esta diferencia se pasa por un filtro perceptual. El error perceptual se utiliza después para realizar la búsqueda en el diccionario. El procedimiento de búsqueda encuentra las ganancias de los diccionarios estocásticos y adaptativos que minimizan el error cuadrático medio. El filtro de predicción lineal puede determinarse por técnicas de análisis de retardo corto de la señal de entrada.

El objetivo del filtro perceptual es atenuar aquellas frecuencias en las que el oído humano detecta menos los posibles errores y aumentarlas para aquellas en que el oído es más perceptivo. O sea, se lleva a cabo una comparación dando más importancia a aquellas zonas donde el oído aprecia más el ruido.



### 1.2.3.3.2. Decodificador.

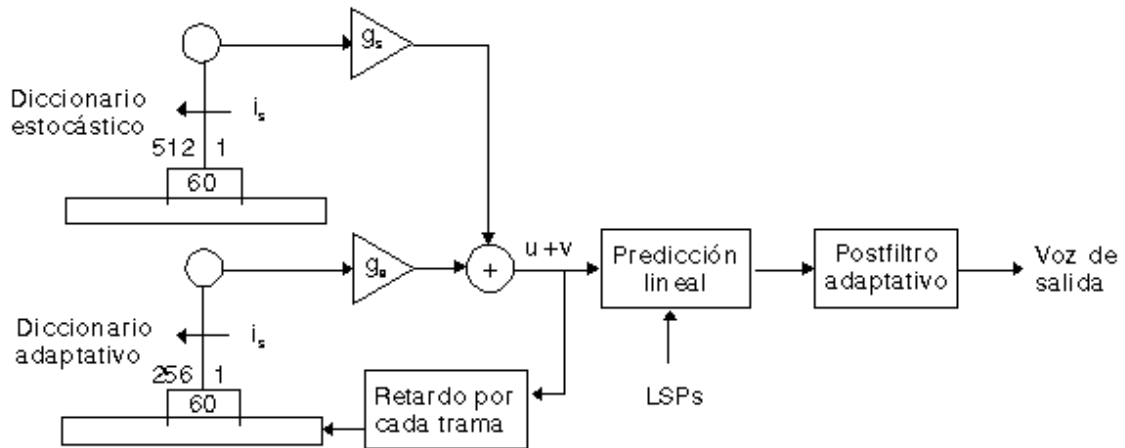


Figura. 1.30. Receptor CELP

El sintetizador CELP se basa en el principio del uso de un filtro de predicción lineal cuya ganancia se busca en una tabla. La excitación se genera usando, en paralelo, un diccionario estocástico (de valores aleatorios) fijo y otro diccionario que no es fijo, sino adaptativo. Después, se actualiza el diccionario adaptativo con esta excitación. En el receptor, se puede mejorar la calidad de la voz sintetizada añadiendo una fase de post-filtrado.

### 1.2.3.3.4. Codificación VSELP.

El codificador VSELP es el estándar en las comunicaciones celulares digitales en Estados Unidos. Permite el procesamiento de voz a un bit-rate de 7950 bps e incluso de 4.8 Kbps.

Este algoritmo es una variación del CELP. La diferencia radica en la forma y estructura del diccionario. El diccionario estocástico de CELP se convierte en VSELP en dos conjuntos de vectores.

La siguiente figura se corresponde con el decodificador VSELP:

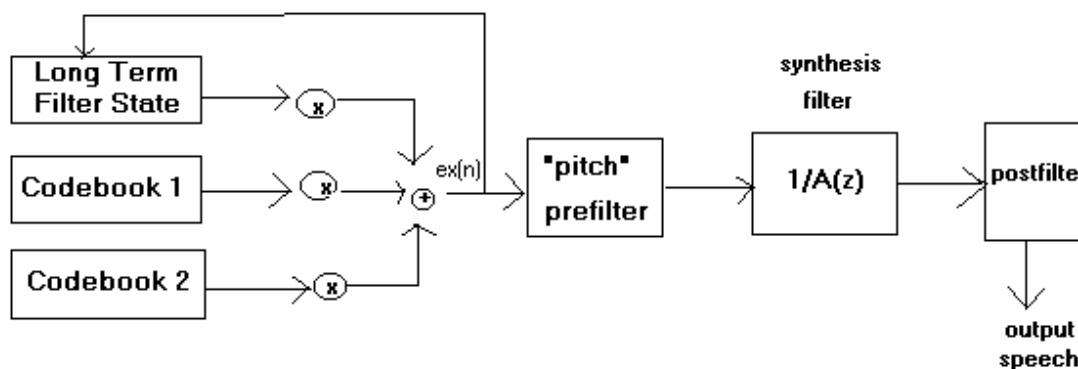
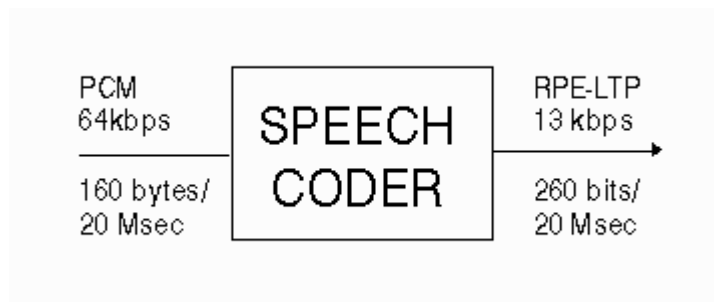


Figura. 1.31. Receptor

El codificador/decodificador VSELP utiliza hasta un total de tres fuentes de excitación. La primera es el diccionario adaptativo (igual que en CELP). Las dos restantes son los dos (o uno) diccionarios VSELP. En las implementaciones de 8 Kbps se usan dos diccionarios VSELP, cada uno de los cuales contiene 128 vectores. A 4.8 Kbps se usa un solo diccionario que contiene 2048 vectores. Los diccionarios VSELP, debido a su estructura, son muy robustos frente a posibles errores. Estos diccionarios VSELP contienen vectores que se calculan como combinación lineal de unos vectores básicos. Estas tres fuentes de excitación se multiplican por sus correspondientes ganancias y el resultado se suma, dando la secuencia combinada de excitación  $ex(n)$ . Después, en cada submuestra, se utiliza  $ex(n)$  para actualizar el diccionario adaptativo. El filtro de síntesis (synthesis filter en la figura) es un filtro LPC de décimo orden todo polos. Los coeficientes LPC se codifican una vez por cada trama de 20 mseg. Los parámetros de excitación se actualizan cada 5 mseg. En el codificador a 8 Kbps el número de muestras de cada trama (N) es 40. Finalmente, el post-filtro espectral sirve para mejorar la calidad de la señal sintetizada.

#### 1.2.3.3.5. RPE-LTP.

Este algoritmo de codificación, *RPE-LTP* (Regular Pulse Excitation - Long Term Prediction), es el que se usa en las comunicaciones móviles GSM.



**Figura. 1.32. Proceso RPE – LTP**

La trama o segmento de voz de entrada consta de 160 muestras de señal (muestras PCM uniforme de 13 bits). A estas muestras se le aplica un preprocesado consistente en generar una señal libre de offset y posteriormente se pasan por un filtro de primer orden de preénfasis. La salida obtenida se analiza para obtener los coeficientes del filtro de retardo corto, STP (análisis LPC). Estos parámetros se utilizan para filtrar las 160 muestras. Los coeficientes del filtro, llamados coeficientes de reflexión, se transforman en las razones log-area, LAR, antes de transmitirlos. Para las operaciones siguientes, el segmento de voz se subdivide en 4 sub-segmentos con 40 muestras de la señal residuo del filtro de retardo corto cada uno.

El siguiente paso es calcular los parámetros del filtro de retardo largo (LTP), el pitch (retardo del filtro LTP) y la ganancia, en la etapa de análisis LTP. El bloque resultante de 40 muestras pasa al bloque de análisis RPE. En este bloque las 40 muestras de entrada se representan por una de las 4 sub-secuencias de 13 pulsos cada una. Los parámetros RPE se utilizan en un decodificador local RPE para reconstruir las 40 muestras de la versión cuantizada de la señal residuo del filtro LTP. Sumando estas muestras al bloque anterior del residuo STP estimado, se consigue una versión reconstruida de la señal residuo actual del filtro STP.

Este bloque de señal residuo del filtro STP se pasa al bloque de análisis LTP para producir la nueva estimación de señal residuo STP que se va a utilizar en el siguiente sub-bloque, completando así el lazo de realimentación.

Según esto, se definen tres grupos de datos generados por el codificador: los parámetros del filtro STP, los parámetros del filtro LTP, los parámetros RPE.

### 1.2.3.3.5.1. Codificador RPE-LTP.

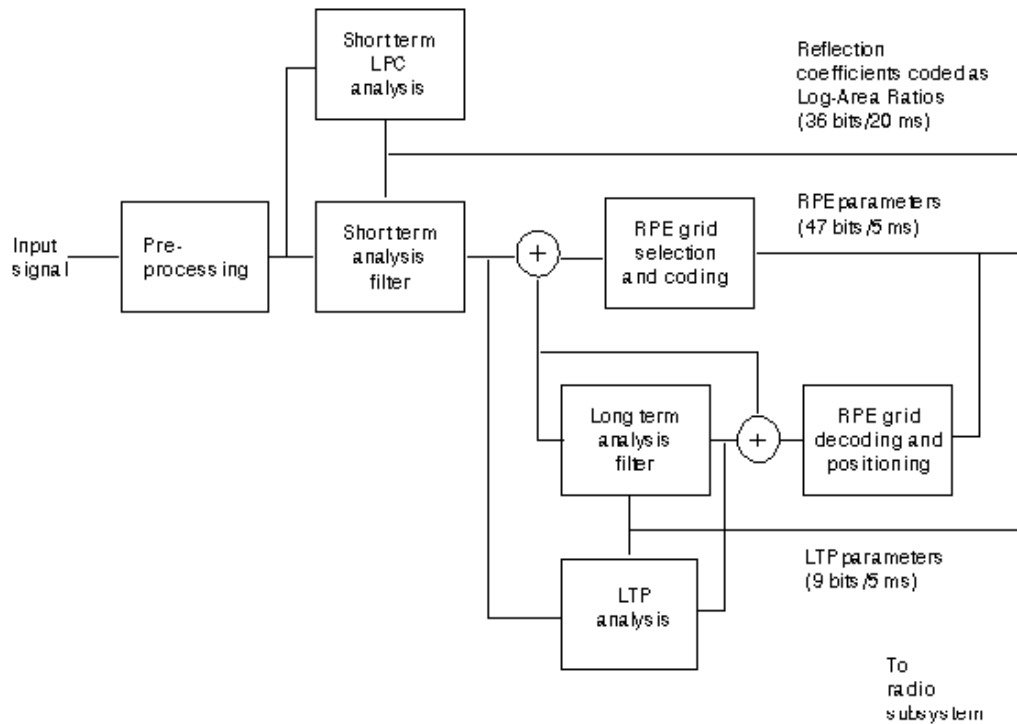


Figura. 1.33. Codificador RPE – LTP

El primer proceso que sufre la señal de voz es una compensación offset para producir una señal libre de offset. Esta es después pasada por un filtro de preénfasis de primer orden.

### 1.2.3.3.5.2. Análisis LPC.

**Segmentación:** La señal de voz  $s(k)$  se divide en segmentos no solapados de 20ms (160 muestras). A cada uno de estos segmentos se le aplica un análisis LPC de orden 8 (orden de predicción  $p=8$ ).

Como parámetros del filtro LPC se utilizan los coeficientes de reflexión,  $r(i)$ . Estos proceden de la teoría de que el tracto vocal se puede representar mediante una serie de secciones cilíndricas uniformes. Esta representación se puede describir por los coeficientes de reflexión de las razones de área de las secciones conectadas.

$$-1 \leq r(i) \leq +1$$

Los parámetros que se envían son una transformación de estos. Las razones Log-area definidos por:

$$\text{LAR}(i) = \log_{10} \left( \frac{1+r(i)}{1-r(i)} \right)$$

Una vez calculados se cuantizan los parámetros LAR. Para evitar transiciones espúreas, que pueden ocurrir si los coeficientes del filtro cambian abruptamente, se interpolan linealmente dos conjuntos consecutivos de parámetros LAR. Dentro de un segmento de 160 muestras se trabaja con 4 conjuntos de coeficientes diferentes. Los coeficientes de reflexión se calculan utilizando la transformación inversa.

#### 1.2.3.3.5.3. Filtro de predicción de retardo largo (LTP).

**Sub-segmentación:** El filtro de predicción de retardo largo se evalúa cuatro veces por segmento, para cada 5ms (40 muestras). Para cada sub-segmento se calcula el factor de desplazamiento de retardo largo (pitch) y un factor de ganancia asociado. Como el parámetro de pitch puede tomar valores entre 40 y 120 se necesitan 7 bits para codificarlo. El factor de ganancia se codifica con 2 bits. Si  $d(k)$  es la señal residuo del filtro STP, el residuo del filtro LTP,  $e$ , se calcula restando a  $d$  una estimación  $d''$ .  $d''$  se calcula antes a partir de la señal residuo STP reconstruida previamente, pero ajustada a los valores del sub-segmento actual. La estimación de la señal residuo del filtro LTP se denota  $e'$ , y se suma a la estimación  $d''$  para obtener el residuo STP reconstruido.

#### 1.2.3.3.5.4. Codificación RPE.

La señal residuo LTP se filtra con un filtro FIR con respuesta impulsiva dada por:

<b>i</b>	5	4 (6)	3 (7)	2 (8)	1 (9)	0 (10)
<b>H(i)*2<sup>13</sup></b>	8192	5746	2054	0	-374	-134

$$|H(\Omega=0)|=2.779$$

El propósito de este filtro perceptual es atenuar el espectro en frecuencia donde el error es perceptiblemente menos importante y amplificar aquellas zonas del espectro donde es perceptiblemente más importante. Con esto conseguimos una medida de error subjetiva significativa (propiedad de enmascaramiento del oído humano).

La señal filtrada se sub-muestra por un factor de 4, dando lugar a 4 secuencias entrelazadas de longitud 13. Se elige la secuencia de mayor energía como la representante de la excitación, secuencia RPE.

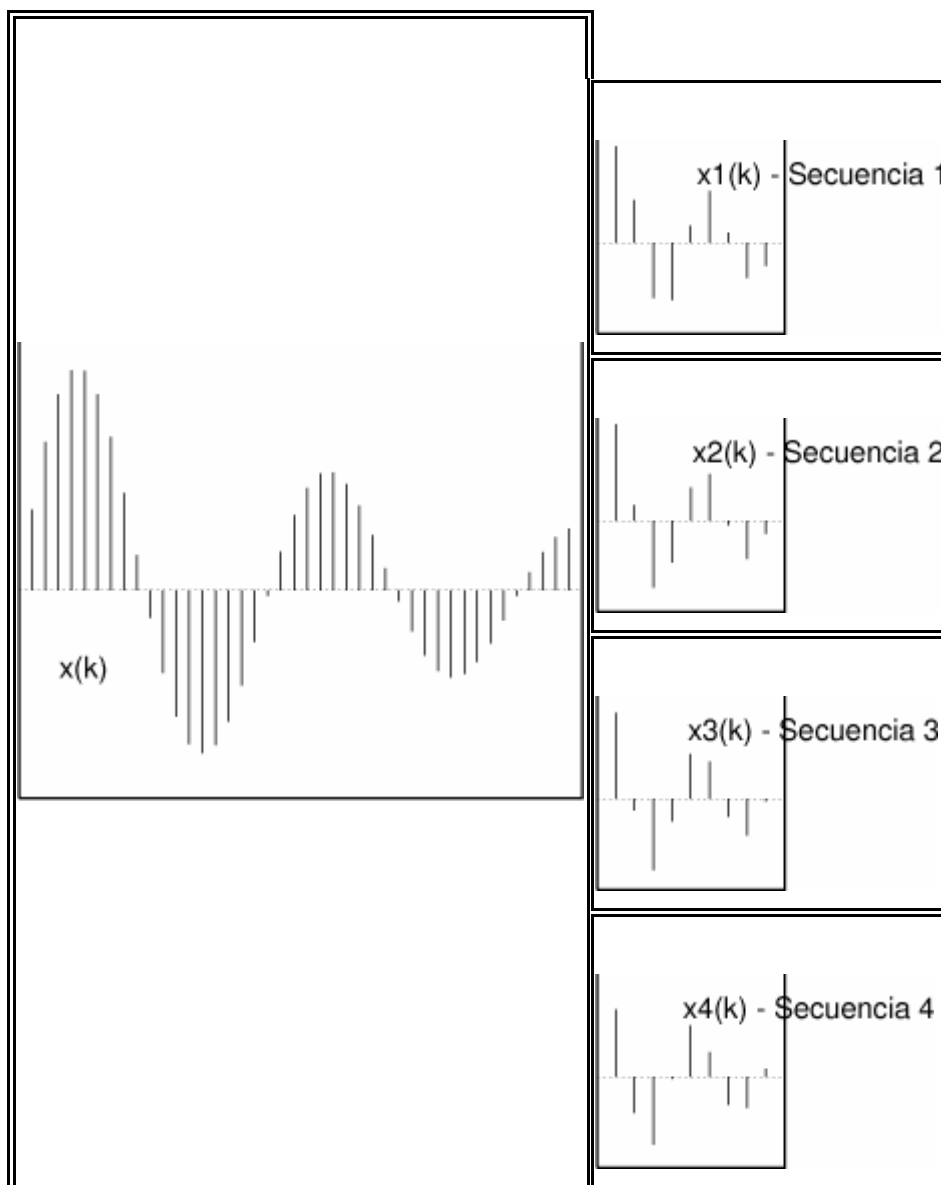


Figura. 1.34. Decimación RPE y Selección de Grid

Para cada secuencia, se selecciona el valor absoluto máximo  $x_{\max}$ , y se cuantiza de forma logarítmica utilizando seis bits. Las muestras se normalizan al valor cuantizado de la amplitud máxima y se cuantizan uniformemente con 3 bits.

### 1.2.3.3.5.5. Decodificador RPE-LTP.

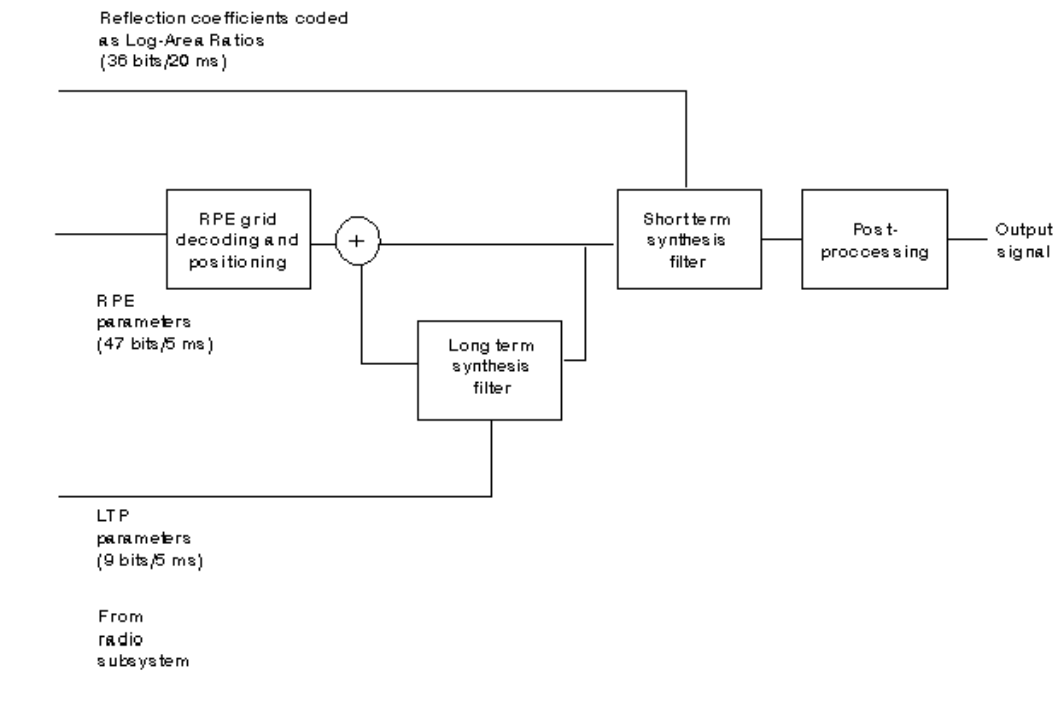


Figura. 1.35. Decodificador RPE – LTP

En el lado receptor se reciben los parámetros codificados y se procede a reconstruir la señal. Consta de las siguientes funciones: Decodificación RPE, Predicción de retardo largo, Filtrado de retardo corto, síntesis.

**Postprocesado:** La decodificación RPE consiste en decodificar y desnormalizar las muestras de la señal residuo LTP. Posteriormente se colocan en su posición temporal adecuada y la frecuencia de muestreo se incrementa por un factor de 3 insertando el resto de muestras como ceros.

Esta señal se aplica al filtro LTP y su salida forma la excitación del filtro STP. Por último, la salida del filtro STP se pasa por un filtro des-enfatizador.

### 1.3. PRÁCTICAS CON MATLAB

#### 1.3.1. AUTORRELACIÓN DE SEÑAL DE VOZ

##### 1.3.1.1. Objetivos

- Encontrar los 12 primeros retardos (lags) de la función de auto correlación de los fonemas de la palabra “**nueve**”.
- Realizar un análisis de autocorrelación entre fonemas.
- Utilizar ventanas adecuadas de análisis para las vocales y para las consonantes.

##### 1.3.1.2. Procedimiento

1. Definir el tamaño del bloque de análisis.
2. En cada bloque estimar los atrasos de la función de autocorrelación.
3. Estimar valor medio temporal.
4. Restar del valor medio estimado de todas las muestras del bloque.
5. estimar los atrasos  $r_x(0)$  a  $r_x(12)$ .
6. Normalizar: atraso /  $r_x(0)$ .
7. comparar los atrasos de la función de autocorrelación en bloques de un mismo fonema.

La señal de voz fue grabada con una  $f_s$  de 11025 Hz y se tomaron en total 5962 muestras. Cada muestra tiene 16 bits de PCM. Esto fue realizado con el grabador de sonidos de Windows® y editado con la ayuda del software de edición de audio **GoldWave 4.0**, aunque se puede utilizar cualquier editor de audio disponible. Con este último lo que se realizó fue la supresión del silencio inicial y final de la grabación inicial.



Propiedad	Valor
<b>Audio</b>	
Velocidad de transmisión	176kbps
Tamaño de muestra de sonido	16 bit
Canales	1 (mono)
Velocidad de muestra de sonido	11 KHz
Formato de audio	PCM

Figura. 1.36. Propiedades de Señal de Voz: Nueve

La señal de voz es la palabra nueve y su espectro muestra en el siguiente gráfico

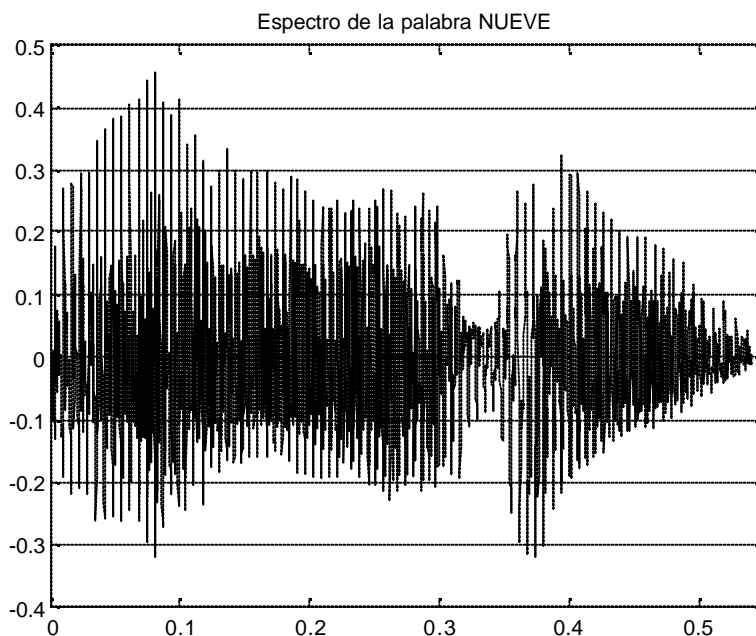


Figura. 1.37. Espectro de la Palabra Nueve

La duración de la señal es de 0,54 segundos. El tiempo de cada muestra es de 0,00009070 segundos. Con estos datos se pueden obtener las ventanas de análisis necesarias para el cálculo de la autocorrelación y los retardos.

Esta señal de voz se encuentra en el archivo **nueve.wav** y el espectro se obtiene de ejecutar el archivo **espectronueve.m**.

Usando MATLAB, para la función de autocorrelación se utiliza el comando **autocorr** que tiene dos parámetros: el vector de entrada para calcular la autocorrelación y el número de retrasos que se desea calcular.

El ejemplo del programa para el análisis de una ventana es el siguiente:

<code>[x,y,z] = wavread('nueve.wav');</code>	para obtener el vector de muestras
<code>w=wavread('nueve.wav','size');</code>	para obtener el numero de muestras
<code>for k=2196:2393</code>	
<code>v1(k-2195)=x(k)</code>	obtención de la ventana desde la muestra
<code>end</code>	2196 hasta la 2393 de tiempo 17,5 ms
<code>m=0;</code>	
<code>for k=1:197</code>	
<code>m=m+v1(k);</code>	
<code>end</code>	
<code>m=m/197;</code>	obtención del valor medio
<code>for k=1:197</code>	
<code>v1(k)=v1(k)-m;</code>	resta del valor medio
<code>end</code>	
<code>v1r=autocorr(v1,12);</code>	función de auto correlación
<code>plot(v1r);</code>	

Para cada ventana se utilizan muestras diferentes entonces este programa se repite para cada fonema y para cada ventana de un fonema pero con diferentes valores en los bucles for que sirven para acceder a las muestras.

### FONEMA N

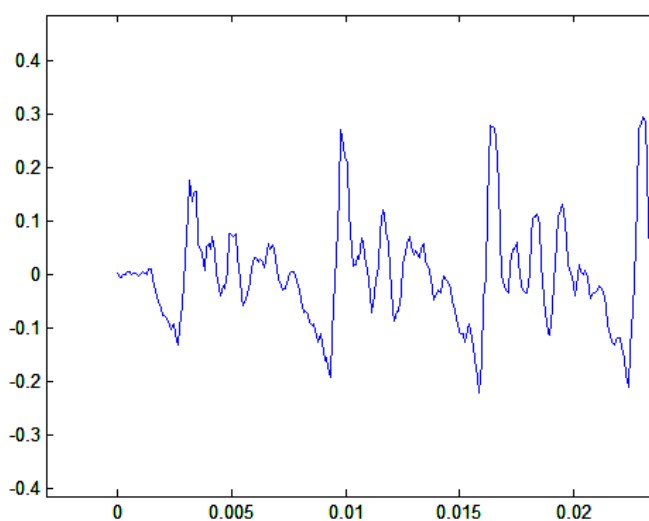
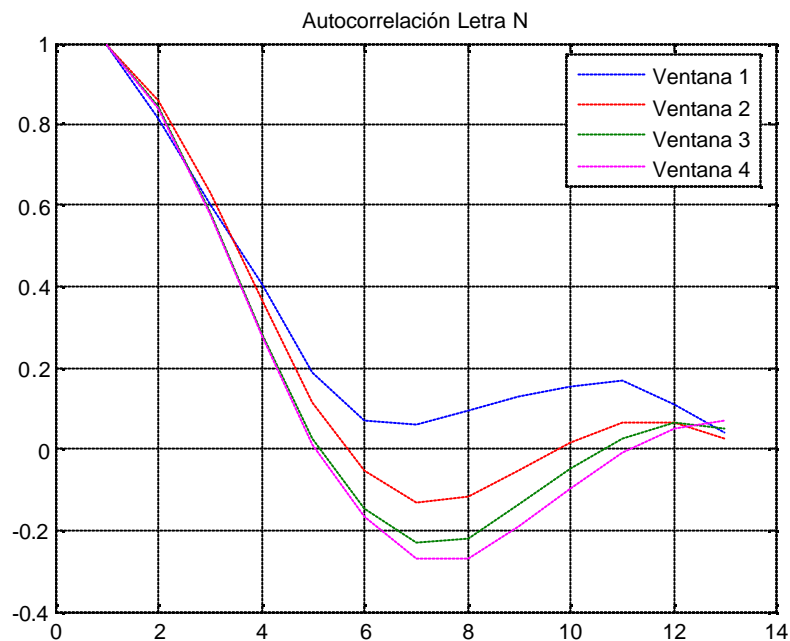


Figura. 1.38. Espectro del Fonema N

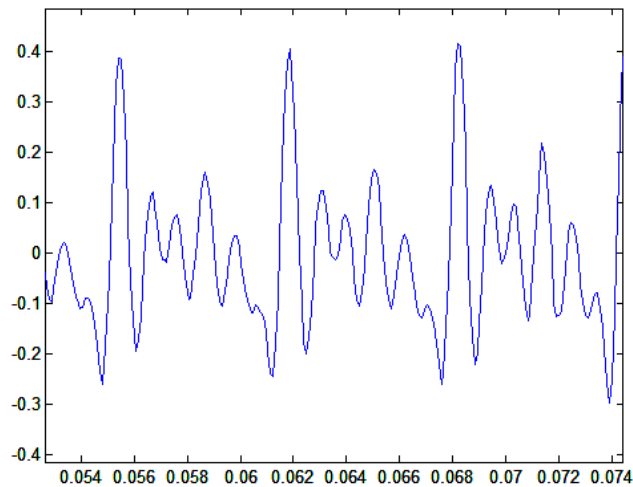
La letra N es una consonante que en esta palabra tiene poca duración. Su forma es muy irregular y se tomaron ventanas de análisis de 6.5 ms, en las cuales hay 72 muestras.



**Figura. 1.39. Autocorrelación Letra N**

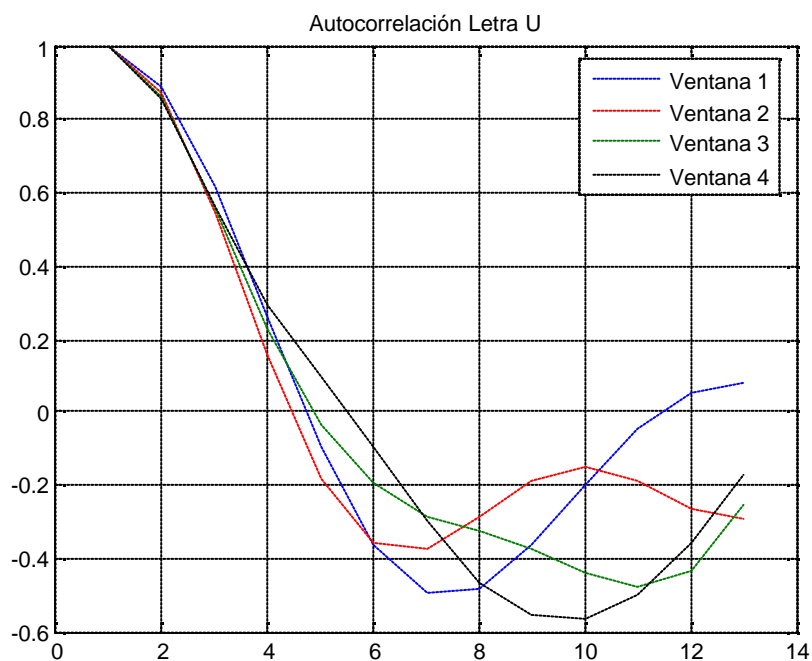
Con este gráfico demostramos la estacionariedad local de las señales de voz y además se comprueba que las auto correlaciones de las ventanas de un mismo fonema son parecidas, guardan características comunes.

En el archivo **autocorr\_n.m** se encuentra el código para obtener las curvas de autocorrelación de este fonema para diferentes ventanas.

**FONEMA U****Figura. 1.40. Espectro del Fonema U**

Para las señales de vocales se toman ventanas de 15 a 20 ms para el análisis de autocorrelación.

En el archivo **autocorr\_u.m** se encuentra el código para obtener las curvas de autocorrelación de este fonema para diferentes ventanas.

**Figura. 1.41. Autocorrelación Letra U**

En el último gráfico se aprecia la auto correlación de las 4 ventanas del fonema u. Los retrasos son parecidos pero se nota algunos diferentes. Esto se debe a que la duración de u es menor a la duración de e, entonces en las muestras no se tiene una u pura.

### FONEMA E

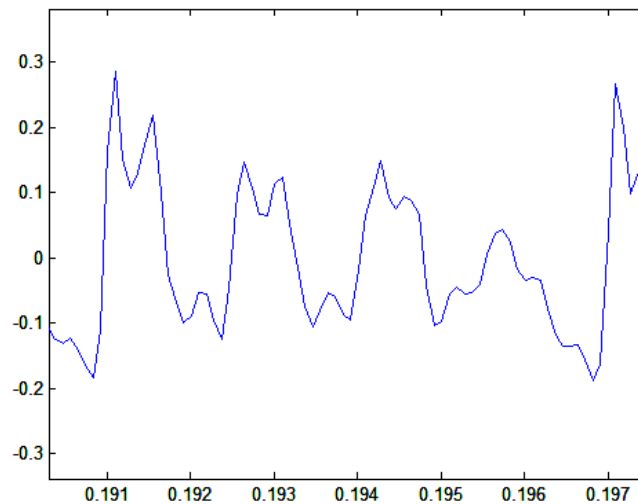


Figura. 1.42. Espectro del Fonema E

Para el análisis de este fonema se utilizaron ventanas de 17,5 ms, en las cuales existen 197 muestras.

Se puede ver en el gráfico que los fonemas vocales son más suavizados que las consonantes.

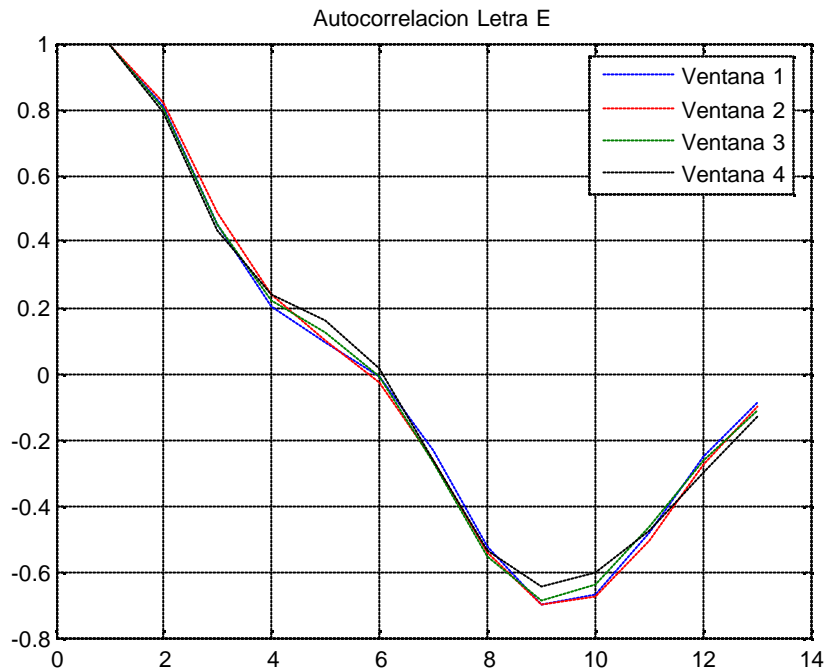


Figura. 1.43. Autocorrelación Letra E

En este último gráfico podemos apreciar la auto correlación de el fonema E. Vemos que los valores de los retrasos en las diferentes ventanas es muy parecido incluso mas que los retrasos vistos en las ventanas del fonema u.

En el archivo **autocorr\_e.m** se encuentra el código para obtener las curvas de autocorrelación de este fonema para diferentes ventanas.

### FONEMA V

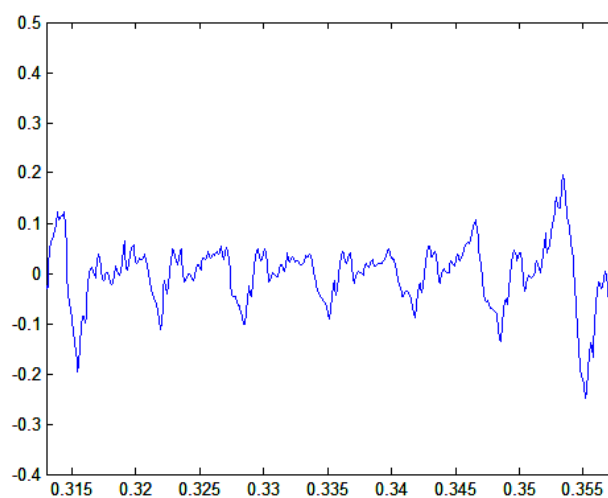


Figura. 1.44. Espectro del Fonema V

Nuevamente se observa que una consonante es mucho mas variante que una vocal entonces para el caso de la letra **v** se tomaron ventanas de 6.5 ms en las cuales existen 72 muestras. A continuación se muestran los gráficos de las funciones de autocorrelación de las ventanas con los retrasos.

En el último grafico que representa los retrasos del fonema v se puede apreciar que los mismos disminuyen casi de una manera lineal a diferencia de los otros fonemas que existían curvas.

En el archivo **autocorr\_v.m** se encuentra el código para obtener las curvas de autocorrelación de este fonema para diferentes ventanas.

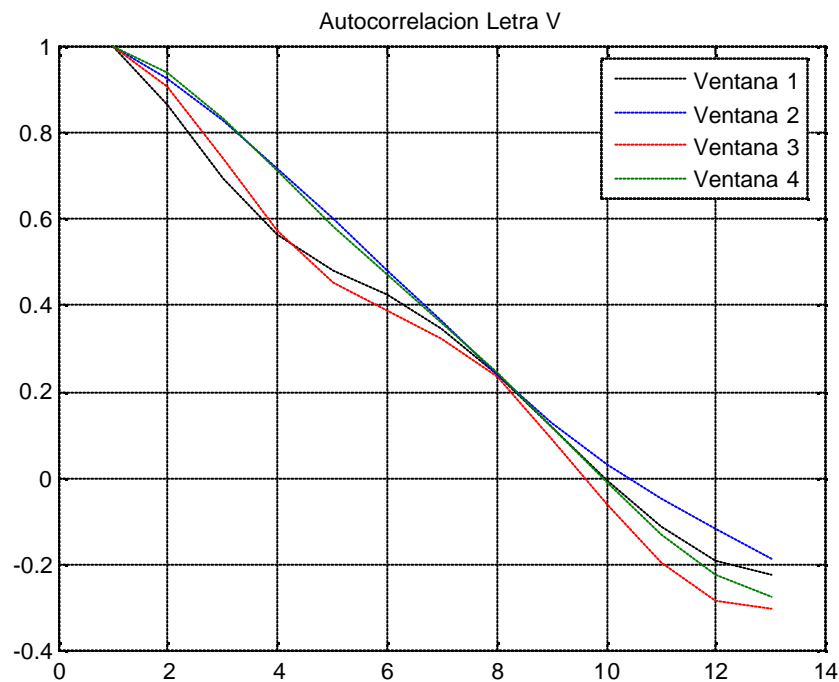


Figura. 1.45. Autocorrelación Letra V

### 1.3.1.3. Conclusiones

- Las señales de voz tienen una estacionariedad local por eso se realiza el análisis de las mismas con ventanas específicas para vocales y consonantes.
- La autocorrelación de una vocal es similar cuando se la calcula en diferentes ventanas de análisis. Así cuando se toma una ventana de alrededor de 20 ms se puede comprobar que los retrasos de la función de autocorrelación son similares y van disminuyendo en amplitud normalizada.
- La autocorrelación de una consonante es similar cuando se la calcula en diferentes ventanas de análisis. Si se toma una ventana de alrededor de 8 ms se puede comprobar que los retrasos de la función de autocorrelación son similares y van disminuyendo en amplitud normalizada.
- En algunos casos los retrasos son bien parecidos en las diferentes ventanas de análisis pero en otros casos estos retrasos difieren más o menos en un máximo de 0,2 amplitud normalizada.
- En los fonemas E y V se encontró una mayor similitud entre los atrasos de la función de autocorrelación teniendo valores y gráficas bien parecidas.
- Además con estos valores se comprueba que las características estadísticas de la señal de voz no varía con el tiempo.



### 1.3.2. ANÁLISIS DE LA SEÑAL DE VOZ POR MODELAMIENTO AUTOREGRESIVO

#### Comando del Estudio: “nueve”

El proceso de análisis de la voz se realiza en el transmisor. El proceso incluye un modelo auto regresivo el cual tiene por objetivo minimizar el error cuadrático medio. Este filtro actúa como un predictor lineal. La forma de este modelo toma las muestras anteriores de la señal para estimar una nueva señal en el instante  $n$  y así calcular el error restando de la señal en el instante  $n$ . El algoritmo es el siguiente:

$$e(n) = x(n) + a_1x(n-1) + \dots + a_px(n-p)$$

Donde  $e(n)$  es el la señal de error,  $p$  es el orden del filtro que es este caso va a ser de 12.

Con la utilización de Matlab se puede obtener el vector de la señal, calcular la estimación de  $x(n)$ , restarla de la señal y obtener el error. Cabe destacar que los valores reales de los coeficientes del filtro son  $a_k = -w_k$

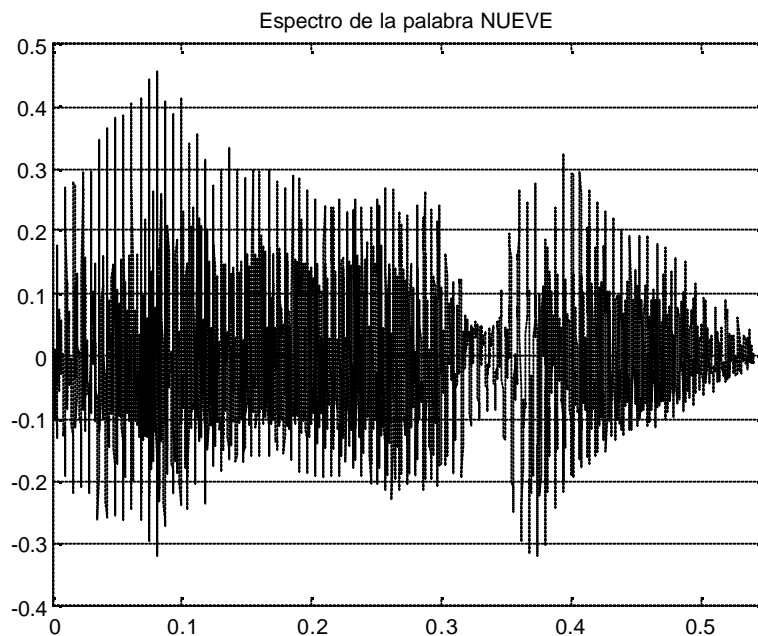
#### 1.3.2.1. Procedimiento

1. Establecer el orden del filtro AR. Se establece  $p = 12$ .
2. Se divide el comando (señal de voz: “nueve”) en bloques de procesamiento de 120 muestras. Con esto se aplica el criterio de estacionariedad.
3. A cada bloque se le aplica una ventana de Hanning. Para suavizar los extremos.
4. Cuando el buffer este lleno (120 muestras), estimar los atrasos  $R_x(0)$ ,  $R_x(1)$ ,...,  $R_x(12)$ .
5. Calcular los coeficientes  $w_k$ :  $w_k = R_x^{-1} \gamma_x$ , donde  $\gamma_x$  es el vector de los retrasos.
6. Procesar el bloque con los  $w$  respectivos.
7. Concatenar todos los bloques de salida para obtener la estimación total.

- Finalmente se resta de la señal original la señal estimada y se obtiene el error para ser transmitido.

Con la utilización del comando `ARYULE` de Matlab se obtienen directamente los coeficientes  $w$  que cambiados de signo proporcionan los coeficientes  $a_k$  del filtro AR.

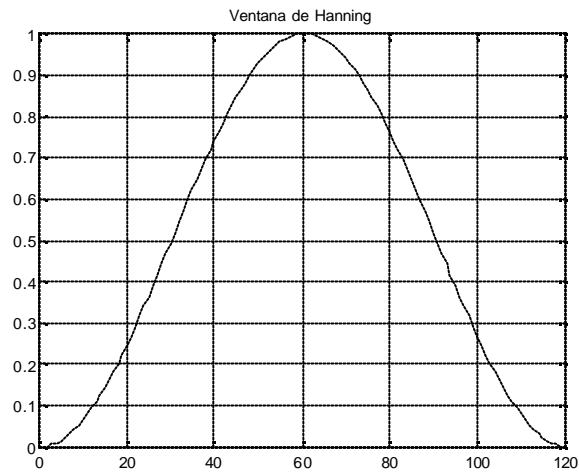
En el algoritmo de la programación de este proceso se toman las 12 primeras muestras de la señal como condiciones iniciales. Así se empieza el bloque desde la muestra 13. A continuación se presentan los gráficos obtenidos para las distintas partes del proceso:



**Figura. 1.46. Señal de Voz, 11025 Hz, 16 bits: Nueve**

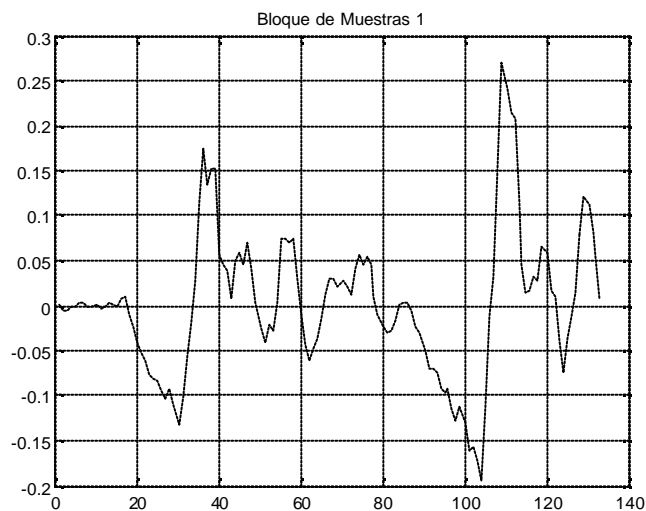
La señal de voz en la cual se representa el comando nueve se muestra en la figura 1.46. Es una señal con una  $f_s$  de 11025 Hz con 16 bits de codificación. La duración del comando es de 0.54 segundos y en total se tienen 5962 muestras.

El comando `Hanning (120)` calcula una ventana de Hanning de 120 muestras para aplicar al bloque o ventana de análisis. La forma de la ventana de Hanning se muestra en la figura 1.47.



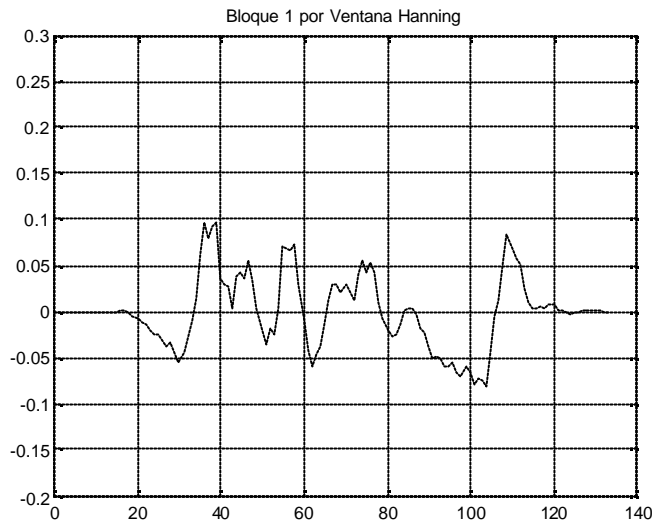
**Figura. 1.47. Ventana de Hanning para 120 Muestras**

En la figura 1.48 se muestra un bloque de 120 muestras de la señal original “nueve”:



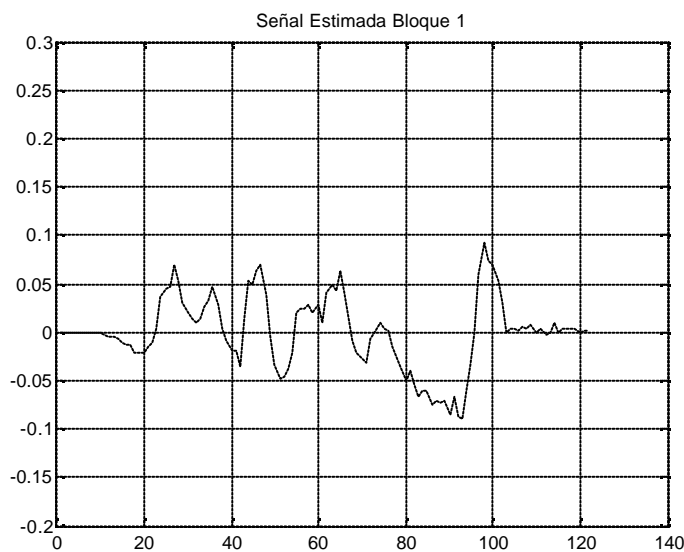
**Figura. 1.48. Bloque de 120 Muestras de la Señal de Voz  
Para el Primer Bloque de Proceso**

Una vez multiplicada la señal de la figura 1.47 con la ventana de Hanning se obtiene la señal de la figura 1.49.



**Figura. 1.49. Señal de Voz (Primer Bloque) Aplicada una Ventana de Hanning**

La figura 1.49 muestra que la señal es menor en amplitud y suavizada en los extremos. Esta señal está lista para ser procesada. Una vez encontrados los coeficientes del filtro se procede a calcular el estimado de la señal y se obtiene la señal de la figura 1.50.



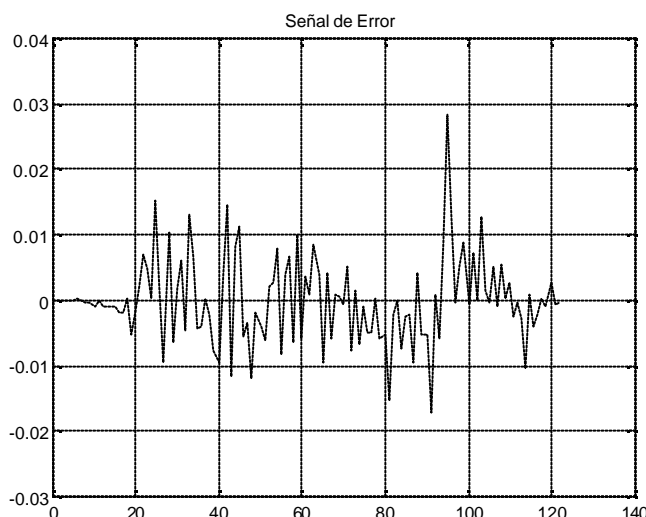
**Figura. 1.50. Señal Estimada del Primer Bloque de Proceso**

Comparando la figura 1.49 en la cual está la señal  $x(n)$  a ser procesada con la señal de la figura 1.50 que es la estimación, se observa una similitud en amplitudes y forma lo que indica que la estimación está bien realizada y se aproxima mucho a la señal original.

El siguiente paso es restar la señal estimada de la señal original para calcular el error:

$$e(n) = x(n) - \hat{x}(n)$$

En la figura 1.51 esta representado  $e(n)$ , en donde se puede apreciar la disminución de la amplitud de la señal conservando la información de la señal original:

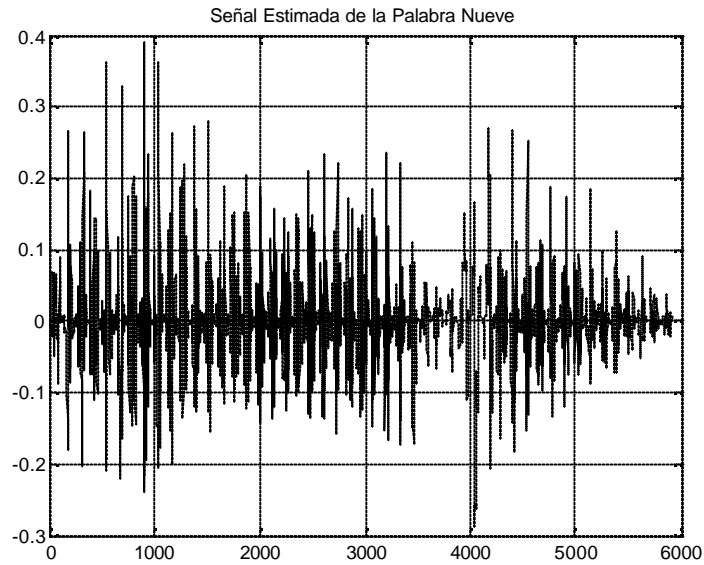


**Figura. 1.51. Señal de Error  $e(n)$  para el Primer Bloque de Proceso.**

Se puede apreciar que la potencia se ha disminuido entonces el valor del error cuadrático medio se ha disminuido en cerca de diez veces. Antes los picos de la señal de voz eran alrededor de 0,3 y -0,2, en el error son de 0,03 y -0,02, para este primer bloque de análisis.

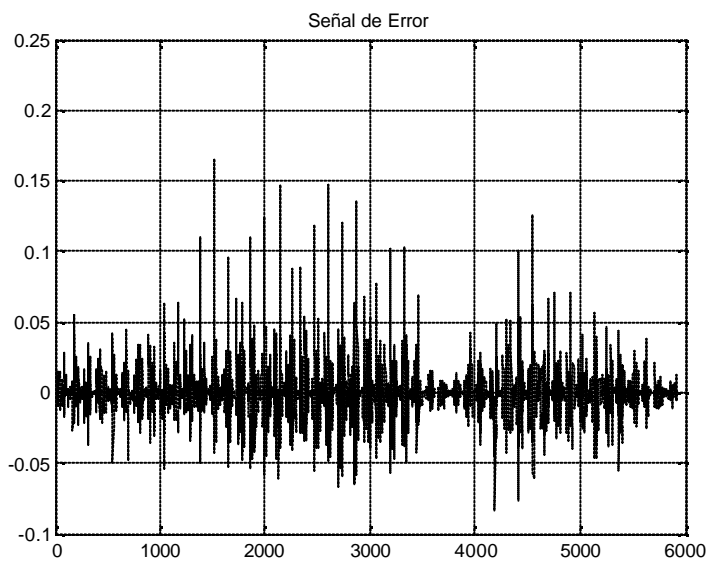
Estas gráficas se obtienen de la ejecución del archivo **autorg\_1.m**, usando el archivo **nueve.wav** como señal de entrada.

En la figura 1.52 se observa el resultado de la señal de estimación de todo el comando. Esta señal es la que se restará de la original de comando para obtener la señal  $e(n)$  completa. Las semejanzas entre las formas de onda de la señal original y la señal estimada son bastante notorias, lo cual es muy deseable.



**Figura. 1.52. Señal de Estimación de la Señal Original**

Finalmente se restan estas dos señales obteniendo la señal de error que podrá ser transmitida. En la figura 1.53 se observa la señal final  $e(n)$ .



**Figura. 1.53. Señal  $e(n)$  a Ser Transmitida.**

La señal  $e(n)$  completa tiene menor potencia que la señal original, pero contiene la misma información de la señal original, prueba de ello es el archivo **numero.wav**, que al reproducirlo es un nueve robotizado.

Estas gráficas se obtienen de la ejecución del archivo **autoregr.m** cuyo código fuente:

```
clc;
clear all;

[x,fs,bit] = wavread('nueve.wav');

z=0;
zz=0;

for v=0:119:5760

    for k=1+v:133+v
        b1(k-v)=x(k);
        b1h(k-v)=0;
    end

    h=hanning(121);

    for j=13:133
        b1h(j)=h(j-12)*b1(j);
        b2h(j-12)=h(j-12)*b1(j);
    end

    w=aryule(b2h,12);
    a=w*(-1);

    for n=13:133
        e(n-12)=b1h(n)+a(1)*b1h(n-1)+a(2)*b1h(n-2)+a(3)*b1h(n-3)+a(4)*b1h(n-4)+a(5)*b1h(n-5)+a(6)*b1h(n-6)+a(7)*b1h(n-7)+a(8)*b1h(n-8)+a(9)*b1h(n-9)+a(10)*b1h(n-10)+a(11)*b1h(n-11)+a(12)*b1h(n-12);
    end

    for g=13:133
        tx(g-12)=b1h(g)-e(g-12);
    end

    estimado=[zz,e];
    etotal=[z,tx];
    z=etotal;
    zz=estimado;
end

figure(1)
plot(x);
grid on;
figure(2)
plot(estimado);
grid on;
figure(3)
plot(etotal);
grid on;

wavwrite(etotal,fs,bit,'numero.wav');
```

### 1.3.3. TASA DE CRUCE POR CERO

Para determinar la tasa de cruce por cero se toman bloques de 120 muestras, se calcula el valor medio de todos los valores y luego se cuenta el número de veces que la señal cruza por cero. Este valor nos ayuda a diferenciar las vocales de las consonantes.

#### 1.3.3.1. Procedimiento

En la figura 1.54 se observa el espectro de la señal correspondiente a la palabra **nueve**, que se encuentra grabado en el archivo **nueve.wav**. La señal fue muestreada a 11025 KHz. y codificada con 16 bits PCM con el grabador de sonidos de Windows® y editado con la ayuda de GoldWave 4.0, aunque se puede utilizar cualquier editor de audio disponible. Con este último lo que se realizó fue la supresión del silencio inicial y final de la grabación inicial.

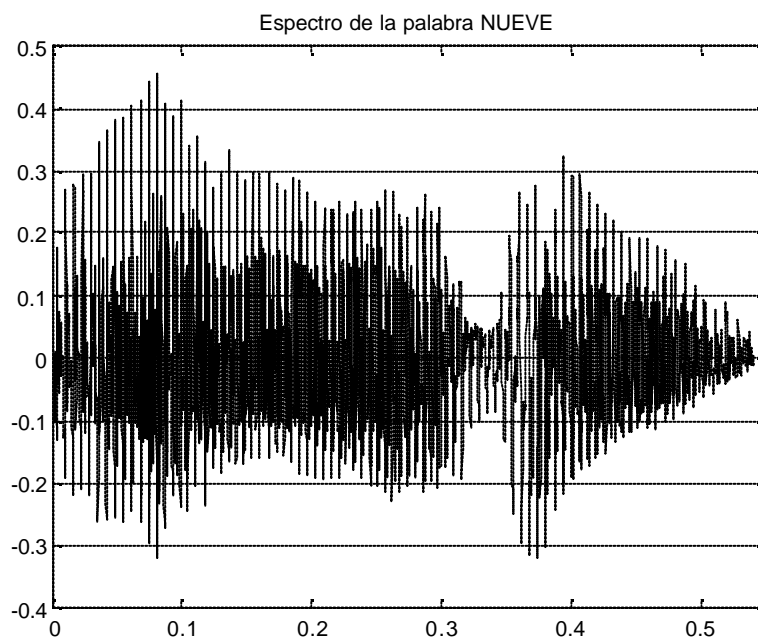
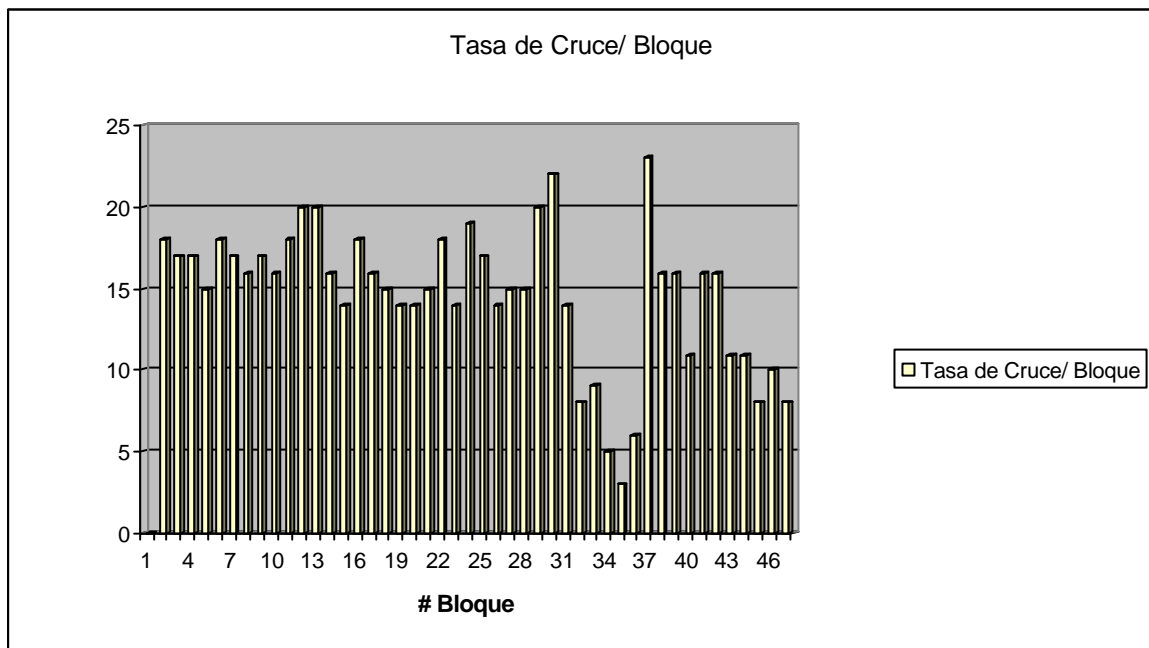


Figura. 1.54. Señal de Voz, 11025 Hz, 16 bits: Nueve



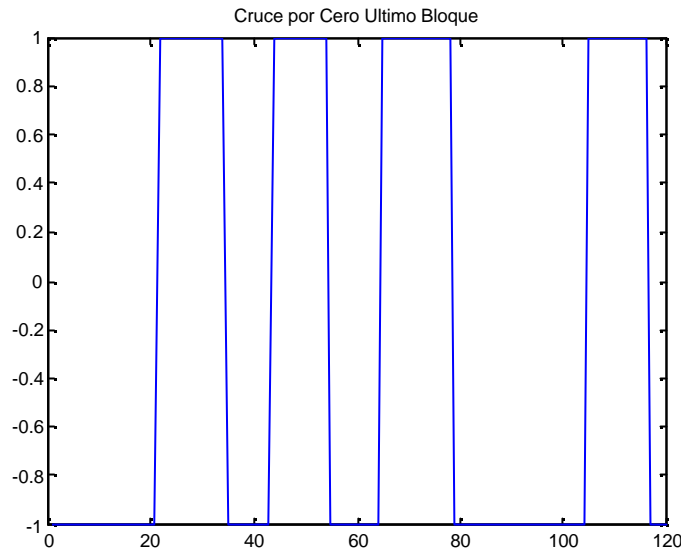
Se debe contar cuantas veces la señal cruza por cero para obtener la tasa de cruce por cero de cada bloque. Para calcular la tasa de toda la señal se utiliza un algoritmo que cuente las veces de cruce por cero para todos los bloques de procesamiento que en total son 47. Al final se encuentra el programa para el cálculo de la tasa de cruce por cero, que consta en el archivo **cruce\_zero.m**.



**Figura. 1.55. Variación de la Tasa de Cruce por Cero por Bloques**

En la figura 1.55 se observa la variación de la tasa de cruce por cero a lo largo de cada uno de los bloques de 120 muestras cada uno, con ello se puede apreciar donde existe mayor concentración de cruces por cero y así se podrá determinar las posiciones de vocales y consonantes respectivamente entre mayor sea esta concentración.

Por ejemplo se puede determinar la posición de la letra V de la palabra nueve por el cambio brusco que se produce al pronunciarla, comparadas con las vocales U y E que son adyacentes a ella.



**Figura. 1.56. Esquema de Cruce por Cero de la Señal de Voz**

En la figura 1.56 se puede apreciar a través de una función **signo(x)** los cambios de polaridad de la señal, es decir, los cruces por cero para el último bloque procesado por el algoritmo realizado.

En la siguiente tabla se muestra de forma aproximada la tasa de cruce por cero para cada letra. Se nota que las vocales tienen menor tasa de cruce por cero que las consonantes. Los valores completos de cruce por cero en cada bloque se encuentran en el archivo **CruceZero.xls**.

Letra	Tasa de cruce por cero
n	18
u	15
e	18
v	19
e	12

**Tabla. 1.2. Tabla Resumen de la Tasa de Cruce por Cero para Cada Fonema**

La figura 1.57 muestra la tasa de cruce por cero para los diferentes bloques procesados de todo el comando de voz utilizado.

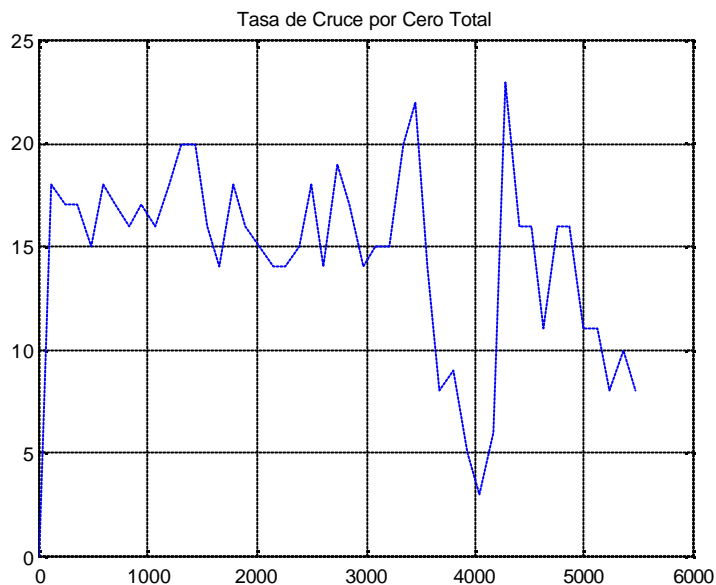


Figura. 1.57. Tasa de Cruce por Cero de la Señal de Voz

### 1.3.3.2. Conclusiones.

- Con el cálculo de la tasa de cruce por cero se puede diferenciar las vocales de las consonantes debido a que las vocales tienen un valor de tasa menor mientras que las consonantes como la N o la V tienen un valor de tasa de cruce por cero alto.
- Se puede apreciar la diferenciación de los fonemas en la gráfica de la señal y la tasa de cruce por cero. Como se puede apreciar en la figura 5, la tasa de cruce es mayor en las posiciones aproximadas de las consonantes, con lo que se ratifica lo considerado para la figura 3 y la tabla 1.
- Se debe utilizar una palabra bien pronunciada para poder realizar el cálculo de la tasa de cruce por cero y diferenciar las vocales de las consonantes.
- Las consonantes tienen mayor frecuencia por eso es que tienen mayor tasa de cruce por cero.

El código fuente de esta práctica es el siguiente y se encuentra en el archivo **cruce\_zero.m**.

```
clc;
clear all;
[x,fs,bit] = wavread('nueve.wav');

y=0;
```

```
tcz=0;
zz=0;
z=0;

for k=0:119:119*45

    for h=1+k:120+k
        x1(h-k)=x(h);
    end

    z=mean(x1);

    xx=x1-z;
    tcz=0;
    xx=sign(xx);

    for j=1:119
        if (xx(j)*xx(j+1))>0
            tcz=tcz;
        else
            tcz=tcz+1;
        end
    end

    etotal=[zz,tcz];
    zz=etotal;

end

d=etotal';
plot(x)
grid on;

figure(2)
plot(xx)

figure(3)
gg=1:119:119*47;
plot(gg,etotal)
grid on;
```

## 1.4. REFERENCIAS

### 1.4.1. LIBROS

- SKLAR, Bernard, **Digital Communications**, Segunda Edición, Prentice Hall, Estados Unidos 2001, páginas 62-100.
- STREMLER, Ferrel, **Sistemas de Comunicación**, Segunda Edición, Editorial AlfaOmega, México 1989, páginas 124-132.
- TOMASI, Wayne, **Sistemas de Comunicaciones Electrónicas**, Segunda Edición, Prentice Hall, México 1996, páginas 608-638.
- COUCH, Leon, **Sistemas de Comunicación Digitales y Analógicos**, Quinta Edición, Prentice Hall, México 1998, páginas 136-141, 185-194.
- FREEMAN, Roger, **Ingeniería de Sistemas de Telecomunicaciones**, Primera Edición, Editorial Limusa, México 1991, páginas 444-461.
- ESPE, **Codificación Digital de Señales**, apuntes de la asignatura, semestre septiembre 2004 – marzo 2005.

### 1.4.2. INTERNET

- <http://www.eveliux.com/fundatel/analogdigital.html>, Conversión Analógica – Digital.
- <http://prof.usb.ve/tperez/docencia/2422/contenido/muestreo/muestreo.htm>, Muestreo.
- [http://www.portalgsm.com/documentacion\\_extendida/100\\_0\\_17\\_0\\_C24/](http://www.portalgsm.com/documentacion_extendida/100_0_17_0_C24/), Vocoders.
- [http://www.asifunciona.com/electronica/af\\_conv\\_ad/conv\\_ad\\_5.htm](http://www.asifunciona.com/electronica/af_conv_ad/conv_ad_5.htm), Conversión Análoga Digital.
- <http://ceres.ugr.es/~alumnos/luis/vocoders.htm>, Vocoders.

## CAPÍTULO II

### TRANSMISIÓN BANDA BASE

#### 2.1. INTRODUCCIÓN.

Una vez que se ha revisado como convertir una señal analógica de una fuente de información en una secuencia de dígitos binarios, y las ventajas, como los problemas relacionados con la conversión análoga-digital, se continúa en este capítulo con el estudio de la transmisión de esa información en banda base.

La banda base es aquella que incluye generalmente la frecuencia cero, es decir desde *DC* hasta un valor finito de unos pocos megahertz; ejemplos de canales de este tipo son los cableados como el de coaxial y par trenzado telefónico. En estos canales la información puede transmitirse directamente mediante técnicas de modulación de pulsos PCM, que es el nombre dado a las señales banda base obtenidas de las señales PAM (Pulse Amplitud Modulation) por codificación de cada muestra en una palabra digital. Las señales PAM son las obtenidas del proceso de muestreo natural.

Es decir, que las señales en banda base no necesitan una portadora sinusoidal para efectuar traslación en frecuencia del espectro de la señal transmitida.

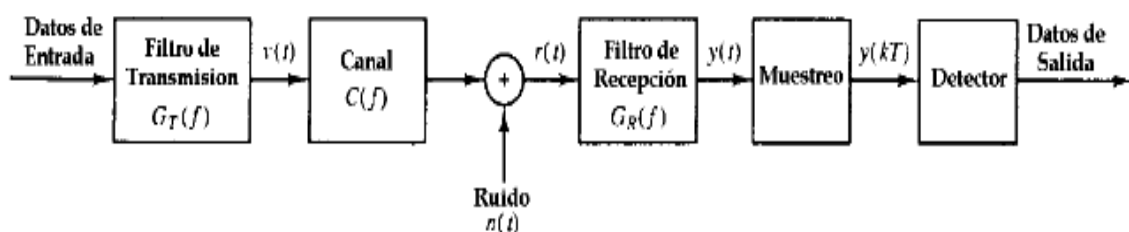


Figura 2.1. Diagrama de Bloques de una Señal PAM Digital

Pero existen también varias fuentes de corrupción que afectan la recuperación de las señales, entre ellas los efectos producidos por el muestreo y cuantización, así como los producidos por el canal de comunicaciones empleado.

Entre los primeros se encuentran el ruido de cuantización (revisado en el capítulo I), saturación del cuantizador, y el **timing Jitter**. La saturación se produce cuando se ingresa una señal que supera el rango de operación para el cual ha sido diseñado el cuantizador. Por otro lado, el Jitter es un fenómeno asociado a la temporización y además es un proceso aleatorio que afecta las posiciones de la señal, volviéndolas desconocidas y este desplazamiento es similar a la modulación en frecuencia o **FM**.

Los efectos del canal están relacionados con el ruido introducido por el canal mismo como tal y la interferencia intersimbólica (ISI). El ruido es causado por ruido térmico o por la interferencia de otras señales, lo cual se puede corregir aumentando la potencia de la señal transmitida. La ISI se produce cuando el ancho de banda del canal es menor al ancho de banda de la señal, traslapándose los pulsos (PCM) y produciendo altas tasas de error en la transmisión.

Específicamente en este capítulo se revisa varios tipos de formas de onda modulante o códigos de línea como retorno a cero, no retorno a cero, fase codificada, entre otros. Las causas y efectos de la Interferencia Intersimbólica, así como algunas de las técnicas usadas para su tratamiento y se finaliza con el estudio del receptor óptimo, para lo cual se analiza el comportamiento y la probabilidad de error producida por el uso de una determinada forma modulante o código de línea.

Al final se incluyen las prácticas de laboratorio relativas a los temas estudiados con sus respectivos desarrollo y análisis de resultados.

## 2.2. MARCO TEÓRICO

### 2.2.1. CÓDIGOS DE LÍNEA

La terminología **Código de Línea** se origina en la telefonía con la necesidad de transmitir información digital a través del par de cobre, sin embargo rápidamente se aplicó a cualquier canal o línea de transmisión. En un sistema de comunicación digital existe un conjunto de símbolos conocidos a ser transmitidos que pueden ser designados como  $\{m_i\}$ ,  $i = 1, 2, \dots, N$ , con una probabilidad de ocurrencia  $\{p_i\}$ ,  $i = 1, 2, \dots, N$ , donde los símbolos transmitidos en secuencia son estadísticamente independientes. La codificación de estos símbolos abstractos en formas de onda reales temporales a ser transmitidas en banda base es la codificación de línea. Y ya que el tipo más común de codificación es para datos binarios, tal forma de onda se puede llamar un formato directo para bits seriales.

Las diferentes características del canal, así como las diferentes aplicaciones y requerimientos de desempeño de los sistemas de comunicación han impulsado el desarrollo y estudio de varios tipos de códigos de línea. Además de ello, la complejidad de un codificador y el costo de su decodificador pueden determinar el código de línea a escogerse, una vez que cada uno de ellos tiene distintas propiedades.

Las consideraciones más importantes y usuales que se tienen para la selección de un código de línea son:

**1) Sincronización:** la forma de onda producida debe contener suficiente información de sincronización de forma que el receptor se sincronice con el transmisor y se decodifique la señal apropiadamente. Ese contenido debe ser relativamente independiente de la estadística de la fuente, es decir una serie de **1s** o **0s** no debe verse como jitter en el receptor.

**2) Contenido de DC:** debido a que los repetidores usados en telefonía son acoplados con AC, es deseable tener cero DC en la forma de onda a usar. Si se usa una señal conteniendo DC significativamente en un acople AC, producirá **DC wander** en la onda recibida. Es decir, la línea base de la señal recibida variará en el tiempo.



**3) Espectro de Potencia:** el espectro de potencia y ancho de banda de la señal transmitida debe enmarcarse a la respuesta en frecuencia del canal para evitar distorsiones significativas. Además el espectro de potencia debe ser tal q la mayoría de la energía este contenida en el menor ancho de banda como sea posible. Entre menor sea el ancho de banda, mayor es la eficiencia de la transmisión.

**4) Monitoreo de Desempeño:** es muy deseable detectar los errores causados por un canal de transmisión ruidoso. La capacidad de detección de errores debe permitir evaluar el desempeño cuando el canal esta en uso.

**5) Probabilidad de Error:** el promedio de la probabilidad de error debe ser tan pequeña como sea posible para una potencia de transmisión dada. Esta refleja la confiabilidad de un código de línea.

**6) Transparencia:** el código de línea debe permitir el procesamiento de todos los posibles patrones de **1s** y **0s**.

### 2.2.1.1. Códigos de Línea Comunes

Un formato de código de línea consiste en una definición formal que especifica como una cadena de dígitos binarios es convertido en una forma de onda. Existen dos grandes clases de códigos de línea binarios: **códigos por nivel** y **códigos por transición**

Los primeros llevan la información en su nivel de voltaje, que puede ser alto o bajo para un período de bit completo o parte de él. Son generalmente instantáneos por lo que codifican un digito binario en una forma de onda distinta, independientemente de cualquier dato binario anterior. Sin embargo, algunos de ellos pueden poseer memoria.

Los códigos de transición, por otro lado, llevan la información en el cambio de nivel de la señal. Pueden ser instantáneos, a pesar de que generalmente tienen memoria, es decir, utilizan los datos binarios anteriores para dictar la forma de onda presente.

Existen dos tipos de códigos de nivel: **retorno a cero (RZ)** y **no retorno a cero (NRZ)**.

En la codificación RZ, el nivel del pulso retorna a cero por una porción del intervalo de bit. En la codificación NRZ, el nivel del pulso se mantiene durante todo el intervalo de bit.

Los códigos de línea se clasifican más a fondo de acuerdo a la polaridad de los niveles de voltaje utilizados para representar los datos. Si se usa solo un nivel de polaridad, positivo o negativo (además el nivel cero), entonces se conoce como *señalización unipolar*. Y, si se usan ambas polaridades, con o sin el nivel cero, se conoce como *señalización polar*. Se suele llamar en algunos casos *señalización bipolar* al código de línea que incluye las polaridades positiva, negativa y el nivel cero.

La definición de los cinco códigos de línea comunes se presenta a continuación con su forma de onda representativa, densidad espectral de potencia (PSD), la probabilidad de error y sus ventajas y desventajas.

#### 2.2.1.1.1. NRZ Unipolar (Binary On-Off Keying - OOK)

En este código un **1** binario se representa por un nivel de voltaje no-cero y un **0** binario se representa con un nivel de voltaje cero, como se muestra en la figura 2.2 (a). Este es un código de nivel instantáneo. Su Densidad Espectral de Potencia, en adelante PSD, con igual probabilidad de **1**s y **0**s está dada por:

$$S_1(f) = \frac{V^2 T}{4} \left( \frac{\sin p f T}{p f T} \right) + \frac{V^2}{4} d(f)$$

Donde  $V$  es el nivel de voltaje del **1** binario,  $T = 1/R$  es la duración de bit, y  $R$  es la tasa de bits en bits por segundo.

El espectro de NRZ Unipolar se muestra en la figura 6.3, donde se muestra solo la mitad del mismo ya que esta PSD es un espectro par de dos lados. Si la probabilidad de un **1** es  $p$ , y la de un **0** es  $(1 - p)$ , entonces la PSD de este código, en el caso más general es:  $4p(1-p) S_I(f)$ . Considerando nula la frecuencia del primer espectro como del ancho de banda de la forma de onda, el ancho de banda de este código es  $R$  (Hz). La tasa de error (probabilidad

de error de bit), para una misma probabilidad de datos, con ruido blanco aditivo Gaussiano (AWGN) y un receptor óptimo (filtro casado), en la detección está dada por:

$$P_e = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{Eb}{2No}} \right)$$

Donde  $Eb/No$  es una medida de la relación señal a ruido (SNR) de la señal recibida. En general,  $Eb$  es la energía de bit o por bit, y  $No/2$  es la PSD de dos lados del AWGN. Específicamente, para este caso,  $Eb$  es la energía de un **1**, la cual es  $V^2T$ . El desempeño del código unipolar NRZ se muestra en la figura 2.5.

Las principales ventajas del unipolar NRZ son: la facilidad de generación, ya que requiere solo una fuente de alimentación, y un relativamente bajo ancho de banda de R (Hz).

Entre sus desventajas están: la pérdida de sincronización y jitter resultantes de largas secuencias de **1s** o **0s** debido a que no existe transición de pulsos. El código no tiene capacidad de detección de error y, por lo tanto, no se puede monitorear su desempeño. Existe una componente significativa de DC y su tasa de error no es tan buena como la de un código polar.

#### 2.2.1.1.2. Unipolar RZ

En este código un **1** binario se representa con un nivel de voltaje no-cero durante una porción de la duración de bit, usualmente la mitad de ese período, y un nivel de voltaje cero el resto del período de bit. Un **0** es representado por un nivel de voltaje cero durante todo el periodo de bit. Así, este es un código de nivel instantáneo, y su representación gráfica se muestra en la figura 2.2 (b).

La PSD de este código de línea, con igual probabilidad de dígitos binarios, está dada por:

$$S_2(f) = \frac{V^2 T}{16} \left( \frac{\sin p f T / 2}{p f T / 2} \right)^2 + \frac{V^2}{4 p^2} \left[ \frac{p^2}{4} d(f) + \sum_{n=-\infty}^{\infty} d(f - (2n+1)R) \right]$$

Donde  $V$  es el nivel de voltaje de un **1** binario, y  $T = 1/R$  es el período de bit.

Su espectro se muestra en la figura 2.3. en el caso más general, cuando la probabilidad de un **1** es  $p$ , la parte continua de la PSD en la ecuación anterior es escalada por el factor  $4p(1-p)$  y la parte discreta por el factor  $4p^2$ . El ancho de banda del unipolar RZ es  $2R$  (Hz). La probabilidad de error de bit para este código es el mismo que para el unipolar NRZ, con tal que se aumente el nivel de voltaje de este código de forma que la energía de bit  $E_b$  para **1** binario, es la misma para ambos códigos y se muestra en la gráfica 2.5. Si el nivel de voltaje y el período de bit son los mismos para un unipolar NRZ y uno RZ, entonces la energía en un **1** binario para unipolar RZ será  $V^2 T / 2$  y la probabilidad de error de es peor por 3 dB.

Las principales ventajas del unipolar RZ son: la facilidad de generación, requiere solo una fuente de alimentación y la presencia de una componente espectral discreta a la tasa de símbolo, lo que permite recuperación de sincronización simple.

Las desventajas son: la existencia de una componente DC, que produce DC wander. Una secuencia larga de **0**s carecerá de transiciones de pulso y puede generar pérdida de la sincronización. No existe detección de error y por tanto no se puede monitorear el desempeño. El requerimiento de ancho de banda es mayor que el de NRZ, el doble. Y la probabilidad de error de bit también es peor que la de los códigos polares.

Tanto el unipolar NRZ como el unipolar RZ son ejemplos de señalización de tipo pulso / no pulso, en la cual, el pulso para un **0** binario  $g_2(t)$ , es cero y para **1** binario generalmente es  $g_1(t) = g(t)$ . Usando  $G(f)$  como la transformada de Fourier de  $g(t)$ , la PSD de esta señalización está dada por:

$$S_{PNP} = p(1-p)R |G(f)|^2 + p^2 R^2 \sum_{n=-\infty}^{\infty} |G(nR)|^2 d(f - nR)$$

Donde  $p$  es la probabilidad de un **1**, y  $R$  la tasa de bit.

### 2.2.1.1.3. Polar NRZ

En este código, el **1** binario se representa con un voltaje positivo  $+V$  y un **0** binario se representa con un voltaje negativo  $-V$  sobre el periodo de bit completo. Se lo conoce como NRZ (L), porque cada bit se representa manteniendo el nivel; su forma de onda se muestra en la figura 2.2 (c), y también es un código de nivel instantáneo. Alternativamente, un **1** puede ser representado por un nivel de voltaje  $-V$  y un **0** con un nivel de voltaje  $+V$ , sin que cambien sus características espectrales y desempeño. La PSD de este código de línea con igual probabilidad de dígitos binarios está dada por:

$$S_3(f) = V^2 T \left( \frac{\sin p f T}{p f T} \right)^2$$

La gráfica de la PSD se muestra en la figura 2.4. Cuando la probabilidad de un **1** es  $p$ , y  $p$  no es 0.5, existe una componente de DC y la PSD se convierte en

$$S_3(f) = 4V^2 T p(1-p) \left( \frac{\sin p f T}{p f T} \right)^2 + V^2 (1-2p)^2 \mathbf{d}(f)$$

El ancho de banda para este código es  $R$  (Hz), independiente de  $p$ . La probabilidad de error cuando  $p = 0.5$  está dada por

$$P_e = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{Eb}{No}} \right)$$

El desempeño del NRZ polar se muestra en la figura 6.5 y es mejor que la probabilidad de error mostrada por los códigos unipolares por 3 dB.

Las ventajas del NRZ polar incluyen un bajo requerimiento de ancho de banda, comparado con el NRZ unipolar, muy buena probabilidad de error de bit, y componente de DC reducida notablemente debido a que la forma de onda tiene DC nula cuando  $p = 0.5$ , aunque el contenido de DC nunca es cero.

Tiene pocas desventajas como no tener detección de error, y que largas secuencias de **1s** o **0s** pueden resultar en una pérdida de sincronización, ya que no hay transiciones mientras dura la secuencia. Se requieren dos fuentes de alimentación para generar este código.

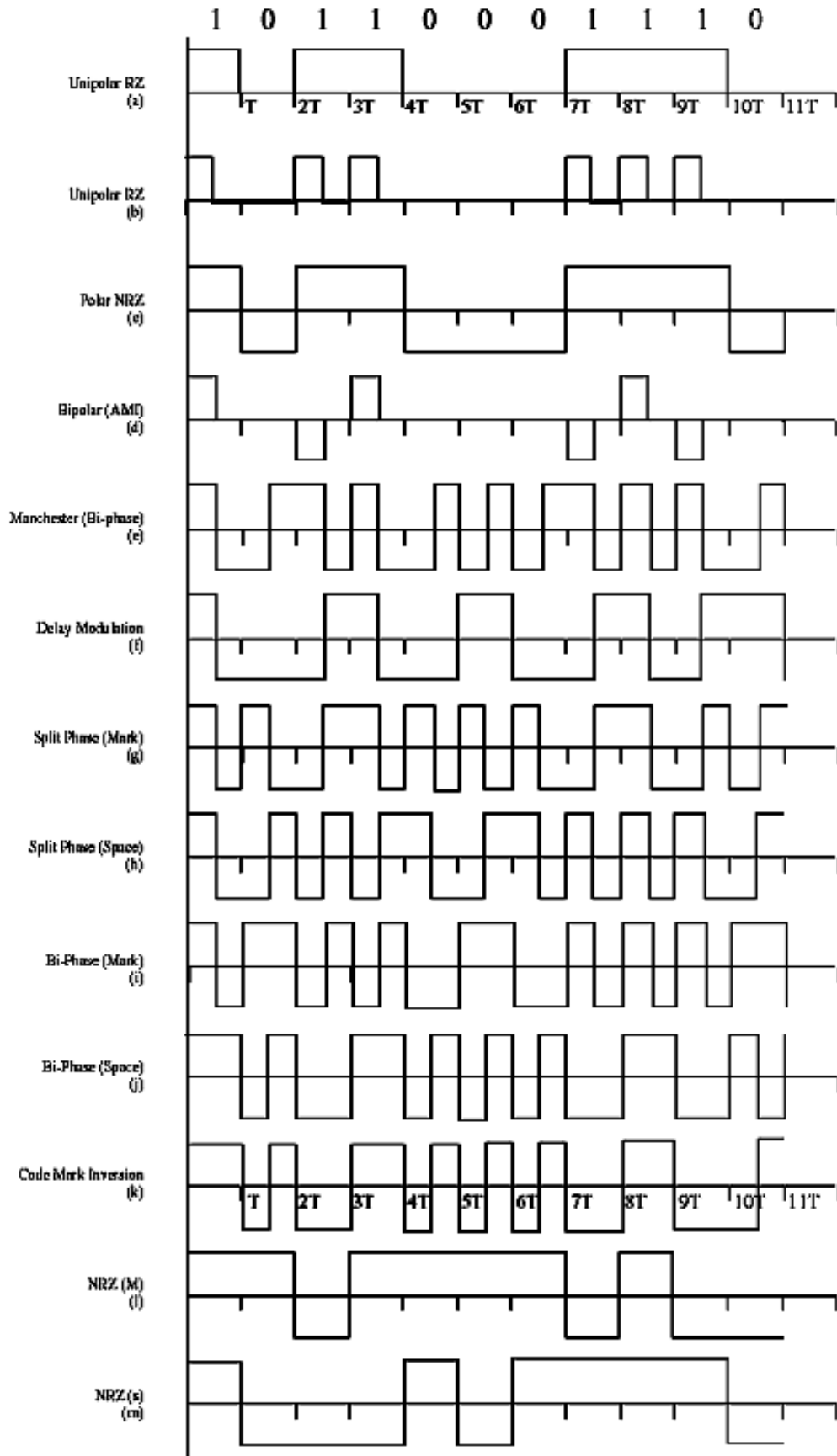


Figura 2.2. Formas de Onda de los Diferentes códigos de Línea

#### 2.2.1.1.4. Polar RZ -Bipolar, Alternate Mark Inversion (AMI), o Pseudoternaria.

Para este esquema, un **1** binario se representa alternando los niveles positivo y negativo de voltaje, los cuales retornan a cero durante una porción de la duración del bit, generalmente la mitad de este. Un **0** binario se representa con un nivel de voltaje cero durante un periodo de bit complete. Se lo llama de *inversión de marca alterna (AMI)* porque los **1s** (marcas) se representan así, alternando pulsos negativos y positivos.

Se llama también **pseudoternario** porque usa tres niveles de voltaje diferente para representar datos binarios. Algunos lo llaman bipolar RZ (BRZ). Su forma de onda se muestra en la figura 2.2 (d).

Este es un código de nivel con memoria y es bastante conocido por su uso en telefonía. Su PSD está dado por:

$$S_4(f) = 2p(1-p)R |G(f)|^2 \left( \frac{1 - \cos 2p f T}{1 + (2p-1)^2 + 2(2p-1)\cos 2p f T} \right)$$

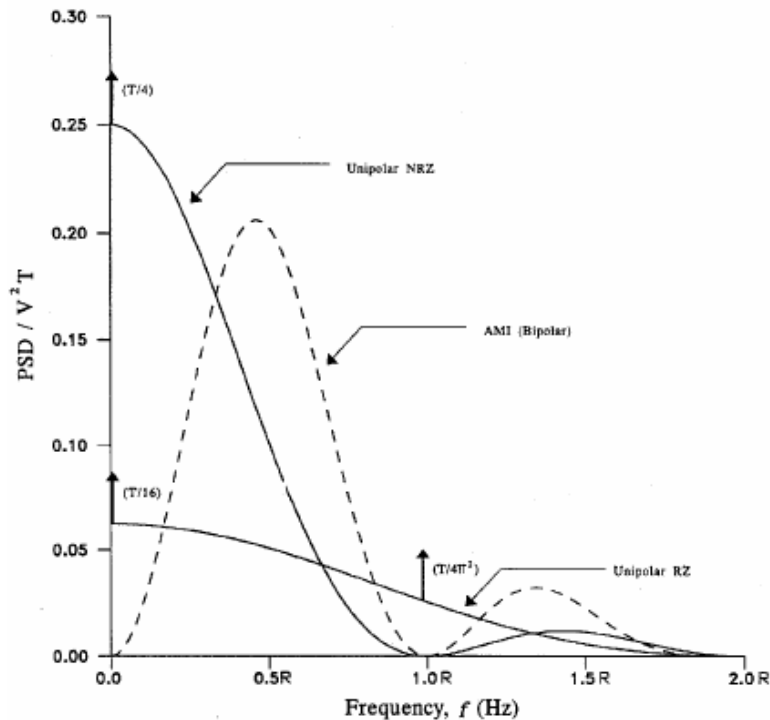
Donde  $G(f)$  es la transformada de Fourier del pulso usado para representar un **1** binario, y  $p$  es la probabilidad de ese **1**. Cuando  $p = 0.5$  y se usan pulsos cuadrados de amplitud  $\pm V$  y duración  $T/2$  para representar **1s**, la PSD se vuelve:

$$S_4(f) = \frac{V^2 T}{4} \left( \frac{\sin p f T / 2}{p f T / 2} \right)^2 \sin^2(p f T)$$

Esta PSD se gráfica en la figura 2.3. Su ancho de banda es  $R$  (Hz); lo cual es cierto para pulsos rectangulares RZ, independientemente del valor de  $p$  en la ecuación anterior. La probabilidad de error de bit cuando datos binarios igualmente probables está dada por:

$$P_e \approx \frac{3}{4} \operatorname{erfc} \left( \sqrt{\frac{E_b}{2N_o}} \right) \quad E_b / N_o > 2$$

Esta curva se muestra en la figura 2.5 y se observa que no es peor que 0.5 dB que para los códigos unipolares.

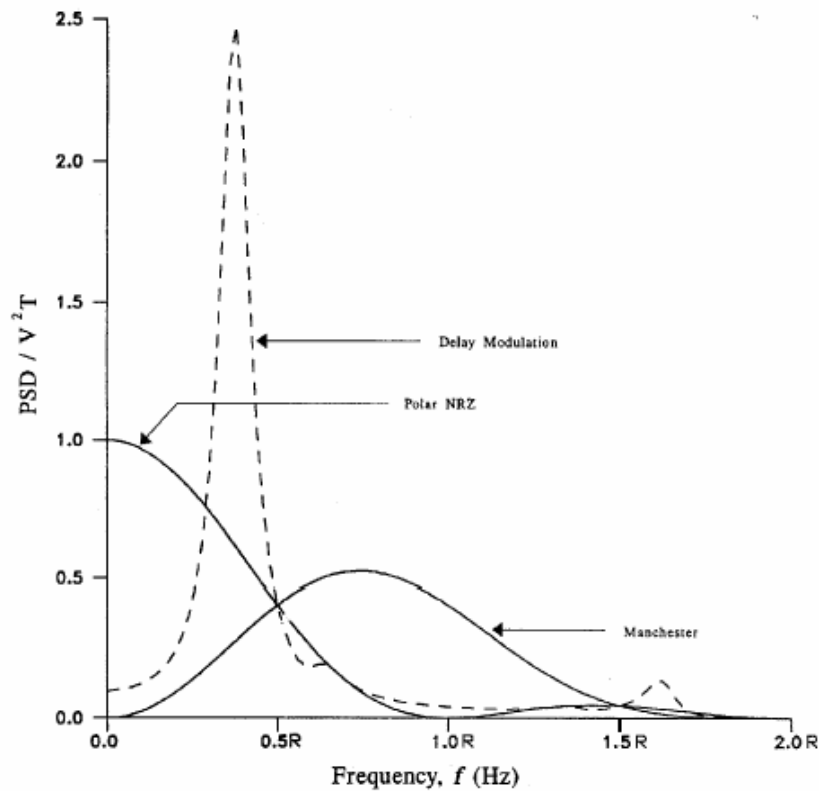


**Figura 2.3. Densidad Espectral de Potencia de los Diferentes códigos de Línea, donde  $R = 1 / T$  es la Tasa de Bit**

Las ventajas del código AMI compensa las desventajas. No tiene componente ni contenido de DC, no hay problemas de wander. La recuperación de sincronización es algo fácil debido a la rectificación de la onda que permite una forma de onda de un RZ unipolar con una componente discreta a la tasa de bit,  $R$ . Debido a la alternabilidad de la polaridad de los pulsos para los 1s, posee detección de error, y por tanto capacidad de monitoreo del desempeño. Requiere un ancho de banda pequeño de  $R$  Hz, a comparación del NRZ unipolar.

Las desventajas son: su peor probabilidad de error en relación con los códigos unipolar y polar. Una secuencia larga de 0s puede derivar una pérdida de sincronización y necesita dos fuentes de alimentación para ser generado.





**Figura. 2.4. Densidad Espectral de Potencia de los Diferentes códigos de Línea, donde  $R = 1 / T$  es la Tasa de Bit**

#### 2.2.1.1.5. Codificación Manchester (Fase Dividida o Bi-fase Digital).

En este modelo, un **1** binario se representa por un pulso que tiene un voltaje positivo la primera mitad de su periodo de bit y voltaje negativo en la segunda mitad de ese periodo. Un **0** binario se representa por un pulso que es negativo la primera mitad de la duración de bit y es positivo durante la segunda mitad. La transición negativa o positiva del medio bit es la que determina un **1** o un **0**, respectivamente. Así, el código Manchester se clasifica como uno de transición instantánea, que no tiene memoria. Se lo llama también bi-fase porque se usa una onda cuadrada de fase  $0^\circ$  para representar un **1** y una onda cuadrada de fase  $180^\circ$  para representar un **0**; o viceversa.

Se usa esta codificación de línea para redes de área local LAN en Ethernet. Su forma de onda se muestra en la figura 2.2 (e). Y su PSD para bits con igual probabilidad de ocurrencia está dada por:

$$S_s(f) = V^2 T \left( \frac{\sin p f T / 2}{p f T / 2} \right)^2 \sin^2(p f T / 2)$$

Donde  $\pm V$  son usados como los niveles de voltaje positivo/negativo para este código.

Su espectro se gráfica en la figura 2.4. Cuando la probabilidad  $p$  de un **1** binario, no es igual a un medio, la parte continua de la PSD se reduce en amplitud y las componentes discretas aparecen en múltiplos enteros de la tasa de bit,  $R = 1/T$ . La PSD resultante es:

$$S_s(f) = V^2 T p(1-p) \left( \frac{\sin p f T / 2}{p f T / 2} \right)^2 \sin^2 \left( \frac{p f T}{2} \right) \\ + V^2 (1-2p)^2 \sum_{n=-\infty, n \neq 0}^{\infty} \left( \frac{2}{n p} \right)^2 \delta(f - nR)$$

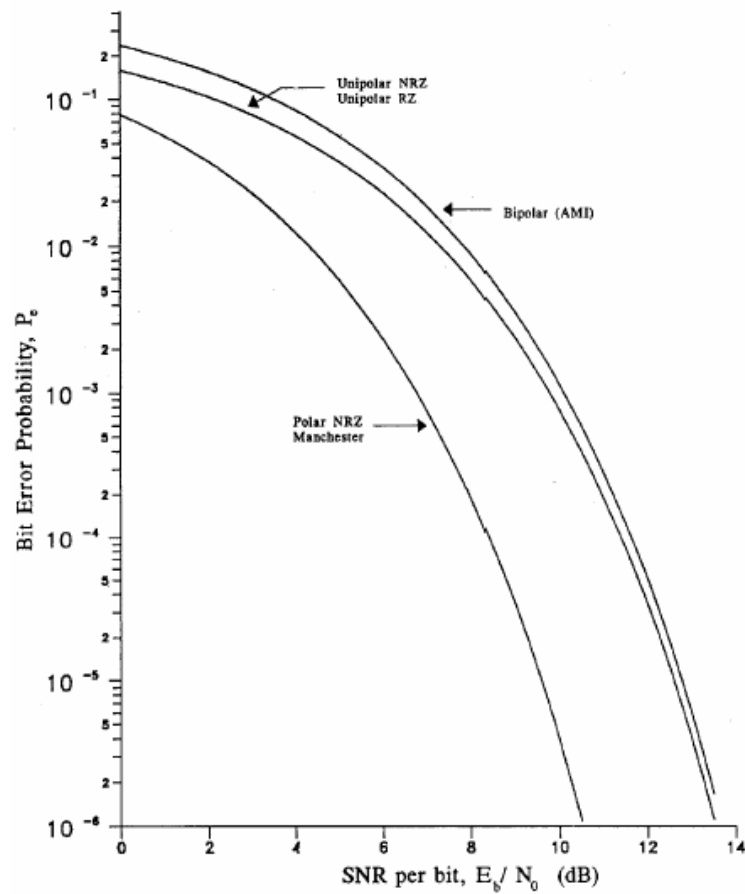


Figura 2.5. Probabilidad de Error de Bit para los Diferentes códigos de Línea

El ancho de banda de para esta forma de onda es  $2R$  (Hz). La probabilidad de error de bit cuando  $p = 0.5$  es la misma que la de NRZ polar y se grafica en la figura 2.5.

Las ventajas de este código incluyen contenido cero DC sobre un pulso base individual, de forma que ningún patrón de bits puede causar acumulación de DC; las transiciones de medio bit siempre están presentes, facilitando la extracción de información de sincronización; y tiene una muy buena probabilidad de error, idéntica al NRZ polar.

La principal desventaja es el requerimiento de ancho de banda mayor a los demás códigos de línea, y tampoco tiene detección de errores.

### 2.2.1.2. Tamaño de la Palabra PCM

Para poder determinar el número de bits o niveles a utilizar por muestra, se debe saber cuanta distorsión se esta dispuesto a tolerar en un sistema con el formato PCM. Es útil manejar esta relación entre la cantidad de distorsión permitida y el tamaño de la palabra PCM, para lo cual, se puede especificar a la magnitud del error de cuantización  $|e|$  como una fracción  $p$  del voltaje análogo pico-pico

$$|e| \leq pV_{pp}$$

y dado que si el paso de cuantización es  $q$ , el error no puede ser mayor que  $q/2$

$$|e|_{\max} = \frac{q}{2} = \frac{V_{pp}}{2(L-1)} \approx \frac{V_{pp}}{2L}$$

$$\frac{V_{pp}}{2L} \leq pV_{pp}$$

$$2^l = L \geq \frac{1}{2p}$$

$$l \geq \log_2 \frac{1}{2p}$$

Siendo finalmente  $p$  el porcentaje de distorsión permitida.

### 2.2.1.3. Señalización Multinivel, de Respuesta Parcial y Codificación Duobinaria

Para incrementar la eficiencia de transmisión, se necesitan más niveles y/o mayor procesamiento de la señal. La señalización multinivel permite una mejora en la eficiencia de transmisión a expensas de un incremento en la tasas de error, se necesita mayor potencia en el transmisor para mantener una probabilidad de error dada.

En la señalización de respuesta parcial, se introduce interferencia intersímbolo deliberadamente, usando pulsos más anchos y que requerirán menor ancho de banda en frecuencia. La cantidad controlada de interferencia de cada pulso puede ser removida en el receptor. La codificación duobinaria, es un caso especial de la respuesta parcial, requiere el mínimo ancho de banda teórico, es decir  $0.5 R$  Hz.

#### 2.2.1.3.1. Señalización Multinivel.

El número de niveles que se pueden usar para un código de línea no se restringe a dos o tres, ya que a mayor número de niveles o símbolos, mayor es la eficiencia de transmisión y por eso se usa esta técnica en aplicaciones con ancho de banda limitado. Si la tasa de símbolo o baudío es  $R_s$  y se usan  $L$  niveles, la tasa de bit  $R_b$  equivalente esta dada por:

$$R_b = R_s \log_2 [L]$$

$$L = 2^k$$

$$R_s = \frac{R_b}{k}$$

*Siendo  $k$  el número de bits empleados para los  $L$  niveles.*

#### 2.2.1.3.2. Señalización de Respuesta Parcial y Codificación Duobinaria.

Introduciendo interferencia correlacionada entre los pulsos, y cambiando el procedimiento de detección, se puede cancelar la ISI en el receptor y lograr una eficiencia de ancho de banda ideal de dos símbolos por Hz.

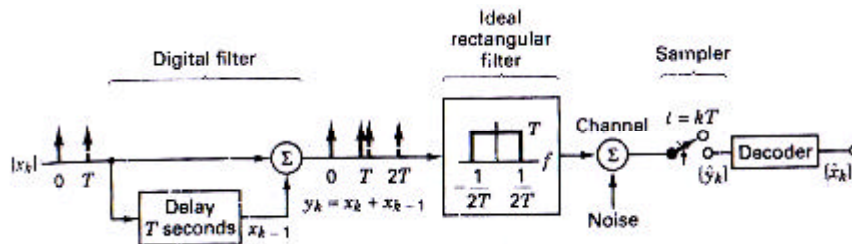


Figura 2.6. Señalización Duobinaria

Para entender su operación, podemos considerar el proceso de la figura 2.6, para el que asumimos que tenemos una secuencia de símbolos binarios  $\{x_k\}$  para ser transmitidos a una tasa  $R$  sobre un sistema de ancho de banda  $W=R/2$ . Como se puede observar, los pulsos pasan por un filtro digital simple antes de ser adecuados por el filtro ideal.

El filtro digital incorpora un retraso de un dígito; para cada pulso, el filtro añade el valor del pulso anterior. De esta forma, cada pulso de la secuencia  $\{y_k\}$  a la salida del filtro digital se puede expresar como

$$y_k = x_k + x_{k-1}$$

Por lo tanto cada dígito  $y_k$  lleva consigo la memoria del dígito anterior. Esta correlación entre amplitudes de pulsos de  $\{y_k\}$  se puede considerar como la ISI controlada introducida por la señalización duobinaria.

Lo que la hace importante también es la facilidad con que puede ser removida en recepción, al igual como fue introducida dicha interferencia. Debido a la existencia de ruido en todos los sistemas prácticos, no se puede obtener  $\{y_k\}$  exactamente a la recepción, sino, su estimación a la que llamaremos  $\{\hat{y}_k\}$ , la cual una vez removida la ISI añadida será la secuencia estimada de  $\{x_k\}$ , llamada  $\{\hat{x}_k\}$ .

Entonces, si el dígito binario  $x_k$  es  $\pm 1$ ,  $y_k$  puede ser  $+2$ ,  $0$ , o  $-2$ , es decir, que para una transmisión M-aria en general, el número de salidas de la señalización de respuesta parcial es igual a  $2M-1$  niveles.

Para la decodificación es al contrario de la codificación, por tanto se debe restar  $x_{k-1}$  del dígito  $y_k$ , como en el siguiente ejemplo:

Si se tiene la secuencia  $\{x_k\} = 0\ 0\ 1\ 0\ 1\ 1\ 0$ , considerando el primer bit como el de partida:

<i>Secuencia binaria <math>\{x_k\}</math></i>	0	0	1	0	1	1	0
<i>Amplitudes en Bipolar <math>\{x_k\}</math></i>	-1	-1	+1	-1	+1	+1	-1
<i>Regla de codificación <math>y_k = x_k + x_{k-1}</math></i>		-2	0	0	0	2	0
<i>Regla de Decisión para Decodificación</i>	Si $\hat{y}_k = 2$ , $\hat{x}_k = +1$ (1 binario) Si $\hat{y}_k = -2$ , $\hat{x}_k = -1$ (0 binario) Si $\hat{y}_k = 0$ , opuesto a la decisión anterior						
		-1	+1	-1	+1	+1	-1
		0	1	0	1	1	0

Tabla. 2.1. Ejemplo Señalización Duobinaria

### 2.2.1.3.3. Precodificación.

Esto se hace codificando primero  $x_k$  en una nueva secuencia  $\{w_k\}$  binaria que resulte de:

$$w_k = x_k \oplus w_{k-1}$$

que representa la adición en módulo dos o una suma ExOr de dígitos binarios.

La nueva secuencia  $\{w_k\}$  se convierte en una secuencia de pulsos bipolares y se procede a la codificación duobinaria de igual forma que en el ejemplo anterior.

Para la decodificación se toma en cuenta la siguiente regla:

$$\text{Si } \hat{y}_k = \pm 2, \hat{x}_k = 0 \text{ binario}$$

$$\text{Si } \hat{y}_k = 0, \hat{x}_k = 1 \text{ binario}$$

De esta forma, usando la precodificación, se puede realizar la decodificación de una secuencia de manera sencilla sin tener que recurrir a las decisiones anteriores, las cuales podrían producir errores. Otra de las ventajas es que si existe un error producido por ruido,

tal error no se propagará a los otros dígitos; y, aunque el primer dígito de la secuencia  $\{w_k\}$  sea determinado de forma arbitraria, el resultado decodificado siempre será el mismo.

<b>Secuencia binaria <math>\{x_k\}</math></b>	0	0	1	0	1	1	0
<b>Secuencia Precodificada <math>w_k = x_k \oplus w_{k-1}</math></b>	0	0	1	1	0	1	1
<b>Amplitudes en Bipolar <math>\{w_k\}</math></b>	-1	-1	+1	+1	-1	+1	+1
<b>Regla de codificación <math>y_k = x_k + x_{k-1}</math></b>		-2	0	+2	0	0	+2
<b>Secuencia Decodificada</b>		0	1	0	1	1	0

Tabla. 2.2. Ejemplo Precodificación para Señalización Duobinaria

### 2.2.2. INTERFERENCIA INTERSIMBÓLICA

La Interferencia Inter Simbólica se produce en los sistemas de comunicación debido a un mal filtraje de las señales durante recorrido, es decir en el transmisor, receptor y canal, lo que provoca que los pulsos de cada símbolo se dispersen en el tiempo y así ocupen cuotas de tiempo adyacentes, interfiriendo con el proceso de detección y degradando el desempeño de la tasa de error.

En el transmisor se modulan los pulsos conteniendo la información y luego se filtran a fin de cumplir con restricciones de ancho de banda. En el canal se encuentran reactancias que producen distorsión de los pulsos. Y en la recepción también se realiza filtrado, pero usando *filtros ecualizados*, los cuales se configuran de forma que compensen la distorsión causada anteriormente por el transmisor y el canal.

Debido a las restricciones de ancho de banda, los pulsos dejan de ser cuadrados y se vuelven redondeados, y entre menor sea el ancho de banda mayor será este problema.

En la figura 2.8 se muestra un modelo del sistema, y del cual se puede concluir una función de transferencia equivalente para todo el proceso:

$$H(f) = H(f)_t H(f)_c H(f)_r$$

Donde  $H(f)_t$  caracteriza al filtraje de transmisión,  $H(f)_c$  al filtraje dentro del canal, y  $H(f)_r$  al filtro ecualizador de recepción.

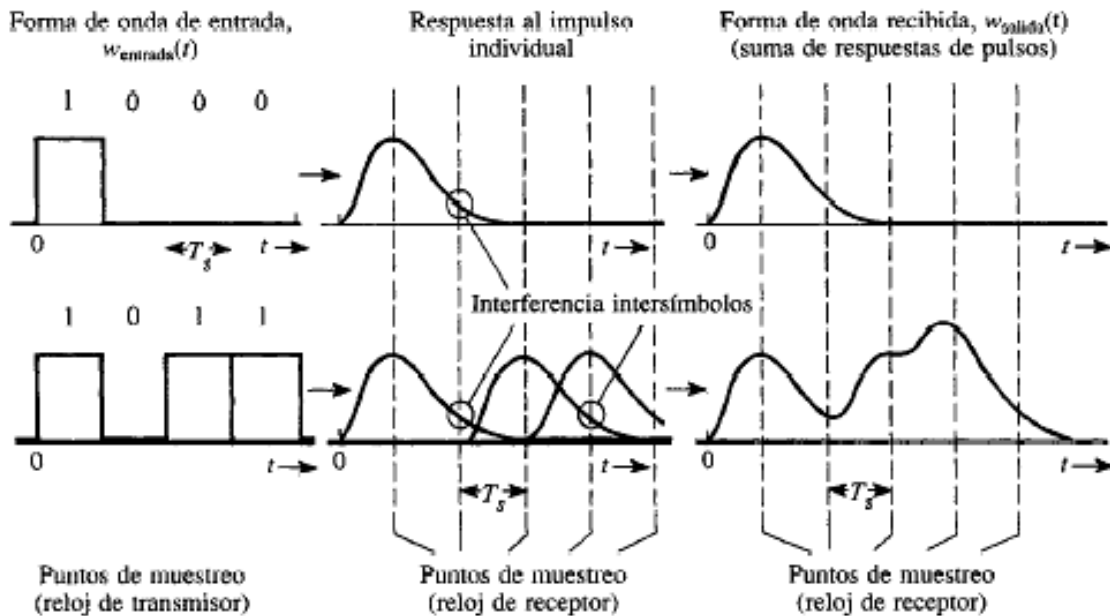


Figura 2.7. Interferencia InterSimbólica en recepción de un Sistema Binario

Cuando  $H(f)$  se selecciona para reducir al mínimo la ISI, entonces el  $H(f)_r$  obtenido es lo que se conoce como **filtro de ecualización**, y depende del  $H(f)_c$ . Cuando la función de transferencia del canal cambia (conmutación telefónica), es posible que se requiera que el filtro de ecualización además sea **adaptable**. Es decir, que se autoajuste para reducir la ISI al mínimo, para lo cual se realizan pruebas que adaptan al filtro electrónicamente, antes de cada cambio, para que la abertura del diagrama del ojo sea la máxima. Estas pruebas se conocen como *secuencias o preámbulos de aprendizaje o entrenamiento*.

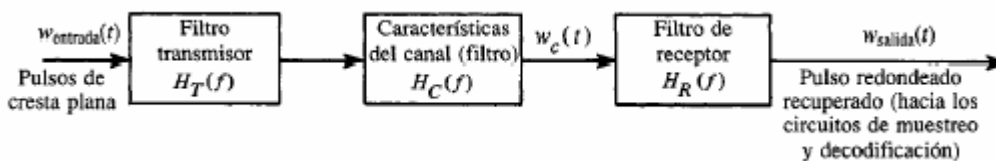


Figura 2.8. Sistema de Transmisión de Pulsos en Banda Base



Nyquist fue el primero en investigar este problema, y mostró que el ancho de banda mínimo para detectar  $R_s$  (tasa de símbolo) sin ISI era  $R_s/2$  Hz. Descubrió tres métodos diferentes para configurar los pulsos que se podían utilizar para eliminar la ISI.

### 2.2.2.1. Primer Método De Nyquist – Cero ISI

Consiste en utilizar una función de transferencia  $H(f)$  tal que la respuesta al impulso satisfaga la condición:

$$h(kTs + t) = \begin{cases} C, & k=0 \\ 0, & k \neq 0 \end{cases}$$

Donde  $k$  es un entero,  $T_s$  el periodo de sincronización de las muestras,  $t$  la separación en los tiempos de sincronización de muestreo en el receptor comparados con los tiempos de sincronización de los símbolos de entrada y  $C$  una constante diferente de cero.

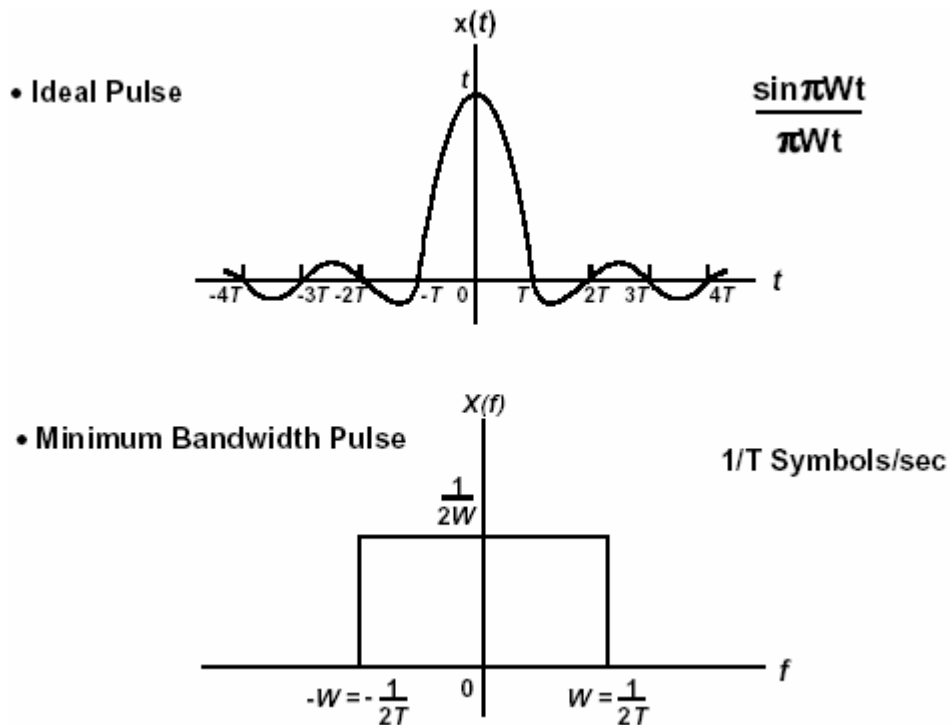


Figura 2.9. Pulso Ideal y mínimo Ancho de Banda de Nyquist

Es decir, para un pulso único de cresta plana de nivel  $a$  presente a la entrada del transmisor cuando  $t$ , el pulso recibido sería  $ah(t)$ . Su valor sería  $aC$  cuando  $t = t$ , aunque no provocaría interferencia en otros tiempos de muestreo debido a la condición de  $h(kTs + t)$  para  $k \neq 0$ .

Ahora si consideramos el pulso ideal de Nyquist de la figura 2.9, y si  $t=0$ , se cumple el primer criterio para ISI cero y el ancho de banda absoluto es  $B = fs/2$ . De esto también se desprende que la señalización en baudios permitida es  $fs = 1/Ts = 2B$ . Sin embargo esta forma de pulso ideal (sinc) plantea dos dificultades prácticas:

- La característica de transferencia  $H(f)$  tiene que ser plana en  $-B < f < B$  y cero en cualquier otra parte, y esto es irrealizable.
- La sincronización del reloj del decodificador tiene que ser casi perfecta puesto que el impulso del sinc ( $x$ ) decae solo en  $1/x$ , y cuando el tiempo de muestreo es exactamente correcto, lo contrario produciría ISI.

Por ello se deben considerar otras formas de pulso con mayor ancho de banda, es decir, formas que pasen por cero en puntos de muestreo adyacentes y que no obstante tengan una envolvente que decae más rápido que  $1/x$ , evitando la fluctuación entre los tiempos de muestreo y por tanto ISI. Una solución para ello, y que tiene muchas características deseables es el filtro reductor en coseno levantado.

#### 2.2.2.1.1. Filtro de Coseno Levantado.

Se caracteriza por la siguiente función de transferencia:

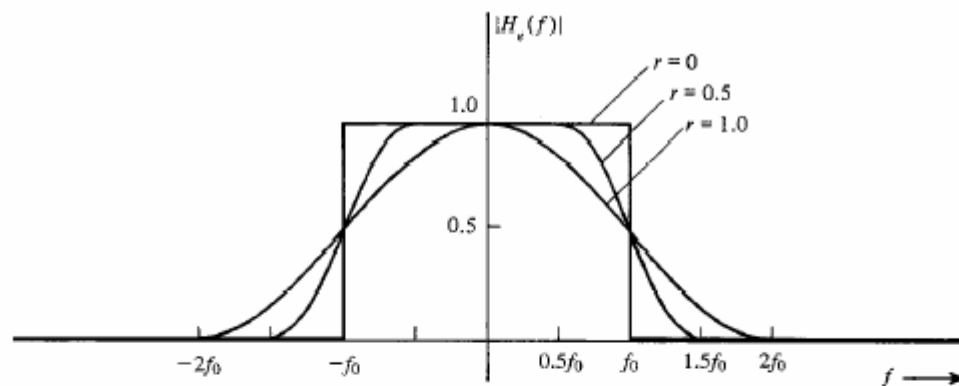
$$H(f) = \begin{cases} 1 & |f| < 2W_0 - W \\ \cos^2\left(\frac{\pi}{4} \frac{|f| + W - 2W_0}{W - W_0}\right) & 2W_0 - W < |f| < W \\ 0 & |f| > W \end{cases}$$

Donde  $W$  es el ancho de banda absoluto,  $W_0 = 1/2T$  representa el mínimo ancho de banda de Nyquist y los 6 dB del espectro del filtro de coseno levantado.

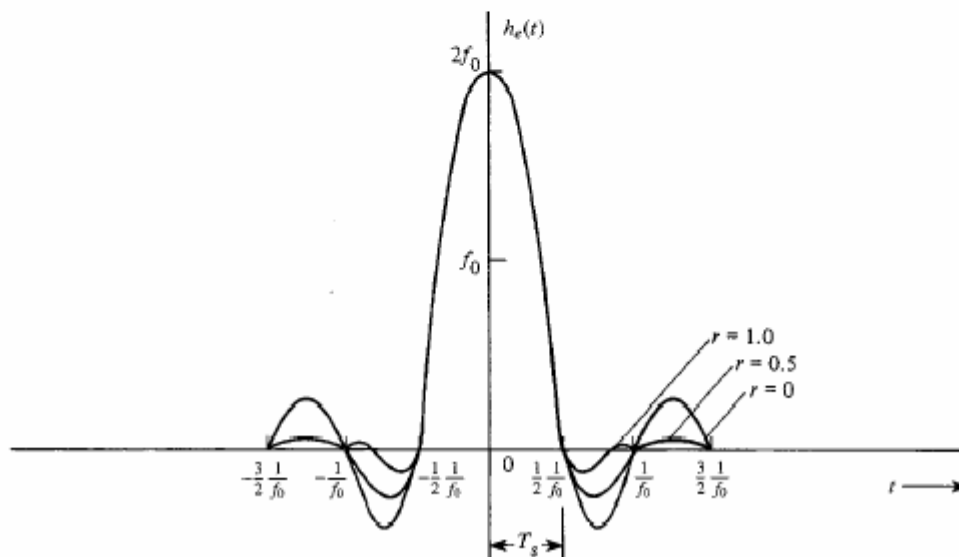
El factor de reducción, o de *roll-off*, se obtiene de la relación entre el “exceso de ancho de banda” y el ancho de banda mínimo, y puede tomar valores de  $\theta = r = 1$

$$r = \frac{W - W_0}{W_0}$$

Cuando es 0, indica el mínimo ancho de banda de Nyquist, y cuando es 1, el máximo exceso (100%) de ancho de banda y produce los lóbulos laterales mas reducidos en el espectro de la función *sinc*, o pulso ideal de Nyquist.



a) Respuesta a la frecuencia de magnitud



b) Respuesta al pulso

Figura 2.10. Características del Filtro de Coseno Levantado para varios factores de Roll-Off

Podemos observar que a medida que aumenta  $r$ , los requisitos de filtración se aligeran, al igual que los de sincronización, puesto que la envolvente de respuesta al impulso decrece más rápido que  $1/|t|$ .

La velocidad de transmisión en baudios del sistema, sin ISI, se relaciona con el ancho de banda absoluto del sistema y el factor de roll-off, y está dada por:

$$R_s = \frac{2W}{1+r}$$

$$W = \frac{1}{2}(1+r)R_s$$

Las señales de modulación pasabanda como ASK o PSK, que requieren un ancho de banda de transmisión de las señales equivalentes banda base, y que ocupan dos veces ese ancho de banda, se conocen normalmente como señales DSB (Double Side Band); y la relación entre su ancho de banda y su tasa de símbolos está dada por:

$$W_{DSB} = (1+r)R_s$$

#### 2.2.2.2. Segundo y Tercer Métodos de Nyquist

El **segundo método de Nyquist** permite introducir ISI de manera controlada, de modo que se pueda eliminar en el receptor y que se puedan recuperar los datos sin error, siempre y cuando no haya ruido presente.

Esta técnica permite la posibilidad de duplicar la velocidad de transmisión en baudios, o reducir el ancho de banda requerido a la mitad. Es lo que se conoce normalmente como Codificación Duobinaria y que se revisó previamente en el presente capítulo.

El **tercer método de Nyquist** para el control o eliminación de ISI comprende la selección de un  $h(t)$  de modo que el área bajo el pulso  $h(t)$  dentro del mismo intervalo de símbolos deseado,  $T_s$ , no es cero, aunque las áreas bajo  $h(t)$  en intervalos de símbolos adyacentes son cero. Para la detección de datos, el receptor evalúa el área bajo la forma de onda del receptor en cada intervalo  $T_s$ . De hecho se han encontrado pulsos que satisfacen este tercer criterio, pero su rendimiento en presencia de ruido es inferior al de los otros ejemplos revisados.

### 2.2.2.3. DIAGRAMA DE OJO.

El diagrama de ojo representa una técnica cómoda para visualizar y analizar la influencia de las imperfecciones de la transmisión sobre la señal, la degradación que se introduce en los pulsos. Este diagrama está definido como la superposición sincronizada de todas las posibles realizaciones de la señal de interés vistas dentro un particular intervalo de señalización.

El diagrama ocular se puede utilizar como método de diagnóstico, representándolo en un osciloscopio cuya base de tiempo (sistema de deflexión horizontal) esté regulada a la frecuencia del reloj de los símbolos, y cuyo sistema de deflexión vertical esté conectado a la señal.

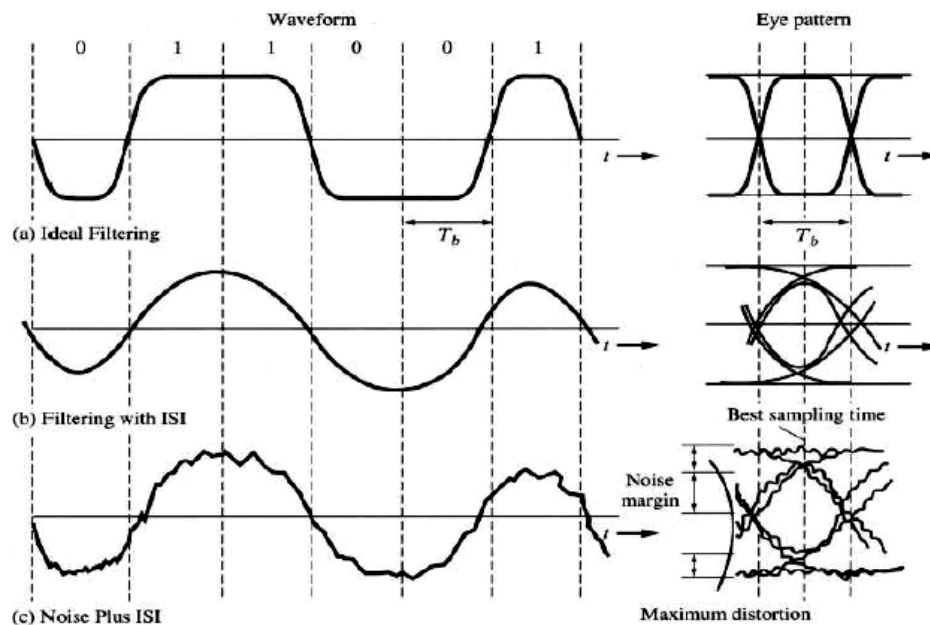


Figura 2.11. Diagramas de Ojo para una Señal: a) Ideal, b) Con ISI, c) Con ISI y Ruido

La abertura del ojo en el diagrama es una medida de la calidad del radio y la probabilidad de error, además es una función del número de niveles de código y la interferencia intersímbolo. El Jitter se puede visualizar con el diagrama de ojo.

Para regenerar los pulsos sin error debe existir:

- Ojo debe estar abierto.
- Las líneas de decisión deben estar dentro del área abierta.

El efecto de la degradación es una reducción en el tamaño del ojo, definiéndose la degradación como:

$$20 \log h/H$$

$h$  = Apertura vertical degradada

$H$  = Apertura vertical ideal

A partir del diagrama ocular también podemos encontrar el mínimo margen de ruido, que consiste en la diferencia de tensión entre el umbral de decisión y la línea del límite interno del ojo, medida en el instante de muestreo.

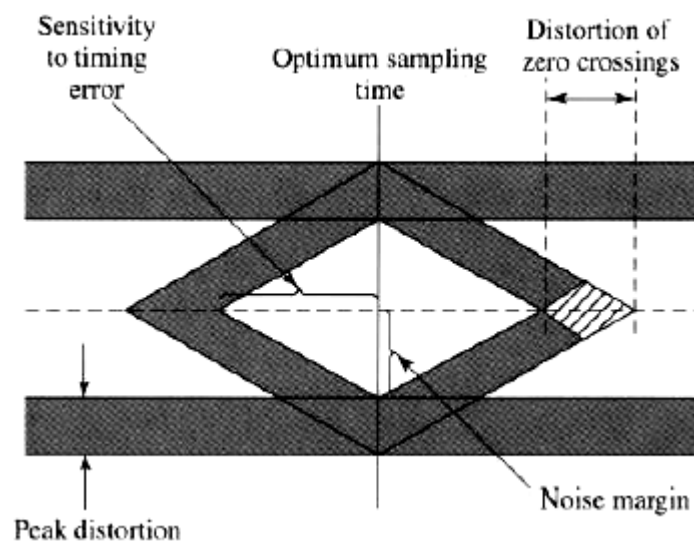


Figura 2.12. Características de un Diagrama de Ojo

### 2.2.3. RECEPTOR ÓPTIMO: PROBABILIDAD DE ERROR

La probabilidad de error de bit, es decir la posibilidad de un bit errado en una secuencia de datos, está dada por

$$P_B = Q\left(\frac{a_1 - a_2}{2s_0}\right)$$

Donde  $\mathbf{a}_1$  y  $\mathbf{a}_2$  representan cuando se transmiten las señales binarias  $\mathbf{s}_1(\mathbf{t})$  y  $\mathbf{s}_2(\mathbf{t})$ ,  $\mathbf{s}_0$  la desviación estándar y  $Q(x)$  la simbología usada generalmente para calcular la probabilidad de error de ruido Gaussiano, y en general, cuando se utiliza la función de densidad de potencia Gaussiana. Conocida también como **función de error complementario** o **función de co-error**, que se define como

$$Q(x) \approx \frac{1}{\sqrt{2\mathbf{p}}} \int_x^{\infty} \exp\left(-\frac{u^2}{2}\right) du$$

Pero  $Q(x)$  no se puede utilizar en esta forma, por lo que para efectos prácticos se utiliza una aproximación, la cual es válida para  $x > 3$ .

$$Q(x) \approx \frac{1}{x\sqrt{2\mathbf{p}}} \exp\left(-\frac{x^2}{2}\right)$$

$Q(x)$  también se denota como **erfc(x)**, como se mostró en estudio de la probabilidad de error para cada uno de los códigos de línea. Sus valores se encuentran incluidos en tablas en los anexos.

Para minimizar  $P_B$  en un canal AWGN, se debe seleccionar el filtro de recepción y el umbral de decisión óptimos. Para el caso binario el umbral de decisión óptimo nos permite tener  $P_B = Q\left(\frac{a_1 - a_2}{2\mathbf{s}_0}\right)$ , y para completar esta optimización es necesario elegir el filtro (**filtro casado**) que maximice el argumento de  $Q(x)$  o su equivalente

$$\frac{(a_1 - a_2)^2}{\mathbf{s}_0^2}$$

Donde  $(a_1 - a_2)$  es la diferencia de las componentes de la señal deseada a la salida del filtro en un tiempo  $t = T$ , y el cuadrado de esta diferencia es la potencia instantánea. Si consideramos que un filtro casado alcanza su máxima salida de SNR en  $2E/N_0$ , se puede establecer la SNR en  $t = T$  como

$$\left(\frac{S}{N}\right)_T = \frac{(a_1 - a_2)^2}{\mathbf{s}_0^2} = \frac{2Ed}{N_0}$$

Donde  $N_0/2$  es la densidad de potencia de ruido de doble lado a la entrada del filtro y  $Ed$  es la energía de la diferencia de señales a la entrada del filtro.

Si utilizamos esta última ecuación en  $P_B = Q\left(\frac{a_1 - a_2}{2s_0}\right)$ , podemos obtener

$$P_B = Q\left(\sqrt{\frac{Ed}{2N_0}}\right)$$

Desde esta ecuación se puede desarrollar otra más general en función de la energía de bit recibido, tomando en cuenta que  $Ed = 2Eb(1-p)$ , por lo que la nueva ecuación es

$$P_B = Q\left(\sqrt{\frac{Eb(1-r)}{N_0}}\right)$$

Donde  $-1 = p = 1$  y es el valor de la correlación de  $S_1$  y  $S_2$ , por lo que si  $p = 1$ , quiere decir que están perfectamente correlacionadas y eso no es posible, si  $p = -1$  son señales antipodales y si  $p = 0$ , son señales ortogonales.

Es decir, que si el filtro casado detecta señales antipodales, la probabilidad de error de bit es

$$P_B = Q\left(\sqrt{\frac{2Eb}{N_0}}\right)$$

Y si son señales ortogonales  $P_B = Q\left(\sqrt{\frac{Eb}{N_0}}\right)$

Lo que significa que las magnitudes de las señales utilizadas son  $\sqrt{Eb}$  y que estas ecuaciones son función de la distancia entre  $S_1$  y  $S_2$ , por lo que a mayor distancia, menor  $P_B$ .

Para los diferentes casos de señalización binaria, es decir, para los diferentes códigos de línea, la probabilidad de error y su desempeño (eficiencia de ancho de banda) se encuentran descritos en la primera parte de este capítulo, y que podrán ser mejor comprendidos luego del estudio de esta última sección.



## 2.3. PRÁCTICAS CON MATLAB

### 2.3.1. CÓDIGOS DE LÍNEA.

#### 2.3.1.1. Objetivos

- Investigar como se codifica de forma serial la información binaria para la transmisión en frecuencia de banda base.
- Revisar de forma gráfica los diferentes métodos que son usados en aplicaciones de comunicación de datos en la actualidad en banda base.
- Observar las principales características que diferencian a cada uno de los códigos de los demás y entender el significado de su nomenclatura.
- Obtener las funciones de densidad espectral de potencia asociados con los diferentes códigos de línea.

#### 2.3.1.2. Procedimiento

Para la realización de la presente práctica se utilizan funciones realizadas para MATLAB y cuyo código fuente se encuentran disponibles en los archivos del mismo nombre. Estas funciones simplifican el desarrollo y mejoran el entendimiento y alcance de los objetivos planteados. Las funciones han sido colocadas en el mismo directorio de la práctica y son compatibles con MATLAB7, que es la versión utilizada en el desarrollo de este proyecto.

1. Se genera una secuencia binaria aleatoria la cual va a ser codificada con los diferentes códigos de línea. Para ello se usa la función **binary()**, cuyo argumento es el número de dígitos binarios deseados para la secuencia.

También se puede utilizar una secuencia fija, para lo cual nosotros designamos un vector de la longitud deseada con los valores binarios que sean requeridos.

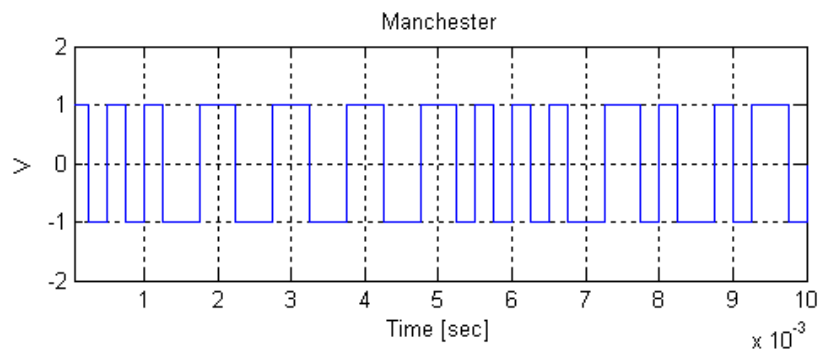
La secuencia generada con la función `binary(20)`, para este caso es

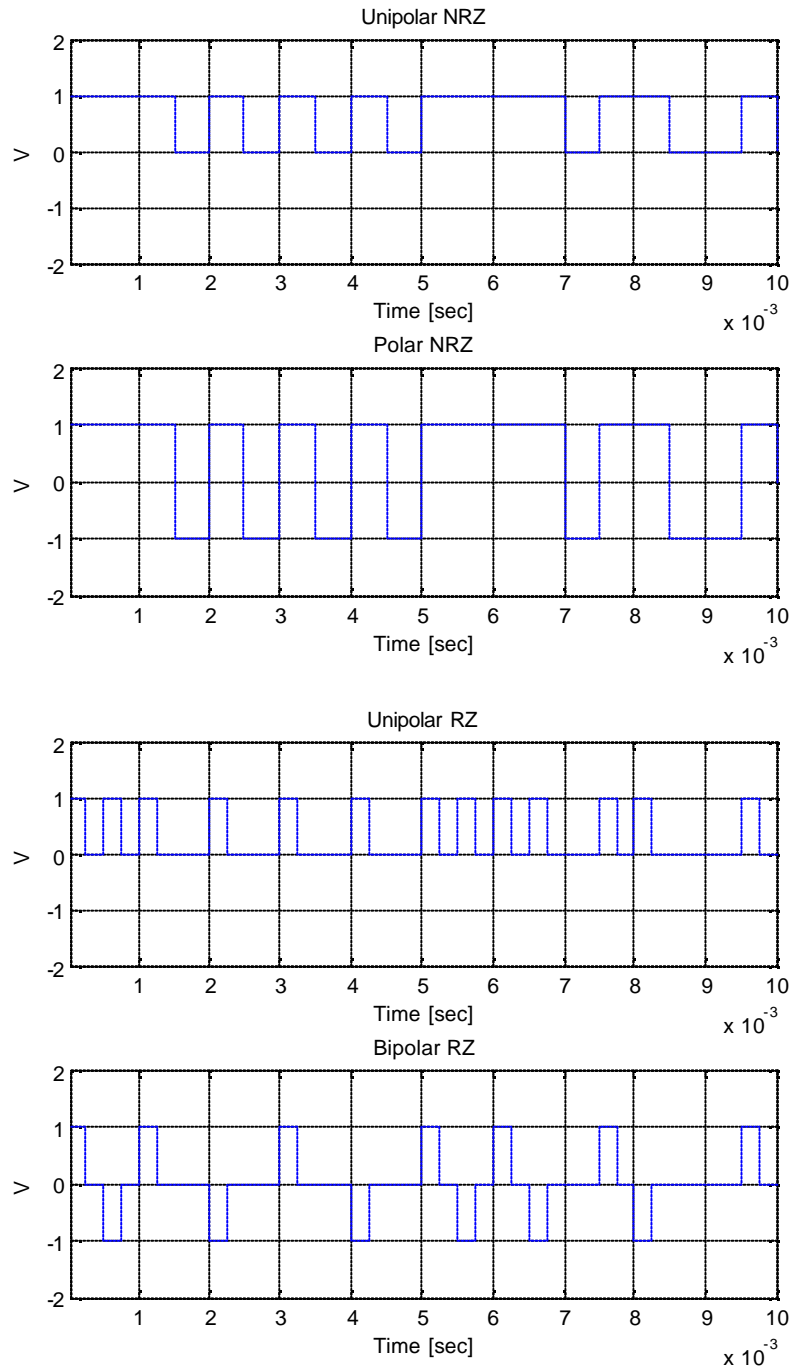
$$\mathbf{b} = [1\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1]$$

2. Con ayuda de la función **wave\_gen** (**secuencia**, '**codigo\_linea**', **R<sub>b</sub>**) generamos los diferentes códigos de línea para la secuencia dada. Sus argumentos son la "secuencia" generada o determinada en el punto anterior, el "código de línea" en el que se desea codificar y **R<sub>b</sub>** es la tasa de bit de la secuencia de datos en Hz.

Los códigos posibles son:

- Unipolar NRZ ('unipolar\_nrz')
  - Polar NRZ ('polar\_nrz')
  - Unipolar RZ ('unipolar\_rz')
  - Bipolar RZ ('bipolar\_rz')
  - Manchester ('manchester')
3. Luego se dibuja los códigos de línea generados en el paso anterior por medio de la función **waveplot()**, cuyo argumento es el la secuencia codificada. Para poder apreciar las diferencias entre los diferentes códigos de una misma secuencia a una misma tasa de bits (**R<sub>b</sub>**), se presentan la figura 2.13, gráficas obtenidas de la ejecución del archivo **Cod\_linea.m** que es el programa principal de esta práctica y que contiene el código fuente de lo descrito anteriormente y lo que se presenta adelante en la misma práctica. La señal se encuentra muestreada 10 veces en cada tiempo de bit, para obtener resultados más claros. Este código se incluye al final.



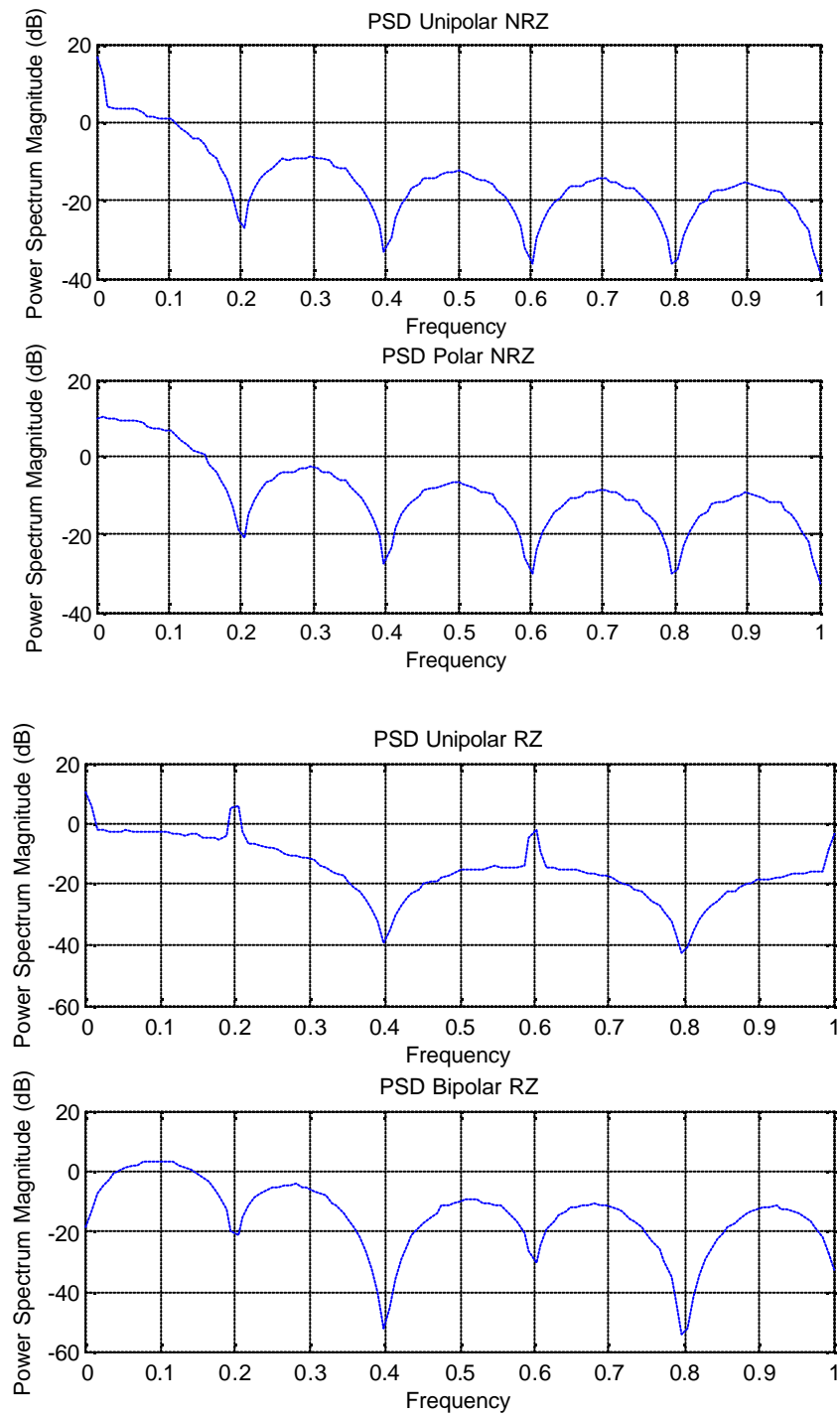


**Figura 2.13. Formas de Onda de Diferentes códigos de Línea**

- Para poder determinar la densidad espectral de potencia de los diferentes códigos de línea usamos la función **psd()**, cuyo argumento es la secuencia de datos codificada como se describió en el numeral anterior.

En la figura 2.14, se muestra la densidad espectral de todos los códigos de línea generados en el paso anterior para poder apreciar sus diferencias. Para ello se ha generado una nueva secuencia de datos binarios, **bn**, de 2500 elementos, que

permite apreciar la densidad espectral de potencia de mejor manera que si se lo hiciera con pocos datos (b).



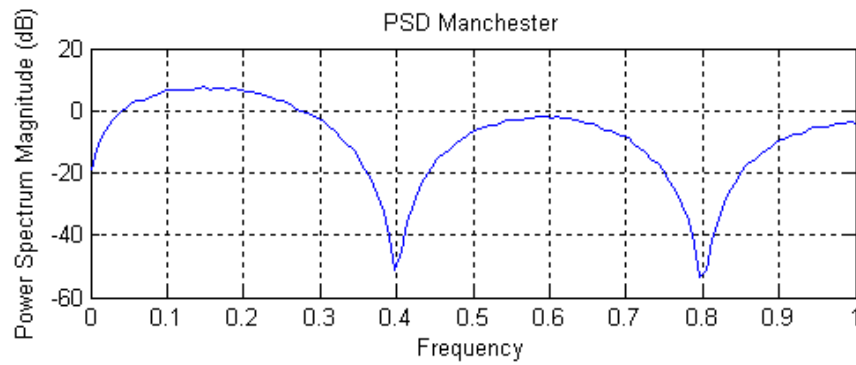


Figura. 2.14. PSD para Diferentes códigos de línea, Igual Secuencia

### 2.3.1.3. Conclusiones

- La información serial binaria se puede codificar de diferentes formas para poder ser transmitida por medio de un canal de comunicación banda base haciendo uso de los denominados códigos de línea.
- Se pudo apreciar gráficamente porque un código es de retorno a cero o de no retorno a cero, así como su polaridad si es una o doble.
- Los códigos de línea unipolar de retorno a cero (RZ) y Manchester ocupan el doble de ancho de banda que los sistemas de no retorno a cero (NRZ) y el bipolar. Esto se puede visualizar en los diagramas de densidad espectral de potencia.
- La densidad espectral de potencia también depende la tasa de bits, a mayor tasa de bit mayor será la densidad espectral de potencia (PSD).

### Código Cod\_linea.m

```
%Generacion de los Diferentes Codigos de Linea para
%una secuencia binaria de datos y densidad espectral de potencia (PSD).

clc;
clear all;
%*****

SAMPLING_CONSTANT = 10;
BINARY_DATA_RATE = 1000;
SAMPLING_FREQ = BINARY_DATA_RATE * SAMPLING_CONSTANT;

global SAMPLING_CONSTANT;
```

```
% *****  
  
b=binary (20);  
  
unr = wave_gen(b,'unipolar_nrz',2000);  
pnr = wave_gen(b,'polar_nrz',2000);  
upr = wave_gen(b,'unipolar_rz',2000);  
bpr = wave_gen(b,'bipolar_rz',2000);  
man = wave_gen(b,'manchester',2000);  
  
figure(1)  
subplot(2,1,1),waveplot(unr)  
subplot(2,1,2),waveplot(pnr)  
figure(2)  
subplot(2,1,1),waveplot(upr)  
subplot(2,1,2),waveplot(bpr)  
figure(3)  
subplot(2,1,1),waveplot(man)  
  
% *****  
  
bn=binary (2500);  
  
unr2 = wave_gen(bn,'unipolar_nrz',3000);  
pnr2 = wave_gen(bn,'polar_nrz',3000);  
upr2 = wave_gen(bn,'unipolar_rz',3000);  
bpr2 = wave_gen(bn,'bipolar_rz',3000);  
man2 = wave_gen(bn,'manchester',3000);  
  
figure(4)  
subplot(2,1,1),psd(unr2)  
subplot(2,1,2),psd(pnr2)  
figure(5)  
subplot(2,1,1),psd(upr2)  
subplot(2,1,2),psd(bpr2)  
figure(6)  
subplot(2,1,1),psd(man2)  
  
clc;  
b
```

## 2.3.2. ERRORES EN MODULACION BANDABASE

### 2.3.2.1. Objetivos

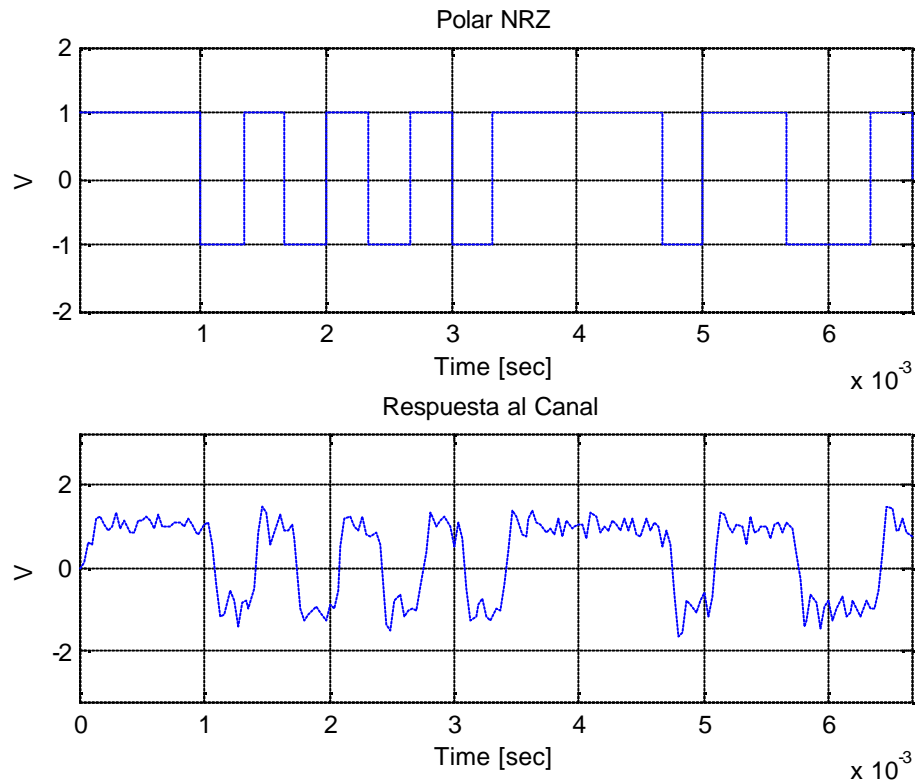
- Observar la forma en que es afectada una señal transmitida en modulación banda base por el incremento de ruido AWGN en el canal de transmisión.
- Aprender la degradación sufrida por una señal transmitida en banda base ante la restricción del ancho de banda disponible.
- Obtener los diagramas de ojo de las señales afectadas por el incremento de ruido en el canal de transmisión banda base.
- Obtener los diagramas de ojo de las señales degradadas por la restricción de ancho de banda del canal de transmisión.

### 2.3.2.2. Procedimiento

Para la realización de la presente práctica se utilizan funciones realizadas para MATLAB y cuyo código fuente se encuentran disponibles en los archivos del mismo nombre. Estas funciones simplifican el desarrollo y mejoran el entendimiento y alcance de los objetivos planteados. Las funciones han sido colocadas en el mismo directorio de la práctica y son compatibles con MATLAB7, que es la versión utilizada en el desarrollo de este proyecto. El código fuente de esta práctica se encuentra disponible en el archivo **Error\_BB.m**.

1. Se pueden determinar los efectos del ruido en un canal, así como del ancho de banda del mismo. Para ello, se usa la función **channel(entrada, ganancia, potencia\_ruido, ancho\_banda)**, la cual permite obtener la respuesta de una señal al canal, pudiendo modificarse los valores de potencia de ruido en Watts y ancho de banda en Hz. La entrada es la secuencia codificada que ingresa al canal, la ganancia del canal generalmente se considera la unidad.

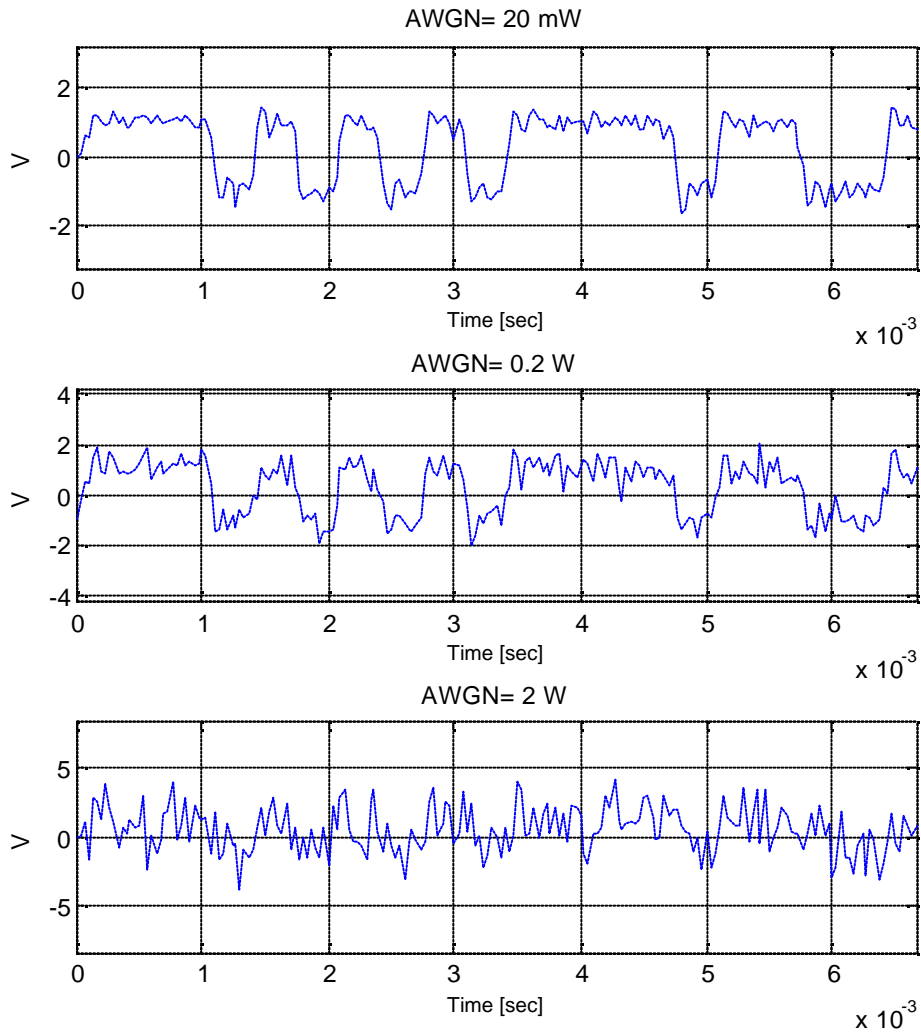
Se puede observar la respuesta a un canal de 10 KHz con un AWGN de 20 mW, con ganancia 1 y cuya entrada sea la señal 'polar\_nrz', en la figura 2.15.



**Figura 2.15. Respuesta al Canal para Polar NRZ y Señal Original**

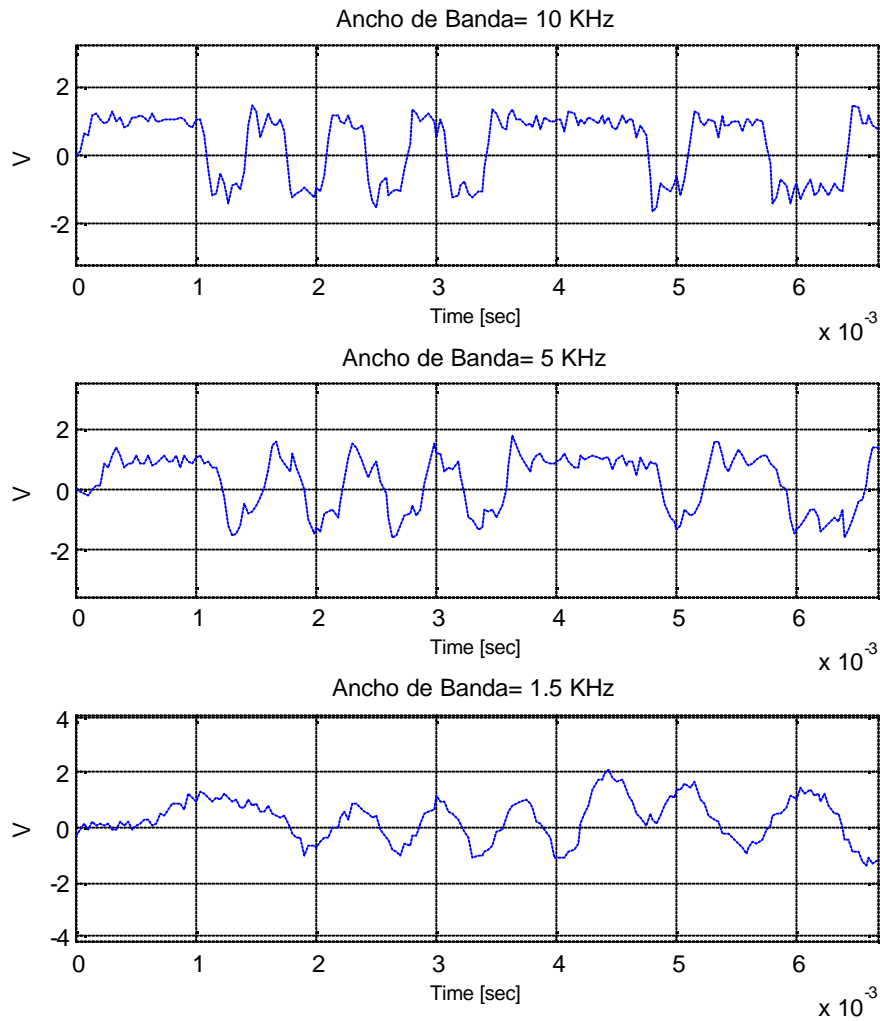
2. Si se altera la cantidad de AWGN introducido en el canal, se puede observar la degradación que sufre la señal para diferentes valores de potencia de ruido. Los resultados de ello, para valores de 20 mW, 0.2W y 2W se muestran en la figura 2.16 para la misma secuencia del paso anterior.





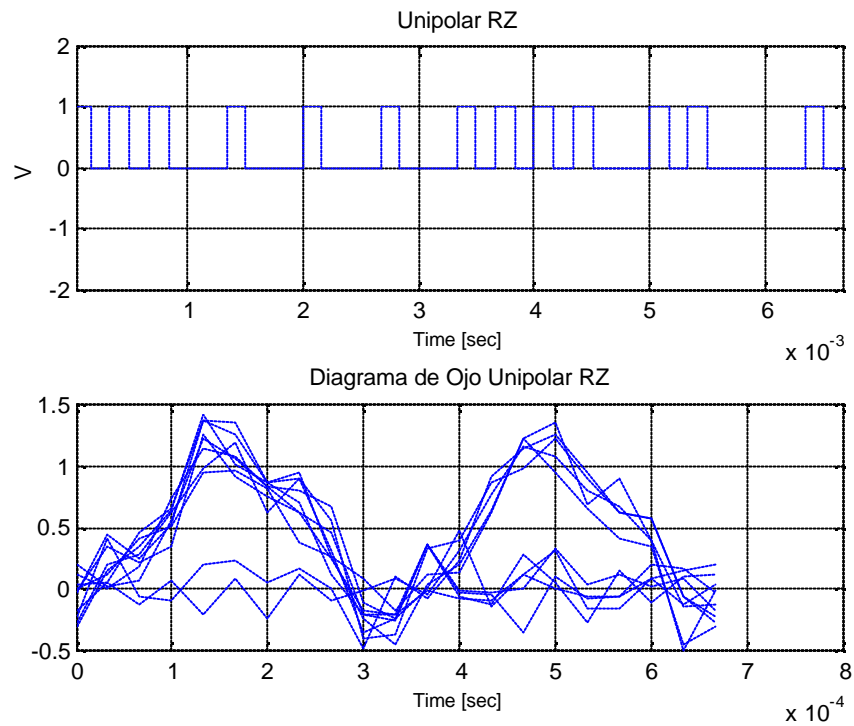
**Figura 2.16. Degradación de Señal causada por AWGN**

3. Si se altera el ancho de banda disponible para realizar la transmisión, se puede observar la degradación que sufre la señal para diferentes valores de ancho de banda. Los resultados de ello, para valores de 10 KHz, 5 KHz, 1500 Hz se muestran en la figura 2.17 para la misma secuencia del paso 1.



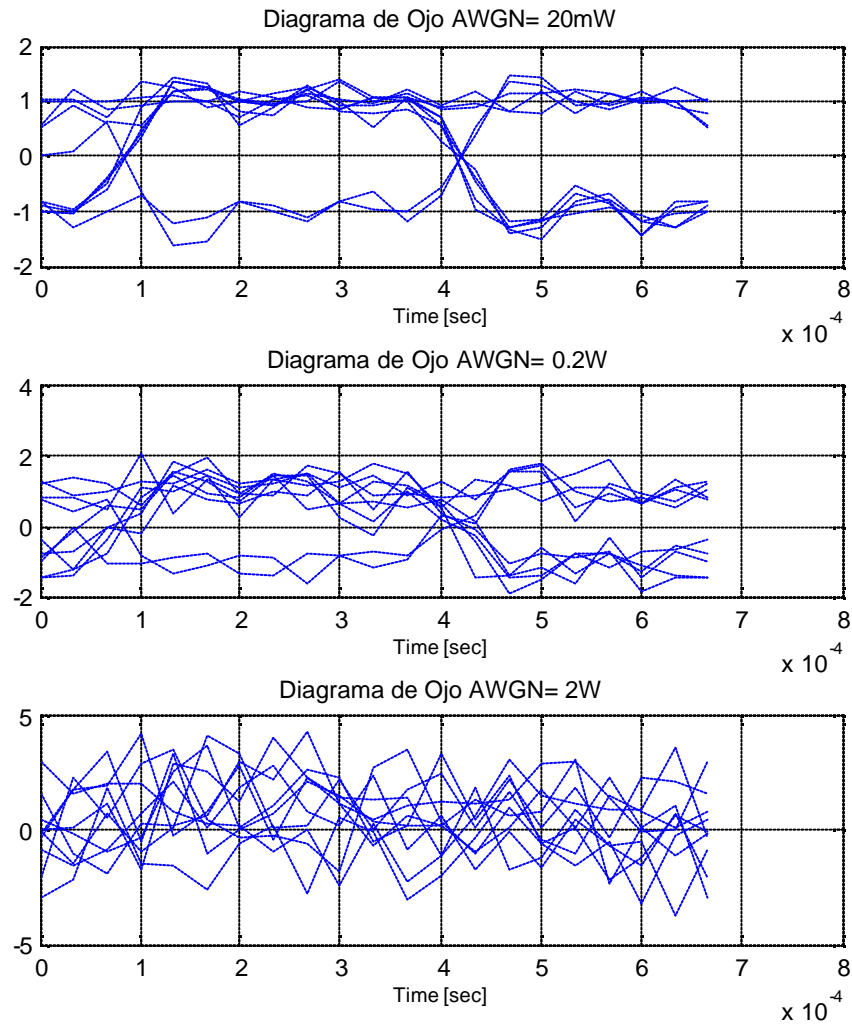
**Figura 2.17. Degradación de Señal Causada por Restricción de Ancho de Banda**

- Los efectos del canal y el ruido pueden ser mejor visualizados mediante el diagrama del ojo. Este diagrama de ojo se genera mediante barridos, que son ligeramente más anchos que el periodo de bit  $T_b = 1/R_b$ . En este caso se usará un ancho de barrido de  $2T_b$ . La figura 2.18 muestra el diagrama del ojo para la secuencia generada para un código unipolar RZ.



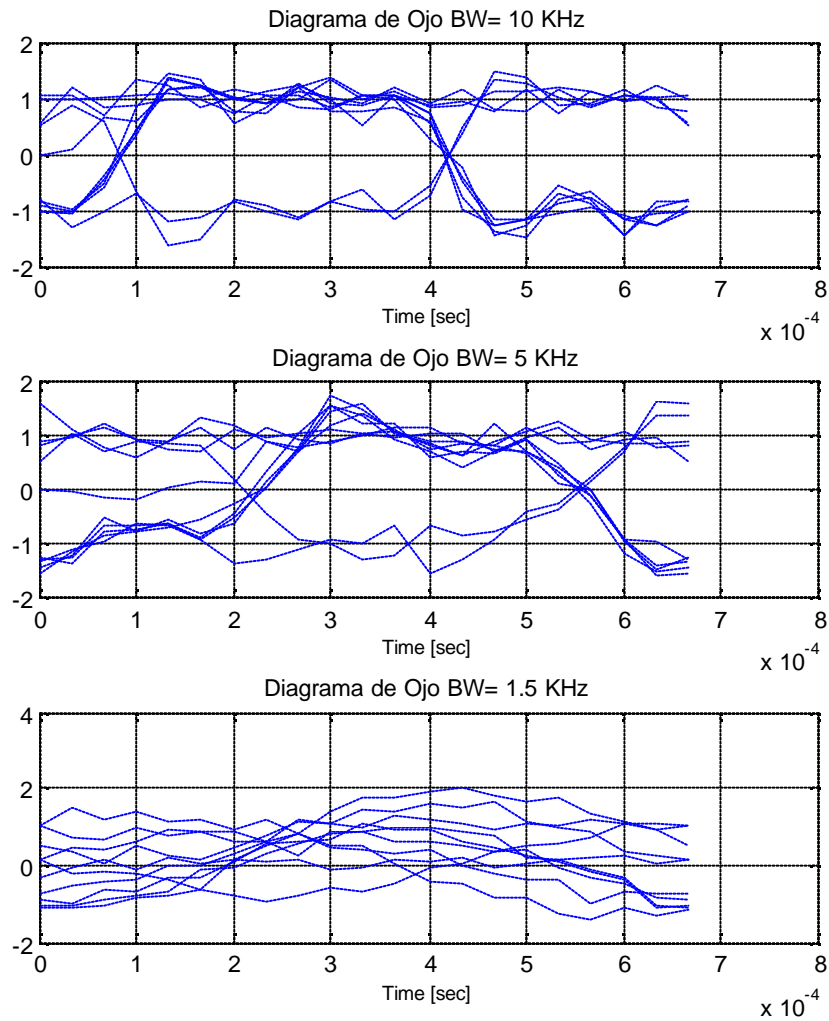
**Figura 2.18. Diagrama de Ojo para Señal Unipolar RZ y Señal Considerada**

5. Para poder visualizar el efecto de la introducción de AWGN en un canal a través del diagrama del ojo, consideramos los valores del paso 2 y los resultados de esta simulación se muestran en la figura 2.19.



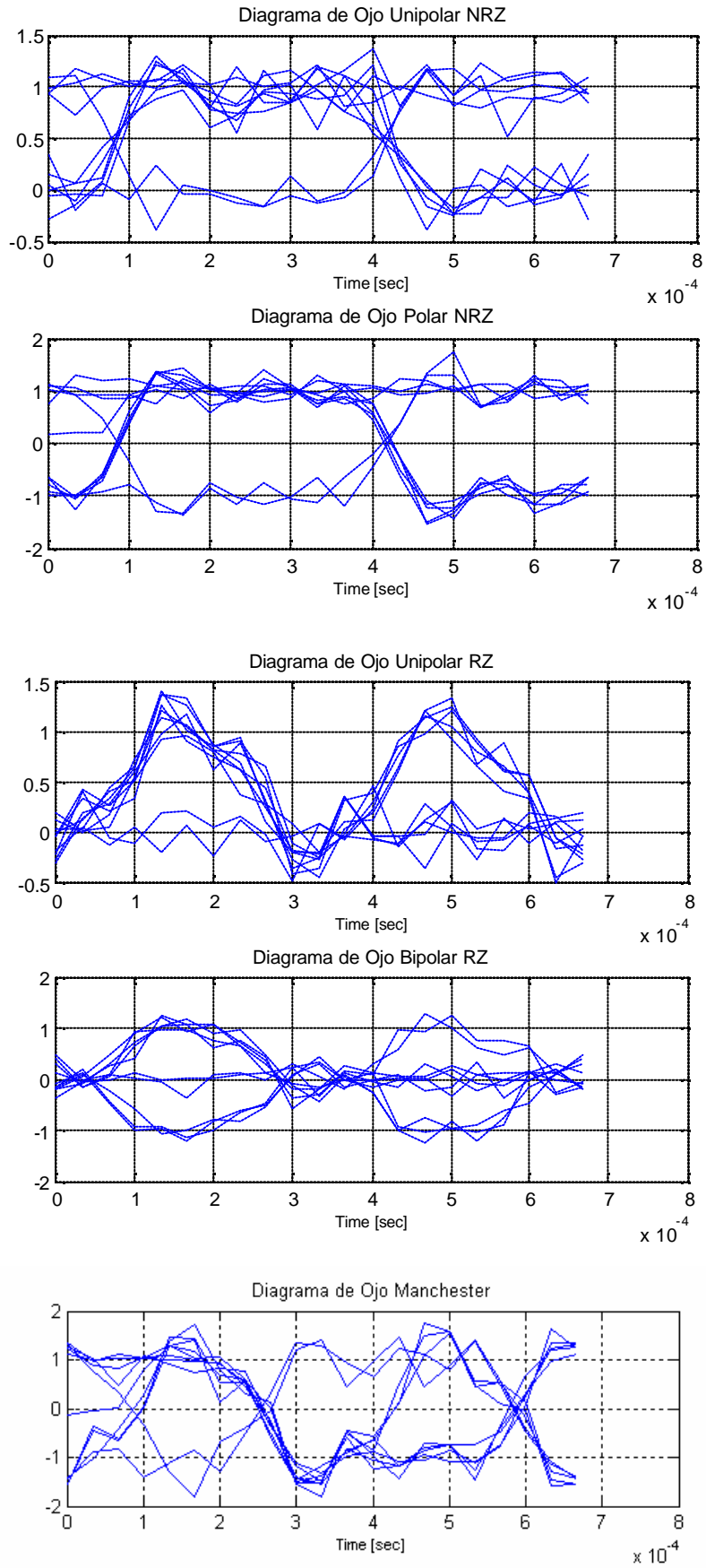
**Figura 2.19. Diagrama de Ojo para Señal Degradada por Ruido**

6. Para poder visualizar el efecto de la restricción de ancho de banda en un canal, a través del diagrama del ojo, consideramos los valores del paso 3 y los resultados de esta simulación se muestran en la figura 2.20.



**Figura 2.20. Diagrama de Ojo para Señal Degradada por Reducción Ancho de Banda**

7. Los diagramas de ojo para los diferentes códigos de línea se muestran en la figura 2.21, para un canal de ganancia 1, AWGN de 20mW y 10000 Hz de ancho de banda. Se deben de generar previamente las secuencias de cada código de línea como se revisó en la práctica anterior



**Figura 2.21. Diagrama de Ojo para Diferentes códigos de línea**

### 2.3.2.3. Conclusiones

- Como se puede observar en la figura 2.15, las variantes de ruido en un canal de transmisión cambian la forma de onda de la señal transmitida volviéndola irregular; lo que si ocurre en alto grado hace imposible la comunicación.
- A mayor densidad de ruido introducido en un canal mayor es la degradación que sufre una señal, como se observa en la figura 2.16. Llegando a perderse totalmente la apertura del ojo para el último caso.
- A menor disponibilidad de ancho de banda en un canal de transmisión, mayor es la degradación que sufre una señal, como se observa en la figura 2.17. En el último caso la señal se ve muy afectada y prácticamente se ha perdido toda la información.
- Los diagramas de ojo para una misma secuencia difieren bastante en su forma, su nivel y apertura dependiendo del código de línea usado en su codificación.
- Por lo mostrado en la figura 2.21, se puede considerar que para las condiciones de canal dadas en el paso 7, los códigos de no retorno a cero, son los que ofrecen menor afectación y son probablemente los que posean menor número de errores en su decodificación.

### Código Error\_BB.m

% Respuesta al Canal, Insercion de Ruido, Restriccion de  
% Ancho de Banda. Diagrama de Ojo.

clc;

clear all;

SAMPLING\_CONSTANT = 10;

BINARY\_DATA\_RATE = 1000;

SAMPLING\_FREQ = BINARY\_DATA\_RATE \* SAMPLING\_CONSTANT;

global SAMPLING\_CONSTANT;

% \*\*\*\*\*

b=binary (20);

unr = wave\_gen(b,'unipolar\_nrz',2000);

pnr = wave\_gen(b,'polar\_nrz',2000);

upr = wave\_gen(b,'unipolar\_rz',2000);

bpr = wave\_gen(b,'bipolar\_rz',2000);

man = wave\_gen(b,'manchester',2000);

% \*\*\*\*\*

```
bn=binary (2500);

pnr2 = wave_gen(bn,'polar_nrz',3000);

y=channel(pnr,1,0.02,10000);

figure(1)
subplot(2,1,1),waveplot(pnr)
subplot(2,1,2),waveplot(y)

yn2=channel(pnr,1,0.2,10000);
yn3=channel(pnr,1,2,10000);

figure(2)
subplot(3,1,1),waveplot(y)
subplot(3,1,2),waveplot(yn2)
subplot(3,1,3),waveplot(yn3)

ybw2=channel(pnr,1,0.02,5000);
ybw3=channel(pnr,1,0.02,1500);

figure(3)
subplot(3,1,1),waveplot(y)
subplot(3,1,2),waveplot(ybw2)
subplot(3,1,3),waveplot(ybw3)

figure(4)

upry=channel(upr,1,0.02,10000);
subplot(2,1,1),waveplot(upr)
subplot(2,1,2),eye_diag(upry)

figure(5)
subplot(3,1,1),eye_diag(y)
subplot(3,1,2),eye_diag(yn2)
subplot(3,1,3),eye_diag(yn3)

figure(6)
subplot(3,1,1),eye_diag(y)
subplot(3,1,2),eye_diag(ybw2)
subplot(3,1,3),eye_diag(ybw3)

unry=channel(unr,1,0.02,10000);
pnry=channel(pnr,1,0.02,10000);
bpriy=channel(bpr,1,0.02,10000);
many=channel(man,1,0.02,10000);

figure(7)
subplot(2,1,1),eye_diag(unry)
subplot(2,1,2),eye_diag(pnry)
figure(8)
subplot(2,1,1),eye_diag(upry)
subplot(2,1,2),eye_diag(bpriy)
figure(9)
subplot(2,1,1),eye_diag(many)

clc;
```



## 2.4. REFERENCIAS

### 2.4.1. LIBROS

- SKLAR, Bernard, **Digital Communications**, Segunda Edición, Prentice Hall, Estados Unidos 2001, páginas 106-152.
- STREMLER, Ferrel, **Sistemas de Comunicación**, Segunda Edición, Editorial AlfaOmega, México 1989, páginas 505-558.
- COUCH, Leon, **Sistemas de Comunicación Digitales y Analógicos**, Quinta Edición, Prentice Hall, México 1998, páginas 157-184.
- ESPE, **Comunicación Digital**, apuntes de la asignatura, semestre septiembre 2004 – marzo 2005.

### 2.4.2. INTERNET.

- <http://bips.bi.ehu.es/~inma/psc/tema2.pdf>, Transmisión Banda Base.
- <http://www.ingelec.uns.edu.ar/fsc2718/CIII-IV.pdf>, Pulsos Banda Base.

## CAPÍTULO III

### TRANSMISIÓN PASA BANDA M-ARIA

#### 3.1. INTRODUCCIÓN.

Como se vio en los capítulos anteriores, la modulación digital es el proceso por el cual los símbolos digitales son transformados en formas de onda que son compatibles con las características de un canal para que puedan ser transmitidas.

En el caso de la modulación en banda base, las ondas toman la forma de pulsos desde el origen (frecuencia cero) hasta un ancho de banda determinado, pero existen canales telefónicos, de radio y satelitales con frecuencias que están lejos del origen y se los conoce como **pasabanda**. Al igual que en la transmisión analógica, la información puede ser transmitida por estos canales pero para ello los pulsos deben modular una sinusoidal u **onda portadora**.

Esta modulación se puede ser en amplitud, fase o frecuencia o una combinación de amplitud y fase, y lo que persigue es trasladar el contenido de la señal de información a una frecuencia apropiada para pasar por dichos canales. Esto se realiza ya que la señal de la portadora se convierte en un campo electromagnético y se requieren antenas para su transmisión en el espacio libre, y dado que el tamaño de dichas antenas depende de la longitud de onda de la señal y es inversamente proporcional a la frecuencia utilizada en la transmisión. Es decir, a mayor frecuencia, menor tamaño de las antenas, lo que convierte a este tipo de modulación en un paso esencial para la transmisión de radio señales.

También con la modulación pasabanda se puede realizar la utilización de un mismo canal por más de una señal y reducir al mínimo los efectos de la interferencia, y además colocar una señal de información en la frecuencia más adecuada para cumplir con requerimiento de diseño de una manera más fácil. Esto último es el caso cuando las señales de Radio Frecuencia (RF) son convertidas en señales de frecuencia intermedia (IF) en un receptor.

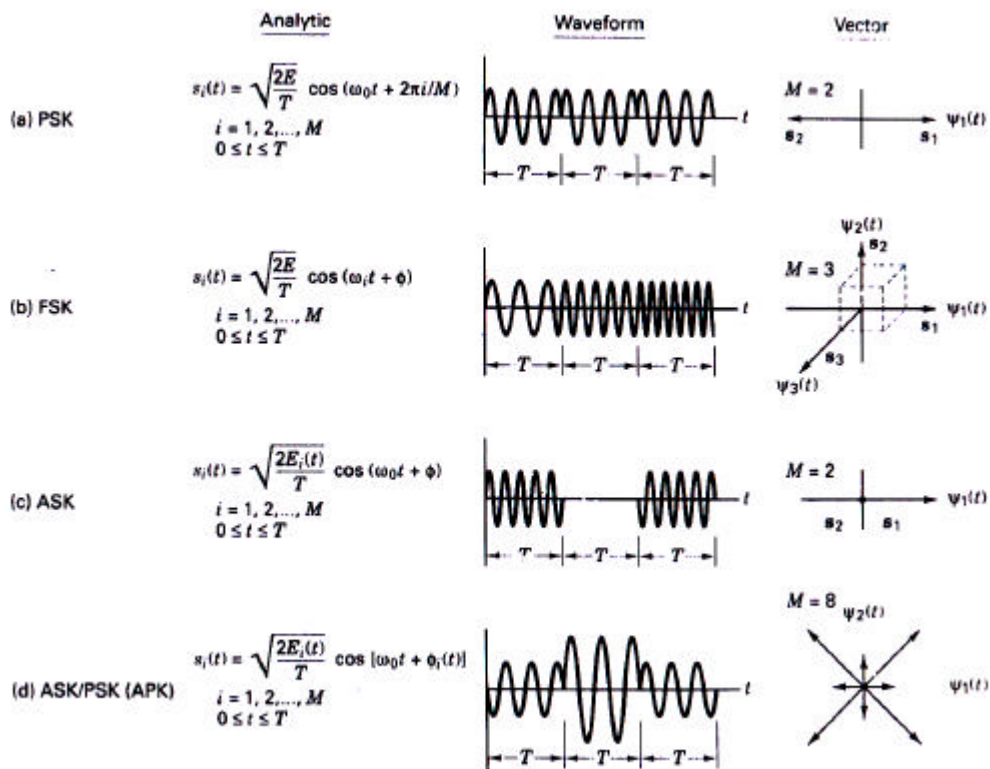


Figura 3.1. Varios Tipos de modulación Digital Pasabanda

En este capítulo se revisan los diferentes tipos de modulación en amplitud, fase y frecuencia, así como sus características espectrales, los moduladores y demoduladores que se requieren.

Se estudian también los diferentes tipos de receptores y de detección para poder trabajar con estos tipos de modulación. Y, finalmente se evalúa el desempeño de los diferentes sistemas en función de la probabilidad de error asociada a su utilización en canales AWGN.

## 3.2. MARCO TEÓRICO

### 3.2.1. RECEPTORES

En la modulación / demodulación pasabanda, cuando un receptor cuenta con información de la fase de portadora para detectar las señales, se dice que la detección es **coherente**, y por el contrario si no cuenta con esa información de referencia de la fase, la detección es **no coherente**. Los términos demodulación y detección se usan indistintamente en comunicaciones, aunque específicamente, la demodulación está referida a la recuperación de la forma de onda y la detección al proceso de detección de símbolos.

Para la **detección coherente** en el receptor existen una serie de prototipos de las señales esperadas, durante la demodulación el receptor correlaciona la señal a la entrada con estos prototipos. Dentro de los tipos de modulación con este tipo de detección entran PSK, FSK, ASK, CPM y algunos híbridos de estos.

La **demodulación no coherente** debido a que carece de la información de la fase de la señal recibida no requiere de estimación de fase. Así, una ventaja de estos sistemas es su menor complejidad frente a los coherentes, aunque tienen mayor probabilidad de error  $P_E$ . Los tipos de modulación que usan esta detección son DPSK, FSK, ASK, CPM y los híbridos. La inclusión de DPSK dentro de este grupo se debe a que es un tipo especial de modulación de fase (**diferencial**) que utiliza la información de fase del símbolo anterior como referencia para poder detectar el símbolo actual.

El proceso de detección es virtualmente idéntico al usado en banda base, ya que la señal recibida en pasabanda se transforma primero en banda base para antes de realizar la detección final. En sistemas lineales las matemáticas de la detección no son afectadas por un cambio o desplazamiento en la frecuencia, cumpliéndose con ello el teorema del heterodinaje, y permitiendo entender el por qué de que la mayoría de las simulaciones de procesamiento de señales lineales se realicen en banda base, aprovechando su mayor simplicidad.

### 3.2.1.1. Regiones de Decisión

Si se considera un espacio bidimensional para una señal con dos vectores binarios perturbados con ruido ( $\mathbf{s}_1 + \mathbf{n}$ ) y ( $\mathbf{s}_2 + \mathbf{n}$ ) y dado que el vector  $\mathbf{n}$  es de media cero, nos queda en la recepción el vector  $\mathbf{r}$ , con media  $\mathbf{s}_1$  o  $\mathbf{s}_2$ . Así, la tarea del detector es determinar cual de las dos posibilidades es la que ha llegado, y usualmente se lo hace en función de la mínima  $P_E$ .

Para el caso donde  $\mathbf{s}_1$  o  $\mathbf{s}_2$  son igualmente probables, con AWGN, la regla de decisión del mínimo error esta dada por

$$d(r, s_i) = \|r - s_i\|$$

Por la cual se escoge la señal cuya distancia sea la mínima.

Para delimitar las regiones de decisión, en el plano, se dibuja una línea conectando los extremos de los vectores  $\mathbf{s}_1$  y  $\mathbf{s}_2$ , luego una línea que pase por la mitad de la línea de unión y el origen de coordenadas, esa línea dará el lugar de los todos los puntos equidistantes de ambos vectores. Ese lugar geométrico es el límite de la región de decisión 1 y la región de decisión 2. Con esto, si la señal recibida  $\mathbf{r}$  cae en la región 1, entonces se escoge  $\mathbf{s}_1$  y de forma similar para  $\mathbf{s}_2$  en la región 2.

### 3.2.1.2. Receptor de Correlación

Este tipo de detección es muy similar a la que se hace en la modulación banda base, se referirá básicamente al uso del filtro casado o correlator y se considerará el caso general de detección M-aria, con degradación producida solo por AWGN y con una señal transmitida de tipo

$$r(t) = s_i(t) + n(t) \quad \left. \begin{array}{l} 0 \leq t \leq T \\ i = 1, \dots, M \end{array} \right\}$$

Recibida la señal se realizan dos pasos, primero la forma de onda recibida,  $\mathbf{r}(\mathbf{t})$ , se reduce a una variable aleatoria simple  $\mathbf{z}(\mathbf{T})$  a la salida del demodulador en  $\mathbf{t} = \mathbf{T}$ , donde  $\mathbf{T}$

es la duración del símbolo. En el segundo paso, se compara  $z(T)$  con un determinado umbral.

El paso uno se conoce como **punto de predetección** y es al que se encuentran referidos los valores de potencia de señal recibida o ruido de interferencia recibido ( $E_b/N_0$ ). En cada tiempo de símbolo, la señal disponible en el punto de predetección es una muestra de pulso bandabase, todavía no se tienen bits.

El paso dos entonces puede pensarse como el que determina en que región de decisión se encuentra el punto actual. Como se sabe, un filtro casado es aquel que provee la máxima SNR a la salida del filtro en  $t = T$ , con ello se puede definir un receptor de correlación comprendido de  $M$  correladores, que transforma una señal  $r(t)$  en una secuencia de  $M$  números de salidas de correlator  $\mathbf{z}_i(T)$ .

Cada salida de correlator es de la forma

$$z_i(T) = \int_0^T r(t) s_i(t) dt \quad i= 1, \dots, M$$

Que indica la correlación de la señal recibida con los prototipos de señal conocidos por el receptor. Así, la decisión se realizará por la forma de onda que tenga la mayor correlación con  $\mathbf{r}(t)$ , es decir, la  $\mathbf{s}_i(t)$  sería la máxima  $\mathbf{z}_i(T)$ .

Ahora, el banco de  $M$  correladores se puede sustituir por un banco de  $N$  correladores,  $N=M$ , con un conjunto de **funciones base** que forman señales de referencia. Para el caso de detección binaria, se puede configurar el receptor de correlación para que funcione como un filtro casado simple o un integrador de producto y su salida alimentar directamente a la fase de decisión.

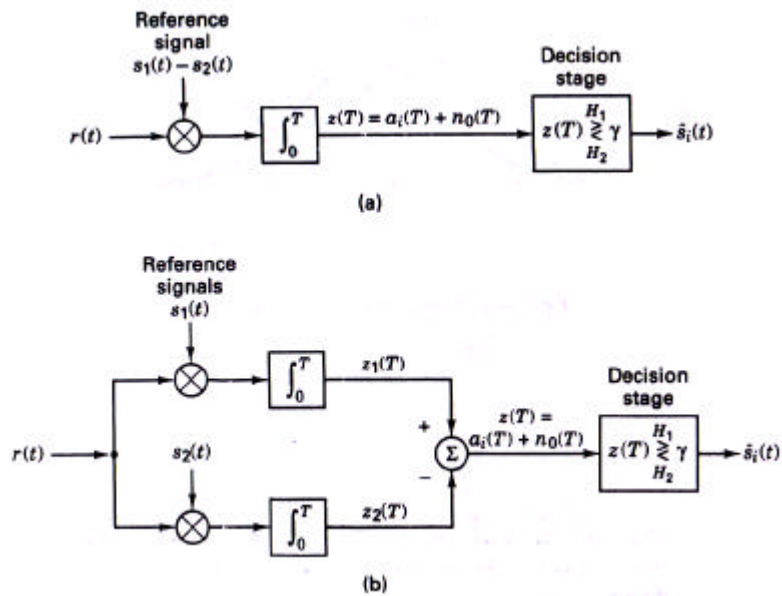


Figura 3.2. Receptor de correlación Binario. a) Un Correlator, b) Dos Correlatores

3.2.1.3. Detección Coherente

3.2.1.3.1. Detección Coherente de MPSK.

Las formas de onda empleadas por un sistema MPSK coherente típico, se pueden expresar como:

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos\left(\omega_0 t - \frac{2\pi i}{M}\right) \quad \begin{cases} 0 \leq t \leq T \\ i = 1, \dots, M \end{cases}$$

E es la energía recibida por la forma de onda en cada T duración de símbolo y  $\omega_0$  es la frecuencia de portadora. Si se asume un espacio ortonormal para las señales, se podrían determinar sus ejes como

$$\Psi_1(t) = \sqrt{\frac{2}{T}} \cos \omega_0 t$$

$$\Psi_2(t) = \sqrt{\frac{2}{T}} \sin \omega_0 t$$

La amplitud  $\sqrt{2/T}$  se ha escogido para normalizar la salida esperada del detector, de forma que la forma de onda en este nuevo sistema, es de la forma

$$s_i(t) = a_{i1}(t)\Psi_1(t) + a_{i2}(t)\Psi_2(t)$$

$$s_i(t) = \sqrt{E} \cos\left(\frac{2pi}{M}\right)\Psi_1(t) + \sqrt{E} \sin\left(\frac{2pi}{M}\right)\Psi_2(t)$$

Con esto se describen un conjunto de M múltiples formas de onda, no ortogonales, en términos de dos componentes de portadora ortogonales. La regla de decisión para saber que forma de onda es la recibida, es escoger la i-ésima forma si  $\hat{z}(T)$  es la mayor de las salidas del correlator.

De lo revisado anteriormente se espera que existan M correlatores para M formas de onda, cada uno con un cambio de fase determinado, pero en la práctica se usa la consideración de usar N integradores de producto, que es menor a M. esto se puede realizar debido a que toda forma de onda integrable puede ser expresada como combinación lineal de formas de onda ortogonales. Con ello la señal recibida  $r(t)$  es de la forma

$$r(t) = \sqrt{\frac{2E}{T}} (\cos f_i \cos \omega_0 t + \sin f_i \sin \omega_0 t) + n(t) \quad \begin{cases} 0 \leq t \leq T \\ i = 1, \dots, M \end{cases}$$

Donde  $f_i = 2pi/M$ . Como podemos ver en la figura 3.3, solo hay dos correlatores con las funciones base  $\psi_1(t)$  (superior) y  $\psi_2(t)$  (inferior), los cuales producirán las componentes en fase (X) y en cuadratura (Y) del vector  $r$ , que determinan  $\hat{f}$  como el arctan de Y/X como una estimativa de la fase de la señal recibida, la cual se compara con cada prototipo  $f_i$  en el demodulador y se escoge la que tenga la menor diferencia.

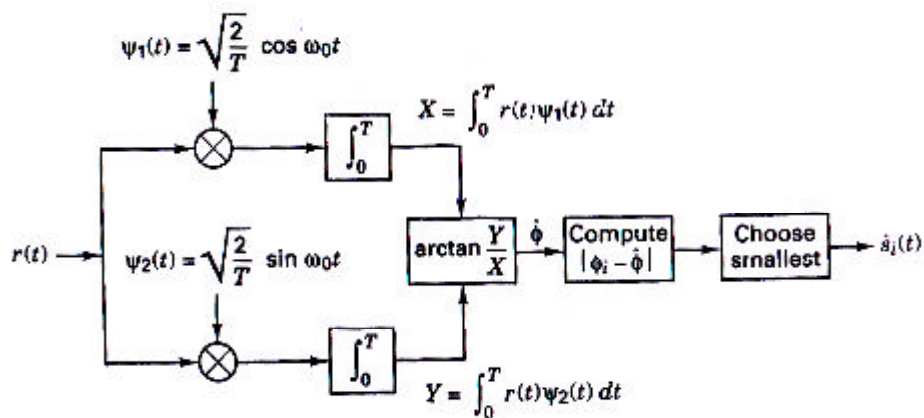


Figura 3.3. Demodulador de Señales MPSK



### 3.2.1.3.2. Detección Coherente de FSK.

Dado que la modulación FSK se caracteriza por llevar la información contenida en la frecuencia de la portadora, su ecuación típica es

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos(\omega_i t + f) \quad \begin{cases} 0 \leq t \leq T \\ i = 1, \dots, M \end{cases}$$

En la que E es la energía de  $s_i(t)$  en cada duración de símbolo T, y  $(\omega_{i+1} - \omega_i)$  se asume como un múltiplo entero de  $\pi/T$  y dado que la fase f es una constante arbitraria, se puede asumir igual a cero.

Si al igual que en caso anterior se consideran las funciones base  $\psi_i(t)$  formando un conjunto ortonormal, se tiene

$$\psi_j(t) = \sqrt{\frac{2}{T}} \cos \omega_j t$$

Y

$$a_{ij} = \int_0^T \sqrt{\frac{2E}{T}} \cos(\omega_i t) \sqrt{\frac{2E}{T}} \cos(\omega_j t) dt$$

En donde  $\sqrt{2/T}$  normaliza la salida esperada del filtro casado. Por lo tanto

$$a_{ij} = \begin{cases} \sqrt{E} & \text{si } i = j \\ 0 & \text{en otro caso} \end{cases}$$

Es decir, el  $i$ ésimo prototipo de vector se localiza en la  $i$ ésima coordenada a una distancia  $\sqrt{E}$  del origen de coordenadas. Para este esquema, y para un caso M-ario dado E, la distancia entre dos vectores prototipo es constante

$$d(s_i, s_j) = \|s_i - s_j\| = \sqrt{2E} \quad \text{para } i \neq j$$

En la figura 3.4 se muestra como se hace la división de las regiones de decisión, y aunque  $M$  es potencia de dos, se asume una  $M=3$  para mostrar la ortogonalidad de los vectores y que ahora son planos en lugar de líneas los límites de dichas regiones.

La regla de decisión establece a que región pertenece la señal transmitida de acuerdo con el índice de la región cuyo índice corresponde al que se encontró la señal.

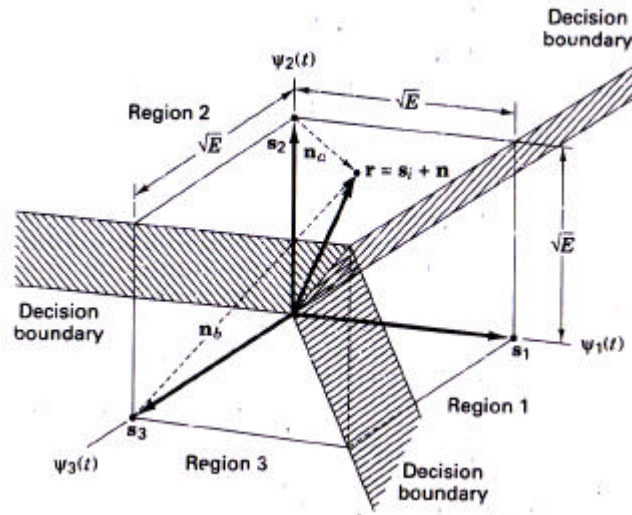


Figura 3.4. Regiones de Decisión para MFSK,  $M=3$  (ideal)

### 3.2.1.4. Detección No Coherente

#### 3.2.1.4.1. Detección de PSK Diferencial – DPSK.

A pesar de que este tipo de modulación puede ser en ciertos casos detectado coherentemente, por ser modulación de “fase”, pertenece a los no coherentes porque no requiere una referencia en fase con la señal recibida.

En los sistemas no coherentes no existe tentativas por determinar el valor actual de la fase de la señal entrante, por lo tanto, si la señal transmitida es de la forma

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos[\omega_0 t + q_i(t)]$$

La señal recibida se puede caracterizar de la forma

$$r(t) = \sqrt{\frac{2E}{T}} \cos[\omega_0 t + \mathbf{q}_i(t) + \mathbf{a}] \quad \begin{cases} 0 \leq t \leq T \\ i = 1, \dots, M \end{cases}$$

En la que  $\mathbf{a}$  es una constante arbitraria y asume valores aleatorios entre cero y dos pi.

Para detección coherente se usan filtros casados o sus equivalentes, pero para detección no coherente no es posible ya que la salida de este tipo de filtros es función de un ángulo desconocido  $\mathbf{a}$ .

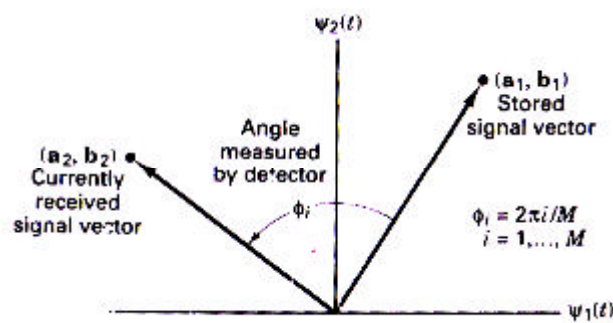


Figura 3.5. Espacio Ortogonal para DPSK

La base de la detección diferencialmente coherente (DPSK) es la fase de portadora del intervalo anterior de señalización que puede ser empleada como una fase de referencia para la demodulación, es decir, que la codificación debe ser diferencial en el transmisor y que la información es llevada por la diferencia en fase entre dos ondas consecutivas.

El detector calcula las coordenadas de la señal que llega correlacionándola con formas de onda generadas localmente como  $\sqrt{2/T} \cos \omega_0 t$  y  $\sqrt{2/T} \sin \omega_0 t$ , entonces mide el ángulo entre el vector de la señal recibida y el vector de la señal anterior como se ve en la figura 3.5.

La señalización DPSK en general es menos eficiente que PSK debido a la tendencia a la propagación de errores resultante de la correlación entre las diferentes ondas de señalización. Además se puede considerar que tiene dos veces más ruido, lo que a fin de cuentas significarían 3 dB por debajo del PSK.

### 3.2.1.4.2. Detección de FSK.

Se pueden usar los correladores del receptor de correlación para la detección de FSK no coherente, pero se requieren configurar como detectores de energía sin usar mediciones de fase. Por esta razón, el detector no coherente requiere el doble de canales de un coherente como se ve en la figura 3.6.

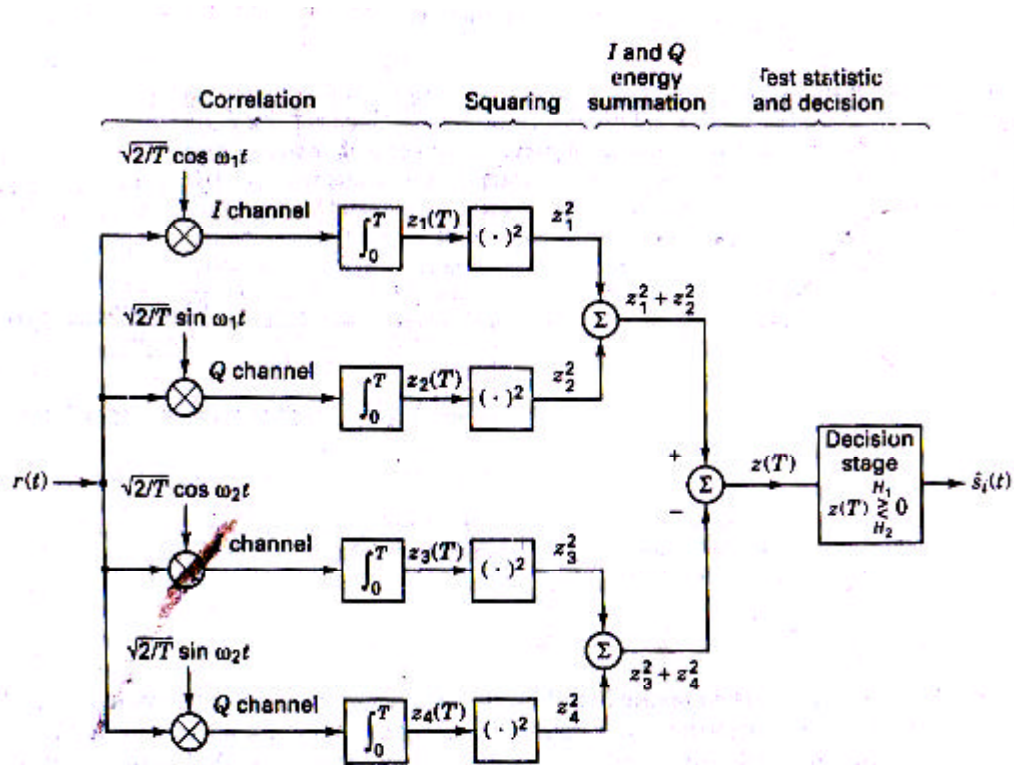


Figura 3.6. Detector No Coherente FSK en Cuadratura

Existen canales en fase (I) y en cuadratura (Q) para detectar esta BFSK no coherentemente. Como podemos ver los dos bloques superiores están configurados para una frecuencia  $\omega_1$  y los dos inferiores para una  $\omega_2$ . Los canales en fase son componentes  $\sqrt{2/T} \cos \omega_i t$  y los canales en cuadratura  $\sqrt{2/T} \sin \omega_i t$ . Dado que la señal recibida  $r(t)$  es de la forma  $\cos(\omega_i t + f) + n(t)$ , la señal recibida se correlacionará parcialmente con una componente en fase y con una en cuadratura, **de una misma frecuencia**, luego son elevadas al cuadrado para así evitar valores negativos, se añaden los resultados y en la fase

final se escoge la señal con la frecuencia a la cual pertenecen los detectores  $q$  entregaron la mayor salida.

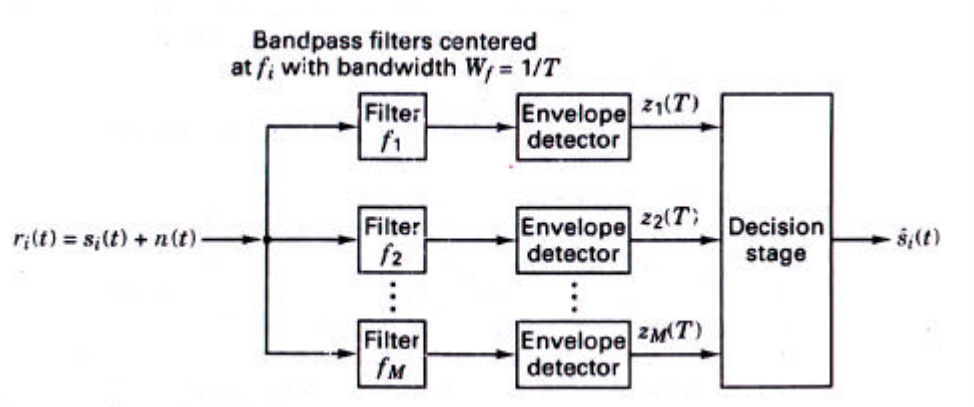


Figura 3.7. Detector No Coherente de FSK

Hay otra forma de implementar la detección no coherente para FSK, la cual consiste en usar filtros pasabandas seguidos de detectores de envoltura como se observa en la figura 3.7, los mismos que se forman de un rectificador y un filtro pasabajos. Los detectores están ajustados a la envoltura de las señales y la fase de la portadora no es importante para definir la envoltura, por lo tanto no se usa información de fase.

La decisión sobre la señal recibida entonces se realiza sobre la base de cual de los detectores tiene la mayor salida al momento de la medición. Aunque esto parezca funcionalmente más simple que un demodulador en cuadratura, es realmente más complejo y más costoso que aquel, y además el de cuadratura es digitalmente implementable mientras que este nuevo modelo usa filtros análogos.

### 3.2.2. MODULADORES / DEMODULADORES: MPSK, MFSK, MQAM

La  $M$ , que acompaña a las siglas que representan los distintos tipos de modulación digital, es un dígito que indica el número de condiciones posibles, por ejemplo, para en sistemas Binarios  $M = 2$ . Para la modulación digital generalmente es ventajoso codificar con niveles mayores, es decir, usar más de un bit por cada símbolo a transmitir. Esto se puede expresar con

$$N = \log_2 M$$

Donde  $N$  es el número de bits y  $M$  el número de condiciones de salida posibles con  $N$  bits.

### 3.2.2.1. Límite de Shannon para Capacidad de Información.

La capacidad de información de un sistema de comunicación representa el número de símbolos independientes que pueden pasarse, a través del sistema, en una unidad de tiempo determinada. Debido a que el símbolo fundamental es un bit binario, es conveniente expresar esta capacidad en **bits por segundo (bps)**.

La capacidad de información es una función lineal del ancho de banda y de la línea de transmisión, y es directamente proporcional a ellos. Es decir, que un cambio en alguno de ellos, cambiará de forma proporcional la capacidad de información.

E. Shannon trabajando para laboratorios Bell, estableció matemáticamente la capacidad de información de un canal relacionando el ancho de banda y la relación señal a ruido, de la siguiente manera:

$$I = B \log_2 \left( 1 + \frac{S}{N} \right)$$

$$I = 3,32 \log_{10} \left( 1 + \frac{S}{N} \right)$$

Donde  $I$ , es la capacidad de información en bps  $B$ , el ancho de banda en Hz y  $S/N$  la relación señal-ruido sin unidades.

Así pues, un canal de voz estándar con un SNR = 30 dB y un ancho de banda de 2,7 KHz, tiene un límite de capacidad de información de 26900 bps. Esto es posible pero no con un sistema de transmisión binario sino con un M-ario como los que se analizan adelante en este capítulo.

La probabilidad típica del error versus  $E_b/N_0$  tiene la forma de una cascada al igual que las características de probabilidad de error de bit de varios esquemas de modulación

binaria en AWGN. En la figura 3.8 se muestra las características ideales como el límite de Shannon, este límite representa el umbral de  $E_b/N_0$  bajo el cual no puede ser mantenida una comunicación fiable.

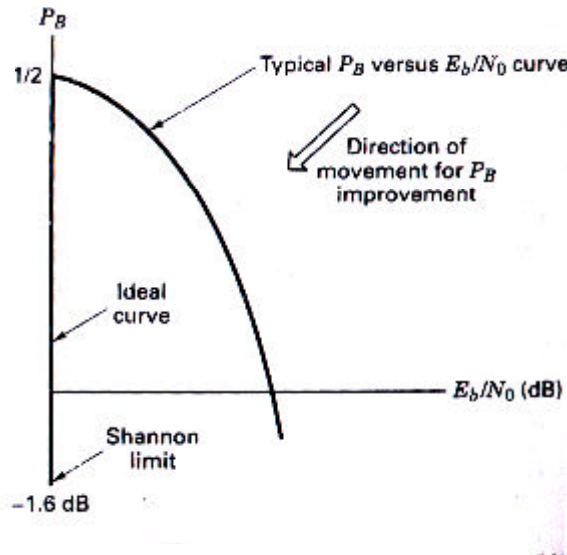


Figura 3.8. Curva Ideal PB versus Eb/No

Para todos los valores de  $E_b/N_0$  sobre el límite de Shannon de  $-1.6$  dB,  $P_B$  es cero. Una vez que  $E_b/N_0$  es reducida bajo el límite de Shannon,  $P_B$  cae hacia el peor caso de valor  $1/2$ . La flecha grande en la figura 3.8 describe la dirección de movimiento deseada para lograr un mejoramiento del rendimiento de  $P_B$ .

### 3.2.2.2. Desplazamiento en Frecuencia FSK (Frequency Shift Keying)

Caracterizada por llevar la información en la frecuencia de la portadora, es una forma de modulación de bajo rendimiento. Su amplitud es constante y la fase entre cada uno de las formas de onda que representan los diferentes símbolos no es necesariamente continua. Es similar a la modulación en frecuencia convencional FM, con la diferencia de que la señal modulante ahora es de tipo discreto. Su expresión analítica general es:

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos(\omega_i t + f) \quad \begin{cases} 0 \leq t \leq T \\ i = 1, \dots, M \end{cases}$$

Las formas de onda para FSK se caracterizan por ser ortogonales, es decir, cada una de los formas de onda está representada por un eje mutuamente perpendicular a los otros con una senoide de diferente frecuencia. Esto depende del cumplimiento del requerimiento de espaciamento entre los diferentes tonos. Es decir, para que el grupo de señales sean ortogonales, cualquier par de tonos adyacentes deben tener una separación de frecuencia de un múltiplo de  $1/T$  hertz, que es la diferencia de frecuencia entre el centro del lóbulo principal y el primer cruce por cero.

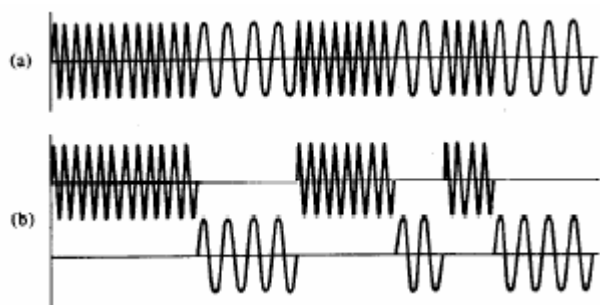


Figura 3.9. Onda FSK, a) Ideal, b) Descompuesta.

Para que dos tonos no se interfieran entre ellos durante la detección, el pico del espectro del tono uno debe coincidir con uno de los cruces por cero del espectro del tono dos y similarmente, el pico del espectro del tono dos debe coincidir con uno de los cruces por cero del espectro del tono uno.

#### 3.2.2.2.1. FSK Binario.

La salida de un modulador binario es una función escalón en el dominio del tiempo; su salida se desplaza entre dos frecuencias, una de marca o de 1 lógico y una de espacio o de 0 lógico. Su salida cambia cada vez que la entrada cambia. La rapidez de cambio en la entrada se llama **razón de bit** y se mide en bps. La rapidez de cambio en la salida se llama **baudio** o **razón de baudios** y es igual al recíproco del tiempo de un elemento de señalización de salida, es decir, símbolos por segundo.



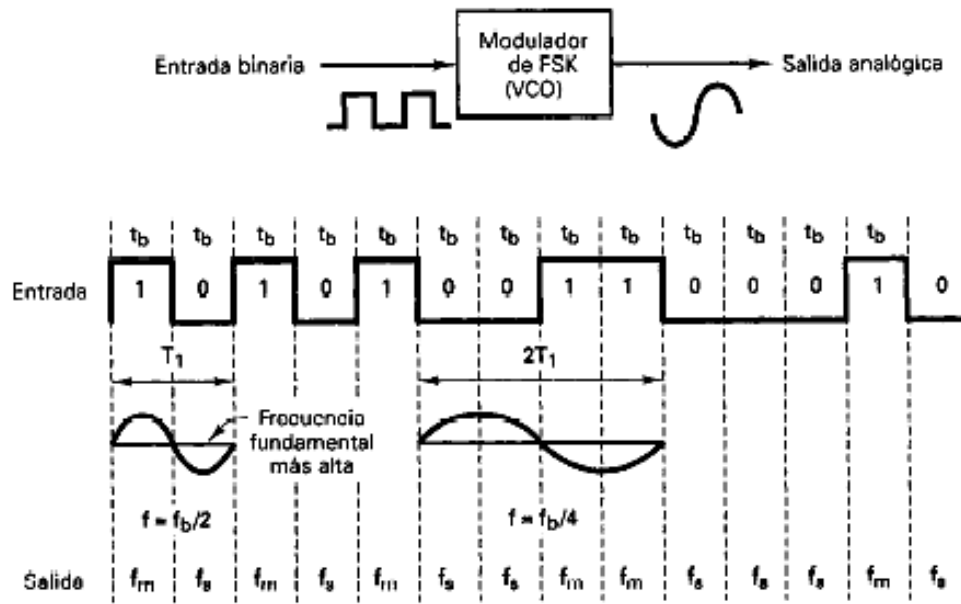


Figura 3.10. Modulador de FSK

Debido a que el FSK binario es una forma de modulación en frecuencia, el índice de modulación utilizado en FM es valido y está dado por

$$MI = \frac{\Delta f}{f_a}$$

$$MI = \frac{\left| \frac{f_m - f_s}{2} \right|}{\frac{f_b}{2}} = \frac{|f_m - f_s|}{f_b}$$

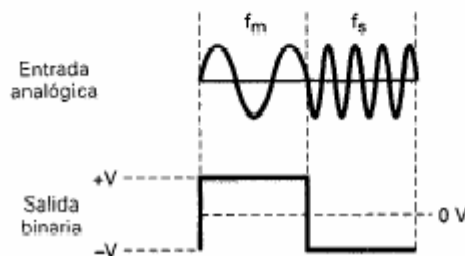
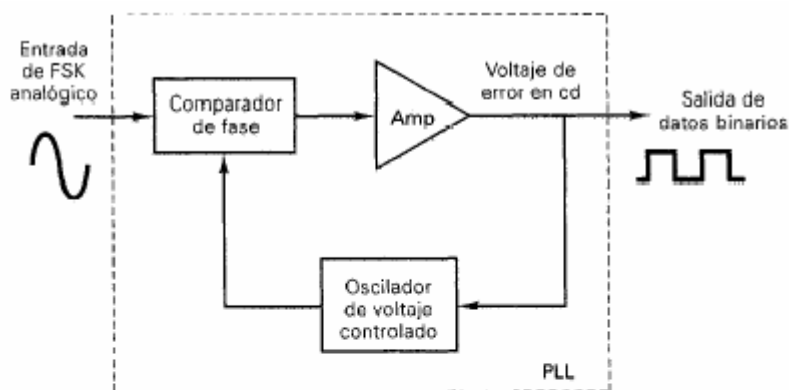
El pero caso del índice de modulación es el que da el ancho de banda de salida las amplio, llamado **relación de desviación**, y sucede cuando la desviación de frecuencia  $|f_m - f_s|/2$  y la frecuencia modulante o fundamental,  $f_a = f_b / 2$ , están en sus valores máximos.

Por lo general MI se mantiene bajo 1, produciendo un espectro de salida relativamente angosto y para su graficación se utiliza la función de Bessel, cuyos valores se encuentran en la tabla 3.1, y que corresponden a las amplitudes de la portadora y de hacia cada lado. La separación entre componentes es igual a en  $f_a$  Hz.

$Ml$	$J_0$	$J_1$	$J_2$	$J_3$	$J_4$
<b>0.0</b>	1.00				
<b>0.25</b>	0.98	0.12			
<b>0.5</b>	0.94	0.24	0.03		
<b>1.0</b>	0.77	0.44	0.11	0.02	
<b>1.58</b>	0.51	0.56	0.23	0.06	0.01
<b>2.0</b>	0.22	0.58	0.35	0.13	0.03

**Tabla. 3.1. Función de Bessel**

Es circuito de fase cerrada PLL es el que más se utiliza para demodular FSK binario. Conforme cambia la entrada del PLL entre las frecuencias de marca y espacio, el error de voltaje a la salida del comparador de fase sigue el desplazamiento de frecuencia. Debido a que solo hay dos frecuencias de entrada, hay dos voltajes de error, que son 1 o 0 lógicos.



**Figura 3.11. Demodulador FSK con PLL**

### 3.2.2.2. FSK M – ario.

La señalización M-aria ortogonal FSK puede ser de gran utilidad en canales de potencia limitada para conservar la potencia transmitida a costa mayor ancho de banda y mayor complejidad en el equipo.

En forma general un demodulador M-ario de cualquier esquema tiene la forma del receptor óptimo básico, solo que debe tenerse en cuenta las señales base ortogonales para tomarlas como referencia tal como se muestra en la figura.

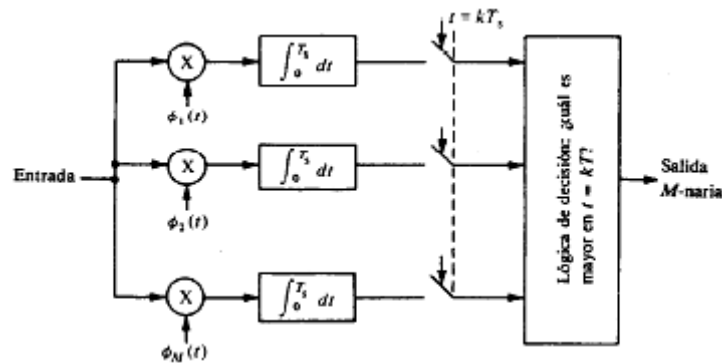


Figura 3.12. Receptor Óptimo para M-aria Ortogonal

Se evalúa la energía en los tiempos  $t = kT$  y se determina cual de las salidas del filtro acoplado (correlator) es mayor y ese símbolo se toma como el correcto para un intervalo de símbolos. En la estructura del receptor, podemos implementar la etapa de detección haciéndola de forma coherente y no coherente. En el caso práctico se utilizan más las estructuras de detección no coherente ya sea a través de correladores o de combinaciones de filtros pasabanda y detectores de envolvente.

La separación mínima entre las frecuencias de las señales  $s_i(t)$  está determinada por el tipo de demodulador utilizado en la recepción. Si este es del tipo no coherente entonces  $\Delta f_{\min} = 1/T$  y si es del tipo coherente entonces la separación  $\Delta f_{\min} = 1/2T$ , por lo que el ancho de banda utilizado será:

$$B_{\min} = \Delta f_{\min} (M - 1)$$

$$T_s = T_b \log_2 M$$

### 3.2.2.3. Desplazamiento en Fase PSK (Phase Shift Keying)

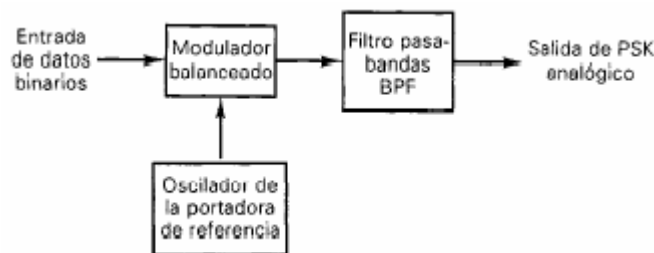
Es una forma de modulación digital de amplitud constante que lleva la información en la fase de la portadora. Su expresión analítica general es:

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos\left(w_0 t - \frac{2\pi i}{M}\right) \quad \begin{cases} 0 \leq t \leq T \\ i = 1, \dots, M \end{cases}$$

Las formas de onda utilizadas pueden representadas como vectores o fasores en un esquema polar, donde la longitud del vector corresponde a la amplitud de la señal, y el vector dirección, para cualquier caso  $M$ -ario, corresponde a la fase relativa de la señal hacia otra señal  $M-1$  del conjunto.

#### 3.2.2.3.1. BPSK – PSK Binario.

Modulación de forma de onda cuadrada de portadora suprimida que presenta dos posibles fases de salida, existiendo así  $180^\circ$  entre una y otra fase de portadora. Se le conoce también como **transmisión de fase inversa PRK**.



**Figura 3.13. Modulador BPSK**

En la figura 3.13, el modulador balanceado es el conmutador que se usa para invertir la fase. La razón de bit es igual a la razón de baudio, y dado que el mayor ancho de banda se da cuando la secuencia de datos de entrada es alternativa 1/0, (lo que significa que la frecuencia fundamental  $f_a$  sea  $f_b/2$ ), el mínimo ancho de banda de Nyquist de doble lado  $f_N$  es

$$f_N = 2 \left( \frac{f_b}{2} \right) = f_b$$

El espectro de salida de un modulador BPSK es una señal de doble banda lateral con portadora suprimida donde las frecuencias laterales están separadas de la portadora por el valor de  $f_a$  en Hz.

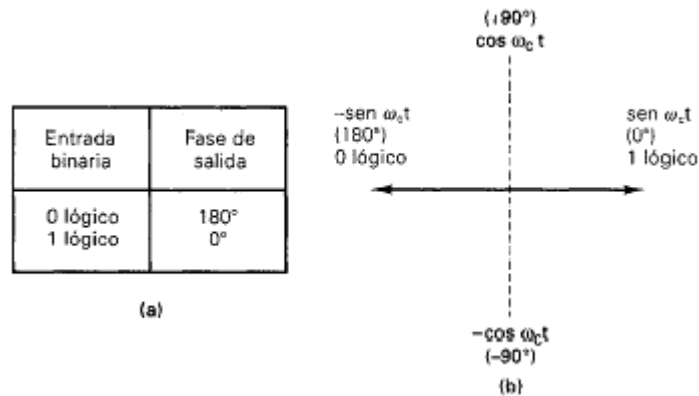


Figura .3.14. Tabla de Verdad y Diagrama Fasorial de BPSK

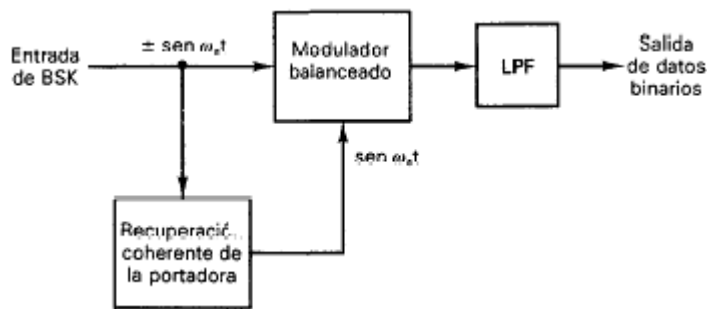


Figura .3.15. Demodulador BPSK

En la figura 3.15 se muestra el demodulador BPSK, en el que hay un recuperador de portadora que detecta y regenera una señal de portadora coherente en frecuencia y en fase con la original del transmisor. Luego de que la señal pase por el detector de producto (modulador balanceado), llega al filtro pasabajos en donde se separa los datos binarios recuperados de la señal demodulada compleja. Matemáticamente, con una señal de entrada  $+ \sin(\omega_c t)(1 \text{ lógico})$  la salida del modulador balanceado es:

$$(\sin \omega_c t)(\sin \omega_c t) = \sin^2(\omega_c t)$$

$$-\sin^2(\omega_c t) = \frac{1}{2}(1 - \cos 2\omega_c t) = \frac{1}{2} + \frac{1}{2} \cos 2\omega_c t$$

$$\text{salida} = +\frac{1}{2} V = 1 \text{ lógico}$$

La función del LPF es bloquear la segunda armónica de la portadora, que presenta doble frecuencia.

### 3.2.2.3.2. QPSK – PSK Cuaternario.

Modulación angular de amplitud constante de  $M = 4$ , es decir, cuatro posibles fases de salida diferentes para una sola frecuencia portadora. Existen cuatro condiciones de entrada diferentes que se logran con dos bits, **dibit**. Con esto, por cada dibit entrado al modulador existe un cambio en la salida, la razón de baudio es la mitad de la razón de bit.

En la figura 3.16 se puede apreciar como un dibit entra en forma serial y se van en forma paralela, uno hacia el canal I (modula en fase) y el otro al canal Q (modula en cuadratura).

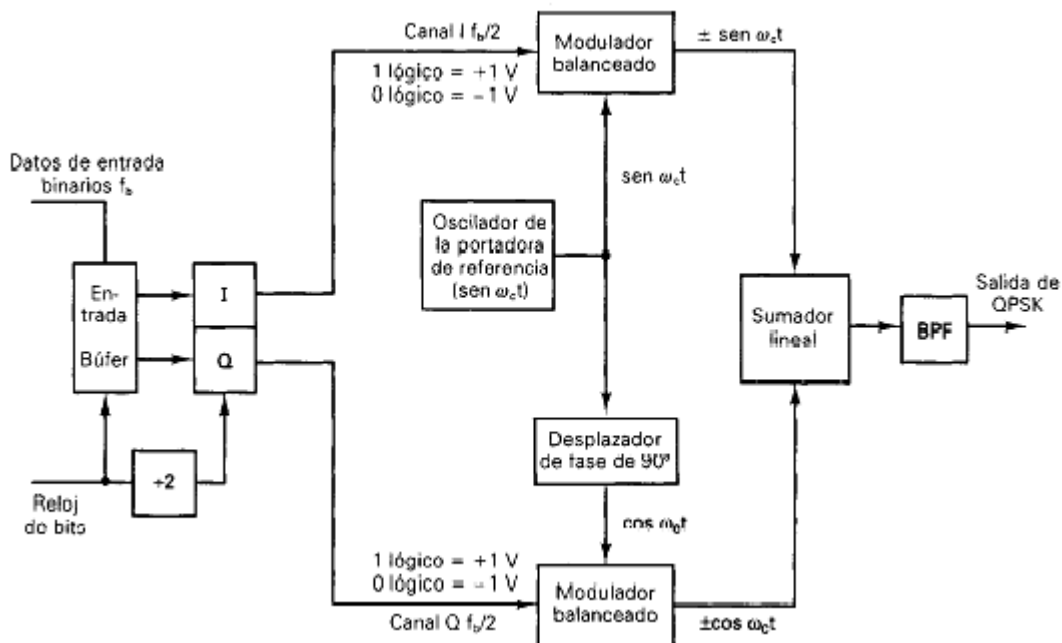


Figura 3.16. Modulador QPSK

Este modulador es equivalente a dos BPSK en paralelo y, en efecto el sumador lineal, luego de los moduladores balanceados, entrega cuatro fases resultantes.

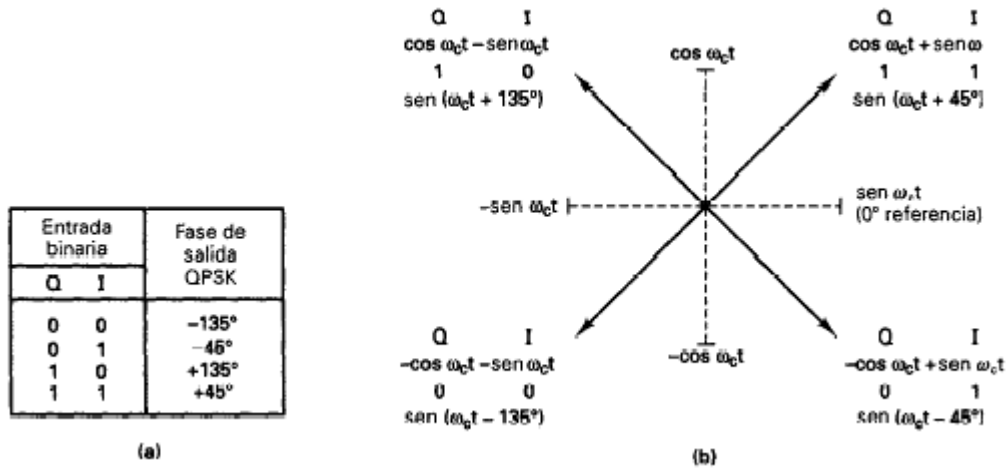


Figura 3.17. Tabla de Verdad (a) y Diagrama fasorial (b) de QPSK

Observando la figura 3.17 se puede apreciar que cada una de las posibles salidas tiene exactamente la misma amplitud, lo cual es la característica más importante que la diferencia de QAM; la separación entre dos fasores es de 90°, así, una señal de este tipo puede variar ± 45° durante la transmisión y mantener la información correcta al demodularse.

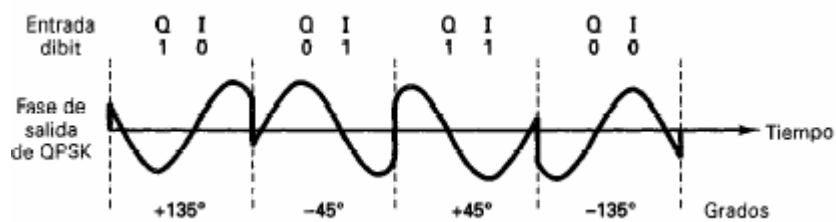


Figura 3.18. Posibles Señales de Salida QPSK

Al dividirse los datos de entrada en dos canales, la tasa de bits en cada uno de ellos es la mitad de la tasa de entrada  $f_b/2$ , así la frecuencia fundamental más alta presente en la entrada de datos al modulador balanceado es la mitad de lo que entra en cada canal  $f_a = f_b/4$ , es decir, un cuarto de la tasa de bits de entrada.

Por tanto, el mínimo ancho de banda de Nyquist de doble lado requerido es

$$f_N = 2 \frac{f_b}{4} = \frac{f_b}{2}$$

Existiendo una compresión de ancho de banda, el cual es igual a la tasa de baudios en Hz.

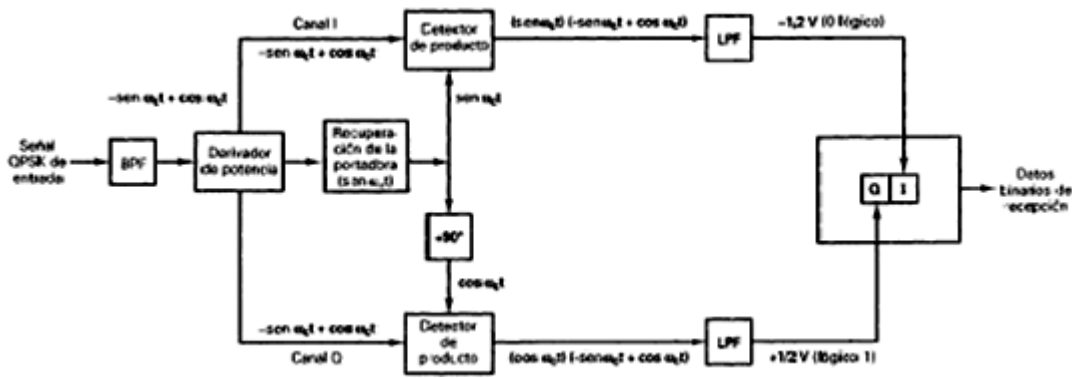


Figura 3.19. Demodulador QPSK

El proceso de demodulación es similar al visto para BPSK, siendo el derivador de potencia el encargado de dirigir la señal QPSK a los detectores de producto I y Q y seguir el proceso revisado para BPSK.

### 3.2.2.3.3. OQPSK – QPSK de Compensación (Offset).

Es una modificación del QPSK en el cual las formas de onda de los bits en los canales I y Q se compensan o se cambian en fase, entre sí, por la mitad del tiempo de bit.

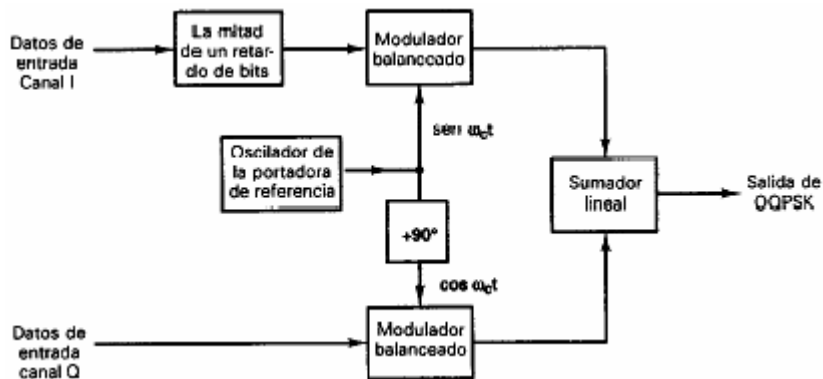


Figura 3.20. Modulador OQPSK



En la figura 3.21 se puede observar la alineación de la secuencia de bits, debido a que ocurren cambios en el canal I en los puntos medios de los bits del canal Q, y viceversa, nunca hay más de un solo cambio de bit en el dibit, lo que significa que nunca hay un más de un cambio de  $90^\circ$  en la fase de salida.

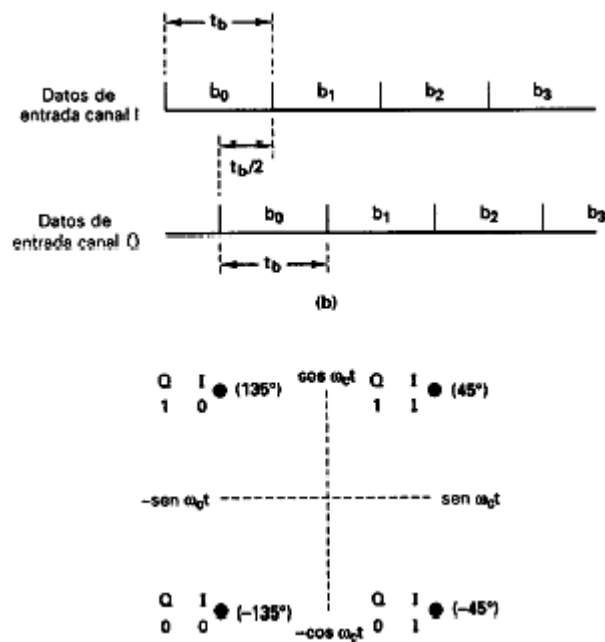


Figura 3.21. Alineación de Bits (b) y Constelación (c) de OQPSK

Así, una ventaja de OQPSK es el cambio de fase limitado durante la modulación y una desventaja es que, los cambios en la fase de salida ocurren al doble de la tasa de datos en los canales I y Q. como resultado de esto último, el ancho de banda y el baudio mínimo son el doble del QPSK convencional.

#### 3.2.2.3.4. PSK de Ocho Fases.

Posee ocho posibles fases de salida,  $M = 8$ , por lo que requiere que los bits entren en grupos de tres, **tribits**, por símbolo.

Los bits entran serialmente y se distribuyen a los canales I, Q y C (control), con lo que la tasa de bit por canal es  $f_b/3$ . Como se muestra en la figura 3.17, existen convertidores de

nivel 2 a 4, que son **convertidores digital a analógico** de entrada paralela. En los DAC, el bit I o Q determina la polaridad de la señal analógica de salida, mientras C o  $\bar{C}$  determinan la magnitud de las mismas. Así, con dos magnitudes y dos polaridades son posibles cuatro posibilidades de salida diferentes.

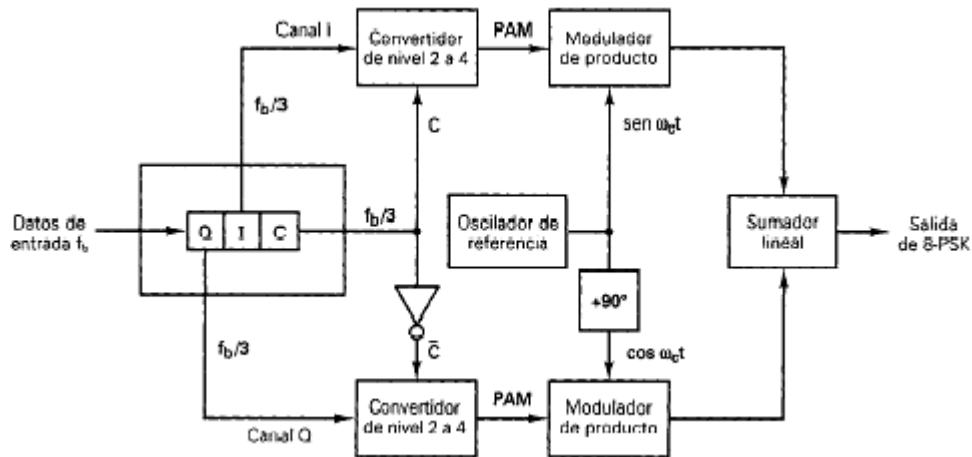


Figura 3.22. Modulador 8PSK

La salida de un convertidor de nivel 2 a 4, es una PAM M-aria, donde  $M = 4$ .

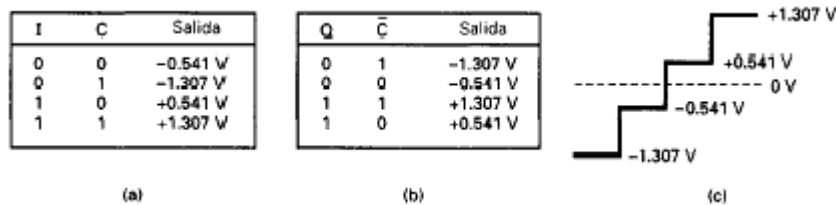


Figura 3.23. Convertidores de Nivel 2 a 4, a) Tabla de Verdad Canal I, b) Tabla de Verdad Canal Q, c) Niveles PAM

La separación angular entre dos favores adyacentes es de  $45^\circ$ , la mitad de QPSK y cada fador es de igual magnitud. Los valores de PAM son relativos, se puede usar cualquier valor mientras se mantenga la relación  $0.541 / 1.307$  y  $\arctan$  de  $22.5^\circ$ . Se utiliza **código Gray**, o **código de máxima distancia** para los favores, es decir, cada fador adyacente cambia solo por un bit. Así se disminuyen los errores, ya que si se recibe en error un fador adyacente, este cambia solo por un bit, y no todo el símbolo.

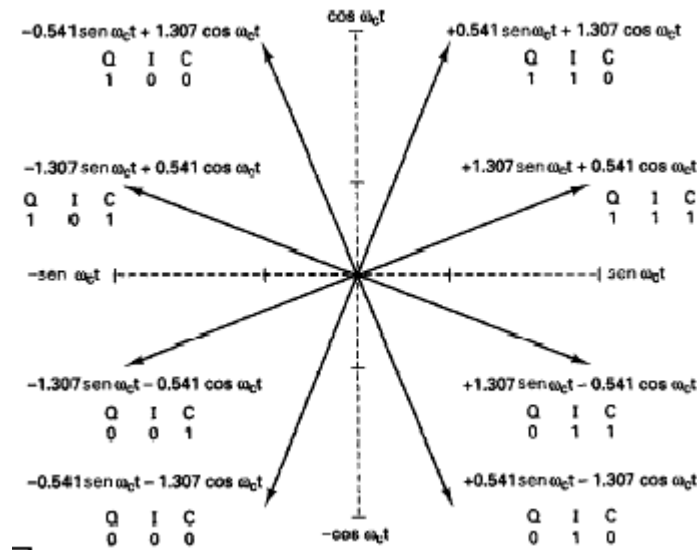


Figura 3.24. Constelación de 8PSK

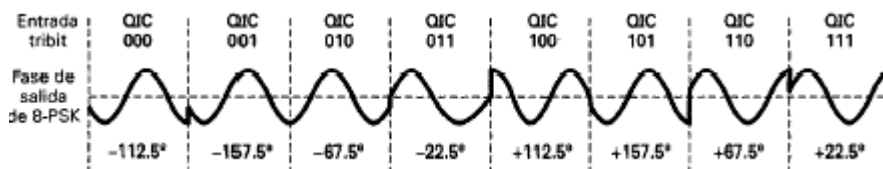


Figura 3.25. Posibles Señales de Salida de 8PSK.

Debido a que existe un cambio a la salida por cada tres en la entrada del modulador, la razón de baudio es un tercio de de la razón de bit,  $f_s = f_b/3$ , al igual que el ancho de banda mínimo. Con ello la frecuencia fundamental más alta es  $f_a = f_b/6$ .

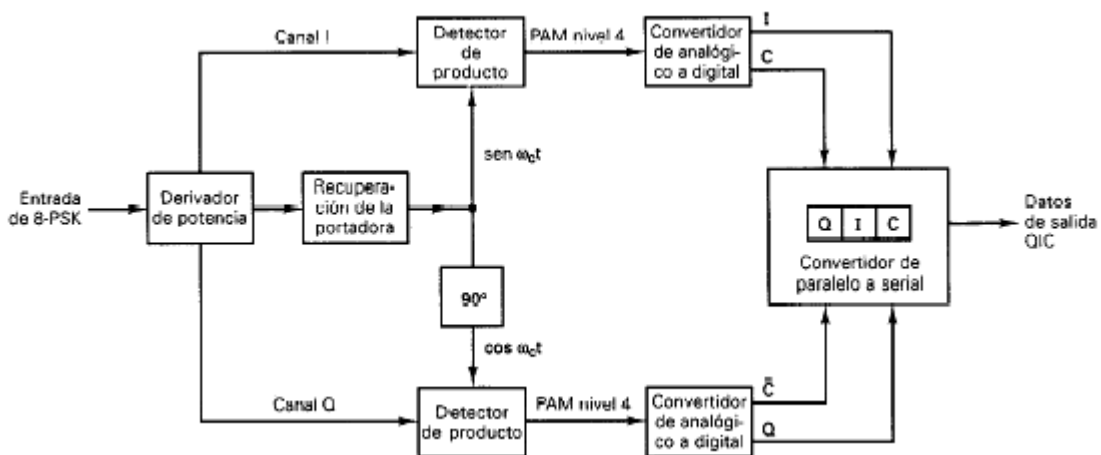


Figura 3.26. Demodulador 8PSK.

Para la demodulación, como se ve en la figura 3.26, las salidas de los detectores de producto son señales PAM de nivel 4 que alimentan convertidores analógico-digitales de nivel 4 a 2. Luego el circuito lógico de paralelo a serial convierte los pares  $I/C$  y  $Q/\bar{C}$  a flujos de salida  $I, Q, C$ .

### 3.2.2.3.5. PSK de Dieciséis Fases.

Tiene dieciséis posibles fases de salida,  $M = 16$ , en cuyo modulador entran los datos en grupos de cuatro bits, **quadbits**. Así su razón de baudio es la cuarta parte de su razón de bit  $f_b/4$ , al igual que su mínimo ancho de banda en Hz.

Código de bits	Fase	Código de bits	Fase
0000	11.25°	1000	191.25°
0001	33.75°	1001	213.75°
0010	56.25°	1010	236.25°
0011	78.75°	1011	258.75°
0100	101.25°	1100	281.25°
0101	123.75°	1101	303.75°
0110	146.25°	1110	326.25°
0111	168.75°	1111	348.75°

(a)

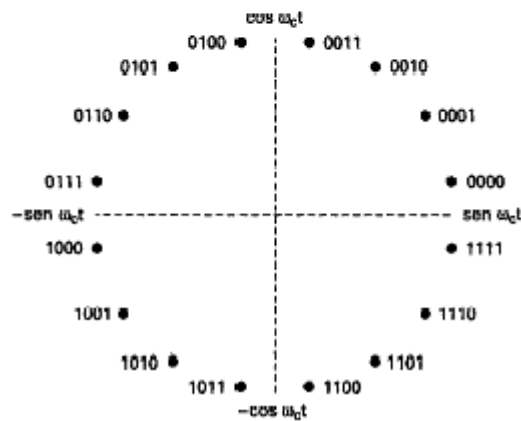


Figura 3.27. Tabla de Verdad y Constelación 16PSK

Como se puede apreciar en la figura 3.27, la separación angular de las fases es de 22.5°, debido a esto, es altamente susceptible al deterioro en la fase, por lo que rara vez se utiliza.

**3.2.2.3.6. BPSK Diferencial.**

Es una forma alterna de modulación en la que la información se encuentra en la diferencia entre dos elementos sucesivos de señalización, en lugar de la fase absoluta. No es necesario recuperar una portadora coherente en fase.

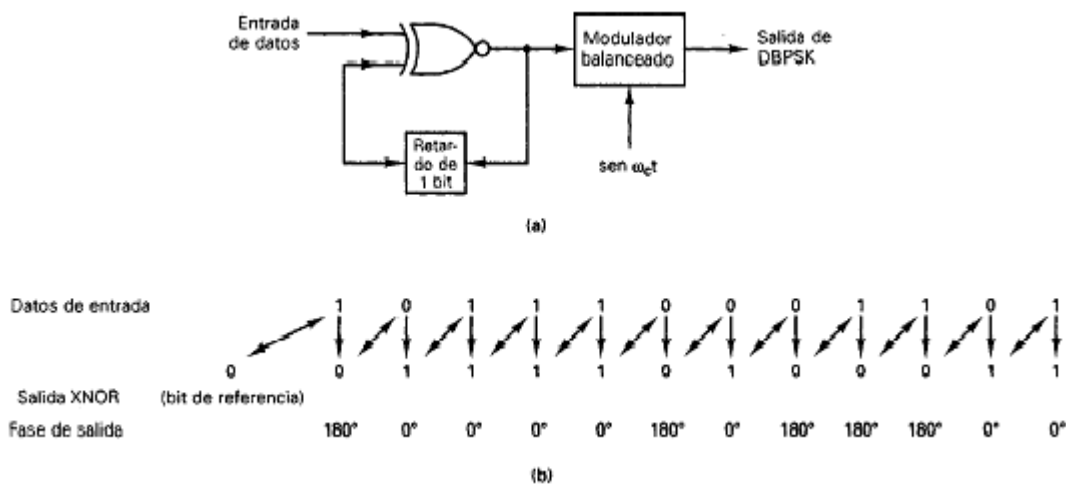


Figura 3.28. Modulador y relación de Señales DBPSK

Para la modulación, un bit de información entrante usará XNOR con el bit anterior antes de entrar al modulador balanceado. En la figura 3.28 se puede observar la relación entre los datos de entrada, los datos XNOR, y la fase de salida.

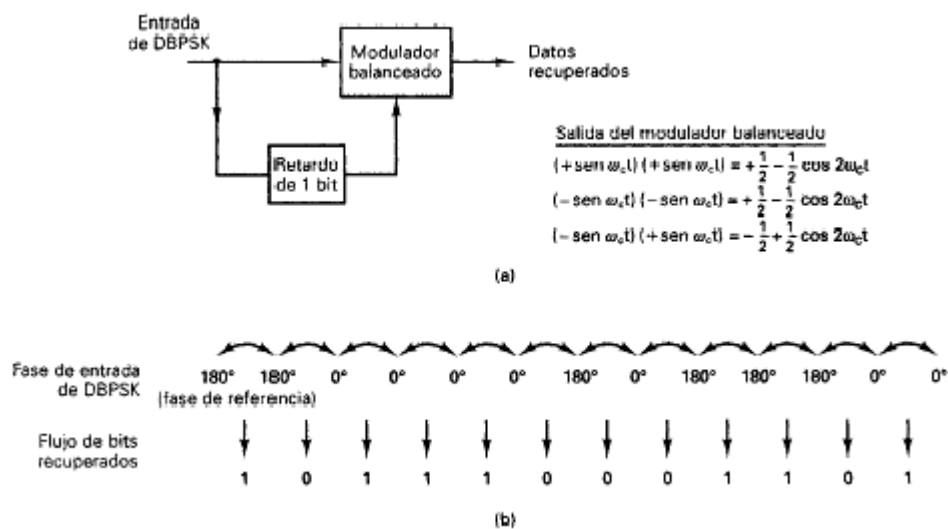


Figura 3.29. Demodulador y Secuencia de Tiempo DBPSK

Para demodular en cambio, la señal recibida se retrasa un tiempo de bit y luego se compara con el siguiente elemento. Si son iguales se genera un 1 lógico, si son diferentes un 0 lógico. Si se supone mal el bit de referencia, solo el primer bit demodulado estará en error.

Su ventaja es la simplicidad con la que se puede implantar y una desventaja, es que requiere entre 1 y 3 dB más de SNR para alcanzar la misma tasa de errores que el PSK absoluto.

#### 3.2.2.4. Desplazamiento en Amplitud ASK (Amplitude Shift Keying)

Este tipo de modulación lleva la información en la amplitud de la portadora y su expresión analítica general es

$$s_i(t) = \sqrt{\frac{2E_i(t)}{T}} \cos(\omega_0 t + f) \quad \begin{cases} 0 \leq t \leq T \\ i = 1, \dots, M \end{cases}$$

Donde la amplitud  $\sqrt{2E_i(t)/T}$  tiene M valores discretos y la fase f es una constante arbitraria.

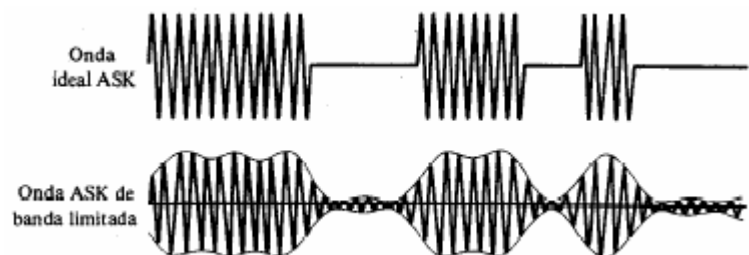


Figura 3.30. Onda Binaria ASK

A la forma binaria de este tipo de modulación se la conoce generalmente como OOK (On Off Keying) y fue una de las primeras modulaciones digitales utilizadas en radio telegrafía, aunque actualmente ya no es muy usada en los sistemas de comunicaciones digitales.

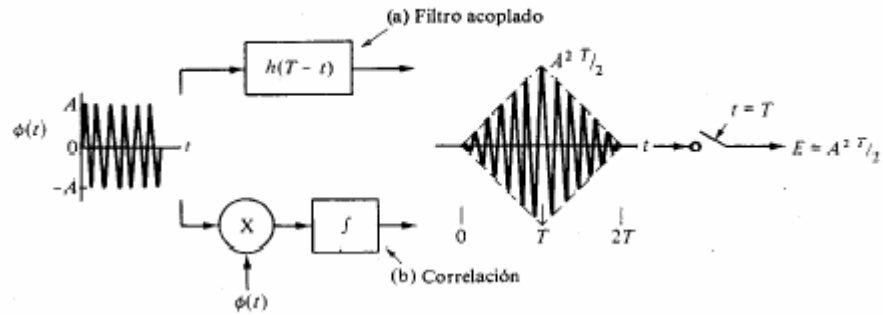


Figura 3.31. Detección de Filtro Apareado de ASK

Para un sistema binario, la tabla de verdad y sus diagramas serian:

Entrada Binaria	Amplitud de la Señal de Salida
0	0
1	$\sqrt{\frac{2E}{T}}$

DIAGRAMA FASORIAL

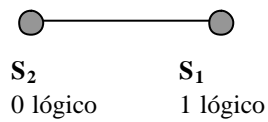


DIAGRAMA DE CONSTELACIÓN

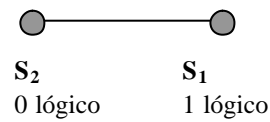


Figura 3.32. Tabla de Verdad y Diagramas ASK

3.2.2.5. Desplazamiento en Amplitud y Fase APK (Amplitude Phase Keying)

Resultado de la mezcla entre ASK y PSK, su expresión analítica general es:

$$s_i(t) = \sqrt{\frac{2E_i(t)}{T}} \cos[w_0 t + f_0(t)] \quad \begin{cases} 0 \leq t \leq T \\ i = 1, \dots, M \end{cases}$$

Donde constan la indexación de la amplitud y la fase.

Si sus  $M$  símbolos se arreglan en una constelación rectangular en el espacio bidimensional, la señalización pasa a ser una QAM como se muestra a continuación.

### 3.2.2.6. Modulación de Amplitud en Cuadratura QAM.

Como se dijo en APK, esta es una forma de modulación en donde la información digital se encuentra contenida en la amplitud y en la fase de la portadora transmitida.

#### 3.2.2.6.1. 8 – QAM.

Tiene 8 posibilidades de salida,  $M = 8$ , y a diferencia de PSK, la señal de salida del modulador no es de amplitud constante. Como puede verse en la figura 3.33, la diferencia con el modulador PSK es la omisión del inversor entre el canal C y el modulador de producto Q.

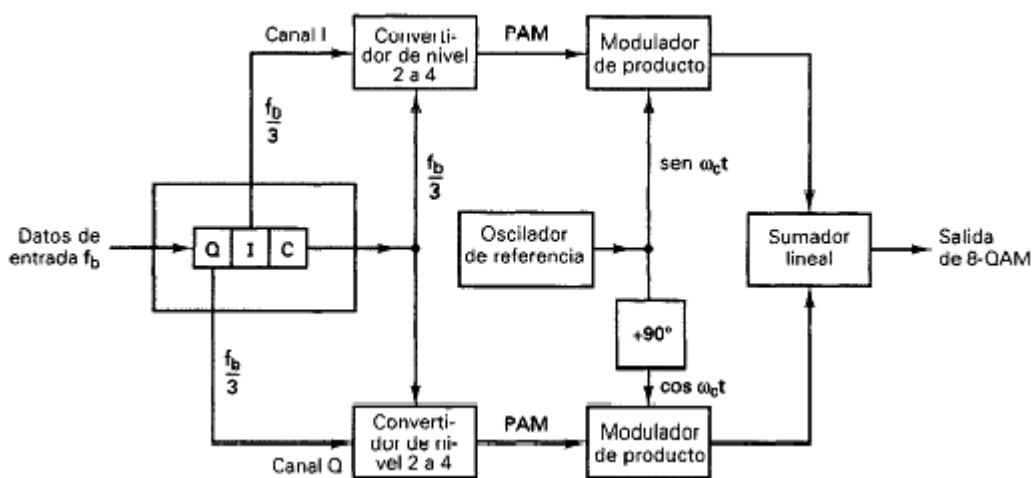


Figura 3.33. Modulador 8 QAM

Al igual que el 8 PSK, I y Q determinan la polaridad y C la magnitud, pero al no haber inversión de signos, la gráfica de fasores, su tabla de verdad y su constelación son los mostrados en la figura 3.34.



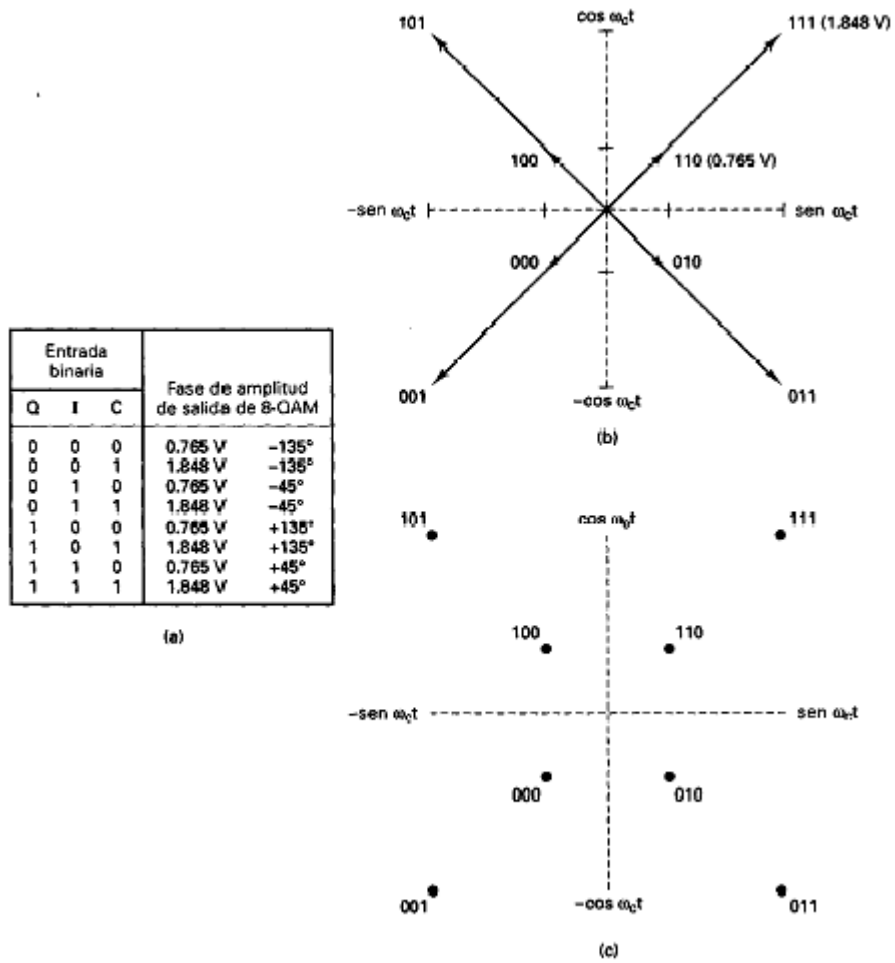


Figura 3.34. a) Tabla de Verdad, b) Fasores, c) constelación 8 QAM

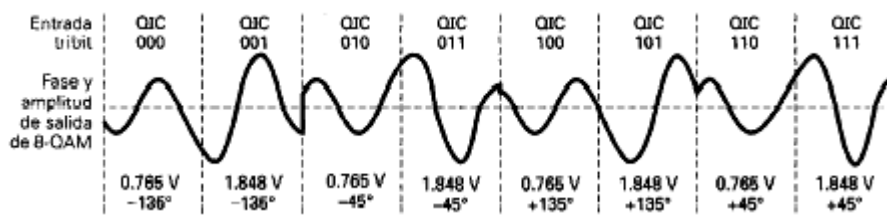


Figura 3.35. Posibles Fases de Salida 8 QAM

El ancho de banda y el baudio al igual que en PSK es de  $f_b/3$ . Así mismo, el receptor es el mismo mostrado para 8 PSK salvo por que las señales de salida de los convertidores analógicos digitales ahora son I/C y Q/C.

3.2.2.6.2. 16 – QAM.

Al igual que el 16 PSK,  $M = 16$ , con entrada de quadbits en el modulador y con variaciones en fase y en amplitud.

Ahora se tienen cuatro canales I, I', Q, Q', que reciben una tasa de bits  $f_b/4$  cada uno. Los bits I, Q determinan la polaridad a la salida de los convertidores de nivel y los bits I', Q' determinan la magnitud de las señales.

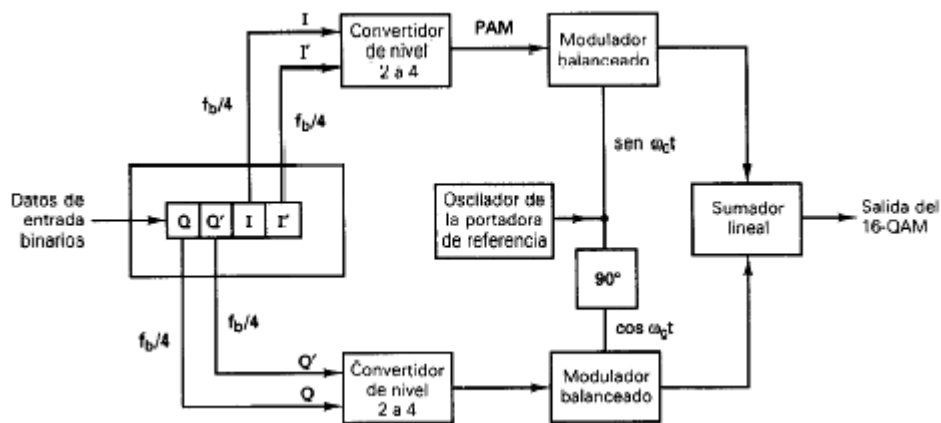


Figura 3.36. Modulador 16 QAM

I	I'	Salida	Q	Q'	Salida
0	0	-0.22 V	0	0	-0.22 V
0	1	-0.821 V	0	1	-0.821 V
1	0	+0.22 V	1	0	+0.22 V
1	1	+0.821 V	1	1	+0.821 V

Figura 3.37. Tablas de Verdad Convertidores de Nivel, a) Canal I, b) Canal Q

La tasa de baudios al igual que en 16 PSK, es igual a  $f_b/4$ , y de manera similar el mínimo ancho de banda es  $f_b/4$  en Hz. La frecuencia fundamental para esta modulación es  $f_b/8$ .

Entrada binaria				Salida de 16-QAM	
Q	Q'	I	I'		
0	0	0	0	0.311 V	-135°
0	0	0	1	0.850 V	-165°
0	0	1	0	0.311 V	-45°
0	0	1	1	0.850 V	-15°
0	1	0	0	0.850 V	-105°
0	1	0	1	1.161 V	-135°
0	1	1	0	0.850 V	-75°
0	1	1	1	1.161 V	-45°
1	0	0	0	0.311 V	135°
1	0	0	1	0.850 V	175°
1	0	1	0	0.311 V	45°
1	0	1	1	0.850 V	15°
1	1	0	0	0.850 V	105°
1	1	0	1	1.161 V	135°
1	1	1	0	0.850 V	75°
1	1	1	1	1.161 V	45°

(a)

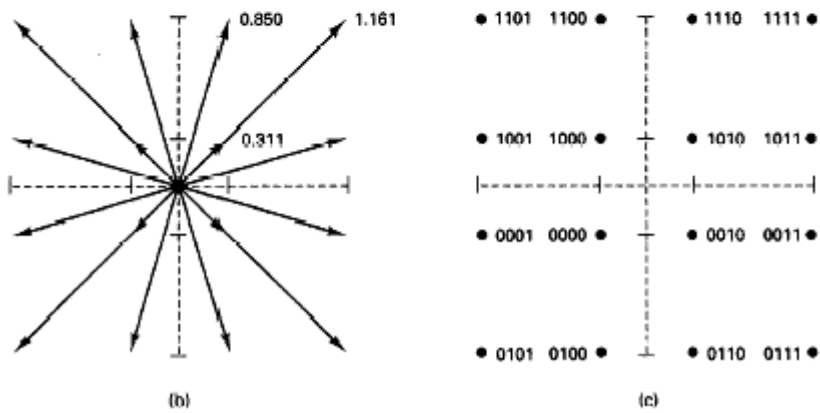


Figura 3.38. a) Tabla de Verdad, b) Fasores, c) Constelación 16 QAM

### 3.2.2.7. Desplazamiento Mínimo De Fase MSK (Minimum Shift Keying)

El método de transmisión por desplazamiento mínimo es una FSK de fase continua con un índice de modulación mínimo,  $h= 0.5$ , que produce señalización ortogonal. Tiene la ventaja de ser una señal de amplitud constante y, por consiguiente, se puede amplificar con amplificadores clase C sin distorsión. MSK es equivalente a OQPSK con forma de pulso senoidal.

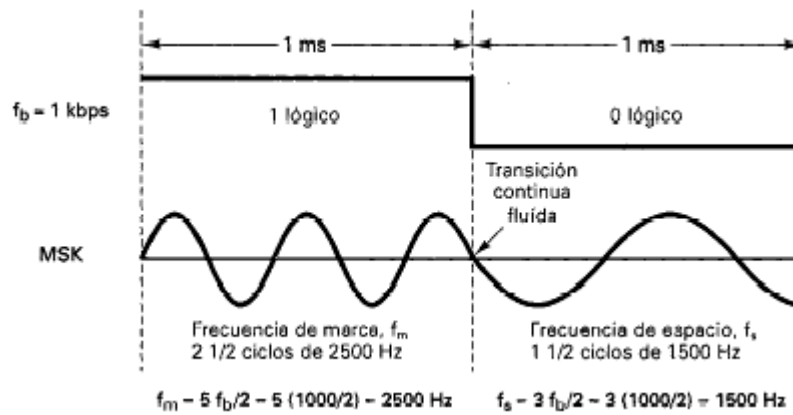


Figura 3.39. Forma de Onda de MSK

MSK a su vez es un caso de lo que se conoce como CPFSK, FSK de fase continua, la cual es una FSK en que se controla la fase de tal manera que se evita toda discontinuidad en la señal modulada mientras se realiza el corrimiento de frecuencia para conducir la información. Por ello tiene un mejor rendimiento de error de bit que un FSK convencional, pero es más costosa de implementar. De esto resulta que la densidad espectral de potencia disminuye a una razón proporcional, al menos, a  $\omega^{-4}$  lejos de la frecuencia portadora. Además, con CPFSK se consigue una **eficiencia teórica de banda de 2 bps / Hz**. La onda CPFSK como tal se define como

$$f(t) = A \cos[\omega_c t + g(t)]$$

Se puede observar que la fase es una función lineal del tiempo en el intervalo  $0 < t < T_b$ , es decir:

$$g(t) = \Delta \omega t \pm g(0)$$

$$g(t) = \pm \frac{P}{2T_b} + g(0) \quad 0 < t < T_b$$

Si se elige por conveniencia  $g(0) = 0$ , los posibles valores de  $g(t)$  para  $t > 0$  corresponden a los del siguiente gráfico, para varios intervalos de bit sucesivos.

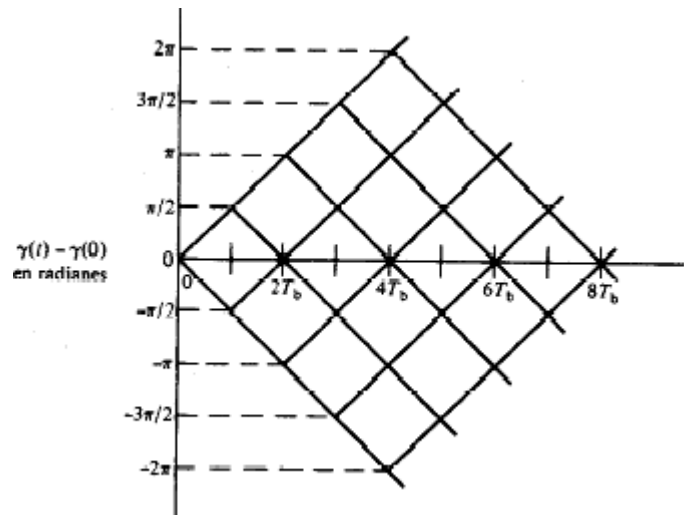


Figura 3.40. Trama de Exceso de Fase MSK

Cada línea de izquierda a derecha en el entramado corresponde a una secuencia binaria de entrada específica, se muestra que sobre cada intervalo de bit, la fase de la onda MSK se adelanta o retrasa exactamente  $90^\circ$ , dependiendo si el dato en el intervalo es 0 o 1, con respecto a la fase portadora en el intervalo de bit precedente. Como todos los corrimientos de fase son de módulo  $2p$ , la fase solo puede tomar valores  $\pm p/2$  en múltiplos impares de  $T_b$  y solo los valores 0,  $p$  en múltiplos pares de  $T_b$ .

Al igual que en QPSK, la corriente binaria de entrada se agrupa en pares de datos, por ejemplo, muestras con subíndice par a la componente en fase y las de subíndice impar a la de cuadratura. Haciendo esto, se nota que para valores sucesivos de cualquiera de los valores de la componente, la fase en exceso siempre aumentará en 0,  $p$ , módulo  $2p$ . Las fases asignadas a una componente diferirán de la otra en  $\pm p/2$  que, excepto por el signo, podrán regularse con un retraso de  $T_b$ . De esta forma, la analogía es mayor con OQPSK, el signo del corrimiento  $\pm p/2$  necesario se elegirá para mantener la continuidad de fase.

Ahora puede tomarse a MSK como un caso especial de OQPSK con una ponderación de pulso senoidal, no rectangular; y además

$$a_i(t) = \cos \mathbf{g}_k$$

$$a_o(t) = p_k \cos \mathbf{g}_k$$

Y recordando que  $\mathbf{g}_k = 0, p$ , módulo de  $2p$ , se tiene

$$p_k = a_i(t)a_q(t)$$

Con esta ecuación y el requisito de continuidad de fase pueden bosquejarse los canales de datos en cuadratura y las ponderaciones sinusoidales.

Como la señalización MSK es equivalente a OQPSK, el rendimiento de probabilidad de error de la MSK con detección de filtro acoplado en presencia de AWGN es el mismo que en QPSK y OQPSK. De esto se deduce que la detección de MSK como dos canales binarios ortogonales proporciona una ventaja de 3 dB  $E_b/N_0$  sobre la detección de FSK ortogonal. La eficiencia de banda de la MSK es definitivamente de 2 bps / Hz, al igual que en QPSK y OQPSK.

La densidad espectral de potencia de OQPSK es  $S_x(w) = \frac{16A^2Tb}{P^2} \left[ \frac{\cos w'Tb}{1 - (2w'Tb/P)^2} \right]^2$

$$S_x(w) = 2A^2Tb[Sa(w'Tb)]^2$$

Y la densidad espectral de potencia de la MSK es

$$S_x(w) = \frac{16A^2Tb}{P^2} \left[ \frac{\cos w'Tb}{1 - (2w'Tb/P)^2} \right]^2$$

Donde  $w'$  es la frecuencia en radianes medida desde la frecuencia portadora.

A continuación se muestra estas ecuaciones en forma gráfica, y se observa que la MSK tiene un lóbulo principal más amplio (primer cero en  $0.75/Tb$ ) que la QPSK y OQPSK (primer cero en  $0.50/Tb$ ), pero la densidad espectral de potencia de la MSK tiene lóbulos más bajos a frecuencias más lejanas de la portadora.

Los circuitos de modulación / desmodulación de la MSK no son mucho más difíciles de elaborar que los de OQPSK, en la figura 3.42 se muestra un diagrama de bloques del modulador MSK. El receptor MSK de filtro acoplado es similar al usado en OQPSK excepto en que las señales de referencia para la detección sincrónica de las componentes en fase y en cuadratura son  $\cos(\pi t/2Tb) \cos w_c t$  y  $\sin(\pi t/2Tb) \sin w_c t$ .

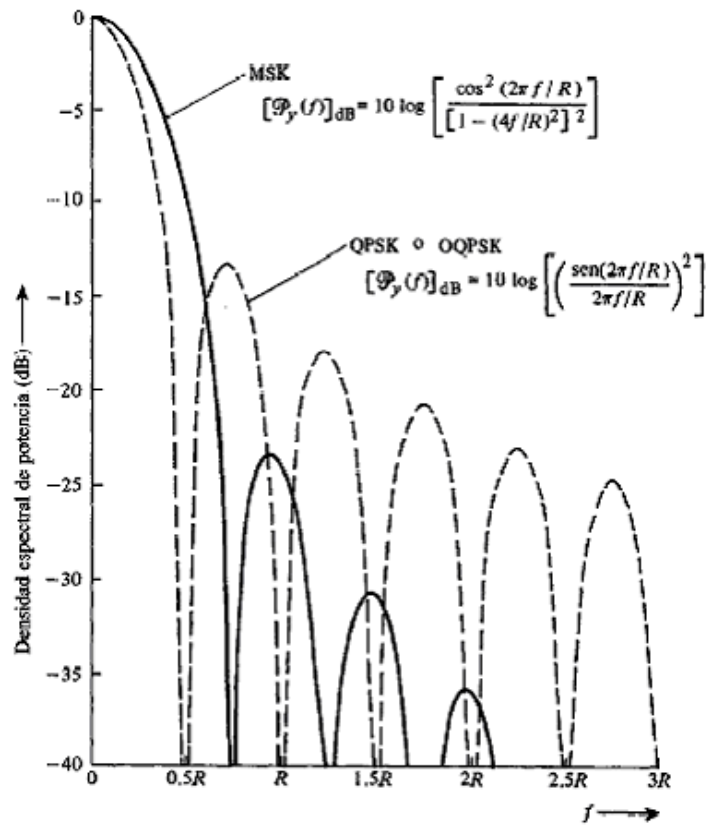


Figura 3.41. PSD de Envolvente Compleja de MSK, QPSK, OQPSK

Se concluye por tanto, que MSK se considere como una OQPSK con ponderación de pulso senoidal o bien una CPM con una separación de frecuencia  $2 \cdot f$  igual a la mitad de la razón de bit. La onda MSK tiene una fase continua en los tiempos de transmisión de bits y la ventaja de una envolvente constante y una densidad espectral de potencia que decrece a razón de  $w^{-4}$  lejos de la frecuencia portadora. La razón de disminución del espectro puede mejorarse usando otras ponderaciones en las componentes de fase y en cuadratura, como  $Sa(2\pi t / T_b)$ , pero solo a expensas de un lóbulo principal más amplio en la densidad espectral.

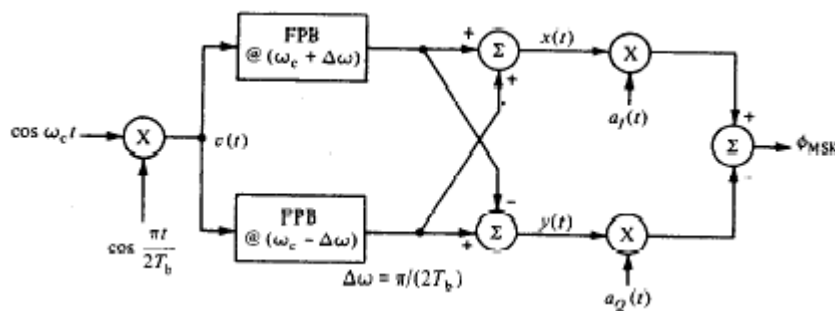


Figura 3.42. Modulador MSK

En general, MSK ofrece un compromiso para sistemas digitales en los que tanto la conservación de la banda como el uso de transmisores de saturación de amplitud (no lineales) son requisitos importantes.

### 3.2.3. DESEMPEÑO DE SISTEMAS

#### 3.2.3.1. Eficiencia del Ancho de Banda.

Es la relación de la tasa de bits al mínimo ancho de banda requerido para un esquema de modulación en particular. Por lo general se normaliza a 1 Hz, por lo tanto, indica el número de bits que pueden propagarse por el medio por cada hertz de ancho de banda.

$$Eficiencia_{BW} = \frac{tasade\ transmision\ (bps)}{minimo\ ancho\ de\ banda\ (Hz)}$$

$$Eficiencia_{BW} = \frac{bits / segundo}{hertz} = \frac{bits / segundo}{ciclos / segundo}$$

$$Eficiencia_{BW} = \frac{bits}{ciclo}$$

Entendido esto, se puede realizar una tabla de las diferentes tipos de modulación para comparar sus requerimientos de codificación, su ancho de banda, su baudio y su eficiencia de ancho de banda.

<i>modulación</i>	<i>Codificación</i>	<i>Ancho de Banda</i> (Hz)	<i>Baudio</i>	<i>Eficiencia BW</i> (bps / Hz)
<b>FSK</b>	Un bit	$=f_b$	$f_b/2$	= 1
<b>BPSK</b>	Un bit	$f_b$	$f_b$	1
<b>QPSK</b>	Dibit	$f_b/2$	$f_b/3$	2
<b>8-PSK</b>	Tribit	$f_b/3$	$f_b/3$	3
<b>8-QAM</b>	Tribit	$f_b/3$	$f_b/3$	3
<b>16-PSK</b>	Quadbit	$f_b/4$	$f_b/4$	4
<b>16-QAM</b>	Quadbit	$f_b/4$	$f_b/4$	4

**Tabla. 3.2. Modulación Digital.- Eficiencia.**



### 3.2.3.2. Probabilidad de Error $P_b(E)$

Términos que se usan indistintamente en la práctica para hablar del error en sistemas de modulación digital, aunque en la realidad son distintos.  $P(e)$  es una expectativa matemática de la tasa de error de bit, mientras que BER es un registro histórico del verdadero rendimiento de error de bit de un sistema.

La probabilidad de error es una función de la relación de *densidad de potencia de energía por bit a ruido* “ $E_b / N_o$ ” y el número  $M$  de posibles condiciones de codificación utilizadas

$$\frac{E_b}{N_o} = \frac{C / f_b}{N / B} = \frac{CB}{Nf_b}$$

$$\frac{E_b}{N_o} = \frac{C B}{N f_b}$$

Es decir, que el  $E_b/N_o$  es igual a la SNR de la señal por la relación de ancho de banda a tasa de bits.

En general la relación de potencia de la portadora a ruido mínima requerida para QAM, es menor a la requerida para sistemas PSK comparables. Además mientras más alto sea el nivel de codificación,  $M$ , más alta será la relación de potencia a ruido mínimo.

La probabilidad de que el detector haga una decisión incorrecta se denomina Probabilidad de Error de Símbolo ( $P_E$ ). Es conveniente también especificar el funcionamiento del sistema por la Probabilidad de Error de Bit ( $P_B$ ), aunque las decisiones se toman para los símbolos cuando  $M > 2$ .

#### 3.2.3.2.1. BPSK Detectado Coherentemente.

Para este caso en particular  $P_E = P_B$ . Se asume que las señales son iguales y que cuando  $s_i(t)$  donde ( $i=1,2$ ) es transmitida, la señal  $r(t)$  recibida es igual a  $s_i(t) + n(t)$ , donde  $n(t)$  es

un proceso AWGN (ruido blando aditivo Gaussiano) y cualquier degradación debida al canal o a la circuitería son desechadas. El ángulo entre  $s_1(t)$  y  $s_2(t)$  es de  $180^\circ$ .

$$P_B = \int_{\sqrt{2E_b/N_0}}^{\infty} \frac{1}{\sqrt{2p}} \exp\left(-\frac{u^2}{2}\right) du$$

$$P_B = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

### 3.2.3.2.2. DBPSK Detectado Coherentemente.

Algunas veces los mensajes (y sus formas de onda asignadas) son codificados diferencialmente y detectados coherentemente para evitar estas ambigüedades de fase. La probabilidad de error de bit para sistemas coherentemente detectados, codificados diferencialmente PSK está dada por

$$P_B = 2Q\left(\sqrt{\frac{2E_b}{N_0}}\right)\left[1 - Q\left(\sqrt{\frac{2E_b}{N_0}}\right)\right]$$

Este tipo de señalización requiere aproximadamente 1 dB más que BPSK. Es más fácil implementar un sistema un sistema DPSK que un sistema PSK, porque DPSK no requiere un sistema sincronizador pero es en algunos casos menos eficiente por la misma razón.

$$P_B = \frac{1}{2} \exp\left(-\frac{Eb}{N_0}\right)$$

### 3.2.3.2.3. FSK Binario Detectado Coherentemente.

$$P_B = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{(1-p)E_b/N_0}}^{\infty} \exp\left(-\frac{u^2}{2}\right) du$$

El coeficiente de correlación cruzado en el tiempo es  $r = \cos \mathbf{q}$ , entre la señal  $s_1(t)$  y  $s_2(t)$ , donde  $\mathbf{q}$  es el ángulo entre los vectores de las señales  $\mathbf{s}_1$  y  $\mathbf{s}_2$ . Para señales antipodales como BPSK,  $\mathbf{q} = \mathbf{p}$ , por lo tanto  $r = -1$ .

Para señales ortogonales como BFSK,  $\mathbf{q} = \mathbf{p}/2$ , por lo tanto  $r = 0$ , por lo tanto:

$$P_B = \frac{1}{\sqrt{2\mathbf{p}}} \int_{\sqrt{E_b/N_0}}^{\infty} \exp\left(-\frac{u^2}{2}\right) du$$

$$P_B = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

#### 3.2.3.2.4. FSK Binario Detectado No Coherentemente.

$$P_B = \frac{1}{2} \exp\left(-\frac{A^2}{4N_0W_f}\right)$$

De esta ecuación se obtiene que el error depende del filtro pasa banda o que  $P_B$  es más pequeño cuando  $W_f$  decrece. Este resultado es válido solamente cuando la ISI no es importante. La mínima  $W_f$  permitida se obtiene cuando en el filtro roll-off el factor  $r = 0$ . Entonces  $W_f = R \text{ bits/s} = 1/T$ , por lo tanto

$$P_B = \frac{1}{2} \exp\left(-\frac{A^2T}{4N_0}\right)$$

$$P_B = \frac{1}{2} \exp\left(-\frac{Eb}{2N_0}\right)$$

Donde  $E_b = (1/2)A^2T$  (energía de bit). Cuando se compara el error de la señalización FSK coherente con la no coherente, se puede observar que para la misma  $P_B$ , FSK no coherente requiere aproximadamente 1 dB más que para FSK coherente, por ejemplo para  $P_B \leq 10^{-4}$ .

MODULACIÓN	$P_B$
PSK (coherente)	$Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$
DPSK (diferenciación coherente)	$\frac{1}{2}\exp\left(-\frac{E_b}{N_0}\right)$
FSK ortogonal (coherente)	$Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$
FSK ortogonal (no coherente)	$\frac{1}{2}\exp\left(-\frac{1}{2}\frac{E_b}{N_0}\right)$

Tabla. 3.3. Probabilidad de Error en Sistemas Binarios.

El receptor no coherente es más fácil de implementar, ya que las señales de referencia coherentes necesitan ser generadas, por lo que casi todos los receptores usan detección no coherente.

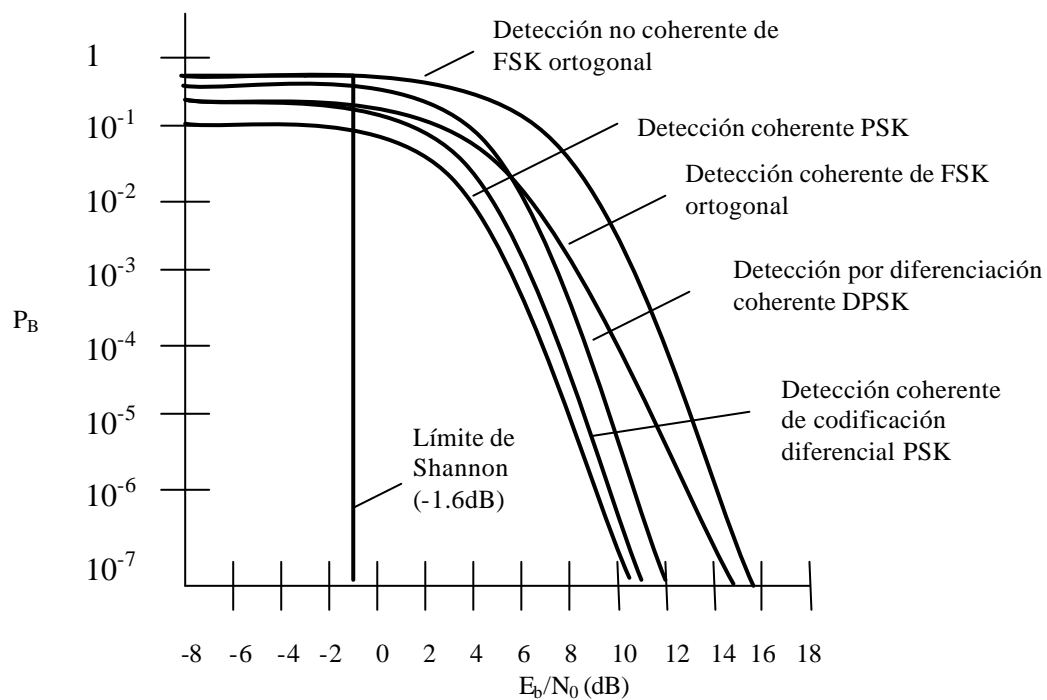


Figura. 3.43. Curvas de Probabilidad de Error en Sistemas Binarios

### 3.2.3.2.5. Señalización M-aria.

La señalización m-aria mejora o degrada el rendimiento del error. En la figura 3.44 se ilustra la probabilidad de error de bit versus  $E_b/N_0$  para detección coherente de MFSK y de MPSK, sobre un canal Gaussiano. En la gráfica de MFSK mientras  $k$  se incrementa, las curvas se mueven en la dirección del mejoramiento de rendimiento del error.

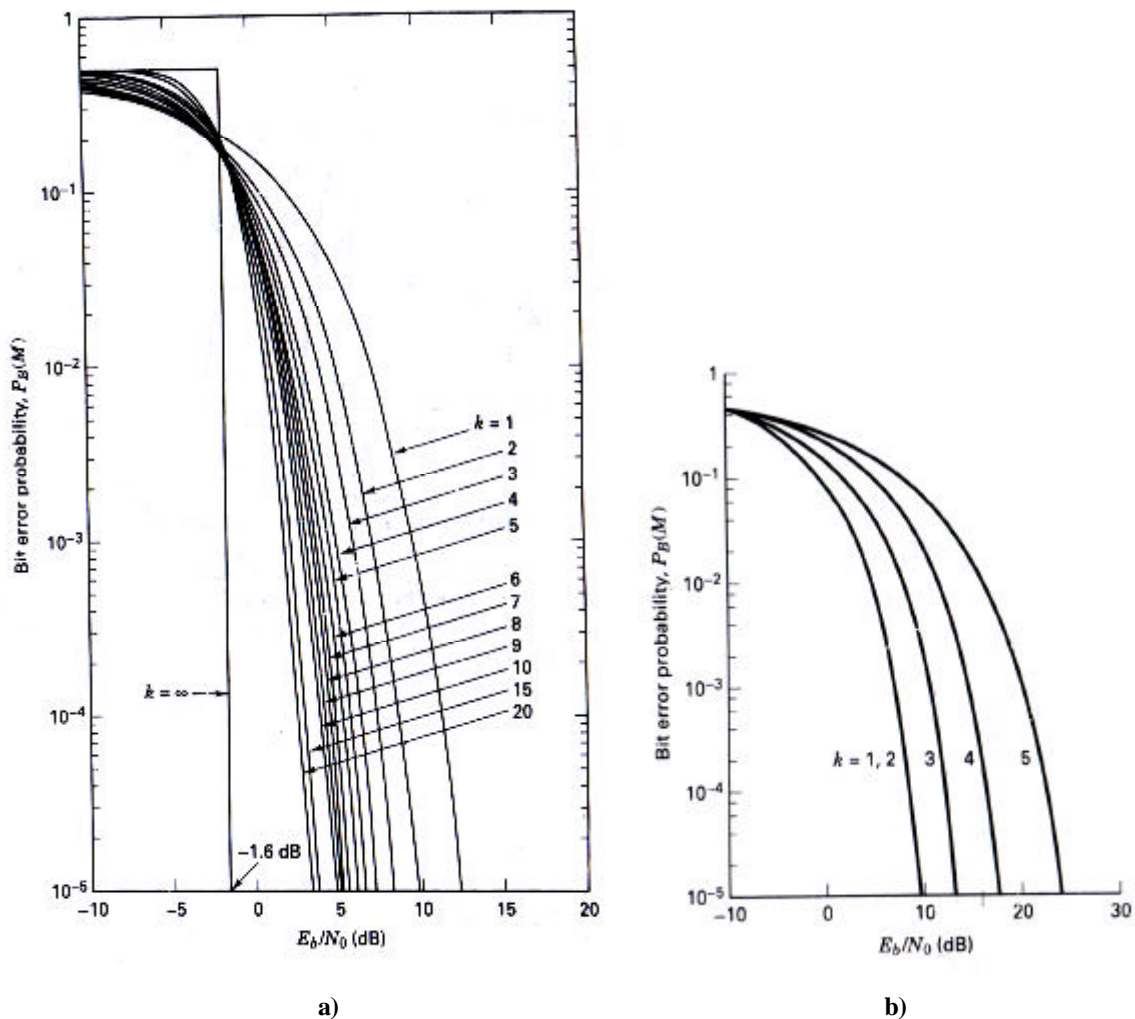


Figura. 3.44. Probabilidad de Error de Bit Coherentemente Detectada

a) Señalización M-aria Ortogonal (MFSK), b) Señalización Fase Múltiple (MPSK)

En cambio en la gráfica de MPSK mientras  $k$  se incrementa las curvas se mueven en la dirección del degradamiento de rendimiento del error. Con ello se concluye que la señalización m-aria produce el mejoramiento de rendimiento del error con la señalización ortogonal y el degradamiento del error de rendimiento con la señalización de fase múltiple.

Sin embargo, muchos sistemas hacen uso de la señalización en la fase múltiple, debido a que existen otros criterios de rendimiento para la selección de un sistema, como ancho de banda, complejidad, costo, entre otras.

Así, una característica del rendimiento que no está explícitamente visto en las gráficas es el requerimiento de un sistema de ancho de banda, el mismo que: para la caracterización de señales m-arias ortogonales, mientras  $k$  incrementa, el requerimiento de ancho de banda también se incrementa. En cambio, para señales m-arias en fase múltiple mientras  $k$  se incrementa, aumenta la tasa de bits que puede ser transmitida dentro de un mismo ancho de banda, es decir que este requerimiento es menor para una misma  $R_b$ .

En el caso de señalización ortogonal, el mejoramiento del error de rendimiento puede ser logrado con el gasto de ancho de banda. En el caso de señalización en fase múltiple, el rendimiento del ancho de banda puede ser logrado con el gasto del error de rendimiento.

### 3.2.3.2.6. Probabilidad de Error de Símbolo para MPSK.

La representación del error de símbolo  $P_E(M)$ , para una señal igualmente probable coherentemente detectada M-aria PSK, se puede describir con

$$P_E(M) = 2Q\left(\sqrt{\frac{2E_s}{N_0}} \sin \frac{\pi}{M}\right)$$

$$E_s = E_b (\log_2 M) \quad \text{Energía de Símbolo}$$

$$M = 2^k \quad \text{Número de Niveles M-arios}$$

### Relación $P_B / P_E$ .

$P_B$  es menor o igual  $P_E$  al igual en los sistemas MFSK, con la diferencia que en ellos seleccionar uno de los  $(M-1)$  símbolos erróneos es igualmente probable, mientras que en la señalización de fase cada vector de señalización no es equidistante de los demás de su grupo.

Para la asignación de los vectores se puede usar código binario, en el que la secuencia binaria corresponde a símbolos adyacentes o código Gray, en el cual los símbolos adyacentes difieren solo en un bit. Con esto último la ocurrencia de un error en más de un bit es menos probable en un error de símbolo que usando secuencias binarias.

La implementación del código Gray no requiere circuitería especial o adicional. Considerando esto, la relación  $P_B / P_E$  está dada por

$$P_B = \frac{P_E}{\log_2 M} \quad \text{para } P_E \ll 1$$

Entendido esto, y sobre la base de que QPSK tiene la **misma probabilidad de bit** que un BPSK, debido a las características de sus moduladores iguales y en cuadratura, se debe tener muy en cuenta que la **probabilidad de error de símbolo es distinta** para estos dos sistemas. Para BPSK será  $P_E = P_B$ , mientras que para QPSK  $P_E = 2 P_B$ .

### 3.2.3.2.7. M- QAM.

Para sistemas M-arios mayores a 4, QAM funcionará mejor que PSK debido a la menor distancia que existe entre los diferentes puntos de la constelación. La expresión general de la distancia entre puntos es

$$d = \frac{\sqrt{2}}{M-1} D$$

Donde  $d$  es la distancia de error,  $M$  el número de niveles en cada eje y  $D$  la amplitud pico de la señal.

La expresión general para el error de bit de un sistema QAM de  $M$  niveles es

$$P_B(e) = \frac{1}{\log_2 M} \left( \frac{M-1}{M} \right) Q \left( \frac{\sqrt{\log_2 M}}{M-1} \sqrt{\frac{Eb}{No}} \right)$$

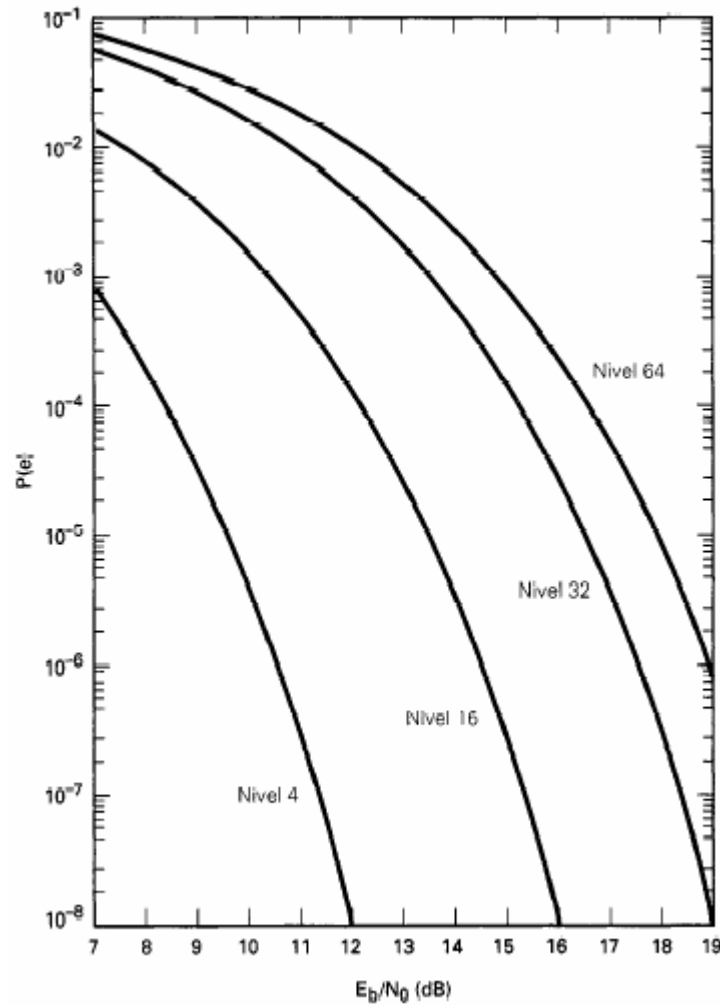


Figura. 3.45. Probabilidad de Error de Bit para QAM

### 3.2.3.2.8. Probabilidad de Error de Símbolo para MFSK.

La representación del límite superior del error de símbolo  $P_E(M)$ , para una señal igualmente probable coherentemente detectada M-aria ortogonal FSK, se puede describir con

$$P_E(M) \leq (M-1) Q \sqrt{\frac{E_s}{N_0}}$$

El desempeño de error de símbolo para una señal igualmente probable **no coherentemente** detectada M-aria ortogonal FSK, se puede describir con



$$P_E(M) = \frac{1}{M} \exp\left(-\frac{E_s}{N_0}\right) \sum_{j=2}^M (-1)^j \binom{M}{j} \exp\left(-\frac{E_s}{jN_0}\right)$$

Donde

$$\binom{M}{j} = \frac{M!}{j!(M-j)!}$$

Por lo que para el caso binario resulta

$$P_B = \frac{1}{2} \exp\left(-\frac{E_b}{2N_0}\right)$$

Y su límite superior al igual que en la detección coherente, está dado por

$$P_E(M) < \frac{M-1}{2} \exp\left(-\frac{E_s}{2N_0}\right)$$

### Relación $P_B / P_E$ .

Para las señales ortogonales, la relación entre la probabilidad de error de bit  $P_B$  y la probabilidad de error de símbolo  $P_E$  para un sistema M-ario es

$$\frac{P_B}{P_E} = \frac{2^{k-1}}{2^k - 1} = \frac{M/2}{M-1}$$

Relación que, cuando k tiende a infinito se reduce a

$$\lim_{k \rightarrow \infty} \frac{P_B}{P_E} = \frac{1}{2}$$

### **3.3. PRÁCTICAS CON MATLAB**

#### **3.3.1. MODULACIÓN DIGITAL**

##### **3.3.1.1. Objetivos**

- Utilizar los conocimientos de transmisión digital banda base y de modulación de onda continua para el estudio de transmisión digital pasabanda.
- Generar secuencias en banda base para luego utilizando las funciones desarrolladas para MATLAB poder modularlas en pasa banda.
- Modular una secuencia bandabase con modulación por desplazamiento de Amplitud ASK y observar la densidad espectral de potencia con respecto a la densidad espectral de potencia (PSD) de la secuencia inicial en banda base.
- Modular una secuencia bandabase con modulación por desplazamiento de Fase PSK y observar la densidad espectral de potencia con respecto a la densidad espectral de potencia (PSD) de la secuencia inicial en banda base.
- Modular una secuencia bandabase con modulación por desplazamiento de Frecuencia FSK y observar la densidad espectral de potencia con respecto a la densidad espectral de potencia (PSD) de la secuencia inicial en banda base.

##### **3.3.1.2. Procedimiento**

Para la realización de la presente práctica se utilizan funciones realizadas para MATLAB y cuyos códigos fuente se encuentran disponibles en los archivos del mismo nombre. Estas funciones simplifican el desarrollo y mejoran el entendimiento y alcance de los objetivos planteados. Las funciones han sido colocadas en el mismo directorio de la práctica y son compatibles con MATLAB7, que es la versión utilizada en el desarrollo de este proyecto.

1. Para esta práctica la señal de datos binarios tiene una tasa de bits de 1000 bps y la señal modulada una amplitud pico de 1 V.
2. Para la generación de una señal ASK:
  - Se genera una secuencia de 60 datos binarios cuyos primeros 6 dígitos sean [1 1 0 0 1 0] usando la función **binary** para los 54 datos restantes.
  - Para obtener la señal ASK, con una frecuencia de portadora de 6 KHz, se puede generar una secuencia unipolar NRZ de la secuencia de datos binarios generada anteriormente.
  - Se mezcla o multiplica esta secuencia NRZ con la salida de un oscilador que opera a 6 KHz, habiendo uso de la función **mixer (entrada, freq\_portadora)**.
  - Debido a que se tienen 40 muestras por bit, se grafican las 400 primeras muestras para obtener los 6 bits conocidos de la secuencia de entrada y cuatro adicionales generados por **binary**. La figura 3.46 muestra las señales antes y después de la modulación.

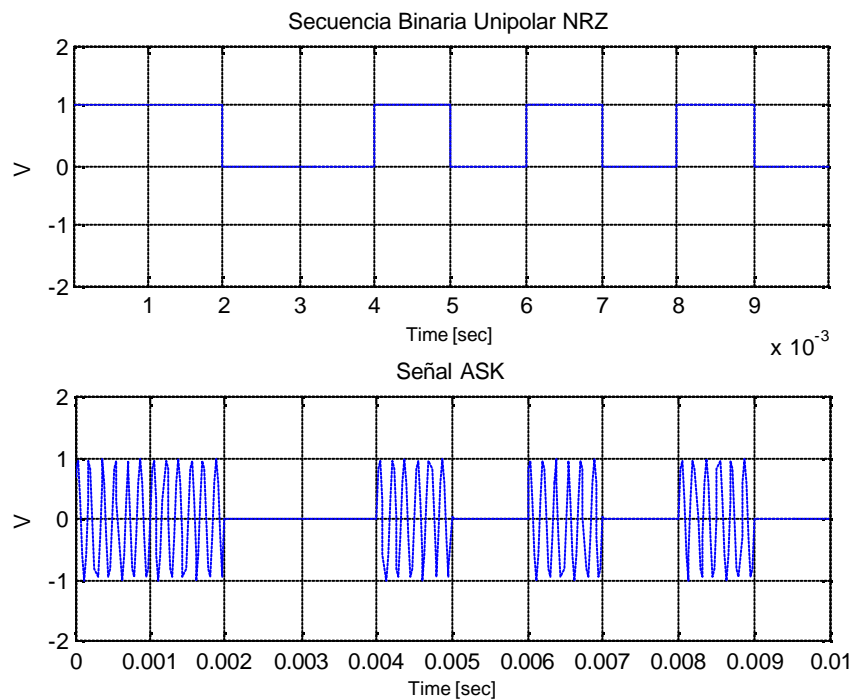
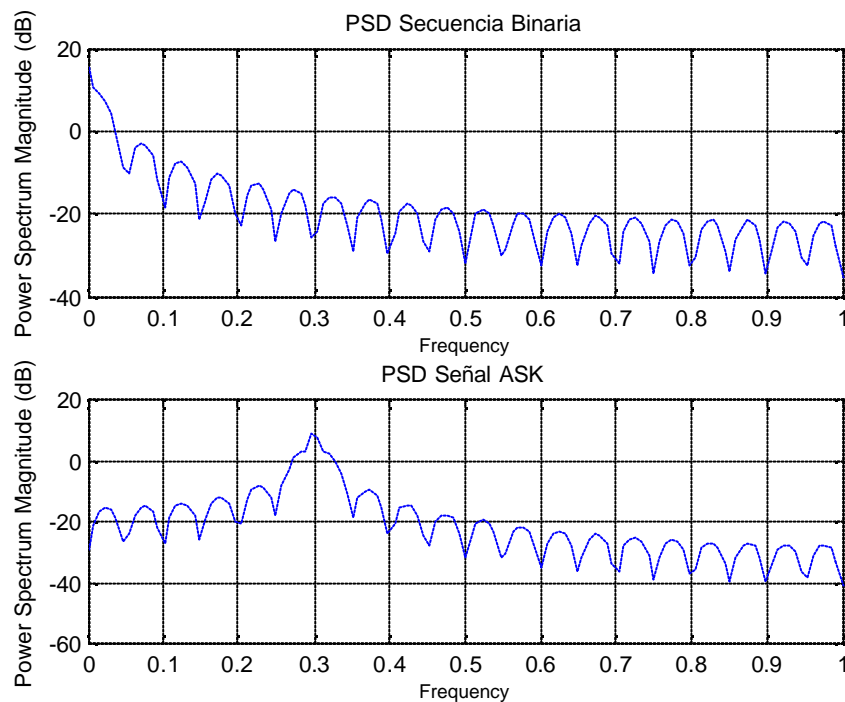


Figura. 3.46. Señal Modulada ASK, portadora 6 KHz

- Luego se visualizan la densidad espectral de potencia de ambas señales usando la función **psd (señal)**, siendo el argumento para ella, la señal antes y después de la modulación. El resultado se muestra en la figura 3.47.

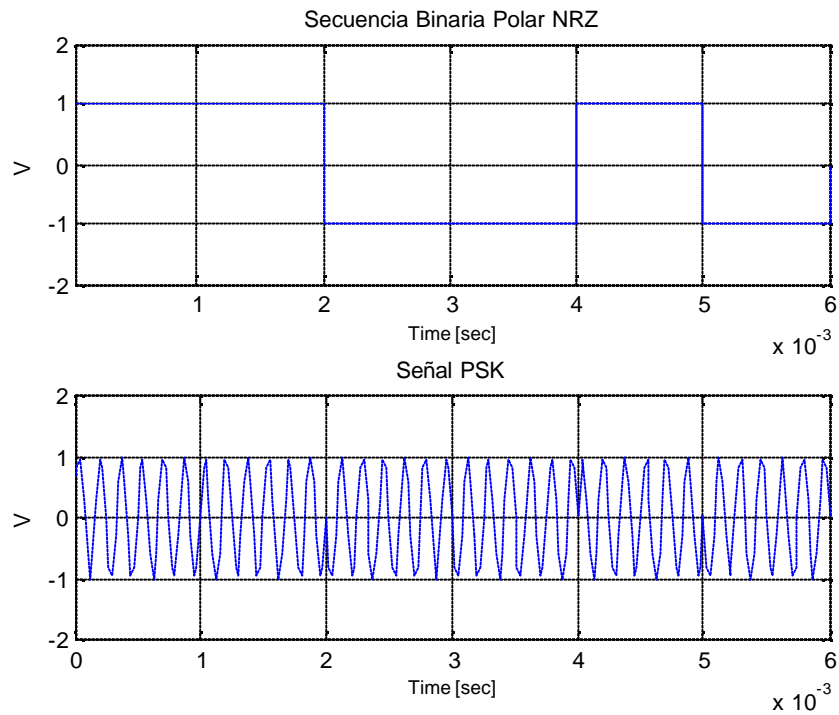


**Figura. 3.47. Densidad Espectral de Potencia: Secuencia Binaria y Señal ASK**

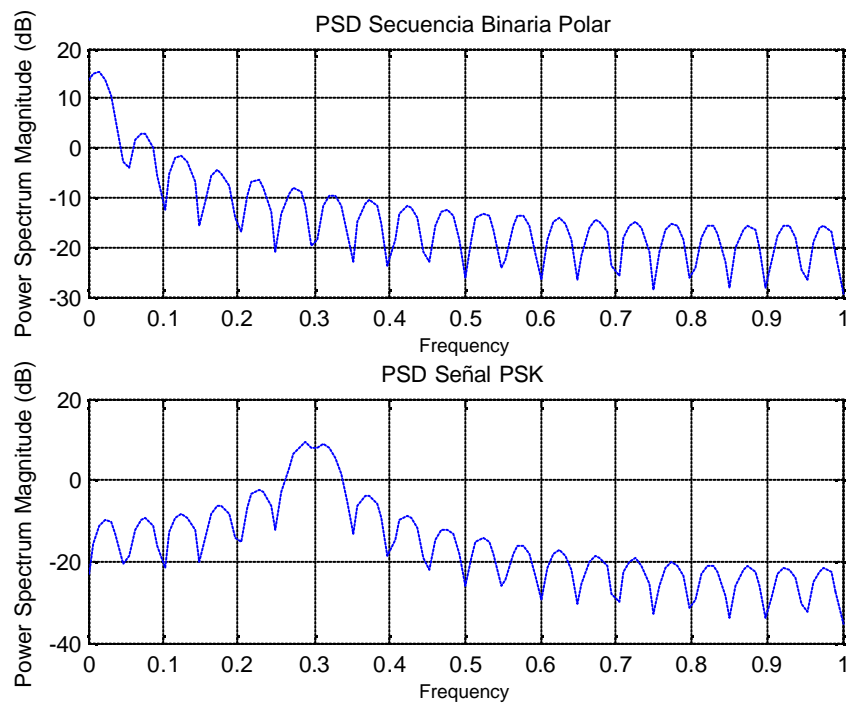
### 3. Para la generación de una señal PSK:

- Para obtener una señal PSK con una portadora de 6 KHz, se lo puede tener tras generar una secuencia polar NRZ a partir de la secuencia de datos binarios determinados anteriormente.
- Se multiplica la señal de la secuencia NRZ con la salida de un oscilador operando a 6 KHz, haciendo uso de la función **mixer**.
- Debido a que se tienen 40 muestras por bit, se grafican las 240 primeras muestras para obtener los 6 bits conocidos de la secuencia de entrada. La figura 3.48 muestra las señales antes y después de la modulación.

- Luego se visualizan la densidad espectral de potencia de ambas señales usando la función **psd (señal)**, siendo el argumento de esta función, la señal antes y después de la modulación. El resultado se muestra en la figura 3.49.



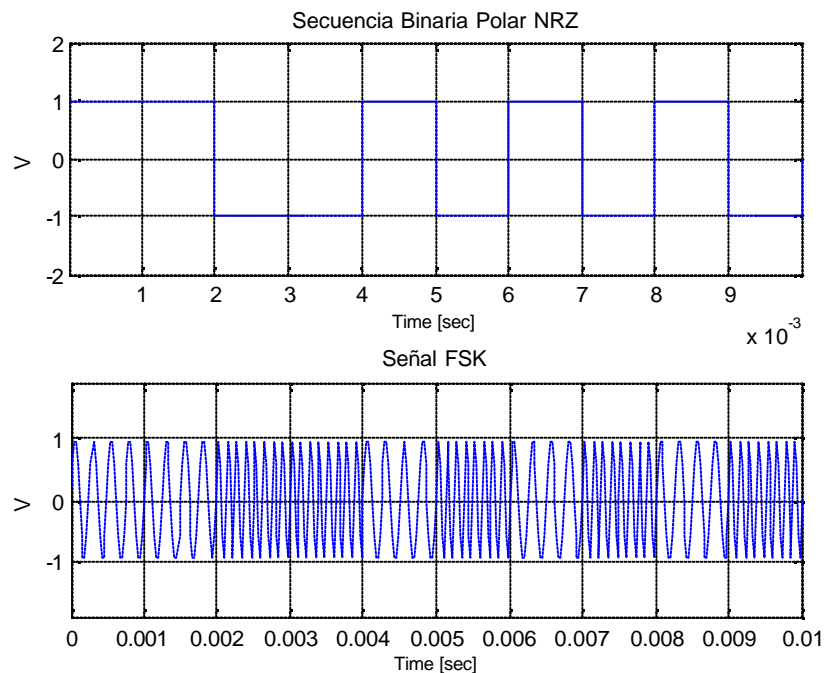
**Figura. 3.48. Señal Modulada PSK, Portadora 6 KHz.**



**Figura. 3.49. Densidad Espectral de Potencia: Secuencia Binaria y Señal PSK**

## 4. Para la generación de una señal FSK

- Para obtener una señal FSK de fase continua con frecuencia de marca y espacio de 4KHz y 8KHz respectivamente, se puede partir de una secuencia de datos codificados en polar NRZ de la secuencia de entrada ya conocida.
- Se aplica la forma de onda polar en la entrada de un oscilador controlado por voltaje, **VCO**, que para esta practica tiene una frecuencia de trabajo libre de 6 KHz y una sensibilidad de frecuencia de  $-2 \text{ KHz} / \text{V}$ .
- Debido a que se tienen 40 muestras por bit, se grafican las 400 primeras muestras para obtener los 6 bits conocidos de la secuencia de entrada más cuatro generados por la función **binary**. La figura 3.50 muestra las señales antes y después de la modulación.



**Figura. 3.50. Señal Modulada FSK, Marca: 8KHz, Espacio: 4 KHz**

- Luego se visualizan la densidad espectral de potencia de ambas señales usando la función **psd (señal)**, siendo el argumento de esta función, la señal antes y después de la modulación. El resultado se muestra en la figura 3.51.

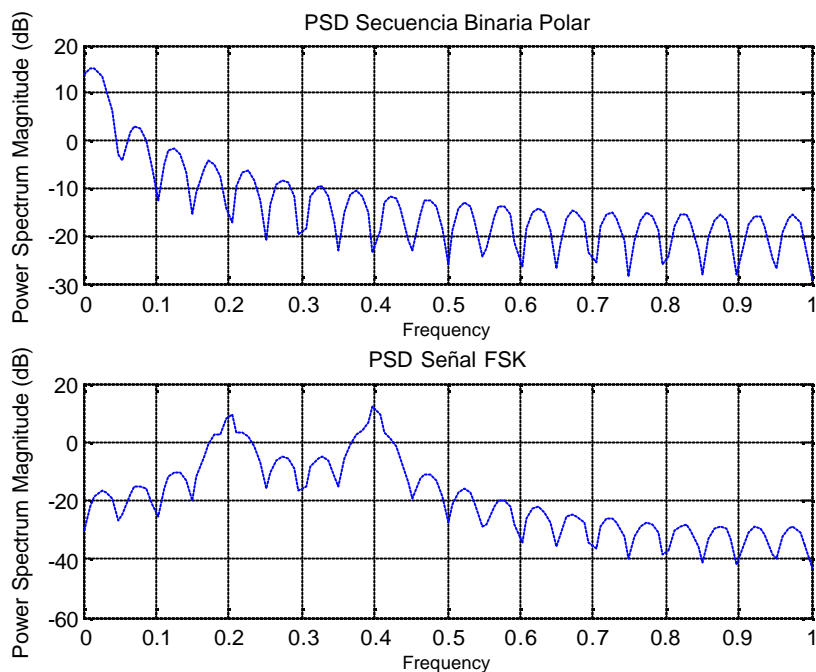


Figura. 3.51. Densidad Espectral de Potencia: Secuencia Binaria y Señal FSK

### 3.3.1.3. Conclusiones

- Se observa gráficamente como se constituye una señal ASK asignando formas de onda de amplitud 1V para cuando se tiene un “1” o de amplitud nula para denotar un “0”.
- Se puede apreciar de forma clara los cambios de fase de  $180^\circ$  entre las señales que representan la presencia de un “1” y de un “0” de la modulación PSK, debido a que se trata de un caso binario.
- En las gráficas de la densidad espectral de potencia se puede observar que para la secuencia binaria el lóbulo principal se encuentra centrado en el origen del eje de la frecuencia, mientras que para ASK y PSK el lóbulo principal se encuentra desplazado hacia una frecuencia portadora.
- En la densidad espectral de potencia de la señal FSK se puede apreciar que existen dos lóbulos principales centrados en las frecuencias de marca y espacio, mientras que el lóbulo principal de la secuencia binaria inicial se encuentra centrado en la frecuencia cero.

**Código Mod\_Dig.m**

```

%Modulacion Digital,
%Generacion de Señales ASK, PSK y FSK
%Densidad Espectral de Potencia.

clc;
clear all;

% *****

SAMPLING_CONSTANT = 40;
BINARY_DATA_RATE = 1000;
SAMPLING_FREQ = BINARY_DATA_RATE * SAMPLING_CONSTANT;

global SAMPLING_CONSTANT;

% *****

b=[1 1 0 0 1 0 binary(54)];

unr=wave_gen(b,'unipolar_nrz',1000);
ask=mixer(unr, osc(6000));

tt=[1:400];

figure(1)
subplot(2,1,1), waveplot(unr(tt))
subplot(2,1,2), waveplot(ask(tt))

figure(2)
subplot(2,1,1), psd(unr)
subplot(2,1,2), psd(ask)

pnr=wave_gen(b,'polar_nrz',1000);
psk=mixer(pnr, osc(6000));

tt=[1:240];

figure(3)
subplot(2,1,1), waveplot(pnr(tt))
subplot(2,1,2), waveplot(psk(tt))

figure(4)
subplot(2,1,1), psd(pnr)
subplot(2,1,2), psd(psk)

fsk= vco (pnr);

tt=[1:400];

figure(5)
subplot(2,1,1), waveplot(pnr(tt))
subplot(2,1,2), waveplot(fsk(tt))

figure(6)
subplot(2,1,1), psd(pnr)
subplot(2,1,2), psd(fsk)
clc;

```



### 3.3.2. DETECCIÓN DE SEÑALES PASABANDA

#### 3.3.2.1. Objetivos

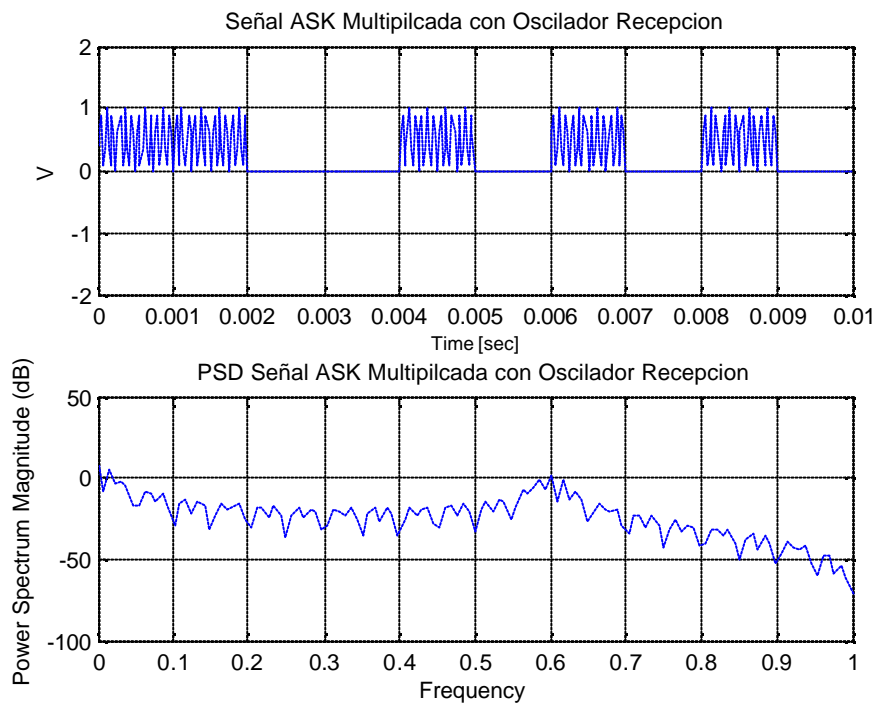
- Comprobar mediante los resultados de la simulación las diferencias entre la detección coherente y la no coherente de señales moduladas digitalmente.
- Realizar la detección de señales moduladas digitales de forma Coherente y No Coherente utilizando las funciones que han sido desarrolladas para MATLAB.
- Observar los efectos de tener error en la fase del oscilador de recuperación en la recepción del sistema de comunicación.
- Observar la influencia y consecuencias de la existencia de desplazamientos de frecuencia en el oscilador de recuperación en el receptor del sistema.
- Aprender el desempeño de sistemas de modulación digital pasabanda ante la presencia de ruido AWGN en el canal de transmisión contaminando las señales que pasan por él.

#### 3.3.2.2. Procedimiento

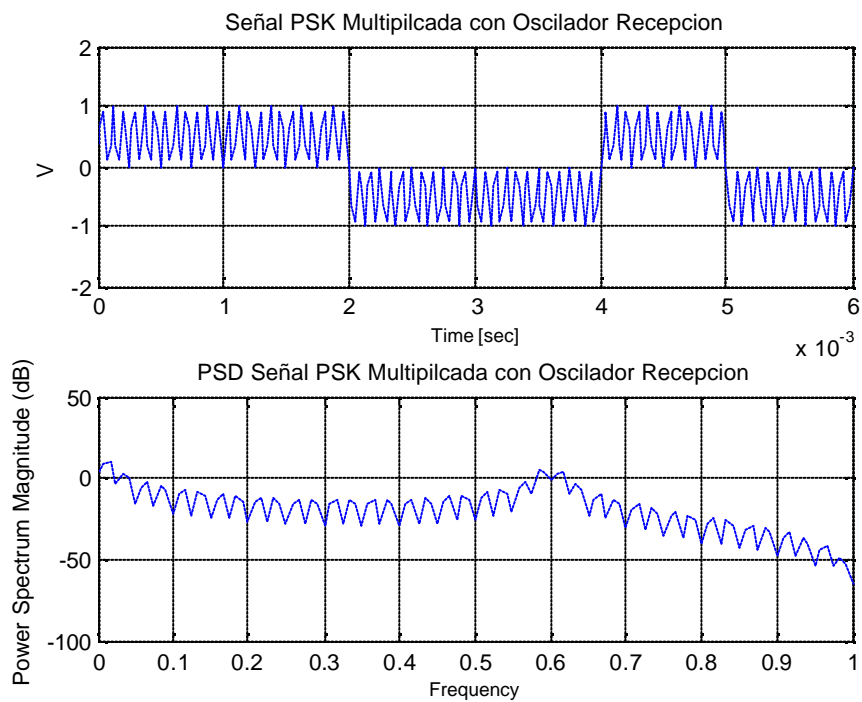
Para la realización de la presente práctica se utilizan funciones realizadas para MATLAB y cuyos códigos fuente se encuentran disponibles en los archivos del mismo nombre. Estas funciones simplifican el desarrollo y mejoran el entendimiento y alcance de los objetivos planteados. Las funciones han sido colocadas en el mismo directorio de la práctica y son compatibles con MATLAB7, que es la versión utilizada en el desarrollo de este proyecto.

1. Para realizar la detección Coherente de las señales ASK y PSK:
  - a) Se multiplica la señal modulada ASK o PSK por una portadora generada localmente en el receptor, a la misma frecuencia y fase de la portadora usada en el transmisor.

- b) Se visualiza la señal obtenida a la salida del multiplicador (**mixer**), junto con su densidad espectral como se muestra en la figura 3.52 para ASK y 3.53 para PSK.

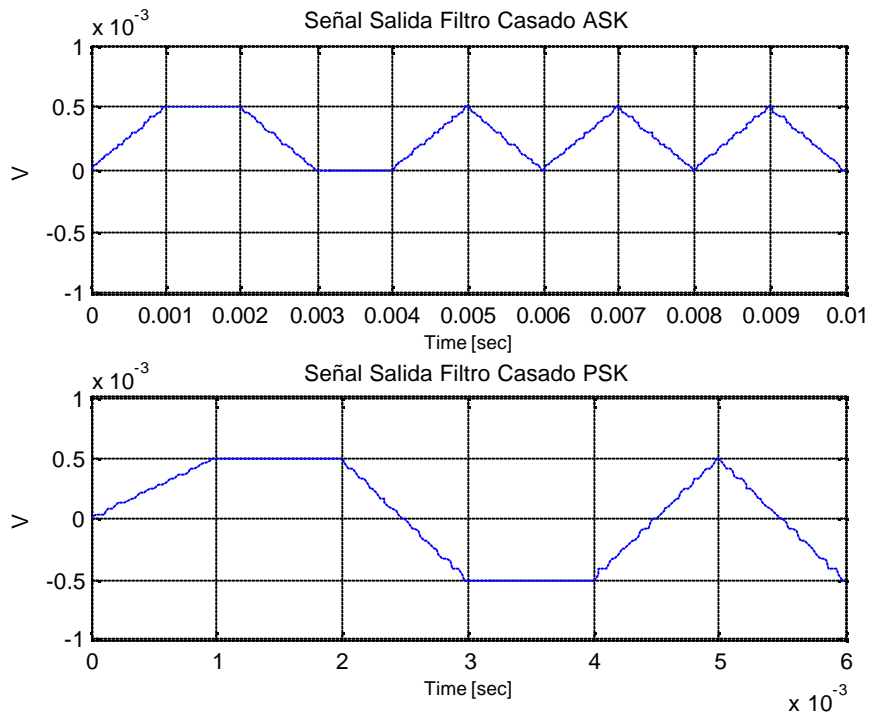


**Figura. 3.52.** Señal ASK Multiplicada por Oscilador de Recuperación y su PSD



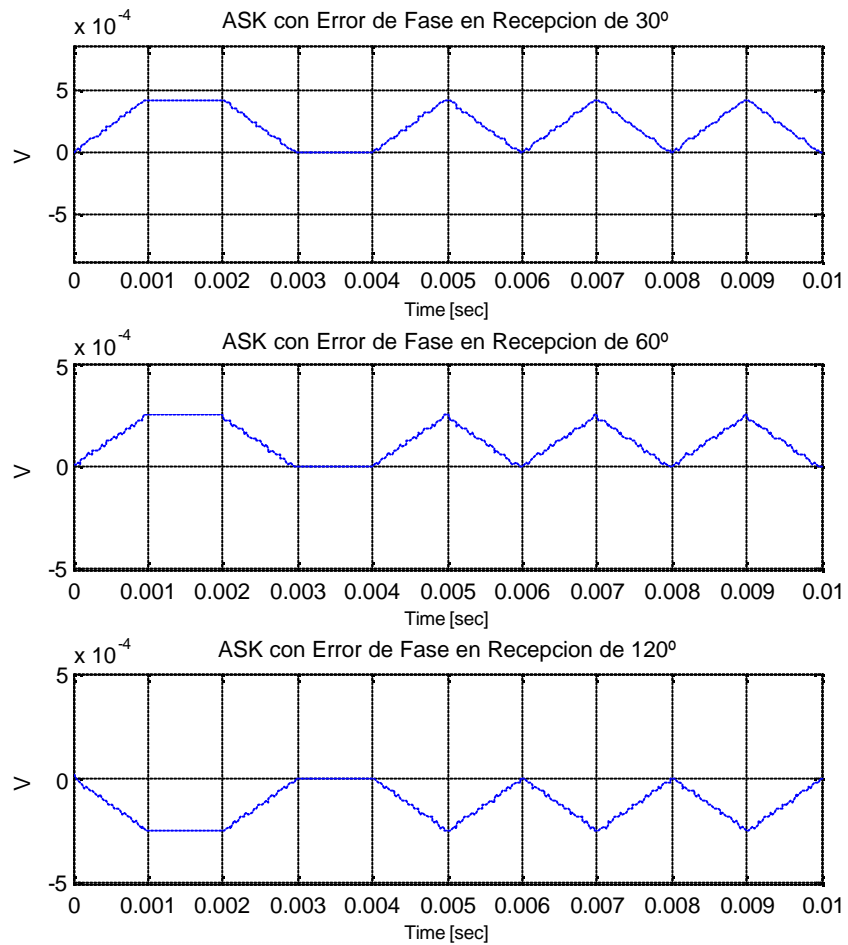
**Figura. 3.53.** Señal PSK Multiplicada por Oscilador de Recuperación y su PSD

- c) Se aplica esa señal a un filtro casado mediante la función **match** ('código', **entrada**), donde código se refiere al código de línea usado en el transmisor y la entrada es la señal salida del multiplicador.
- d) Se muestra en la figura 3.54 la salida del filtro casado para ASK y PSK.



**Figura. 3.54. Señal ASK y PSK a la Salida del Filtro Casado del Receptor**

- Para observar los efectos del error de fase, se demodula la señal ASK usando un oscilador que incluya este error de la forma **osc (freq\_portadora, error\_fase)**, con respecto a la fase de la portadora. En la figura 3.55 se muestra las secuencias demoduladas para errores de fase de  $30^\circ$ ,  $60^\circ$  y  $120^\circ$ .



**Figura. 3.55. Señal ASK con Errores de Fase en el Oscilador del Receptor**

- Para observar los efectos del error de fase, se demodula la señal PSK usando un oscilador que incluya este error de la forma  $\text{osc}(\text{freq\_portadora}, \text{error\_fase})$ , con respecto a la fase de la portadora. En la figura 3.56 se muestra las secuencias demoduladas para errores de fase de 30°, 60° y 120°.

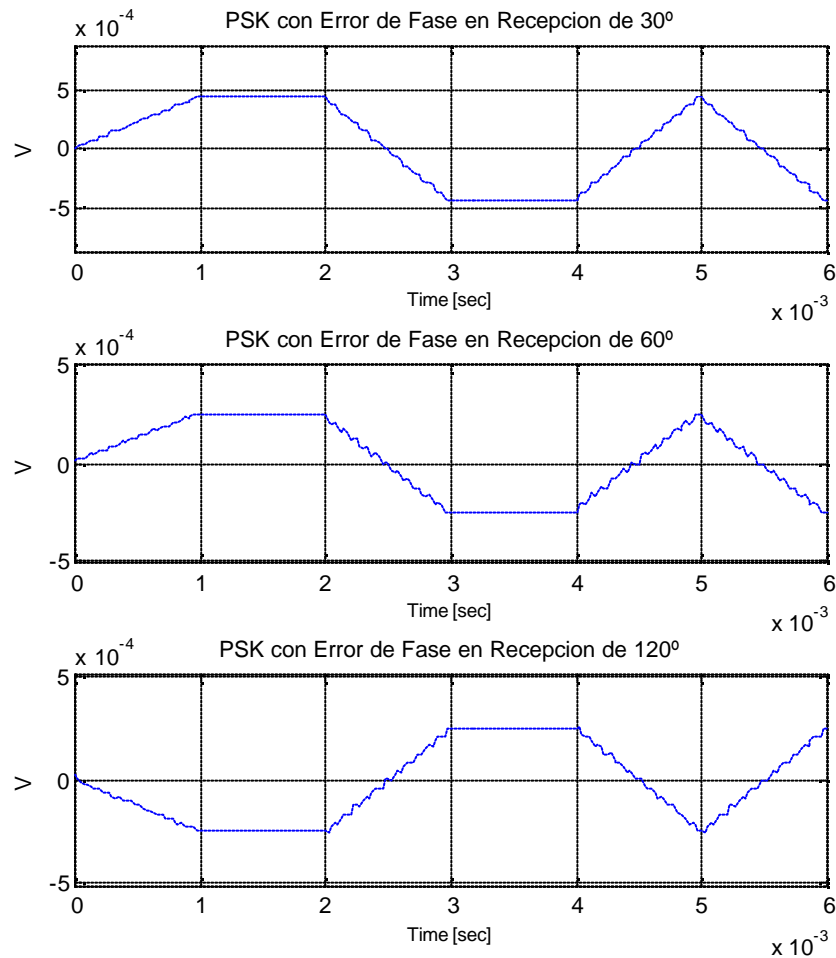
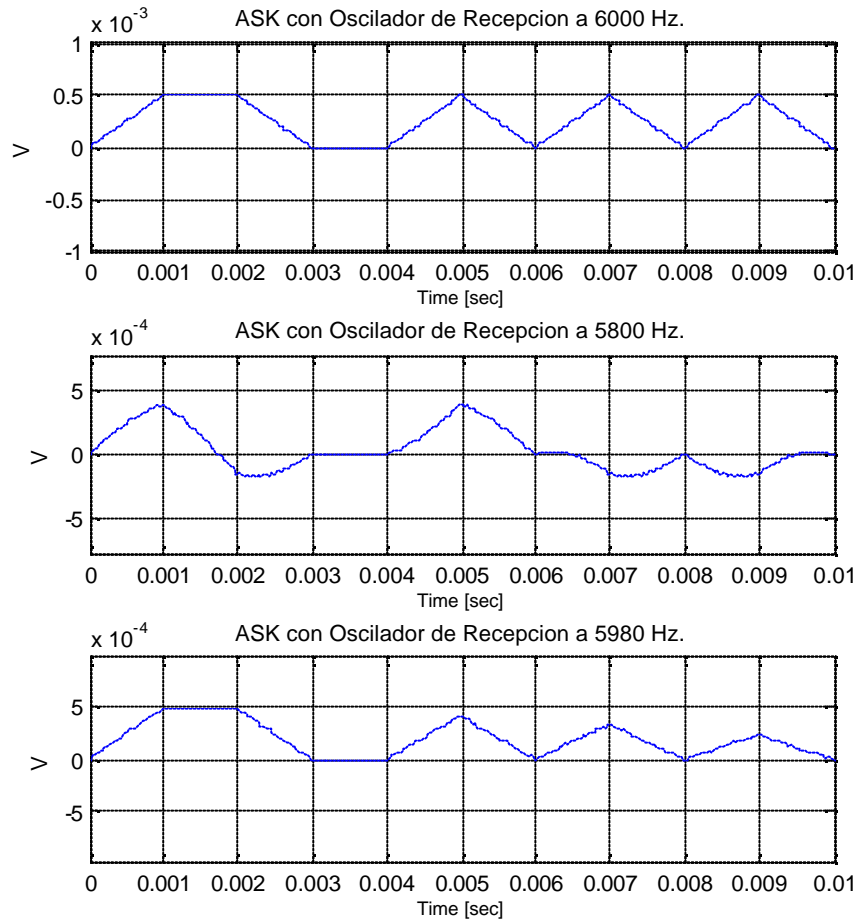


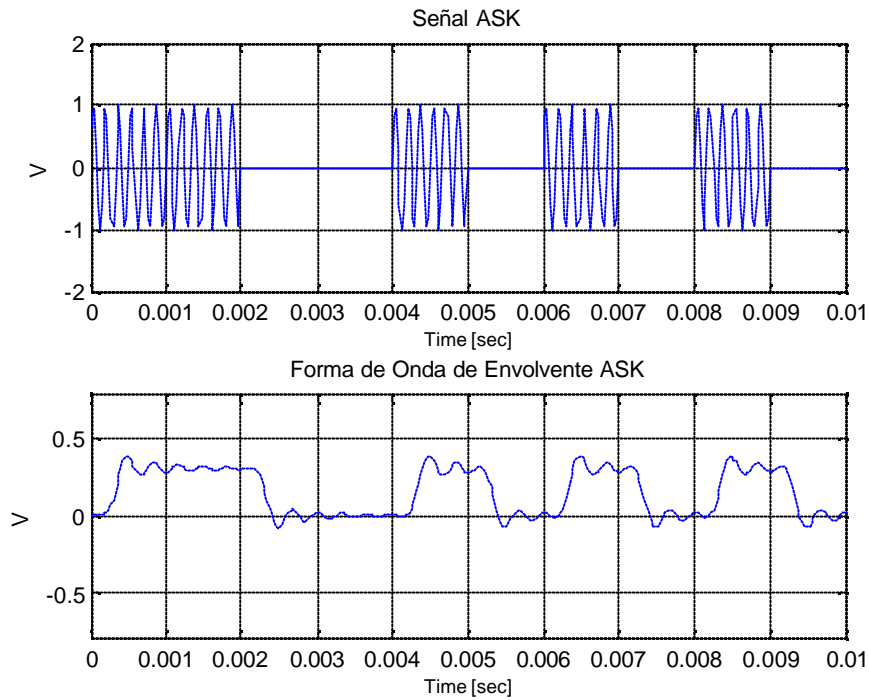
Figura. 3.56. Señal PSK con Errores de Fase en el Oscilador del Receptor

4. Para visualizar los efectos de la desviación de frecuencias en la demodulación de señales ASK, se demodula dicha señal con osciladores a 5800 y 5980 Hz, y los resultados de ello se muestran en la figura 3.57.
5. Para la detección No Coherente de señales, se usa un filtro pasa banda que reduce el ruido fuera de banda y la interferencia, un detector de envolvente conformado por un rectificador y un filtro pasa bajos, cuyo ancho de banda es mayor ancho de banda de la señal y menor a la frecuencia de portadora.
6. Se selecciona por tanto un ancho de banda del LPF de 4000 Hz a ser usado en la función **envelope (entrada, BW)**. El resultado de esto junto con la señal ASK se muestran en la figura 3.58.

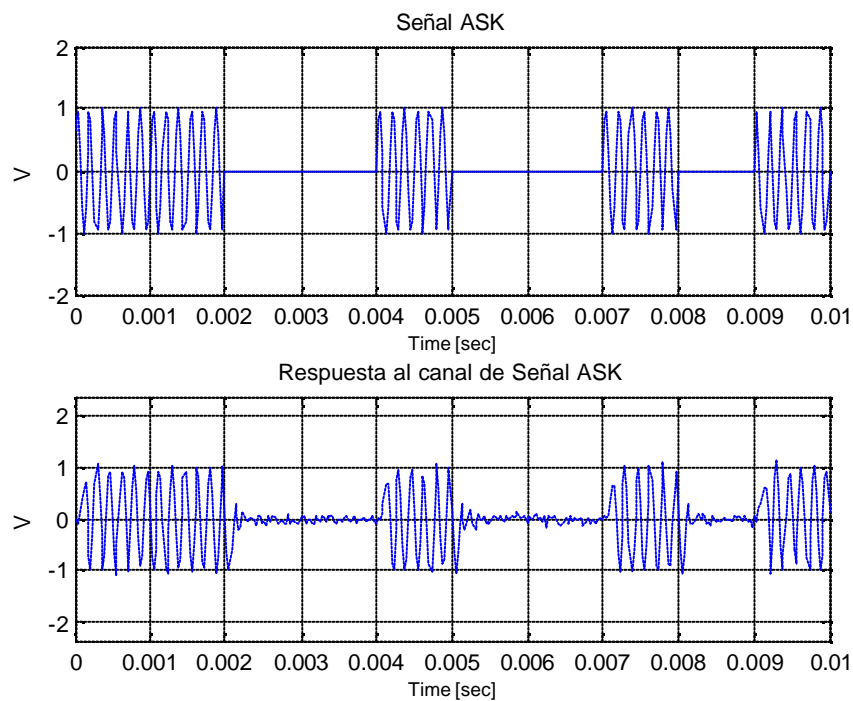


**Figura. 3.57. Señal ASK con Desviación de Frecuencia del Oscilador en Recepción**

7. Finalmente para comprender mejor el desempeño del sistema pasabanda frente al ruido, se genera una señal binaria de 600 datos, siendo los primeros bits [1 1 0 0 1 0], la cual se modula como ASK usando la función **mixer**.
8. Esto se introduce en un canal con ganancia 1, ruido AWGN= 1 W, y suficiente ancho de banda para que no se introduzcan distorsiones. En la figura 3.59 se muestran los primeros diez datos de la señal transmitida antes y después de pasar por el canal.



**Figura. 3.58. Detección No Coherente de Señal ASK**



**Figura. 3.59. Respuesta al Canal para Señal ASK**

9. Se usa un detector Coherente para demodular la señal obtenida después de pasar por el canal. En la figura 3.60 se muestran este resultado con el diagrama de ojo correspondiente a la salida del filtro casado.

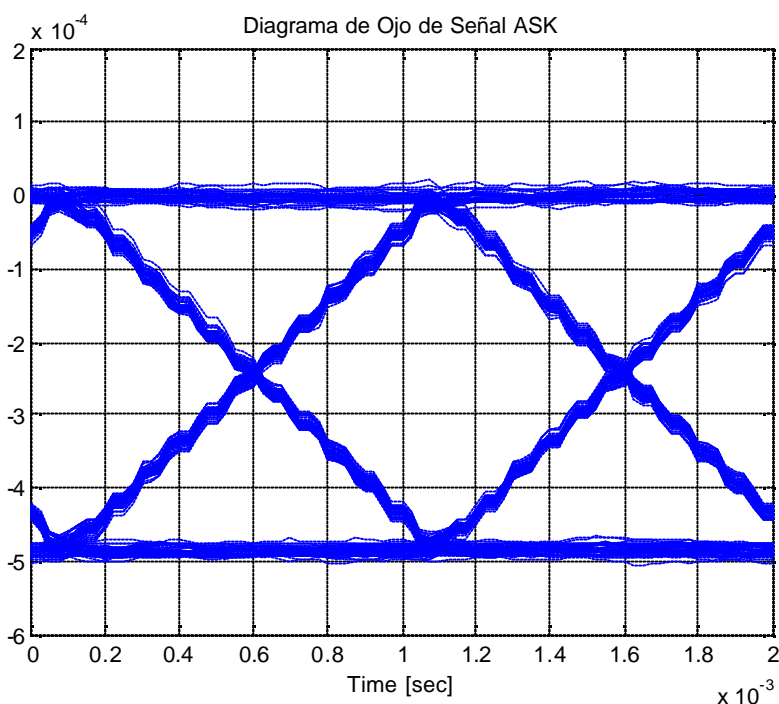


Figura. 3.60. Diagrama de Ojo para Señal ASK

### 3.3.2.3. Conclusiones

- La detección No Coherente de señales, no se requiere sincronización del oscilador en recepción con el de transmisión. Sin embargo, un demodulador de este tipo presenta mayores BER comparado con un sistema de detección Coherente.
- La salida del filtro casado entrega una señal atenuada 33 dB (2000 veces), la cual ya no conserva la forma de los pulsos de la secuencia binaria.
- La mayor dificultad en la implementación de detectores Coherentes es lograr la sincronización de la portadora. Como se puede ver en las gráficas de error de fase, la señal se va degradando de tal forma que para  $120^\circ$  se tiene una señal con datos totalmente distintos a la secuencia enviada.
- La desviación de frecuencias afecta también severamente a las señales en la detección, mostrando formas de onda muy diferentes a la secuencia de recibida. Si es pequeña la desviación (20Hz), aumenta gradualmente la atenuación de la señal recibida hasta llegar a perderla.



- Los efectos de un canal sobre una señal producen distorsiones de la forma de onda que los atraviesa, pero si el AWGN presente es bajo, estas alteraciones son mínimas y ello se comprueba con el diagrama de ojo. Como se observa, para un ruido de 4 mW el diagrama de ojo es bastante bueno. Se debe contar con suficiente ancho de banda, caso contrario el diagrama de ojo es sumamente malo y por tanto la alteración de la señal es muy alta.

### Codigo Coh\_NonCo.m

```

% Detección Coherente y No Coherente
% Efectos de Desplazamiento de Fase
% Efecto de Diferentes Frecuencia de Oscilador de Recuperación
% Respuesta al Canal

clc;
clear all;

% *****

SAMPLING_CONSTANT = 40;
BINARY_DATA_RATE = 1000;
SAMPLING_FREQ = BINARY_DATA_RATE * SAMPLING_CONSTANT;

global SAMPLING_CONSTANT;

% *****

b=[1 1 0 0 1 0 binary(54)];

unr=wave_gen(b,'unipolar_nrz',1000);
ask=mixer(unr, osc(6000));

pnr=wave_gen(b,'polar_nrz',1000);
psk=mixer(pnr, osc(6000));

fsk= vco (pnr);

c_ask=mixer(ask, osc(6000));

tt=[1:400];

figure(1)
subplot(2,1,1), waveplot(c_ask(tt))
subplot(2,1,2), psd(c_ask(tt))

c_psk=mixer(psk, osc(6000));

tt=[1:240];

figure(2)
subplot(2,1,1), waveplot(c_psk(tt))
subplot(2,1,2), psd(c_psk(tt))

```

```

ask_mat=match('unipolar_nrz',c_ask);
psk_mat=match('polar_nrz',c_psk);

figure(3)
subplot(2,1,1), waveplot(ask_mat(1:400))
subplot(2,1,2), waveplot(psk_mat(1:240))

c_ask1=mixer(ask, osc(6000,30));
c_ask2=mixer(ask, osc(6000,60));
c_ask3=mixer(ask, osc(6000,120));

ask_mat1=match('unipolar_nrz',c_ask1);
ask_mat2=match('unipolar_nrz',c_ask2);
ask_mat3=match('unipolar_nrz',c_ask3);

c_psk1=mixer(psk, osc(6000,30));
c_psk2=mixer(psk, osc(6000,60));
c_psk3=mixer(psk, osc(6000,120));

psk_mat1=match('polar_nrz',c_psk1);
psk_mat2=match('polar_nrz',c_psk2);
psk_mat3=match('polar_nrz',c_psk3);

tt=[1:400];

figure(4)
subplot(3,1,1), waveplot(ask_mat1(tt))
subplot(3,1,2), waveplot(ask_mat2(tt))
subplot(3,1,3), waveplot(ask_mat3(tt))

tt=[1:240];

figure(5)
subplot(3,1,1), waveplot(psk_mat1(tt))
subplot(3,1,2), waveplot(psk_mat2(tt))
subplot(3,1,3), waveplot(psk_mat3(tt))

ask_dev1=match('unipolar_nrz',mixer(ask, osc(5800)));
ask_dev2=match('unipolar_nrz',mixer(ask, osc(5980)));

tt=[1:400];

figure(6)
subplot(3,1,1), waveplot(ask_mat(tt))
subplot(3,1,2), waveplot(ask_dev1(tt))
subplot(3,1,3), waveplot(ask_dev2(tt))

nc_ask=envelope(ask,3000);

figure(7)
subplot(2,1,1), waveplot(ask(tt))
subplot(2,1,2), waveplot(nc_ask(tt))
clc;

% *****

bn=[1 1 0 0 1 0 binary(594)];

unrn=wave_gen(bn,'unipolar_nrz',1000);
askn=mixer(unrn, osc(6000));

```

```
ch_ask=channel(askn,1,0.004,10000);

tt=[1:400];

figure(8)
subplot(2,1,1), waveplot(askn(tt))
subplot(2,1,2), waveplot(ch_ask(tt))

ask_zm=match('unipolar_nrz', mixer(ch_ask, osc(6000)));

figure(9)
eye_diag(ask_zm);

clc;
```

### 3.4. REFERENCIAS

#### 3.4.1. LIBROS

- SKLAR, Bernard, **Digital Communications**, Segunda Edición, Prentice Hall, Estados Unidos 2001, páginas 169-234.
- STREMLER, Ferrel, **Sistemas de Comunicación**, Segunda Edición, Editorial AlfaOmega, México 1989, páginas 567-611.
- TOMASI, Wayne, **Sistemas de Comunicaciones Electrónicas**, Segunda Edición, Prentice Hall, México 1996, páginas 457-504.
- COUCH, Leon, **Sistemas de Comunicación Digitales y Analógicos**, Quinta Edición, Prentice Hall, México 1998, páginas 332-356.
- ESPE, **Comunicación Digital**, apuntes de la asignatura, semestre septiembre 2004 – marzo 2005.

#### 3.4.2. INTERNET.

- <http://www.ingelec.uns.edu.ar/fsc2718/CIII-IV.pdf>, Modulación de Pulsos Digital.
- <http://bips.bi.ehu.es/~inma/psc/tema3.pdf>; QAM.

## CAPÍTULO IV

### CODIFICACIÓN DEL CANAL

#### 4.1. INTRODUCCIÓN.

El propósito de cualquier sistema de comunicaciones es transmitir información desde una **fuentes** a un **destino** mediante un **canal** de comunicaciones. Un ingeniero en telecomunicaciones usualmente tiene poco control sobre los tres componentes, su rol es diseñar transmisores y receptores que envían la salida de la fuente al destino con alta fidelidad, baja distorsión, alto rendimiento de la probabilidad de error entre otras.

En ocasiones se producen diferencias o errores entre las secuencias de datos enviadas a través de un canal y las secuencias de datos recibidas debidas a la existencia de ruido en el canal, interferencias, y fading. Por ello es necesario realizar una codificación a la entrada del canal, cuyo objetivo es que el receptor sea capaz de detectar y corregir los errores producidos en los datos durante su transmisión por el canal.

La codificación del canal consiste en introducir redundancia, de forma que sea posible detectar e incluso corregir los errores introducidos por el canal, para reconstruir la secuencia de datos original de la forma más fiable posible.

Con el uso de circuitos de larga escala de integración LSI, y de procesadores digitales de señales DSP de alta velocidad, se ha logrado mejorar los sistemas hasta en 10 dB y con costos mucho menores que usando transmisión de alta potencia o antenas de alta ganancia.

La codificación de canal puede subdividirse en dos áreas de estudio:

**Codificación de forma de onda**, la cual busca la transformación de las formas de onda en otras mejores de manera que el proceso de detección este sujeto a menores errores.

**Secuencias estructuradas**, que buscan transformar las secuencias de datos en otras secuencias mejores que tengan redundancia estructurada de bits (insertan bits adicionales para detectar y/o corregir errores).

Las secuencias estructuradas a su vez emplean dos técnicas de corrección de errores: Solicitud de Repetición Automática – ARQ, Corrección de Errores de Transmisión (Anticipada) – FEC.

La elección entre usar una técnica ARQ o FEC depende de la situación en particular. Con frecuencia se utiliza la ARQ en sistemas de comunicación por computadora por su bajo costo de implementación y posee un canal dúplex de dos vías, de modo q el receptor puede transmitir una confirmación ACK de conformidad con los datos recibidos o un NAK si se reciben errores. Las FEC en cambio utilizan un canal símplex donde no hay regreso de ACK o NAK y es más útil en sistemas con largas demoras en la transmisión.

Estos a su vez se dividen en códigos de bloques, convolucionales, entrelazados, concatenado y Turbo códigos, que se estudian en este capítulo, así como la forma de ser decodificados. Es decir, se incluye una revisión de las diferentes técnicas y algoritmos para la codificación de canal y para la decodificación, con sus principales características a fin de facilitar su comprensión.

Al final se incluyen las prácticas de laboratorio correspondientes al capítulo con su respectivo desarrollo y resultados.

## 4.2. MARCO TEÓRICO

### 4.2.1. CODIFICACIÓN DE FORMA DE ONDA

Debido a que el proceso de codificación busca proporcionar la señal codificada con las mejores distancias entre vectores. Dentro de este grupo se encuentran las técnicas de:

1. Códigos ortogonales
2. Códigos biortogonales
3. Códigos trans-ortogonales

Para poder entenderlos de mejor forma, se describe previamente que son las señales antipodales y ortogonales y su correlación.

#### 4.2.1.1. Señales Antipodales y Ortogonales

Dos señales son antipodales cuando se encuentran desfasadas 180°, es decir, la una es la negada de la otra, así:

$$\begin{aligned} s_1(t) &= \text{Sin } \omega_o t \\ s_2(t) &= -\text{Sin } \omega_o t \end{aligned} \quad 0 \leq t \leq T$$

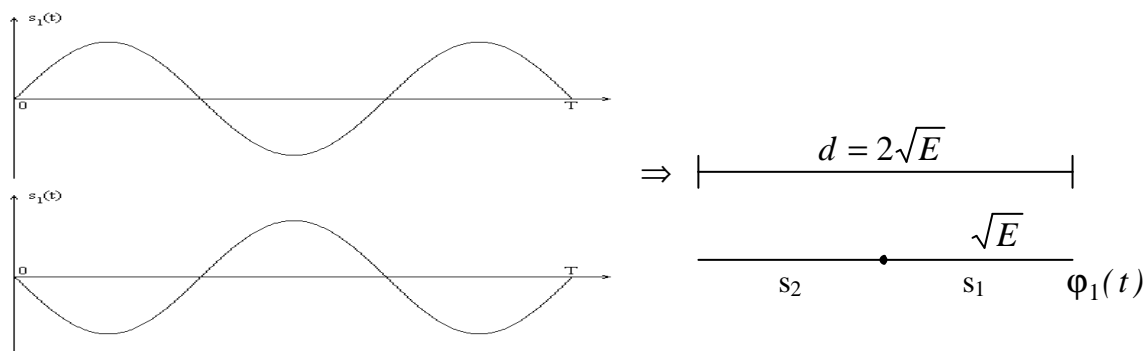


Figura. 4.1. Señales Antipodales

Como se puede observar, la distancia entre las señales es  $d = 2\sqrt{E}$ , que es la mayor que se puede obtener entre un par de señales cualesquiera.

Dos señales son ortogonales, en general cuando son perpendiculares entre si, y se definen en base al **coeficiente de correlación cruzada**  $Z_{ij}$ :

$$Z_{ij} = \frac{1}{E} \int_0^T s_i(t)s_j(t)dt = \begin{cases} 1 & i = j \\ 0 & \text{otro caso} \end{cases}$$

De esta forma, la correlación de dos señales es una expresión numérica que indica la similitud y de alguna forma la disposición de las mismas así:

$$\begin{cases} 1 & \rightarrow \text{sobrepuestas} \\ 0 & \rightarrow \text{ortogonales} \\ -1 & \rightarrow \text{antipodales} \end{cases}$$

La distancia en este tipo de señales, entre dos vectores cualesquiera es  $d = \sqrt{2E}$  y es considerada muy buena, para un nivel de energía dado, según el número de niveles que se utilicen.

#### 4.2.1.2. Codificación de Correlación Cruzada

Con la generación de un nuevo conjunto de formas de onda, provista por la codificación de forma de onda, se busca también mejorar el  $\mathbf{P}_B$ , y los códigos más comunes son los ortogonales y bi-ortogonales. El proceso de codificación pretende encontrar señales lo más distinta posibles y que  $Z_{ij}$  sea lo más pequeño posible.

Esto debido a que la correlación cruzada entre dos señales es una medida de la distancia entre sus vectores, pudiéndose llegar al mínimo valor (-1) solo en  $M= 2$  y señales antipodales, por lo que tratar de reducir su  $Z_{ij}$  al mínimo es acercarla a cero, formando con ello grupos de vectores de señalización **ortogonales**.



#### 4.2.1.2.1. Códigos Ortogonales.

Un conjunto de señales de datos de un bit puede ser transformado usando palabras de código ortogonales de dos bits cada una, como las filas de la matriz  $\mathbf{H}_1$ :

<i>DATOS</i>	<i>CÓDIGO ORTOGONAL</i>
0	$H_1 = \begin{Bmatrix} 0 & 0 \\ 0 & 1 \end{Bmatrix}$
1	

Para codificar conjuntos de datos de dos bits, se extiende el código anterior de forma horizontal y vertical, resultando:

<i>DATOS</i>	
0    0	$H_2 = \begin{Bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{Bmatrix} = \begin{Bmatrix} H_1 & H_1 \\ H_1 & H_1 \end{Bmatrix}$
0    1	
1    0	
1    1	

En general se puede construir un conjunto de palabras de código  $\mathbf{H}_k$ , conocida como matriz **Hadamard** para un conjunto de datos de k-bits con la matriz  $\mathbf{H}_{k-1}$  con:

$$H_k = \begin{Bmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & \overline{H_{k-1}} \end{Bmatrix}$$

Así, para datos de dos bits se codifican con cuatro bits, para datos de tres bits se codifican con ocho bits, y así sucesivamente.

Con este tipo de codificación en añadidura con detección coherente, se puede lograr el alto desempeño de  $P_B$  tal como con MFSK. Por lo que su límite superior de  $P_B$  y  $P_E$  serían:

$$P_E(k) \leq (2^k - 1) Q\left(\sqrt{\frac{kE_b}{N_o}}\right)$$

$$P_B(k) = (2^{k-1}) Q\left(\sqrt{\frac{kE_b}{N_o}}\right)$$

El número de dígitos redundantes que se insertan es igual a  $(2^k - k)$  y el ancho de banda requerido también aumenta en  $2^k / k$ .

#### 4.2.1.2.2. Códigos Bi – Ortogonales.

Un conjunto de palabras de código bi-ortogonales de M señales se puede obtener de un código ortogonal de M/2 señales aumentando la negación de cada palabra de código:

$$B_k = \left\{ \begin{array}{l} H_{k-1} \\ H_{k-1} \end{array} \right\}$$

El código biortogonal es una combinación de señales ortogonales (dentro de un conjunto) y antipodales (entre dos conjuntos) de forma que:

$$z_{ij} = \begin{cases} 1 & \text{para } i = j \\ -1 & \text{para } i \neq j, |i - j| = M/2 \\ 0 & \text{para } i \neq j, |i - j| \neq M/2 \end{cases}$$

Y debido a que usa la mitad de bits por palabra de código, ocupa la mitad de ancho de banda de los códigos ortogonales.

La probabilidad de error de palabras de código (símbolos) también es ligeramente mejor:

$$P_E(k) = (2^k - 2) Q\left(\sqrt{k \frac{E_b}{N_o}}\right) + Q\left(\sqrt{2k \frac{E_b}{N_o}}\right)$$

$$P_B(k) \approx \frac{P_E(k)}{2} \quad \text{para } k > 3$$

#### 4.2.1.2.3. Códigos Trans–Ortogonales.

Llamado también código **símplex**, se obtiene quitando el primer dígito de cada palabra ortogonal, quedando el código

$$Z_{ij} = \begin{cases} 1 & \text{para } i = j \\ -1/M - 1 & \text{para } i \neq j \end{cases}$$

Para este tipo de código el  $E_b/N_o$  es mínimo respecto a los ortogonales y bi-ortogonales para una tasa de errores determinada.

#### 4.2.2. ALGORITMOS FEC Y ARQ: CÓDIGOS DE BLOQUES, CONVOLUCIONALES, ENTRELAZADOS Y CONCATENADOS

Los algoritmos FEC (Forward Error Correction) y ARQ (Automatic Repeat Request) se utilizan para el control de errores, diferenciando con esto que existen sistemas que detectan y corrigen los errores y otros que solo los detectan.

- **Sistemas correctores de errores (FEC):** se implementan con códigos que, como su propio nombre indica, detectan y corrigen los errores, evitando, en principio, la retransmisión. Por este motivo, se utilizan, por ejemplo en sistemas símplex en los cuales no se puede enviar de vuelta un mensaje pidiendo una retransmisión. También se usan en procesos en tiempo real y en captura de datos. En este grupo estarán los códigos de bloques y convolucionales.
- **Sistemas detectores de errores (ARQ):** se utilizan solo para detectar si hay errores, y, en caso afirmativo, si es necesario (dependiendo del protocolo utilizado) solicitar una retransmisión. En este caso están por ejemplo los códigos de chequeo de paridad.

Aunque se usa a veces la corrección de errores, normalmente se prefiere la detección de errores ya que es más eficiente.

Por ejemplo, si se asume un canal con una tasa de errores de  $10^{-6}$  (es decir, un bit errado en cada  $10^6$ ) y que el mensaje tiene una longitud de 1000 bits y que para la corrección debemos añadir 10 bits por mensaje. De forma que, en la transmisión de  $10^6$  bits de datos se envían 10.000 bits de chequeo para detectar y corregir un bit de error esperado. Que esperamos. En cambio, para la detección se puede usar solamente un bit de paridad por mensaje. Para  $10^6$  bits de datos (mensajes de 1000 bits) se usarían solamente 1000 bits. Pero uno de los mensajes tiene un error, así que tendría que retransmitirse con su bit de paridad (1001 bits). En total usaría 2001 bits para este esquema.

#### **4.2.2.1. ARQ (Automatic Repeat Request)**

Existen tres clases de ARQ:

##### **4.2.2.1.1. ARQ de Parada y Espera (stop-wait ARQ).**

El transmisor envía un paquete y espera una respuesta del receptor para poder continuar. Si la transmisión es correcta recibe una señal de aceptación (ACK). Si un paquete tiene error, el receptor envía una señal NAK y se vuelve a transmitir el paquete errado antes de continuar con la secuencia. Su implementación requiere solo una conexión half-dúplex.

##### **4.2.2.1.2. ARQ Continuo con PullBack.**

Requiere una conexión full-dúplex, de forma que ambos terminales están transmitiendo simultáneamente. Este requiere números referenciales de los mensajes y saber los retardos de propagación. Cuando se detecta un error, los datos son transmitidos desde el dato errado hasta donde se encuentra la secuencia

#### 4.2.2.1.3. ARQ Continuo con Repetición Selectiva

También requiere una conexión full-dúplex y funciona de forma similar al anterior, pero en este caso solo el dato errado es retransmitido y vuelve a retomar la secuencia donde se había quedado.

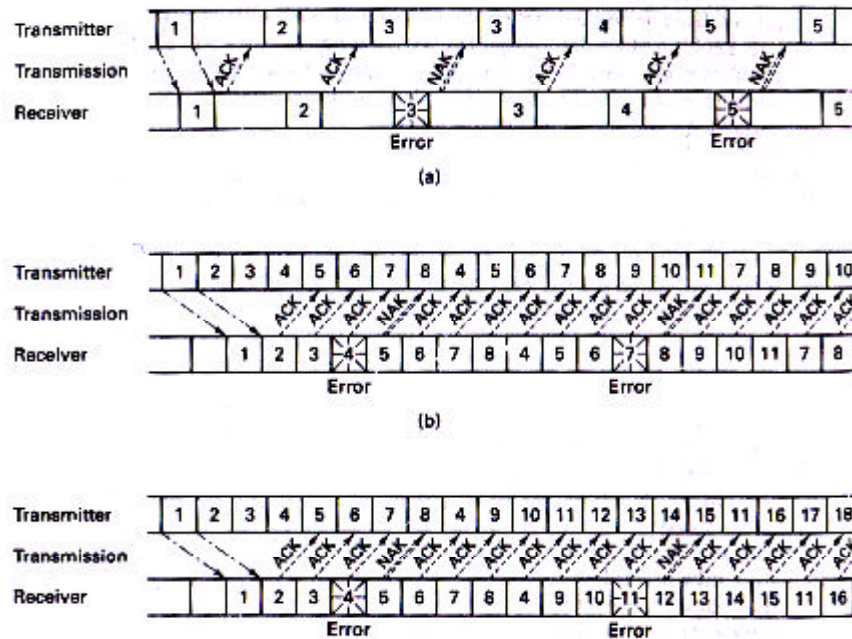


Figura. 4.2. ARQ, a) Parada-Espera, b) Continuo, c) Continuo Selectivo

ARQ tiene una ventaja en lo que es implementación sobre FEC, debido a que requiere mucho menos equipo y redundancia para la detección de errores. Además, ARQ es adaptativa ya que la información es retransmitida solo cuando ocurre un error.

Pero se puede preferir FEC si: no hay disponibilidad de canal de inverso, retardo excesivo de ARQ, el número de errores esperados requiera excesivas retransmisiones.

#### 4.2.2.2. Secuencias Estructuradas.

Como se vio anteriormente el uso de códigos ortogonales contaba con la gran desventaja de requerir un mayor ancho de banda a medida que crecía  $M$ , por lo que a continuación se revisan otro tipo de técnicas, que no concentran en si la señal es ortogonal o antipodal, conocidos como códigos de paridad.

Estos se clasifican en secuencias estructuradas, las cuales representan métodos de inserción de redundancia estructurada para la corrección y/o detección de errores.

#### 4.2.2.2.1. Códigos de Paridad.

Los códigos de detección de paridad usan sumas lineales de los bits de información, llamados símbolos de paridad o bits de paridad, para detección o corrección de errores.

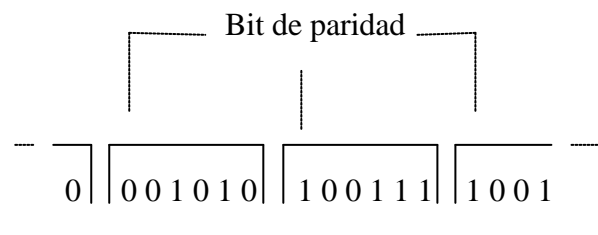
Existen dos tipos:

- Código de detección de paridad singular.
- Código de detección de paridad rectangular.

##### 4.2.2.2.1.1. Paridad Singular.

Un código de detección de paridad singular se construye añadiendo un bit de paridad a un bloque de bits de datos. El bit de paridad puede ser 1 ó 0 según sea necesario para asegurar que la suma de todos los bits sea par (o impar). Esta suma es realizada usando aritmética en módulo-2 (**ExOr**).

Se conoce como paridad par cuando el bit añadido permite que el resultado de la suma sea par, caso contrario se denomina paridad impar. En la figura 4.3 se ilustra una transmisión serial de datos, siendo el primer bit el de la derecha, y un bit de paridad singular es añadido para obtener paridad par, el último bit de la izquierda de cada bloque.



**Figura. 4.3. Código de Paridad Singular (Serial)**

En el receptor, el decodificador, comprueba que la suma módulo-2 de los datos transmitidos de como resultado 0 (paridad par). Si el resultado es 1 en vez de 0, el error será detectado.

La tasa de código puede ser expresado como  $k/(k+1)$ , siendo  $k$  el número de bits de datos por símbolo. Un error no detectado se obtiene cuando existe un número par de bits invertidos. La probabilidad de que  $j$  errores ocurran en un bloque de  $n$  símbolos es:

$$P(j, n) = \binom{n}{j} p^j (1-p)^{n-j}$$

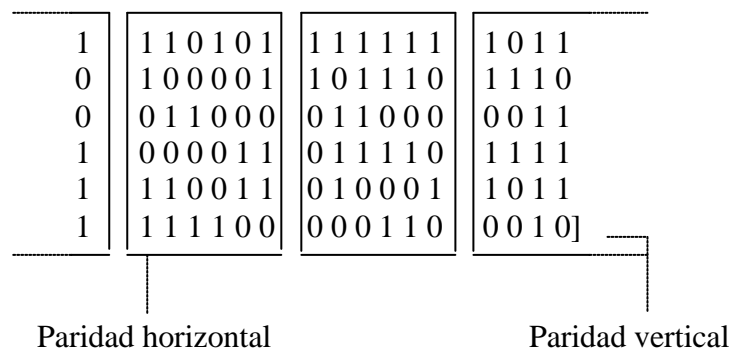
$$\binom{n}{j} = \frac{n!}{j!(n-j)!}$$

Donde  $p$  es la probabilidad de error de símbolo del canal, y la probabilidad de un error no detectado es en bloques de  $n$  bits es:

$$P_{nd} = \sum_{j=1}^{\substack{n/2(par) \\ (n-1/2)(impar)}} \binom{n}{2j} p^{2j} (1-p)^{n-2j}$$

#### 4.2.2.2.1.2. Paridad Rectangular.

Un código rectangular, también llamado un código producto, puede ser visto como una transmisión paralela de datos, como se muestra en la figura 4.4.



**Figura. 4.4. Código de Paridad Rectangular (Paralela)**

Primero se forma un rectángulo de bits de mensaje compuesto de **M filas y N columnas**, entonces a cada fila se aplica detección de paridad horizontal, y a cada columna detección de paridad vertical. Como resultado se logra un arreglo de dimensiones **(M + 1) x (N + 1)**. La tasa de código rectangular es:

$$\frac{k}{n} = \frac{MN}{(M+1)(N+1)}$$

El código rectangular es más eficiente que el singular pues permite una detección más exacta del error, localizando la fila y columna a las que pertenece.

La probabilidad de un error en el mensaje, también llamado error de bloque o error de palabra,  $P_M$ , está dado por:

$$P_M = \sum_{j=1}^n \binom{n}{j} p^j (1-p)^{n-j}$$

#### 4.2.2.3. Códigos de Bloques Lineales

Se toman **k** bits de datos, se le agregan **r** bits redundantes de manera de crear nuevos bloques de datos de longitud **n= k+r**. Así por ejemplo si **k= 3, r= 3** entonces **(n, k)= (6,3)**. Los bits de datos originales **m<sub>1</sub>, m<sub>2</sub>, m<sub>3</sub>** y los bits redundantes se construyen usando relaciones de suma módulo 2 entre los bits de mensaje. Por ejemplo.

m <sub>1</sub>	m <sub>2</sub>	m <sub>3</sub>	m <sub>1</sub> +m <sub>3</sub>	m <sub>1</sub> +m <sub>2</sub>	m <sub>2</sub> +m <sub>3</sub>
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	0	1	1
0	1	1	1	1	0
1	0	0	1	1	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	0	0	0

Tabla. 4.1. Código de Bloques Lineales (6,3)



Obsérvese que se utilizan solamente  $2^k = 8$  combinaciones de las  $2^n = 64$  posibles. Cuando el mensaje forma parte de la palabra código, se dice que este es **sistemático**.

Un código de bloque lineal sistemático será capaz de detectar  $X-1$  bits erróneos, donde  $X$  viene dado por la distancia de Hamming mínima entre 2 palabras cualesquiera del código, y será capaz de corregir  $(X-1)/2$  bits erróneos. Estos códigos cumplen la siguiente propiedad:

**La suma módulo-2 de dos palabras del código da lugar a otra palabra de código.**

Se requiere que las combinaciones enviadas estén lo más alejadas posible para disminuir la probabilidad de error, por otra parte hay que agregar pocos bits adicionales para que no crezca indefinidamente el ancho de banda necesario.

Para agilizar el proceso se utiliza una matriz generadora  $\mathbf{G}$ , a partir de la cual pueda generarse el mensaje codificado de la siguiente manera:

$$[\mathbf{m}][\mathbf{G}] = [\mathbf{C}]$$

Donde la matriz generadora está dada por:

$$[\mathbf{G}] = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1r} & 1 & 0 & \dots & 0 \\ p_{21} & p_{22} & \dots & p_{2r} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ p_{k1} & p_{k2} & \dots & p_{kr} & 0 & 0 & \dots & 1 \end{bmatrix}$$

La primera parte de la matriz  $[\mathbf{G}]$  será la porción de paridad del arreglo, y la última parte es la matriz identidad. Para el ejemplo de un código (6,3) la matriz generadora será:

$$[\mathbf{G}] = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Se observa que para generar los códigos solo hay que almacenar la porción de paridad ( $\mathbf{P}$ ), de la matriz  $[\mathbf{G}]$ .

Para realizar la decodificación, en destino se recibe un vector (**c**) de tamaño **n** y lo que se puede hacer es repetir la operación realizada en la codificación: se toman los primeros **k** bits y se calcula la redundancia usando la matriz generadora y se comprueba si la redundancia obtenida es igual a la redundancia recibida.

Otra opción más eficiente es la basada en el concepto de síndrome. En el proceso de decodificación basado en el síndrome se utiliza la matriz de chequeo de paridad, que se define como:

$$\mathbf{H} = [ \mathbf{I} \mid \mathbf{P}t ]$$

$$[H^T] = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 \\ p_{11} & p_{12} & \dots & p_{1r} \\ p_{21} & p_{22} & \dots & p_{2r} \\ \vdots & \vdots & \vdots & \vdots \\ p_{k1} & p_{k2} & \dots & p_{kr} \end{bmatrix}$$

**H** tiene la propiedad de que sólo las palabras de código verifican que al multiplicarlas por **H<sup>t</sup>** el resultado es el vector nulo. Esta propiedad será utilizada para la detección y corrección de errores.

A cada palabra que el receptor recibe a través del canal se denominada palabra recibida **r**. Una palabra recibida la podemos expresar como:

$$\mathbf{r} = \mathbf{c} + \mathbf{e}$$

Donde **c** es la palabra de código enviada por el emisor y **e** es una palabra de error. Cada componente **e<sub>i</sub>** de la palabra de error podrá valer 1 si hay un error en esa posición y 0 si no lo hay.

El receptor para realizar la codificación utiliza la matriz **H** para calcular el vector de síndrome de error a partir de la palabra recibida; el cual se obtiene de la siguiente forma:

$$[S] = [e][H^T]$$

El vector **S** permite crear un modelo de error en cualquier vector de código transmitido, y tiene tantos elementos como bits de paridad se estén usando. El vector de síndrome sólo depende de la secuencia de error y no de la palabra de código transmitida. Si en la transmisión no se ha producido un error, el síndrome es el vector nulo. Si se ha producido un error la multiplicación de la palabra recibida por  $H^t$  nos da un vector que es igual a una de las filas de  $H^t$ . La posición que ocupa esa fila es la posición donde hay un error.

La matriz [**H**] debe cumplir dos condiciones:

- a) Ninguna columna debe ser nula ya que el error en esa columna no sería detectado.
- b) Todas las columnas deben ser diferentes.

Todas estas operaciones se hacen en módulo-2 (sin acarreo).

**Ejemplo:** Se supone el vector de error  $[e] = [0\ 0\ 1\ 1\ 1\ 0]$ , entonces la relación  $[S] = [e][H^t]$  resulta ser:

$$[0\ 0\ 1\ 1\ 1\ 0] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = [1\ 0\ 0]$$

Entonces el síndrome es  $S = [1\ 0\ 0]$  y por tanto es posible hacer corresponder este resultado al error que se cometió.

#### 4.2.2.3.1. Fortaleza del Código.

Permite cuantificar la capacidad de corrección de errores de un código y consecuentemente determinar su fortaleza frente a los efectos del ruido. Para el efecto, es necesario definir los términos siguientes:

**Peso Hamming**, de un vector  $U = w(U)$ , es el número de “unos” de  $U$ .

**Distancia Hamming**, entre dos vectores  $U$  y  $V = d(U, V)$ . Es el número de elementos colocados en la misma posición en los cuales difieren entre sí los dos vectores. Debido a las propiedades de la suma módulo 2 puede demostrarse que:

$$w(U+V) = d(U, V)$$

por lo que

$$w(V) = d(V, 0)$$

La mínima distancia  $d_{\min}$  de un código define su fortaleza frente al ruido y como la suma de dos vectores de un subespacio dará otro elemento  $Z$  del subespacio, entonces sólo necesitamos ver el peso de cada vector y se elige el menor. La distancia mínima define la capacidad de detectar y corregir errores. En general un código  $(n, k)$  es capaz de detectar  $d_{\min}-1$  errores y corregir  $2^t$  patrones de error o  $t$  errores

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$$

Siendo  $t$  la capacidad de corrección de un código.

#### 4.2.2.3.2. Ganancia de Codificación.

La codificación no solamente tiene su efecto en la probabilidad de error final y en el ancho de banda sino también en la potencia transmitida por el sistema. A este parámetro se lo conoce como ganancia de codificación. En general podemos decir que:

**“Si se mantiene la potencia promedio de la señal codificada igual a la de la no codificada, la Energía por Bit tiene que bajar ya que cada bit codificado durará menos”.**

Para la detección y corrección de errores **simples** la matriz  $H$  debe cumplir:

- Todas las columnas de la matriz  $H$  deben ser diferentes. Esta condición hace que se pueda localizar la posición del error.
- Ninguna de las columnas de  $H$  pueden ser todas ceros. Esta condición se debe a que el síndrome es el vector nulo cuando no hay error.

Cuando se quieren corregir más de un error, por ejemplo **dos errores**, la matriz H debe verificar:

- Todas las columnas de la matriz H deben ser diferentes.
- Ninguna de las columnas de H pueden ser todas ceros.
- La suma de las columnas dos a dos debe ser diferente (Si la matriz H tiene 5 columnas hay 10 sumas diferentes que se pueden hacer. Lo que dice esta propiedad es que todas ellas deben dar resultados diferentes).

Esta última condición complica el cálculo de códigos correctores de dos bits y en la práctica no se suelen utilizar. Cuando hay más de un error se pedirá una repetición de la secuencia al emisor (vuelta atrás).

#### 4.2.2.4. Códigos de Hamming.

Un código de Hamming es un código de bloque lineal. Al igual que en los códigos de bloque lineales sistemáticos, podemos denotar un código de Hamming mediante un par  $(n, k)$ . El número de bits de paridad será  $m = n - k$ , pero deberá cumplirse la siguiente relación entre la longitud de la palabra de código y el número de bits de paridad:

$$n = 2^m - 1 \quad m \geq 3$$

Según esto también se cumplirá la siguiente relación entre el número de bits de datos y el número de bits de paridad:

$$k = 2^m - m - 1$$

Estos códigos tienen la propiedad de que la mínima distancia es **tres** independiente del valor asignado a  $m$ , lo que significa que los códigos Hamming pueden corregir 1 error.

Por lo tanto, a cada palabra original la añadiremos unos bits de paridad para obtener la palabra de código, de forma que estos bits de paridad sirvan posteriormente para encontrar y corregir errores que se produzcan en la transmisión. Cada uno de los bits de paridad que

añadimos a una palabra original va a afectar a unas determinadas posiciones de la nueva palabra de código, de forma que tomarán un valor adecuado para que se cumpla el criterio de paridad (par o impar) preestablecido en las subcombinaciones afectadas por cada uno de estos bits de paridad. El siguiente paso consistirá en determinar a que posiciones de los bits de las palabras de código afecta cada bit de paridad. Para ello construiremos todas las combinaciones posibles con  $m$  bits de paridad e interpretamos cada una en binario natural:

$b_m$ .....	$b_3$	$b_2$	$b_1$	Posición
0 .....	0	0	0	0
0 .....	0	0	1	1
0 .....	0	1	0	2
0 .....	0	1	1	3
.....				

Cada bit de paridad va a afectar a aquellas posiciones en las que ese bit vale 1.

Bit de paridad	Posiciones
$b_1$ .....	1,3,5,7,.....
$b_2$ .....	2,3,6,7,.....
.....	
$b_m$ .....	$2^m, 2^m+1, 2^m+2, \dots$

Por último sólo nos queda determinar que posiciones de cada palabra de código ocupará cada bit de paridad. Los bits de paridad han de colocarse en aquellas posiciones en las que no se vean afectados por otros bits de paridad. Estas posiciones serán:

Bit de paridad	Posición
$b_1$ .....	$2^0$
$b_2$ .....	$2^1$
$b_3$ .....	$2^2$
.....	

De esta forma queda completado el proceso de construcción de un código de Hamming.

En el proceso de decodificación, el receptor recibe una palabra de un código de Hamming, y deberá comprobar si es correcta o no, y en el caso de que no fuera correcta deberá comprobar en que bit se produjo el error y corregir ese error. Para comprobar si la palabra recibida es correcta, el receptor debe utilizar los bits de paridad de la palabra y hacer con ellos un control de paridad. Para realizar el control de paridad creamos una palabra que tendrá un bit por cada uno de los bits de paridad utilizados. Cada uno de los bits de esta palabra tomará el valor 0 o 1 dependiendo de si el número de unos de las posiciones de la palabra de código afectadas por su correspondiente bit de paridad cumplen o no el criterio de paridad establecido.

Interpretando la combinación resultante en binario natural tendremos dos posibilidades:

- Que sea un 0, lo cual quiere decir que no se han producido errores en la transmisión.
- Que se corresponda a un número distinto de 0, lo cual quiere decir que durante la transmisión ha variado el bit situado en la posición indicada por ese número.

Una vez obtenida la palabra de código correcta, basta con quitar los bits de paridad para obtener la palabra original enviada por el emisor.

#### 4.2.2.5. Códigos BCH (Bose-Chaudhuri-Hocquenghem)

Es una generalización del código Hamming, que permite múltiple corrección de error. Son una clase potente de los códigos cíclicos que proporcionan una amplia selección del largo de los bloques, tasas de código, tamaño de alfabeto y capacidad de corrección de error.

Los códigos BCH comúnmente usados, emplean un alfabeto binario y una longitud del bloque de palabras de código  $n = 2^m - 1$  donde  $m$  debe ser mayor o igual a 3.

El número de bits del mensaje es:  $k = n - mt$  y la distancia mínima  $d_{\min} = 2t + 1$ .

Cada código BCH es un código corrector de  $t$  errores; esta versatilidad de diseño y la existencia de un algoritmo eficiente de decodificación (Derlekamp – Massey), hace que estos sean particularmente atractivos. Una subclase especial de códigos BCH es el particularmente útil conjunto **no binario** llamado código **Reed-Solomon**.

$n$	$k$	$t$	$g(p)$
7	4	1	13
15	11	1	23
	7	2	721
	5	3	2467
31	26	1	45
	21	2	3551
	16	3	107657
	11	5	5423325
	6	7	313365047
63	57	1	103
	51	2	12471
	45	3	1701317
	39	4	166623567
	36	5	1033500423
	30	6	157464165547
	24	7	17323260404441
	18	10	1363026512351725
	16	11	6331141367235453
	10	13	472622305527250155
	7	15	5231045543503271737
127	120	1	211
	113	2	41567
	106	3	11554743
	99	4	3447023271
	92	5	624730022327
	85	6	130704476322273
	78	7	26230002166130115
	71	9	6255010713253127753
	64	10	1206534025570773100045
	57	11	335265252505705053517721
	50	13	54446512523314012421501421
	43	14	17721772213651227521220574343
	36	15	3146074666522075044764575721735
	29	21	403114461367670603667530141176155
	22	23	123376070404722522435445626637647043
15	27	22057042445604554770523013762217604353	
8	31	7047264052751030651476224271567733130217	
255	247	1	435
	239	2	267543
	231	3	156720665
	223	4	75626641375
	215	5	23157564726421
	207	6	16176560567636227
	199	7	7633031270420722341
	191	8	2663470176115333714567

Tabla. 4.2. Generadores de código BHC Primitivo.



#### 4.2.2.6. Códigos Cíclicos

Son un tipo de códigos de bloques lineales más fáciles de implementar. Un código lineal es llamado cíclico si cumple con:

- **Linealidad:** La suma de 2 palabras códigos es otra palabra código.
- **Desplazamiento Cíclico:** Cualquier desplazamiento cíclico de una palabra código es otra palabra código.

Estos contienen una cantidad razonable de estructuras matemáticas que permite diseñar códigos de corrección de órdenes mayores. Además se pueden implementar fácilmente utilizando simples registros de corrimiento. Los códigos cíclicos son de tal naturaleza que las palabras de código son simplemente corrimientos laterales de unas a otras.

Se utiliza la siguiente notación; si:  $\mathbf{c} = (c_1, c_2, \dots, c_n)$

Es un vector de un código  $\mathbf{C}$ , entonces  $\mathbf{c}^{(i)}$  representa a  $\mathbf{c}$  desplazada en forma cíclica  $i$  lugares hacia la izquierda, esto es  $\mathbf{c}^{(i)} = (c_{i+1}, c_{i+2}, \dots, c_n, c_1, c_2, \dots, c_i)$

Los códigos cíclicos se pueden escribir en forma de polinomio. El vector  $\mathbf{c}$  se puede expresar como un polinomio de grado  $(n-1)$

$$\mathbf{c}(\mathbf{x}) = c_1x^{n-1} + c_2x^{n-2} + \dots + c_n$$

Los coeficientes de los polinomios son ya sea 0 o 1, y obedecen las siguientes propiedades.

$$\begin{array}{ll} \mathbf{0} + \mathbf{0} = \mathbf{0} & \mathbf{0} \times \mathbf{0} = \mathbf{0} \\ \mathbf{0} + \mathbf{1} = \mathbf{1} + \mathbf{0} = \mathbf{1} & \mathbf{0} \times \mathbf{1} = \mathbf{1} \times \mathbf{0} = \mathbf{0} \\ \mathbf{1} + \mathbf{1} = \mathbf{0} & \mathbf{1} \times \mathbf{1} = \mathbf{1} \end{array}$$

El polinomio de código  $\mathbf{c}^{(i)}(\mathbf{x})$  para el vector de código  $\mathbf{c}^{(i)}$  es:

$$\mathbf{c}^{(i)}(\mathbf{x}) = c_{i+1}x^{n-1} + c_{i+2}x^{n-2} + \dots + c_n x^i + c_1 x^{i-1} + \dots + c_i$$

$$C^{(i)}(x) = x^i C(x) \bmod (x^n + 1)$$

Se probará ahora un teorema importante de los códigos cíclicos. Se dice que un polinomio de código cíclico  $c(x)$  se puede generar mediante el polinomio de datos  $d(x)$  de grado  $k-1$  y un polinomio generador  $g(x)$  de grado  $(n-k)$  como

$$\mathbf{c}(\mathbf{x}) = \mathbf{d}(\mathbf{x}) \mathbf{g}(\mathbf{x})$$

en donde el polinomio generador  $g(x)$  es un factor de orden  $(n-k)$  de  $(x^n + 1)$ .

Para un vector de datos  $(d_1, d_2, \dots, d_k)$ , el polinomio de datos es

$$\mathbf{d}(\mathbf{x}) = d_1 x^{k-1} + d_2 x^{k-2} + \dots + d_k$$

### Ejemplo

**Se determinará un polinomio generador  $g(x)$  para un código cíclico (7, 4), y se encontrará un vector de código para el siguiente vector de dato 1010.**

#### Solución:

En este caso  $n = 7$  y  $n - k = 3$

$$x^7 + 1 = (x + 1)(x^3 + x + 1)(x^3 + x^2 + 1)$$

Para un código (7, 4), el polinomio generador debe ser del orden  $n - k = 3$ . En este caso, existen dos elecciones posibles para  $g(x)$ . Elijamos  $x^3 + x^2 + 1$  como posible polinomio generador:

$$g(x) = x^3 + x^2 + 1$$

$$\mathbf{d} = [ \mathbf{1010} ]$$

$$\mathbf{d}(\mathbf{x}) = x^3 + x$$

Y el polinomio de código es

$$\begin{aligned} \mathbf{c(x)} &= \mathbf{d(x) g(x)} \\ &= (x^3 + x) (x^3 + x^2 + 1) \\ &= x^6 + x^5 + x^4 + x \end{aligned}$$

Por lo tanto

$$\mathbf{c} = (1110010)$$

Al observar la estructura de la palabra de código, los primeros  $k$  dígitos no necesariamente son los dígitos de los datos, por consiguiente, éste no es un código sistemático.

En un código sistemático, los primeros  $k$  dígitos son de datos, y los últimos  $m = n - k$  son los dígitos de comprobación de paridad.

#### 4.2.2.6.1. Códigos Cíclicos Sistemáticos.

Se demostrará que para un código sistemático, el polinomio de la palabra de código  $c(x)$  correspondiente al polinomio de datos  $d(x)$  es

$$c(x) = x^{n-k} d(x) + p(x)$$

En donde  $p(x)$  es el residuo de la división de  $x^{n-k} d(x)$  entre  $g(x)$ .

$$p(x) = \text{Res} \frac{x^{n-k} d(x)}{g(x)}$$

Para probar esto se observa que

$$\frac{x^{n-k} d(x)}{g(x)} = q(x) + \frac{p(x)}{g(x)}$$

En donde  $q(x)$  es de grado  $(k - 1)$  o menor. Se suma  $p(x) / g(x)$  a los dos miembros de la ecuación anterior y como  $f(x) + f(x) = 0$ , bajo la operación módulo 2, se tiene

$$q(x) g(x) = x^{n-k} d(x) + p(x)$$

**Ejemplo:** Partiendo de  $(x^n + 1)$

$$g(x) = x^3 + x^2 + 1$$

Y se considera un vector de datos  $d = (1010)$

$$d(x) = x^3 + x$$

$$x^{n-k} d(x) = x^6 + x^4$$

Por lo tanto

$$\begin{array}{r}
 x^6 + x^4 \\
 \underline{x^6 + x^5 + x^3} \\
 x^5 + x^4 + x^3 \\
 \underline{x^5 + x^4 + x^2} \\
 x^3 + x^2 \\
 \underline{x^3 + x^2 + 1} \\
 1 \quad p(x)
 \end{array}
 \begin{array}{r}
 \underline{x^3 + x^2 + 1} \\
 x^3 + x^2 + 1 \quad q(x)
 \end{array}$$

De donde,

$$\begin{aligned}
 c(x) &= q(x) g(x) \\
 &= (x^3 + x^2 + 1)(x^3 + x^2 + 1) \\
 &= x^6 + x^4 + 1
 \end{aligned}$$

Observe que  $d_{\min}$ , la mínima distancia entre dos palabras de código, es 3. Por lo tanto, éste código logra corregir un solo error.

#### 4.2.2.6.2. Códigos de Redundancia Cíclica – CRC.

Son códigos cíclicos usados para detectar errores. Los más típicos usan los siguientes polinomios generadores

código	Polinomio $g(X)$	$n-k$
CRC-12	$1+X+X^2+X^3+X^{11}+X^{12}$	12
CRC-16	$1+X^2+X^{15}+X^{16}$	16
CRC-CCITT	$1+X^5+X^{12}+X^{16}$	16

**Tabla. 4.3. Polinomios Generadores CRC**

Estos códigos todos contienen como factor primo  $(1+X)$ . El CRC-12 se usa para caracteres de 6 bits; los otros dos son usados para caracteres de 8 bits.

Los Códigos CRC binarios pueden detectar los siguientes patrones de error:

- Ráfagas de longitud  $n-k$  o menores.
- Una fracción de ráfagas de longitud igual a  $n-k+1$ ; la fracción igual a  $1-2^{-(n-k-1)}$ .
- Una fracción de ráfagas de longitud mayor que  $n-k+1$ ; la fracción igual a  $1-2^{-(n-k-1)}$ .
- Todas las combinaciones de  $d_{\min}-1$  (o menos) errores.
- Todos los patrones de error con un número impar de errores si el polinomio  $g(x)$  tiene un número par de coeficientes distintos de cero.

#### 4.2.2.6.3. Generación de Códigos Cíclicos.

La ventaja de los códigos cíclicos es que la codificación y la decodificación pueden ser implementadas con elementos tan sencillos como registros de desplazamiento y sumadores módulo 2.

Es la simple implementación práctica de los polinomios generadores, que como ya se pudo ver, tienen una forma definida en la que  $g_k$  son los pesos o ganancias entre 0 o 1. Este tipo de circuitos generadores lo que hacen es mandar primero los  $k$  datos originales y luego envían en cola los datos de redundancia cíclica.

#### 4.2.2.6.4. Decodificación de Códigos Cíclicos.

Todo polinomio de código  $c(x)$  válido es un múltiplo de  $g(x)$ . Si ocurre un error durante la transmisión, el polinomio de palabra recibido  $r(x)$  no será múltiplo de  $g(x)$ .

En estos cálculos aparece el polinomio  $s(x)$  que se conoce como polinomio de síndrome, y siempre tiene un grado  $n-k-1$  o menor. Así:

$$\frac{r(x)}{g(x)} = ml(x) + \frac{s(x)}{g(x)}$$

y

$$s(x) = \text{res} \left[ \frac{r(x)}{g(x)} \right]$$

Si  $e(x)$  es el polinomio de error entonces:

$$r(x) = c(x) + e(x)$$

Si se recuerda que  $c(x)$  es múltiplo de  $g(x)$  entonces.

$$s(x) = \text{Res} \left[ \frac{e(x)}{g(x)} \right]$$

A continuación se muestran los pasos para el proceso de la decodificación:

- Primero se construye la tabla de decodificación considerando el error y el síndrome asociado. Para esto, se divide el polinomio generador para el polinomio de error, obteniéndose el polinomio de síndrome debido al error introducido.
- Ahora se calcula el síndrome del dato específico dividiendo el dato recibido  $r(x)$  para el polinomio generador  $g(x)$ . El residuo de éste será el  $s(x)$  particular para el dato  $r(x)$ .
- Con el  $s(x)$  del paso anterior, se va a la tabla de decodificación y se identifica el error. Si consideramos que con el error correspondiente y el dato recibido podemos

hallar la palabra de código por medio de  $c = e \oplus r$  entonces podemos ir a la tabla en el transmisor y saber cual fue el dato transmitido.

#### 4.2.2.7. Códigos Entrelazados y Concatenados

En las técnicas descritas anteriormente hemos asumido que los canales no tienen memoria. Un canal con memoria es aquel que exhibe deterioro de la señal de forma dependiente de los errores anteriores, por tanto, en un canal con memoria, los errores tienden a ocurrir en ráfagas (bloques de bits ó símbolos). Un ejemplo de aquello es las huellas y rayaduras de la superficie de un disco compacto.

En un canal con memoria usando códigos de bloques ó convolucionales sin previamente entrelazar y luego concatenarlos, el rendimiento del sistema se vuelve pobre, debido a que se afectan en un solo bloque los bits de datos como los bits de paridad. Mal puede el receptor decidir si la información llegó bien si también es probable que los datos de comprobación también estén probablemente errados.

El entrelazamiento es una técnica para disimular la memoria del canal haciendo que los errores en ráfaga se dispersen en el tiempo y estos se convierten en errores “aleatorios”, que, con los códigos analizados anteriormente, se pueden detectar y corregirlos. El proceso consiste en reordenar los símbolos antes de la transmisión, para que el receptor los vuelva a su orden.

En la figura se observa que si en el canal se dañan tres símbolos consecutivos, con el uso de un entrelazador en el transmisor y un des-entrelazador en el receptor, los errores no se quedan como consecutivos.

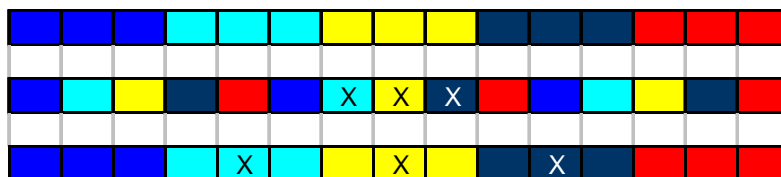


Figura. 4.5. Código Entrelazado

Luego de entrelazar los datos entonces será necesario concatenar dos códigos de tal manera que uno de ellos logre control de errores a nivel de bits y el otro control de errores a nivel de símbolos. Un ejemplo práctico de aquello resulta ser la utilización de un código convolucional concatenado con un código Reed-Solomon en los sistemas de almacenamiento de información en discos compactos.

Un código concatenado es aquel que usa dos niveles de codificación, una interna y una externa para lograr el desempeño de error deseado. En la figura 4.6 se muestra el orden de la codificación y decodificación.

La codificación interna, la que se ve con el modulador / demodulador y el canal es usualmente configurada para corregir los errores del canal. La codificación externa posee por lo general una tasa de código alta (baja redundancia) para reducir la probabilidad de error a un nivel específico.

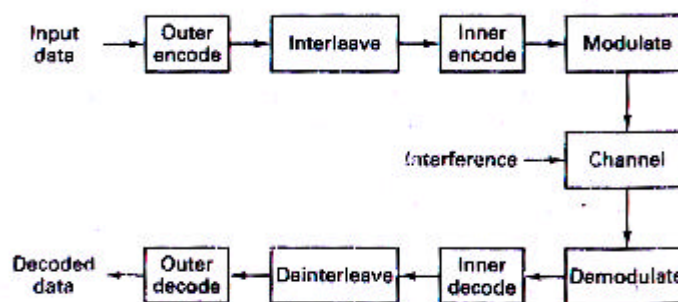


Figura. 4.6. Esquema código Concatenado

Existe un entrelazador entre las dos etapas de codificación. Esto es requerido para ensanchar cualquier ráfaga de errores que pueda aparecer a la salida del codificador interno.

Uno de los códigos concatenados más utilizados emplea un código convolucional como interno y un Reed – Solomon como código externo con entrelazado entre ellos.



#### 4.2.2.8. Códigos Convolutionales

Se diferencian de los códigos de bloque en su forma estructural y las propiedades para corregir errores. Los códigos de bloque suelen tener limitada la capacidad de corrección de errores alrededor de 1 o 2 símbolos erróneos por palabra de código. Estos códigos son buenos para utilizar en canales con baja probabilidad de error. Los códigos convolutionales en cambio son adecuados para usar sobre canales con mucho ruido (alta probabilidad de error).

Los códigos convolutionales son códigos lineales, donde la suma de dos palabras de código cualesquiera también es una palabra de código. Y al contrario que con los códigos lineales, se prefieren los códigos no sistemáticos.

El sistema tiene memoria: la codificación actual depende de los datos que se envían ahora y que se enviaron en el pasado.

Un código convolucional queda especificado por tres parámetros ( $n$ ,  $k$ ,  $m$ ):

$n$  es el número de bits de la palabra codificada.

$k$  es el número de bits de la palabra de datos.

$m$  es la memoria del código o longitud restringida.

#### Ejemplos:

##### **Código (2, 1, 3)**

La palabra codificada tiene 2 bits de longitud.

La entrada son bloques de 1 bit.

La salida depende de los dos bloques anteriores y del actual.

##### **Código (4, 2, 3)**

La palabra codificada tiene 4 bits de longitud.

La entrada son bloques de 2 bit.

La salida depende de los dos bloques anteriores y del actual.

#### 4.2.2.8.1. Codificación.

El proceso de codificación de estos códigos se realiza utilizando un dispositivo lógico en el codificador.

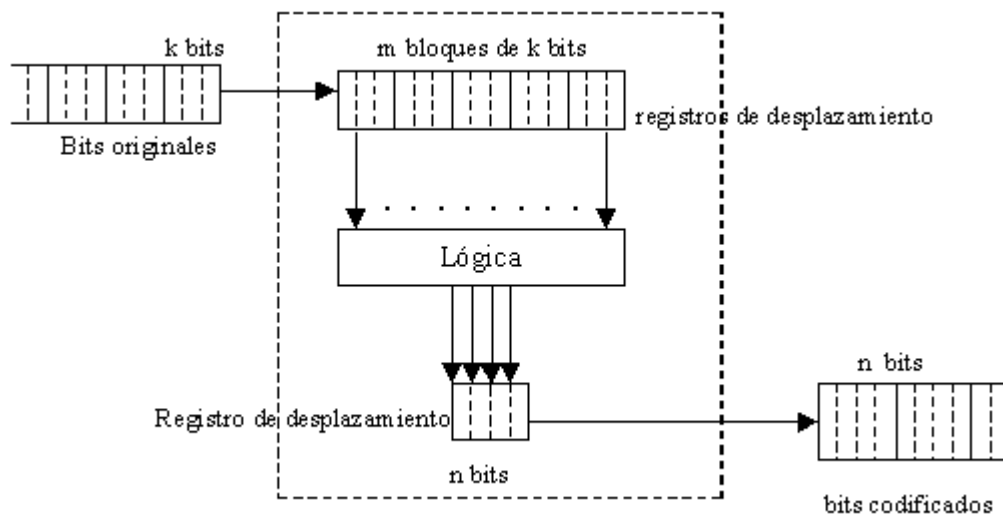


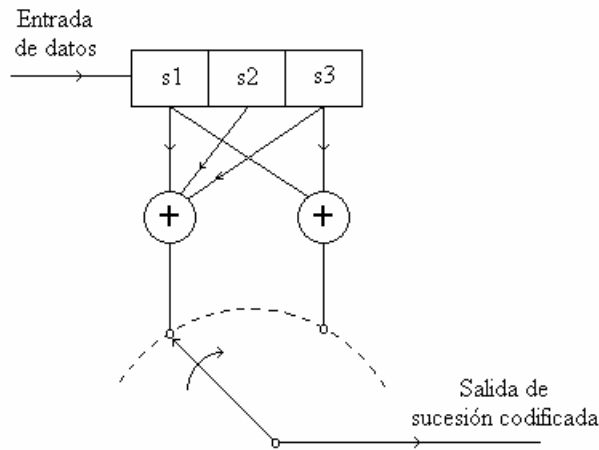
Figura. 4.7. Codificador Convolucional (4, 3, 5)

La palabra codificada se obtendría como el resultado de realizar una serie de operaciones lógicas entre determinados bits que están almacenados en los registros intermedios.

En un código de convolución el bloque de  $n$  dígitos de código generado por el codificador en una unidad de tiempo particular depende no sólo del bloque de  $k$  dígitos de mensaje, sino también del bloque de dígitos de datos dentro del lapso previo de  $N-1$  unidades de tiempo ( $N > 1$ ). Para los códigos de convolución  $k$  y  $n$  son generalmente pequeñas, además se pueden diseñar para la corrección de errores aleatorios, ráfaga de errores, o ambos. La codificación se implementa fácilmente mediante registros de corrimiento.

Un codificador de convolución con restricción de longitud  $N$  consta de un registro de corrimiento de  $N$  etapas y  $O$  sumadores módulo 2. En la figura 4.6 los dígitos de mensaje se aplican a la entrada del registro de corrimiento. La corriente de dígitos codificados se obtiene a la salida del conmutador. El conmutador muestrea los  $v$  sumadores de módulo 2 en secuencia, una vez durante cada intervalo de bit de entrada.

Hay en total  $n = (N+k)O$  dígitos en la salida codificada por cada  $k$  dígitos de datos. en la práctica,  $k \gg N$ , y, en consecuencia, hay aproximadamente  $k\nu$  dígitos codificados de salida por cada  $k$  dígitos de datos, dando una eficiencia  $h \approx 1/O$



**Figura. 4.8. Codificador Convolucional  $N= 3, \nu= 2$**

El conmutador con las dos entradas hace el papel de un registro de desplazamiento de dos estados. El código convolucional es generado introduciendo un bit de datos y dando una revolución completa al conmutador. Inicialmente se supone que los registros intermedios contienen ceros.

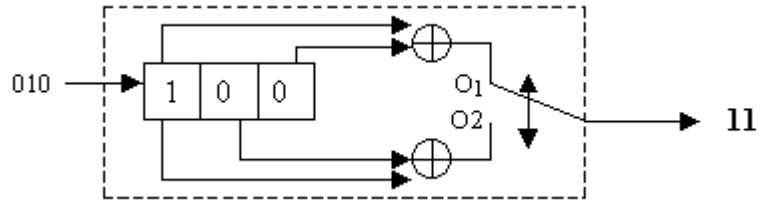
Para un codificador (2, 1, 3) como el de la figura 4.8, las secuencias de salida son:

Entrada (S3,S2,S1)	Salida (O1,O2)
000	00
001	11
010	01
011	10
100	10
101	01
110	11
111	00

**Tabla. 4.4. Secuencias de Salida para Codificador (2, 1, 3)**

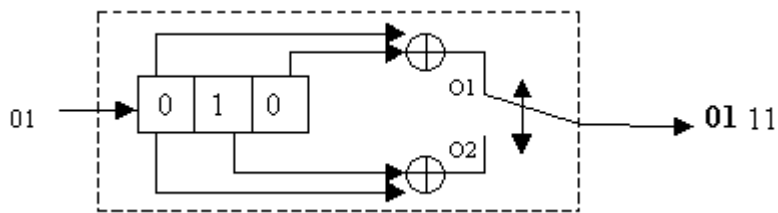
Como ejemplo del funcionamiento de este codificador, suponer que se quiere enviar la secuencia de bits 0101 (donde los bits más a la derecha son los más antiguos). El proceso de codificación es el siguiente:

- Se introduce el primer bit de la secuencia en el codificador:



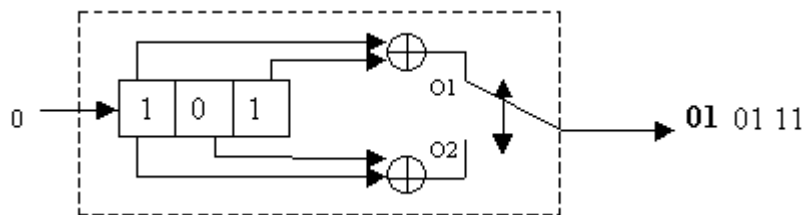
**Figura. 4.9a. Codificador Convolucional (2, 1, 3) secuencia 0101**

- Se introduce el segundo bit de la secuencia en el codificador:



**Figura. 4.9b. Codificador Convolucional (2, 1, 3) secuencia 0101**

- Se introduce el tercer bit de la secuencia en el codificador:



**Figura. 4.9c. Codificador Convolucional (2, 1, 3) secuencia 0101**

- Se introduce el cuarto bit de la secuencia en el codificador:

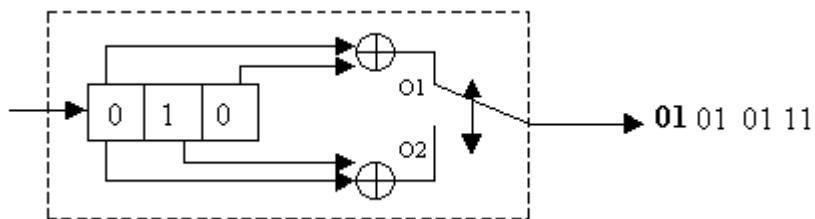


Figura. 4.9d. Codificador Convolucional (2, 1, 3) secuencia 0101

Al final del proceso de codificación obtenemos que la secuencia codificada es 01 01 01 11. Debido a la memoria del código es necesario de disponer de medios adecuados para determinar la salida asociada a una determinada entrada.

Hay tres métodos gráficos:

- **Diagrama de estados:** es la forma menos utilizada.
- **Diagrama árbol o árbol del código:** representa mediante un árbol binario las distintas posibilidades.
- **Diagrama de Trellis o enrejado:** es la forma más utilizada porque es la que permite realizar la decodificación de la forma más sencilla.

#### 4.2.2.8.1.1. Diagrama de Estados.

En la representación por diagrama de estados, el estado de un codificador se define como los contenidos de los registros de las  $k-1$  etapas, habiendo por tanto en total  $2^{k-1}$  estados.

Para un codificador (2, 1, 2) y una secuencia de entrada 101, los estados estarán representados por 2 registros, cuyos valores posibles serán 00, 01, 10, 11. La transición de estados toma en cuenta el valor de los registros en la transición del estado actual y el próximo estado en la ejecución de un ciclo de reloj.

De esta manera, si llamamos a los estados como  $a=00$   $b=10$   $c=01$   $d=11$ , existirán transiciones de estados con las salidas respectivas como se muestra en la figura 4.10. Las líneas sólidas indican una transición debida a la entrada de un 0 y las líneas punteadas indican una transición por un 1.

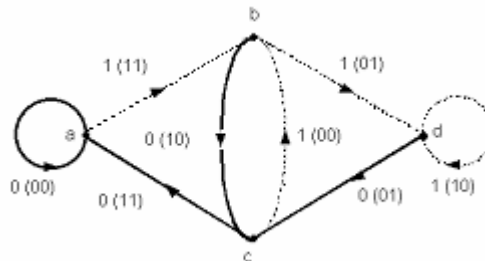


Figura. 4.10. Diagrama de Estados Codificador (2, 1, 2) secuencia 101

#### 4.2.2.8.1.2. Árbol de código.

Para el ejemplo del codificador (2, 1, 3) anteriormente especificado, cuyas secuencias de salida son las de la tabla 4.4, se cuenta con el Árbol del código de la figura 4.10.

La profundidad del árbol es  $2(m-1)$ , y el número de estados es  $2^{(m-1)k}$ . La interpretación del árbol del código es la siguiente:

- Hay dos ramas en cada nodo. La rama superior corresponde a una entrada de un 0. La rama inferior corresponde a la entrada de un 1.
- En la parte exterior de cada rama se muestra el valor de salida.
- El número de ramas se va multiplicando por dos con cada nueva entrada.
- A partir del segundo nivel el árbol se vuelve repetitivo. En realidad, solo hay cuatro tipos e nodos: **A,B,C,D**. Estos tipos de nodos en realidad son estados del codificador. A partir de estos nodos, se producen los mismos bits de salida y el mismo estado. Por ejemplo, de cualquier nodo etiquetado como **C** se producen el mismo par de ramas de salida:

- Salida 10 y estado A.

- Salida 01 y estado B.

#### **4.2.2.8.1.3. Diagrama de Trellis.**

A partir de la identificación de los estados del codificador se puede incorporar esta información en el diagrama de Trellis, el cual 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.

Partiendo 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.

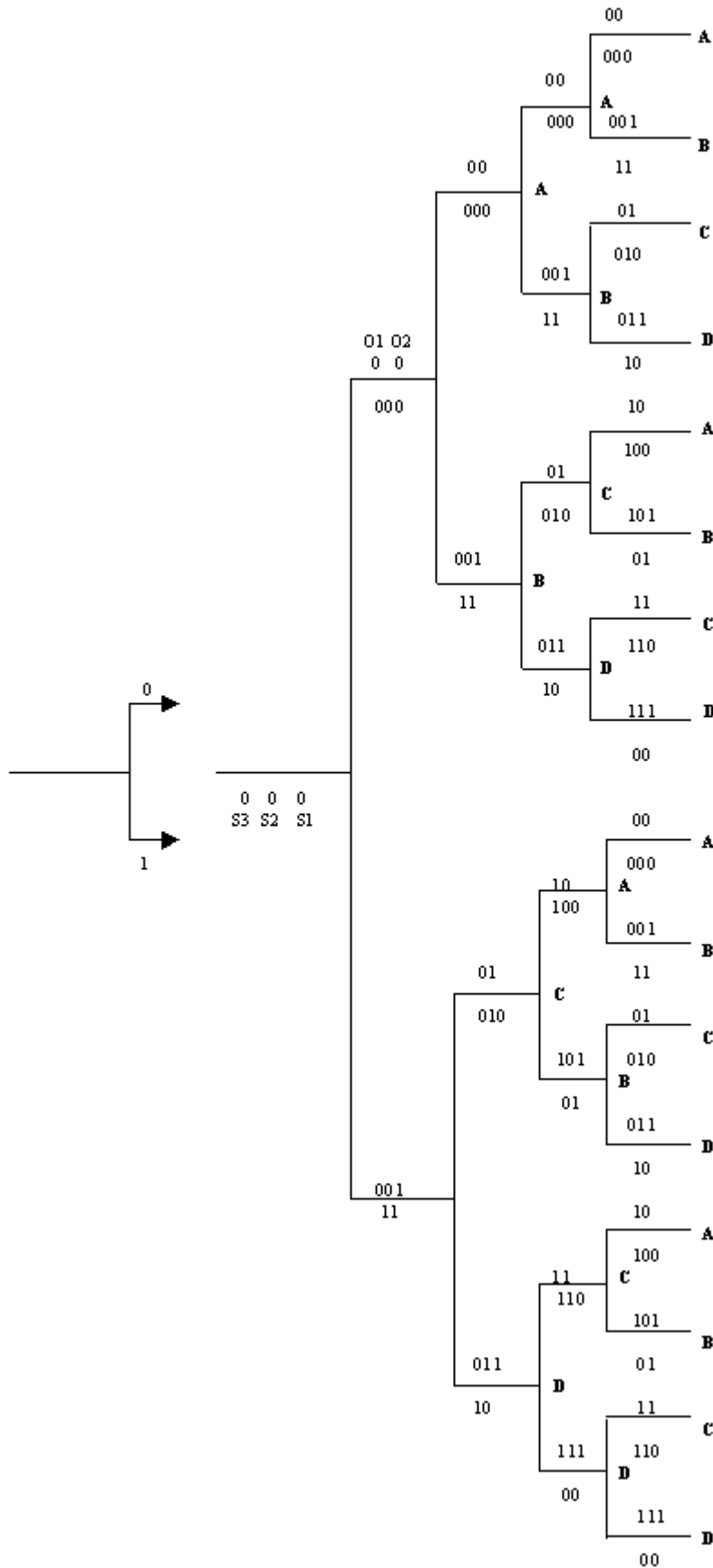


Figura. 4.11. Árbol de código para Codificador (2, 1, 3) secuencia 0101



Para cada nivel del árbol se hace lo mismo desde todos los estados en los que el codificador se puede encontrar.

Según todo esto, el diagrama de Trellis para el codificador (2,1,3) :

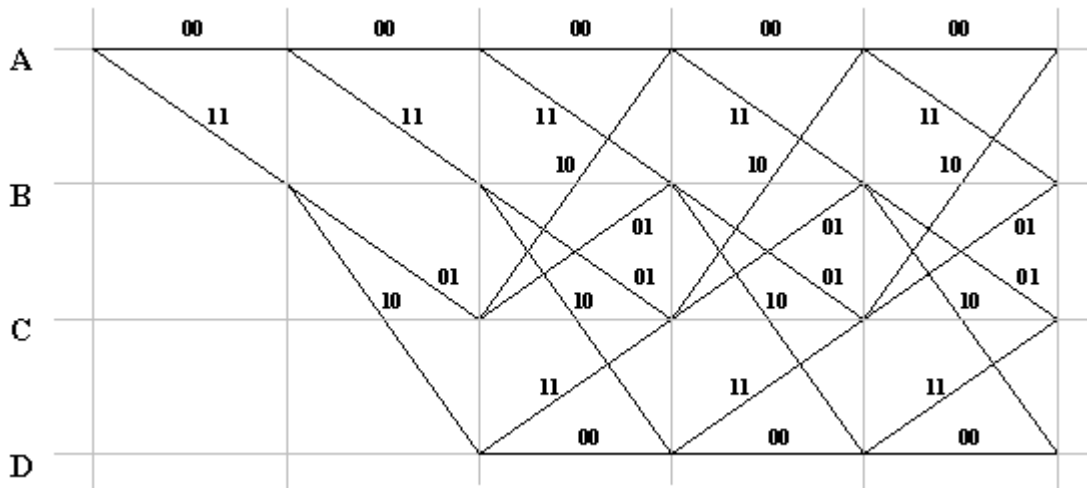


Figura. 4.12. Diagrama de Trellis Codificador (2, 1, 3)

Para la secuencia 0101 del ejemplo, el diagrama de Trellis es:

Secuencia entrada:	1	0	1	0
Secuencia codificada:	11	01	01	01

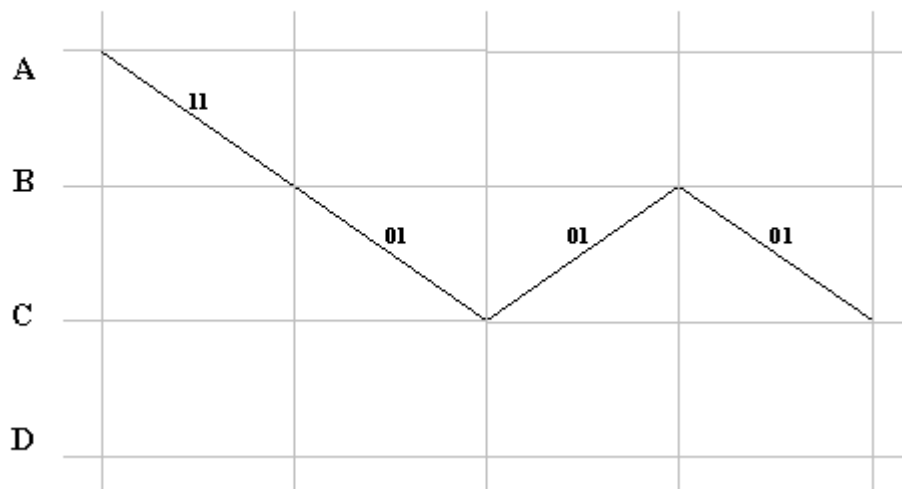


Figura. 4.13. Diagrama de Trellis Codificador (2, 1, 3) secuencia 0101

#### 4.2.2.8.2. Decodificación.

El proceso de decodificación consiste en buscar un camino en el diagrama de Trellis (o en el árbol del código) que nos dé la secuencia de bits más probable (si no hay errores obtendremos la secuencia exacta).

El codificador convolucional añade una estructura a la secuencia de bits. Incluso aunque la entrada sea totalmente aleatoria, se fuerza a que la salida siga unas determinadas secuencias. Esta restricción es la que da la capacidad correctora a los códigos convolucionales. El procedimiento de decodificación es equivalente a comparar la secuencia recibida con todas las posibles secuencias que pueden obtenerse con el correspondiente codificador y seleccionando la secuencia que está más próxima a la secuencia recibida. Para realizar la decodificación se utiliza un algoritmo denominado Algoritmo de Viterbi.

##### 4.2.2.8.2.1. Algoritmo de Viterbi.

El fundamento de este algoritmo está en que no se almacenan todas las secuencias a las que da lugar el codificador. Se basa en el principio de optimalidad: el mejor camino (menor distancia de Hamming) a través del diagrama de Trellis que pasa por un determinado nodo, necesariamente incluye el mejor camino desde el principio del diagrama de Trellis hasta este nodo.

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.

Descripción del algoritmo de Viterbi:

1. En el nivel  $j$ , calcular la distancia de Hamming de cada camino entrante en cada nodo (estado) desde el nodo del nivel  $j-1$  hasta el nodo del nivel  $j$  a través del camino superviviente.

2. Para cada nodo (estado) del diagrama de Trellis en el nivel  $j$ , descartar todos los caminos que entran en el nodo, excepto el de distancia mínima. Cuando a un nodo llegan dos caminos con la misma distancia se toma el superior.
3. Pasar al nivel  $j+1$  y repetir los pasos 1 y 2.

Estos pasos se aplican para  $j$  mayor o igual que 2. Hasta ese valor se expanden los caminos. A continuación se muestra un ejemplo de aplicación de este algoritmo sobre una cadena codificada con el codificador (2,1,3) del ejemplo de codificación.

El codificador envía la secuencia: 11 01 10 11 10 00 11 correspondiente a la codificación de la cadena 1 0 0 1 1 1 0. El receptor recibe la secuencia: 11 01 00 11 11 00 11, con dos errores (posiciones 5 y 10), entonces:

### Paso 1

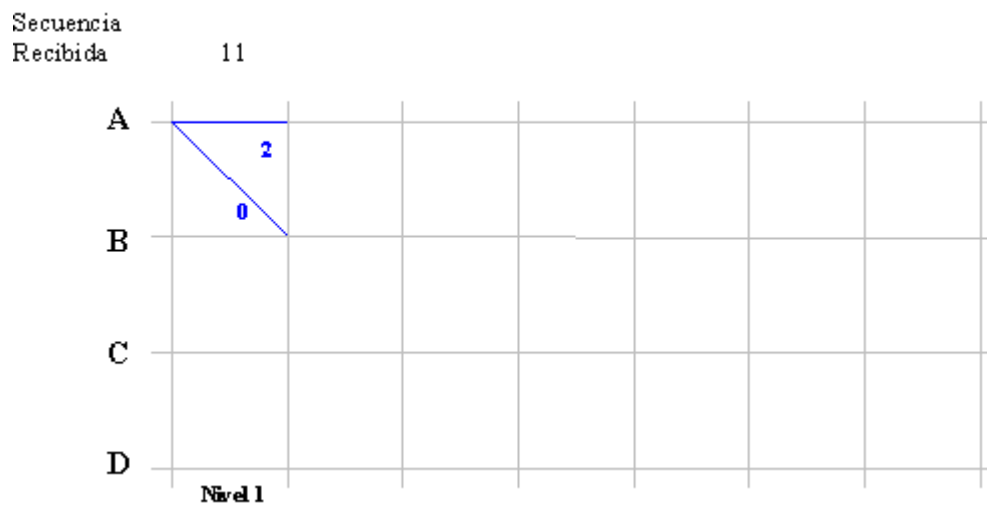
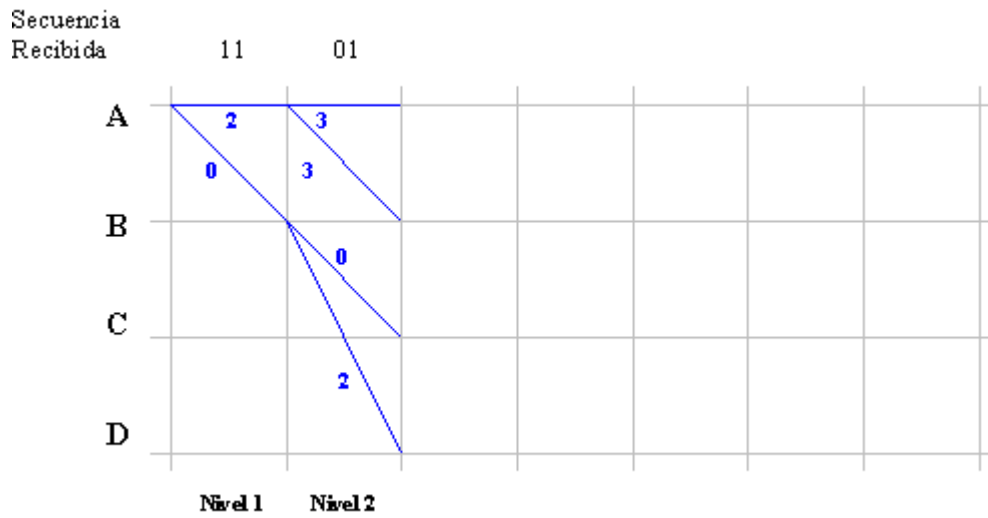


Figura. 4.14a. Algoritmo Viterbi Codificador (2, 1, 3)

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

**Paso 2**



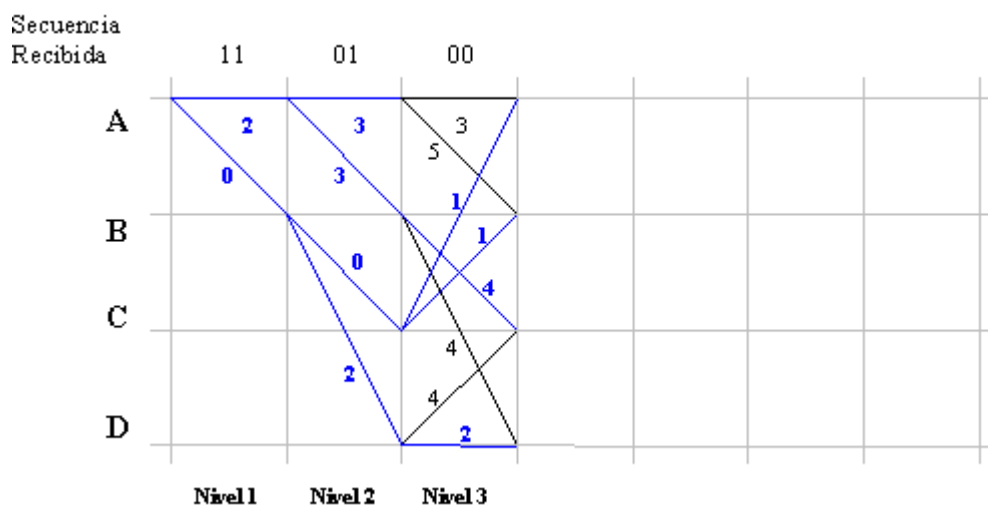
**Figura. 4.14b. Algoritmo Viterbi Codificador (2, 1, 3)**

$$2 + d(01,00) = 2 + 1 = 3, 0 + d(01,01) = 0 + 0 = 0$$

$$2 + d(01,11) = 2 + 1 = 3, 0 + d(01,10) = 0 + 2 = 2$$

En este momento se encuentra en el nivel 2. A partir de ahora se comienza a aplicar el algoritmo. A cada estado del nivel 2 llega un único camino, por lo tanto nos quedamos con todos los caminos hasta el momento.

**Paso 3**



**Figura. 4.14c. Algoritmo Viterbi Codificador (2, 1, 3)**

$$3 + d(00,00) = 3 + 0 = 3$$

$$0 + d(00,10) = 0 + 1 = 1$$

$$3 + d(00,11) = 3 + 2 = 5$$

Para cada estado del nivel 3 nos quedamos con el camino de menor peso que llegue a él. Los caminos que se han seleccionado hasta el momento aparecen marcados de color azul.

**Paso 4**

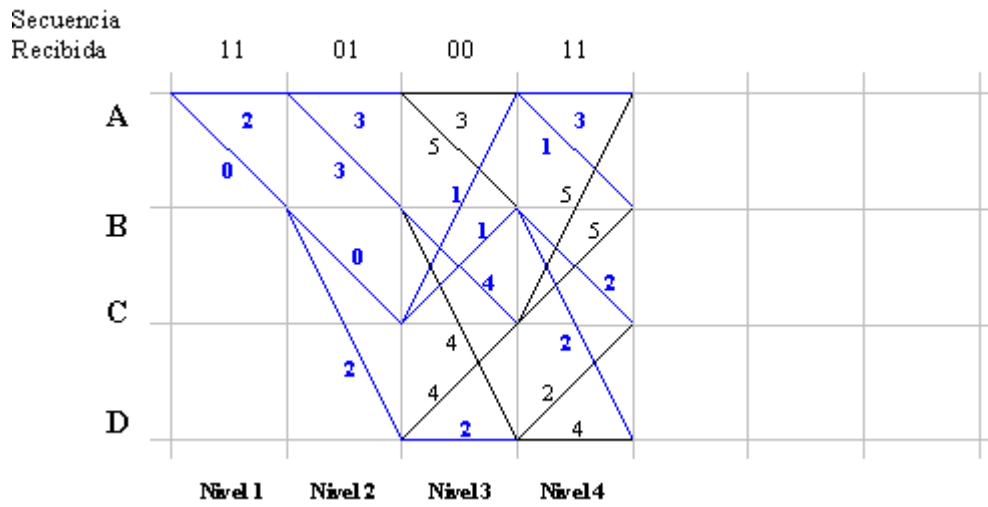


Figura. 4.14d. Algoritmo Viterbi Codificador (2, 1, 3)

**Paso 5**

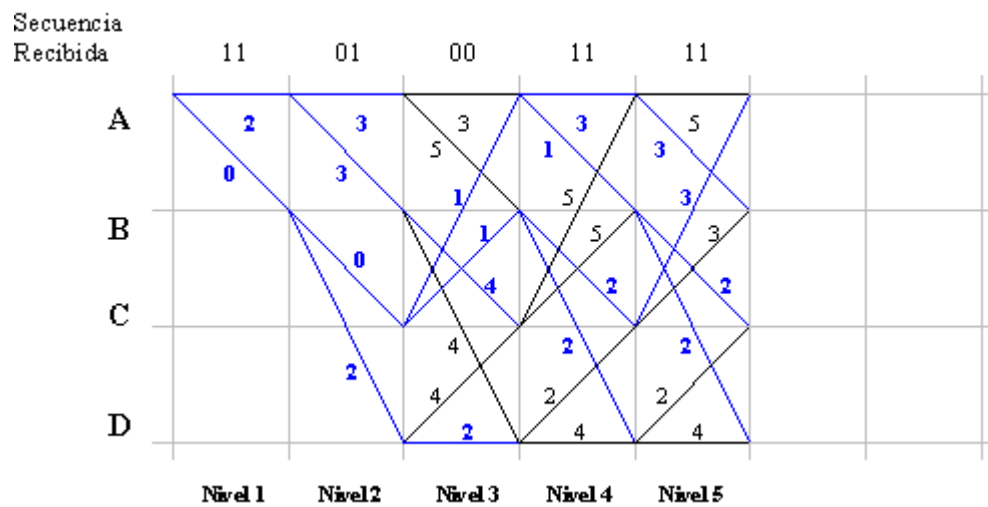


Figura. 4.14e. Algoritmo Viterbi Codificador (2, 1, 3)

**Paso 6**

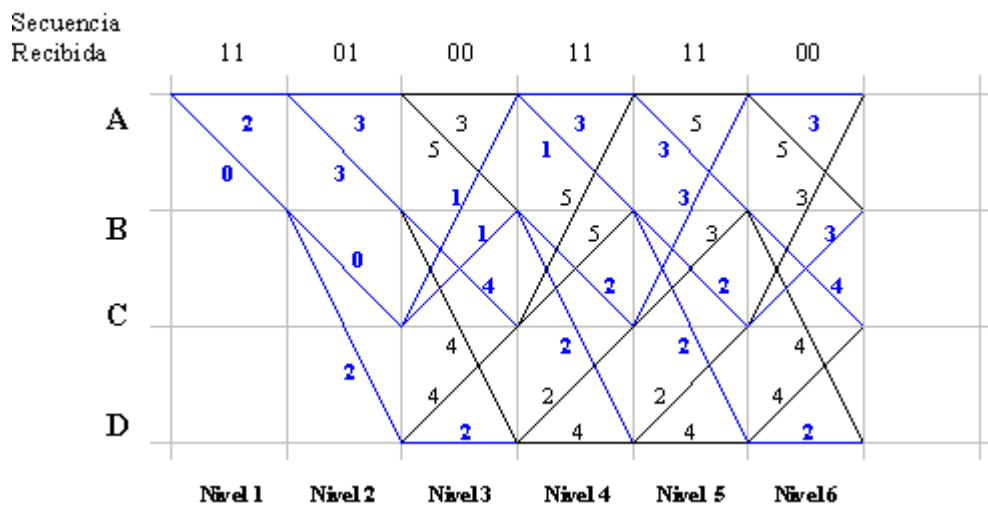


Figura. 4.14f. Algoritmo Viterbi Codificador (2, 1, 3)

**Paso 7**

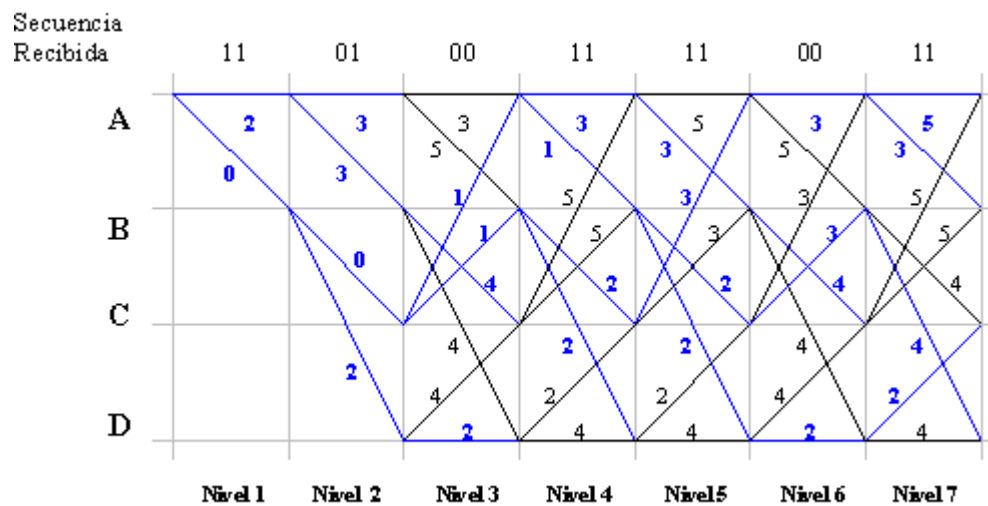


Figura. 4.14g. Algoritmo Viterbi Codificador (2, 1, 3)

Una vez que se ha llegado al final, se escoge el camino que nos da la distancia más corta en el último nivel. Una vez seleccionado el camino, elegimos 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.

Para este ejemplo, se tiene que la distancia más corta en el último nivel es **2** y viene dado por:

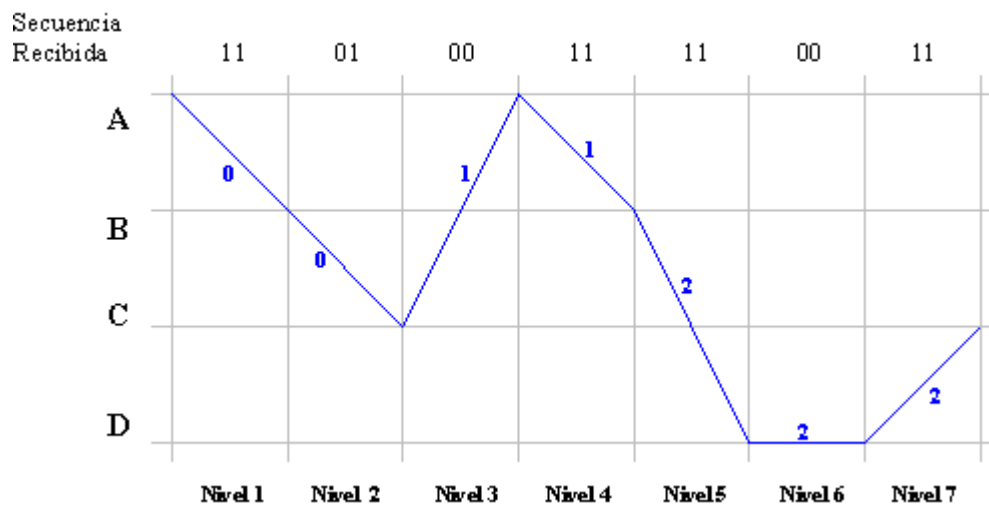


Figura. 4.15. Distancia más Corta para Codificador (2, 1, 3)

Ahora se selecciona en cada nivel el bit de entrada que provoca la transición de estados que se refleja según el camino obtenido.

Nivel	bit
1	1
2	0
3	0
4	1
5	1
6	1
7	0

Tabla. 4.5. Secuencias de Salida por Niveles

Por lo tanto, la secuencia decodificada obtenida es 1 0 0 1 1 1 0, y coincide con la cadena codificada enviada al decodificador.

El código utilizado ha sido capaz de detectar y corregir dos errores.

#### 4.2.2.9. Códigos Reed – Solomon

Los códigos R-S son muy usados en la práctica para lo relacionado con la tecnología de los discos compactos CDs y DVDs. Estos son códigos cíclicos no binarios. Un código R-S  $(n, k)$  expande un bloque de  $k$  símbolos a un bloque de  $n$  símbolos agregando  $n-k$  símbolos redundantes. Cada símbolo procesado consta de  $m$  bits, de modo que debe satisfacerse la relación

$$0 < k < n < 2^m + 2.$$

De esta manera para un código R-S  $(n, k)$  convencional

$$(n, k) = (2^m - 1, 2^m - 1 - 2t)$$

Siendo  $t$  la capacidad de corrección de símbolos. Se tiene también un código **R-S extendido** para el cual

$$n = 2^m \quad \text{o} \quad n = 2^m + 1$$

Los códigos R – S alcanzan la menor distancia de código más grande posible para ningún código lineal con la misma longitud de bloques de entrada y salida. Para los códigos no binarios, la distancia entre dos palabras e código se define como el número de símbolos en los cuales difieren las secuencias, para los códigos R – S, la mínima distancia de código es

$$d_{\min} = n - k + 1$$

Y por lo tanto la máxima capacidad de corrección de errores se define como

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor = \left\lfloor \frac{n - k}{2} \right\rfloor$$

Los códigos R – S pueden ser diseñados para tener cualquier redundancia, pero la complejidad de una implementación de alta velocidad se incrementa con ello. Es por esta razón, que los R – S más atractivos tienen alta tasa de código, es decir, baja redundancia.



Estos también son muy útiles para corrección de error de ráfagas, es decir, son efectivos para canales con memoria. Pueden ser usados eficientemente en canales donde las secuencias de entrada son grandes.

Una característica importante de los códigos R – S es que se pueden añadir dos símbolos de información a un código de longitud  $n$  sin reducir su mínima distancia. Ahora el código será de longitud  $n+2$  y los mismos símbolos de control de paridad como el código original.

#### 4.2.2.9.1. Codificación.

El polinomio generador para un código R – S es de la forma

$$g(X) = g_0 + g_1X + g_2X^2 + \dots + g_{2t-1}X^{2t-1} + X^{2t}$$

El grado del polinomio generador es igual al número de símbolos de paridad  $2t = n-k$ , de forma similar a los códigos BCH, ya que los códigos R – S son un subconjunto de ellos. Debido al grado de  $g(X)$ , existirán  $2t$  potencias de  $a$  que son las raíces del mismo. Por ejemplo podemos considerar el código de corrección de doble error (7, 3), en el cual el grado es  $2t = n-k = 4$ .

$$\begin{aligned} g(X) &= (X - a)(X - a^2)(X - a^3)(X - a^4) \\ &= (X^2 - (a + a^2)X + a^3)(X^2 - (a^3 + a^4)X + a^7) \\ &= (X^2 - a^4X + a^3)(X^2 - a^6X + a^0) \\ &= X^4 - (a^4 + a^6)X^3 + (a^3 + a^{10} + a^0)X^2 - (a^4 + a^9)X + a^3 \\ &= X^4 - a^3X^3 + a^0X^2 - a^1X + a^3 \end{aligned}$$

De forma que ordenando  $g(X)$  de menor a mayor orden y cambiando los signos negativos por positivos, ya que en binario  $+1 = -1$ ,

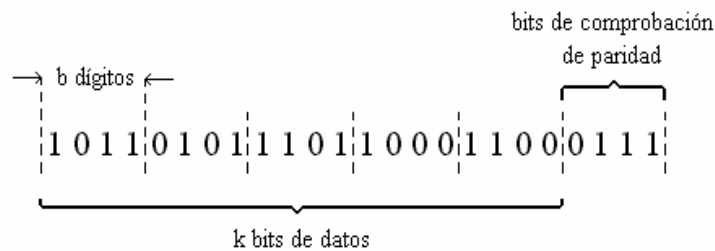
$$g(X) = a^3 + a^1X + a^0X^2 + a^3X^3 + X^4$$

#### 4.2.2.10. Código de Detección y Corrección de Ráfaga de Errores

Los errores en ráfaga son aquellos que destruyen una parte o el total de dígitos de un conjunto secuencial. Los códigos de corrección de errores aleatorios no son eficientes para corregir los errores en ráfaga.

Una ráfaga de longitud  $b$  es una sucesión de dígitos en los cuales el primero y el  $b$ -ésimo se encuentran con error, con los  $b - 2$  dígitos intermedios sin error o recibidos correctamente. Por ejemplo, un vector de error  $e = 0010010100$  tiene una ráfaga de longitud 6. Para detectar todos los errores de la ráfaga de longitud  $b$ ,  $b$  bits de comprobación de paridad son necesarios y suficientes.

Para ilustrar esto, agrupamos  $k$  dígitos de datos en segmentos de  $b$  dígitos de longitud; para esto se agrega un último segmento de  $b$  dígitos de comprobación de paridad; la suma módulo 2 del  $i$ -ésimo dígito de cada segmento, incluyendo el segmento de comprobación de paridad, debe ser cero. Como los dígitos de comprobación de paridad son una combinación lineal de los dígitos de datos, éste es un código lineal de bloque y además es un código sistemático.



**Figura. 4.16. Ráfaga de Errores**

Si una sucesión de dígitos de longitud  $b$  o menor tiene un error, se violará la paridad y el error será detectado, pero no se corregirá, y el receptor podrá solicitar la retransmisión de los dígitos perdidos. Cabe mencionar que el número de dígitos de comprobación de paridad, es independiente de  $k$ , lo que los hace útiles en sistemas en los cuales los dígitos de datos varían de paquete a paquete.

### 4.2.3. TURBO CÓDIGOS

Un gran avance en la teoría de la codificación se dio en 1993, cuando un grupo de investigadores desarrolló en Francia los **turbo códigos**. Los resultados iniciales demostraron que estos podían lograr eficiencias de energía a solo medio decibel del límite de Shannon, resultado que en un principio produjo escepticismo. Pero otros investigadores empezaron a validar los resultados y un esfuerzo masivo estuvo pronto con la meta de explicar de mejor manera la importante eficiencia de los turbo códigos.

Mucha de esta investigación se enfocó en mejorar la practicidad de estos códigos, los cuales tienen ciertas peculiaridades que tornan la implementación menos directa. Actualmente se usa los turbo códigos a nivel de la NASA, para comunicaciones espaciales (CCSDS), difusión de video digital (DVB-T), y para telefonía celular de tercera generación (UMTS y CDMA 2000).

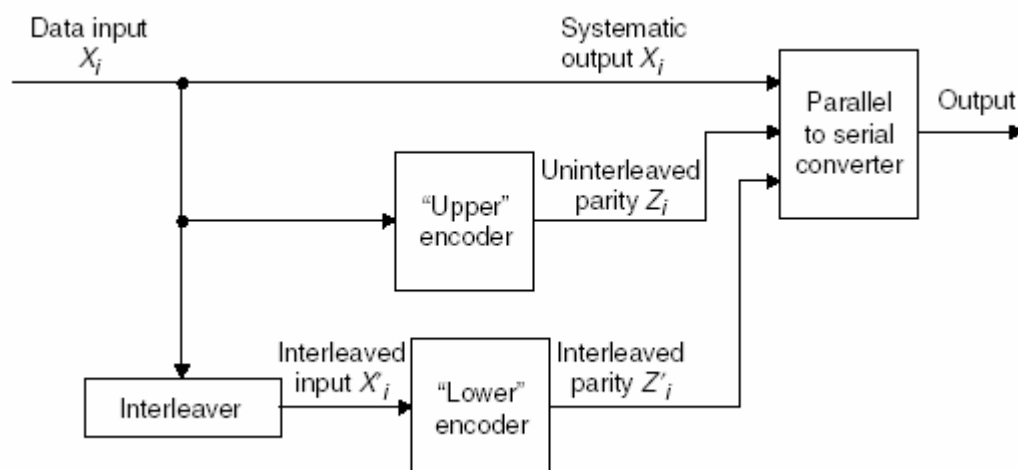
Los turbo códigos son una alternativa poderosa para la codificación de canal en las nuevas generaciones de comunicaciones móviles. Actualmente, la investigación está centrada en el desarrollo e implantación de algoritmos de turbo codificación-decodificación usando plataformas programables de alta velocidad (DSP's y FPGA's), para optimizar la corrección de errores, el consumo de potencia y la velocidad. El turbo codificador, como elemento principal del esquema de codificación-decodificación de canal, contiene los bloques que determinan la eficiencia del turbo código: tamaño del entrelazador y parámetros característicos de los codificadores convolutivos que lo forman. Los FPGA's son una alternativa para el diseño de turbo códigos sobre plataformas reconfigurables que permitan implementar bloques reconfigurables de decodificación y estructuras veloces de codificación en un mismo circuito. La arquitectura propuesta representa una solución sencilla para la turbo codificación dadas las ventajas que ofrecen los FPGA's en términos de programabilidad y reconfigurabilidad sobre otras tecnologías.

El éxito de los turbo códigos radica en la complementación y redundancia de la secuencia codificada, esto es gracias al entrelazador y a su comportamiento pseudoaleatorio que da lugar a rendimientos muy cercanos a los límites de Shannon.

La arquitectura de un turbo-codificador se basa en dos elementos esenciales: El codificador convolutivo y el entrelazador. El codificador convolutivo consiste en una máquina de estados que proporciona una salida que depende del contenido de sus registros y los bits de entrada. El objetivo del entrelazador es modificar la estructura de la secuencia binaria para proporcionarle un grado de aleatoriedad. La secuencia de bits después del entrelazado es la misma, es decir, contiene los mismos bits, sólo que en distinto orden.

El entrelazador representa un factor determinante en el rendimiento y desempeño de un turbo código, entre más grande y más aleatoria sea la secuencia que brinda, se mejorará mucho el desempeño utilizando un mismo esquema de decodificación.

Una de las cosas más interesantes de estos códigos, es que no son un código simple. Son, de hecho, la combinación de dos códigos, aunque pueden ser tres o más (turbo códigos múltiples) que trabajan juntos para lograr la sinergia que no puede ser alcanzada usando un solo código por si mismo. Particularmente, un turbo código está formado por la concatenación paralela de dos codificadores y un entrelazador. Cada código constitutivo puede ser un tipo de FEC. Aunque los dos codificadores pueden ser diferentes, en la práctica son normalmente idénticos. En la figura 4.17 se muestra una estructura general de generación de turbo códigos. El flujo de entrada de datos y las salidas de paridad de los codificadores son serializadas en una palabra de turbo código simple.



**Figura. 4.17. Diagrama General de un Turbo Codificador**

Con el uso del entrelazador, los datos  $\{X_i\}$  se arreglan de forma que el codificador los recibe en otro orden, llamado  $\{X_{i'}\}$ . Así, la salida del Segundo codificador será diferente a la salida del primero, a menos que los datos después del entrelazador sean iguales a la entrada. El entrelazador aquí usado es diferente de los rectangulares usados generalmente en sistemas inalámbricos que buscan espaciar los datos e acuerdo a un patrón establecido, ya que los de los turbo códigos tratan de cambiar el orden de los datos de forma aleatoria e irregular.

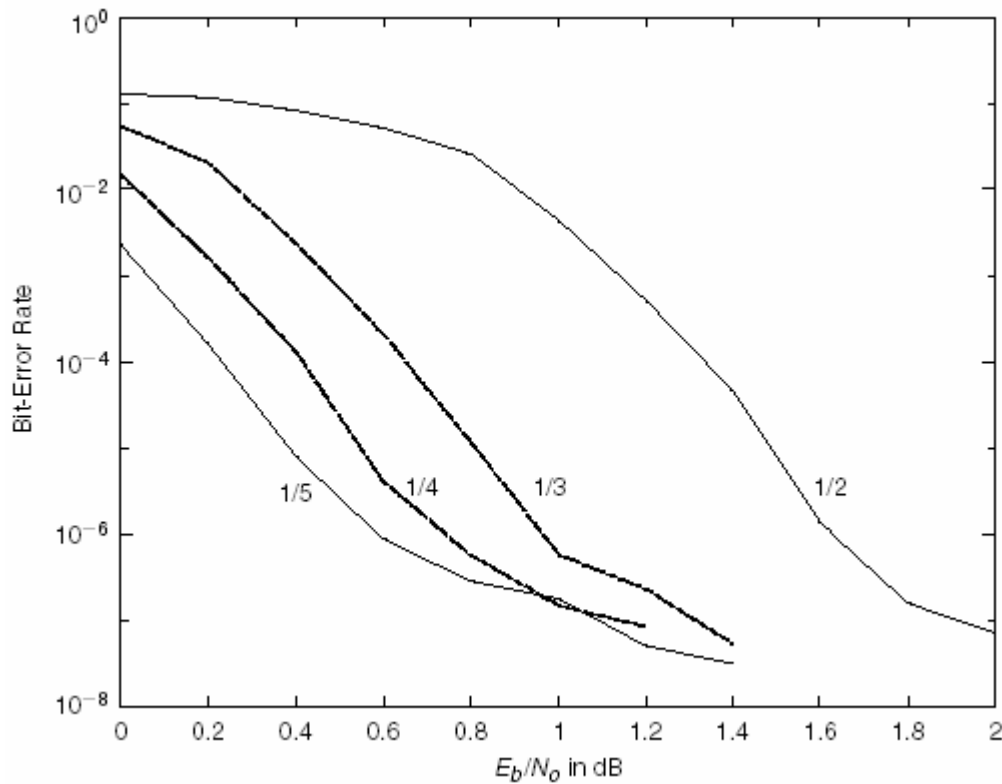
Para entender la importancia de los turbo códigos se debe considerar que un buen código lineal es aquel que tiene la mayor cantidad de palabras de código con gran peso, estas son deseadas por cuanto ello significa que son más distantes y así al decodificador le será más fácil distinguirlas. Cuando se pueden tolerar palabras de código de bajo peso, su frecuencia de ocurrencia debe ser minimizada. Una forma de reducir el número de palabras de bajo peso es usando turbo codificadores. Debido a que el peso de una palabras de turbo código es simplemente la suma de los pesos de la entrada y las salidas de paridad de las dos palabras de código constituyentes, se puede permitir que una de ellas sea de bajo peso ya que la entrada del segundo codificador ha sido revuelta por el entrelazador, su paridad es usualmente muy diferente del primer codificador. Así, aunque es posible que un codificador ocasionalmente produzca una palabra de bajo peso, la probabilidad d que ambos lo hagan es bastante remota. Esta mejora llamada **ganancia del entrelazador** es una de las principales razones que determinan la eficiencia de los turbo códigos.

#### 4.2.3.1. Problemas Prácticos

Aunque los turbo códigos tienen el potencial de ofrecer eficiencias de energía sin precedentes, existen ciertas peculiaridades que deben considerarse.

Primero, cuando la curva de BER cae agudamente con el incremento de SNR para tasas de error moderadas ( $BER > 10^{-5}$ ), la curva de ver comienza a aplanarse a SNR altos. Esta característica puede ser observada en la figura 4.18. La región donde la curva se vuelve plana se conoce como **piso del error** y obstaculiza la habilidad del turbo código de alcanzar menores tasas de error. La existencia de este piso de error se debe a la presencia

de palabras de bajo peso, las cuales a un SNR pequeño son insignificantes, pero en cuanto crece, dominan la eficiencia del código.



**Figura. 4.15. BER de Turbo Código para CDMA2000 Para Varias Tasas de código**  
Palabra de entrada de 1530 bits, BPSK y canal AWGN

El piso de error puede combatirse en varias formas. Una es usando un codificador convolucional sistemático recursivo – RSC con una mayor distancia de espectro. Sin embargo, la eficiencia a bajos SNR disminuye.

Otra forma de reducir el piso de error es colocar dos codificadores constitutivos en concatenación serial, en vez de paralelos. Tal código convolucional serialmente concatenado (SCCC) ofrece alta eficiencia en altos SNR, así como el piso de error es virtualmente eliminado, llevándolo a un BER de  $10^{-10}$ . Sin embargo, la eficiencia en bajos SNR es considerablemente peor que para los códigos convolucionales concatenados paralelos (PCCC).

Otro problema con los turbo códigos es la complejidad, si el turbo decodificador fuere implementado usando el algoritmo de max-log-MAP, entonces cada media iteración

requeriría que el algoritmo de Viterbi se ejecute dos veces. Es decir, si se realizan ocho iteraciones completas, el algoritmo de Viterbi se utilizará 32 veces. Lo que contrasta con un código convolucional convencional que requiere ejecutar Viterbi una sola vez. Esta es la razón por la que la longitud restringida de un constitutivo de turbo codificador es generalmente más corta que la de un código convencional.

Una forma fácil de reducir la complejidad es simplemente parar las iteraciones del decodificador una vez que un bloque ha sido completamente corregido. Esto previene la sobre iteración, lo que significa menor desperdicio de ciclos de reloj. Sin embargo, si el decodificador es parado adaptativamente, el tiempo requerido para la decodificación de cada palabra será altamente variable.

Otra opción de reducir complejidad es implementar el decodificador con circuitería análoga en vez de hardware digital. Usando circuitería análoga, las salidas del decodificador en cientos de Mbps son posibles para simples turbo códigos. Pero para poder hacer eso con sistemas CDMA2000 o UMTS los turbo códigos aun tienen que ser producidos.

Un problema final es la estimación del canal y la sincronización. Para poder transformar la señal recibida en una forma LLR, se requieren algunos conocimientos estadísticos del canal. Para un canal AWGN, se debe conocer la SNR. Para un canal de desvanecimiento con variaciones de amplitud aleatorias la ganancia por bit del canal debe ser conocida. Si el canal además induce un desplazamiento aleatorio de fase en la señal, entonces podría ser necesaria una estimativa de la fase para detección coherente; la sincronización de símbolo debe estimarse usando un algoritmo para el efecto. Además de ello, el decodificador necesita saber cual bit recibido en una secuencia de datos recibidos corresponde al primer bit de la palabra de turbo código.

### 4.3. PRÁCTICAS CON MATLAB

#### 4.3.1. CÓDIGOS DE CONTROL DE ERRORES

##### 4.3.1.1. Objetivos

- Revisar de forma experimental el funcionamiento de algunos de los algoritmos de codificación de canal estudiados.
- Observar por medio de la simulación la respuesta de los códigos de bloques ante un error.
- Comprender de mejor manera la generación de códigos cíclicos y determinar si son o no sistemáticos.
- Realizar la codificación convolucional de una secuencia de datos de longitud variable y su respectiva decodificación mediante el algoritmo de Viterbi
- Corregir una secuencia de código con bits erróneos por medio del algoritmo de Viterbi.

##### 4.3.1.2. Procedimiento

Para la realización de la presente práctica se utilizan funciones realizadas para MATLAB y cuyos códigos fuente se encuentran disponibles en los archivos del mismo nombre. Estas funciones simplifican el desarrollo y mejoran el entendimiento y alcance de los objetivos planteados. Las funciones han sido colocadas en el mismo directorio de la práctica y son compatibles con MATLAB7, que es la versión utilizada en el desarrollo de este proyecto.

1. Códigos de Bloques (7,4). Se ejecuta el archivo **bloques.m**, en el cual se pueden cambiar el vector de datos (`dat_vec`) o el vector de código errado (`code_err`).



- Como se tiene  $n=7$  y  $k=4$ , quiere decir que  $m=3$ . Es decir, que la palabra de datos es de 4 bits, los bits redundantes son 3 y el código final es de 7 bits.
- Se ingresa un vector de datos de 4 bits.
- Se ingresa la matriz generadora de la codificación.
- Por medio de la función **mult\_mod2 (vect\_datos, matr\_generadora)**, que realiza el producto de las matrices ingresadas como parámetros, se realiza la codificación del vector  $x$  con la matriz  $G$ , dando un vector de código.
- Se obtiene la matriz de chequeo de paridad por medio de la función **check\_matrix (mat\_generadora)** a partir de la matriz  $G$ .
- Para verificar la presencia de errores se utiliza nuevamente la función **mult\_mod2 ( )**, pero los parámetros para este caso son el vector de código y la transpuesta de la matriz de chequeo de paridad.
- Se realiza nuevamente el paso anterior pero se cambia el vector de código de manera que se conserve su longitud y se comparan los resultados.

code =

1 0 1 0 0 1 1

H =

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

ans =

0 0 0

code\_err =

1 0 1 0 0 1 0

ans =

0 0 1

2. Códigos cíclicos (7,4). Se ejecuta el archivo **ciclico.m**, en el cual se pueden cambiar el vector de datos (`dat_vec`) o el polinomio generador (`pol_g`).

- Como tenemos que  $n=7$  y  $k=4$ , quiere decir que  $m=3$ . Es decir, que la palabra de datos es de 4 bits, los bits redundantes son 3 y el código final es de 7 bits. Se considera el ejemplo revisado en la teoría de códigos cíclicos.
- Se ingresa un vector de datos de 4 bits.
- Se ingresa el polinomio generador de la codificación, que puede ser cualquiera de las tres opciones que se revisaron:  $x + 1$ ,  $x^3 + x + 1$ ,  $x^3 + x^2 + 1$ .
- Por medio de la función **cyclic\_matrix (pol\_g, longitud)**, que obtiene la matriz generadora a partir del polinomio generador y la longitud es la misma que la del vector de datos, en este caso cuatro.
- Se genera la forma sistemática de la matriz generadora por medio de la función **sys\_matrix ( )**, cuyo argumento es precisamente la matriz obtenida en el paso anterior.
- Finalmente se obtiene la palabra de código por medio de la función **mult\_mod2( )**, cuyos parámetros son el vector de datos y la matriz generadora para código no sistemático, o el vector de datos y la forma sistemática de la generadora, para códigos sistemáticos.
- Las respuestas obtenidas demuestran el ejemplo considerado y las diferencias propias de ellas.

dat\_vect =

1 0 1 0

Gen =

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

Gsis =

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

code =

1 1 1 0 0 1 0

sis\_code =

1 0 1 0 0 0 1

3. Códigos Convolucionales (3,1, 3). Se ejecuta el archivo **convoluc.m**, en el cual se puede cambiar el vector de datos (dat\_vec) al que se desee.

- Como se tiene  $n=3$  y  $k=1$  y  $m=3$ , quiere decir que los datos entran bit a bit, el código generado es de 3 bits por cada bit que ingresa y que la memoria de código es de 3 bloques (actual y dos anteriores).
- Se ingresa un vector de datos cuya longitud puede ser determinada por el usuario; un vector más largo incrementará el código generado y el cálculo computacional.
- Se ingresa la matriz generadora, que viene a ser la matriz de conexiones de los bloques con los sumadores en binario, tal como se muestra en las gráficas de códigos convolucionales revisadas en la parte teórica.

- Se define el número de desplazamientos por ciclo,  $k=1$ .
- Se genera el código convolucional del vector de datos ingresado por medio de la función **conv\_encode (Gen, dat\_vec, k)**, cuyos parámetros son la matriz generadora, el vector de datos ingresado y el número de desplazamientos.
- Una vez obtenida la secuencia codificada, se genera un código errado a partir de la misma. Se realiza de forma que exista un error en cada tercio del código generado.
- Para decodificar este tipo de códigos se utiliza generalmente el algoritmo de Viterbi, el cual se puede simular a través de la función **viterbi(Gen, Cod\_Bip)**, cuyos argumentos son la matriz generadora y la secuencia de código pero de forma bipolar por lo que se la multiplica por dos y se le resta uno, obteniendo valores  $\pm 1$ .
- Se aplica el mismo procedimiento pero ahora con la secuencia de código modificada para que contenga errores. El algoritmo de Viterbi debe detectar y corregir dichos errores. Se compara las respuestas.

dat\_vec =

0 1 0 1

code =

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

code\_err =

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

dat =

0 1 0 1

```
dat_corr =
    0  1  0  1
```

#### 4.3.1.3. Conclusiones

- Se puede comprobar para la codificación por bloques que por el método de la matriz de paridad se obtiene un vector nulo cuando el código es correcto u otro vector en caso de que se presenten errores en la codificación.
- La generación de códigos cíclicos se puede realizar de forma sencilla por lo que este tipo de codificación es el más utilizado en la vida práctica, pudiendo incluirse la información en el código (sistemáticos) o prescindiendo de ella en la palabra de código.
- El uso de codificación convolucional junto con el algoritmo de Viterbi resuelve de forma efectiva la presencia de errores en una secuencia de código, por lo que se entiende el por qué de su uso en canales ruidosos.
- Se pudo comprobar a través de la práctica la corrección efectiva de los errores generados, reafirmando los conceptos repasados en la teoría.

#### Codigo bloques.m

```
%Codificación en Bloques

clear all;
clc;

dat_vec= [ 1 0 1 0 ];
mat_gen= [ 1 0 0 0 1 0 1 ; 0 1 0 0 1 1 1 ; 0 0 1 0 1 1 0 ; 0 0 0 1 0 1 1 ];

code= mult_mod2(dat_vec,mat_gen)
H= check_matrix(mat_gen)

mult_mod2(code,(H'))

code_err=[ 1 0 1 0 0 1 0]
mult_mod2(code_err,(H'))
```

### Codigo ciclico.m

```
%Códigos Cíclicos

clear all;
clc;

dat_vect=[1 0 1 0]

pol_g= [ 1 1 0 1 ];

Gen= cyclic_matrix(pol_g,4)

Gsis= syst_matrix(Gen)

code= mult_mod2(dat_vect,Gen)

sis_code= mult_mod2(dat_vect,Gsis)
```

### Codigo convoluc.m

```
%Códigos Convolucionales
%Decodificacion y Corrección de Errores Con Viterbi

clear all;
clc;

dat_vec= [0 1 0 1]

Gen= [ 1 0 0 ; 1 0 1 ; 1 1 1 ];

k=1;

code= conv_encode(Gen,dat_vec,k)

[l,ln]=size(code);

code_err=code;

for (i=1:ln/3:ln)
    code_err(i)=~code_err(i);
end

code_err

dat=viterbi(Gen,2*code-1)

dat_corr=viterbi(Gen,2*code_err-1)
```

## 4.4. REFERENCIAS

### 4.4.1. LIBROS

- SKLAR, Bernard, **Digital Communications**, Segunda Edición, Prentice Hall, Estados Unidos 2001, páginas 304-370, 382-420, 437-504.
- TOMASI, Wayne, **Sistemas de Comunicaciones Electrónicas**, Segunda Edición, Prentice Hall, México 1996, páginas 519-528.
- COUCH, Leon, **Sistemas de Comunicación Digitales y Analógicos**, Quinta Edición, Prentice Hall, México 1998, páginas 19-26.
- ESPE, **Comunicación Digital**, apuntes de la asignatura, semestre septiembre 2004 – marzo 2005.

### 4.4.2. INTERNET

- <http://www.isa.cie.uva.es/proyectos/codec/teoria4.html>, Códigos de Codificación del Canal.
- <http://www.isa.cie.uva.es/proyectos/codec/teoria1.html>, Conceptos Generales sobre Codificación.
- <http://jungla.dit.upm.es/~trdt/apuntes/t4.html>, Capacidad de Canales.
- <http://www.tecnun.es/Asignaturas/transdat/ficheros%5Cviterbi.pdf>, Codificación Convolutiva y Decodificación de Viterbi.
- <http://gemini.udistrital.edu.co/comunidad/profesores/jruiz/jairocd/htmls/comunicdig.htm>, Codificación de Canal.
- <http://personales.mundivia.es/jtoledo/angel/error/error51.htm>, Códigos Reed-Solomon.
- <http://www.us.es/da/apuntes/ae/ae0405b.pdf>, Códigos Lineales y Cíclicos.
- <http://www.gts.tsc.uvigo.es/gpsc/publications/wmark/icip01.pdf>, Turbo Coding.
- <http://www.csee.wvu.edu/~mvalenti/documents/DOWLA-CH12.pdf>, Turbo Codes.

## CAPÍTULO V

### MÚLTIPLEX

#### 5.1. INTRODUCCIÓN.

Se denomina multiplexar a repartir un canal de comunicación, con capacidad  $C$ , entre varios subcanales que requieran capacidades inferiores. Demultiplexar consistirá en realizar la operación inversa, es decir, volver a obtener cada uno de los subcanales que se están transmitiendo sobre el mismo canal.

Esto facilita el que los canales que comparten el medio de transmisión puedan prestar servicios diferentes y transportar tráfico de distinta naturaleza. Es usual que en un mismo nodo de red se realicen tareas de multiplexión y conmutación.

Hay muchos motivos que justifican la utilización de técnicas de multiplexión, como: Si se utiliza la red telefónica conmutada, compartir un canal vocal entre varias comunicaciones que requieran un ancho de banda inferior al de un canal telefónico analógico (3100 Hz); usar un canal de banda ancha, es decir con capacidad del orden de Mbps, para la transmisión de varias comunicaciones simultáneas con velocidades inferiores, combinar comunicaciones que necesiten distinta capacidad sobre un mismo enlace; y, por supuesto, el ahorro en los costes de transmisión al utilizar circuitos de mayor capacidad. Teniendo muy en cuenta que el coste de instalar y mantener medios de transmisión de baja capacidad viene a ser igual que el de medios de alta capacidad.

La UIT-T normaliza la forma de realizar esta multiplexión. También se establecen jerarquías de multiplexión, que para el caso de multiplexión en el tiempo, TDM, existen



dos jerarquías muy diferentes. La más antigua es la jerarquía digital plesiócrona (PDH) y la más actual es la jerarquía digital síncrona (SDH). Posteriormente se verán ambas jerarquías más en detalle.

En los primeros pasos de la tecnología óptica cada compañía telefónica disponía de su propio sistema TDM óptico. Tras la disgregación de AT&T en el año 1984, dichas compañías se encontraron con el problema de compatibilidad entre portadoras para largas distancias debido a esos diferentes sistemas. Se hizo entonces necesaria una estandarización.

En 1985, Bellcore empezó a trabajar en ello estableciendo las bases de SONET (Synchronous Optical Network, Red Óptica Síncrona). Más tarde el entonces CCITT se unió al trabajo, lo que daría lugar de forma paralela al estándar SONET por un lado y a las recomendaciones que originarían SDH (Synchronous Digital Hierarchy) por otro.

Hoy las redes de telecomunicación han tenido un incremento en el tráfico de datos, por lo que las redes que utilizan el multiplexado por división de tiempo (TDM), originalmente inventado para ser la técnica de multiplexado más eficiente para la posible transmisión de voz en una velocidad de 64 Kbits/seg, ya no son tan adecuadas para el tráfico de datos. Un router o un ATM se pueden conectar a una red de transporte mediante WDM, a través de mapeo de paquetes o células directamente dentro de una longitud de onda sin la intervención del uso de una SONET o SDH TDM. De hecho, un TDM puede ser reemplazado por DWDM, el cual puede incrementar la utilización del ancho de banda, facilitando la red, y reduciendo costo.

DWDM ha revolucionado la industria de las telecomunicaciones proveyendo de infraestructura para largo transporte de redes ópticas.

## 5.2. MARCO TEÓRICO

### 5.2.1. SINCRONISMO

Cuando el modo de transmisión entre dos equipos es asíncrono éstos no tienen un reloj común. Sin embargo es imprescindible que el receptor sepa en qué instante recibe un bit y cuál es la duración del mismo, ya que tiene que recogerlo del enlace para interpretar la información que le está llegando. Para conseguir esta imprescindible sincronización el emisor envía la información en bloques de palabras (o caracteres), cada vez que va a transmitir una palabra nueva envía un bit especial denominado bit de comienzo o arranque, que indicará al receptor que a partir de ese momento va a comenzar a recibir una palabra. Terminada la transmisión de la palabra el emisor envía uno (o varios bits) denominados de final o parada, que indican al receptor que la palabra ha terminado. Como la longitud de la palabra es fija (normalmente 8 bits) el receptor puede leer e interpretar esta información sin problemas y sin la necesidad de compartir un reloj con el emisor.

El mayor problema de este mecanismo es la necesidad de insertar bits adicionales de parada y espera que disminuyen el rendimiento de las líneas de transmisión.

Cuando el modo de transmisión es síncrono los datos se transmiten con un ritmo fijo, marcado por un reloj o base de tiempo *común* a los equipos transmisor y receptor. Se denomina sincronismo al establecimiento de este reloj con la finalidad de interpretar adecuadamente las señales recibidas. Cuando existe este sincronismo los equipos involucrados en una transmisión conocen el instante exacto en que debe aparecer determinada información, de este modo las tareas de inserción, recuperación o conmutación de los datos del canal resultan mucho más sencillas.

Además no es necesario transmitir bits de arranque y parada para indicar el comienzo y el final de la transmisión, como ocurría con las transmisiones asíncronas. Esto conlleva un mejor aprovechamiento de la capacidad del enlace, ya que toda la información transmitida es útil.

Todos los componentes de un sistema de transmisión digital necesitan una señal de reloj para su funcionamiento. Utilizando ésta señal se generarán las secuencias de unos y ceros, es decir la información digital, con un régimen binario determinado.

Dentro de un dispositivo, donde las distancias son pequeñas, existe un reloj que proporciona el sincronismo para todos los elementos que lo componen. Así en un computador, o en una central de conmutación, todos los componentes funcionan exactamente a la misma frecuencia. Si las distancias entre los componentes del sistema aumentan, como ocurre en un sistema de transmisión, conseguir un reloj común resulta más complejo y puede requerir de toda una red adicional de sincronización.

#### **5.2.1.1. Modos de Operación**

Existen dos modos fundamentales de operación.

##### **5.2.1.1.1. Plesiócrono.**

Cada reloj del sistema opera de forma independiente. Es necesario entonces utilizar relojes de alta estabilidad y resintonizarlos periódicamente de forma manual con el fin de que operen dentro de unos límites muy cercanos a la frecuencia nominal de la red. De esta forma se intentará mantener los deslizamientos, **Jitter**, en un nivel aceptable. La jerarquía digital plesiócrona sigue este modo de operación y se usa, sobre todo, en enlaces entre redes de distintos operadores o internacionales que conectan redes funcionando en modo síncrono.

##### **5.2.1.1.2. Síncrono.**

Todos los relojes están controlados por un mecanismo automático, de forma que todos operan en la misma frecuencia nominal. Existen uno o dos relojes de muy alta calidad y precio, por ejemplo relojes atómicos, que sirven de referencia y controlan las frecuencias de los demás que son más baratos y de menor calidad. Todos los relojes operan en la misma frecuencia, con lo que se evitan los desplazamientos en condiciones normales. La información de temporización se disemina desde el reloj primario de referencia (PRC) a

todos los demás por una red superpuesta de control de sincronización. La jerarquía digital síncrona sigue este modelo y se suele utilizar en comunicaciones dentro de la red de un mismo operador.

### **5.2.1.2. Métodos de Sincronismo**

Para conseguir el sincronismo de los relojes de los distintos elementos de red se utilizan principalmente dos métodos: Imposición de sincronismo y sincronismo mutuo. En el primer caso se necesita un reloj que sirva de referencia (el PRC) que es el que gobierna la sincronización de red, en el segundo caso el nodo se sincroniza con las señales de información de entrada.

#### **5.2.1.2.1. Reloj Primario de Referencia –PRC.**

En el método de imposición de sincronismo el reloj de referencia primario dicta la frecuencia a los demás componentes de la red bien directamente o bien a través de un reloj de jerarquía inferior. Este reloj tiene que tener unos requisitos especiales, en concreto debe tener una variación de fase reducida y una deriva de frecuencia inferior a  $10^{-11}$ , es decir que debe ser un reloj de alta tecnología. Éste puede funcionar de forma autónoma, normalmente en condiciones controladas de temperatura, humedad, o bien puede sincronizarse con un sistema externo más exacto.

#### **5.2.1.2.2. Escala de Tiempo Universal Coordinado – UTC.**

También se conoce como GMT (Greenwich Mean Time) u hora Zulu. Es una escala solar es decir que se basa en la cuenta de las rotaciones de la tierra desde un origen de tiempo determinado, el 1 de enero de 1972. Utiliza un conjunto de relojes de cesio en condiciones fuertemente controladas en laboratorios de estandarización.

### 5.2.1.2.3. Sistema de Posicionamiento Global – GPS.

Necesita una perfecta sincronización entre sus satélites y utiliza una escala de tiempo atómica basada en contar los ciclos de una señal en determinada resonancia atómica (relojes atómicos). Utiliza relojes en estaciones terrenas de control y en los propios satélites. El origen de tiempo es el 6 de enero de 1980 y está 13 segundos adelantado a UTC.

### 5.2.1.2.4. LORAN – C.

El sistema Long Range Navigation proporciona servicios de navegación, localización y temporización para usuarios civiles o militares en tierra, mar o aire utilizando una escala de tiempos atómica. Determina posiciones con exactitud de 50m. Se desarrolló para las costas americanas. El origen de tiempo de este sistema son las 0 horas del 1 de enero de 1958 y está 22 segundos adelantado a UTC.

### 5.2.1.2.5. TAI.

El sistema Temps Atomique International (Tiempo atómico internacional) también utiliza una escala de tiempo atómica mantenida por la BIPM (International Bureau of Weights and Measures). Se basa en la comparación de unas 200 frecuencias localizadas en todo el mundo. Está 32 segundos por delante de UTC.

## 5.2.2. PDH – PLESIOSYNCHRONOUS DIGITAL HIERARCHY

Los sistemas de multiplexión digital de primera generación se denominan sistemas plesiócronicos y se habla de la Jerarquía digital plesiócrona (PDH). El término plesiócrono procede del griego, su significado es **casi – síncrono**, es decir, señales digitales de diferentes fuentes disponen de sus propios relojes. Todas ellas operan aproximadamente a

la misma tasa ( $X$  Mbps), pero pueden existir pequeñas desviaciones, de tal forma que alguna puede operar a una tasa ligeramente mayor o menor ( $X \pm x$  Mbps).

**Señal digital X** es un término referido a la serie de tasas de transmisión digital basadas en DS<sub>0</sub>, que tiene una velocidad de transmisión de 64Kbps (el ancho de banda usado normalmente en el canal telefónico). Tanto el sistema norteamericano y japonés T-carrier como el europeo E-carrier operan usando la serie DS como base.

En esta jerarquía de multiplexión el número de canales de un orden es siempre múltiplo entero del número de canales en el nivel anterior, sin embargo no ocurre lo mismo con los regímenes binarios medidos en Mbps. La razón radica en que al pasar de un nivel a otro es necesario agregar información de control que es para uso exclusivo del nivel siguiente. De esta forma los relojes usados en un nivel son independientes de los usados en otros niveles. Esto obliga a que para acceder a un canal en concreto sea necesario ir deshaciendo todas las multiplexiones realizadas, recuperando las tramas de cada nivel, hasta llegar de nuevo al grupo básico, donde ya se puede identificar la información de interés.

Dentro del mismo nivel de la jerarquía se utilizan relojes con la misma frecuencia nominal, sin embargo la frecuencia instantánea de estos relojes puede variar, es lo que se conoce como deriva de frecuencia. Esto implica que se deben utilizar procedimientos para ajustar estas posibles fluctuaciones en la frecuencia. Para ello se reservan, dentro de la trama de cada nivel, espacios denominados de justificación o **bit stuffing**. El método más sencillo consistirá en asignar a cada afluente, o flujo de menor orden, que conforma el múltiplex una capacidad superior a la nominal. Cuando la frecuencia instantánea supere a la nominal estos bits de justificación contienen información útil, mientras que si no es así simplemente llevan información de relleno. La justificación positiva/nula/negativa es algo más compleja. Se define, dentro de la trama, una zona para justificación positiva, que se utiliza cuando la frecuencia instantánea del afluente es menor que la nominal, enviando información de relleno. Además existe una zona de justificación negativa, en la que se transmite información útil sólo cuando la frecuencia instantánea supera a la nominal.

### 5.2.2.1. Jerarquías

Cada nuevo nivel de la jerarquía se forma mediante la multiplexión de varias señales del nivel inmediatamente inferior, a las que se le denomina afluentes. Se añade también información de control, que usará sólo ese nivel, para tareas como alineación de trama, señalización o justificación. Mientras que el grupo básico se forma por entramado de caracteres los niveles más altos se forman por entramado de bits. Dentro de PDH existen tres jerarquías de multiplexión distintas, la europea, la americana y la japonesa.

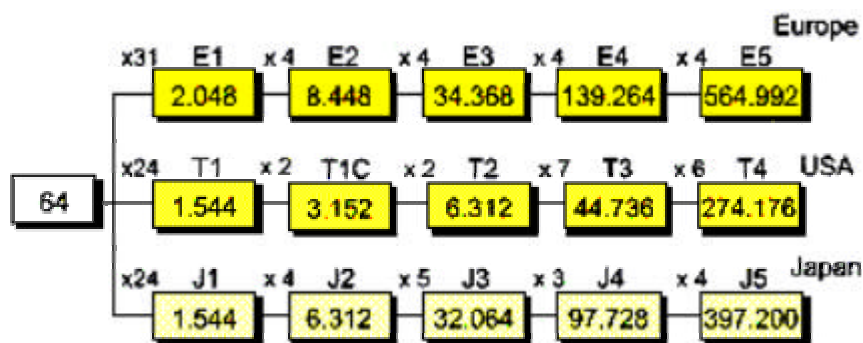


Figura. 5.1. Jerarquías PDH.

#### 5.2.2.1.1. Europea.

En la norma europea cada nivel de multiplexión se denomina E1, E2... que significa estándar europeo. La tabla 5.1, que se presenta a continuación muestra las características de cada uno de los niveles. Se puede observar que la duración de la trama en cada nivel es diferente, cosa que no ocurrirá en la jerarquía síncrona. El grupo básico utilizado es el dado en la recomendación G.732. Los sistemas E5 y E6 son propietarios, ya que no se encuentran normalizados y cada fabricante puede implementarlos como quiera.

Llamada también Jerarquía CEPT (Conference of European Postal and Telecommunication Administration), está regulada por la norma ITU-T G702. Su zona geográfica de utilización es el resto del Mundo (excluyendo a Norteamérica y Japón), aunque con mayor desarrollo en Europa.

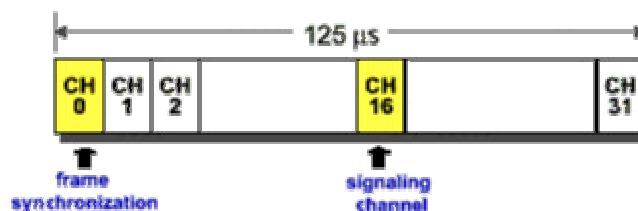
<i>T. trama <math>\mu</math>s</i>	<i>NIVEL</i>	<i>CIRCUITOS</i>	<i>VELOCIDAD</i>	<i>COMPOSICIÓN</i>
125	E1	32	2,048Mbps	32 señales 64Kbps
100,38	E2	128	8,448Mbps	4 de 2Mbps
44,69	E3	512	34,368Mbps	4 de 8Mbps
20,85	E4	2048	139,264Mbps	4 de 34Mbps
4,7	E5	8192	564,992Mbps	4 de 140Mbps
	E6	32768	2,5Gbps	4 de 565Mbps

**Tabla. 5.1. Jerarquía Europea**

La portadora E1 transporta datos a una tasa de 2048 Mbps, incluyendo 32 canales DS0 (de 64 Kbps, telefónicos). Las sucesivas portadoras van multiplicando por 4 su capacidad. Cada uno de los 32 canales de E1 transporta 8 bits y, a diferencia de T-1, la señalización va incluida en cada una de las tramas mediante el uso reservado de 2 de los canales, dejando pues 30 para datos codificados con precisión de 8 bits (con lo cual cada usuario dispone de una tasa de 64 Kbps).

Cada grupo de 4 tramas permite 64 bits de señalización, la mitad de los cuales son para los canales y la otra mitad para las propias tramas o reservados para uso particular en cada país. Por tanto, la trama está compuesta por  $32 \times 8 = 256$  bits que se transmiten en 125 $\mu$ s, lo cual origina una tasa de 2048 Mbps, de los cuales:

- Tasa de información: 1920 Mbps.
- Señalización total (de cada canal + trama): 128 Mbps.



**Figura. 5.2. Trama PDH Europea**

La velocidad de datos por canal es proporcionalmente superior al T-1 diseñado originalmente porque, a diferencia de éste, los 8 bits de cada canal son usados para



codificar información (sin 'robo de bit'). Sin embargo para señalización se emplean 2 canales por trama, con lo cual el número real de canales disponibles para datos es de 30.

A diferencia del caso de portadora T (Americano/Japonés), el factor de combinación para las sucesivas portadoras en el sistema CEPT es siempre de cuatro.

#### 5.2.2.1.2. Americana.

En las normas Americana y japonesa cada grupo u orden se denomina T1, T2... El grupo básico utilizado es ahora el dado en la recomendación G.733.

<i>NIVEL</i>	<i>CIRCUITOS</i>	<i>VELOCIDAD</i>	<i>Grupos de orden inferior</i>
T1	24	1,544Mbps	
T2	96	6,312Mbps	4
T3	672	44,736Mbps	7
T4	2016	139,264Mbps	3

**Tabla. 5.2. Jerarquía Americana**

Sus características más relevantes, así como su trama se explican en la jerarquía japonesa al corresponder a una misma clasificación.

#### 5.2.2.1.3. Japonesa.

<i>NIVEL</i>	<i>CIRCUITOS</i>	<i>VELOCIDAD</i>	<i>Grupos de orden inferior</i>
T1	24	1,544Mbps	
T2	96	6,312Mbps	4
T3	480	32,064Mbps	5
T4	1440	97,728Mbps	3

**Tabla. 5.3. Jerarquía Japonesa**

Regulada por la norma ITU-T G704 se utiliza principalmente en Norteamérica y Japón. La portadora T-1 transporta datos a una tasa de 1544 Mbps, incluyendo 24 canales DS0 (de 64 Kbps, telefónicos). Cuatro de estas tramas se multiplexan para originar la portadora T-2.

Las sucesivas portadoras (T-3, T-4 y T-5) difieren en los casos norteamericano y japonés. En el caso americano también se estandarizó la T-1C, consistente en dos fundamentales T-1, es decir, 48 canales.

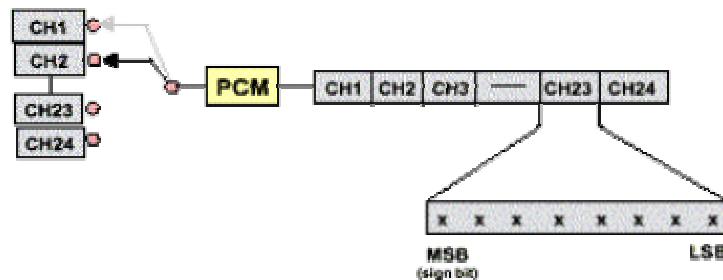


Figura. 5.3. Portadora T – 1

Cada uno de los mencionados 24 canales de T-1 transporta 8 bits, de los cuales:

- Siete son de datos (y el usuario 've' una tasa de  $7 \cdot 8000 = 56$  Kbps)
- Uno de señalización de canal ( $1 \cdot 8000 = 8$  Kbps/canal).

Una trama consiste, pues, en  $24 \cdot 8 = 192$  bits, más uno extra para señalización de trama, lo que arroja un total de 193 bits cada 125us. Esto hace necesaria una capacidad de 1544 Mbps, de la cual:

- Velocidad de información: 1344 Mbps.
- Señalización total (de cada canal + trama): 200 Mbps.

El bit 193 se emplea para sincronización, y sigue el patrón 01010101. Normalmente, el receptor chequea este bit para asegurarse que no ha perdido el sincronismo. Si lo pierde, puede emplear este bit para recuperarlo. El patrón está pensado para que los usuarios no puedan generarlo, ya que corresponde con un tono de 4 Khz, el cual sería previamente filtrado.

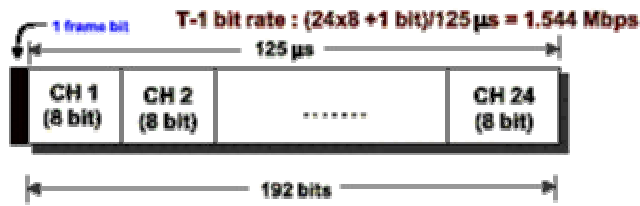


Figura. 5.4. Trama PDH Americana

El sistema TDM permite que varias portadoras T-1 sean multiplexadas en otras de mayor orden. Esta operación se realiza a nivel de bit y no de byte, esto es, de cada una de las portadoras T-1 se toma 1 bit y así sucesivamente hasta completar todas las T-1 que se combinan.

**5.2.3. SONET (SYNCHRONOUS OPTICAL NETWORK)**

La trama básica SONET consiste en un bloque de 810 bytes cada 125 μs, al ser un sistema síncrono estas tramas son enviadas exista o no información útil. Con un total de 8000 tramas por segundo se ajusta a la frecuencia de muestreo de canal PCM para telefonía digital.

Estas tramas se estructuran en bloques de 9 filas y 90 columnas, lo que arroja un total de  $810 \times 8 = 6480$  bits transmitidos 8000 veces por segundo, que suponen una tasa de 51.84 Mbps. Éste es el canal básico de SONET, llamado STS-1 (Synchronous Transport Signal-1). El resto de las jerarquías superiores son múltiplos de STS-1, como se verá más adelante.

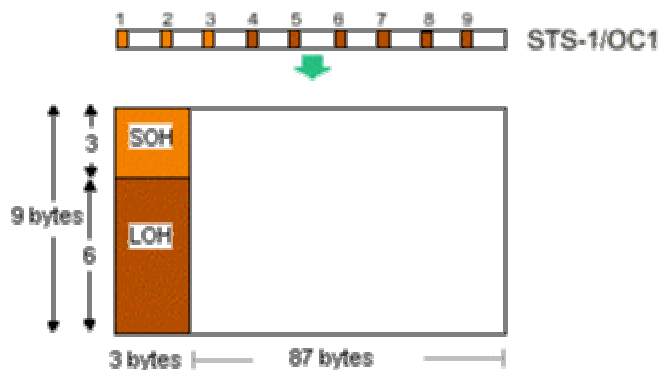


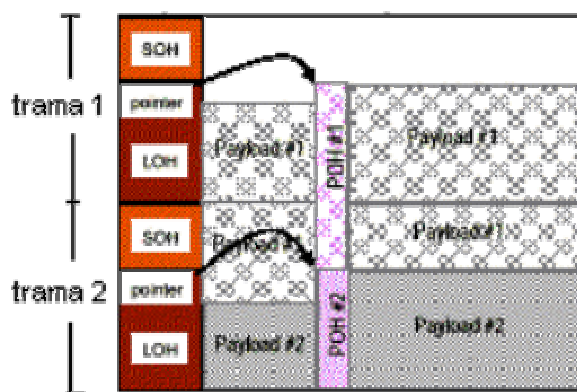
Figura. 5.5. Trama SONET

Las primeras 3 columnas de cada trama están reservadas para información del sistema:

- Las 3 primeras filas (SOH - Section OverHead) forman la cabecera de sección.
- Las 6 restantes (LOH - Line OverHead), la cabecera de línea.

Las restantes 87 columnas transportan los  $87 \cdot 9 \cdot 8 \cdot 8000 = 50.112$  Mbps de datos para usuarios. Sin embargo, esta información llamada SPE (Synchronous Payload Envelope), no tiene porqué comenzar en la fila 1, columna 4. Puede hacerlo en cualquier punto de la zona; un puntero situado en la 1ª fila del LOH apunta ese primer byte de datos. Asimismo, la primera columna del SPE es la cabecera de línea (la que faltaba), llamada POH (Path OverHead).

Todo esto se puede entender mejor examinando la figura 5.6.



**Figura. 5.6. Synchronous Payload Envelope**

Esta característica del SPE de empezar en cualquier parte y de poder incluso extenderse en varias tramas dotan al sistema de una gran flexibilidad, permitiendo por ejemplo la compatibilidad con celdas ATM (Asynchronous Transfer Mode).

Las cabeceras de sección, línea y camino contienen una serie de bytes usados para OAM (Operation, Administration and Maintenance). Como cada byte se repite 8000 veces por segundo, representa un canal PCM. Se encargan de la señalización para canales de voz, de trama, paridad, monitorización de errores, sincronización.

Partiendo de varios flujos de baja tasa formamos la velocidad básica STS-1 empleando, de ser necesario, rellenos para llegar hasta los 51.84 Mbps. A continuación, combinamos tres para obtener la siguiente portadora STS-3 de 155.52 Mbps. Tres de éstas se vuelven a multiplexar y en caso óptico se hace un barajado previo para prevenir secuencias demasiado largas de 0's o 1's que puedan interferir con el reloj, así como la conversión de señal eléctrica a óptica.

SONET	Tasa total	Tasa SPE	Tasa Datos usuario
STS-1 / OC -1	51.84 Mbps	50.122	49.536
STS-3 / OC -3	155.52 Mbps	150.336	148.608
STS-9 / OC -9	466.56 Mbps	451.008	445.824
STS-12 / OC -12	622.08 Mbps	601.344	594.432
STS-18 / OC -18	933.12 Mbps	902.016	891.648
STS-24 / OC -24	1244.16 Mbps	1202.688	1188.864
STS-36 / OC -36	1866.24 Mbps	1804.032	1783.296
STS-48 / OC -48	2488.32 Mbps	2405.376	2377.728

**Tabla. 5.4. Características de SONET**

Se han definido tasas de STS-1 a STS-48. La portadora óptica correspondiente a STS-n se denomina OC-n y es idéntica a excepción del barajado y la conversión que se realizan.

#### 5.2.4. SDH – SYNCHRONOUS DIGITAL HIERARCHY

La jerarquía digital síncrona viene descrita en la recomendación G.707 de la UIT-T. En esta jerarquía se utiliza la misma base de tiempo para todos niveles. Además la deriva en frecuencia de los relojes del sistema está muy limitada, gracias a la utilización de redes de sincronismo superpuestas que transmiten información de sincronización.

Esto facilita algunas tareas, como la extracción e inserción de afluentes en tramas de niveles altos. En esta jerarquía para recuperar una carga en concreto no es necesario

deshacer todas las multiplexiones realizadas, alineándose a la trama de cada nivel, ya que se conoce la posición en la que se encuentra esta información, gracias a la utilización de punteros, y se puede acceder directamente a ella. Todas las tramas tienen la misma duración,  $125\mu\text{s}$ , al contrario que ocurría en la jerarquía plesiócrona. SDH puede transportar cargas de cualquier tipo: flujos PDH, desde el nivel más bajo al más alto, flujos ATM, señales de TV, etc. La capacidad adicional, o tara, dentro de las tramas permite transportar canales de servicio de gran capacidad. Estos canales se pueden usar para transportar cualquier tipo de información que pudiera ser útil a la red, por ejemplo para tareas de señalización o de mantenimiento y gestión de red.

La trama básica se denomina STM-1 y su velocidad es de 155,52 Mbps. Existen tramas de mayor nivel, que se forman con la multiplexión, octeto a octeto, de N tramas básicas, sin incorporar ninguna información adicional, estas tramas se denominan STM-N. La duración de todas las tramas es  $125\mu\text{s}$  y la longitud depende del nivel (N) y será  $270 \times 9 \times N$  octetos.

Dentro de la STM-1 existe un espacio de carga, siempre del mismo tamaño ( $261 \times 9$  octetos), reservado para transportar los afluentes que viajan en la trama, que como ya se ha comentado, pueden ser de muy diversa naturaleza. La formación de la STM-1 variará en función de las cargas que se pretende transmitir. Cada flujo se acomoda dentro de un contenedor virtual, cuyo tamaño dependerá del tipo de carga. Así, para cargas PDH de tipo E4 o T3 los contenedores virtuales se denominan VC-4 y VC-3 respectivamente, éstos son los contenedores virtuales de orden alto. Los demás contenedores se denominan de orden bajo y acomodan flujos más lentos, como E1, en el VC-12. A partir de los contenedores de orden bajo se forman unidades tributarias (TU), simplemente añadiendo un puntero que identifica en qué posición se encuentra el contenedor virtual.

Multiplexando, octeto a octeto, distintas TU y sin añadir ninguna información adicional, se obtienen los grupos de unidades tributarias (TUG). Estas TUG se acomodan dentro de contenedores virtuales de orden alto. Si a los contenedores de orden alto se les añade un puntero, que identifica dónde comienza el contenedor virtual dentro del espacio de carga, se forma la denominada unidad administrativa. Multiplexando diversas unidades administrativas se forma un grupo de unidades administrativas, añadiendo las cabeceras

necesarias se obtiene el STM-1. La siguiente figura está extraída de la norma y refleja el proceso explicado.

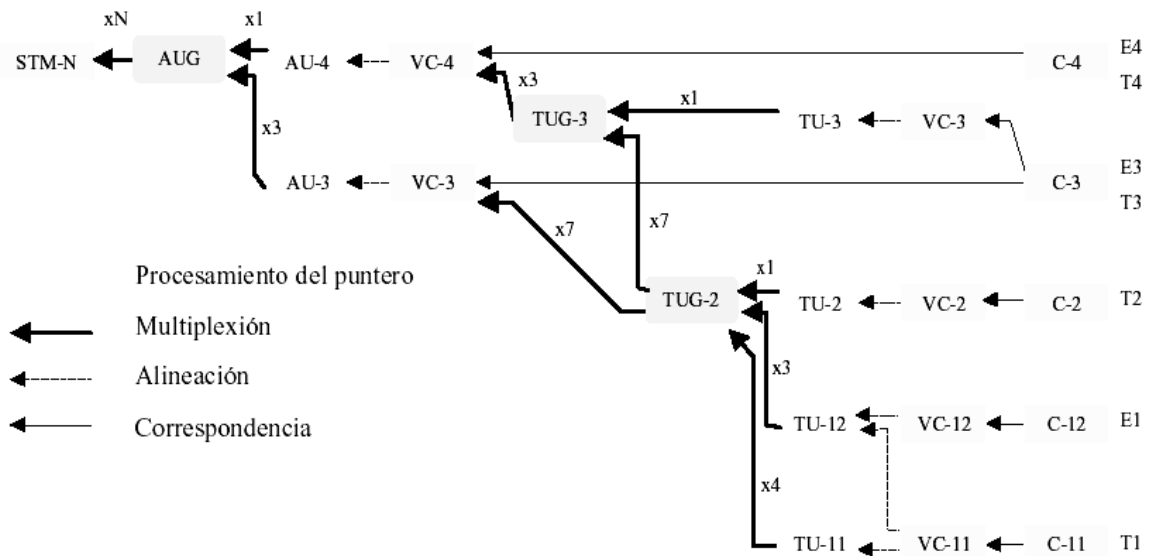


Figura. 5.7. Multiplexado SDH

La trama fundamental de la jerarquía SDH, STM-1(Synchronous Transport Module), tiene exactamente la misma tasa que el segundo nivel de SONET; el resto de las combinaciones superiores coinciden. Esto es, la única diferencia en este aspecto es que SDH no define un equivalente del nivel STS-1 inicial de SONET. El manejo de cabeceras y campos es también idéntico.

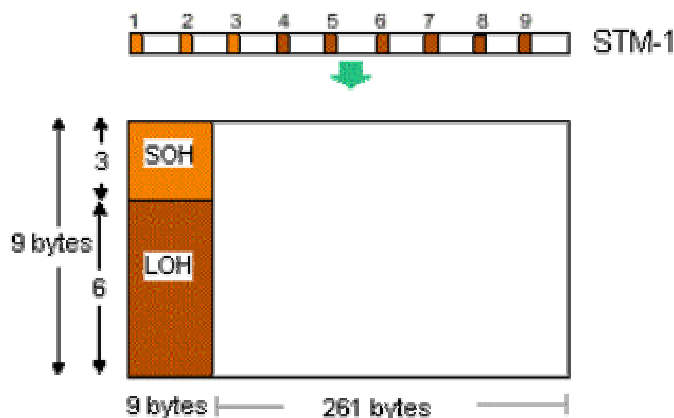


Figura. 5.8. Trama SDH

Salvo el mencionado hecho de que la portadora básica de SDH, STM-1, es la equivalente a la STS-3 de SONET, con lo cual la tasa inicial es de 155.52 Mbps en lugar de 51.84, ambos sistemas crecen de manera análoga hasta alcanzar el máximo de 2.488 Gbps.

Las tasas de velocidad típicas para STS (Synchronous Time Stamps) y STM (Synchronous Transport Module) se muestran en la tabla 5.5.

Tasa Eléctrica	Tasa Óptica	Velocidad
STS-1	STM-0	51.84 Mbps
STS-3	STM-1	155.52 Mbps
STS-9	STM-3	466.56 Mbps
STS-12	STM-4	622.08 Mbps
STS-18	STM-6	933.12 Mbps
STS-24	STM-8	1.244 Gbps
STS-36	STM-12	1.866 Gbps
STS-48	STM-16	2.488 Gbps

**Tabla. 5.5. Comparación de las Tasas de STS y STM**

Solo tres de los niveles jerárquicos están actualmente definidos en el estándar y están disponibles comercialmente: STM-1, STM-4 y STM-16. Las otras tasas serán disponibles según la necesidad. La tabla 5.6. refleja la información útil que se transporta en cada nivel STM 1, STM 4 y STM 16.

Nivel STM	Tasa de datos	Contenido de información útil
1	155.52 Mbps	150.112 Mbps
4	622.08 Mbps	601.344 Mbps
16	2.4883 Gbps	2.40537 Gbps

**Tabla. 5.6. Multiplexado SDH**

La tabla 5.7. es una comparación de los servicios combinados, esta comparación solo incluye información con respecto al tamaño de la trama.



<b>SDH</b>	<b>SONET</b>	<b>Mbps</b>	<b>Tamaño Trama</b>	<b>Fila / Trama</b>	<b>Bytes Encabezado</b>
STM-0	STS-1	51.84	810	90	774
	STS-3	155.52	2,430	270	2,322
STM-1	STS-3c	155.52	2,430	270	2,340
	STS-12	622.08	9,720	1,080	9,288
STM-4	STS-12c	622.08	9,270	1,080	9,288
	STS-48	2,488.32	38,880	4,320	37,152
STM-16	STS-48c	2,488.32	38,880	4,320	37,575
STM-64	STS-192	9,953.28	155,520	17,280	148,608
STM-256	STS-768	39.81312	622,080	69,120	594,432

**Tabla. 5.7. Estructura de Multiplexión basada en G707**

## 5.2.5. COMPARACIÓN DE PDH CONTRA SDH

La jerarquía plesiócrona ha sido ampliamente superada por la síncrona debido a sus deficiencias como por ejemplo:

### 5.2.5.1. Falta de Flexibilidad

La mux/demux es llevada a cabo por un dispositivo llamado ADM ('Add-Drop-Multiplexer'), y es complicado, usando estos equipos, la extracción de un flujo de una determinada tasa a partir de un flujo de tasa varios órdenes mayor; por ello, y tal como vemos en la ilustración, es imperativo ir demultiplexando sucesivamente hasta la tasa final y a continuación, volver a multiplexar en el sentido inverso hasta alcanzar otra vez la tasa original.

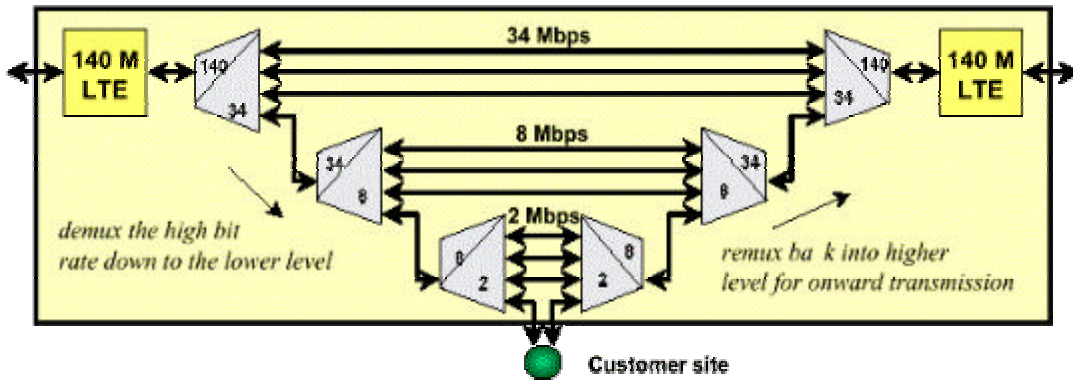


Figura. 5.9. Falta de Visibilidad de PDH

**5.2.5.2. Sin Monitorización**

Actualmente PDH no dispone de ningún estándar para la monitorización y control del rendimiento y funcionamiento de los canales, ni siquiera mediante el uso reservado de canales para tal fin.

No ofrece, pues, un soporte adecuado para OAM (Operaciones, Administración y Mantenimiento).

**5.2.5.3. Ausencia de Especificación de Líneas de Salida**

Aunque PDH especifica el formato exacto de cada flujo de bits a medida que se crece en la jerarquía, no hace tales especificaciones sobre el flujo de bits en las líneas terminales, con lo que cada fabricante puede usar su propio interfaz y código para dichas líneas, haciendo imposible la interconexión a partir de varios productos incompatibles.

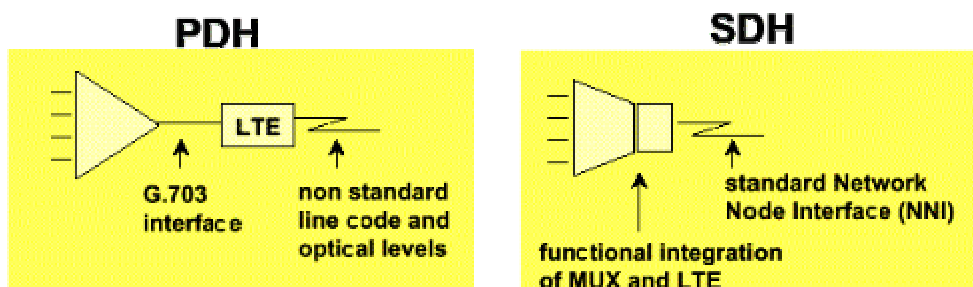


Figura. 5.10. Especificación de Líneas de Salida

A continuación se encuentran tabuladas a manera de comparación algunas de las propiedades o características más importantes de cada tipo de jerarquía, sincronía y plesiócrona, que ayudan a reflejar y comprender de mejor manera las diferencias que ellas presentan.

	<b>PDH</b>	<b>SDH</b>
<b><i>Entramado</i></b>	Octeto a octeto en el nivel básico. Bit a bit en los posteriores.	Octeto a octeto en todos los niveles.
<b><i>Duración de la trama</i></b>	Distinta en cada nivel	125µs en todos los niveles
<b><i>Recuperación de carga</i></b>	Alineándose a la trama de cada nivel y deshaciendo la multiplexión hasta llegar a la carga.	Identificando la posición de la carga, gracias a los punteros. Inserción y extracción rápida de tributarios.
<b><i>Canales de servicio</i></b>	Baja capacidad de los canales de servicio. Se forman multitramas para utilizar los bits de reserva.	Canales de servicio de gran capacidad. Útiles para labores de mantenimiento y gestión de red.
<b><i>Interfaces normalizados</i></b>	Sólo los eléctricos.	Se definen interfaces ópticos para altas velocidades.
<b><i>Capacidad de transporte de carga</i></b>	Limitada.	Muy variada , manejo muy flexible de capacidades diversas: PDH, ATM, TV
<b><i>Usado preferentemente en enlaces</i></b>	Internacionales	Nacionales

**Tabla. 5.8. Comparación de Sistemas PDH y SDH y Capacidades G707**

### 5.2.6. WDM – WAVELENGTH DIVISION MULTIPLEXING

Los dos métodos tradicionales para la multiplexación de señales en un sistema de fibra óptica que utiliza luz coherente (láser) han sido TDM (Time División Multiplexing) y FDM (Frequency Division Multiplexing), al que se viene a añadir WDM. Al contrario que las otras técnicas, WDM suministra cada señal en una frecuencia láser diferente, de tal manera que puede ser filtrada ópticamente en el receptor.

Una propiedad fundamental de la luz establece que ondas de luz individuales de diferentes longitudes de onda no interfieren unas con las otras dentro del medio. Los láseres son capaces de crear pulsos de luz con una longitud de onda muy precisa. Cada longitud de onda individual de luz puede representar un canal de información diferente. Combinando pulsos de luz de diferentes longitudes de onda, pueden transmitirse simultáneamente muchos canales a través de una sola fibra.

A pesar de existir desde hace varios años, hoy en día es uno de los temas de mayor interés dentro del área de la infraestructura de redes ópticas. En los Estados Unidos, en donde las redes de fibra óptica han evolucionado considerablemente, WDM y sus evoluciones (CWDM y DWDM) se han consolidado como una de las tecnologías favoritas, debido a las enormes ventajas que ofrece en la optimización del uso del ancho de banda.

Su implementación en los mercados de Europa, Asia y América Latina crece día con día, y son cada vez más las redes de cable que la utilizan para ofrecer multi – servicios.



**Figura. 5.11. Como Viaja WDM en una Fibra Óptica**

Para entender el funcionamiento de la tecnología WDM, podemos considerar un enlace punto a punto de fibra óptica, en el que existe una fuente de luz localizada en el extremo transmisor y un fotodetector en el extremo receptor. Las señales originadas por diferentes fuentes ópticas utilizan fibras diferentes y únicas como medio de transmisión. Puesto que toda fuente óptica tiene un ancho de línea limitado, es decir, el rango de longitudes de onda que puede emitir es pequeño, el ancho de banda de la fibra es desperdiciado puesto que únicamente se hace uso de una pequeña porción de éste en la fibra. Los multiplexores WDM permiten acoplar diferentes longitudes de onda dentro de una fibra común. De la misma manera, un dispositivo WDM puede recuperar las longitudes de onda que se transmitieron a través de la fibra óptica.

Cada uno de los canales WDM es diseñado para dejar pasar una longitud de onda o una banda de longitudes de onda en particular. Por ejemplo, un sistema WDM de dos canales podría estar pensado para dejar pasar las bandas de 1310 y 1550 nm. El sistema podría utilizarse para enviar dos señales ópticas a través de una fibra común: una señal con longitud de onda de 1310 nm se transmitiría a través del canal de 1310 nm, y una señal de 1550 nm se enviaría por el canal WDM de 1550 nm.

Los canales WDM se comportan como filtros que únicamente permiten el paso de las señales ópticas especificadas para cada canal, de tal forma que transmitir una señal de 1310nm a través de un canal de 1550nm no funcionaría. A pesar de que actualmente se desarrollan técnicas para alojar más de 2000 canales en una sola fibra, los multiplexores más comunes que existen cuentan con 2, 4, 8, 16, 32 ó 64 canales. Aquéllos que integran dos canales cuyas longitudes de onda se localizan entre las bandas de 1310 y 1550 nm se conocen como WDMs de banda amplia.

La multiplexación por división de longitud de onda densa es una tecnología que combina y separa longitudes de onda, usa múltiples señales ópticas para transmitir muchas longitudes de onda de luz simultáneamente sobre una única fibra óptica, llegando a enviar 32/40/64/80/96 longitudes de onda a 2,5 Gbps y 10 Gbps.

Aun así, pronto se verán los sistemas ultra-densos (UDWDM) con transmisión de 128 y 256 longitudes de onda a 10 Gbps y 40 Gbps por canal, ya que la infraestructura actual de fibra óptica no será suficiente para cubrir la demanda.

El estándar (UIT) define una cuadrícula de longitudes de onda permitidas dentro de la ventana que va desde los 1525 nm hasta los 1565 nm tal y como se muestra en la figura. Así, el espaciado entre dos de estas longitudes de onda permitidas puede ser de 200 GHz (1,6 nm), 100 GHz (0,8 nm), o incluso menos. La técnica WDM se considera "densa" (DWDM) cuando este espaciado es de 100 GHz o inferior.

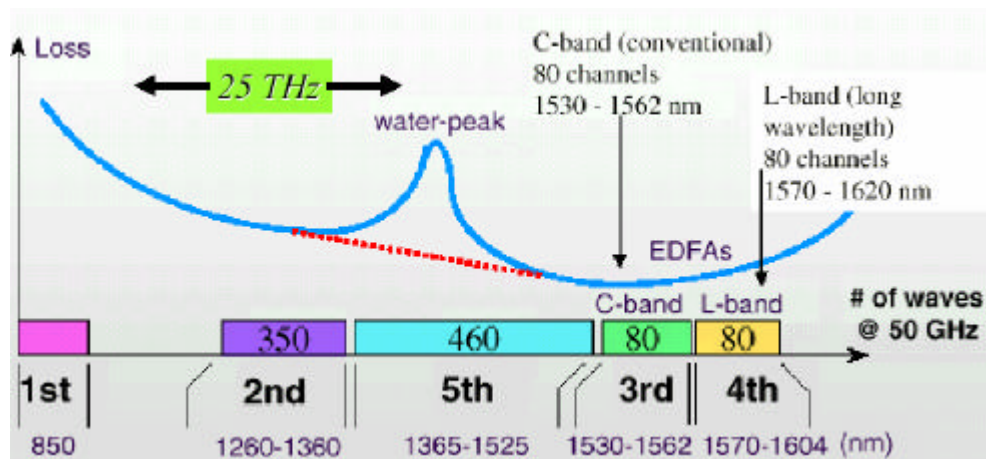


Figura. 5.12. División del Espectro en Ventanas de WDM

Un multiplexor WDM de banda angosta es aquél que integra dos o cuatro canales dentro de la banda de los 1550 nm. DWDM pertenece a esta categoría WDM de banda angosta y está diseñado para un espaciado entre canales de 100 GHz (~0.8nm). Debido a este espaciado, DWDM puede acoplar ocho o más canales dentro de la banda de los 1550nm como se puede observar en la siguiente figura.

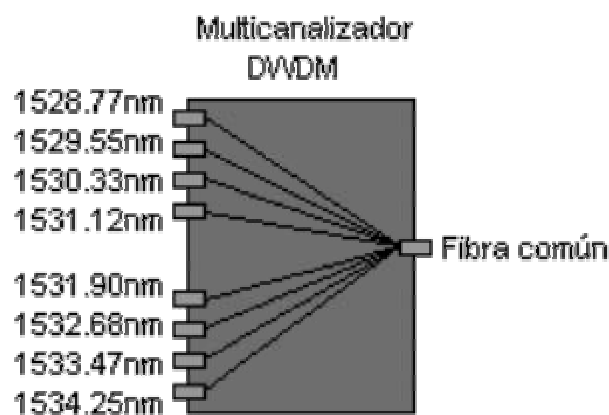


Figura. 5.13. Integración de canales para WDM.

Al momento de implementar tecnologías WDM, es muy importante que los multiplexores utilicen fuentes láser con diferentes longitudes de onda, y que estas fuentes se sintonicen de acuerdo a las longitudes de onda o bandas específicas del multiplexor. De no utilizar las longitudes de onda correctas, el sistema podría no funcionar adecuadamente.

Valdría la pena señalar que antes de implementar tecnología WDM eran necesarias dos fibras en un sistema de comunicación. Una de ellas estaría conectada al transmisor óptico, mientras que la otra provendría del receptor, permitiendo una comunicación bidireccional, conocida como "full-dúplex". Con la llegada de WDM, sólo se requiere de una fibra para proporcionar comunicación "full-dúplex", sin importar el número de canales que se tengan.

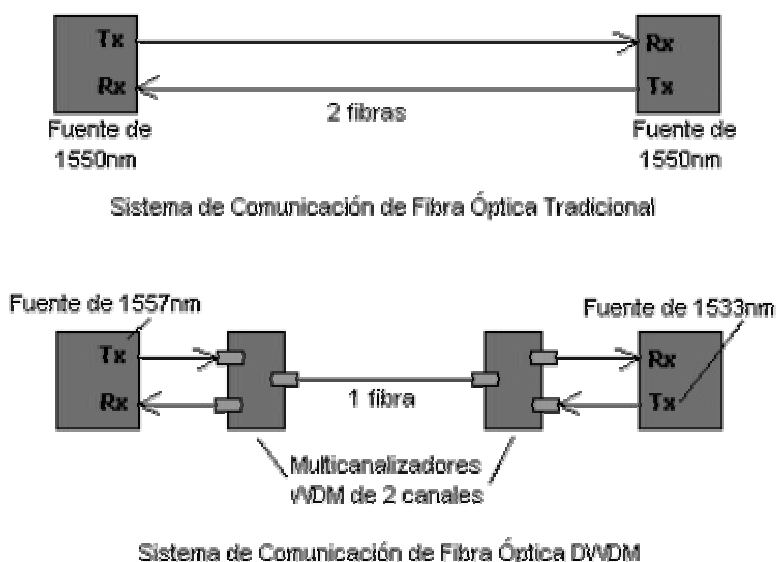


Figura. 5.14. Sistema de Comunicación Normal y Usando WDM

Las bandas que se utilizan en las comunicaciones ópticas tienen sus propios nombres. La banda C, por ejemplo, corresponde a las longitudes de onda entre 1530 y 1565 nm, y sus sub-bandas son conocidas como banda azul (1527.5 – 1542.5 nm) y banda roja (1547.5 – 1561.0 nm). Existen otras bandas conocidas como banda L (1570 – 1610 nm) y banda S (1525 – 1538 nm), actualmente siendo investigadas para su posterior implementación.

Algunos sistemas comerciales tienen un espaciamiento de 50-GHz. Los desarrolladores están hablando de espaciamientos de 25 GHz e incluso 12.5 GHz, pero aun no está definido cuando y si realmente esta tecnología se vuelva de uso comercial.

## Estandares ITU-T

G.692 describe la arquitectura de referencia y la grilla de longitudes de onda para sistemas DWDM

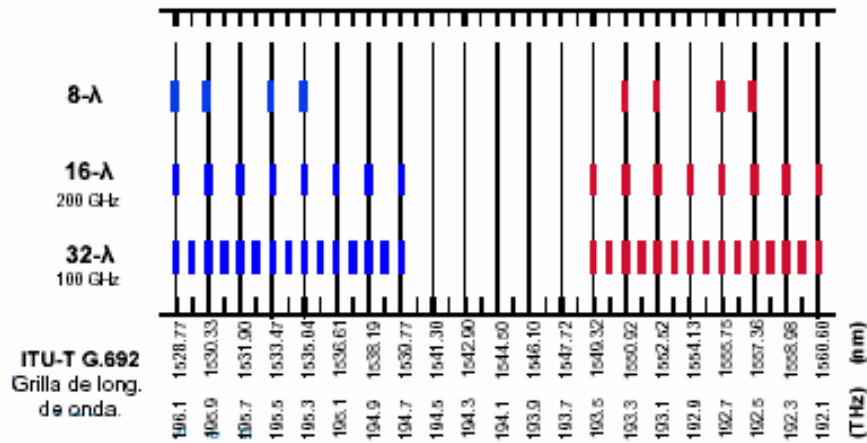


Figura. 5.15. Separación de Canales Según Recomendación ITU-R G.692

Los espaciamientos anchos generalmente son conocidos como Wide Wavelength-Division Multiplexing, o WWDM. Los espaciamientos anchos como 20 a 25 nm son usados en algunos sistemas, especialmente en aquellos operando en las bandas de 850 o 1300 nm.

Pocos sistemas con mayor espaciamiento han sido demostrados con longitudes de onda visibles con fibras plásticas pero no son usados ampliamente. Entre mayor sea el espaciamiento, es más fácil y más barato realizar la multiplexación y demultiplexación ópticas. Por otro lado, el ancho de banda del sistema limita el rango total de longitudes de onda utilizables.

Las técnicas WDM se especifican en términos de la longitud de onda de sus canales (en nanómetros) y con base en su configuración de transmisión – recepción. En ocasiones suele especificarse la longitud de onda de un canal como su frecuencia correspondiente en terahertz (THz). La relación entre frecuencia y longitud de onda se da por la siguiente expresión, en la que el valor 299792 es el estimado de la velocidad de la luz en el vidrio ( $\sim 2.99 \times 10^8$  m/s), con el punto decimal ajustado para su correcto uso en la fórmula:



$$f \text{ (THz)} = \frac{2,99 \cdot 10^8 \text{ (m/s)}}{I \text{ (nm)}}$$

A partir de esta expresión se podrá comprender que para un sistema DWDM, cuya separación de canal, según el estándar de la UIT, es de 100 GHz (~0.8nm), dos canales adyacentes, el primero con una frecuencia de 192.0 THz y el segundo de 192.1 THz, cuentan con longitudes de onda respectivas de 1561.42 nm y 1560.61 nm.

Mientras dos canales adyacentes se encuentren más cerca uno del otro, menor será la banda espectral correspondiente a cada uno de ellos, lo cual hace posible alojar más canales dentro de una misma fibra. Sin embargo, no es el número de canales lo más importante, sino la velocidad que cada uno de ellos puede alcanzar y la flexibilidad que esta tecnología ofrece a los operadores en términos de escalabilidad. Por poner un ejemplo, un sistema cuya estrategia de crecimiento sea gradual, podrá iniciar implementando DWDM a 100 Mbps por canal e incrementar la capacidad de cada uno hasta más de 40 Gbps.

Su capacidad y flexibilidad permiten integrar el tráfico de una variedad de redes diferentes, incrementando el número de usuarios, proveyendo aplicaciones y servicios complejos y acelerando las tasas de transmisión. Definitivamente se trata de una tecnología prometedora para los sistemas de fibra óptica.

Se puede pensar en un multiplexor y demultiplexor como un espejo de un mismo dispositivo. El multiplexor toma longitudes de onda separadas y las combina, y el demultiplexor toma las señales combinadas y las separa. Las principales consideraciones de operación difieren entre ambos. Los Multiplexores deben tener una baja pérdida de inserción y evitar la retrodispersión de cualquiera de los transmisores. Los demultiplexores en cambio deben separar canales de forma confiable con poca fuga de luz de un canal óptico a un canal adyacente. En la práctica, ambos dispositivos pueden usarse como un espejo del otro aunque muchas de las veces los multiplexores tienen un mayor espaciamiento de canal que los demultiplexores para reducir las pérdidas por inserción.

El OADM proporciona una función diferente, separando una o más longitudes de onda de una señal combinada, así ellas pueden ser sacadas en cualquier parte a lo largo del

sistema. También se pueden añadir señales transmitidas desde una estación a lo largo del sistema en los canales vacíos.

La demultiplexación óptica requiere que la separación de canales sea lo más limpia posible, proporcionando un alto grado de aislamiento entre ellos. Este aislamiento debe ser entre 20 y 40 dB, es decir, la entrada de canales adyacentes debe estar reducida de 20 a 40 dB bajo la entrada del canal deseado.

La mejor forma de ver como trabaja esto es considerar la transmisión de un demultiplexor como función de la longitud de onda para un puerto de salida, es decir un canal óptico individual. Idealmente, el puerto debe transmitir toda la luz en el centro del canal óptico, y nada de luz fuera de él. Normalmente el pico de transmisión no es del 100%, con una pérdida típica de 3 a 5 dB.

En los sistemas actuales, algunos de los canales no están poblados por transmisores y receptores, es decir, el espacio está disponible pero no es usado por el momento. Además, algunos diseños dejan intencionalmente algunos espacios para evitar solapamientos de las señales por dispersión. Algunos sistemas WDM dejan algunos huecos entre bloques de longitudes de onda, dejando libres uno o dos espaciados de 100 GHz entre grupos de ocho canales ópticos.

En distancias cortas, como es en el entorno de una oficina, la atenuación de la fibra (mínima para una longitud de onda de 1,55 (mm) y la dispersión (mínima para 1,3 (mm) no presenta un gran problema, pero a distancias mayores, como las que se requieren en los enlaces de comunicaciones a larga distancia, realmente lo es y se requiere el uso de amplificadores/repetidores que regeneren la señal cada cierta distancia. Por ejemplo en los cables transatlánticos se colocan repetidores cada 75 km que, primero, convierten la señal óptica degradada en eléctrica, la amplifican y la vuelven a convertir en óptica mediante un diodo láser, para inyectarla de nuevo en la fibra óptica, todo un proceso complejo y que introduce retardos debido a los dispositivos electrónicos por los que ha de pasar la señal.

Este inconveniente se evitaría si todo el camino pudiese ser óptico (all-optical), algo que ya es posible gracias a los resultados obtenidos, hace ya más de una década, por investigadores de la Universidad de Southampton, que descubrieron la manera de

amplificar una señal óptica en una longitud de onda de 1,55 mm haciéndola pasar por una fibra de 3 metros de longitud dopada con iones erbio e inyectando en ella una luz de láser a 650 nm (fenómeno que se conoce como bombeo o pumping).

### 5.3. REFERENCIAS

#### 5.3.1. LIBROS

- TOMASI, Wayne, **Sistemas de Comunicaciones Electrónicas**, Segunda Edición, Prentice Hall, México 1996, páginas 519-528.
- COUCH, Leon, **Sistemas de Comunicación Digitales y Analógicos**, Quinta Edición, Prentice Hall, México 1998, páginas 645-672.
- ESPE, **Comunicación Digital**, apuntes de la asignatura, semestre septiembre 2004 – marzo 2005.

#### 5.3.2. INTERNET

- <http://trajano.us.es/~isabel/publicaciones/pdh.pdf>, Jerarquía Digital Plesiócrona PDH.
- <http://trajano.us.es/~isabel/publicaciones/multiplexion.pdf>, Multiplexión PDH, SDH.
- [http://200.14.84.223/apuntesudp/docs/civil\\_inf/\(INF2010\)Sistemas\\_Comunicaciones/\(01-07-2004\\_-\\_01-07-23\)-sistemas\\_de\\_comunicaciones-guia-apunte-id1924.pdf](http://200.14.84.223/apuntesudp/docs/civil_inf/(INF2010)Sistemas_Comunicaciones/(01-07-2004_-_01-07-23)-sistemas_de_comunicaciones-guia-apunte-id1924.pdf), Sistemas de Multiplexión, PDH, SDH, SONET.
- <http://com.uvigo.es/asignaturas/scvs/trabajos/curso9900/tdm/Contenido.html>, Multiplexación.

## CAPÍTULO VI

### SPREAD SPECTRUM

#### 6.1. INTRODUCCIÓN.

El interés de mejorar la transmisión de una señal de un punto a otro ha llevado a intentar diversos sistemas de comunicaciones. Al principio bastaba con que la señal se recibiera aceptablemente; luego se insistió en reducir el ancho de banda o de subirlo un poco a cambio de relación señal a ruido. Durante la segunda guerra mundial lo más importante era que las comunicaciones no fuesen interceptadas.

La tecnología de **Espectro Ensanchado** o **Spread Spectrum (SS)** fue descrita por primera vez por la actriz Hedy Lamarr y el músico George Antheil, en 1941, como un enlace seguro de radio para controlar torpedos, bajo la patente americana número 2.292.387. En la que se incluyen los fundamentos sobre la teoría de espectro disperso en los cuales al dispersar el espectro a través del uso de un código personalizado, se lograba un aumento del ancho de banda pero también se fortalecía la transmisión frente al ruido y a las interferencias, intencionales o no. Su sistema fue la base de la técnica conocida como Frequency Hopping (FHSS).

No fue tomada muy en serio por el ejercito americano hasta la década de los 80s donde recobró importancia y se torno ampliamente popular para aplicaciones que involucran radio comunicaciones en ambientes hostiles, y hacia la década de los 90s aparecen los primeros equipos que explotan esta técnica.

Hoy en día la tecnología de espectro disperso se aplica en telefonía celular de tercera generación 3G, transmisión de datos inalámbrica, redes de comunicación personal (PCN),

redes de área local inalámbricas WLAN (IEEE802.11a, IEEE802.11b, IEEE802.11g), Sistemas de Posicionamiento Satelital (GPS) y Bluetooth.

Un sistema debe cumplir con ciertos requerimientos para ser considerado como de Spread Spectrum, como:

- La señal ocupa un ancho de banda mucho mayor al del mínimo necesario para enviar información.
- El ensanchamiento se encuentra realizado en términos de una señal ensanchadora, llamada **señal de código**, independiente de los datos.
- En la recepción, el de-ensanchamiento se realiza mediante la correlación de la señal ensanchada recibida con una replica sincronizada de la señal de código utilizada para ensanchar la de información.

Los esquemas de modulación como FM y PCM también ensanchan el espectro de una señal de información, pero no son SS debido a que no satisfacen las otras condiciones.

En este capítulo se estudiarán las principales características de esta técnica, sus ventajas, la forma como se la realiza y sus clasificaciones. Dentro de sus clasificaciones se analizará las conocidas como: secuencia directa DSSS, saltos de frecuencia FHSS, saltos de tiempo THSS y alguno de los híbridos existentes de estas.

Se finalizará con prácticas en MATLAB que permitan un mejor entendimiento del manejo de la técnica, así como de sus características.

## 6.2. MARCO TEÓRICO

### 6.2.1. DEFINICIONES

Existen varias técnicas de Espectro Ensanchado (SS), cuyo común denominador es el uso de un código o secuencia unido al canal de comunicación. La forma como se inserta este código es lo que define el tipo de técnica de SS que se está utilizando.

Así, un sistema de espectro ensanchando es un sistema de comunicaciones de Radio Frecuencia, en el cual, el ancho de banda de la señal banda base es ensanchado intencionalmente sobre uno mucho mayor inyectando una señal de mayor frecuencia. Como consecuencia de ello, la energía usada es distribuida sobre el ancho de banda más amplio, apareciendo como ruido. La relación entre una señal ensanchada y la original se conoce como ganancia del proceso y generalmente va de 10 dB a 60 dB.

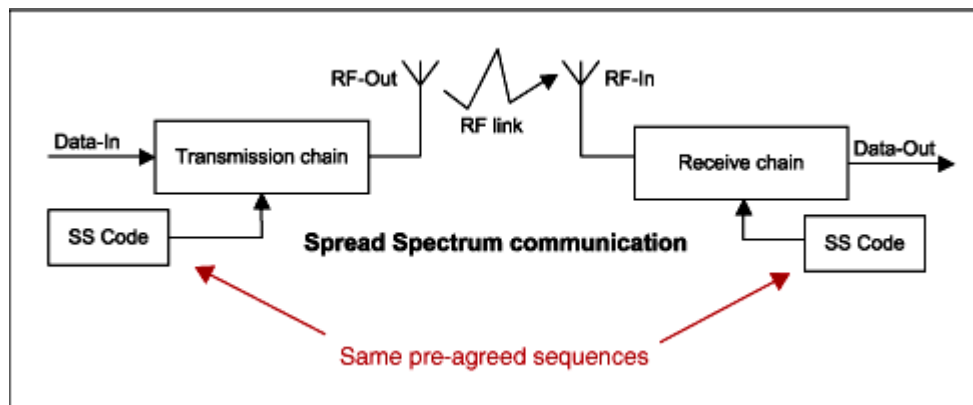


Figura. 6.1. Sistema de Comunicación Spread Spectrum

En la figura 6.2, se muestra los anchos de banda de la señal en un enlace de comunicaciones de SS, luego del ensanchamiento, y en la figura 6.3, el proceso inverso.

En este caso, la modulación SS se aplica sobre una señal modulada convencionalmente como BPSK, y solo se afecta a la señal que se desea ensanchar, pudiendo tener otras que permanecerán tal cual su forma original.

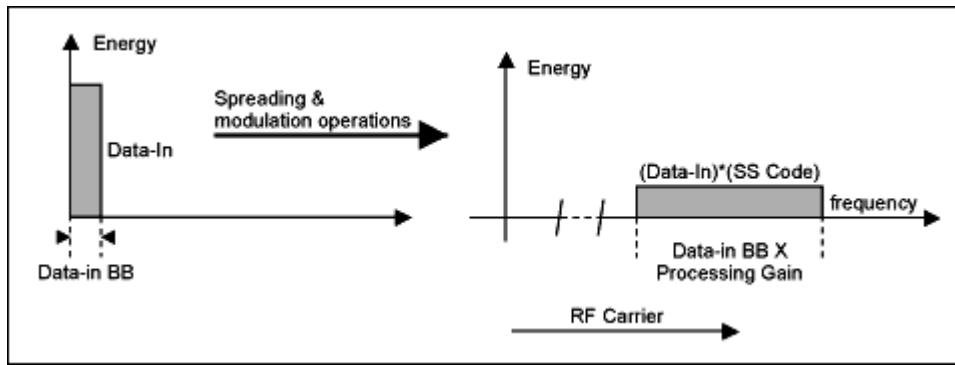


Figura. 6.2. Efecto de Ensanchamiento de una Señal

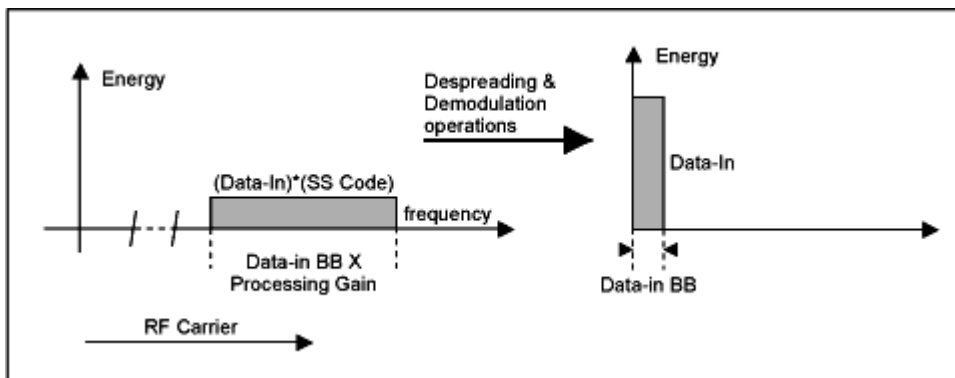


Figura. 6.3. Efecto de De-ensanchamiento de una Señal

Una característica del uso de SS, es que dicho ensanchamiento resulta directamente en el uso de una mayor banda de frecuencias, de forma proporcional a la ganancia de procesamiento, pero ese sobre uso se compensa adecuadamente con la posibilidad de aumentar el número de usuarios que comparten esa banda ampliada de frecuencias.

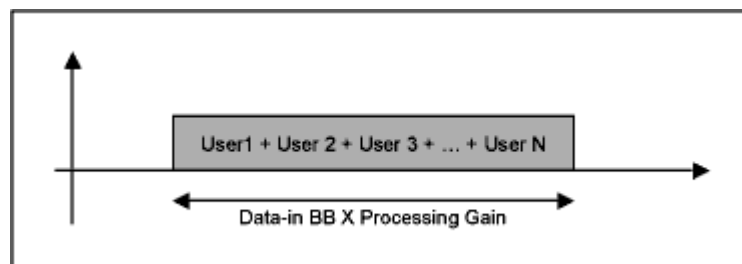


Figura. 6.4. Rendimiento en usuarios del Espectro Ensanchado



### 6.2.1.1. Sistemas de Espectro Disperso y Secuencias Pseudo Aleatorias

Para lograr una mayor ocupación del espectro y además personalizar la transmisión, se puede simplemente multiplicar la señal digital que se quiere transmitir por una secuencia pseudo-aleatoria o de pseudo-ruido (PN):

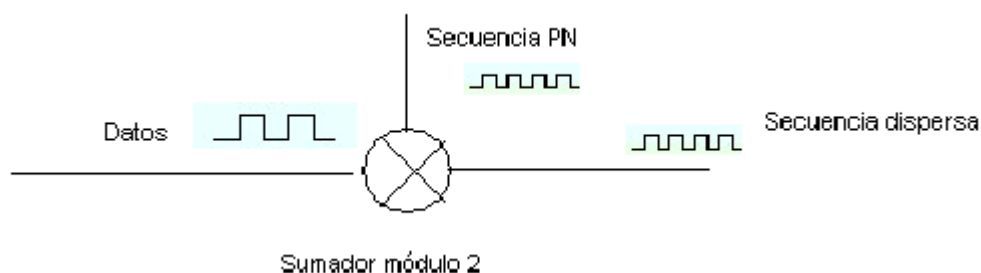


Figura. 6.5. Formación de la Señal Ensanchada

Multiplicar dos señales en tiempo implica hacer una convolución en frecuencia lo cual esparce o dispersa el espectro. Al dispersar el espectro, la Densidad Espectral de Potencia disminuye, disminuyendo así la interferencia entre sistemas; además, se puede multiplexar casi indefinidamente un canal, asignado códigos diferentes a cada usuario.

Se comprende entonces que la secuencia o código que esparce el espectro es de vital importancia, y se define una secuencia pseudos-aleatoria o de ruido (**Pseudo Noise-PN**) como un conjunto de señales binarias, periódicas y de cierta longitud de tal forma que, dentro de cada período, la señal puede aproximarse a una señal aleatoria. Se hace esto para tener la certeza de que la misma secuencia puede generarse tanto en el transmisor como en el receptor. Si fuese totalmente aleatoria esto no sería posible.

Dentro de estas secuencias PN se define un **chip** como la duración de cada elemento dentro de la misma. Se le llamará **T<sub>c</sub>**. Este tiempo será mucho menor que la duración de 1 bit en la secuencia que se quiere codificar.

Existen muchos tipos de secuencias pseudos aleatorias periódicas; las más útiles, por su sencillez de implementación en base a registros de corrimiento, son las secuencias de máxima longitud, conocidas como **m-sequences**.

### 6.2.1.2. Secuencias de Máxima Longitud

Es el tipo más importante de secuencias de PN utilizadas; la denominación de máxima longitud significa que es el código de mayor longitud que se puede generar por medio de un registro de desplazamiento de longitud N, esto es  **$2^N-1$  chips de longitud**. Algunas secuencias podían ser de  $N=7, 15, 31, 63, 127, 255, \dots$ ; las cuales serían secuencias cortas.

Por ejemplo si se usa un reloj de 10MHz, entonces  $T_c=0.1 \mu\text{seg}$ ; si se usa un registro de 41 etapas, la longitud de la secuencia será de  $2.199 \times 10^{12}$ , y  $NT_c= 2.199 \times 10^5$  segundos= 2545 días.

Una secuencia de este tipo puede conseguirse con el siguiente arreglo de registros de desplazamiento y sumadores módulo 2.

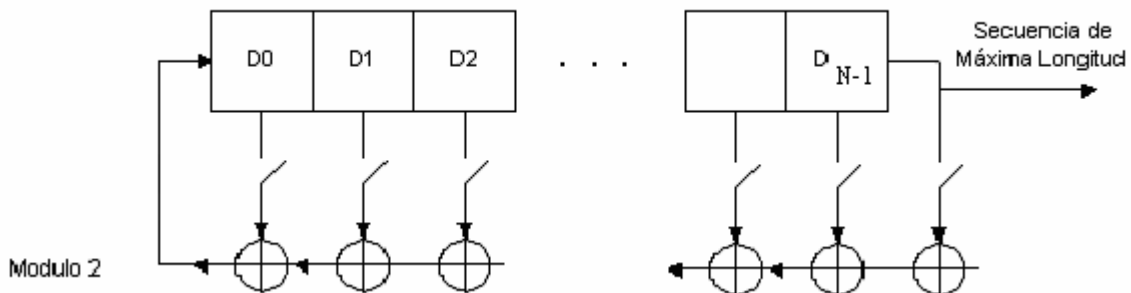


Figura. 6.6. Generación de una Secuencia de Pseudo -Ruido

#### Ejemplo:

Suponga en el siguiente generador, que el estado inicial de los registros es 1000.

Las conexiones se describen como [4,3]. Encuentre la secuencia de salida.

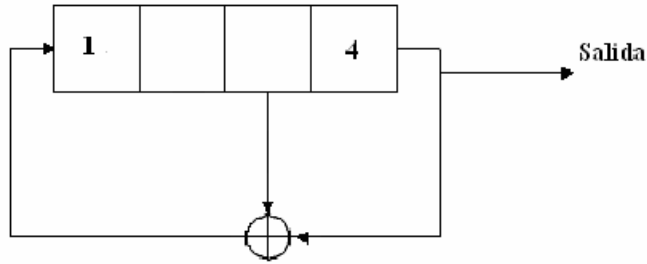


Figura. 6.7. Generador Pseudo-Ruido

Número Desplazamiento	Contenido Registros	Salida Sumadores	Salida
0	1000		
1	0100	0+0=0	0
2	0010	0+0=0	0
3	1001	1+0=1	0
4	1100	0+1=1	1
5	0110	0+0=0	0
6	1011	1+0=1	0
7	0101	1+1=0	1
8	1010	0+1=1	1
9	1101	1+0=1	0
10	1110	0+1=1	1
11	1111	1+0=1	0
12	0111	1+1=0	1
13	0011	1+1=0	1
14	0001	1+1=0	1
15	1000	0+1=1	1

Tabla. 6.1. Generación de Secuencia Pseudo-Ruido

La secuencia de salida es 000100110101111 (El período es  $16-1=15$ )

Dado un arreglo de  $N$  registros, el número de secuencias de longitud  $2^N - 1$  es limitado. Por ejemplo para  $N=4$ , solo hay 2 secuencias de longitud 15; para  $N=5$  y  $N=6$  hay solo 6 secuencias. Habrá mas secuencias si  $N$  es un número primo.

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.

Cuando el período de la  $m$ -secuencia es muy larga, sobre todo comparado con la longitud del mensaje, que llamaremos  $M$ , hay que investigar la correlación sobre este periodo que no es el de la  $m$ -secuencia, a ver que propiedades tiene. Cuando las secuencias son largas 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.

Otro punto interesante de considerar es el comportamiento de la cross-correlación entre las secuencias posibles, ya que si el generador PN se quiere usar para diferenciar usuarios, es deseable que la cross correlación sea baja. Las  $m$ -secuencias no tiene un buen comportamiento respecto a este parámetro.

En aplicaciones donde se desee identificar usuarios asignándoles secuencias o códigos distintos se utilizan otro tipo de secuencias diferentes a las  $m$ -secuencias, por ejemplo secuencias Gold, Walsh o Kasami.

### 6.2.1.3. Ganancia del Procesamiento ( $G_p$ )

Da una idea del aumento del ancho de banda con relación al de la señal en banda base; se define como la razón de las relaciones señal a ruido entre la salida y la entrada al receptor

$$G_p = \frac{SNR_{\text{salidareceptor}}}{SNR_{\text{entradareceptor}}}$$

En la práctica se utiliza la siguiente aproximación equivalente:

$$G_p = \frac{W_c}{W}$$

$$G_p = \frac{2/T_c}{2/T} = \frac{T}{T_c} = N$$

Siendo  $G_p$  la ganancia del proceso,  $W_c$  el ancho de banda de la señal transmitida y  $W$  el ancho de banda del mensaje original.  $N$  es la longitud de la secuencia PN. Esta medida implica sólo la mejora en la relación señal a ruido que llega al receptor del sistema pero no una medida del nivel de interferencia presente.

#### 6.2.1.4. Margen Sobre la Interferencia (J)

Da idea de la máxima interferencia que puede tolerar el sistema; se define como

$$J = G_p - SNR_{\min}$$

Donde  $SNR_{\min}$  es la mínima relación señal a ruido aceptable a la entrada del receptor.

El margen de interferencia indica el límite de interferencia en que puede trabajar el sistema con una tasa de error aceptable, por ejemplo, si  $J=10$  dB entonces la máxima interferencia aceptada deberá tener un nivel de 10dB ya que una interferencia mayor a este valor afectará el enlace en cuanto a probabilidad de error.

#### 6.2.1.5. Secuencias Gold

En algunas aplicaciones no solo es importante lograr ciertas características de correlación de la secuencias PN sino también es importante que los valores de correlación cruzada cumplan ciertos requisitos. En los años 60 Gold y Kasami probaron que existían ciertos pares de  $m$ -secuencias, que tenían 3 valores de cross-correlación:

$$-1, -t(m) \text{ y } t(m)-2,$$

Donde  $t(m)$  es igual a

$$2^{0.5(m+1)} + 1 \text{ para } m \text{ impar y,}$$

$$2^{0.5(m+2)} + 1 \text{ para } m \text{ par.}$$

A dos secuencias que cumplan estas condiciones se les denomina secuencias preferidas. Suponga que se tiene dos secuencias de longitud  $n$ ; se puede construir un conjunto de secuencias de longitud  $n$  sumando, en módulo 2, una de ellas con versiones desplazadas de la otra, o al revés. La nueva secuencia tendrá período  $n = 2^m - 1$ . Las secuencias originales generadoras también forman parte de ese conjunto de manera que surgen  $n+2$  secuencias, llamadas secuencias Gold. Un ejemplo de la combinación de dos secuencias preferidas para generar una secuencia Gold es el de la figura 6.8.

Algo similar hizo **Kasami**. Aquí se parte de una  $m$  secuencia, a la que llamaremos **a**, y en base a ella se forma otra secuencia **b** tomando 1 de cada  $2^{0.5m} + 1$  bits de **a**. La secuencia **b** también es periódica, pero su período es de  $2^{0.5m} - 1$ . Luego se combinan **a** y los desplazamientos cíclicos de **b**. Aparecen entonces  $2^{0.5m}$  secuencias de longitud  $2^{m-1}$ , incluyendo **a**. Estas son las llamadas secuencias Kasami; tanto la autocorrelación, como la cross correlación de estas secuencias toman valores de entre los tres siguientes:

$$-1, -(2^{0.5m} + 1) \text{ y } 2^{0.5m} - 1.$$

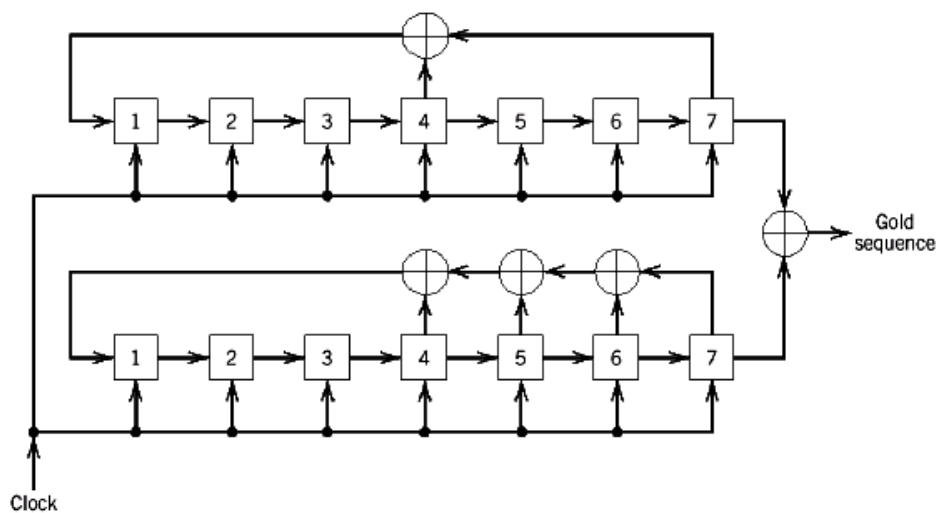


Figura. 6.8. Generación de Secuencias Gold

### 6.2.1.6. Secuencias Walsh

Las secuencias Walsh tienen la ventaja de ser ortogonales por lo tanto estará libre de interferencias de acceso múltiple. Pero existen sin embargo algunas desventajas como no

tener un simple y estrecho pico de correlación, el ensanchamiento no se produce sobre todo el ancho de banda, en su lugar, la energía es dispersa sobre un número discreto de componentes de frecuencia; como se puede observar en la figura 6.9.

La ortogonalidad es afectada por la falta de correlación cruzada y por las propiedades de los canales como multi trayectos. En los sistemas prácticos se aplica ecualización para recuperar la señal original.

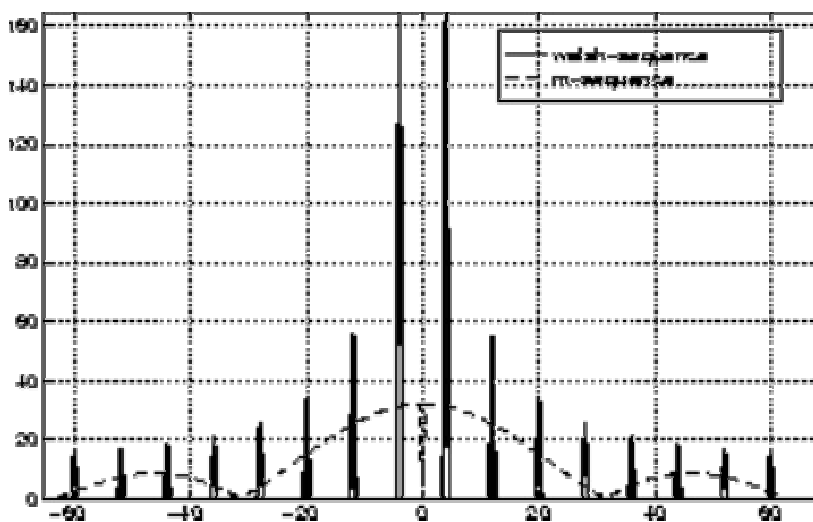


Figura. 6.9. Comparación de Secuencia M y Walsh en Dominio de la Frecuencia

Estas desventajas hacen que las secuencias Walsh no sean aconsejables para sistemas que no sean celulares. Las secuencias Walsh son utilizadas por ejemplo en CDMA de múltiple portadora y en el sistema celular CDMA IS-95.

En estos sistemas, todos los usuarios están sincronizados entre si, y en IS-95 se usa una combinación de secuencias Walsh con registros de desplazamiento para habilitar dicha sincronización.

#### 6.2.1.7. Técnicas de Modulación de Ensanchamiento.

Las diferentes técnicas de SS se distinguen de acuerdo al punto del sistema donde se inserta el código pseudo aleatorio (PRN) en el canal de comunicaciones.

Si el PRN se inserta en el nivel de datos, se tiene la forma de **secuencia directa** de espectro ensanchado DSSS. En la práctica, la secuencia PRN se mezcla o multiplica con la señal de información, dando la impresión de que el flujo de datos está inmerso en el PRN. Si el PRN actúa a nivel de la portadora de frecuencia, se tiene la forma de **salto de frecuencia** FHSS, que fuerza a la portadora a cambiar o saltar de acuerdo con la secuencia pseudo-aleatoria. Si el PRN actúa como una compuerta on/off sobre la señal transmitida, se tiene la técnica de **salto en el tiempo** THSS. Existen también técnicas híbridas de que resultan de mezclar las técnicas descritas como DSSS + FHSS. Siendo DSSS y FHSS las técnicas más utilizadas en la actualidad.

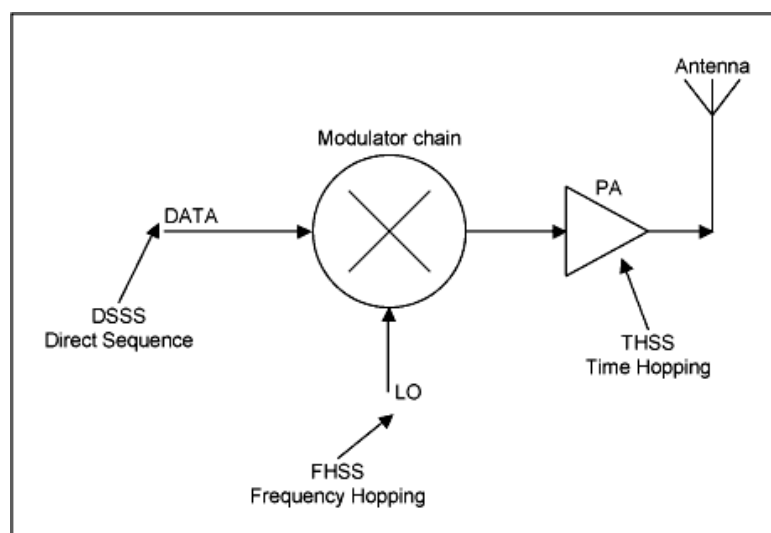


Figura. 6.10. Diferentes Técnicas de SS

### 6.2.2. SECUENCIA DIRECTA

En esta técnica la PRN se aplica directamente a los datos entrando en el modulador de la portadora, de forma que el modulador observa una tasa de bit mucho más larga que la de la señal original y que corresponde a la tasa de chip de la secuencia PRN. El resultado de modular una portadora RF con tal secuencia de código produce una DSSS con un espectro de frecuencia  $\text{sinc}^2x$  centrado en la frecuencia de portadora.

El lóbulo principal de este espectro tiene un ancho de banda igual al doble de la tasa de chip del código modulante. En la figura 6.11, se ilustra la forma más común de señales



DSSS, la cual presenta ligeras variaciones dependiendo del tipo de portadora y de modulación utilizada. La señal de la figura es una BPSK, la cual es la modulación más utilizada en los sistemas de secuencia directa.

Para representar la señal DS/SS (antes de la modulación) se puede usar la siguiente expresión:

$$r_n = \mathbf{e}b(t)c_n + w_m \quad n = 0, 1, \dots, N-1$$

Donde  $\mathbf{e}$  es la energía por bit de la señal,  $\mathbf{b}(t)$  representa la señal de información y es una señal polar  $\{\pm 1\}$ ,  $\{\mathbf{c}_n\}$  es una secuencia PN polar con valores  $\{\pm 1\}$  equi-probables, y de período  $N$ , es decir que  $\mathbf{c}_{i+N} = \mathbf{c}_i$ . Cada bit de información duración  $T$  se codificará con  $N$  chips de duración  $T_c = T/N$ , el incremento de la razón de muestreo de la señal será entonces de  $N$ . Finalmente  $w_m$  representa el ruido blanco gaussiano aditivo (AWGN) de media cero y desviación estándar  $s$ .

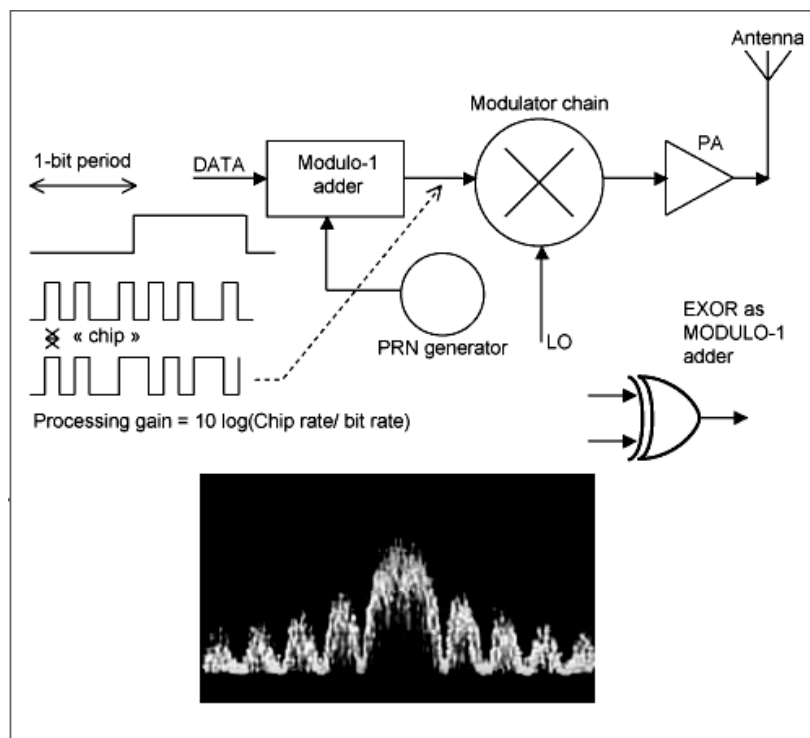
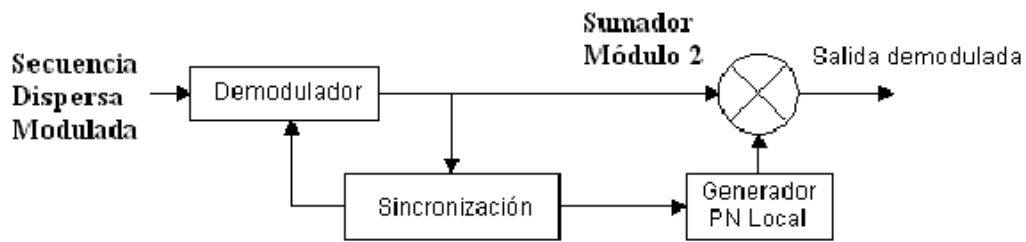


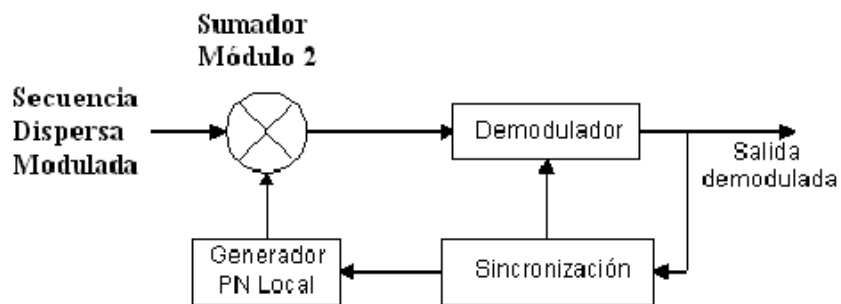
Figura. 6.11. Espectro Ensanchado en Secuencia Directa

La demodulación de la señal se realiza en el receptor por medio de una correlación; esto es multiplicar la señal por una réplica de la secuencia PN.

Para rescatar la señal  $b(t)$  se pueden usar cualquiera de los dos esquemas mostrados en la figura 6.12.



a) Demodulación antes de multiplicar por el código



b) Multiplicación por el código antes de demodular

Figura. 6.12. Demoduladores de DSSS

El ancho de banda de la señal modulada es  $2/T_c$  o  $2 \cdot f_{chip}$ . Por lo tanto resulta mucho mayor que el ancho de banda que produciría modular directamente los datos originales.

El principal problema de los sistemas SS-DS es que el rechazo a la interferencia depende del valor de correlación que resulte entre el código original y la secuencia interferente.

### Ejemplo

En un sistema DS/SS con modulación PSK se desea conseguir una tasa de error por bit de  $10^{-6}$ . El factor de expansión posible permitido es de 1000. Determine el margen sobre la interferencia.

En PSK para lograr  $P_e=10^{-6}$  es porque  $10\log(E_b/\sigma^2)=10.5$  dB. Como la ganancia de procesamiento es 1000 o, lo que es lo mismo, 30 dB, entonces  $10\log(J/SR)=30-10.5=19.5$  dB.

Este es el margen de tolerancia ante las interferencias que puede tener este sistema. Para mejorarlo se puede utilizar codificación de canal.

### 6.2.3. SALTO DE FRECUENCIA

En los sistemas DS/SS a mayor dispersión del ancho de banda mas fortaleza frente a las interferencias. Sin embargo esto también aumentaría la complejidad del hardware. Eso da pie a otro tipo de sistemas conocidos como de multiplicidad de frecuencias. Este método realiza exactamente lo que su nombre dice, hace que la portadora realice saltos de frecuencia a frecuencia sobre una banda amplia de frecuencias de acuerdo a la secuencia definida por el PRN. La velocidad a la cual se realizan los saltos depende de la tasa de datos de la información original, pero se pueden distinguir entre salto de frecuencia rápido FFHSS y salto de frecuencia lento LFHSS, siendo este último el más utilizado y permite el paso de algunos bits de datos a modularse en una misma frecuencia. Por otro lado FFHSS se caracteriza por realizar varios saltos dentro de la duración de cada bit de datos.

El espectro transmitido del salto de frecuencias es absolutamente diferente del de una secuencia directa. En lugar de una envolvente  $\text{sinc}^2x$ , la salida de un salto de frecuencia es plana sobre la banda de frecuencias utilizadas como se muestra en la figura 6.14.

El ancho de banda de una señal FHSS es simplemente  $N$  veces el número de ranuras de frecuencia disponibles, donde  $N$  es el ancho de banda de cada canal de salto.

Estos saltos de frecuencia se hacen con dos objetivos fundamentales: incrementar la capacidad del sistema y mejorar su calidad (disminuyendo la tasa de bits errados). La variación en frecuencia podría lucir como la figura 6.13:

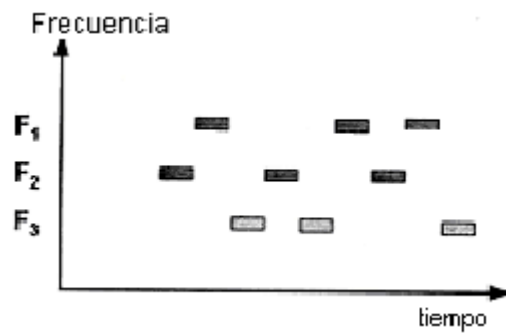


Figura. 6.13. Esquema de Frecuencias para FHSS

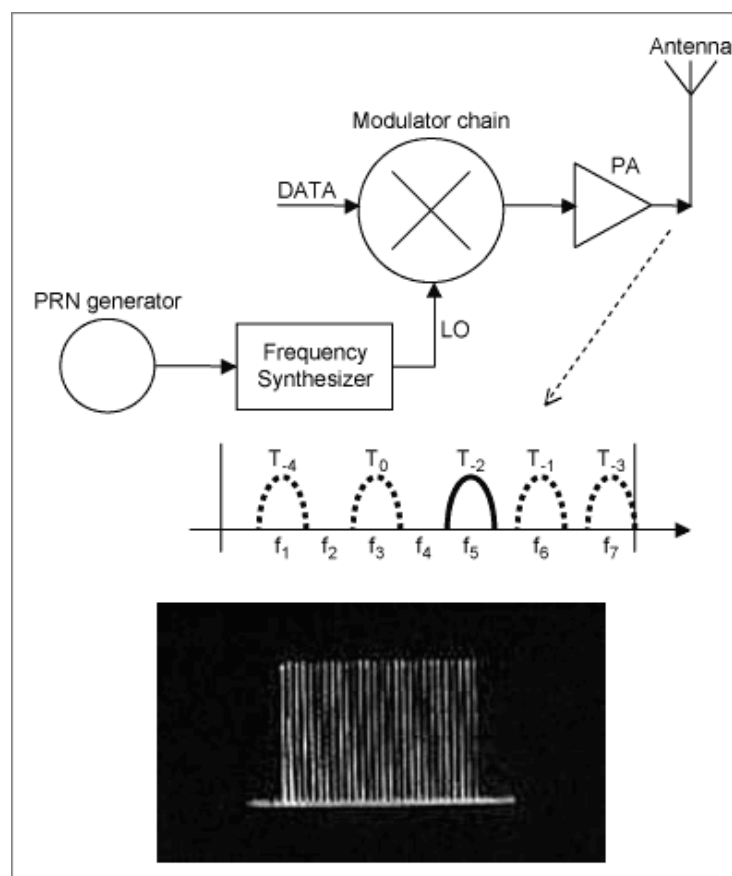


Figura. 6.14. Espectro Ensanchado en Salto de Frecuencias FHSS

Dependiendo de la forma como varía la frecuencia en un sistema FS/SS, se definen el FH cíclico y el FH aleatorio.

**Frequency Hopping Cíclico:** Las frecuencias varían de una manera secuencial y repetitiva. Provee buena diversidad de frecuencia pero no de interferencia.

**Frequency Hopping Aleatorio:** La frecuencia cambia según una secuencia pseudoaleatoria. Tiene mejor desempeño frente a las interferencias.

Usualmente la modulación es M-FSK. La señal modulada es multiplicada por un tono de una de entre  $2^N$  posibles frecuencias generadas aleatoriamente. Se puede imaginar un sistema FH/SS proveniente de modular en 8FSK ( $M=8$ ,  $N=3$ ); el generador de PN es de tres bits y cambia cada cuatro símbolos del modulador 8FSK. Este es un sistema FH lento porque  $T_h > T_s$ . En este caso  $T_c = T_s$ .

Para que sea un sistema de mínima separación entre las frecuencias, la distancia entre las frecuencias de símbolo será  $1/T_s$ . Entre una y otra el PN las hará moverse entre  $f_0$  y  $f_{0+(1/T_s)}$ . En la figura 6.15, se muestra que los símbolos del generador PN ocurren en el siguiente orden: 000 110 010 100.

Dentro de cada uno de estos símbolos del PN la salida del modulador cambia de la siguiente forma:

- Dentro del PN 000, el modulador ofrece las frecuencias asociadas a los símbolos: 000,001,010,011.
- Dentro del PN siguiente aparecerán las frecuencias asociadas a los símbolos: 100, 101, 110, 111.
- Luego, para el PN 010, los símbolos que manejan al modulador son 101, 101,101,101

Cada  $T_h$  segundos el sintetizador manejado por el PN genera una de  $2^n - 1$  frecuencias.

El ancho de banda de esta señal será  $(2^n - 1) f_s + 2f_s = (2^n + 1) f_s$ .

El ancho de banda de la señal original es  $1/T_s$ . La ganancia de procesamiento será entonces:

$$G_p = (2^n + 1)$$

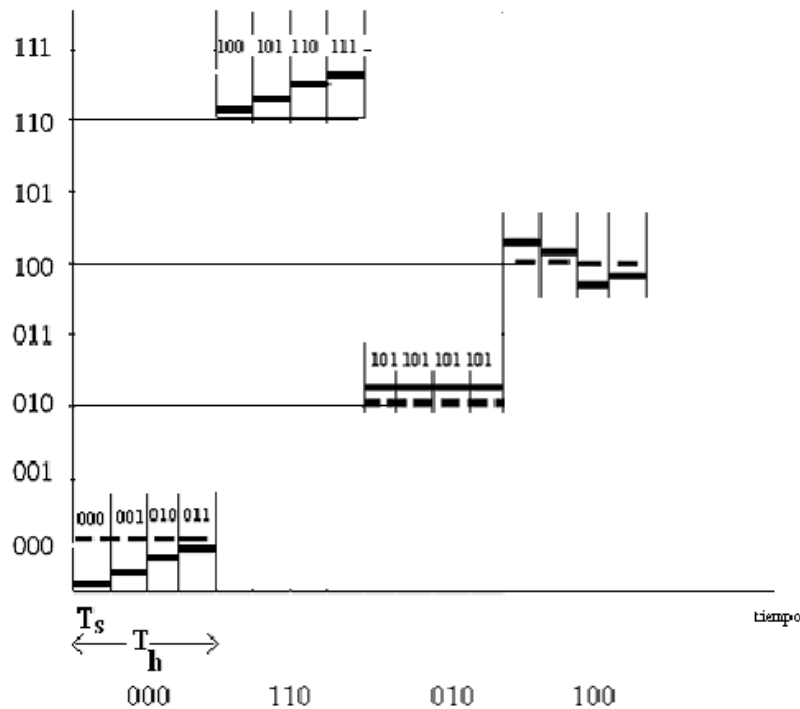


Figura. 6.15. Salidas de LFHSS para 8FSK

#### 6.2.4. SISTEMA HIBRIDO DS/(F)FH

La técnica de espectro ensanchado DS/FFH es una combinación de las técnicas de secuencia directa y saltos de frecuencia rápida. Un bit de dato es dividido en canales de salto de frecuencia (frecuencias portadoras). En cada canal de salto de frecuencia un código completo PN es multiplicado con la señal de datos como se muestra a continuación en la figura 6.16.

Al tiempo que la secuencia FH y los códigos PN son acoplados, una dirección es la combinación de estas características. Para limitar la posibilidad de que dos usuarios compartan el mismo canal de frecuencia en un determinado periodo de tiempo, las secuencias de salto de frecuencia son seleccionadas de tal forma que dos transmisores con diferentes secuencias FH compartan en la mayoría dos frecuencias al mismo tiempo (los desplazamientos de tiempo son aleatorios).

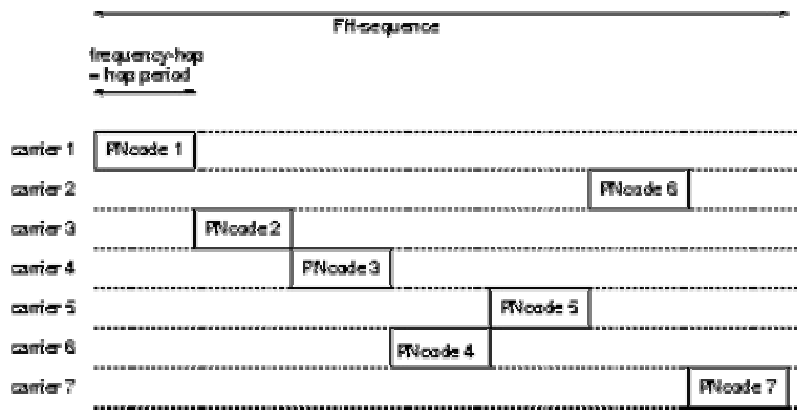


Figura. 6.16. Esquema de Espectro Ensanchado DS/FFH

### 6.2.5. SALTO DE TIEMPO

Este es un método que no ha sido bien desarrollado en la actualidad. Las secuencias on/off son aplicadas en el amplificador de potencia PA son controladas por la secuencia PRN. En la figura 6.17 se muestra el esquema de esta modulación.

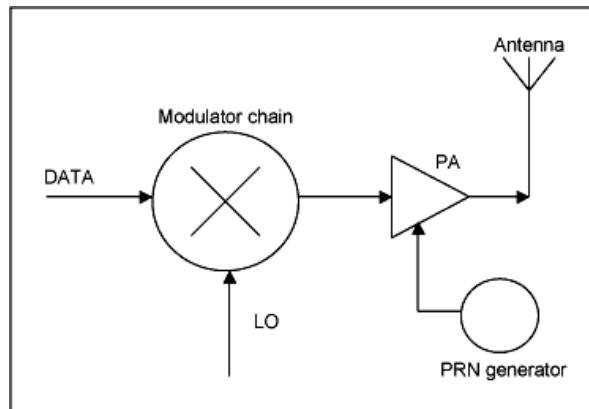


Figura. 6.17. Esquema de Espectro Ensanchado THSS

## 6.3. PRÁCTICAS CON MATLAB

### 6.3.1. SISTEMA DE COMUNICACIÓN CDMA CON SPREAD SPECTRUM

#### 6.3.1.1. Objetivos

- Implementar un sistema de comunicación con Acceso Múltiple por División de Código CDMA, usando la técnica de Espectro Ensanchado.
- Comprender la forma de utilización de los códigos Gold dentro de la técnica de espectro ensanchado.
- Analizar la forma de transmisión y recepción de las secuencias de datos, a las que se les aplica los códigos Gold.
- Verificar el desempeño de este tipo de sistemas frente al incremento de la señal de ruido en un canal de transmisiones.
- Apreciar la influencia en el incremento del error de la comparación de un canal de comunicaciones por dos usuarios que utilizan la técnica de espectro ensanchado.
- Observar el desempeño del sistema frente al incremento del número de usuarios dentro de un mismo canal, interferencia multi usuario.

#### 6.3.1.2. Procedimiento

1. El código de esta práctica se encuentra en el archivo **CDMA\_SS.m**, junto con el cual se encuentran los archivos de las funciones necesarias para su ejecución y que han sido creadas para este caso particular.
2. Se parte de un conjunto de códigos Gold, compuesto por 33 códigos de longitud 31 (realmente se dispone de códigos de longitud 32, pero para efectos de la simulación, se descarta el primer chip de cada uno de ellos para conseguir la longitud deseada de 31 chips).
3. Para ello se realiza la lectura del archivo **gold32**, y se los limita a 31 chips.



4. Primero se realiza la prueba de funcionamiento del sistema de comunicación CDMA con espectro ensanchado, para lo cual:
  - a. Se establece la secuencia de datos que se va a transmitir.
  - b. Se obtiene la secuencia que va a ingresar al canal por medio del comando  $\mathbf{x0} = \mathbf{cdma\_tx(datos0,1,gold31(1,:),0)}$ ; el cual hace uso de la función `cdma_tx()` que requiere como parámetros la secuencia de datos y la secuencia de Gold.
  - c. Se obtiene la salida del detector de correlación por medio del comando  $\mathbf{r0} = \mathbf{cdma\_rx(x0,gold31(1,:))}$ ; en donde se usa la función `cdma_rx()`, en la que se ingresan como parámetros la secuencia ingresada al canal y la secuencia de Gold con la que va a correlacionarse.
  - d. Se obtiene la probabilidad de error del sistema. Los resultados de esto se muestran en la figura 6.18.

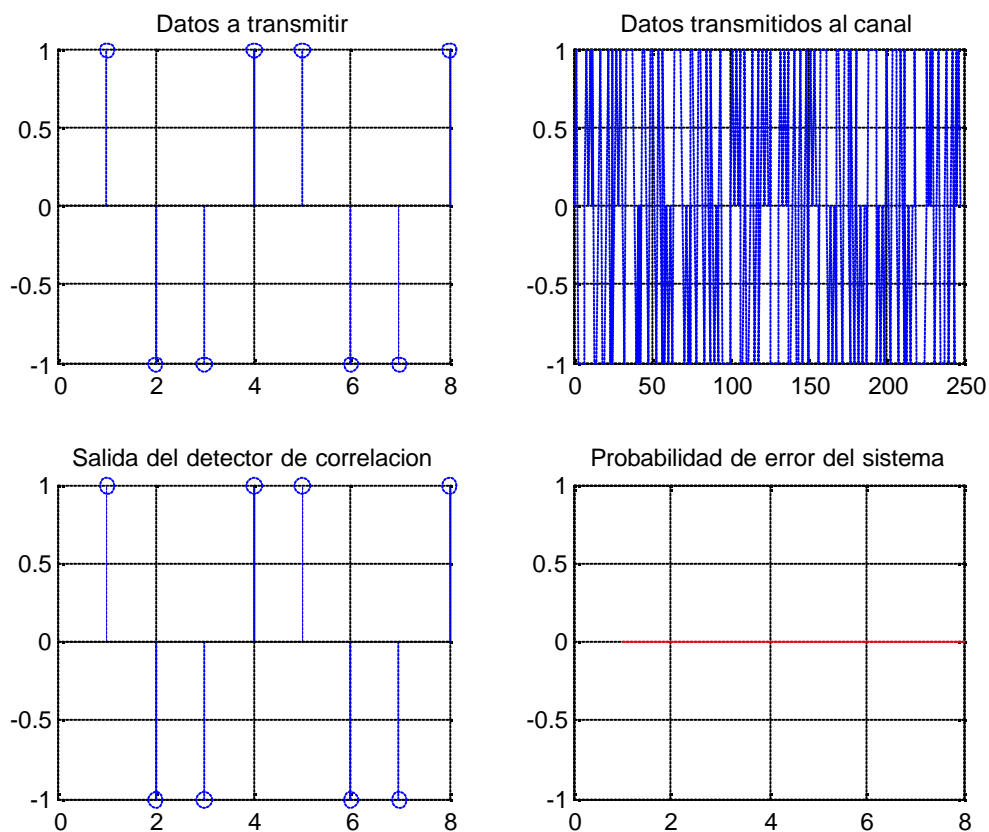
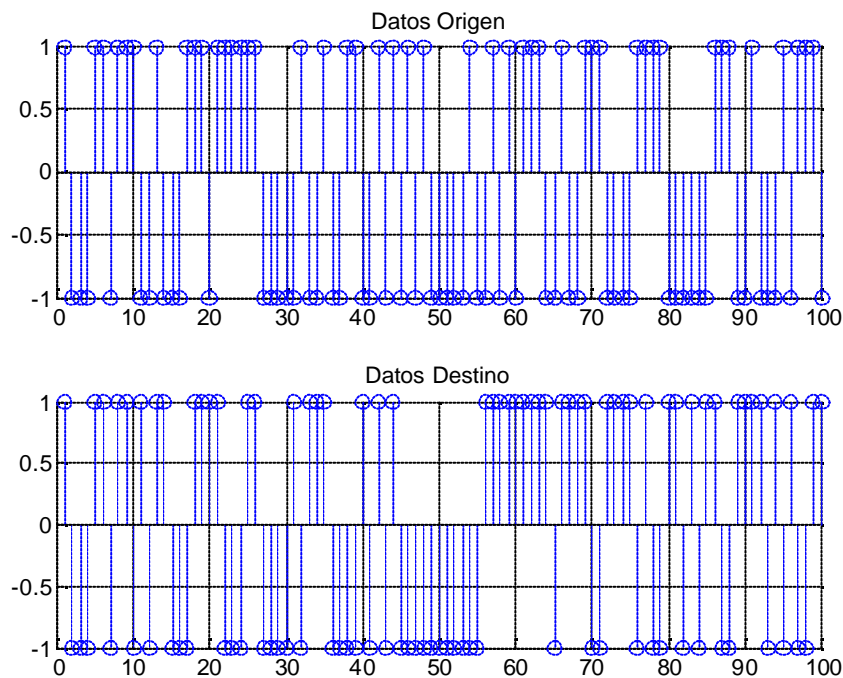
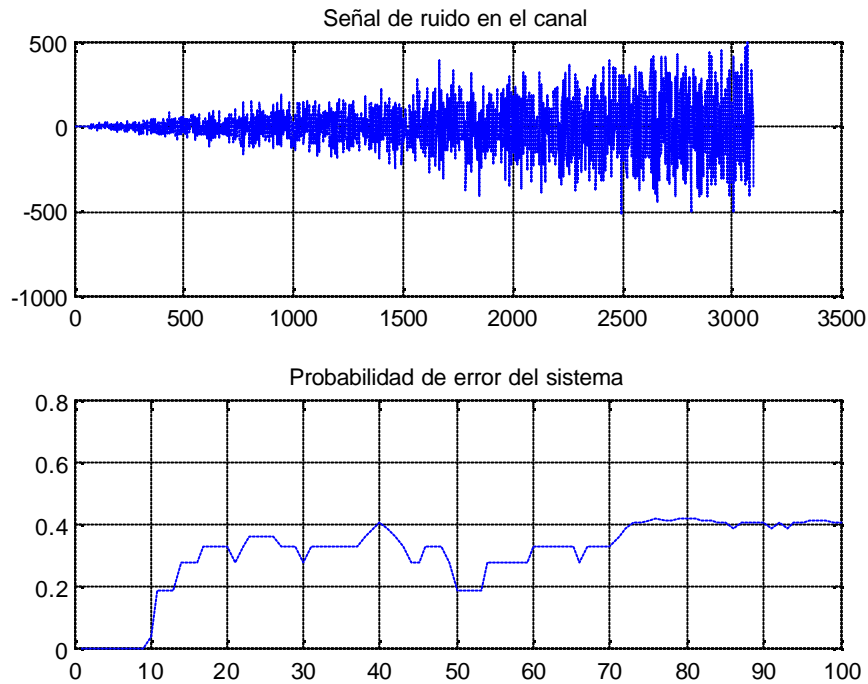


Figura. 6.18. Secuencia de Prueba de Sistema de Comunicación CDMA-SS

5. En primer lugar se simula el funcionamiento del sistema para un usuario único, con un nivel de ruido creciente en el tiempo. Para ello se realizan algunas modificaciones al paso anterior como:
  - a. Se genera una secuencia aleatoria bipolar de datos con 100 elementos.
  - b. Se obtiene la secuencia de datos que ingresa al canal de manera análoga a lo revisado en el paso previo (4).
  - c. Se genera un vector de ruido de igual longitud que la secuencia que ingresa al canal y se los suma.
  - d. Se obtiene la secuencia de salida del detector de correlación de forma análoga a lo que se hizo en el paso anterior (4).
  - e. Se obtiene la probabilidad de error del sistema. Los resultados de esto se muestran en las figuras 6.19 y 6.20.



**Figura. 6.19. Secuencias de Datos Origen y Destino para un Usuario**



**Figura. 6.20. Ruido Creciente en el Canal y Probabilidad de Error para un Usuario**

6. Ahora se simula el funcionamiento del sistema para dos usuarios, con un nivel de ruido creciente en el tiempo. Para lo cual:
  - a. Se genera una secuencia aleatoria bipolar de datos con 100 elementos para el nuevo usuario.
  - b. Se obtiene la secuencia de datos que ingresa al canal para el segundo usuario, de la misma manera como se realiza para un usuario.
  - c. El nuevo usuario transmite con una potencia diez veces superior a la del primer usuario, por lo que se suma las secuencias de ambos usuarios más el ruido generado en el paso 5, y el resultado es lo que ingresa realmente al canal
  - d. Se obtiene la secuencia de salida del detector de correlación de forma análoga a lo que se hizo para un usuario.
  - e. Se obtiene la probabilidad de error del sistema. Los resultados de esto se muestran en las figuras 6.21 y 6.22.

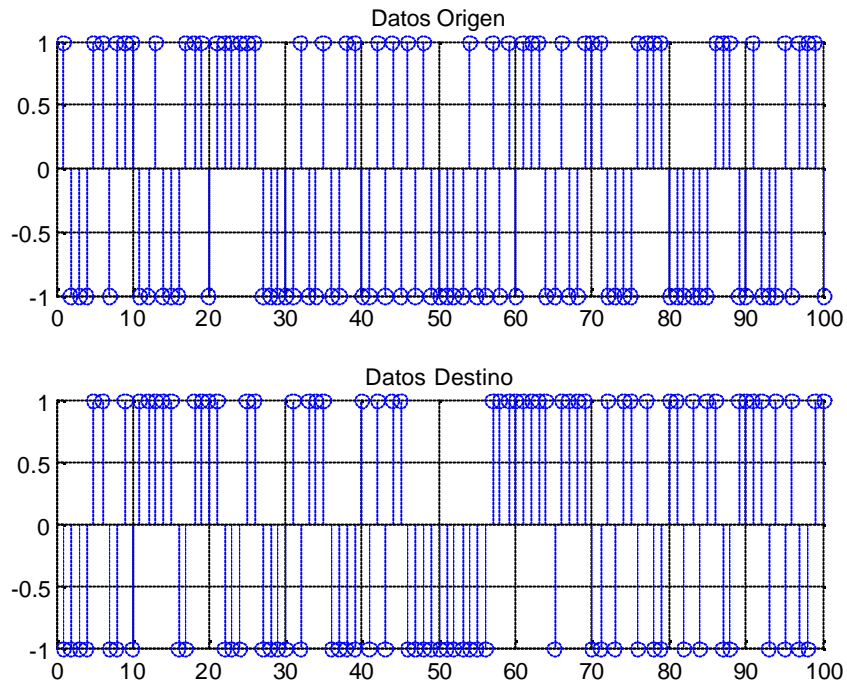


Figura. 6.21. Secuencias de Datos Origen y Destino para Dos Usuarios

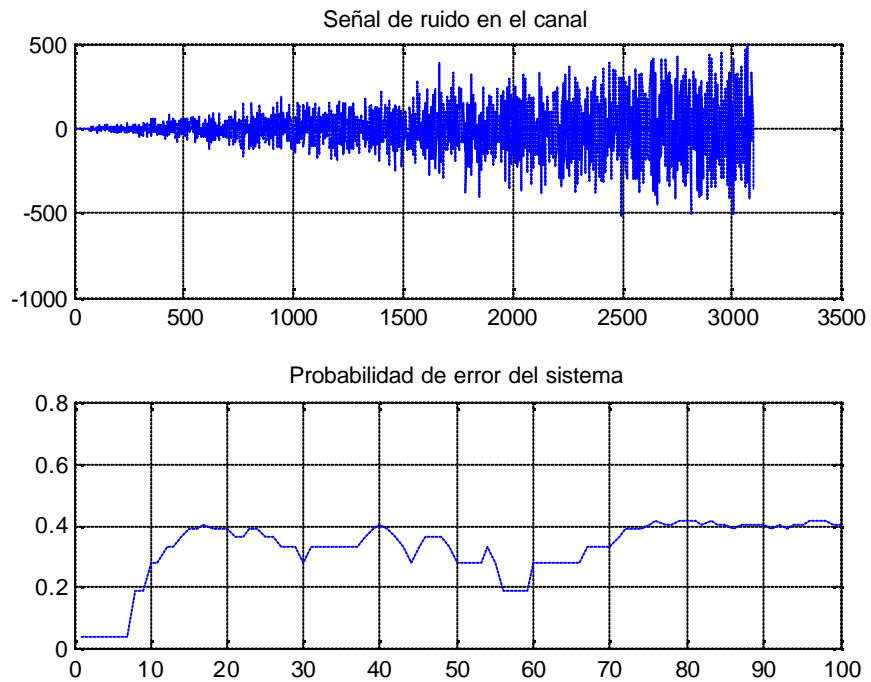
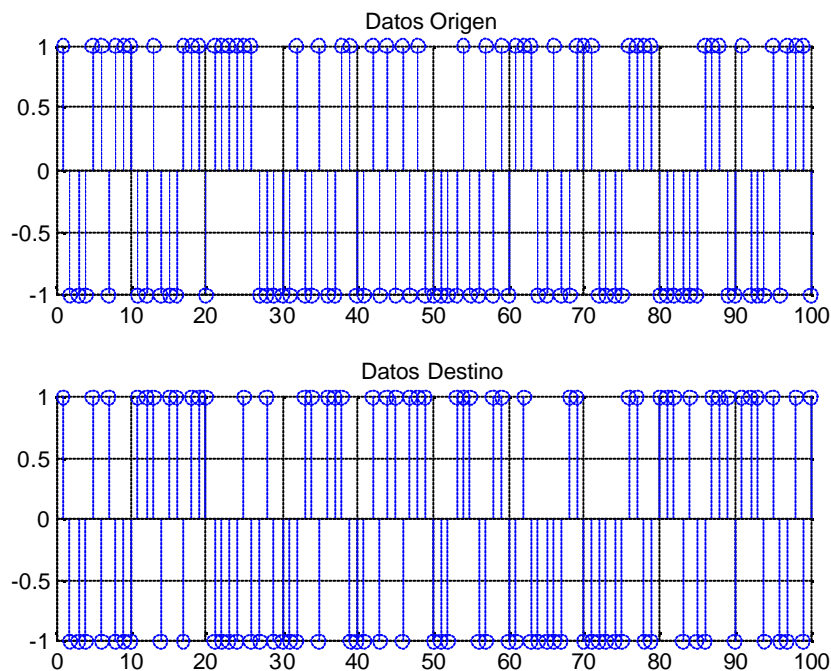


Figura. 6.22. Ruido Creciente en el Canal y Probabilidad de Error para Dos Usuarios

7. Otra simulación del funcionamiento del sistema se realiza manteniendo constante el nivel de ruido del canal y variando la amplitud de la señal del segundo usuario de forma creciente en el tiempo. Para lo cual:
  - a. Se modifica la señal del segundo usuario de forma que aumente su amplitud a lo largo del tiempo de simulación.
  - b. La señal del segundo usuario modificada se suma con la del primer usuario para determinar que es lo que ingresa realmente al canal, y que es el parámetro que debe utilizar el receptor CDMA.
  - c. Se obtiene la secuencia de salida del detector de correlación de forma análoga a lo que se hizo para un usuario.
  - d. Se obtiene la probabilidad de error del sistema. Los resultados de esto se muestran en las figuras 6.23 y 6.24.



**Figura. 6.23. Secuencias Origen y Destino para Dos Usuarios, Ruido Constante**

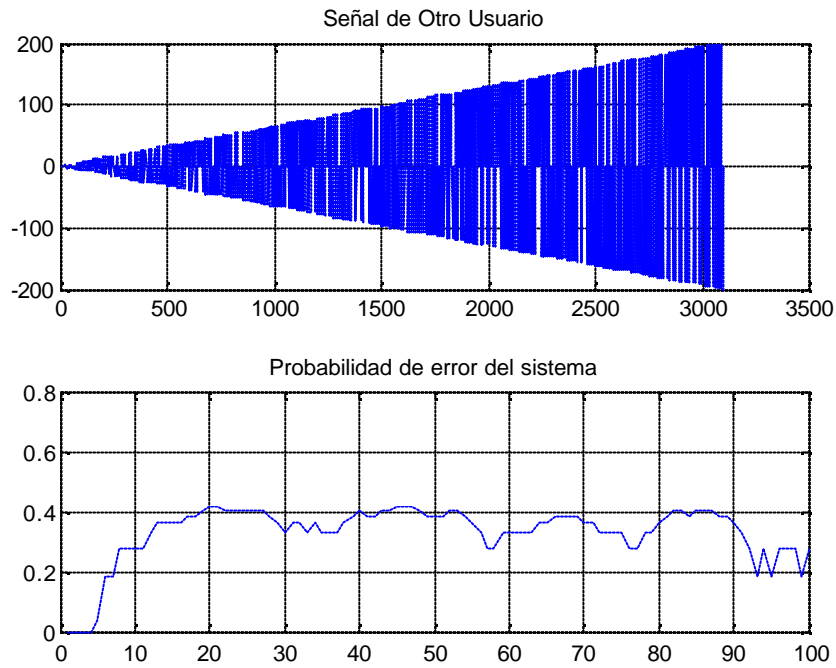


Figura. 6.24. Señal Creciente del Segundo Usuario y Probabilidad de Error

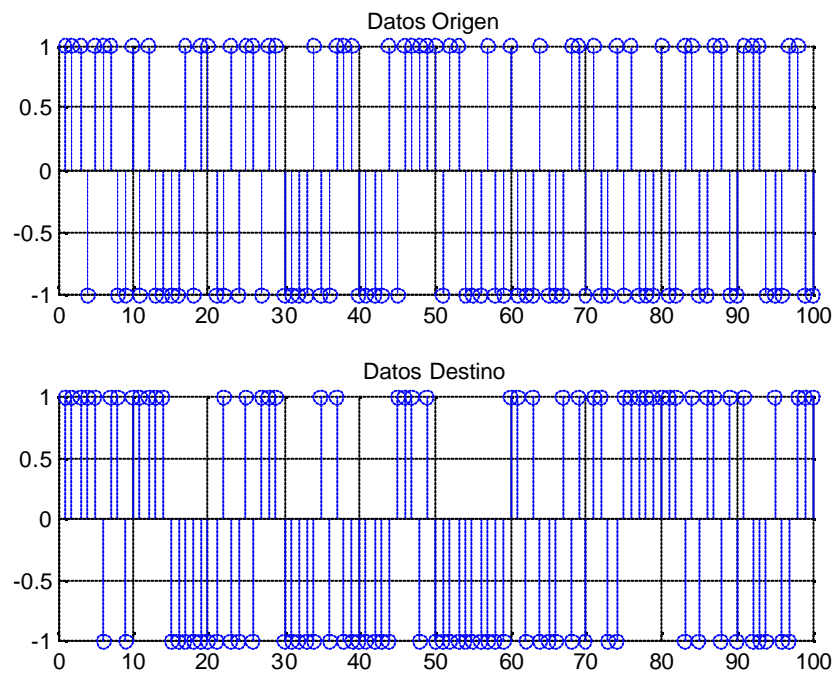
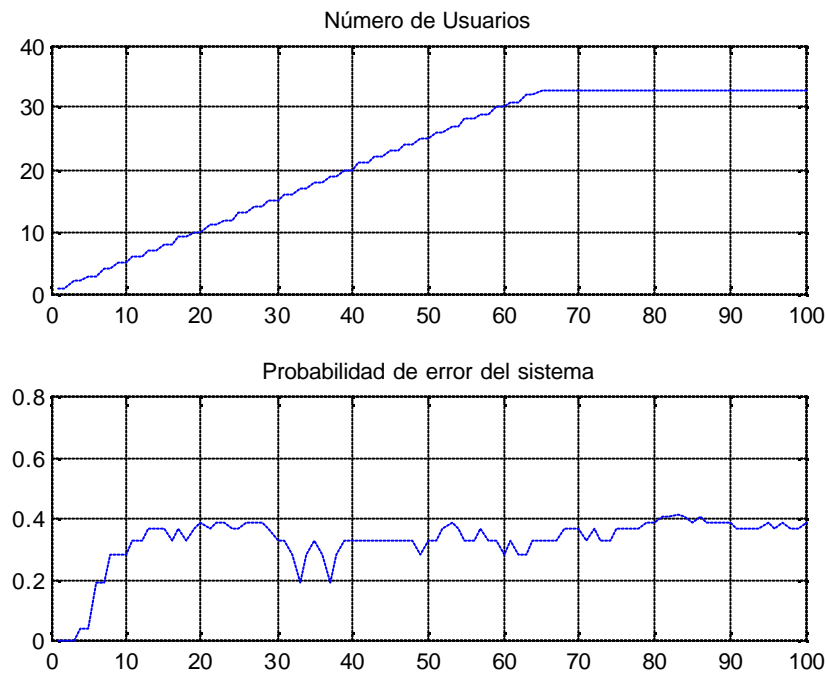


Figura. 6.25. Secuencias Origen y Destino para Múltiples Usuarios

8. La última simulación del funcionamiento del sistema se realiza aumentando el número de usuarios del sistema de comunicaciones, el nivel de ruido del canal es

constante y las señales de los usuarios son hasta 100 veces mayores de la que se considero en un principio. Para esta parte final, se realiza el mismo proceso que para un usuario, con la diferencia que no se generan ni se obtienen vectores, sino matrices de 33 vectores, o usuarios. Los resultados de esta última parte se muestran en las figuras 6.25 y 6.26.



**Figura. 6.24. Número Creciente de Usuarios y Probabilidad de Error**

### 6.3.1.3. Conclusiones

- Para un usuario puede comprobar cómo al inicio de la simulación el sistema funciona de forma adecuada, degenerándose a continuación conforme el nivel de ruido en el sistema se hace mayor.
- En las gráficas de la probabilidad de error de las dos primeras simulaciones, se puede comprobar esta evolución a peores prestaciones del sistema. Con todo, es importante tener presente la gráfica de la señal de ruido que se añade al canal, cuyos valores llegan a ser elevadísimos en comparación con la propia señal de información, y aun así, es posible en algunos casos la detección. Esto muestra una

de las cualidades principales de las técnicas de espectro ensanchado, que es su robustez al hacer uso de un ancho de banda tan elevado.

- Se observa un aumento en la probabilidad de error del sistema, debida a la presencia de un usuario adicional transmitiendo en el mismo ancho de banda. Esta diferencia, sin embargo, no es todo lo grande que podría ser, considerando el elevado nivel de ruido introducido en el canal.
- En el último caso de nuevo se observa una probabilidad de error dependiente de cuánto de fácil sea la detección de la señal deseada, y en el caso de tener un número elevado de usuarios, la interferencia multiacceso hará que esta detección sea una tarea más difícil.
- Es importante destacar, por otro lado, que en el caso de tener a 33 usuarios transmitiendo al mismo tiempo, la probabilidad de error es totalmente independiente del número de usuarios en cuestión.
- Se hace patente de esta forma la presencia necesaria de un control automático de ganancia en los equipos transmisores que vayan a utilizarse en sistemas CDMA, pues si bien esta modulación funciona a la perfección para el número máximo de usuarios teórico, si estos usuarios varían sus niveles de transmisión pueden llegar a perjudicarse mutuamente.



## 6.4. REFERENCIAS

### 6.4.1. LIBROS

- SKLAR, Bernard, **Digital Communications**, Segunda Edición, Prentice Hall, Estados Unidos 2001, páginas 719-745.
- COUCH, Leon, **Sistemas de Comunicación Digitales y Analógicos**, Quinta Edición, Prentice Hall, México 1998, páginas 357-365.
- ESPE, **Comunicación Digital**, apuntes de la asignatura, semestre septiembre 2004 – marzo 2005.

### 6.4.2. INTERNET

- <http://www.xs4all.nl/~bslash/muren/spectrum.htm>, Espectro Disperso.
- <http://www.kmj.com/proxim/pxhist.html>, Spread Spectrum Technology.
- <http://cas.et.tudelft.nl/~glas/ssc/techn/techniques.html>, Principios de Spread Spectrum.
- <http://electronics.howstuffworks.com/question326.htm>, Digital Spread Spectrum Usado en Teléfonos Inalámbricos.
- [http://www.maxim-ic.com/appnotes.cfm/appnote\\_number/1890](http://www.maxim-ic.com/appnotes.cfm/appnote_number/1890), Spread Spectrum Secuencia Directa.
- <http://www.answers.com/topic/spread-spectrum>, Información de Spread Spectrum.
- <http://www.conatel.gov.ec/espanol/baselegal/reglmtoespectroensanchado.htm>, Espectro Ensanchado en Ecuador, Normativa.

## CAPÍTULO VII

### ACCESO MÚLTIPLE

#### 7.1. INTRODUCCIÓN.

El acceso múltiple es utilizado para permitir a muchos usuarios compartir en forma simultánea una cantidad finita del espectro de radio. Al compartir el espectro se consigue una alta capacidad, ubicando al mismo tiempo una gran cantidad de usuarios en el ancho de banda disponible.

Existen varias técnicas para llevar a cabo este fin, entre ellas:

- TDMA.- Acceso Múltiple por División de Tiempo.
- FDMA.- Acceso Múltiple por División de Frecuencia.
- CDMA.- Acceso Múltiple por División de Código.
- SDMA.- Acceso Múltiple por División de Espacio.
- PDMA.- Acceso Múltiple por División de Polarización.

Las tres mayores técnicas de acceso utilizados para compartir el ancho de banda disponible en un sistema de comunicación inalámbricas son, FDMA, TDMA y CDMA). Estas técnicas pueden ser divididas en sistemas de banda estrecha y banda ancha, dependiendo de como el ancho de banda disponible es asignado para el usuario.

En los sistemas de comunicaciones inalámbricas, a menudo es deseable permitir al usuario enviar información a la Estación Base mientras se recibe información desde la misma. Por ejemplo, en los sistemas actuales es posible hablar y escuchar en forma

simultánea, y este efecto es llamado duplexación, para lo cual existen dos maneras de lograrlo:

- **FDD: Frequency Division Duplexing:** La cual proporciona dos bandas diferentes de frecuencias para cada usuario. La banda directa provee el tráfico desde la estación base al móvil y la banda reversa desde el móvil a la base en FDD. Cualquier canal dúplex consiste en la actualidad de dos canales simples, y un dispositivo llamado duplexor usado dentro de cada unidad de usuario y en la estación base para permitir la transmisión y recepción en forma simultánea sobre el par de canal dúplex. La separación de frecuencia entre el canal directo y reverso es constante a través del sistema sin importar que canal en particular sea utilizado.
- **TDD: Time Division Duplexing:** Utiliza el tiempo en vez de frecuencia para proporcionar el enlace directo y reverso. Si el tiempo de separación entre la ranura de tiempo directa y reversa es pequeño, entonces la transmisión y la recepción de datos aparecen simultáneos al usuario. TDD permite la comunicación sobre un canal simple (al contrario de los dos canales simples que se requieren en FDD) y simplifica el equipo del usuario debido que no se requiere duplexor.

Todas estas técnicas se han utilizado ampliamente durante las últimas décadas en el desarrollo de la comunicación móvil celular, las cuales se han difundido e incrementado sus prestaciones de forma vertiginosa a nivel mundial, permitiendo a los usuarios acceder a mayores y mejores beneficios manejando diferentes tipos de aplicaciones sobre un mismo terminal y sobre una misma red, o permitiendo la adecuada interconexión entre las diferentes tecnologías y sistemas como CDMA, GSM, GPS y UMTS.

A continuación se describen las técnicas de acceso nombradas de manera que puedan ser entendidas de forma clara y proporcionen una idea clara de cómo se hace para aprovechar el espectro radioeléctrico de mejor manera, considerando que es un recurso limitado y regulado en todo el mundo.

## 7.2. MARCO TEÓRICO

### 7.2.1. SISTEMAS DE ACCESO MÚLTIPLE POR BANDA

#### 7.2.1.1. Sistemas de Banda Estrecha

El término banda estrecha es usado para relacionar el ancho de banda de un canal simple al ancho de banda coherente esperado del canal. En un sistema de acceso múltiple de banda estrecha, el espectro de radio disponible es dividido en un gran número de canales de bandas estrechas. Los canales son normalmente utilizados usando FDD. En FDMA banda estrecha un usuario esta asignado a un canal particular el cual no esta compartido por otros usuarios en la cercanía, y si FDD es utilizado (esto es, cada canal tiene un enlace directo y un enlace reverso), entonces el sistema es llamado FDMA/FDD.

TDMA banda estrecha, por otro lado, permite a los usuarios compartir el mismo canal pero colocando una única ranura de tiempo (**time slot**) a cada usuario en una forma cíclica sobre el canal, lo que separa un número pequeño de usuarios en el tiempo sobre un canal simple. Para TDMA banda estrecha generalmente hay un gran número de canales asignados utilizando FDD o TDD y cada canal es compartido utilizando TDMA. Tales sistemas son llamados TDMA/FDD o TDMA/TDD

#### 7.2.1.2. Sistemas de Banda Ancha

En sistemas de banda ancha, el ancho de banda de un canal simple es mucho más grande que el ancho de banda coherente del canal. Por lo tanto, el desvanecimiento por multi trayectos no afecta mucho a la señal recibida dentro del canal de banda ancha, y el desvanecimiento selectivo de frecuencia ocurre en solamente una pequeña parte del ancho de banda de la señal. En sistemas de acceso múltiple de banda ancha, los usuarios están permitidos a transmitir en una gran parte del espectro. Un gran número de transmisores son también permitidos a transmitir sobre el mismo canal. TDMA asigna ranuras de tiempo a muchos transmisores sobre el mismo canal y permite que solamente uno de ellos acceda al

canal en un instante de tiempo, mientras que el CDMA permite que todos los transmisores accedan al canal al mismo tiempo. TDMA y CDMA pueden usar TDD o FDD como técnica de multiplexación.

### 7.2.2. DIVISIÓN DE FRECUENCIA – FDMA

Los sistemas celulares basados en FDMA formaron la base de los primeros sistemas celulares en el mundo. FDMA fue implementada en la banda de 800 MHz utilizando un ancho de banda de 30 kHz por canal.

FDMA subdivide el ancho de banda en frecuencias, cada frecuencia sólo puede ser usada por un usuario durante una llamada. Debido a la limitación en ancho de banda, esta técnica de acceso es muy ineficiente ya que se saturan los canales al aumentar el número de usuarios alrededor de una celda. Esta técnica de acceso múltiple predominó en los sistemas celulares analógicos de la primera generación.

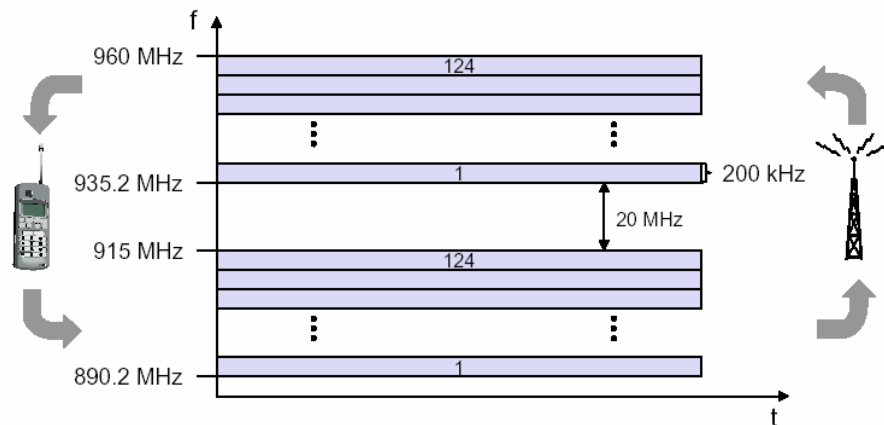


Figura. 7.1. Acceso Múltiple por División de Frecuencia

La tecnología FDMA es mayormente utilizada para la transmisión analógica. Esta tecnología no es recomendada para transmisiones digitales, aun cuando es capaz de llevar información digital.

Es estándar de telefonía celular analógica de 1G, basado en AMPS y TACS y una vez asignado un canal a un único usuario, no pueden acceder otros mientras la conversación este activa o hasta que la llamada original no haya sido pasada a otro canal por el sistema.

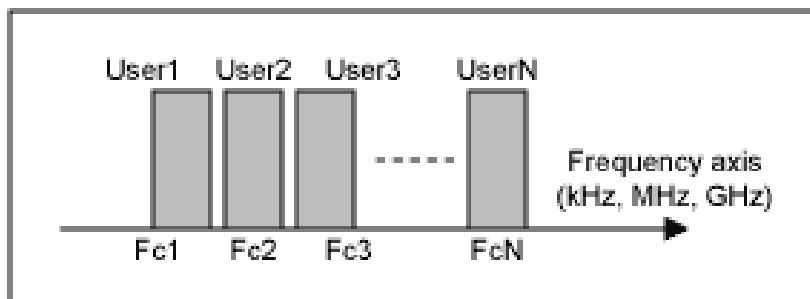


Figura. 7.2. Distribución de Usuarios en FDMA

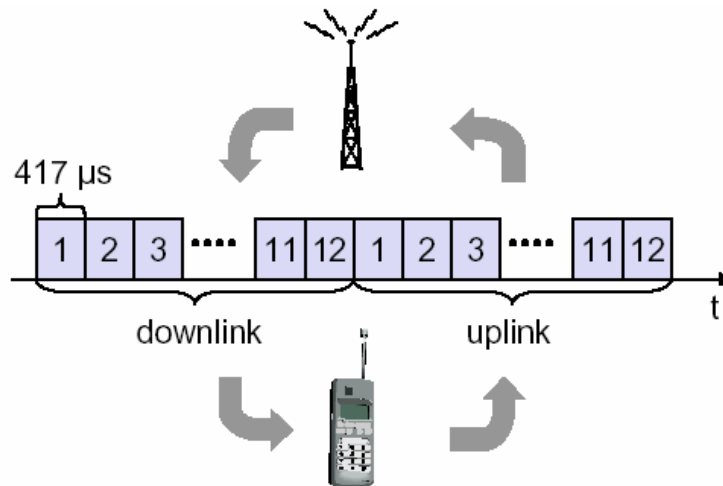
Cuando un teléfono de la célula de FDMA establece una llamada, reserva el canal de frecuencia para la duración entera de la llamada. Los datos de la voz se modulan en esta banda de frecuencia de los canales (que usa la modulación de la frecuencia) y se envían.

En el receptor, se recupera la información usando un filtro pasabanda. El teléfono utiliza un canal común del control numérico para adquirir los canales. Dado la naturaleza de la señal, los teléfonos analógicos de la célula deben utilizar una potencia más alta (entre 1 y 3 vatios) para conseguir una calidad aceptable de la llamada.

### 7.2.3. DIVISIÓN DE TIEMPO – TDMA

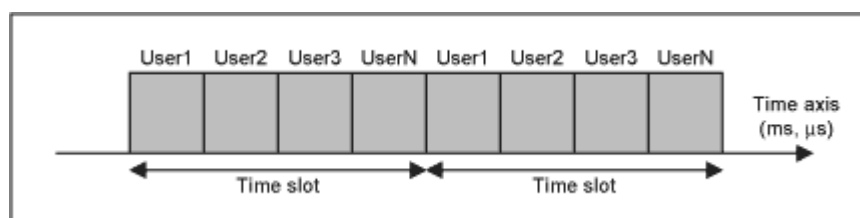
Después de la introducción de FDMA, operadores celulares y fabricantes de equipo inalámbrico reconocieron las limitaciones de esta técnica de acceso analógica. Años más adelante aparecen los primeros sistemas celulares digitales basados en TDMA. Con el fin de continuar la compatibilidad con la asignación de espectro del sistema anterior ocupado por la tecnología AMPS, se desarrolla en Norteamérica a finales de los 80s un sistema conocido como DAMPS (Digital AMPS) también con 30 kHz de ancho de banda por canal. En Europa se desarrolla también un sistema celular digital basado en TDMA conocido como GSM (Groupe Special Mobile) con canales de 200 kHz. Los primeros

sistemas bajo ASM fueron instalados en 1991, mientras el primer sistema instalado en Norteamérica fue instalado en Canadá en 1992. Se basa TDMA en el estándar IS-136, especificado en 1994 como una evolución del IS-54 de DAMPS.



**Figura. 7.3. Acceso Múltiple por División de Tiempo**

Los sistemas celulares bajo TDMA utilizan el espectro de manera similar a los sistemas FDMA, con cada radio base ocupando una frecuencia distinta para transmitir y recibir. Sin embargo, cada una de estas dos bandas es dividida en tiempo (conocidas como ranuras de tiempo) para cada usuario. Por ejemplo, TDMA de tres ranuras divide la transmisión en tres periodos de tiempo fijos (ranuras), cada una con igual duración, con una asignación particular de ranuras para transmisión para uno de tres posibles usuarios. Este tipo de metodología requiere una sincronización precisa entre la terminal móvil y la radio base.



**Figura. 7.4. Distribución de Usuarios en TDMA**

La tecnología TDMA comprime las conversaciones (digitales), y las envía cada una utilizando la señal de radio por un tercio de tiempo solamente, divide el canal análogo

original de 30 kHz en tres canales digitales en tiempo. A más de ello cuenta con un canal de control digital DCCH, que permite incrementar la eficiencia y disminuir costos de implementación, además de proveer la plataforma para nuevas características avanzadas inalámbricas. Debido a esto, la tecnología TDMA tiene tres veces la capacidad de un sistema analógico que utilice el mismo número de canales. Cada usuario habla y escucha al otro de acuerdo a ranuras de tiempo definidas, y se pueden establecer varios canales de comunicación para una única portadora.

TDMA es común en los sistemas de telefonía fija. Las últimas tecnologías en los sistemas de radio son la codificación de la voz y la compresión de datos, que eliminan redundancia y periodos de silencio y decrementan el tiempo necesario en representar un periodo de voz. Los usuarios acceden a un canal de acuerdo con un esquema temporal. Aunque no hay ningún requerimiento técnico para ello, los sistemas celulares, que emplean técnicas TDMA, siempre usan TDMA sobre una estructura FDMA. Un sistema puro TDMA tendría sólo una frecuencia de operación, y no sería un sistema útil. TDMA es un concepto bastante antiguo en los sistemas de radio.

En los sistemas modernos celulares y digitales, TDMA implica el uso de técnicas de compresión de voz digitales, que permite a múltiples usuarios compartir un canal común utilizando un orden temporal. La codificación de voz moderna, reduce mucho el tiempo que se lleva en transmitir mensajes de voz, eliminando la mayoría de la redundancia y periodos de silencio en las comunicaciones de voz. Otros usuarios pueden compartir el mismo canal durante los periodos en que éste no se utiliza. Los usuarios comparten un canal físico en un sistema TDMA, donde están asignadas unas ranuras de tiempo. A todos los usuarios que comparten la misma frecuencia se les asigna una ranura de tiempo, que se repite dentro de un grupo de ranuras que se llama trama. Una ranura GSM es de 577  $\mu$ s, y cada usuario tiene uso del canal cada 4.615 ms ( $577 \mu\text{s} * 8 = 4.615 \text{ ms}$ ), ya que en GSM tenemos ocho ranuras de tiempo.

La tecnología de radio de TDMA es la base de los principales estándares celulares digitales del mundo, GSM (y DCS1800), IS-136 y IS-54.



#### 7.2.4. DIVISIÓN DE CÓDIGO – CDMA

A mediados de los 80s algunos investigadores vieron el potencial de una tecnología conocida como espectro disperso (spread spectrum) la cual era utilizada para aplicaciones militares pero que también podría ser usada para telefonía celular. Esta tecnología de espectro disperso involucra la transformación de la información de banda angosta a una señal de banda amplia para transmisión, la cual puede ser vista como una manera de aumentar las capacidades de los sistemas TDMA que limitan el número de usuarios al número de ranuras de tiempo.

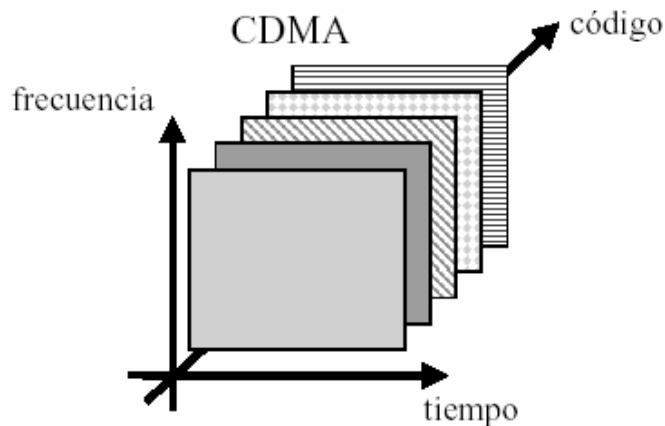


Figura. 7.5. Esquema CDMA en el Tiempo

Al asignar diferentes códigos únicos a los usuarios, un sistema de acceso múltiple es posible. A este método de acceso múltiple se le conoce como CDMA. Las limitaciones de reuso de frecuencia vistas en FDMA y TDMA ya no son tan críticas en CDMA, ya que múltiples terminales móviles y radio bases pueden ocupar las mismas frecuencias a la vez. Es obvio entonces que la capacidad en usuarios en CDMA se incrementa bastante con respecto a las otras dos técnicas de acceso múltiple.

En los sistemas CDMA todos los usuarios transmiten en el mismo ancho de banda simultáneamente, usa códigos matemáticos para transmitir y distinguir entre conversaciones inalámbricas múltiples. Los códigos usados para el esparcimiento tienen valores pequeños de correlación y son únicos para cada usuario. Esta es la razón por la que

el receptor de un determinado transmisor, es capaz de seleccionar la señal deseada.

Para telefonía celular, CDMA es una técnica de acceso múltiple especificada por la TIA (Telecommunications Industry Association) como IS-95. En marzo de 1992, la TIA estableció el subcomité TR 45.5 con la finalidad de desarrollar un estándar de telefonía celular digital con espectro extendido. En julio de 1993, la TIA dio su aprobación al estándar CDMA IS-95.

Los sistemas IS-95 dividen el espectro en portadoras de 1.25 MHz. Uno de los aspectos únicos de CDMA es que a pesar de que existe un número fijo de llamadas telefónicas que pueden ser manipuladas por un proveedor de servicios de telefonía (carrier), éste no es un número fijo. La capacidad del sistema va a depender de muchos factores.

Cada dispositivo que utiliza CDMA está programado con un pseudocódigo, el cual es usado para extender una señal de baja potencia sobre un espectro de frecuencia amplio. La estación base utiliza el mismo código en forma invertida (todos los ceros son unos y los unos ceros) para des-extender y reconstruir la señal original. Todos los otros códigos permanecen extendidos, indistinguibles del ruido de fondo.

Hoy en día existen muchas variantes, pero el CDMA original es conocido como cdmaOne bajo una marca registrada de Qualcomm. A CDMA 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.

#### 7.2.4.1. Ventajas y Beneficios

A continuación se listan algunas ventajas y beneficios de la tecnología de CDMA, que la ponen muy por encima de su competidor TDMA:

**Información paquetizada:** Las redes basadas en CDMA están construidas bajo protocolos basados en IP. En otro tipo de redes, el añadir equipo que soporte paquetes de datos es costoso y requerirá también equipo terminal que lo soporte. El estándar cdmaOne

ya incorpora en sus terminales los protocolos TCP/IP (Transfer Control Protocol/Internet Protocol) y PPP (point to point protocol).

**Seguridad y privacidad:** La técnica de espectro extendido es muy utilizada para aplicaciones militares donde la seguridad de las conversaciones y protección de los datos es importante. En un ambiente de negocios también son vitales los aspectos de seguridad y privacidad. Diseñado con alrededor de 4.4 trillones de códigos, CDMA virtualmente elimina la clonación de dispositivos y es muy difícil capturar y descifrar una señal.

**Control del nivel de potencia:** El control de la potencia es otro beneficio de los sistemas de CDMA. Empleando técnicas de procesamiento de señales, corrección de error, etc., CDMA supera el problema de la potencia con una serie de ciclos de retroalimentación. Con un control automático de la ganancia en las terminales y una supervisión constante del nivel de señal a ruido y tasas de error en la radio base, picos en el nivel de potencia son regulados con circuitería electrónica que ajusta la potencia a una razón de 800 veces en un segundo. Esto tiene gran repercusión en el ajuste dinámico del tamaño de las celdas. En una celda congestionada, la potencia de todas las terminales se elevaría creando interferencia mutua. En el margen, las transmisiones a alta potencia inundarían a las celdas vecinas donde estas podrían ser tomadas por la radio base adyacente. En una celda con poca densidad, la potencia es tan baja que la celda se reduce efectivamente, transmitiendo sin interferencia hacia las celdas vecinas mejorando el desempeño de las mismas. Este tipo de ajustamiento dinámico en el tamaño de las celdas es imposible en TDMA, donde las celdas adyacentes utilizan diferentes frecuencias. Se ha comprobado en diversos estudios, que CDMA es ciento de veces más eficiente en potencia que TDMA.

**Bajo consumo de potencia y baterías más duraderas en las terminales:** Debido al sistema de retroalimentación de CDMA que mantiene la potencia al más bajo nivel permisible, las terminales consumen menos potencia y son más pequeñas y las baterías de CDMA duran mas tiempo que las de TDMA.

**Amplia cobertura con pocas celdas:** La señal de espectro extendido de CDMA provee gran cobertura en la industria inalámbrica, permitiendo a los carriers la instalación de menos celdas para cubrir un área más extensa. Pocas celdas significan para los carriers

mucho ahorro en infraestructura de radio bases. Dependiendo de la carga del sistema y de la interferencia, la reducción de celdas es 50% menor en CDMA que en sistemas como ASM basado en TDMA. Es preciso notar que la reducción de celdas sólo es válida para operadores que empezaron desde un principio con CDMA. Operadores que utilizan sistemas analógicos o basados en otras tecnologías deberán redistribuir las celdas CDMA con las celdas ya existentes.

**Pocas llamadas caídas:** La transferencia de celdas (handoff) de CDMA, método para transferir llamadas entre celdas reduce inteligentemente el riesgo de interrumpirlas durante una transferencia. El proceso conocido como transferencia suave o transparente (soft handoff) entre celdas conduce a pocas llamadas caídas ya que 2 o 3 celdas están monitoreando la llamada todo el tiempo. La transferencia entre celdas es transparente al usuario debido a que como todos los usuarios están utilizando el mismo espectro, es más fácil moverse de una celda a otra sin que el subscriptor se de cuenta.

**Resistencia a la interferencia, ruido del ambiente y multitrayectorias:** Las multitrayectorias en CDMA en vez de ocasionar problemas con la señal, la fortalece más. Esto conduce a una casi eliminación de la interferencia y desvanecimiento. Ambos, el ruido eléctrico de fondo y ruido acústico de fondo son filtrados al usar ancho de banda angosta que corresponde a la frecuencia de la voz humana. Esto mantiene al ruido de fondo fuera de las conversaciones. En TDMA por el contrario, por ser basada en el tiempo, las multitrayectorias son un problema. Señales que vienen de distintas trayectorias a desfasadas en el tiempo ocasionan que están interfieran a las ranuras adyacentes haciendo que se interfieran las llamadas y se caigan.

**Implantación más rápida:** Los sistemas CDMA pueden ser implantados y expandidos más rápidamente debido a que requieren de menos celdas. Pocas celdas significa para los operadores menor gasto de inversión y operación.

**Ancho de banda en demanda:** El canal de 1.25 MHz de CDMA provee un recurso común a todas las terminales en un sistema de acuerdo a sus propias necesidades, como podría ser voz, fax, datos u otras aplicaciones. En un tiempo dado, la porción de este ancho de banda que no sea usada por una terminal, estará disponible para otro usuario. Debido a que CDMA utiliza una porción grande de espectro repartida entre varios usuarios, provee

flexibilidad en el ancho de banda para permitir servicios en demanda. Bajo TDMA donde los canales son fijos y pequeños, esto no es posible. En forma general, está comprobado que CDMA es de 3 a 6 veces más eficiente en ancho de banda que TDMA.

**Compatibilidad hacia adelante y hacia atrás:** Las terminales que funcionan bajo CDMA son compatibles con su versión antecesora. Una terminal de tercera generación, cdma2000 por ejemplo, es compatible con cdmaOne de segunda generación. La compatibilidad hacia adelante se realiza sustituyendo un simple chip en la terminal. Una terminal con cdmaOne puede actualizarse a cdma2000, sustituyendo el únicamente el chip principal. Esto evita la compra de una nueva terminal para acceder a los nuevos servicios.

**Calidad de voz mejorada:** Sofisticados esquemas de control y corrección de error hace que las tramas de información sean interpretadas correctamente. Por otro lado, sofisticados codificadores de voz (vocoders) la codifican a altas velocidades y reducen el ruido de fondo. La transferencia suave entre celdas es otro factor que eleva la calidad de la voz en una conversación. El control preciso de los niveles de potencia asegura que todas las terminales cercanas al nivel óptimo provean la calidad de voz más alta posible.

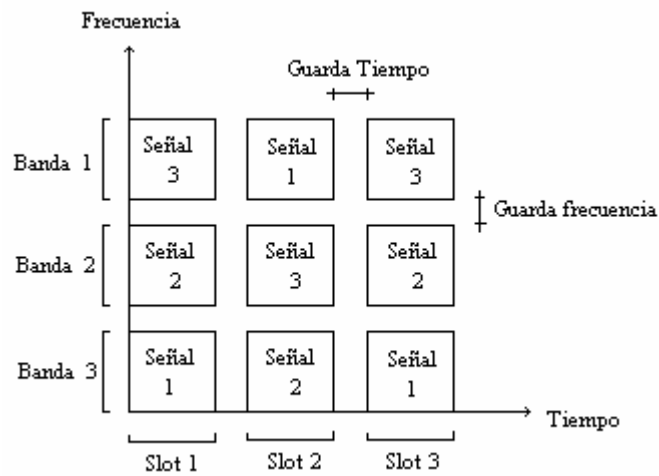
La tecnología CDMA es muy diferente a la tecnología TDMA. La CDMA, después de digitalizar la información, la transmite a través de todo el ancho de banda disponible. Varias llamadas son sobrepuestas en el canal, y cada una tiene un código de secuencia único. Usando a la tecnología CDMA, es posible comprimir entre 8 y 10 llamadas digitales para que estas ocupen el mismo espacio que ocuparía una llamada en el sistema analógico.

En teoría, las tecnologías TDMA y CDMA deben de ser transparentes entre sí (no deben interferirse o degradar la calidad), sin embargo en la práctica se presentan algunos problemas menores, como diferencias en el volumen y calidad, entre ambas tecnologías

#### 7.2.4.2. Funcionamiento

Los usuarios comparten la misma banda de frecuencia y cada señal es identificada por un código especial, que actúa como una clave reconocida por el transmisor y el receptor. La señal recibida es la suma de todas las señales "combinadas", y cada receptor debe

clasificar e identificar las señales que le corresponden de las demás señales. Para hacer esto utiliza un código que corresponde con el código transmitido.



**Figura. 7.6. Acceso Múltiple por División de Código**

La primera operación implica encontrar del código correcto, y así sincronizar el código local con el código entrante. Una vez ha ocurrido la sincronización, la correlación del código local y del código entrante permite a la información apropiada ser extraída y las otras señales ser rechazadas.

También permite que dos señales idénticas que vienen de diversas fuentes, sean demoduladas y combinadas, de modo tal que se mejore la calidad de la conexión, por lo que es también una ventaja el uso simultáneo de varios satélites (diversidad).

A nivel mundial existen muchas propuestas para proveer servicios de banda ancha basadas en la tecnología CDMA. Por ejemplo Qualcomm (EUA) está promoviendo cdma2000, Ericsson (Europa) y NTT DoCoMo (Japón) con WCDMA (Wideband CDMA). En Corea, por su parte están promoviendo WCDMA TTA I y TTA II para servicios de 3G.

### 7.2.5. DIVISIÓN DE ESPACIO – SDMA

El SDMA es un modo de comunicaciones satelital que optimiza el espectro de radio y minimiza los costos aprovechando las características direccionales de las antenas satelitales. Las antenas del satélite transmiten la señal a diversas zonas de la superficie terrestre, permitiendo duplicar el uso de frecuencias para múltiples zonas. Dado que las antenas son altamente direccionales y la señal es estrecha, se corre menos riesgo de sufrir interferencias de otras fuentes que utilicen las mismas frecuencias. Requiere una cuidadosa selección de las zonas a transmitir, además de un alineamiento preciso de las antenas. Un error de alineación podría producir la pérdida de canales, interferencia entre canales, y confusión entre las zonas cubiertas

Es también una de los métodos más promisorios en la solución del problema de la capacidad en los sistemas de comunicación inalámbrica. Para crear algoritmos SDMA más eficientes es vital determinar y entender el límite teórico de la mejora en el desempeño. El más grande reto es extender la ecuación de capacidad del canal de Shannon para cubrir los canales inalámbricos que usan procesamiento de señal espacial.

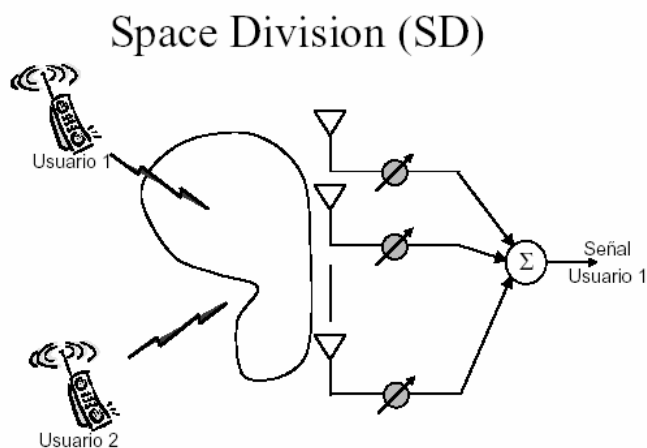
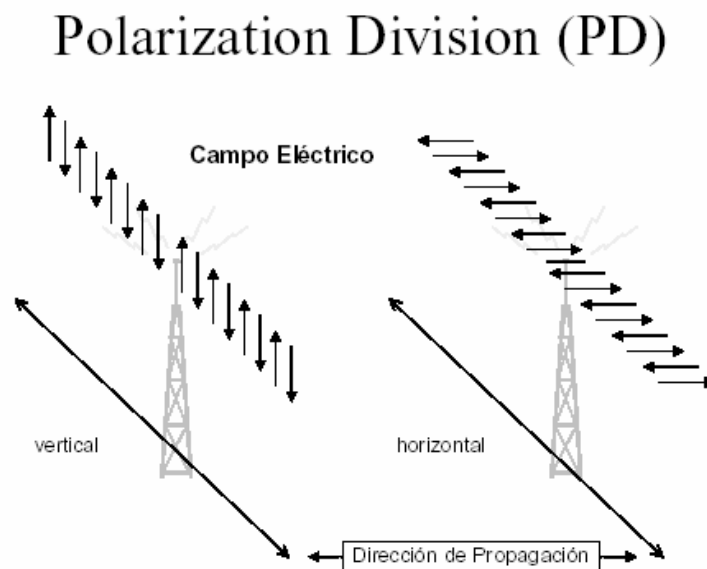


Figura. 7.7. Acceso Múltiple por División de Espacio

### 7.2.6. DIVISIÓN DE POLARIDAD – PDMA

Los arreglos multi polarizados combinados con técnicas adaptativas de formación de lóbulos prometen mejor calidad en sistemas de comunicaciones inalámbricas, ofreciendo reducción de interferencias y la mitigación de mal acoplamiento de polarización causado por la orientación aleatoria de equipos portátiles. Un incremento en la capacidad puede ser posible en algunos sistemas reusando las frecuencias con alguna combinación de SDMA y PDMA.



**Figura. 7.8. Acceso Múltiple por División de Polaridad**

### 7.2.7. COMPARACIÓN DE LOS SISTEMAS DE ACCESO MÚLTIPLE



<b>Consideración</b>	<b>SDMA</b>	<b>TDMA</b>	<b>FDMA</b>	<b>CDMA</b>
<b>Idea</b>	Segmenta el espacio en células / sectores	Segmenta el tiempo en ranuras disjuntas, requiere patrones ajustados	Segmenta la banda de frecuencia en bandas disjuntas	Usa espectro ensanchado con códigos ortogonales.
<b>Terminales</b>	Un solo terminal puede estar activo en una célula / sector	Todos los terminales están activos por cortos periodos de tiempo en la misma frecuencia	Cada Terminal tiene su propia frecuencia, ininterrumpido.	Todos los terminales pueden estar activos en un mismo lugar en un mismo momento, ininterrumpido.
<b>Separación de Señales</b>	Estructura de células, antenas directivas	Sincronización en el dominio del tiempo.	Filtrado en el dominio de la frecuencia	Códigos y receptores especiales
<b>Ventajas</b>	Muy simple, incrementa la capacidad por km <sup>2</sup>	Establecido, totalmente digital, flexible	Simple, establecido, robusto.	Flexible, necesita menor planificación de las frecuencias, posee un hand over suave
<b>Desventajas</b>	Inflexible, antenas generalmente fijas	Necesita espacio de guarda (propagación multi trayecto) dificultad de sincronización.	Inflexible, las frecuencias son un recurso escaso.	Receptores complejos, necesita control de potencia más complicados para los remitentes o transmisores.
<b>Comentario</b>	Útil solo en combinación con TDMA, FDMA o CDMA	Estándar en redes fijas, junto con FDMA/SDMA se usa en diversas redes de telefonía móvil	Combinado típicamente con TDMA (patrones de salto de frecuencia) y SDMA (reuso de frecuencias)	Aun enfrenta algunos problemas como la alta complejidad; se integrará con TDMA/FDMA

**Tabla. 7.1. Comparación de Sistemas de Acceso Múltiple**

### 7.3. PRÁCTICAS CON MATLAB

#### 7.3.1. PROYECTO DE DISEÑO DE UN SISTEMA DE COMUNICACIÓN CELULAR

Se presenta un proyecto de diseño de un sistema de comunicación celular, basado en una serie de simulaciones en MATLAB y con el objeto de aplicar lo aprendido en el desarrollo del curso de Comunicación Digital.

El proyecto consta de tres partes:

- (i) Diseño del codificador de voz con muestreo y cuantización.
- (ii) Diseño de la modulación y forma de pulso.
- (iii) Diseño del esquema de acceso múltiple y el presupuesto del enlace.

##### 7.3.1.1. Diseño del Codificador de Voz con Muestreo y Cuantización

En esta parte se tendrá que diseñar un cuantizador no uniforme que será usado para la transmisión de voz para el sistema celular. Se deberá escoger la situación óptima para estos niveles de cuantización. Aunque existe un diseño óptimo propuesto, se tendrá la mayor parte del crédito por encontrar un buen diseño alternativo. Se tiene libertad de consultar con otras personas, aunque la decisión final del diseño será únicamente la que se tome por uno mismo.

1) Una señal de voz humana tiene un ancho de banda aproximado de 3500 Hz. Se dan muestras de señal de voz tomadas a razón de 8000 muestras/segundo. Se deberá asumir que cada muestra  $x$  es una variable aleatoria Gaussiana con una pdf dada por

$f_x(x) = \frac{1}{\sqrt{2p}} e^{-\frac{x^2}{2}}$ . Se deberá determinar un conjunto de 16 niveles de cuantización para un

cuantizador no uniforme de tal manera que su SNR sea máximo. Se cuenta con un archivo

m-file **quant1.m** que se lo llama desde MATLAB colocando el archivo quant1.m en el directorio de trabajo y tecleando:

**quant1 ([level1 level2...level16], n\_trials)**

Donde level1, level2,..., level16 son los niveles de cuantización (en orden ascendente), y donde n\_trials es el número de iteraciones que se desea correr para determinar los niveles de cuantización.

La respuesta final deberá contar con:

- (a) Un listado de los niveles de cuantización.
- (b) El promedio de la SNR para la simulación.

#### **7.3.1.1.1. Observaciones:**

- (i) Se deberá usar el conocimiento de cuantizador no uniforme. Se debe concentrar los niveles de cuantización más pesados en áreas de densidad de probabilidad más grandes.
- (ii) El tamaño de la PDF será simétrico con respecto a cero, tiene sentido que un buen cuantizador probablemente debe ser también simétrico con respecto a cero.
- (iii) Se deberá correr la simulación del programa unas cuantas veces para conseguir una percepción de para cuántas muestras se necesita simular para conseguir consistencia razonable de resultados: n\_trials=100, son muy pocos ensayos, pero n\_trials=50000, toma demasiado el ejecutar muchos experimentos.

2) Usar el método analítico para calcular el SNR para el cuantizador del problema. Calcular la distorsión promedio necesaria en el ejercicio.

3) Se desea simular un cuantizador diferencial que tenga diferencias entre las muestra de dos señales consecutivas con un nivel de cuantización de 16 niveles.

Para el trabajo final, se debe proveer:

- (a) Una lista de los niveles de cuantización.
- (b) La relación SNR de la simulación.

### 7.3.1.1.2. Primera Parte.- Cuantización No Uniforme.

Se desea obtener un conjunto de valores óptimo que indiquen los niveles de cuantización y que permitan cumplir con las condiciones requeridas. Se utilizará valores entre -2.5 y 2.5 V para este caso, en el cual se va a emplear la función de MATLAB `quant1.m`. Para el caso del cuantizador diferencial, es decir, la tercera parte del proyecto, se utilizará el archivo/función `quant2.m` y en este caso los valores entre -2 y 2 V.

*Niveles de Cuantización:*

Set # 1 = [-1. 0.4 2.1 -2.8 1 -1.03 -1 -0.7 -0.03 1.7 2.5 2.6 2.5 -2.3 2.8 -1.7];

Set # 2 = [-2 -1.8 -1.7 -1.4 -1 -0.7 -0.4 -0.2 0.1 0.3 0.6 0.9 1.1 1.4 1.7 1.9];

Set # 3 = [0.1 -2.1 2.2 2.9 3.9 -1.4 2.8 -1.7 3.6 -2 3.5 2.4 -2.5 0.4 -1.1 -0.5];

Set # 4 = [-2.3 1.9 0.3 0.7 -3.9 2.2 3.2 2.5 -1.6 3.8 3.9 2.7 -3.1 2.7 2.4 -2.5];

Set # 5 = [3.7 2.2 -0.3 -3.5 -2.97 2.2 2.5 -0.1 -2.2 -1.9 -0.2 2.4 3.1 -0.5 -1.3 -3.2];

Set # 6 = [1.6 -2.1 2.9 4 1.3 1.9 0.6 2.3 1.5 -1 -2.4 -2 3.8 3.95 -0.01 2.8];

Set # 7 = [1.4 -1.7 -0.9 3.7 0.5 -3.5 -2.5 -1.8 -0.4 3.6 -1.6 -0.02 1.9 2.6 3.4 -3.8];

Set # 8 = [-2.4 -2.1 -1.8 -1.5 -1.2 -0.9 -0.6 -0.3 0 0.31 0.6 0.9 1.2 1.5 1.8 2.1];

Estos valores fueron obtenidos de forma aleatoria en MATLAB y para los niveles de cuantización mostrados, la mayor SNR se obtuvo para el octavo caso, para lo cual se variaron las iteraciones de la siguiente manera:

<i># Iteraciones</i>	<i>SNR(dB)</i>
200	20.6895
2000	19.1209
4000	19.6426
6000	18.8999

**Tabla. 7.2. SNR para Diferentes Iteraciones  
Cuantización No Uniforme**

Como se puede observar, el valor más alto obtenido es de 20.69 dB y corresponde a 200 iteraciones. Visto de esta manera, la línea de código en MATLAB es la siguiente:

**Snr1=quant1([-2.4 -2.1 -1.8 -1.5 -1.2 -0.9 -0.6 -0.3 0 0.31 0.6 0.9 1.2 1.5 1.8 2.1],2000)**

### 7.3.1.1.3. Segunda Parte.- Analítica.

El valor de SNR requerido para este tipo de sistemas, teóricamente, se puede determinar mediante las leyes  $\mu$  o A, que se usan para sistemas de cuantización no uniforme, ya que si se usa la estimativa encontrada en Sklar para cuantizadores uniformes, se van a determinar valores mucho más altos de SNR y que no corresponden a este tipo de diseño realizado.

Entonces para lo que es cuantización uniforme se tendría una SNR dada por:

$$SNR = 3L^2 = 3 \cdot 16^2 = 768$$

$$SNR_{dB} = 10\log(SNR) = 10\log(768)$$

$$SNR_{dB} = 28.85_{dB}$$

Que como se puede apreciar es un valor muy alto y que se aleja de los resultados obtenidos por la simulación y se hace incluso lejano de alcanzar.

Por el contrario usando ley A o ley  $\mu$  se observan valores más conservadores, los cuales se determinan a partir de las siguientes ecuaciones y cálculos que se encuentran en el libro de Couch (Sistemas de Comunicación Digitales y Analógicos):

$$SNR_{dB} = 6.02n + a$$

$$a = 4.77 - 20\log[\ln(1 + m)]$$

$$a = 4.77 - 20\log[1 + \ln(A)]$$

Para el caso aquí tratado, se escogerá el estándar americano o ley A, cuyos valores resultantes son:

$$a = 4.77 - 20\log[1 + \ln(87.6)]$$

$$a = 4.77 - 20\log(5.473)$$

$$a = -10_{dB}$$

$$SNR_{dB} = 6.02 * 4 - 10$$

$$SNR_{dB} = 24.08 - 10$$

$$SNR_{dB} = 14.08_{dB}$$

Así, se obtiene un valor de SNR de acuerdo a las condiciones de diseño requeridas y que como se puede comprobar es menor a la SNR obtenida con las funciones de MATLAB proporcionadas para las simulaciones.

#### 7.3.1.1.4. Tercera Parte.- Diferencial.

Para esta parte del problema se utiliza de forma semejante a la primera, 16 niveles para la cuantización, con la diferencia que ahora se trata de cuantización diferencial. Estos niveles son más pequeños en las áreas cercanas a cero y más separados entre más se alejan del cero, esto debido a la forma como se distribuyen la pdf de las señales de speech. Considerando estos puntos, se han tomado también valores aleatorios, pero se los ha modificado en ciertas ocasiones de forma que se pueda mejorar la SNR y optimizar los resultados buscados.

#### *Niveles de cuantización*

Set # 1 = [-1. 0.4 2.1 -2.8 1 -1.03 -1 -0.7 -0.03 1.7 2.5 2.6 2.5 -2.3 2.8 -1.7];

Set # 2 = [-2 -1.8 -1.7 -1.4 -1 -0.7 -0.4 -0.2 0.1 0.3 0.6 0.9 1.1 1.4 1.7 1.9];

Set # 3 = [0.1 -2.1 2.2 2.9 3.9 -1.4 2.8 -1.7 3.6 -2 3.5 2.4 -2.5 0.4 -1.1 -0.5];

Set # 4 = [-2.3 1.9 0.3 0.7 -3.9 2.2 3.2 2.5 -1.6 3.8 3.9 2.7 -3.1 2.7 2.4 -2.5];

Set # 5 = [3.7 2.2 -0.3 -3.5 -2.97 2.2 2.5 -0.1 -2.2 -1.9 -0.2 2.4 3.1 -0.5 -1.3 -3.2];

Set # 6 = [1.6 -2.1 2.9 4 1.3 1.9 0.6 2.3 1.5 -1 -2.4 -2 3.8 3.95 -0.01 2.8];

Set # 7 = [1.4 -1.7 -0.9 3.7 0.5 -3.5 -2.5 -1.8 -0.4 3.6 -1.6 -0.02 1.9 2.6 3.4 -3.8];

Set # 8 = [-2 -1.8 -1.6 -1.3 -0.9 -0.6 -0.3 -0.1 0.1 0.3 0.6 0.9 1.3 1.5 1.8 2]

De las pruebas realizadas, para el octavo conjunto de valores se obtuvo el mayor valor de SNR, igual a 22.7 dB. La línea que se ejecutará en la ventana de comandos de MATLAB por consiguiente queda de la siguiente forma:

**snr2=quant2([-2 -1.8 -1.6 -1.3 -0.9 -0.6 -0.3 -0.1 0.1 0.3 0.6 0.9 1.3 1.5 1.8 2],2000)**

<i># Trials</i>	<i>SNR(dB)</i>
400	22.49
2000	22.7
3000	22.11
4000	22.06
6000	22.11

**Tabla. 7.3. SNR para Diferentes Iteraciones  
Cuantización Diferencial**

La SNR promedio entonces para los números de iteraciones dados y con el último conjunto de valores para los niveles de cuantización es de 22.3 dB.

De todo lo revisado se puede concluir que:

- Los mejores valores de SNR obtenidos en la cuantización no uniforme se obtienen en los cuantizadores donde se manejan pasos de cuantización más pequeños para las zonas de menor amplitud de la señal de entrada.
- La cuantización es un proceso que permite discretización de amplitudes, ya que lo realizado básicamente en este trabajo, es asignar un número de niveles al cuantizador.

- Con un cuantizador diferencial se obtiene valores de SNR superiores a los que se pueden obtener con un cuantizador no uniforme por lo que claramente se puede establecer que en términos de mejor SNR es preferible utilizar un cuantizador diferencial a un no uniforme y mucho mejor que un uniforme, según también el tipo de señal aunque, en este caso de una señal de voz.

### 7.3.1.2. Diseño de la Modulación y Forma de Pulso

En la primera parte se realizó un esquema de cuantización diferencial para un sistema celular para transmisión de una señal de voz con SNR > 25 dB, tasa de datos de 32 Kbps. Ahora se realiza el diseño de la modulación y forma de pulso para transmitir una señal de voz sobre un enlace de comunicación celular.

#### 7.3.1.2.1. Requerimientos de diseño.

Se tienen dos consideraciones que se deben satisfacer:

1. La señal transmitida debe ocupar un canal no mayor a 50 KHz. Todos los componentes espectrales fuera de esos 50 KHz de ancho de banda deben estar al menos 40 dB por debajo del valor de la densidad espectral de potencia.
2. Para poder tener una fidelidad aceptable, la señal debe ser transmitida con una BER no mayor a 0.001.

Una vez logrados estos requerimientos, el objetivo es minimizar el costo de manufactura de los equipos de comunicación.

#### 7.3.1.2.2. Condiciones de Diseño.

**Modulación:** Se ha determinado que se usará modulación PSK, pero se debe determinar entre BPSK (M=2), QPSK (M=4), o 8-PSK (M=8) para obtener una mayor eficiencia espectral. Mayores valores de M proporcionan mayor eficiencia espectral, aunque se ha comprobado que M=2 o M=4 son los mejores.



**Forma de Pulso:** Se usará la forma de pulso de coseno levantado, pero se debe determinar el factor de rolloff ( $0 < r < 1$ ). Valores pequeños de  $r$  proporcionan mejor eficiencia espectral pero es ligeramente más costoso y puede producir grandes lóbulos laterales cuando el pulso es truncado.

**Recepción Coherente / Diferencial:** Recepción coherente proporciona mejor desempeño, pero la recepción diferencial es mucho más barata de implementar.

**Filtro Casado / PasaBajos:** Un filtro casado tiene un desempeño óptimo. Como una alternativa más económica algunos receptores celulares usan un simple filtro pasabajos. En términos de desempeño nos cuesta aproximadamente 1 dB.

**Código de Corrección de Error:** Se puede usar un código de corrección de error en el sistema, similar al usado en IS-95. Esto añade complejidad (costos) al sistema y duplica el ancho de banda debido a los bits extra transmitidos. Sin embargo, esto representa también una reducción del  $E_b/N_0$  requerido en aproximadamente 1 a 5 dB, dependiendo de las condiciones de transmisión.

### 7.3.1.2.3. Costos.

Cada uno de las condiciones elegidas son determinantes. De esta forma se determina el costo final del sistema de acuerdo a los siguientes criterios:

**Modulación:** Entre más compleja es la modulación, más cara resulta implementarla. BPSK cuesta \$2, QPSK cuesta \$4 y 8-PSK cuesta \$8.

**$E_b/N_0$ :** Mayor potencia de la señal requiere baterías más grandes, lo que representa mayor costo. Asumimos que cada dB adicional de energía cuesta \$1.

**Forma de Pulso:** Implementar un factor de rolloff muy agudo puede añadir cierta complejidad. Asumimos que el costo del filtro de forma de onda está dado por  $\$(1-r)^2$ .

**Recepción Coherente/Diferencial:** Un receptor coherente cuesta \$5 más que un receptor diferencial.

**Filtro Casado/PasaBajos:** Es \$ 1.50 más costoso implementar un filtro casado que un simple filtro pasabajos.

**Código de corrección de Error:** Si se escoge usar un código de corrección de error para ahorrar energía, esto nos costará \$ 1.50.

#### 7.3.1.2.4. Herramienta de Simulación.

El archivo m (función) de MATLAB **mod\_sim.m** se usa para determinar la modulación y forma de pulso de acuerdo los resultados entregados con su ejecución. Una vez colocado en el directorio de trabajo adecuado, se utiliza MATLAB y se corre la simulación con el comando:

**mod\_sim (M, r, diff, matched\_f, code, Eb\_No\_dB, n\_trials)**

Donde:

**M** = el número de símbolos de modulación (Debe ser 2,4, o 8).  
**r** = es el factor de rolloff del filtro de coseno levantado ( $0 < r < 1$ ).  
**diff** = 1 indica que se está usando detección diferencial.  
 0 indica que se usa detección coherente.  
**matched\_f** = 1 indica que se está utilizando filtro casado.  
 0 indica que se utilice un filtro pasabajos simple.  
**code** = 1 indica que se está usando código de corrección de error.  
 0 indica que no se usa código de corrección de error.  
**Eb\_No\_dB** = el valor de Eb/No en dB (usualmente,  $5\text{dB} < Eb/No < 15\text{dB}$ ).  
**n\_trials** = número de iteraciones para la simulación ( $1000 < n\_trials < 10e6$ ).

Así se tiene por ejemplo:

**mod\_sim (4,0.9, 0, 1, 1, 6.7, 20000);**

Lo que indica que se ha designado una modulación QPSK (4) con factor de rolloff de coseno levantado 0.9, usando filtro de detección coherente (0), un filtro casado receptor (1)

y código de corrección de error (1). También se tiene para este ejemplo un  $E_b/N_0 = 6.7$  dB, y se ejecutan 20000 iteraciones de la simulación.

El programa retornará las siguientes salidas:

- Un dibujo de la forma de pulso y de la potencia espectral de potencia de la señal transmitida. La gráfica de la densidad espectral de potencia ayuda a comprobar si se verifican los requerimientos de ancho de banda.
- El valor simulado de BER para el sistema.
- Una estimativa de la fiabilidad de la simulación de la BER. En general una fiabilidad menor que 0.5 indica que se tienen que realizar más iteraciones, una fiabilidad entre 0.5 y 1 indica que la estimación es casi confiable, y una fiabilidad mayor que 1 nos indica que no se ganaría mucho más exactitud si se aumenta el número de iteraciones.
- El costo calculado de la implementación del sistema.

#### 7.3.1.2.5. Desarrollo.

De acuerdo a lo que se ha dicho hasta el momento, se procede a simular el comportamiento de estos diseños para diferentes valores de los diferentes parámetros utilizados por la función que se debe ejecutar en MATLAB.

Se realiza pues la simulación primero de dos sistemas BPSK, donde se varía el factor de rolloff, la utilización de detección coherente o diferencial, la utilización de código de corrección de errores, el  $E_b/N_0$  y el número de iteraciones. Este mismo proceso se usará tanto para QPSK como para 8-PSK.

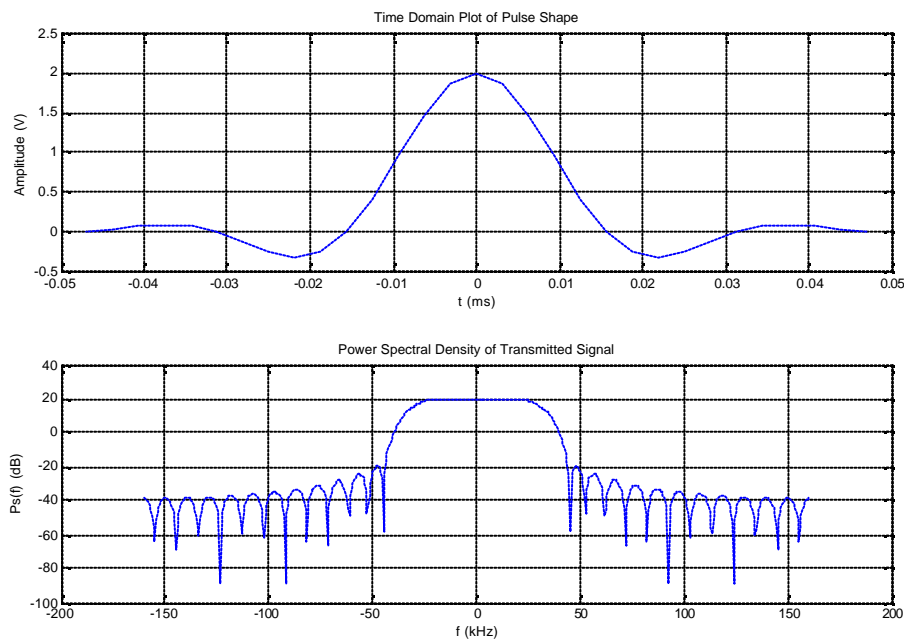
Para todos los esquemas de modulación se han realizado previamente varias pruebas, más de las que se muestran, y solamente se presentan aquí algunas de las más representativas y que cumplen con las condiciones requeridas. Considerando esto último, la decisión que se tome al final del sistema estará relacionado con el costo que importa el diseño, y la evaluación de los parámetros de funcionamiento, por cuanto todos cumplen

con los requerimientos de diseño como ancho de banda, atenuación y BER, y con los requerimientos de funcionamiento y simulación como son la fiabilidad y utilización de potencia, y simplicidad.

Por tanto se puede observar las siguientes simulaciones y luego decidir cual será la que convendría implementar:

1. Modulación BPSK, factor de rolloff de 0.4, detección coherente, filtro casado, código de corrección de errores,  $E_b/N_0$  de 8.7 dB y 40000 iteraciones.

**mod\_sim (2,0.4, 0, 1, 1, 8.7, 40000);**

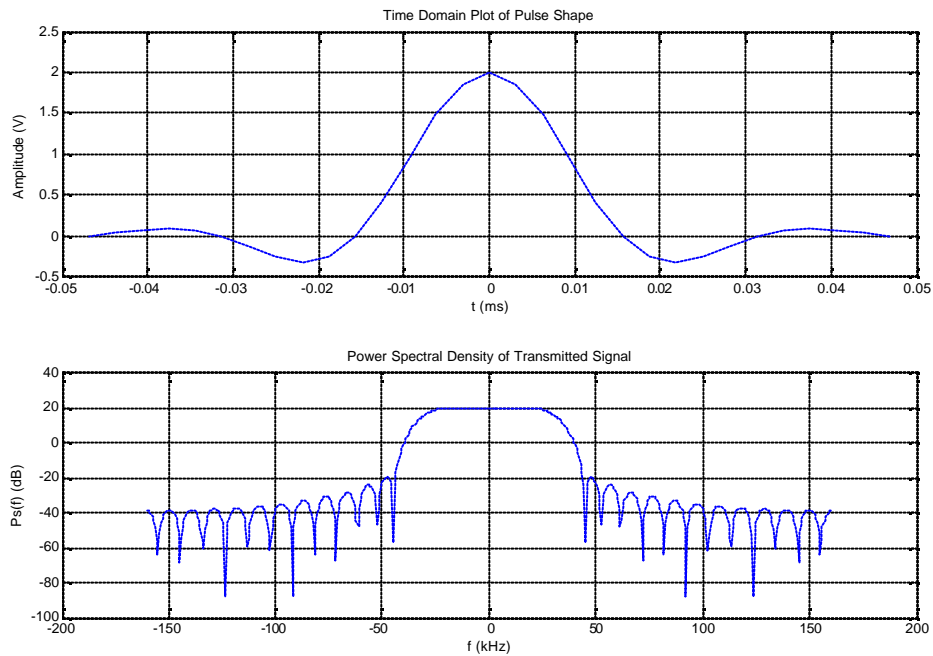


**Figura. 7.9. Modulación BPSK,  $a=0.4$ , Coherente, Filtro Casado y Detección de Errores**

Reliability = 1.0746  
 cost = 18.0600  
 ber = 2.5388e-010

2. Modulación BPSK, factor de rolloff de 0.4, detección diferencial, filtro pasabajos simple, código de corrección de errores,  $E_b/N_0$  de 8.7 dB y 40000 iteraciones.

```
mod_sim(2, 0.4, 1, 0, 1, 8.7, 40000);
```

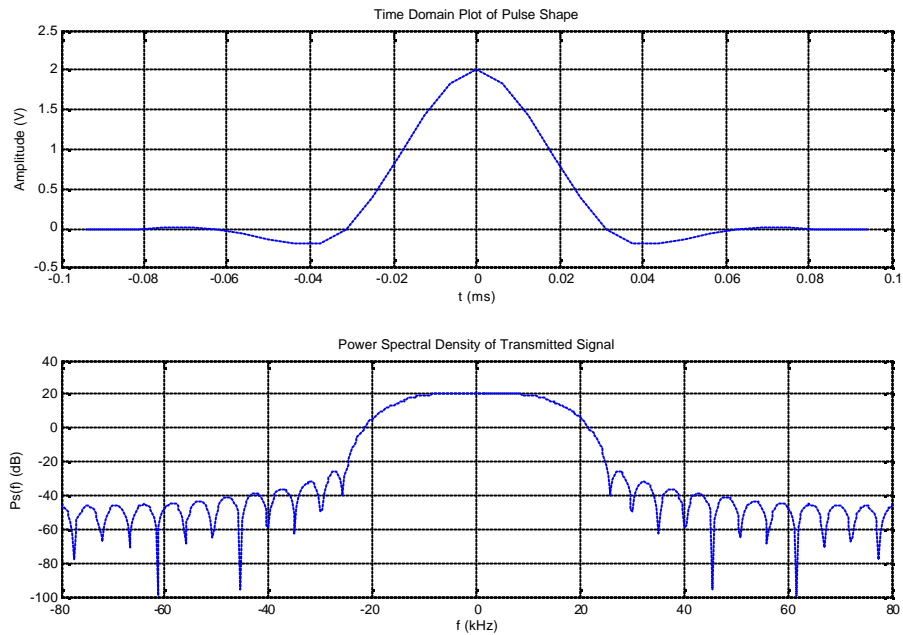


**Figura. 7.10. Modulación BPSK,  $a= 0.4$ , Diferencial, Filtro Pasabajos y Detección de Errores**

```
Reliability = 1.5199
cost = 12.5600
ber = 4.8415e-005
```

3. Modulación QPSK, factor de rolloff de 0.6, detección coherente, filtro casado, código de corrección de errores,  $E_b/N_0$  de 9 dB y 30000 iteraciones.

```
mod_sim(4, 0.6, 0, 1, 1, 9, 30000);
```



**Figura. 7.11. Modulación QPSK,  $a=0.6$ , Coherente, Filtro Casado y Detección de Errores**

Reliability = 1.0966  
 cost = 20.1600  
 ber = 4.1108e-011

4. Modulación QPSK, factor de rolloff de 0.8, detección diferencial, filtro pasabajos simple, código de corrección de errores,  $E_b/N_0$  de 8.7 dB y 20000 iteraciones.

**mod\_sim(4, 0.8, 1, 0, 1, 8.7, 20000);**

Reliability = 1.6196  
 cost = 14.2400  
 ber = 8.0133e-004

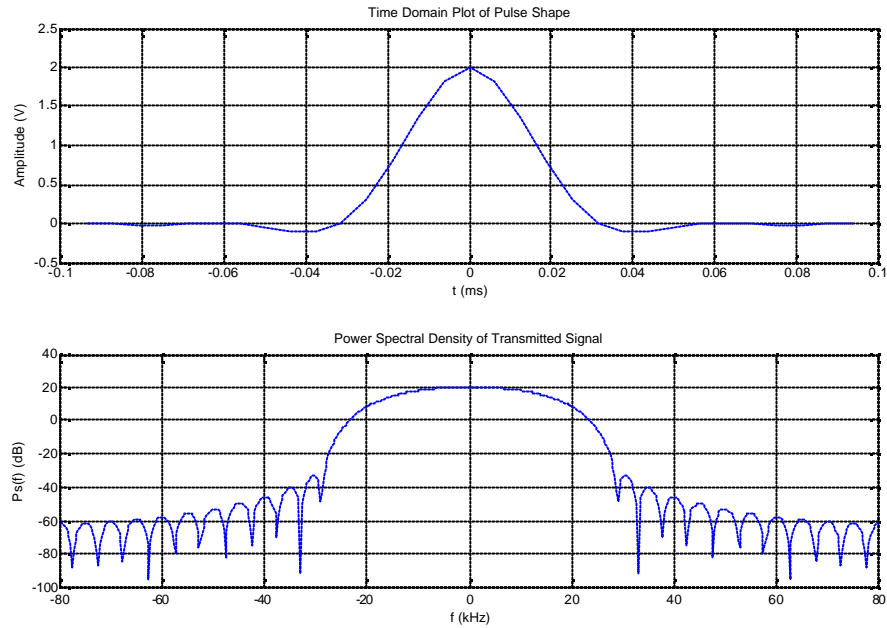


Figura. 7.12. Modulación QPSK,  $a= 0.8$ , Diferencial, Filtro Pasabajos y Detección de Errores

5. Modulación 8-PSK, factor de rolloff de 0.9, detección coherente, filtro casado, código de corrección de errores,  $E_b/N_0$  de 9 dB y 30000 iteraciones.

**mod\_sim (8, 0.9, 0, 1, 1, 9, 30000);**

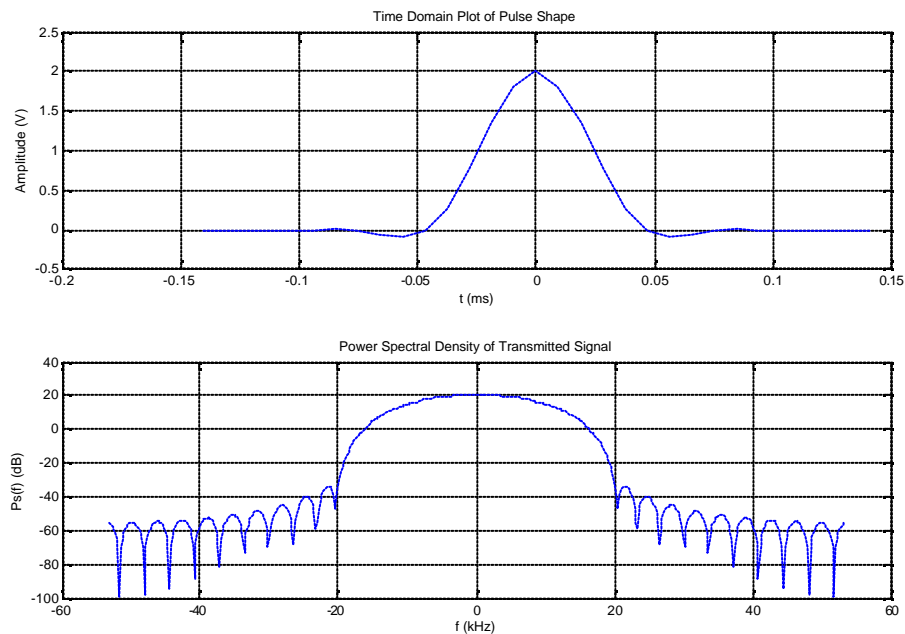
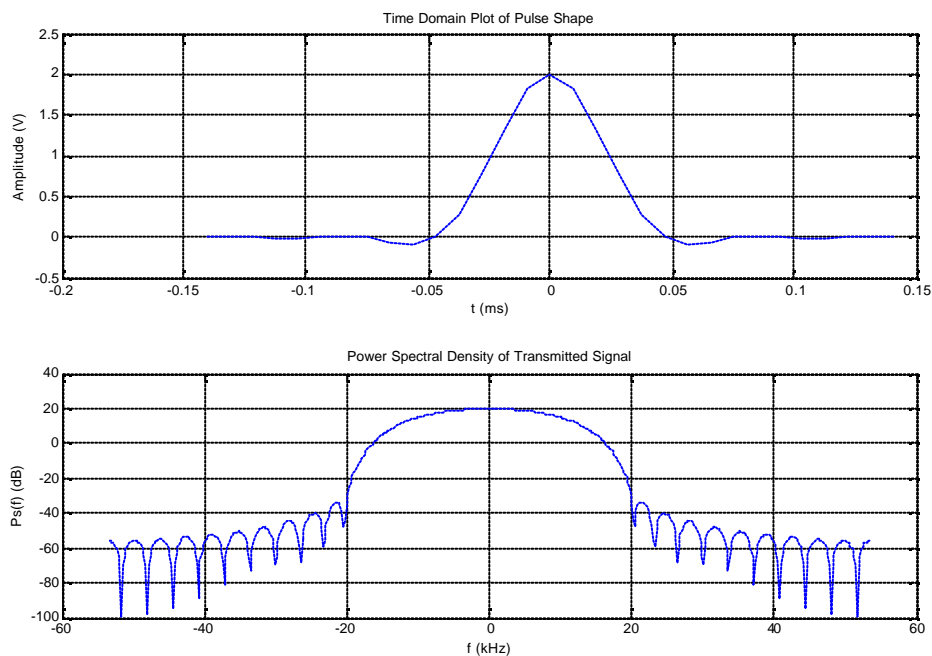


Figura. 7.13. Modulación 8PSK,  $a= 0.9$ , Coherente, Filtro Casado y Detección de Errores

Reliability = 1.6411  
 cost = 22.0100  
 ber = 1.1025e-005

6. Modulación 8-PSK, factor de rolloff de 0.9, detección diferencial, filtro pasabajos simple, código de corrección de errores,  $E_b/N_0$  de 9.6 dB y 30000 iteraciones.

**mod\_sim (8, 0.9, 1, 0, 1, 9.6, 30000);**



**Figura. 7.14. Modulación 8PSK,  $a=0.9$ , Diferencial, Filtro Pasabajos y Detección de Errores**

Reliability = 1.7648  
 cost = 19.1100  
 ber = 3.0764e-004

7. Modulación BPSK, factor de rolloff de 0.4, detección coherente, filtro casado, sin código de corrección de errores,  $E_b/N_0$  de 8.7 dB y 40000 iteraciones.



```
mod_sim(2,0.4,0, 1, 0, 8.7, 40000);
```

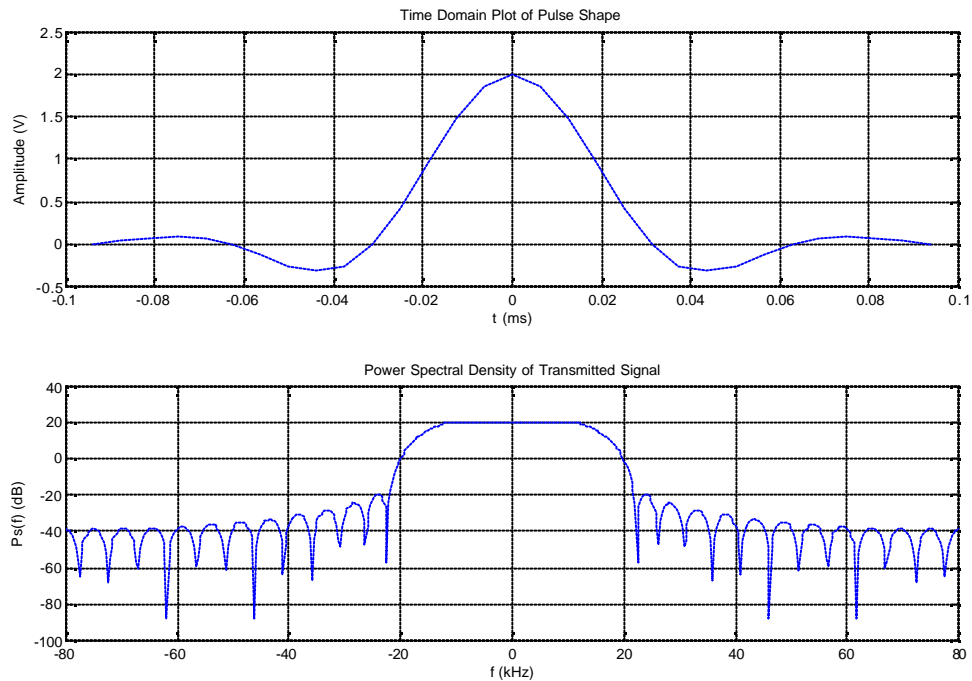


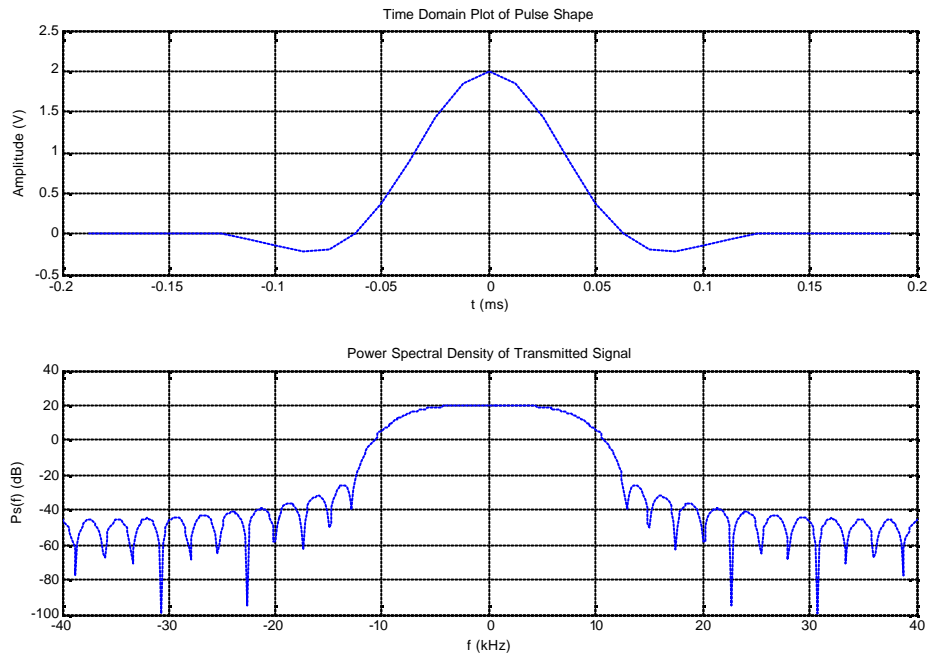
Figura. 7.15. Modulación BPSK,  $a = 0.4$ , Coherente, Filtro Casado y Sin detección de Errores

```
Reliability = 0.3010
cost = 16.5600
ber = 1.0000e-004
```

8. Modulación QPSK, factor de rolloff de 0.6, detección coherente, filtro casado, sin código de corrección de errores,  $E_b/N_0$  de 9 dB y 30000 iteraciones.

```
mod_sim(4, 0.6, 0, 1, 0, 9, 30000);
```

```
Reliability = 0.2386
cost = 18.6600
ber = 5.0000e-005
```



**Figura. 7.16. Modulación QPSK,  $a= 0.6$ , Coherente, Filtro Casado y Sin detección de Errores**

9. Modulación 8-PSK, factor de rolloff de 0.9, detección coherente, filtro casado, sin código de corrección de errores,  $E_b/N_0$  de 9 dB y 30000 iteraciones.

```
mod_sim(8, 0.9, 0, 1, 0, 9, 30000);
```

Reliability = 1.1937

cost = 20.5100

ber = 0.0027

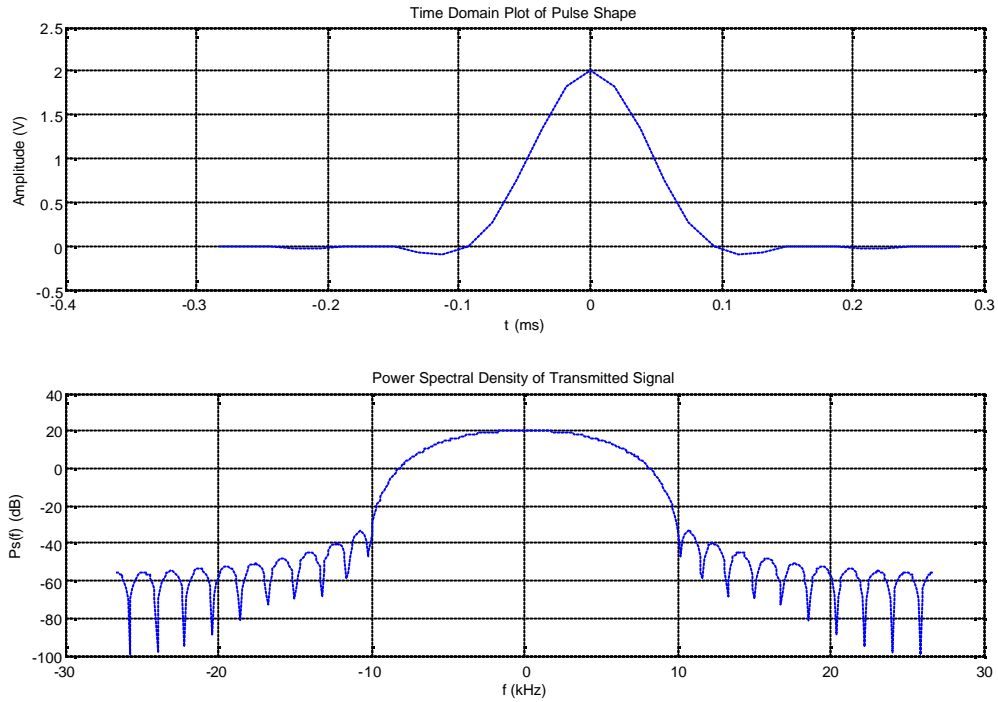


Figura. 7.17. Modulación 8PSK,  $a = 0.9$ , Coherente, Filtro Casado y Sin detección de Errores

### 7.3.1.2.6. Análisis y Selección de Modulación.

Por lo que se ha mostrado en las gráficas de simulación, con todos los sistemas se cumple con los requerimientos de diseño, funcionamiento y optimización de recurso, claro está que hay sistemas más económicos que otros y cuyas respuestas son diferentes pero caen dentro de los rangos sugeridos de funcionamiento.

Se ha seleccionado para este caso, como sistema de modulación y forma de pulso más conveniente y de mejores prestaciones, el sistema cuyos parámetros de la simulación son:

- Modulación BPSK.
- Factor de rolloff 0.4.
- Detección Diferencial, (1).
- Filtro pasabajos simple, (0).
- Código de Corrección de Error, (1).

Como se mostró antes, el comando para su ejecución en MATLAB quedaría de la siguiente forma:

```
mod_sim(2,0.4,1,0, 1, 8.7, 40000);
```

Sus demás características de respuesta son:

- Reliability = 1.5199
- cost = 12.5600
- ber = 4.8415e-005

Se puede observar que es el más económico de todos los sistemas presentados y su fiabilidad es bastante alta. Su BER se encuentra dentro de los límites requeridos incluso más de 10 veces menor. Como se aprecia en la figura 7.10, este diseño de sistema cumple con el ancho de banda y atenuación requeridas, y con el BER.

La selección de las características se realizó de acuerdo a los siguientes criterios:

- **Modulación:** BPSK, aunque QPSK (un modelo de similares características también mostrado) hace un mejor uso del ancho de banda, pero este BPSK se escoge por aminoramiento de costos y además su ancho de banda está dentro de los límites de diseño planteados. Debemos también considerar el hecho de que el uso de código de corrección de error duplica la necesidad de ancho banda y con todo esto, podemos ver en el gráfico que el lóbulo principal se encuentra dentro de los 50KHz. Su implementación es la más económica de todos los modelos PSK propuestos para el diseño, y es la más simple en términos de dispositivos y configuración electrónicos.
- **Factor de rolloff 0.4:** Es un valor que se ha escogido de forma de obtener un mayor uso del ancho de banda disponible y evitar los problemas de lóbulos laterales aumentados que se presentan con el uso de factores de rolloff muy pequeños, ya que este valor no es tan alto, pero se considera más bien medio entre ambos. Se mejora notablemente la respuesta de la señal a transmitirse con el uso de factores de

este factor y en términos de costo, este valor nos resulta de precio medio, aunque si fuera mayor, su costo sería menor.

- **Detección Diferencial:** A pesar de que la detección coherente nos proporciona gran desempeño de recepción, al realizar las diferentes pruebas se pudo observar que la respuesta gráfica no varía significativamente entre tener detección diferencial o coherente, pero si varia notablemente sus costo. Se presenta pues en las simulaciones que no hay mayor diferencia en usar uno u otro sistema de recepción tan significativo como su coste.
- **Filtro Pasabajos:** Se lo seleccionó por ahorro en costos, porque la respuesta del sistema no varía considerablemente (es imperceptible), en relación al uso de un filtro casado en el sistema. También de una u otra forma se tiene un ahorro de  $E_b/N_0$ , que en la práctica nos representa energía de transmisión y mayor coste. Además, esto es algo que ya se viene realizando en algunos receptores celulares de uso comercial y su funcionamiento esta dentro de los parámetros requeridos.
- **Código de Corrección de Error:** Este parámetro se ha llegado a determinar como esencial en el diseño de este sistema de comunicación, y así se ha demostrado que podemos dejar de hacer un receptor con detección coherente, podemos dejar de usar filtros casados, o modificar el  $E_b/N_0$ , o incluso el sistema de modulación, sin que estos cambios afecten gravemente nuestro diseño y sus respuestas, pero, el cambio de este parámetro, es decir su no presencia, realmente afecta todo el diseño. Para poder apreciar esto en toda su magnitud se ha realizado una repetición de las simulaciones 1, 3 y 5 de BPSK, QPSK, 8-PSK, con sus mismos parámetros (coherentes, filtros casados, etc.), pero con la diferencia de no presentar código de corrección de errores, y las respuestas son realmente invalidas para nuestros requerimientos, no cumplen con la fiabilidad del sistema, además no hay mucha diferencia en el costo de implementación y su BER crece descontroladamente. Es decir, de lo observado, el desempeño real de este sistema de comunicación celular con las características de diseño planteadas se arregla o daña abruptamente con la inclusión de la corrección de error, aunque se desprecien otros parámetros de diseño no menos importantes pero que si marcan mucha mayor diferencia de costos.

### 7.3.1.3. Diseño del Esquema de Acceso Múltiple y Presupuesto del Enlace

En la parte primera, se diseñó un esquema de cuantizador diferencial para el sistema celular que era capaz de transmitir una señal de voz con una  $SNR > 25$  dB a una tasa de datos de  $32 \text{ kbps} = 8000 \text{ muestras/seg} * 4 \text{ bits/muestra}$ . En la parte segunda se diseñó la modulación y forma de pulso para que la señal transmitida tenga un ancho de banda de 50 KHz y el BER no sea mayor a 0.001. En la parte tercera, se debe elegir un esquema de acceso múltiple para este sistema de comunicaciones.

#### 7.3.1.3.1. Técnica de Acceso Múltiple.

Cada uno de los enlaces requiere de un ancho de banda de 50 KHz. Se dispone de un total de 7 MHz de ancho de banda disponible para el sistema. Se puede escoger entre dos opciones para la técnica de Acceso Múltiple del sistema:

#### **FDMA/TDMA**

En este esquema se utilizará 4 usuarios por canal para crear 35 canales con un ancho de banda de 200 KHz cada uno. Esto tiene la ventaja de la simplicidad, pero solo 1/7 de los canales pueden ser usados en cada celda.

#### **CDMA**

Si se elige utilizar CDMA como técnica de acceso múltiple, se obtendrá una ganancia de procesamiento de 140. El número de usuarios por celda será determinado mediante la fórmula de aproximación Gaussiana para el BER de un sistema CDMA. Se debe asumir el mismo  $E_b/N_0$  que calculó para la parte 2, y se debe asegurar que con la interferencia de acceso múltiple, el BER no ascienda a más de 0.002.

### 7.3.1.3.2. Desarrollo.

Para realizar la elección del mejor sistema de acceso múltiple entre CDMA y TDMA para este diseño, es necesario primero calcular el número de usuarios por celda que se pueden obtener tomando en cuenta las condiciones requeridas en el planteamiento del problema, entonces:

#### FDMA/TDMA:

- Usuarios por canal: 4
- Canales: 35.
- Ancho de Banda: 200KHz por canal.
- Factor de reutilización (N): 1/7.

Canales por celda

$$N * \#Canales = (1/7)*35 = 5$$

#Usuarios:

$$k = (\#Channels / cell) \times (\#Users / Channel)$$

$$k = 5 \times 4 = 20users / cell$$

GSM es una estándar que trabaja con TDMA/FDMA, con la consideración de que FDMA es la responsable de partir los 25 MHz de ancho de banda asignados para GSM en 124 frecuencias portadoras de 200KHz cada una. TDMA en cambio se utiliza por GSM para obtener una división de ocho ranuras de tiempo.

En lo que se refiere a la planificación del sistema, la ITU ha asignado las frecuencias de 890-915 MHz para el enlace de subida (uplink), y las frecuencias de 935-960Mhz para el enlace de bajada (downlink). La codificación de canal usada por GSM es la convolucional y entrelazado de bloques, pero los algoritmos utilizados dependen de la tasa de transmisión deseada y del speech. La codificación de voz usada en GSM es RPELPC (Regular Pulse Excited - Linear Predictive Coder) con un lazo de predicción a largo tiempo, el cual emplea una tasa de 13kbps.

**CDMA:**

- $G_p = 140$
- $BER \leq 0.002$

Para el cálculo de usuarios se utiliza la fórmula de aproximación Gaussiana del BER, se tiene la siguiente expresión:

$$P_b = Q\left(\sqrt{\frac{3xG_p}{k-1}}\right)$$

Para el análisis se puede partir del caso en que  $BER = 0.002$ , de donde se obtiene:

$$0.002 = Q\left(\sqrt{\frac{3xG_p}{k-1}}\right)$$

Entonces para  $Q(x) = 0.002$ , se calcula que  $x = 2.88$  y con esto se puede obtener el número de usuarios:

$$2.88 = \sqrt{\frac{3xG_p}{k-1}}$$

$$k = \frac{3 \times 140}{2.88^2} = 51.63$$

$$k \approx 52 \text{ usuarios}$$

CDMA utiliza modulación SPREAD SPECTRUM y por medio de ella se pueden manejar de 12 a 45 usuarios por celda. En caso de haber muchos usuarios, la estación base pide a las móviles que aumenten su potencia de transmisión. Además, otra capacidad de CDMA es la de SOFT-HAND OFF

**7.3.1.3.3. Elección.**

El método de multiacceso seleccionado para este diseño es CDMA, principalmente por la relación costo beneficio. Aunque los equipos terminales de CDMA son más costosos



que los de TDMA/FDMA (estándar GSM), el número de usuarios permitidos en un sistema CDMA es más alto. Además, el espectro radioeléctrico es un recurso muy limitado, por lo que se debe de tratar de explotarlo al máximo, es decir, tratar de obtener un mayor número de abonados que compartan un mismo ancho de banda. Por los cálculos mostrados, se puede observar fácilmente que el número de abonados posibles en una celda de CDMA es el doble que en una celda TDMA.

Otra ventaja de CDMA sobre TDMA es la fácil planificación del sistema, porque en TDMA es necesario realizar la división de canales y fijarse que no exista problemas de solapamiento con las celdas adyacentes (evitar interferencia de co-canal y canal adyacente); mientras que para CDMA simplemente requiere de una frecuencia, y la repartición de códigos PN con los cuales se identifica cada usuario. En CDMA no hace falta dividir el ancho de banda en canales, todos los usuarios comparten la misma frecuencia.

Tal vez el único problema de CDMA sucede cuando existen muchos usuarios tratando de acceder a la red o que se encuentran en una zona donde más de una estación base tiene cobertura, en este caso sucede lo que se conoce como “channel pollution” o contaminación del canal.

Una gran ventaja de CDMA sobre TDMA también es el SOFT-HAND OFF, el mismo proceso que en los sistemas CDMA no es notorio, simplemente porque todas las estaciones base trabajan con la misma frecuencia y el móvil puede escoger entre varios códigos PN. Esto nos garantiza que el paso de una celda a otra de un usuario sea mucho más fácil y existan menos posibilidades de interrumpir la llamada. Si se trabajara con TDMA, existe un HARD HAND-OFF, en el cual, el proceso de ser aceptado en otra celda requiere de mucho más tiempo y es un poco más complicado.

Finalmente, CDMA permite la expansión con mayor rapidez y rentabilidad, debido a que requiere de menor número de emplazamientos (división de celdas en sectores) y algo determinante en la selección es que, CDMA se caracteriza por una gran seguridad debido a que utiliza modulación Spread Spectrum de secuencia directa, la cual es operada y controlada con la utilización de pseudos códigos.

## 7.4. REFERENCIAS

### 7.4.1. LIBROS

- SKLAR, Bernard, **Digital Communications**, Segunda Edición, Prentice Hall, Estados Unidos 2001, páginas 660-676.
- ESPE, **Comunicación Digital**, apuntes de la asignatura, semestre septiembre 2004 – marzo 2005.

### 7.4.2. INTERNET

- <http://www.yucatan.com.mx/especiales/celular/tecnologiascelulares.asp>, Tecnologías Celulares.
- [http://www.upv.es/satelite/trabajos/Grupo13\\_99.00/tema3.html](http://www.upv.es/satelite/trabajos/Grupo13_99.00/tema3.html), Tecnicas de Acceso Multiple.
- <http://html.rincondelvago.com/fdma.html>, Acceso Múltiple por División de Frecuencia.
- [http://www.andreas\\_chwope.de/ASIC\\_s/Schnittstellen/Data\\_Lines/body\\_multiplexing.html](http://www.andreas_chwope.de/ASIC_s/Schnittstellen/Data_Lines/body_multiplexing.html), Multiplexing TDMA, FDMA, CDMA.
- <http://scholar.lib.vt.edu/theses/available/etd-04262000-15330030/unrestricted/ch5.pdf>, SDMA.
- [http://www.ece.rochester.edu/courses/ECE586/lectures/MAC\\_I.pdf](http://www.ece.rochester.edu/courses/ECE586/lectures/MAC_I.pdf), Media Access Control.
- [http://www.inf.fu-berlin.de/inst/ag-tech/resources/material/English/PDF-Handout/C03-Media\\_Access.pdf](http://www.inf.fu-berlin.de/inst/ag-tech/resources/material/English/PDF-Handout/C03-Media_Access.pdf), Acceso al Medio Comunicaciones Móviles.
- <http://www.eveliux.com/index.php?option=content&task=view&id=12>, Tecnología de Comunicaciones Inalámbrica.
- <http://cnx.rice.edu/content/m10107/latest/>, TDMA.
- <http://html.rincondelvago.com/multiplexacion-de-canales.html>, Multiplexación de Canales.

- <http://www.iec.org/online/tutorials/tdma/topic06.html>, TDMA vs. CDMA.
- [http://www.mobilecomms-technology.com/projects/tdma\\_is136/](http://www.mobilecomms-technology.com/projects/tdma_is136/), TDMA IS 136.
- <http://computing-dictionary.thefreedictionary.com/SDMA>, SDMA
- [http://paleale.eecs.berkeley.edu/~varaiya/papers\\_ps.dir/Bana\\_Varaiya.pdf](http://paleale.eecs.berkeley.edu/~varaiya/papers_ps.dir/Bana_Varaiya.pdf), SDMA.
- [http://searchnetworking.techtarget.com/gDefinition/0,294236,sid7\\_gci501705,00.html](http://searchnetworking.techtarget.com/gDefinition/0,294236,sid7_gci501705,00.html), SDMA.
- <http://www.nicatech.com.ni/modules.php?name=Encyclopedia&op=content&tid=359>, SDMA.
- [http://bornova.ege.edu.tr/~omermer/dersler/comm\\_sys2/design\\_projects/proj1.pdf](http://bornova.ege.edu.tr/~omermer/dersler/comm_sys2/design_projects/proj1.pdf), Proyecto Diseño Celular.
- [http://bornova.ege.edu.tr/~omermer/dersler/comm\\_sys2/design\\_projects/proj2.pdf](http://bornova.ege.edu.tr/~omermer/dersler/comm_sys2/design_projects/proj2.pdf), Proyecto Diseño Celular.
- [http://bornova.ege.edu.tr/~omermer/dersler/comm\\_sys2/design\\_projects/proj3.pdf](http://bornova.ege.edu.tr/~omermer/dersler/comm_sys2/design_projects/proj3.pdf), Proyecto Diseño Celular.

## CAPÍTULO VIII

### ESQUEMAS ESPECIALES

#### 8.1. INTRODUCCIÓN.

Además de los esquemas de modulación que se revisaron en los capítulos anteriores, en banda base o pasa banda, existen otros que se aplican de igual forma para optimizar los recursos de comunicación como el ancho de banda y potencia de transmisión, mejorar la probabilidad de error, disminuir costos de implementación, así como la facilidad de hacerlos prácticos.

Entre estos esquemas tenemos CAP / DMT, TCM y OFDM, que son ampliamente utilizados en los sistemas de comunicación en la actualidad, en especial este último que cada día presenta nuevas mejoras y prestaciones que lo están convirtiendo en una verdadera revolución tecnológica.

DMT usa multi-carrier para dividir el ancho de banda en varios subcanales y transmitir un número de bits variables en cada uno de ellos después de modular una subportadora; CAP usa una sola portadora la cual es modulada un similar al QAM. Tanto CAP como DMT han sido utilizados para XDSL el cual es usado para transmitir datos digitales usando módems de la banda de voz, debido a que a través de la línea telefónica se obtienen varios servicios. En una primera etapa coexistieron dos técnicas de modulación para el ADSL: CAP ("Carrierless Amplitude/Phase") y DMT ("Discrete MultiTone"). Finalmente los organismos de estandarización (ANSI, ETSI e ITU) se han decantado por la solución DMT.

La modulación Codificada de Trellis es una combinación de modulación y esquemas de código que logra mejorar la probabilidad de error de bit sin expansión del ancho de banda de la señal. Combina señalización de modulación multinivel / multifase con el esquema de codificación de Trellis mostrado en el Capítulo 4 sobre corrección de errores, para transmisiones digitales sobre canales con banda limitada. De igual forma, se usa el algoritmo de Viterbi para la decodificación de estos sistemas, lo que asegura una ganancia de alrededor de 3 dB con respecto a sistemas no codificados.

Este esquema de modulación se basa en **el mapeo por particionamiento de conjuntos** y fue desarrollado en 1982 por Ungerboeck. Permitted que los módems de PC aumentaran su velocidad de 9,6 Kbps a 56 Kbps sobre canales telefónicos. En otras palabras aceleró la transmisión un 583% mediante la teoría de codificación de canal y la modulación de alto nivel. Pero TCM no se quedó ahí, su concepto revolucionó todas las telecomunicaciones desde la transmisión de datos por canales telefónicos hasta las últimas generaciones de transmisión satelital.

La evolución en los últimos años de las Tecnologías de la Información y de las Comunicaciones (TIC's) viene marcada por un aumento en la demanda tanto del ancho de banda como de la calidad en los servicios al usuario. En este sentido han surgido nuevas técnicas de modulación digital de elevada eficiencia espectral como OFDM (Orthogonal Frequency Division Multiplexing).

La técnica OFDM divide un flujo binario de banda ancha en N canales que son transmitidos simultáneamente empleando portadoras ortogonales. Esta modulación supera las limitaciones de las modulaciones tradicionales en canales que presentan dispersión en tiempo y frecuencia.

La técnica OFDM se emplea en sistemas de radiodifusión de televisión digital (ESTI-DVB-T) y radiodifusión sonora digital (ETSI-DAB), en redes de área local inalámbricas (ESTI-HIPERLAN/2 e IEEE-WLAN), en redes de área metropolitana inalámbricas (ETSI-HIPERMAN), y comienza a emplearse en las primeras realizaciones de sistemas de comunicaciones por línea de potencia (denominados PLC o PLT). Se utiliza en algunos tipos de redes radio (WiFi o WiMAX).

## 8.2. MARCO TEÓRICO

### 8.2.1. CAP / DMT

#### 8.2.1.1. Discrete Multitone – DMT

Utilizado por la mayoría de los equipos para ADSL, divide las señales en 247 canales separados, cada 4 Khz. de ancho. Un ejemplo de esto es como que la línea telefónica fuera dividida en 247 líneas de 4 Khz. y le pusiera un módem a cada una. Cada canal es monitoreado y si la calidad falla la señal es movida a otro canal. Este sistema constantemente esta moviendo la señal de canal en canal. Se busca el mejor canal para la transmisión y recepción. Además algunos de los canales bajos (cerca de 8 Khz.) Son usados como canales bidireccionales para subida y bajada de datos.



Figura. 8.1. Esquema de Discrete MultiTone DMT

Variando el número de bits por símbolo en un canal, DMT puede ser de tasa adaptativa. La ITU 992.1 (G.dmt) y la ITU 992.2 (G.lite) usan DMT para la transmisión de datos.

DMT básicamente consiste en el empleo de múltiples portadoras y no sólo una, que es lo que se hace en los módems de banda vocal. Cada una de estas portadoras (denominadas subportadoras) es modulada en cuadratura (modulación QAM) por una parte del flujo total de datos que se van a transmitir. Estas subportadoras están separadas entre sí 4,3125 Khz, y el ancho de banda que ocupa cada subportadora modulada es de 4 KHz. El reparto del flujo de datos entre subportadoras se hace en función de la estimación de la relación Señal/Ruido en la banda asignada a cada una de ellas. Cuanto mayor es esta relación, tanto

mayor es el caudal que puede transmitir por una subportadora. Esta estimación de la relación Señal/Ruido se hace al comienzo, cuando se establece el enlace entre el ATU-R y el ATU-C, por medio de una secuencia de entrenamiento predefinida.

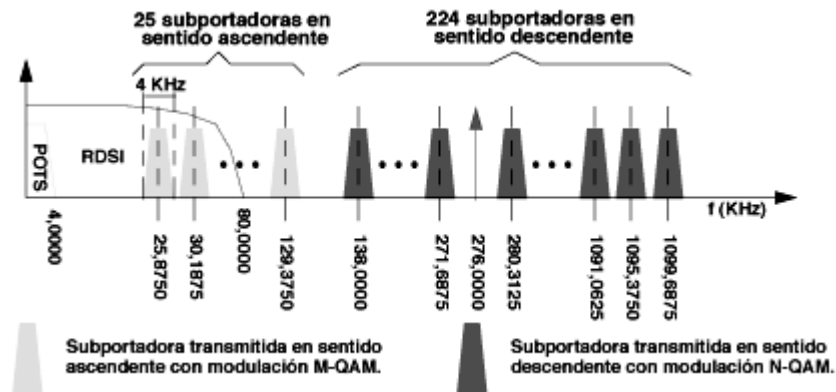


Figura. 8.2. Modulación ADSL DMT con FDM

La técnica de modulación usada es la misma tanto en el ATU-R como en el ATU-C. La única diferencia estriba en que el ATU-C dispone de hasta 256 subportadoras, mientras que el ATU-R sólo puede disponer como máximo de 32. La modulación parece y realmente es bastante complicada, pero el algoritmo de modulación se traduce en una IFFT (Transformada Rápida Inversa de Fourier) en el modulador, y en una FFT (Transformada Rápida de Fourier) en el demodulador situado al otro lado del bucle. Estas operaciones se pueden efectuar fácilmente si el núcleo del módem se desarrolla sobre un DSP.

El modulador del ATU-C, hace una IFFT de 512 muestras sobre el flujo de datos que se ha de enviar en sentido "downstream". El modulador del ATU-R, hace una IFFT de 64 muestras sobre el flujo de datos que se ha de enviar en sentido "upstream". El demodulador del ATU-C, hace una FFT de 64 muestras tomadas de la señal "upstream" que recibe. El demodulador del ATU-R, hace una FFT, sobre 512 muestras de la señal "downstream" recibida.

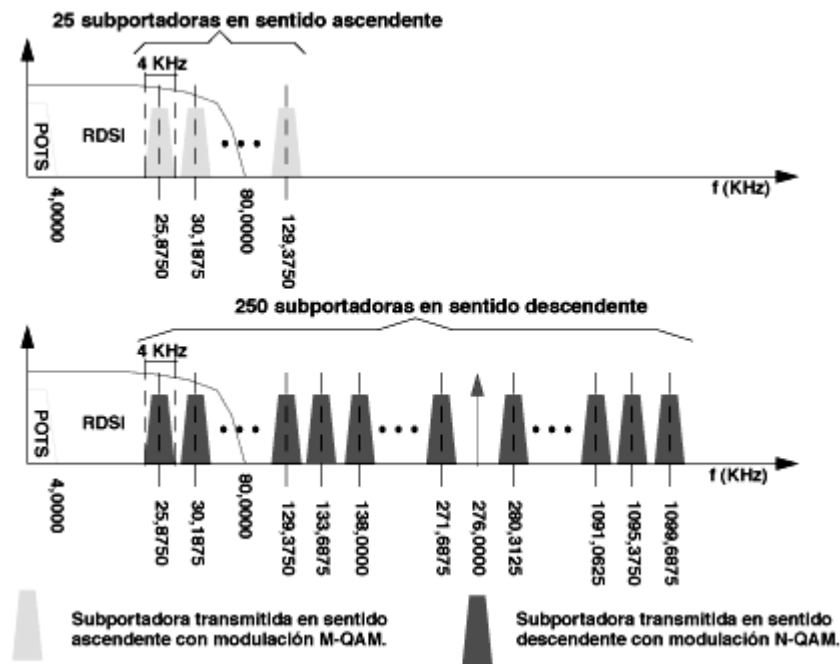


Figura. 8.3. Modulación ADSL DMT con Cancelación de Ecos

En las dos figuras anteriores se han presentado las dos modalidades dentro del ADSL con modulación DMT: FDM y cancelación de ecos. En la primera, los espectros de las señales ascendente y descendente no se solapan, lo que simplifica el diseño de los módems, aunque reduce la capacidad de transmisión en sentido descendente, no tanto por el menor número de subportadoras disponibles como por el hecho de que las de menor frecuencia, aquéllas para las que la atenuación del par de cobre es menor, no están disponibles. La segunda modalidad, basada en un cancelador de ecos para la separación de las señales correspondientes a los dos sentidos de transmisión, permite mayores caudales a costa de una mayor complejidad en el diseño. Como se puede ver, los espectros nunca se solapan con la banda reservada para el servicio telefónico básico (POTS o "Plain Old Telephone Service"), y en cambio sí que se solapan con los correspondientes al acceso básico RDSI. Por ello el ADSL y el acceso básico RDSI son incompatibles.

Los nuevos estándares sobre ADSL han llevado al desarrollo de una nueva generación de módems capaces de transmitir hasta 8,192 Mbps en sentido descendente y hasta 0,928 Mbps en sentido ascendente. Con estas cifras, está claro que el despliegue de esta tecnología supone una auténtica revolución en la red de acceso de las operadoras del servicio telefónico. Pasan de ser redes de banda estrecha capaces de ofrecer únicamente



telefonía y transmisión de datos vía módem, a ser redes de banda ancha multiservicio. De este modo los usuarios podrán disponer de un abanico de servicios inimaginables hasta hace poco. Y todo ello sin afectar a un servicio básico como es la telefonía.

La eficiencia del ancho de banda en cada sub banda se extiende desde 0 hasta 15 bps/Hz. A bajas frecuencias, la atenuación del cable del cobre es baja y SNR es alta la eficiencia es de 10 b/s/ Hz

#### 8.2.1.1.1. Loading.

El proceso de asignar bits de información y energía a cada sub canal es llamado loading. El número de bits asignados al n-ésimo subcanal esta dado por:

$$b_n = \log_2 \left( 1 + SNR \frac{n}{g} \right) \quad (1)$$

Donde  $g$  cuantifica la pérdida efectiva en SNR con respecto a la capacidad, cuyo valor es uno cuando no hay pérdidas en SNR. La tasa de datos total sobre  $N$  canales esta dada por:

$$B = S b_n \quad n = 1, 2, \dots, N \quad (2)$$

El proceso sería fácil: usar la ecuación (1) para asignar los bits y luego la ecuación (2) para obtener el total, pero hay que tomar en cuenta que  $b_n$  es variable en los subcanales y deben ser redondeados a cero. Además para mantener igual tasa de error en todos los canales se usa el scaling en todos los niveles transmitidos obteniendo nuevas SNR. Hay un parámetro  $g_n$  definido por:

$$g_n = [2b_{nr} - 1] / [2b_n - 1]$$

$b_{nr}$  es el valor redondeado de  $b_n$ . El parámetro  $g_n$  varia alrededor de 1 (0.85 - 1.15).

### 8.2.1.1.2. Partición de Canales.

La partición de canales se refiere a la construcción de un conjunto de canales paralelos independientes. Esto se implementa usando funciones bases para la modulación y demodulación. Las N funciones bases deben ser ortogonales y deben mantener esta característica después de la transmisión a través del canal cuya función de transferencia es conocida.

La estimación de la SNR para cada subcanal requiere dos mediciones:

- La respuesta del canal.
- La varianza de la respuesta del canal causada por el ruido del mismo el cual es producto de la suma del ruido convencional, crosstalk, y la distorsión residual después de la ecualización.

El número de símbolos usados debe ser suficiente para reducir el error de estimación de SNR a un nivel aceptable. Si el ruido se asume Gaussiano, un buen valor de SNR es de 8.686.

### 8.2.1.2. Carrierless Amplitude/Phase – CAP

CAP opera dividiendo las señales en tres bandas distintivas:

- La voz es transmitida en la banda de 0 a 4 KHz (POTS).
- El canal de subida (del usuario al servidor) está en la banda de 25 hasta 160 KHz.
- El canal de bajada (del servidor al usuario) está en la banda de 240 KHz. hasta 1.5 Mhz dependiendo del número de usuarios, condiciones de la línea, etc....

Este sistema con tres canales diferentes, minimiza la posibilidad de interferencias entre los canales de la línea o entre señales de diferentes líneas.

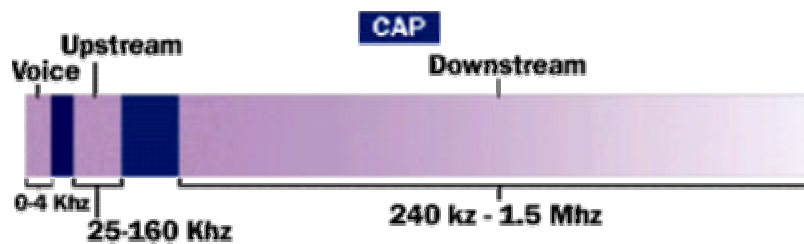


Figura. 8.4. CAP Divide la Línea en Tres Bandas Espaciadas

CAP es un sistema de simple portadora, parecido a QAM en donde dos portadoras ortogonales son moduladas y combinadas. La diferencia principal es que en CAP la modulación de la señal ortogonal es hecha digitalmente usando dos filtros pasa banda digitales con igual amplitud y una diferencia de fase  $\pi/2$ .

Luego las señales son combinadas y un DAC convierte la señal a analógica antes de la transmisión. Cada símbolo ocupa todo el ancho de banda del canal por una corta duración debido a una tasa de transmisión alta. Entonces este sistema es susceptible a errores debido a ruidos impulsivos (ruidos en el dominio del tiempo). Asimismo el ruido en el dominio de la frecuencia presente a lo largo de una frecuencia angosta también contribuye al error total.

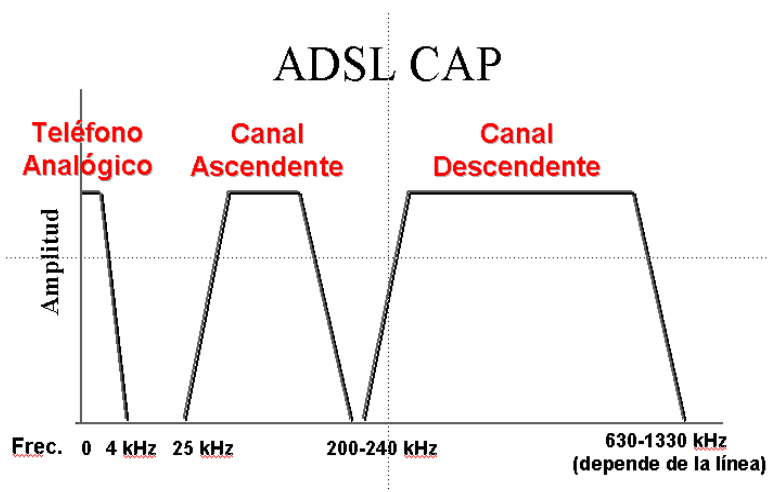


Figura. 8.5. Esquema de ADSL CAP

### 8.2.1.2.1. Transmisor.

La técnica de la modulación del CAP se relaciona de cerca con QAM en que la amplitud y la fase están utilizadas para representar la señal binaria. La diferencia entre el CAP y QAM es en la representación del patrón de la constelación. El CAP no utiliza una señal de portador para representar los cambios de la fase y de la amplitud.

La única diferencia en mezclarse de las dos señales moduladas: QAM genera las señales usadas que son dos señales del seno y del coseno mezcladas antes de transmitir, mientras que el CAP realiza esta operación digital que usa dos filtros pasa banda transversales digitales con la misma amplitud con respuesta de fase del  $90^\circ$ .

Los transmisores CAP no requieren los circuitos híbridos, puesto que transmiten y reciben en dos diversas bandas de frecuencia. Además, no requieren el trazado de circuito de la cancelación del eco y no son muy sensibles como el DMT. Sin embargo, los módems del CAP tienen circuito adicional llamado ecualizadores adaptantes.

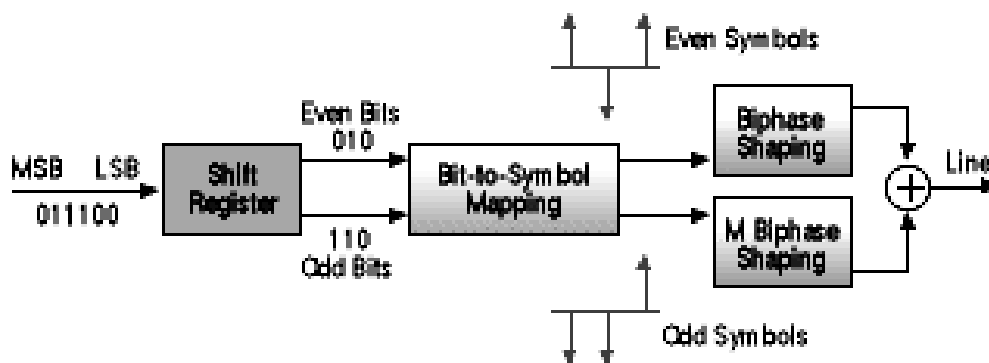


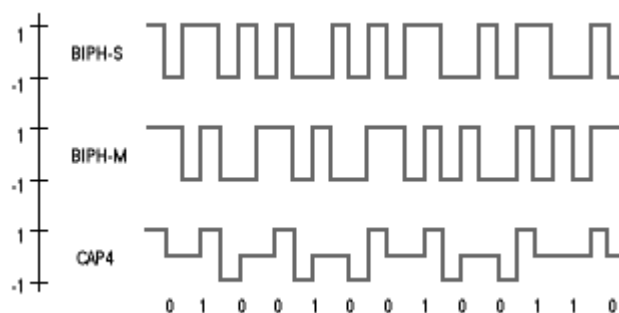
Figura. 8.6. Transmisor CAP Utilizando Desplazamiento de Fase

Al ser muy ancho el canal descendente la atenuación varía mucho en el rango de frecuencias (las frecuencias altas se atenúan mas). Para compensarlo se utiliza una ecualización adaptativa muy compleja. La función del ecualizador es crear la imagen del espejo de la distorsión en la línea causada por las discontinuidades en el lazo del servicio, que se miden durante la inicialización del módem. En vista del coste adicional para el

ecualizador, los sistemas del CAP son más fáciles y menos costosos de poner en ejecución que sistemas de la banda base.

Hay varias variaciones de la modulación del CAP:  $CAP_4$  a  $CAP_{64}$ . Dependiendo de la tasa de datos que se transmitirán y de la longitud, los sistemas del CAP se pueden para utilizar dondequiera a partir 4 a 512 estados de la amplitud y de la fase para codificar los datos. La regla que gobierna para  $CAP_4$  es para la modulación bifásica modificada, que la transición sucede en el comienzo del bit; un "1" da lugar a una transición en el centro del bit, y un "0" da lugar a ninguna transición. Para la modulación bifásica, la transición comienza en el comienzo del bit, los resultados "0" da lugar a una transición en el centro del bit, y un "1" en ninguna transición, como se ve en la figura 8.6. Las dos salidas fase-cambiadas de puesto se agregan para formar la forma de onda final  $CAP_4$  para la configuración de bits 0100100110110.

Para las órdenes más altas de la modulación del CAP, los filtros que forman se substituyen por los filtros en fase y de la cuadratura. Dos formas de onda se utilizan para codificar los bits. Según, el codificador substituye una corriente de datos digitales por una ecuación compleja que simbolice un punto en el diagrama de la constelación. Así, para un  $CAP_{32}$ , habría 32 localizaciones posibles en el diagrama, que se puede representar como vector que consiste en coordenadas verdaderas e imaginarias. Por lo tanto,  $CAP_{32}$  daría lugar a 32 ecuaciones distintas, cada una representando cinco bits de los datos.



**Figura 8.7. Modulación  $CAP_4$**

La fase próxima del transmisor se realiza con dos filtros digitales, que suprimen ciertas partes de la señal dependiendo de los valores en los coeficientes. Entonces, los resultados

digitales del filtro se restan y se pasan a través de un convertidor de D/A que, alternadamente, alimenta el filtro bajo del paso.

### 8.2.1.3. DMT vs. CAP

DMT fue elegido para ser el estándar ANSI de ADSL debido a las siguientes razones:

- **Optimización de subcanales:** DMT verifica cada uno de los subcanales y lo puede hacer durante intervalos regulares de tiempo para optimizar la tasa de transmisión de cada subcanal.
- **Adaptación de la tasa:** DMT tiene adaptación de tasa debido a que monitorea constantemente los subcanales
- **Inmunidad superior al ruido:** DMT usa una tasa de transmisión baja en un canal Ruidoso o incluso evitando canales con esas características.
- **Interoperabilidad con estándares:** DMT es un estándar abierto al cual tienen acceso todos los productores. Hay mucho soporte para chips de DMT, en cambio CAP tiene una sola fuente de apoyo Globespan Semiconductors.

### 8.2.2. TCM – TRELLIS CODED MODULATION

El primer esquema de TCM fue propuesto en 1976 por Gottfried Ungerboeck. Seguido por una publicación en la revista information theory del IEEE en el año de 1982, con la cual se dio una revolución en las investigaciones de su teoría y métodos, además se estableció un lugar a las posibles implementaciones de TCM.

Ungerboeck mostró que con un codificador / modulador como el mostrado en la figura 8.7. para cualquier número entero de bits por segundo por Hertz, es posible lograr una ganancia asintótica de codificación, tanto como 6 dB en la relación  $E_b/N_0$  con precisamente el mismo ancho de banda, pero duplicando el conjunto de constelación de señales de  $M = 2^{k-1}$  a  $M = 2^k$  y empleando una tasa de  $(k-1)/k$  para código convolucional.

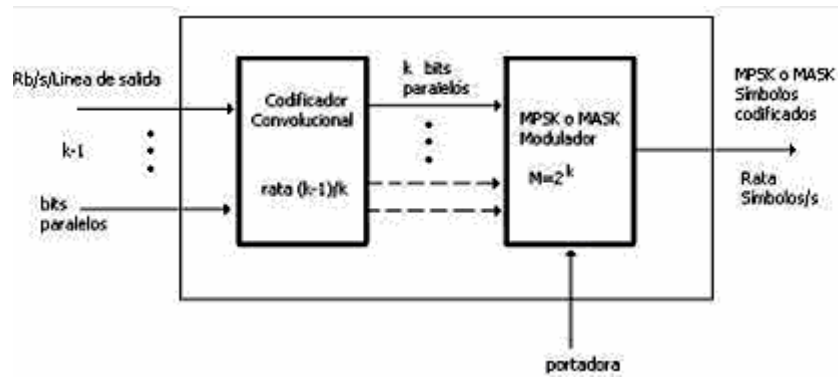


Figura. 8.8. Codificador / Modulador Genérico TCM

Con sistemas más complejos la ganancia de codificación puede ser de 6 dB o más, esta ganancia puede ser obtenida sin la expansión del ancho de banda o reducción de la tasa de información efectiva, como lo requieren los sistemas de codificación tradicionales.

### 8.2.2.1. Particionamiento de Conjuntos

El punto clave en el particionamiento de una constelación es hallar subconjuntos de la constelación que son similares y los puntos dentro de cada partición que estén máximamente separados. Comenzando con la constelación original, se la particiona en dos subconjuntos que son congruentes (complementarios) y los puntos dentro de cada partición están máximamente separados. Luego se aplica el mismo principio a cada partición y se continúa. El punto en el cual se para el particionamiento depende del código que se utilice.

La figura 8.9. muestra un ejemplo de particionamiento, que comienza con una constelación 8PSK con puntos de señal sobre un círculo de radio  $e_s$ . La distancia mínima en la constelación será:

$$d_0 = \sqrt{(2 - \sqrt{2})} e_s$$

Esta constelación se particiona en dos,  $\mathbf{B}_0$  y  $\mathbf{B}_1$ . es posible notar que  $\mathbf{B}_0$  y  $\mathbf{B}_1$  son complementarias. Existen varias formas en las que la constelación original 8PSK puede particionarse en dos subconjuntos complementarios, pero  $\mathbf{B}_0$  y  $\mathbf{B}_1$  proveen la partición de máxima distancia; la cual se deduce como:

$$d_1 = \sqrt{2e_s}$$

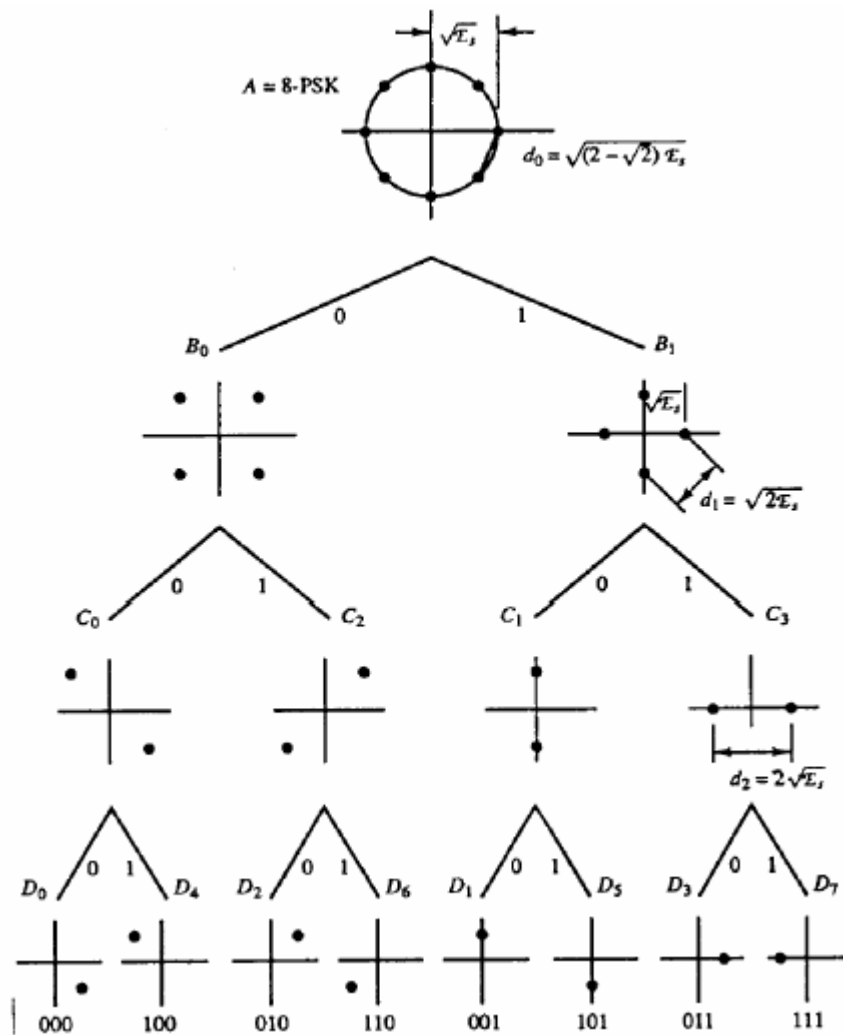


Figura. 8.9. Particionamiento de Conjuntos para 8PSK

Es posible continuar con el proceso y particionar  $B_0$  y  $B_1$  para obtener  $C_0, C_1, C_2$  y  $C_3$ . La partición tiene una distancia interna que ha disminuido a

$$d_2 = 2\sqrt{e_s}$$

Se puede continuar así hasta obtener el último nivel con hasta obtener los subconjunto  $D_0$  a  $D_7$ . A continuación se muestra otro ejemplo de particionamiento para 16QAM.



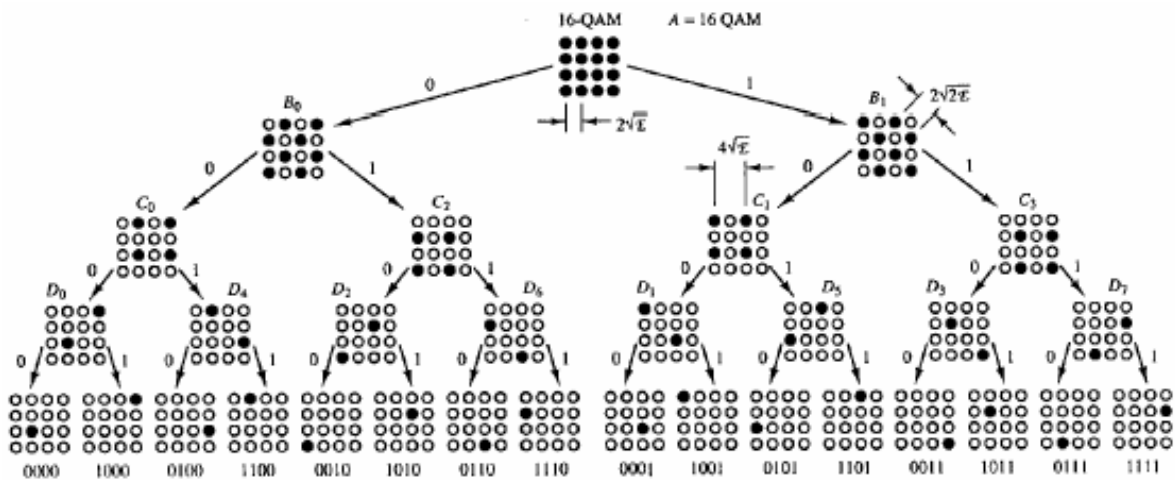


Figura. 8.10. Particionamiento de Conjuntos para 16QAM

Para este caso se siguen las mismas reglas de particionamiento usadas para PSK.

### 8.2.2.2. Modulación Codificada

El diagrama de bloques se muestra en la figura 8.11. Un bloque de  $k$  bits de entrada se subdivide en dos bloques de longitudes  $k_1$  y  $k_2$  respectivamente. Los primeros  $k_1$  bits se aplican a un codificador binario  $(N_1, k_1)$ . La salida del codificador consiste en  $n_1$  bits. Esos bits se utilizan para elegir una de las  $2^{n_1}$  particiones de la constelación. Esto significa que la constelación ha sido particionada en  $2^{n_1}$  subconjuntos. Después que la constelación ha sido elegida, los  $k_2$  bits remanentes se utilizan para elegir uno de los puntos de la constelación elegida. Esto significa que existen  $2^{k_2}$  puntos en cada partición. Esto determina una regla para obtener el tamaño de la constelación requerido y cuantos pasos de particionamiento deben seguirse sobre la constelación.

Underboeck mostró que eligiendo  $n_1 = k_1 + 1$  y  $k_2 = 1$  y utilizando códigos convolucionales simples es posible diseñar esquemas de modulación codificados que logran una ganancia total de codificación entre 3 y 6 dB. Uno de estos esquemas se muestra en la figura 8.12. en este esquema de codificación  $k_1 = 1$ ,  $n_1 = 2$  y  $k_2 = 1$ . La constelación contiene  $2^{n_1+k_2} = 8$  puntos, que están particionados en  $2^{n_1} = 4$  subconjuntos

cada uno de de los cuales tiene  $2^{k_2} = 2$  puntos. La constelación elegida es 8PSK y se particiona como se mostró en la figura 8.9. El código convolucional empleado tiene una tasa  $k_1/n_1 = 1/2$ ; la restricción de longitud de este código es un parámetro de diseño y puede elegirse para proveer la ganancia de codificación deseada, es decir, se obtiene mayor ganancia incrementando la complejidad de codificador y decodificador. En este ejemplo la restricción es tres. El diagrama de Trellis (de una etapa) para este código se muestra en la figura 8.12.

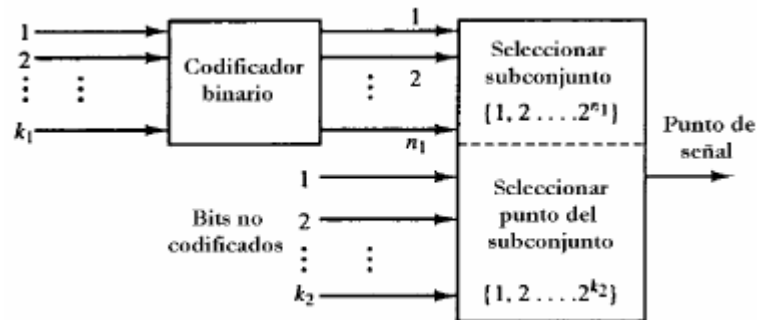


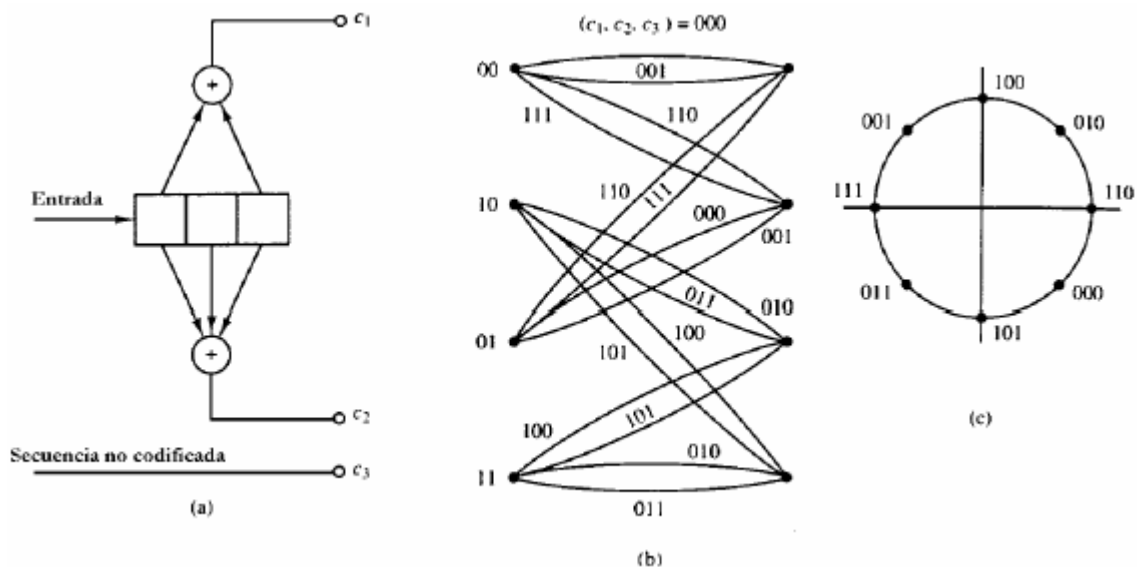
Figura. 8.11. Diagrama de Bloques de un Sistema de Modulación Codificada

El diagrama de Trellis mostrado es el correspondiente a un código convolucional ordinario. La diferencia principal es que aquí se tienen dos caminos que conectan dos estados. La razón de esto es la existencia de un bit extra,  $k_2 = 1$ , que selecciona un punto en cada partición. En realidad, los dos primeros caminos paralelos que conectan dos estados corresponden a una partición, y cualquier camino simple corresponde a un punto en la partición.

Para determinar el mapeo óptimo entre la transición entre los estados de un código convolucional y las particiones, se han determinado las siguientes reglas a través de simulaciones computacionales y razonamiento heurístico.

- Las transiciones en paralelo (cuando ocurren) corresponden a puntos de señal en una única partición en la etapa siguiente del particionamiento. Estos puntos están separados por una distancia Euclidiana máxima de  $d_2 = 2\sqrt{e_s}$ .

- Las transiciones que se originan y vuelven a cualquier estado se asignan en la siguiente etapa del particionamiento que tiene una partición pariente simple en la etapa precedente. La distancia máxima en este caso es  $d_1 = \sqrt{2e_s}$ .
- Los puntos de señal deberían ocurrir con igual frecuencia.



**Figura. 8.12. Esquema TCM Simple: a) Codificador, b) Trellis de Cuatro Estados, c) Mapeo de los Bits Codificados en los puntos de señal.**

Para ver como se desempeña el esquema de modulación codificada en Trellis de la figura 8.12., es posible hallar la distancia Euclidiana mínima entre dos caminos que se originan en un nodo y se juntan en otro nodo. Esta distancia, conocida como **distancia Euclidiana libre**,  $d_{free}$ , es una característica importante de un esquema TCM. Un candidato obvio para  $d_{free}$  es la distancia Euclidiana entre dos transiciones en paralelo.

En lugar de una Trellis de cuatro etapas, una Trellis con mayor número de etapas produce mayores ganancias de codificación. A través de simulaciones computacionales extensivas Ungerboeck indicó que con 8, 16, 32, 128 y 256 etapas, pueden lograrse ganancias de codificación en el rango de 3.6 dB a 5.75 dB. El codificador de una Trellis de ocho estados se muestra en la figura 8.13.

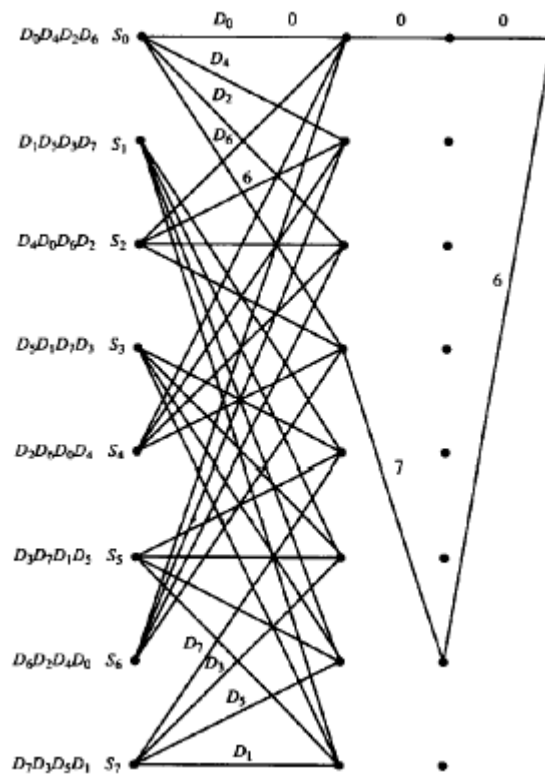


Figura. 8.13. Codificador Trellis de Ocho Estados para modulación Codificada 8PSK

### 8.2.2.3. Decodificación de Códigos TCM

La decodificación de la modulación codificada de Trellis se realiza en dos etapas. Dado que cada transición en la Trellis corresponde a una partición del conjunto de señales y cada partición generalmente corresponde a un número de puntos de señal, el primer paso es hallar los puntos más probables en cada partición. Esto se logra hallando en cada partición el punto más próximo en distancia Euclidiana al punto recibido. Este primer paso en la decodificación de un esquema TCM se denomina **decodificación de subconjuntos**. Después de este paso, a cada transición en la Trellis corresponde un único punto (el más probable) y una única distancia Euclidiana (distancia entre el punto recibido y el punto más probable).

El segundo paso del procedimiento de decodificación es utilizar esta distancia Euclidiana para hallar un camino a través de la Trellis cuya distancia Euclidiana total de la secuencia recibida sea mínima, esto se logra aplicando el algoritmo de **Viterbi**.

La modulación codificada en Trellis, TCM, se utiliza ampliamente en módems de alta velocidad, con constelaciones QAM, generalmente de  $M=16$ . La ganancia de codificación introducida por TCM ha hecho posible incrementar la velocidad de transmisión por lo menos en un factor de dos.

### 8.2.3. OFDM – ORTHOGONAL FREQUENCY-DIVISION MULTIPLEXING

La Multiplexación por División de Frecuencia Ortogonal es un método de modulación digital en el cual cada señal se separa en varios canales de banda angosta a diferentes frecuencias. La tecnología se concibió inicialmente en los años 60 y 70 durante investigaciones para minimizar la interferencia entre canales cercanos uno al otro en frecuencia.

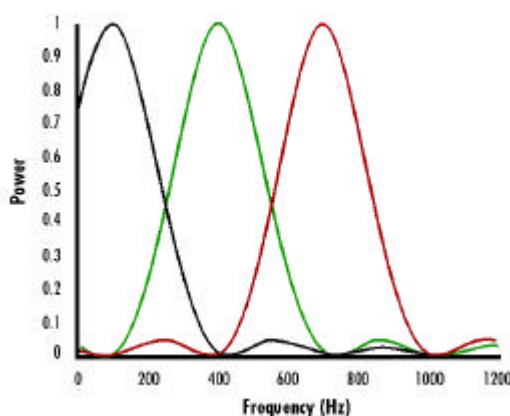


Figura. 8.14. Esquema de Bandas de Transmisión OFDM

En algunos aspectos, el OFDM es similar a la multiplexación por división de frecuencia tradicional (FDM), con la diferencia básica siendo la forma en que las señales se modulan y demodulan. La prioridad se le da a la minimización de interferencia o cruce entre los canales y símbolos en flujo de datos. Se le da menos importancia al perfeccionamiento de los canales individuales. La tecnología se presta para el envío de señales de televisión digital, y también se está considerando como una forma de obtener transmisión de datos a alta velocidad sobre las líneas convencionales de teléfono.

La OFDM es una modulación que consiste en enviar la información modulando en QAM o en PSK un conjunto de portadoras de diferente frecuencia. Normalmente se realiza

la modulación OFDM tras pasar la señal por un codificador de canal con el objetivo de corregir los errores producidos en la transmisión, entonces esta modulación se denomina **COFDM**, del inglés **Coded OFDM**. Debido al problema técnico que supone la generación y la detección en tiempo continuo de los cientos, o incluso miles, de portadoras equiespaciadas que forman una modulación OFDM, los procesos de modulación y demodulación se realizan en tiempo discreto mediante la IDFT y la DFT respectivamente.

Las transformadas digitales de Fourier se utilizan desde hace décadas para la modulación multiportadora. Los sistemas actuales basados en las transformadas de Fourier son los sistemas OFDM (Orthogonal Frequency Division Multiplex) y DMT (Discrete MultiTone). Existe también otro sistema de modulación multiportadora que emplea otro tipo de transformada digital conocida como digital wavelet transform. Estamos hablando del sistema DWMT (Discrete Wavelet MultiTone).

### 8.2.3.1. Características de OFDM

La modulación OFDM es muy robusta frente al multitrayecto, que es muy habitual en los canales de radiodifusión, frente al desvanecimiento debido a las condiciones meteorológicas y frente a las interferencias de RF. Debido a las características de esta modulación, las distintas señales con distintos retardos y amplitudes que llegan al receptor contribuyen positivamente a la recepción, por lo que existe la posibilidad de crear redes de radiodifusión de frecuencia única sin que existan problemas de interferencia.

Un problema común debido a las altas velocidades en las comunicaciones es la interferencia intersimbólica (ISI). ISI ocurre cuando una transmisión interfiere con sí misma y el receptor no puede descifrar la transmisión correctamente. Por ejemplo, en un sistema de comunicación sin hilos, la transmisión se envía en todas las direcciones, la señal reflejada de objetos grandes como montañas o edificios, llegan al receptor, que considera más de una copia de la señal.

Durante los últimos años, la aceptación del OFDM como tecnología de base para el 802.16a que es un estándar de IEEE para redes de área metropolitana inalámbrica que puede proveer extensión inalámbrica para acceso de última milla de banda ancha en

instalaciones de cable y DSL. El mismo cubre el rango de frecuencias de 2 a 11 GHz y alcanza hasta 50 kilómetros lineales, brindando conectividad de banda ancha inalámbrica sin necesidad de que exista una línea directa de visión a la estación de base. La velocidad de transmisión de datos puede llegar a 70 Mbps. Una estación de base típica puede albergar hasta seis sectores. La calidad de servicio está integrada dentro del MAC, permitiendo la diferenciación de los niveles de servicio.

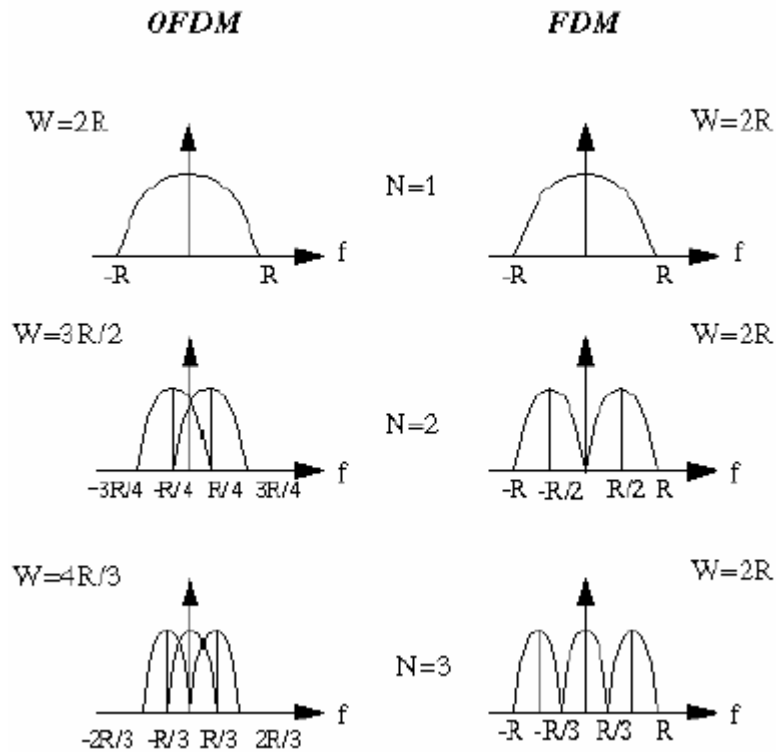


Figura. 8.15. Modulación con Portadoras Ortogonales y Técnica Multiportadora Convencional,

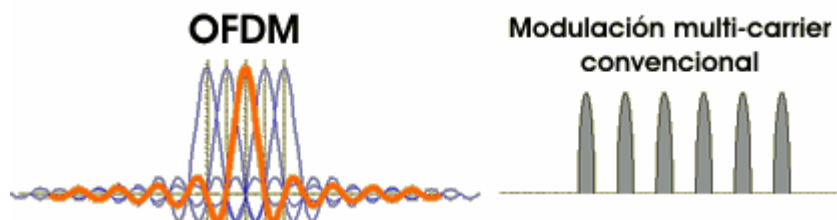


Figura. 8.16. OFDM vs. FDM

### 8.2.3.1.1. Diagrama de Bloques General.

La entrada a cualquier fase del diseño serán señales digitales. Esto puede extenderse de datos binarios simples a las señales muestreadas de sonidos. La salida del transmisor será la representación de OFDM de la fuente.



Figura. 8.17. Diagrama de Bloques General de Transmisión con OFDM

A continuación se muestra cómo una señal de entrada de alta frecuencia se multiplexa sobre muchos portadores ortogonales en una frecuencia más baja. Las señales de una frecuencia más baja tienen tiempos más largos del símbolo que reduce grandemente los problemas de ISI en la comunicación digital de alta velocidad.

Una señal OFDM consiste en un número de subportadoras QAM no filtradas, cuyo espectro decae lentamente ( $\text{SINC}(x)$ ). El espectro decae en un principio rápidamente por efecto de la cancelación de los lóbulos laterales. A mayor número de subportadoras, mayor es la cancelación y más rápidamente decae el espectro.

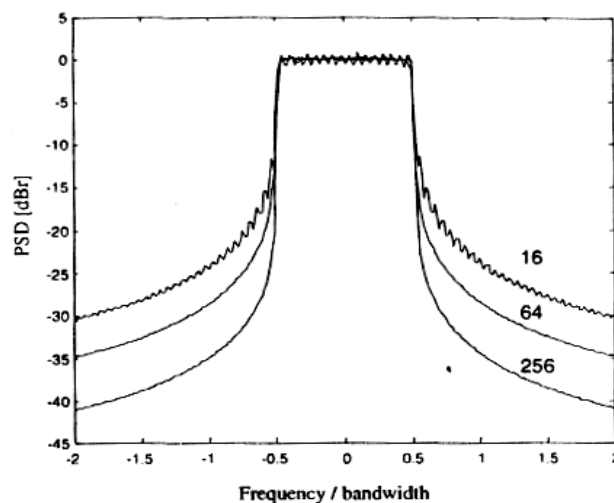


Figura. 8.18. Aplicación de Windowing (Coseno Levantado) para OFDM



Windowing sobre un símbolo OFDM hace que la amplitud se haga cero en los bordes del símbolo. La ventana más utilizada es el Coseno Levantado, que se expresa como:

$$w(t) = \begin{cases} 0.5 + 0.5 \cos\left(\frac{p + tp}{bT_s}\right), & 0 \leq t \leq bT_s \\ 1.0 & , bT_s \leq t \leq T_s \\ 0.5 + 0.5 \cos\left(\frac{p - tp}{bT_s}\right), & T_s \leq t \leq (1+b)T_s \end{cases}$$

Aquí,  $T_s$  es el intervalo del símbolo, que es más corto que la duración total del símbolo pues se permite que subportadoras estén parcialmente montadas en la región de roll-off.

IFFT modula valores QAM en un conjunto de subportadoras para la Tx. FFT demodula las subportadoras en la Rx. Una característica interesante de IFFT/FFT es que IFFT es casi idéntico que FFT, esto hace posible la utilización del mismo HW, con la salvedad de que no es simultáneo.

El procesamiento del módem OFDM es significativamente menor que el módem simple-carrier.

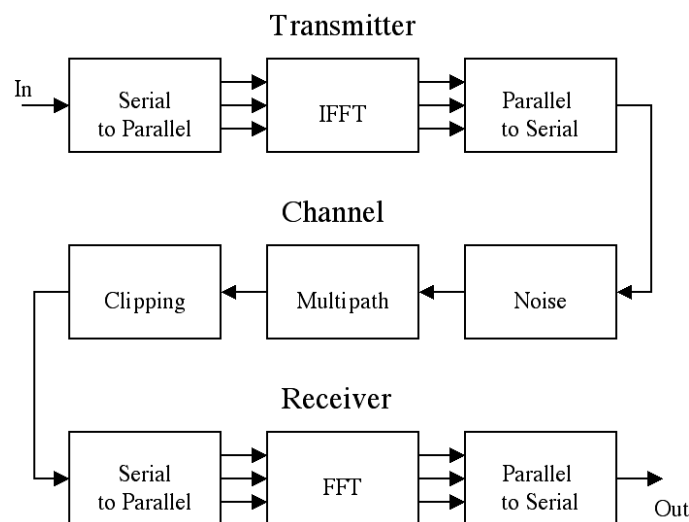


Figura. 8.19. Etapas de Transmisión, Canal y Recepción para OFDM

### 8.2.3.1.2. Transmisor.

El transmisor primero convierte los datos de entrada de forma serial a los sistemas paralelos. Cada sistema de datos contiene una información para cada frecuencia portadora.

Entonces, los datos paralelos se modulan a las frecuencias portadora ortogonal. El IFFT convierte los datos paralelos en formas de onda del dominio de tiempo. Finalmente, estas formas de onda se combinan para crear una sola señal del dominio de tiempo para la transmisión.

Cada dato contiene un símbolo  $S_i$ , para cada subcarrier (subportadora). Por ejemplo, un sistema de cuatro datos sería  $[S_0 S_1 S_2 S_3]$ . Antes de realizar la Transformada inversa de Fourier (IFFT), este juego de datos de ejemplo es colocado en el eje horizontal en el dominio de frecuencia como se muestra en la figura.

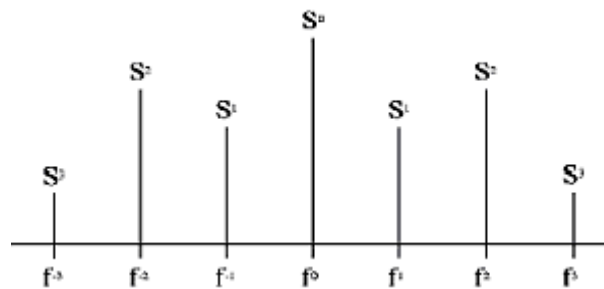
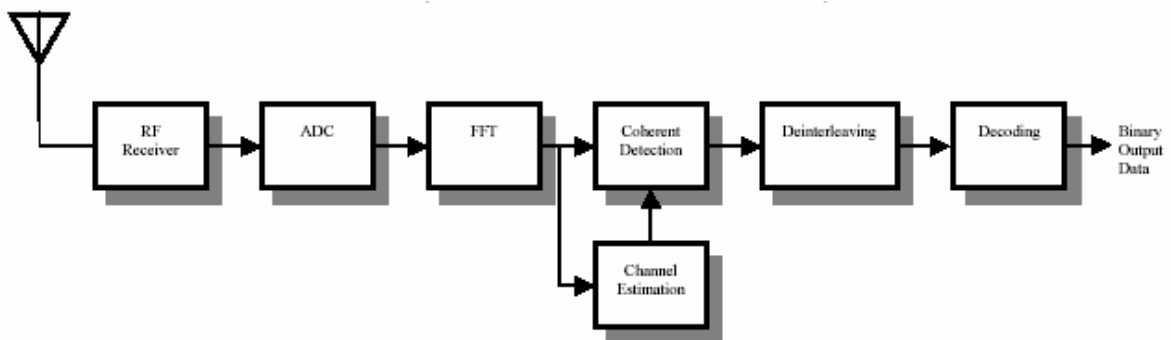


Figura. 8.20. Distribución de Símbolos en el Dominio del Tiempo

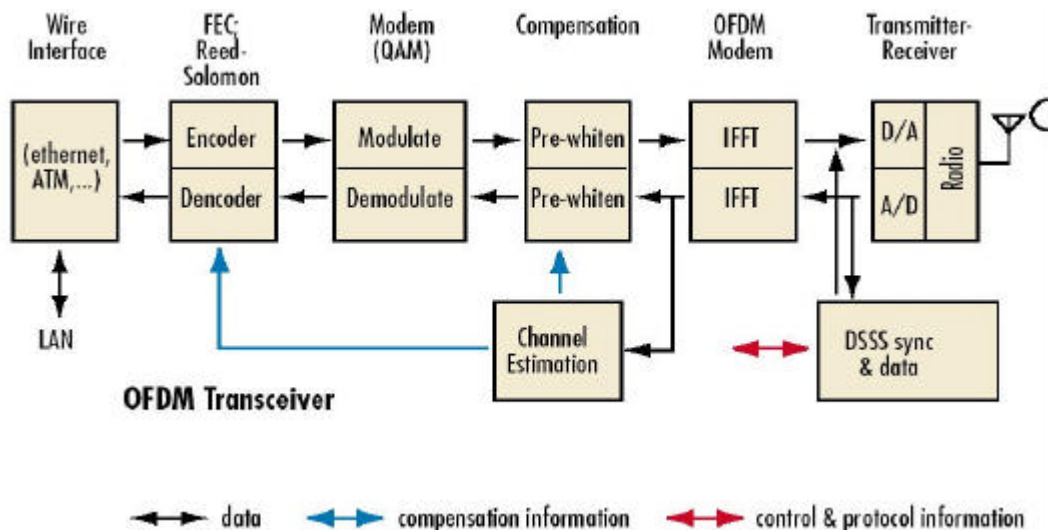
Este arreglo simétrico sobre el eje vertical es necesario para usar el IFFT. La transformada de Fourier convierte el dato al dominio de la frecuencia en muestras, correspondientes en el dominio de tiempo. Específicamente, la IFFT es útil para OFDM porque genera muestras de una forma de onda con los componentes de la frecuencia que satisfacen condiciones del ortogonalidad. Entonces, el bloque serial paralelo crea la señal de OFDM secuencialmente, haciendo salir las muestras del dominio de tiempo.

**8.2.3.1.3. Receptor.**

El receptor realiza básicamente lo contrario del transmisor, primero separando los datos en corrientes paralelas. Entonces, el FFT convierte estas secuencias de datos paralelas en datos del dominio de la frecuencia. Los datos están disponibles ahora en forma modulada en los portadores ortogonal. La demodulación convierte esta información de nuevo a la banda base. Finalmente, estos datos paralelos se convierten nuevamente en forma serial para recuperar la señal original.



**Figura. 8.21. Receptor OFDM con Detección Coherente**



**Figura. 8.22. Tranceptor Usado en HiperLAN OFDM**

### 8.2.3.2. Sistemas que Utilizan OFDM

Entre los sistemas que usan la modulación OFDM destacan:

- La televisión digital terrestre DVT-T.
- La radio digital DAB.
- La radio digital de baja frecuencia DRM.
- El protocolo de enlace ADSL.
- El protocolo de red de área local IEEE 802.11a/g, también conocido como Wireless LAN.
- El sistema de transmisión inalámbrica de datos WiMAX.

### 8.2.3.3. W-OFDM - Wideband OFDM

Es un esquema de transmisión que codifica la información en múltiples radio frecuencias simultáneamente. Dando como resultado, mayor seguridad y mayor velocidad. Esto lo convierte en el esquema más eficiente en el uso del ancho de banda en la industria. W-OFDM es la base del estándar IEEE 802.11a que a su vez es la base para el estándar propuesto IEEE 802.16. Sus características son:

- Ancho de Banda: 30Mbps
- Altamente inmune a interferencias
- Punto-a-Punto, 8 a 10Km
- Multi-Punto, 3 a 5Km
- Próximamente: 45Mbps, 90Mbps, 155Mbps
- Patente: Wi-Lan
  - USA 5,282,222
  - CANADÁ 2,064,975

El reto de todos los días para la industria es lograr mayores velocidades de transmisión en las redes de datos/Internet. Una propuesta es el esquema de transmisión W-OFDM

(Wide-band Orthogonal Frequency Division Multiplexing), este método como otros codifica los datos dentro de una señal de radio frecuencia (RF). Transmisiones convencionales como AM/FM envían solamente una señal a la vez sobre una frecuencia de radio, mientras que OFDM envía una señal de alta velocidad concurrentemente sobre frecuencias diferentes. Esto nos permite hacer un uso muy eficiente del ancho de banda y tener una comunicación robusta al enfrentar ruido y reflejos de señales.

La tecnología OFDM parte una señal de alta velocidad en decenas o centenas de señales de menor velocidad, que son transmitidas en paralelo. Esto crea un sistema altamente tolerante al ruido, al mismo tiempo es muy eficiente en el uso del ancho de banda y por lo tanto permite una amplia cobertura de área punto a punto y multipunto.

Actualmente existen equipos con la capacidad de transmitir desde 1.5Mbps hasta 30Mbps en 25MHz de ancho de banda y pronto se estarán produciendo equipos que superaran velocidades de 100Mbps. Adicionalmente a la velocidad, se cuenta con opciones de seguridad que hacen virtualmente imposible descifrar la señal que se transmite.

Los equipos con tecnología OFDM ayudan a las empresas a evitar los altos costos de instalación de cable, a eliminar rentas mensuales o cargos por licenciamiento. Son la solución ideal en distancias moderadas para redes de información punto a punto, multipunto, acceso de alta velocidad a Internet, extensiones de LAN/WAN, Videoconferencia, Telefonía, Telemetría, Control, Etc.

Los sistemas W-OFDM incorporan además estimación de canal, prefijos cíclicos y códigos Reed-Solomon de corrección de errores.

#### **8.2.3.4. Ventajas de Usar OFDM**

- Hace uso eficiente del espectro permitiendo el solapamiento.
- Al dividir al canal en subcanales estrechos de desvanecimiento plano, OFDM es más resistente al desvanecimiento de frecuencias selectivo como lo son los sistemas mono portadores.

- Elimina la Interferencia InterSímbolos y la IFI (InterFrame Interferente) por medio del uso de prefijos cíclicos.
- Usando la codificación de canal adecuada y entrelazado, se puede recuperar símbolos perdidos debido al desvanecimiento de frecuencias selectivo del canal.
- La ecualización del canal es más simple que la utilización de técnicas de ecualización adaptativa de los sistemas mono portadores.
- Es posible lograr una decodificación de máxima probabilidad con complejidad razonable, OFDM es computacionalmente eficiente debido al uso de técnicas FFT para implementar las funciones de modulación y demodulación.
- En conjunción con la modulación diferencial no hay necesidad de implementar un estimador de canal.
- Proporciona buena protección contra la interferencia co-canal y el ruido impulsivo.

## 8.3. PRÁCTICAS CON MATLAB

### 8.3.1. ORTHOGONAL FREQUENCY-DIVISION MULTIPLEXING

#### 8.3.1.1. Objetivos

- Comprender de mejor manera la forma como se realiza la Multiplexación por División de Frecuencia Ortogonal.
- Apreciar a través del modelamiento en Simulink (MATLAB) la estructura de un sistema de comunicación que utiliza OFDM.
- Conocer algunos de los objetos disponibles en Simulink para efectuar simulaciones en MATLAB de forma gráfica, evitando realizar toda la codificación en scripts.
- Observar la respuesta en frecuencia del modelo de OFDM desarrollado, así como su diagrama de constelación para diferentes tipos de modulación pasabanda: BPSK, QPSK, 16-QAM y 64-QAM.

#### 8.3.1.1.1. Descripción de Bloques.

El modelo de Simulink presenta un sistema simplificado de transmisión OFDM que introduce la aplicación de elementos IFFT / FFT en el proceso de modulación / demodulación de datos de acuerdo con el estándar IEEE802.11a.

El sistema de comunicaciones por tanto presenta tres etapas básicas:

- Transmisor.
- Canal.
- Receptor.

Dentro de cada una de estas etapas se encuentran diferentes procesos y subprocesos que se modelan mediante Simulink a través de la utilización de los objetos disponibles en las

diferentes librerías. En el sistema también se añaden objetos que permiten realizar mediciones, observaciones de espectro y otro tipo de instrumentación virtual.

En este caso particular se usa objetos que permitan observar el espectro de la señal en diferentes puntos del sistema llamados **Spectrum Scope** y que se encuentran en la librería del Blockset de Procesamiento de Señales; y, objetos **Discrete Time Scatter Plot Scope** que permitan observar la constelación de la señal transmitida y se encuentran en la librería del Blockset de Comunicaciones.

Para realizar los modelos en MATLAB, se hace uso de enmascaramientos, los mismos que permiten esconder varios bloques u objetos bajo uno solo, simplificando la vista general del modelo y haciéndolo modular para sus modificaciones, ya que de ponerse todos los objetos en una sola ventana de modelo, la estructura se vuelve muy compleja.

El modelo de esta simulación, el mismo que se puede llamar desde MATLAB entrando en el directorio **laboratorio** de este capítulo bajo el nombre de **simulación\_ofdm.mdl**, se encuentra bajo una máscara formando un subsistema. Cuando se lo ejecuta, lo único que se ve es un bloque como se muestra en la figura 8.23.



**Figura. 8.23. Vista del Modelo de Sistema de Comunicación con OFDM**

Para poder observar los bloques de procesos u objetos bajo la máscara, se hace clic derecho sobre el bloque de la figura 8.23 y se selecciona **Look Under Mask** y aparece otra ventana conteniendo al subsistema que se ejecuta cuando se corre este modelo.



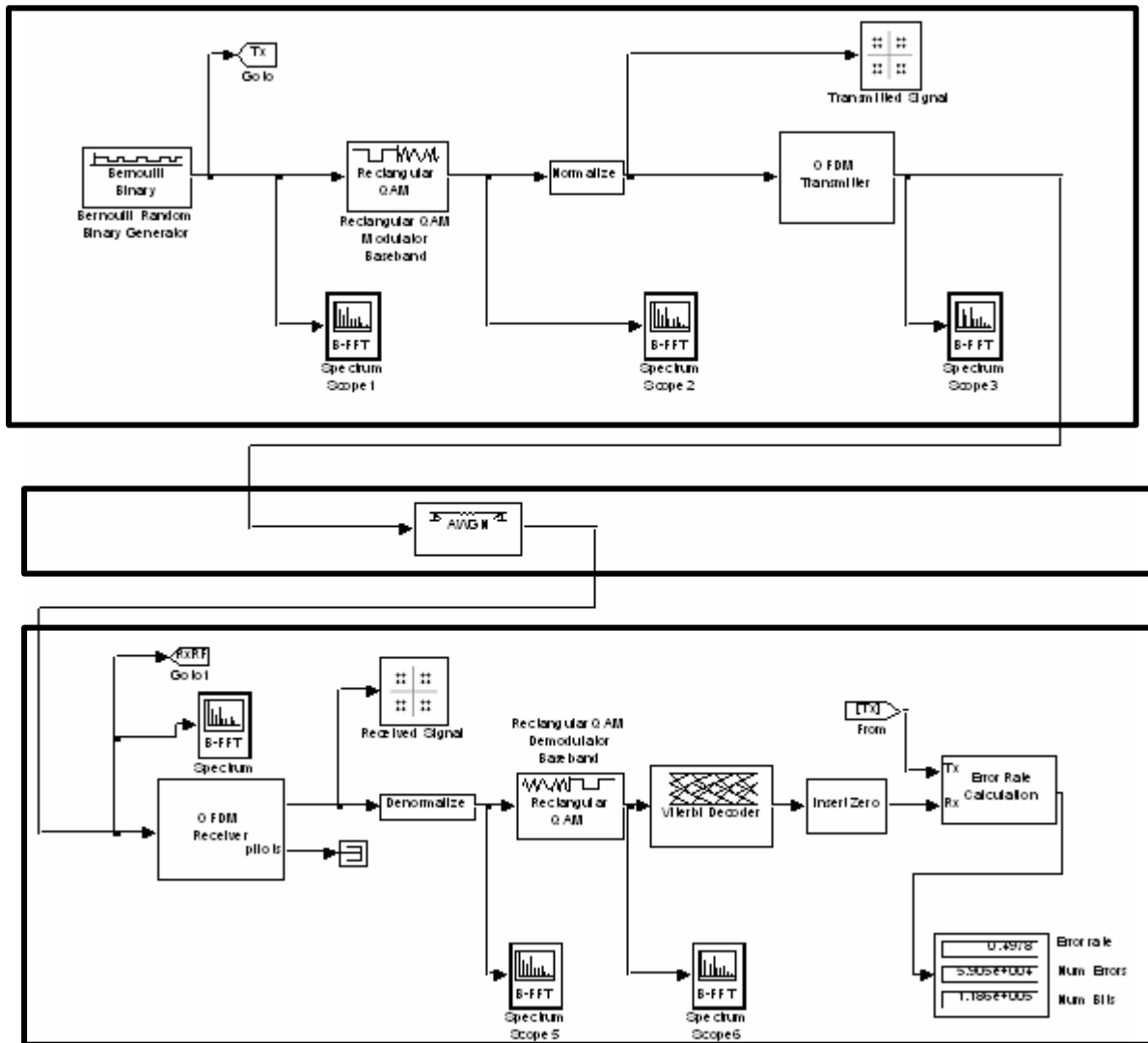


Figura. 8.24. Subsistema Modelo de Sistema de Comunicación con OFDM

Pero este no es el único enmascaramiento que se encuentra en este modelo, ya que una vez dentro del subsistema mostrado en la figura 8.24, que es el principal, se encuentran bajo máscaras el Transmisor y el Receptor de OFDM, un normalizador y un denormalizador.

Como se puede apreciar, en el modelo se incluyen otros objetos que permiten simular un sistema de comunicaciones. Es decir, no se trata solo del transmisor y receptor OFDM utilizando un canal AWGN, sino que se usa un generador de binario aleatorio (bandabase), un modulador QAM (pasabanda), un normalizador, un denormalizador, un demodulador QAM, decodificador Viterbi, calculador de tasa de error, pantallas para lectura de error y la

instrumentación antes detallada. Es decir, todos los elementos básicos necesarios que conforman un sistema de comunicaciones.

Para poder observar la forma como se encuentran conectados todos los subsistemas, podemos recurrir al **model browser** en donde podemos observar las conexiones por subsistemas y por enlaces. Por esto último se muestra a que otros objetos se recurren para realizar uno u otro proceso, mientras que por subsistemas solo se observan los que están en el modelo que se analiza.

**8.3.1.1.2. Transmisión.**

En la parte de transmisión se incluye la generación de la señal, la modulación pasabanda, la normalización de la misma y el transmisor OFDM, como se ve en la figura 8.21. A continuación se muestra lo que existe bajo la máscara del normalizador y del transmisor de OFDM.

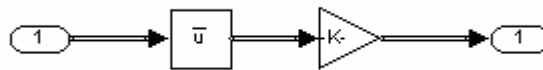


Figura. 8.25. Subsistema Bajo la Máscara del Normalizador

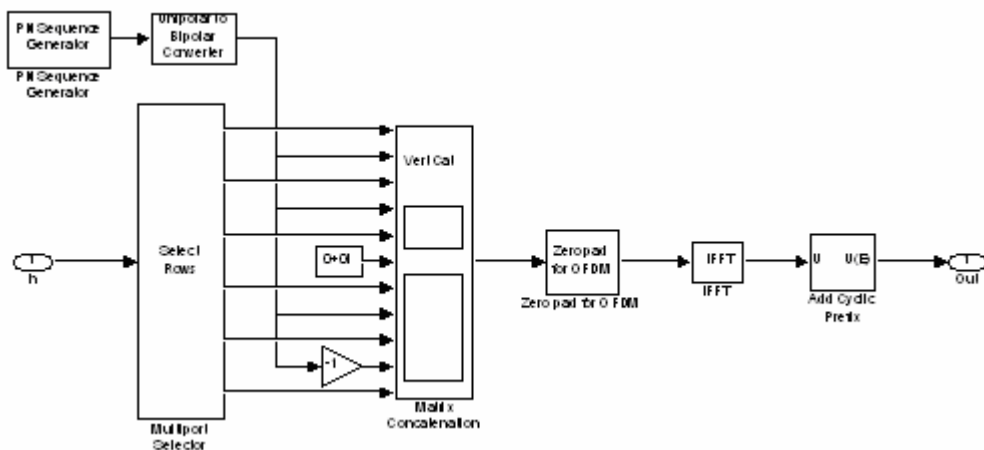


Figura. 8.26. Subsistema Bajo la Máscara del Transmisor OFDM

La figura 8.25 muestra como se forma el normalizador, el cual prepara la señal que alimenta el transmisor OFDM luego de la modulación pasabanda.

La figura 8.26 muestra el esquema del transmisor OFDM, en el que se muestra la generación de la secuencia pseudoaleatoria, un conversor de señal unipolar a bipolar, un selector de filas, un concatenador de matrices, un subsistema **Zero Pad** (cuya estructura se muestra en la figura 8.27), la IFFT y un selector que añade prefijos cíclicos.

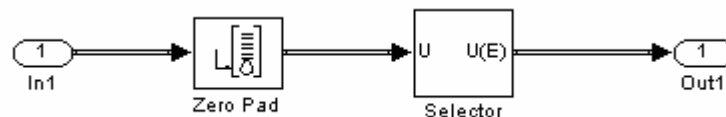


Figura. 8.27. Subsistema Bajo la Máscara de Zero Pad del Transmisor OFDM

### 8.3.1.1.3. Recepción.

En la etapa de recepción se incluyen el receptor de OFDM, cuya estructura de bajo de la máscara de la figura 8.24, se muestra en la figura 8.28. Este receptor trabaja de forma inversa al transmisor, encontrando en primera instancia un bloque que remueve los prefijos cíclicos, luego la FFT para demodular OFDM, conversor de tramas, removedor de **Zero Pad**, selector de filas y entrega los datos. La salida de este subsistema se muestra más adelante, así como las señales en el transmisor para poder compararlas y verificar la correcta transmisión del sistema.

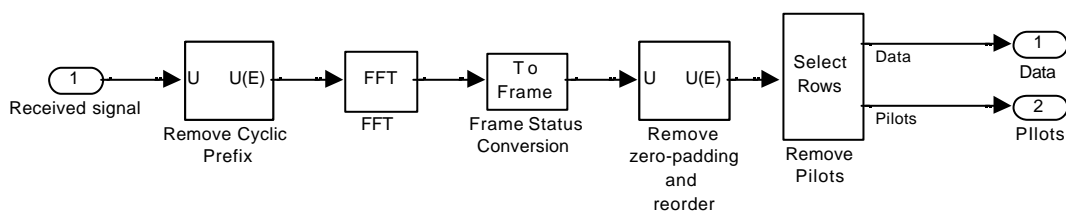


Figura. 8.28. Subsistema Bajo la Máscara del Receptor OFDM

También se encuentran en la recepción un denormalizador que revierte el proceso de normalización, un demodulador de QAM y un bloque que detecta y corrige errores de acuerdo al algoritmo de Viterbi.

Posterior a esto que constituye realmente la etapa de recepción, se encuentra un bloque de inserción de ceros, el cual permite ordenar lo que recibe a la entrada a una forma de vector para que se compare con la señal de entrada (generada) en una calculadora de tasa de error y sus respuestas se visualizan mediante las ventanas al final del modelo expuesto para este sistema de comunicaciones.



**Figura. 8.29. Subsistema Bajo la Máscara del Denormalizador**

Los objetos usados en este modelamiento existen en la librería de **Simulink 6 (R14)** que es la versión que viene incluida en la instalación de **MATLAB 7 /R14)**. Esta aclaración es importante por cuanto existe, al igual que con las funciones vistas en otras prácticas, la posibilidad de crear librerías propias para un determinado modelamiento, así como de objetos y subsistemas de acuerdo con la flexibilidad propia de MATLAB.

#### 8.3.1.1.4. Simulación.

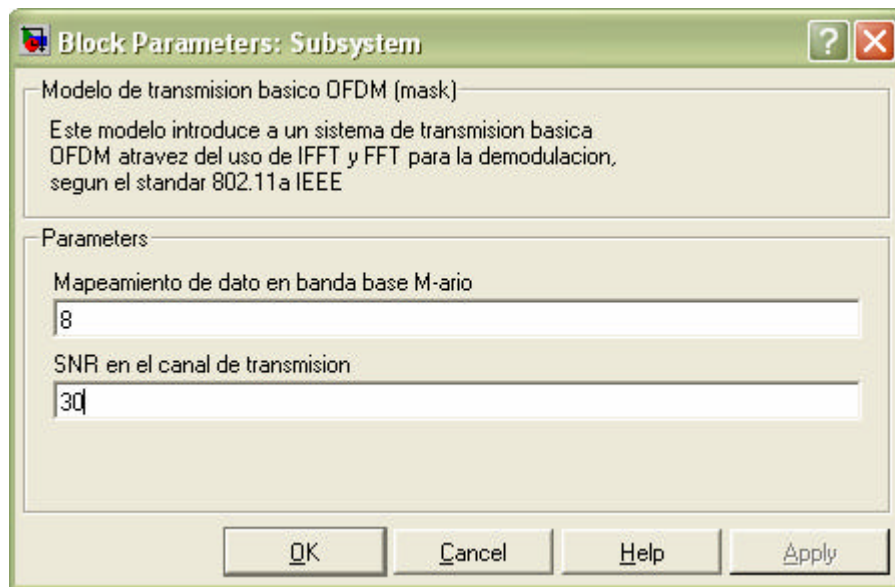
Para ejecutar la simulación se debe seleccionar **start** en el menú **Simulation** o hacer clic en el botón **play** de **Simulink**, con lo cual se empezará a correr el modelo diseñado hasta que se decida hacer una pausa o detenerlo completamente.

Previo a la ejecución se deben de configurar dos parámetros del modelamiento analizado. Estos parámetros son:

- La señalización M-aria a utilizar para el mapeo de los datos.
- La SNR del canal de transmisión.

Esto se debe realizar en la ventana que se muestra en la figura 8.30, a la cual se accede haciendo doble clic sobre el bloque del modelo. El bloque IFFT acepta solo señales complejas en su entrada, por ello la señal real M-aria debe ser convertida en una de tipo complejo, lo cual se logra a través del mapeo complejo de los datos.

De acuerdo con el estándar IEEE802.11a, al cual se refiere este modelamiento, se debe aplicar uno de los métodos de mapeo listados en la tabla 8.1 para la transmisión OFDM. Inicialmente los valores de mapeo y SNR son 8 y 30 respectivamente, por lo que deben ser cambiados para cumplir con las recomendaciones del estándar que se nombran a continuación.



**Figura. 8.30. Ventana de Configuración Principal de Parámetros.**

Así, el parámetro **M** es el que define el método a utilizar y no puede ser alterado mientras se ejecuta la simulación.

<b>M</b>	<b>Mapping Mode</b>
<b>2</b>	BPSK
<b>4</b>	QPSK
<b>16</b>	16-QAM
<b>64</b>	64-QAM

**Tabla. 8.1. Valores de M válidos para Mapeo de Acuerdo al estándar IEEE802.11a**

Para lograr transmisiones en OFDM con alto desempeño del error, la SNR debe ser mayor que 20 dB. Entre más alta sea el método de mapeo utilizado, más alta deberá ser la

SNR necesaria para obtener el mejor desempeño de error posible. Este parámetro si puede ser alterado mientras se este ejecutando la simulación.

Para cumplir con el estándar IEEE802.11a, se deben de cumplir con otros parámetros, los cuales son configurados en la sección **initial commands** en el editor de máscara del sistema. Estos parámetros se listan a continuación en la tabla 8.2

<i>%Subcarriers related parameters according to ch.17.3.2.3 of IEEE Std802.11a</i>
<i>%</i>
<b>Nsd=48;</b> <i>% Number of data subcarriers</i>
<b>Nsp= 4;</b> <i>% Number of pilot subcarriers</i>
<b>Nst =Nsd+Nsp;</b> <i>% Number of subcarriers, total (52)</i>
<i>%</i>
<i>% Parameters of names not following Std802.11a labeling</i>
<i>%</i>
<b>BW=20e6;</b> <i>%Cumulative signal bandwidth (20Msamples/s)</i>
<b>ChTSamp=1/BW;</b> <i>% The time duration between samples in the channel</i>
<b>SubchannelBW=BW/64;</b> <i>%Subchannels signal bandwidth</i>
<b>Tbit=1/SubchannelBW;</b> <i>%Subchannel signal sample time</i>

**Tabla. 8.2. Parámetros Establecidos como Comandos Iniciales del Sistema**

El estándar IEEE802.11a define un total de 52 subcanales sobre los cuales se deben transmitir los datos. Esto incluye 4 subcanales usados para señalización piloto de la transmisión.

#### **8.3.1.1.5. Resultados.**

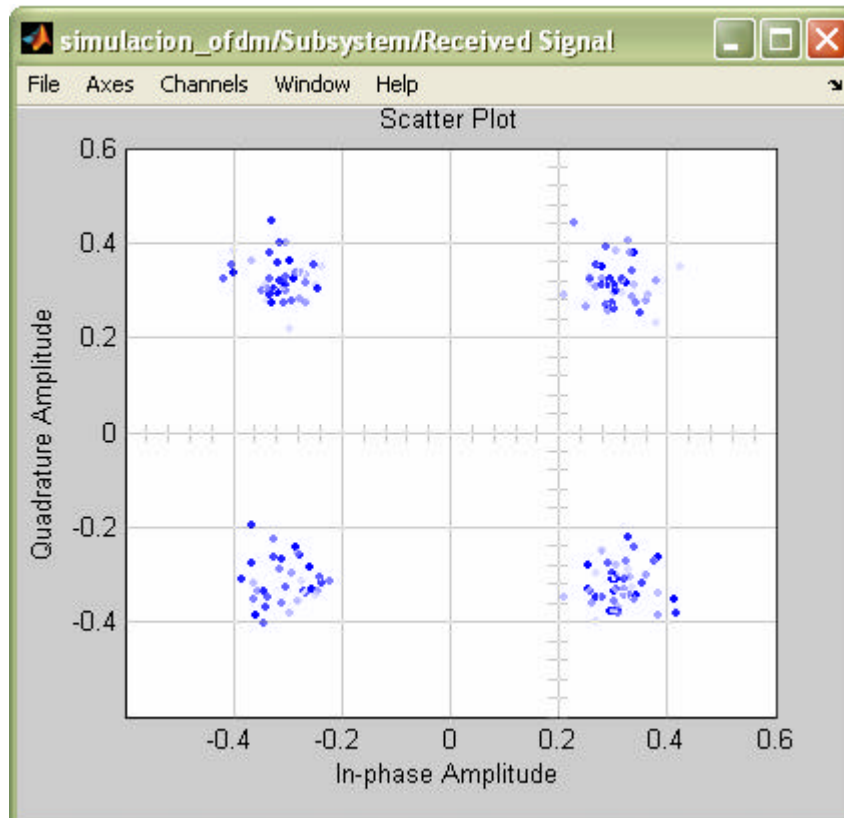
Las gráficas resultantes de la ejecución de este modelo permiten ver la evolución de la señal sobre el sistema de comunicación utilizando OFDM, así como la cantidad de datos y errores obtenidos al final de la cadena de transmisión.

Las figuras de los espectros se colocan de manera secuencial, de forma que se pueda observar la evolución de la señal, así, como comparar lo que sucede en etapas análogas, tal como han sido colocados en el modelo revisado en Simulink. De esta manera, las figuras corresponden a:

- a. Salida del Generador.
- b. Salida del Modulador.
- c. Salida del Transmisor OFDM.
- d. Salida del Canal AWGN / Entrada Receptor OFDM.
- e. Salida Denormalizador (luego del receptor OFDM).
- f. Salida del Demodulador.

Además se incluye el diagrama de constelación correspondiente al mapeo utilizado al inicio de la secuencia de gráficos y los indicadores con tasa de error, número de errores y número de datos transmitidos.

**1.  $M= 4$ ,  $SNR= 22$ .**



**Figura. 8.31. Constelación de Señal Recibida  $M= 4$ .**

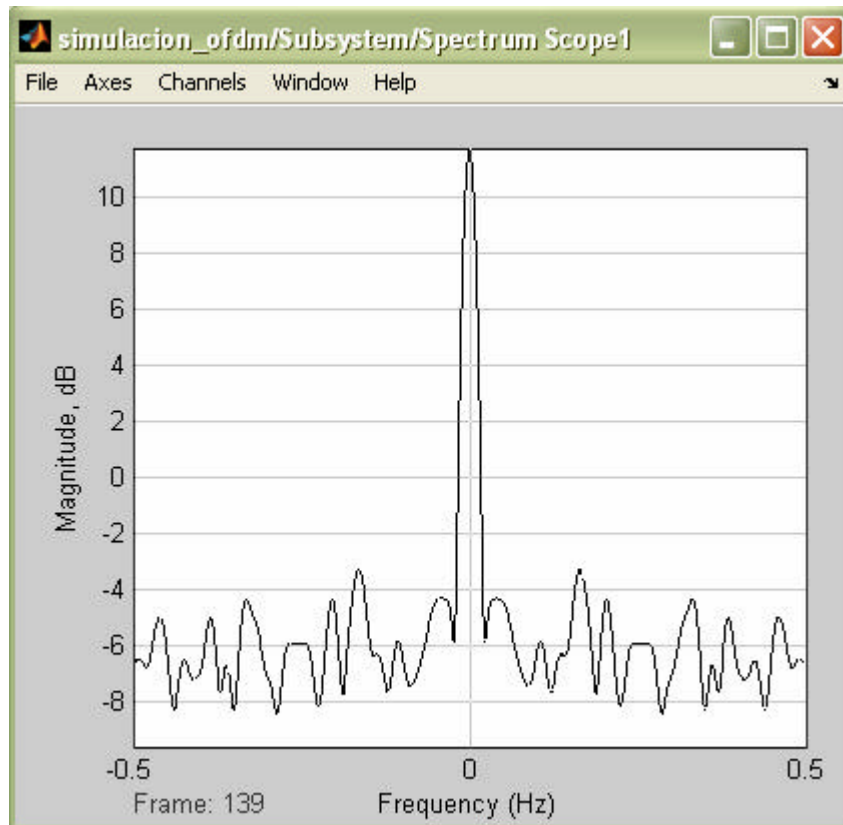


Figura 8.32. Salida del Generador de Datos  $M=4$ ,  $SNR=22$ .

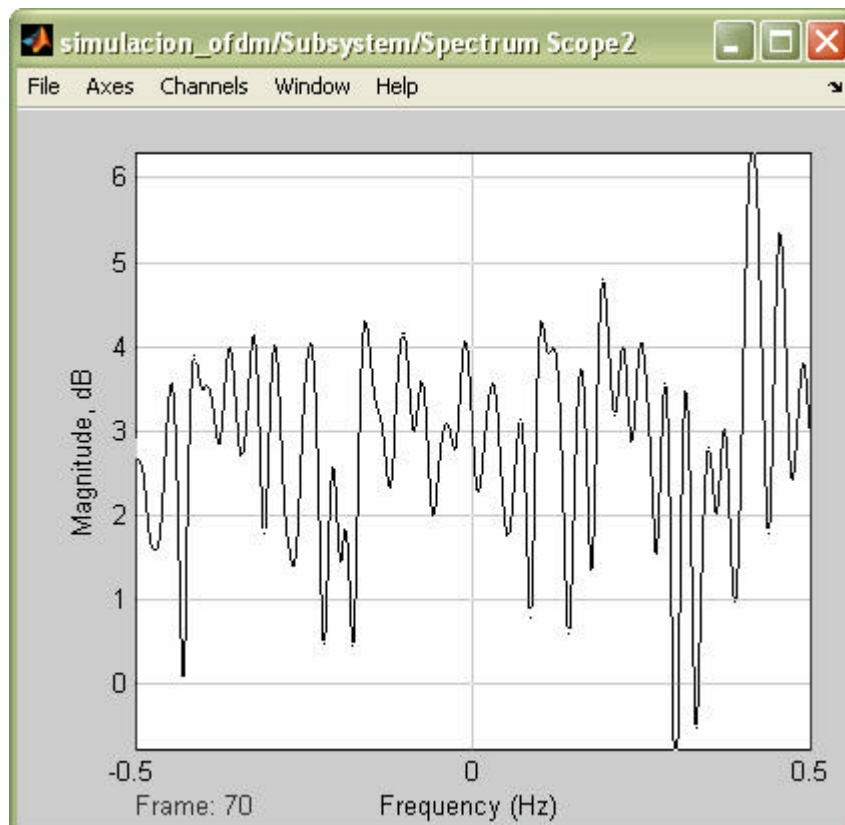


Figura 8.33. Salida del Modulador  $M=4$ ,  $SNR=22$ .



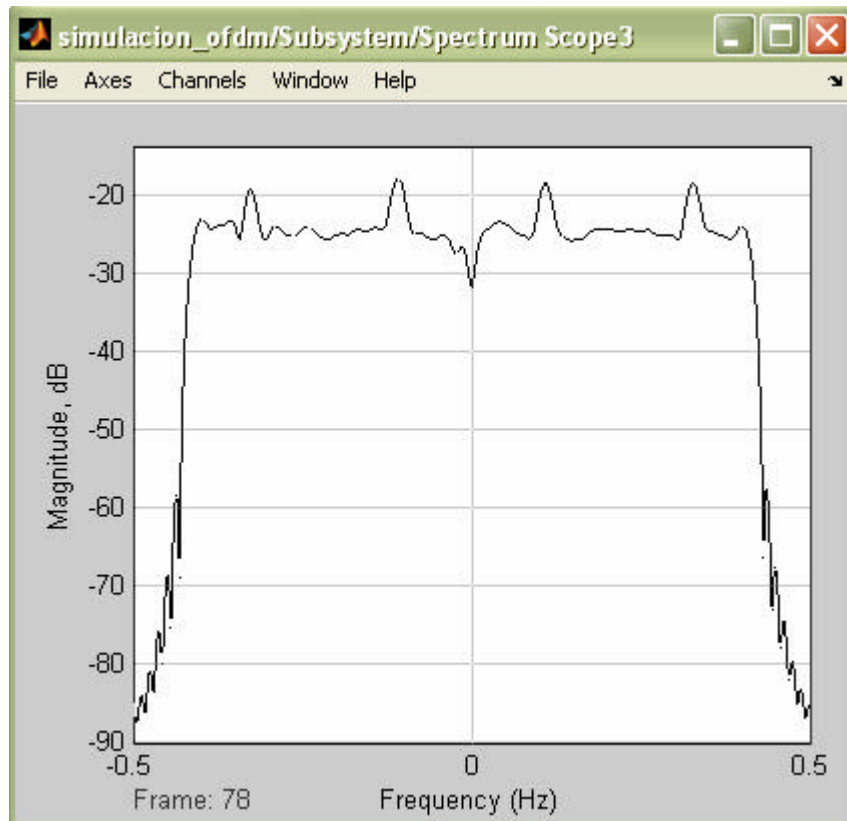


Figura. 8.34. Salida del Transmisor OFDM  $M=4$ ,  $SNR=22$ .

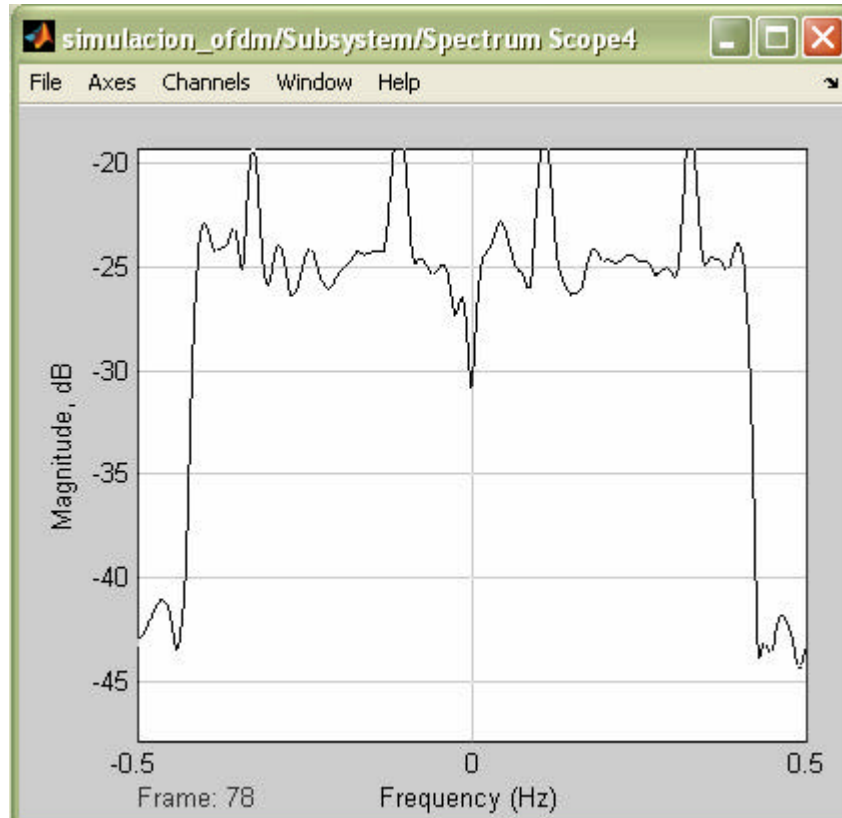


Figura. 8.35. Salida del Canal AWGN  $M=4$ ,  $SNR=22$ .

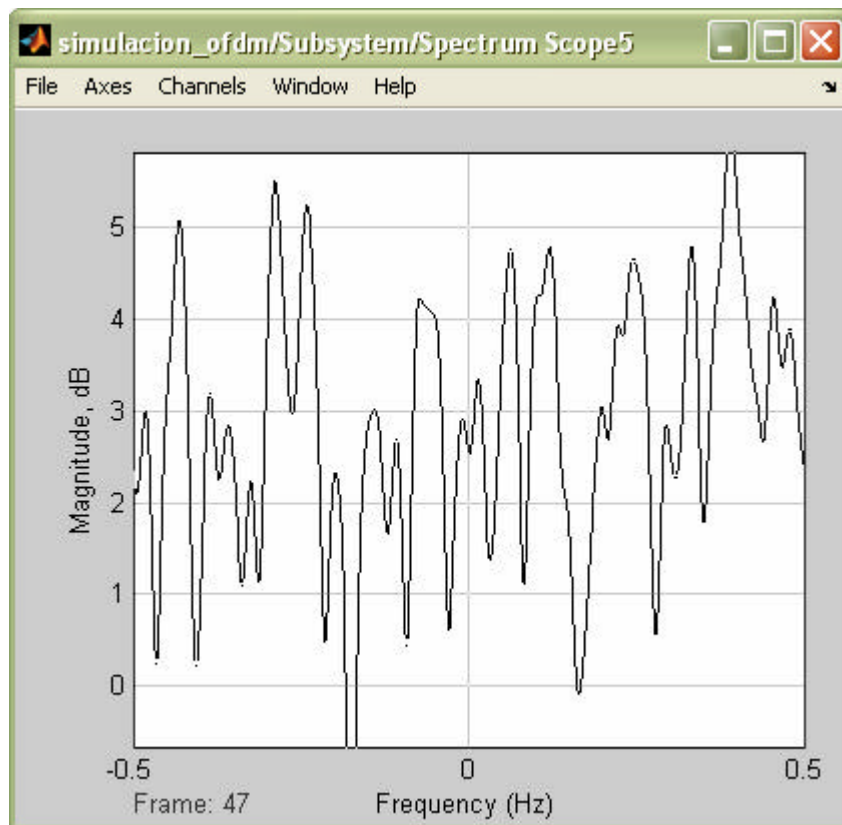


Figura. 8.36. Salida denormalizada del Receptor OFDM  $M=4$ ,  $SNR=22$ .

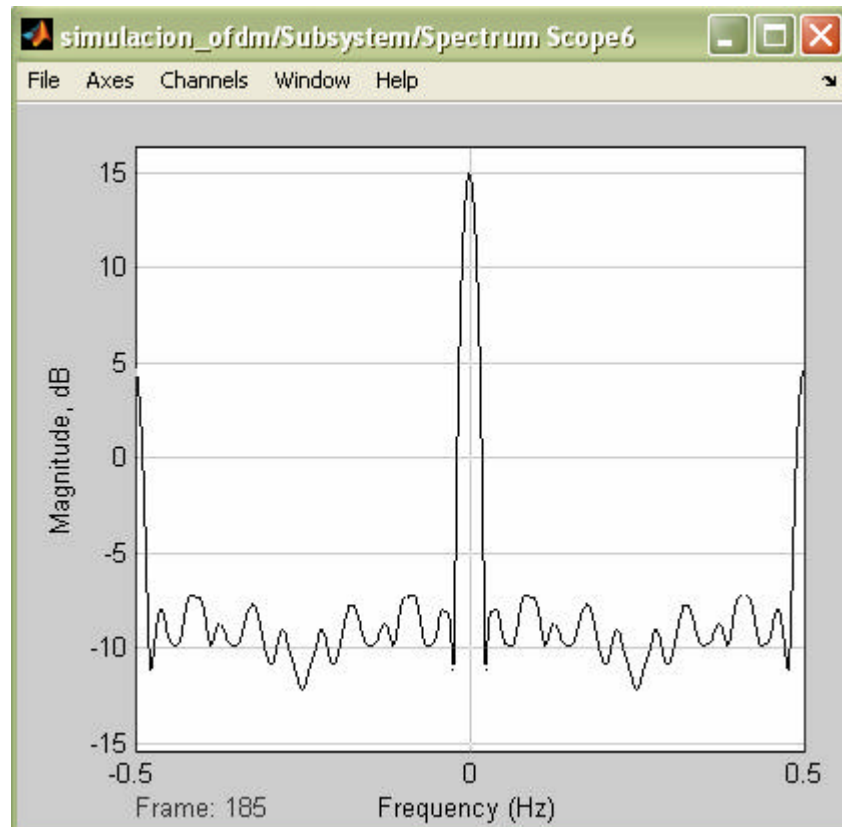


Figura. 8.37. Salida del Demodulador  $M=4$ ,  $SNR=22$ .

0.4951	Error rate
5055	Num Errors
1.019e+004	Num Bits

Figura. 8.38. Indicadores de Datos y Errores del Sistema.

2.  $M=64$ ,  $SNR=50$ .

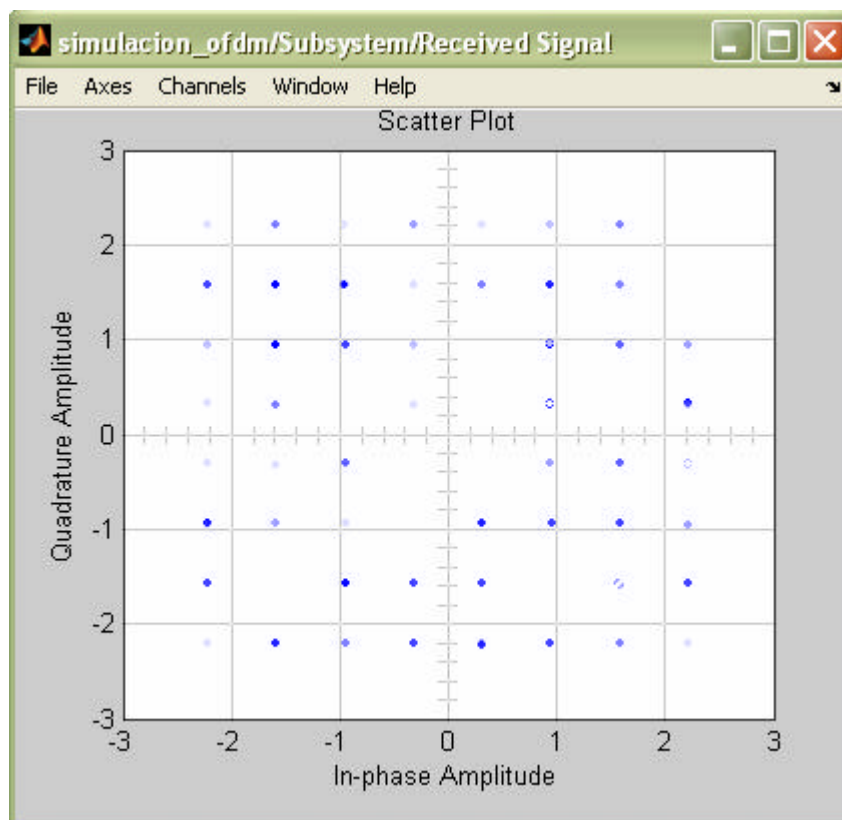


Figura. 8.39. Constelación de Señal Recibida  $M=64$ .

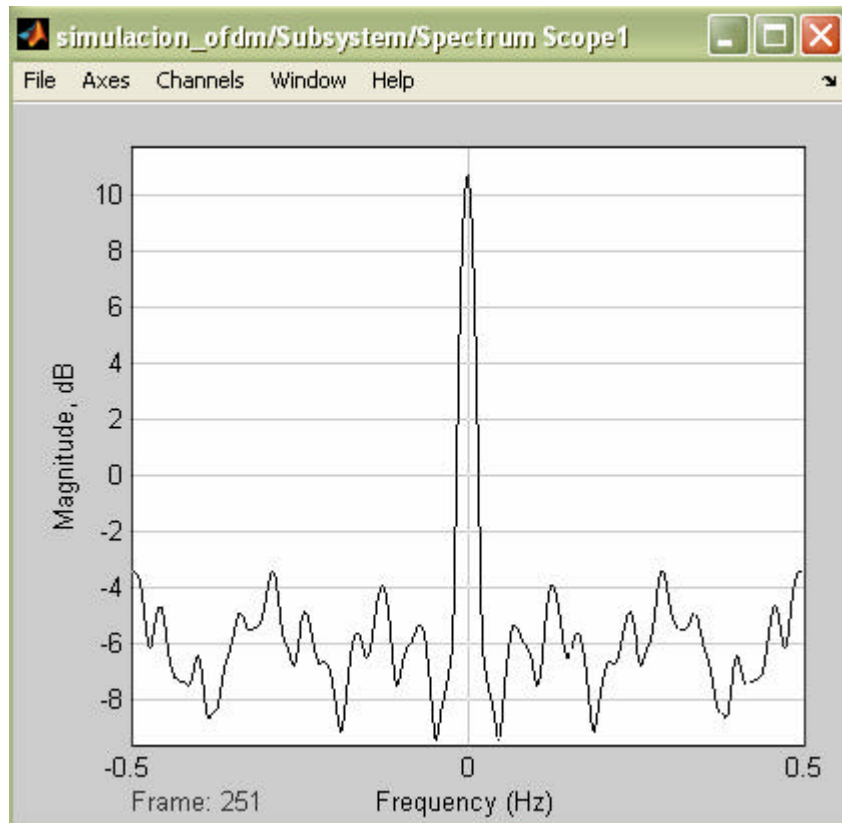


Figura. 8.40. Salida del Generador de Datos  $M=64$ ,  $SNR=50$ .

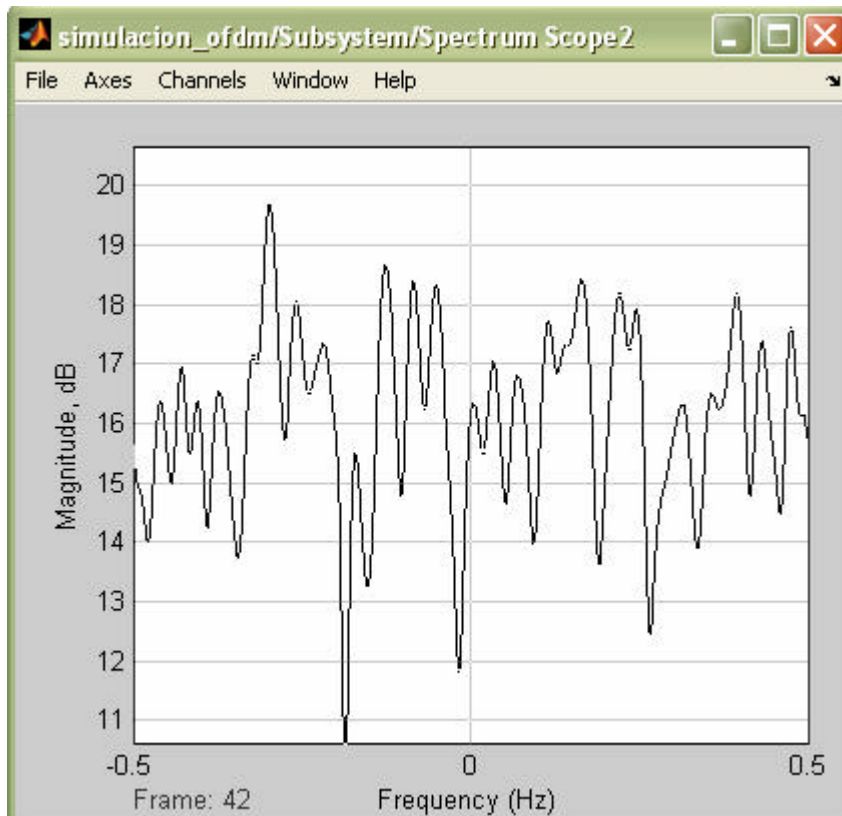


Figura. 8.41. Salida del Modulador  $M=64$ ,  $SNR=50$ .

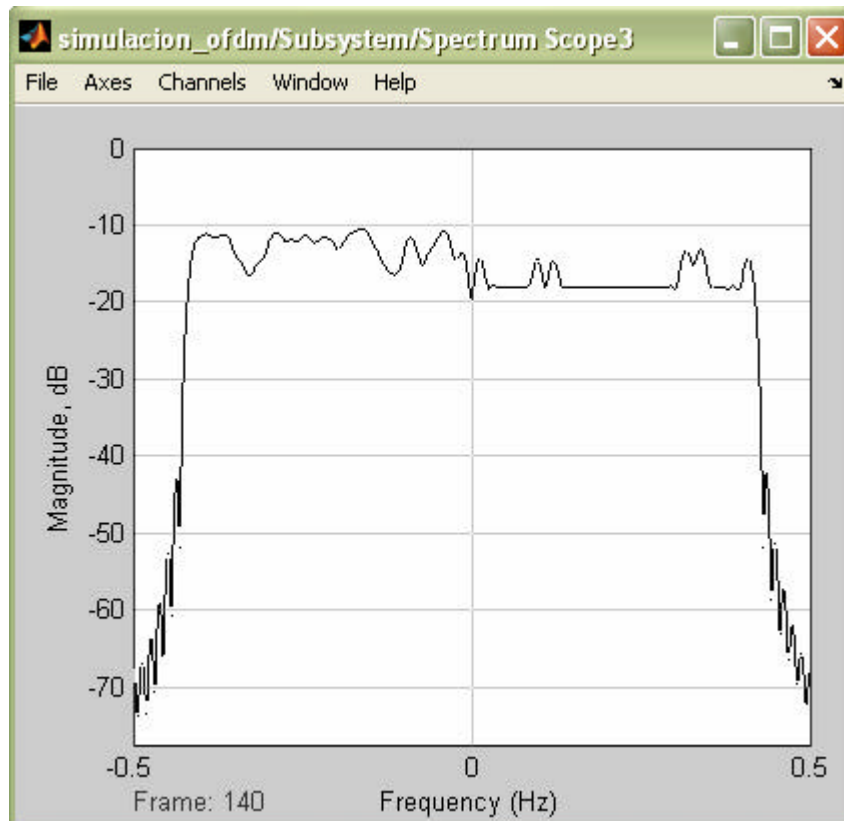


Figura. 8.42. Salida del Transmisor OFDM  $M=64$ ,  $SNR=50$ .

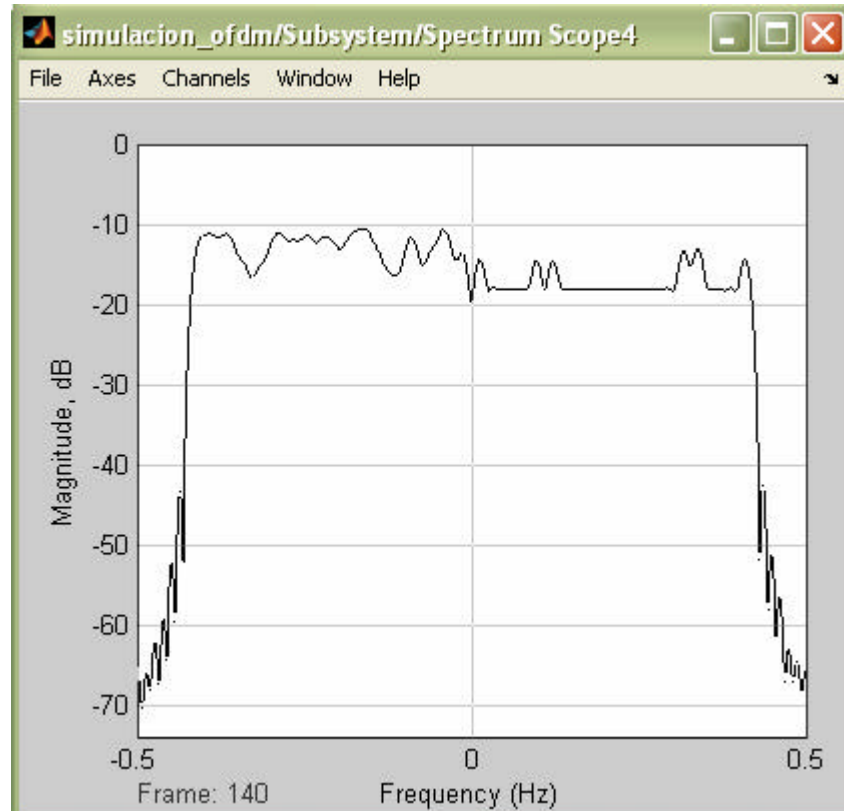


Figura. 8.43. Salida del Canal AWGN  $M=64$ ,  $SNR=50$ .

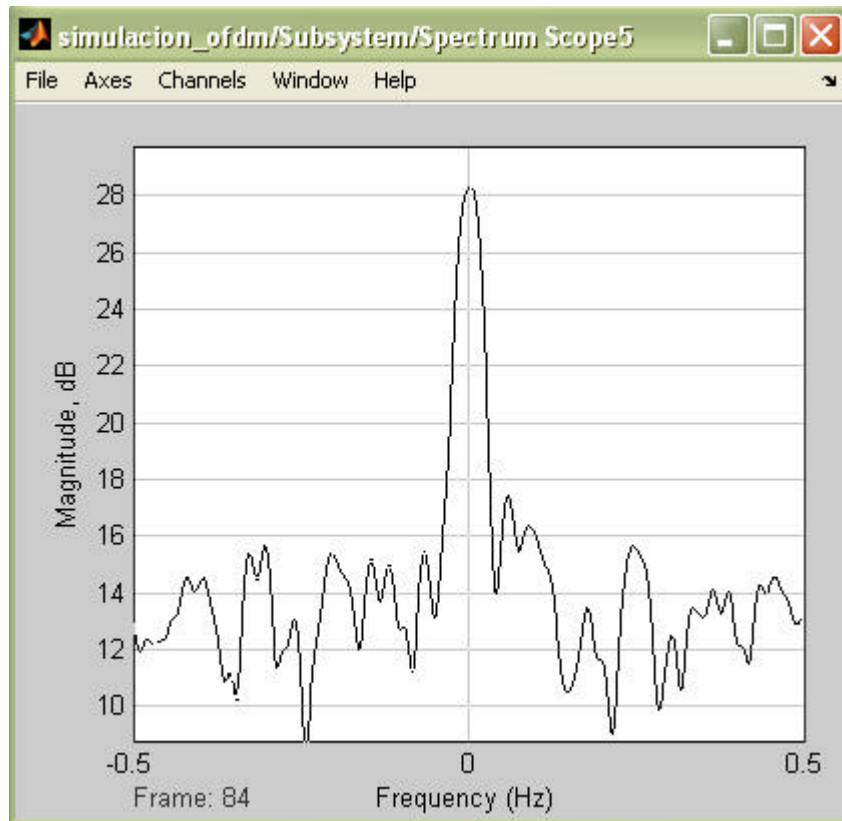


Figura. 8.44. Salida denormalizada del Receptor OFDM  $M=64$ ,  $SNR=50$ .

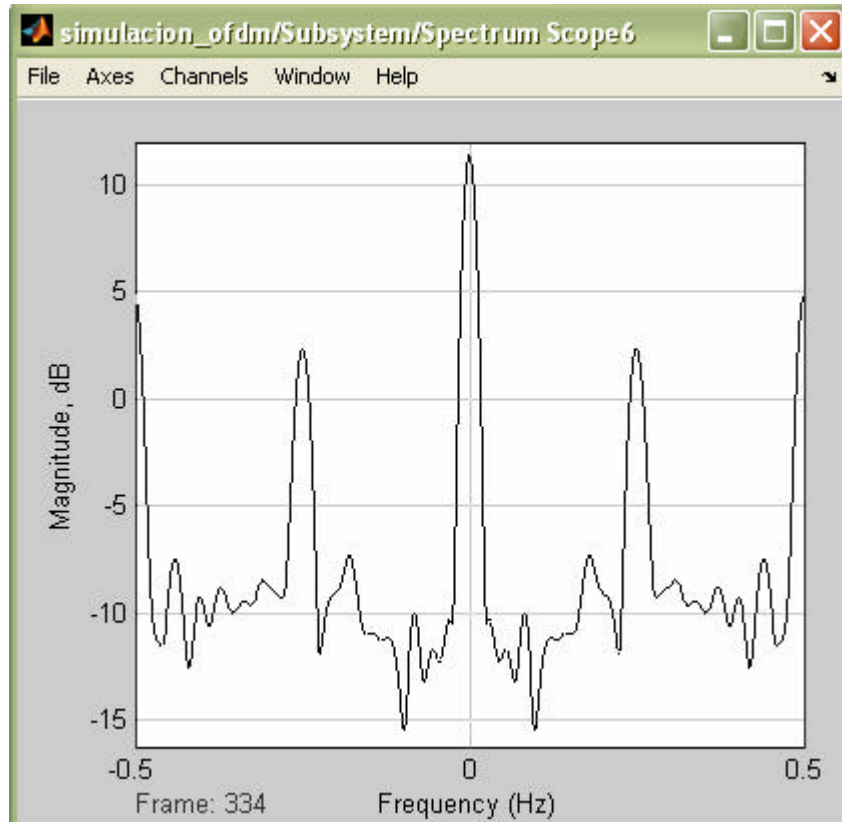


Figura. 8.45. Salida del Demodulador  $M=64$ ,  $SNR=50$ .

0.4879	Error rate
8836	Num Errors
1.811e+004	Num Bits

Figura. 8.46. Indicadores de Datos y Errores del Sistema.

3.  $M= 64$ ,  $SNR= 5$ .

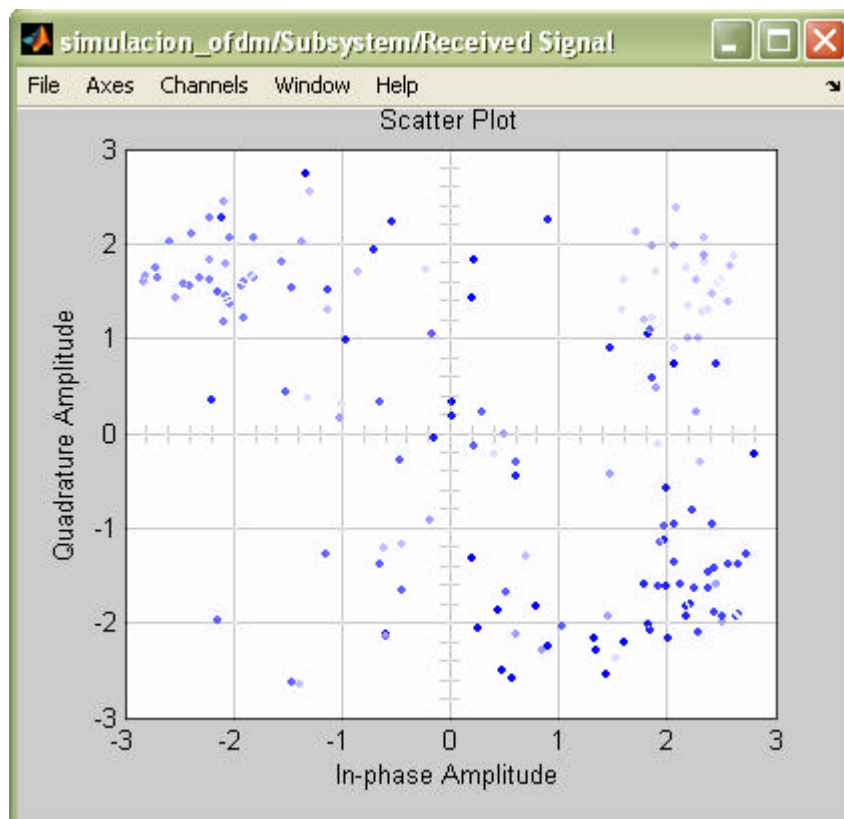
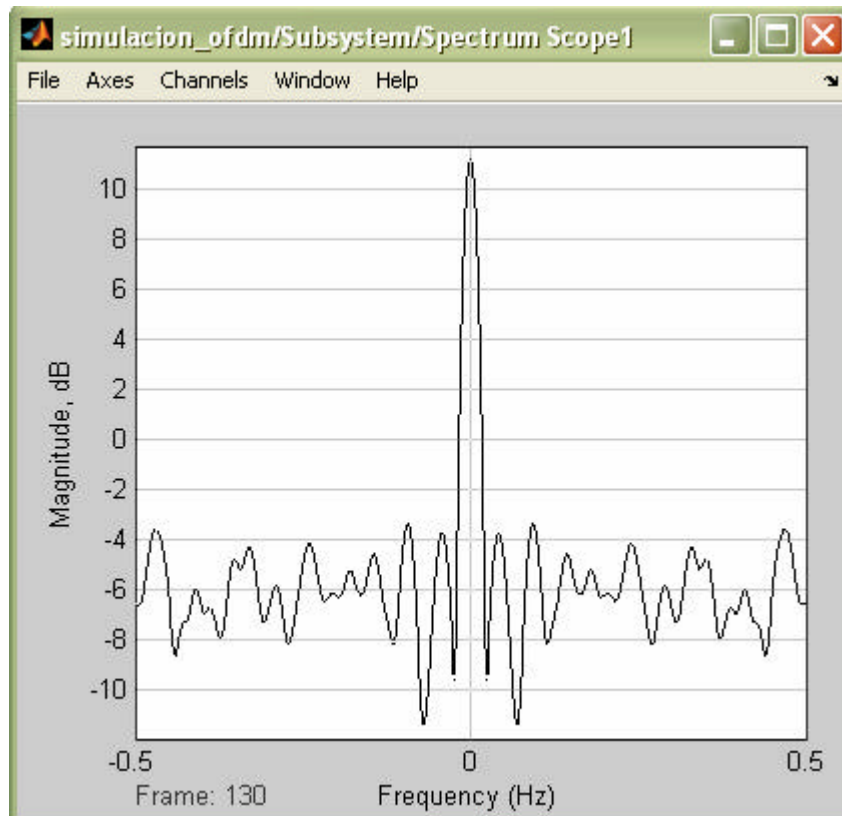
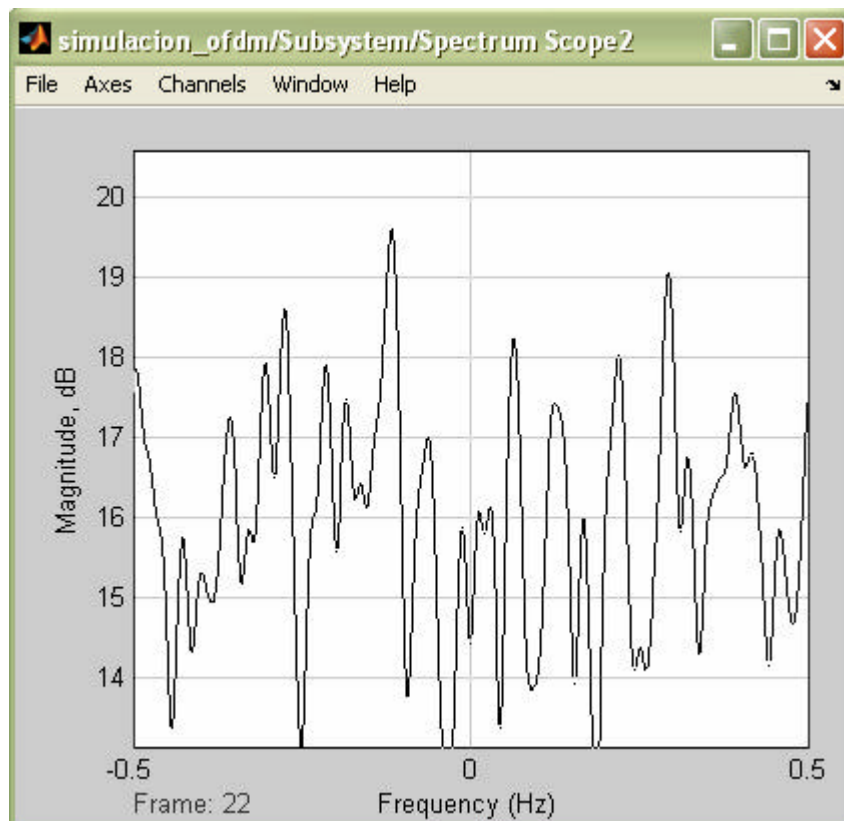


Figura. 8.47. Constelación de Señal Recibida  $M= 64$ .



**Figura. 8.48. Salida del Generador de Datos  $M= 64$ ,  $SNR= 5$ .**



**Figura. 8.49. Salida del Modulador  $M= 64$ ,  $SNR= 5$ .**



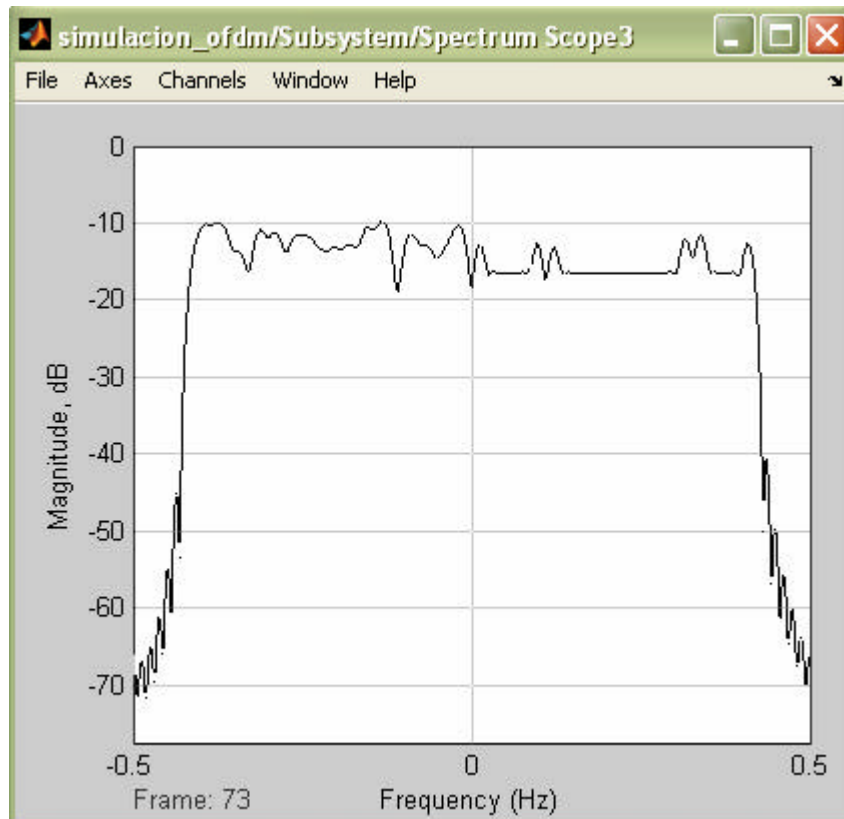


Figura. 8.50. Salida del Transmisor OFDM  $M=64$ ,  $SNR=5$ .

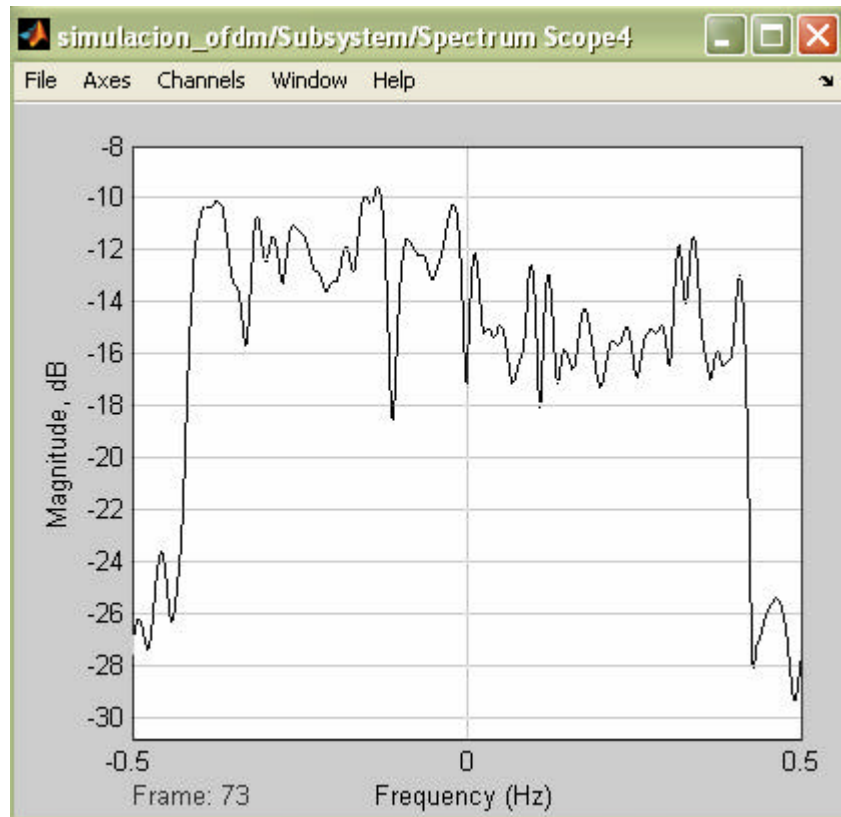


Figura. 8.51. Salida del Canal AWGN  $M=64$ ,  $SNR=5$ .

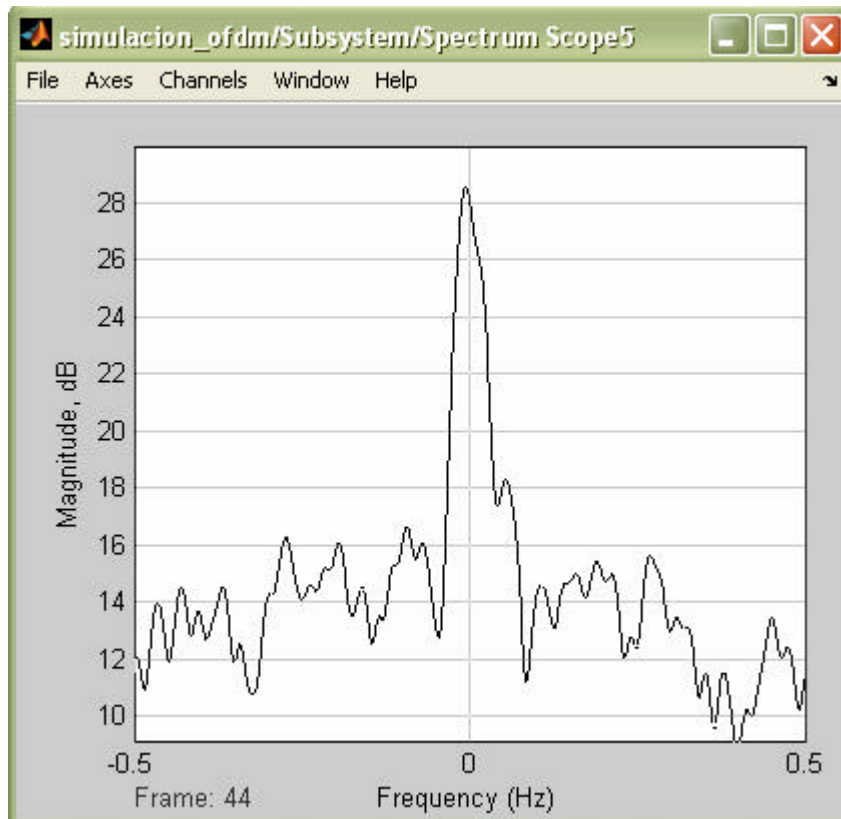


Figura. 8.52. Salida denormalizada del Receptor OFDM  $M=64$ ,  $SNR=5$ .

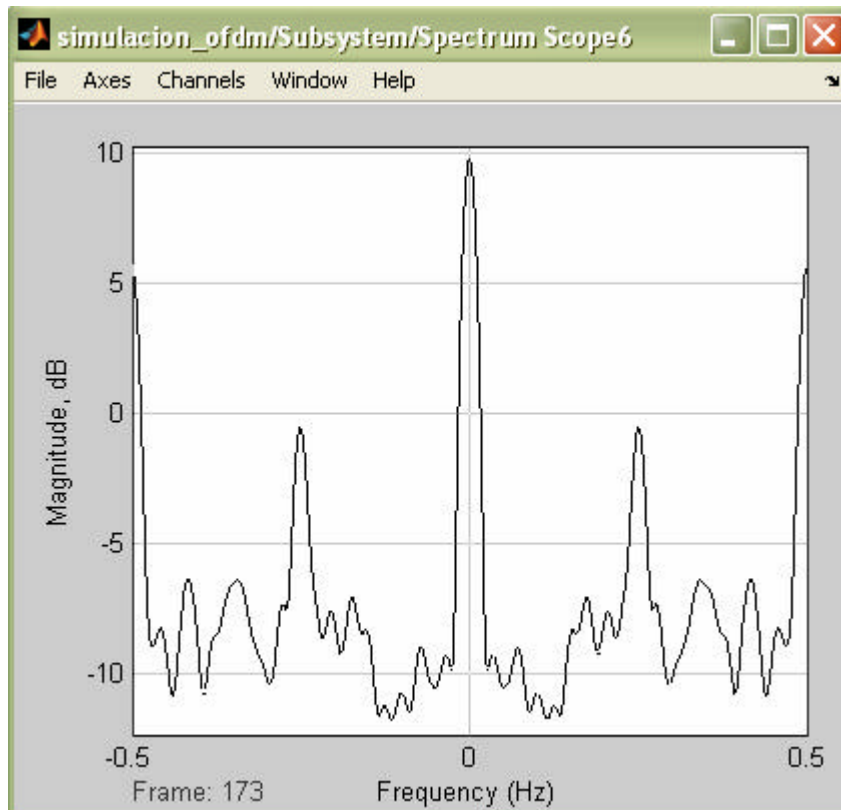


Figura. 8.53. Salida del Demodulador  $M=64$ ,  $SNR=5$ .

0.4945	Error rate
5181	Num Errors
1.048e+004	Num Bits

Figura. 8.54. Indicadores de Datos y Errores del Sistema.

### 8.3.1.2. Conclusiones

- Como se puede observar en los gráficos de constelación de señal recibida, la cantidad de señales mapeadas se define a través del parámetro M antes de ejecutar la simulación.
- Con valores de SNR de canal por encima de 20 dB y con valores más altos a medida que se incrementa M, se puede observar que la constelación de la señal recibida es prácticamente limpia. Es decir, su estructura se apega bastante a la teoría.
- Con valores de SNR bajos, se puede comprobar a través del ultimo ejemplo que la constelación de la señal recibida por el contrario es bastante discontinua y errática.
- Al disminuir también los valores de SNR, el espectro de la señal que entra al canal difiere en mayor proporción de la que se presenta a su salida.
- Se comprende de mejor manera la forma de operación y características de un sistema de comunicaciones que emplea OFDM bajo el estándar 802.11a. El mismo que es empleado para comunicación inalámbrica.
- La familiarización con el manejo de objetos de Simulink, su el entorno y método de desarrollo de los modelos, permite realizar simulaciones de sistemas complejos de forma más fácil y entendible, así como la generación de gráficas. Este modelo requiere 2032 líneas de código script, lo que se convierte realmente en una tarea agotadora de no contar con este simulador en base a objetos.
- El uso de objetos no es restringido y como se observa en esta práctica, se pueden realizar nuevos bloques de procesos o asociar objetos y mascararlos para formar una librería propietaria según las necesidades.

## 8.4. REFERENCIAS

### 8.4.1. LIBROS

- SKLAR, Bernard, **Digital Communications**, Segunda Edición, Prentice Hall, Estados Unidos 2001, páginas 573-590.
- ESPE, **Comunicación Digital**, apuntes de la asignatura, semestre septiembre 2004 – marzo 2005.

### 8.4.2. INTERNET

- <http://www.eie.fceia.unr.edu.ar/ftp/Radioenlaces/1502.pdf>, Modulación de Trellis.
- <http://bips.bi.ehu.es/~inma/psc/tema5.pdf>, Modulación Codificada de Trellis.
- <http://www.internet-solutions.com.co/deacosta/tcm/autor/welcome.html>, Tutor TCM.
- [http://cegt201.bradley.edu/projects/proj2001/ofdmabsh/OFDM\\_Final\\_Report.pdf](http://cegt201.bradley.edu/projects/proj2001/ofdmabsh/OFDM_Final_Report.pdf), Simulación OFDM y Teoría.
- [http://es.wikipedia.org/wiki/Modulaci%C3%B3n\\_por\\_divisi%C3%B3n\\_ortogonal\\_de\\_frecuencia,\\_OFDM](http://es.wikipedia.org/wiki/Modulaci%C3%B3n_por_divisi%C3%B3n_ortogonal_de_frecuencia,_OFDM).
- <http://www.gestiopolis.com/delta/term/TER322.html>, OFDM
- <http://www.depi.itchiuahua.edu.mx/electro/archivo/electro2001/mem2001/articulos/kom3.pdf>, OFDM.
- [http://magnusson.orcon.net.nz/pdfs/V\\_OFDM\\_by\\_IFFT\\_modulation\\_020303.pdf](http://magnusson.orcon.net.nz/pdfs/V_OFDM_by_IFFT_modulation_020303.pdf), Simulación OFDM.

## ANEXOS

### PRÁCTICAS DE LABORATORIO

#### 1.1. CAPÍTULO I

##### 1.1.1. AUTORRELACIÓN DE SEÑAL DE VOZ

###### 1.1.1.1. Objetivos

- Encontrar los 12 primeros retardos (lags) de la función de auto correlación de los fonemas de una palabra, por ejemplo “**palabra**”.
- Realizar un análisis de autocorrelación entre fonemas de dicha palabra.
- Utilizar ventanas adecuadas de análisis para las vocales y para las consonantes.

###### 1.1.1.2. Procedimiento

8. Grabar un comando de voz en un archivo con extensión **.wav**.
9. Utilizando MATLAB, graficar el espectro total de la señal.
10. Definir el tamaño del bloque de análisis.
11. En cada bloque estimar los atrasos de la función de autocorrelación.
12. Estimar valor medio temporal.
13. Restar del valor medio estimado de todas las muestras del bloque.
14. Estimar los atrasos  $r_x(0)$  a  $r_x(12)$ .

15. Normalizar: atraso /  $r_x(0)$ .

16. Comparar los atrasos de la función de autocorrelación en bloques de un mismo fonema, mediante gráficas obtenidas con MATLAB.

Usando MATLAB, para la función de autocorrelación se utiliza el comando **autocorr** que tiene dos parámetros: el vector de entrada para calcular la autocorrelación y el número de retrasos que se desea calcular. El ejemplo del programa para el análisis de una ventana es el siguiente:

<code>[x,y,z] = wavread('nueve.wav');</code>	para obtener el vector de muestras
<code>w=wavread('nueve.wav','size');</code>	para obtener el numero de muestras
<code>for k=2196:2393</code>	
<code>v1(k-2195)=x(k)</code>	obtención de la ventana desde la muestra
<code>end</code>	2196 hasta la 2393 de tiempo 17,5 ms
<code>m=0;</code>	
<code>for k=1:197</code>	
<code>m=m+v1(k);</code>	
<code>end</code>	
<code>m=m/197;</code>	obtención del valor medio
<code>for k=1:197</code>	
<code>v1(k)=v1(k)-m;</code>	resta del valor medio
<code>end</code>	
<code>v1r=autocorr(v1,12);</code>	función de auto correlación
<code>plot(v1r);</code>	

Para cada ventana se utilizan muestras diferentes entonces este programa se repite para cada fonema y para cada ventana de un fonema pero con diferentes valores en los bucles for que sirven para acceder a las muestras.

### 1.1.1.3. Cuestionario

1. Qué concluye de la autocorrelación de las señales obtenida?
2. Cómo se comporta la autocorrelación de señales para las vocales?
3. Cómo se comporta la autocorrelación de señales para las consonantes?
4. Cuáles son sus similitudes y diferencias más notorias?
5. Cuáles de los fonemas presentan mayor similitud entre los atrasos de la función de autocorrelación?

6. Cuáles de los fonemas presentan mayor diferencia entre los atrasos de la función de autocorrelación?

### **1.1.2. ANÁLISIS DE LA SEÑAL DE VOZ POR MODELAMIENTO AUTO REGRESIVO**

El proceso de análisis de la voz se realiza en el transmisor. El proceso incluye un modelo auto regresivo el cual tiene por objetivo minimizar el error cuadrático medio. Este filtro actúa como un predictor lineal. La forma de este modelo toma las muestras anteriores de la señal para estimar una nueva señal en el instante  $n$  y así calcular el error restando de la señal en el instante  $n$ . El algoritmo es el siguiente:

$$e(n) = x(n) + a_1x(n-1) + \dots + a_px(n-p)$$

Donde  $e(n)$  es el la señal de error,  $p$  es el orden del filtro que es este caso va a ser de 12.

Con la utilización de Matlab se puede obtener el vector de la señal, calcular la estimación de  $x(n)$ , restarla de la señal y obtener el error. Cabe destacar que los valores reales de los coeficientes del filtro son  $a_k = -w_k$

#### **1.1.2.1. Procedimiento**

1. Grabar un comando de voz en un archivo con extensión **.wav**. La señal se puede grabar con una fs de 11025 Hz con 16 bits de codificación PCM.
2. Utilizando MATLAB, graficar el espectro total de la señal.
3. Establecer el orden del filtro AR. Se establece  $p = 12$ .
4. Se divide el comando (señal de voz: “nueve”) en bloques de procesamiento de 120 muestras. Con esto se aplica el criterio de estacionariedad.
5. Graficar el espectro de la ventana de Hanning para 120 muestras.
6. A cada bloque se le aplica una ventana de Hanning. Para suavizar los extremos.
7. Graficar el espectro de la señal luego de ser aplicada la ventana de Hanning.
8. Cuando el buffer este lleno (120 muestras), estimar los atrasos  $R_x(0)$ ,  $R_x(1)$ ,...,  $R_x(12)$ .

9. Calcular los coeficientes  $w_k$ :  $w_k = R_x^{-1} \gamma_x$ , donde  $\gamma_x$  es el vector de los retrasos.
10. Procesar el bloque con los  $w$  respectivos.
11. Concatenar todos los bloques de salida para obtener la estimación total.
12. Graficar el espectro de la señal estimada para un bloque de análisis.
13. Finalmente se resta de la señal original la señal estimada y se obtiene el error para ser transmitido.
14. Graficar la señal de error resultante a ser transmitida.
15. Escribir el resultado (señal de error total) en un archivo de formato **.wav**, mediante la función **wavwrite**.

Con la utilización del comando **ARYULE** de Matlab se obtienen directamente los coeficientes  $w$  que cambiados de signo proporcionan los coeficientes  $a_k$  del filtro AR.

En el algoritmo de la programación de este proceso se toman las 12 primeras muestras de la señal como condiciones iniciales. Así se empieza el bloque desde la muestra 13.

### **1.1.2.2. Cuestionario**

1. Cómo se afecta la señal luego de ser aplicada la ventana de Hanning? En sus extremos? En su parte central?
2. Cuáles son las similitudes entre la señal estimada, la señal original y luego de ser aplicada Hanning?
3. Cuáles son las diferencias entre la señal estimada, la señal original y luego de ser aplicada Hanning?
4. Cuáles son las similitudes entre la señal de error y la señal original?
5. Cuáles son las diferencias entre la señal de error y la señal original?
6. Explique que es lo que se escucha al reproducir la señal de error obtenida?
7. Qué concluye de este proceso?



### **1.1.3. TASA DE CRUCE POR CERO**

#### **1.1.3.1. Objetivos**

- Determinar la tasa de cruce por cero para bloques de muestras de una señal de voz.
- Evaluar las estadísticas de la tasa de cruce por cero para determinar las posibles vocales y consonantes.
- Establecer la diferencia de la tasa de cruce por cero para vocales y consonantes.

#### **1.1.3.2. Procedimiento**

1. Grabar un comando de voz en un archivo con extensión **.wav**. La señal se puede grabar con una fs de 11025 Hz con 16 bits de codificación PCM.
2. Utilizando MATLAB, graficar el espectro total de la señal.
3. Dividir la señal en bloques de procesamiento de 120 muestras.
4. Desarrollar un algoritmo que determine cuantas veces la señal cruza por cero.
5. Aplicar este algoritmo a cada uno de los bloques de procesamiento y guardar estas estadísticas.
6. Graficar las estadísticas obtenidas para los bloques de procesamiento de toda la señal de análisis.
7. Determinar en base a esta gráfica donde se encuentran ubicadas las vocales y las consonantes de acuerdo a su mayor o menor concentración.
8. Obtener la gráfica de concentraciones en EXCEL, utilizando los datos estadísticos obtenidos y generar una tabla con dichos valores, donde puedan ser comparados de forma adecuada.

#### **1.1.3.3. Cuestionario**

1. Cómo puede ayudar el uso de la función **signo** a diseñar el algoritmo que determina los cruces por cero de la señal.
2. Dónde se presenta la mayor concentración de la tasa de cruce por cero? Es una vocal o una consonante?
3. Dónde se presenta la menor concentración de la tasa de cruce por cero? Es una vocal o una consonante?
4. Cómo determina gráficamente donde se ubica una u otra vocal o consonante si existen dos continuas de la misma clase?
5. Influye la forma como se grabó el comando de voz en el análisis? El tipo de voz, masculina o femenina?
6. Qué relación encuentra entre la frecuencia y la tasa de cruce por cero? Qué concluye de esta relación?

## 2.1. CAPÍTULO II

### 2.1.1. CÓDIGOS DE LÍNEA.

#### 2.1.1.1. Objetivos

- Investigar como se codifica de forma serial la información binaria para la transmisión en frecuencia de banda base.
- Revisar de forma gráfica los diferentes métodos que son usados en aplicaciones de comunicación de datos en la actualidad en banda base.
- Observar las principales características que diferencian a cada uno de los códigos de los demás y entender el significado de su nomenclatura.
- Obtener las funciones de densidad espectral de potencia asociados con los diferentes códigos de línea.

#### 2.1.1.2. Procedimiento

Para la realización de la presente práctica se utilizan funciones realizadas para MATLAB y cuyo código fuente se encuentran disponibles en los archivos del mismo nombre. Estos archivos se encuentran en el mismo directorio de la práctica y son compatibles con MATLAB7.

1. Se genera una secuencia binaria aleatoria la cual va a ser codificada con los diferentes códigos de línea. Para ello se usa la función **binary()**, cuyo argumento es el número de dígitos binarios deseados para la secuencia.

También se puede utilizar una secuencia fija, para lo cual nosotros designamos un vector de la longitud deseada con los valores binarios que sean requeridos.

2. Con ayuda de la función **wave\_gen** (**secuencia**, **'codigo\_linea'**, **R<sub>b</sub>**) generamos los diferentes códigos de línea para la secuencia dada. Sus argumentos son la “secuencia” generada o determinada en el punto anterior, el “código de línea” en el que se desea codificar y R<sub>b</sub> es la tasa de bit de la secuencia de datos en Hz.

Los códigos posibles son:

- Unipolar NRZ ('unipolar\_nrz')
  - Polar NRZ ('polar\_nrz')
  - Unipolar RZ ('unipolar\_rz')
  - Bipolar RZ ('bipolar\_rz')
  - Manchester ('manchester')
3. Luego se dibuja los códigos de línea generados en el paso anterior por medio de la función **waveplot()**, cuyo argumento es el la secuencia codificada. Para poder apreciar las diferencias entre los diferentes códigos de una misma secuencia a una misma tasa de bits (R<sub>b</sub>), se ejecuta el archivo **Cod\_linea.m** que es el programa principal de esta práctica y que se incluye junto con las funciones. La señal se encuentra muestreada 10 veces en cada tiempo de bit, para obtener resultados más claros.
  4. Para poder determinar la densidad espectral de potencia de los diferentes códigos de línea usamos la función **psd()**. Para ello se genera una nueva secuencia de datos binarios, **bn**, de 2500 elementos, que permite apreciar la densidad espectral de potencia de mejor manera que si se lo hiciera con pocos datos (secuencia b).

### 2.1.1.3. Cuestionario

1. Analizando los archivos de las funciones incluidas, determinar la secuencia de archivos utilizados para la obtención de cada código.
2. Cuál es la consecuencia de utilizar menos muestras por bit? Modifique esta variable en el programa principal y comente los resultados.

3. Cuál es la consecuencia de utilizar más muestras por bit? Modifique esta variable en el programa principal y comente los resultados.
4. Determinar de los gráficos de la forma de onda las diferencias entre los códigos de línea utilizados. Usar la secuencia generada como referencia para este proceso.
5. Cuáles son las diferencias entre la densidad espectral de potencia de los unipolares y polares, y cual es su interpretación?
6. Cómo se determina el ancho de banda usado por cada código de línea? Realizar una tabla indicando el uso de este recurso para los códigos disponibles.
7. De qué depende la densidad espectral de potencia? Modifique la tasa de bit utilizada en el programa principal, así como el número de muestras y analice los resultados.

## 2.1.2. ERRORES EN MODULACIÓN BANDABASE

### 2.1.2.1. Objetivos

- Observar la forma en que es afectada una señal transmitida en modulación banda base por el incremento de ruido AWGN en el canal de transmisión.
- Apreciar la degradación sufrida por una señal transmitida en banda base ante la restricción del ancho de banda disponible.
- Obtener los diagramas de ojo de las señales afectadas por el incremento de ruido en el canal de transmisión banda base.
- Obtener los diagramas de ojo de las señales degradadas por la restricción de ancho de banda del canal de transmisión.

### 2.1.2.2. Procedimiento

Para la realización de la presente práctica se utilizan funciones realizadas para MATLAB y cuyo código fuente se encuentran disponibles en los archivos del mismo nombre. Estos archivos se encuentran en el mismo directorio de la práctica y son compatibles con MATLAB7.

El código fuente de esta práctica se encuentra disponible en el archivo **Error\_BB.m**.

1. Se pueden determinar los efectos del ruido en un canal, así como del ancho de banda del mismo. Para ello, se usa la función **channel(entrada, ganancia, potencia\_ruido, ancho\_banda)**, la cual permite obtener la respuesta de una señal al canal, pudiendo modificarse los valores de potencia de ruido en Watts y ancho de banda en Hz. La entrada es la secuencia codificada que ingresa al canal, la ganancia del canal generalmente se considera la unidad. Utilizar un canal de 10 KHz con un AWGN de 20 mW, con ganancia 1 y cuya entrada sea la señal 'polar\_nrz'.

2. Alterar la cantidad de AWGN introducido en el canal, utilizar cuatro valores de este parámetro, por ejemplo 20 mW, 0.2W, 2W y 5W, para la misma secuencia del paso anterior.
3. Alterar el ancho de banda disponible para realizar la transmisión, empezar con valores altos e ir disminuyendo. Utilizar cuatro valores para este parámetro, para la misma secuencia del paso 1.
4. Los efectos del canal y el ruido pueden ser mejor visualizados mediante el diagrama del ojo. Este diagrama de ojo se genera mediante barridos, que son ligeramente más anchos que el periodo de bit  $T_b = 1/R_b$ . En este caso se usará un ancho de barrido de  $2T_b$ .
5. Para poder visualizar el efecto de la introducción de AWGN en un canal a través del diagrama del ojo, consideramos los valores del paso 2 y obtenemos el diagrama de ojo para cada uno de ellos.
6. Para poder visualizar el efecto de la restricción de ancho de banda en un canal, a través del diagrama del ojo, consideramos los valores del paso 3 y obtenemos el diagrama de ojo para cada uno de ellos.
7. Obtenemos los diagramas de ojo para cada uno de códigos de línea generados en la práctica anterior, para un canal de ganancia 1, AWGN de 20mW y 10000 Hz de ancho de banda.

### **2.1.2.3. Cuestionario**

1. Analice y entienda el funcionamiento del programa principal y de las funciones utilizadas para esta práctica.
2. Observando los espectros generados para las alteraciones de ruido, cómo afecta el aumento o disminución de ruido AWGN en un canal de transmisión?

3. Observando los espectros generados para las restricciones de ancho de banda, cómo afecta el aumento o disminución de la disponibilidad de este recurso en un canal de transmisión?
4. Cómo difieren los diagramas de ojo de los diferentes códigos de línea utilizados? En qué son similares?
5. Repetir los tres primeros pasos para una secuencia unipolar y responda para ello las primeras dos preguntas del cuestionario.
6. Cuáles son las similitudes y diferencias observadas del comportamiento polar vs. El unipolar?



### 3.1. CAPÍTULO III

#### 3.1.1. MODULACIÓN DIGITAL

##### 3.1.1.1. Objetivos

- Utilizar los conocimientos de transmisión digital banda base y de modulación de onda continua para el estudio de transmisión digital pasabanda.
- Generar secuencias en banda base para luego utilizando las funciones desarrolladas para MATLAB poder modularlas en pasa banda.
- Modular una secuencia bandabase con modulación por desplazamiento de Amplitud ASK y observar la densidad espectral de potencia con respecto a la densidad espectral de potencia (PSD) de la secuencia inicial en banda base.
- Modular una secuencia bandabase con modulación por desplazamiento de Fase PSK y observar la densidad espectral de potencia con respecto a la densidad espectral de potencia (PSD) de la secuencia inicial en banda base.
- Modular una secuencia bandabase con modulación por desplazamiento de Frecuencia FSK y observar la densidad espectral de potencia con respecto a la densidad espectral de potencia (PSD) de la secuencia inicial en banda base.

##### 3.1.1.2. Procedimiento

Para la realización de la presente práctica se utilizan funciones realizadas para MATLAB y cuyo código fuente se encuentran disponibles en los archivos del mismo nombre. Estos archivos se encuentran en el mismo directorio de la práctica y son compatibles con MATLAB7.

El código de esta práctica se encuentra en el archivo **Mod\_Dig.m**.

1. Para esta práctica la señal de datos binarios tiene una tasa de bits de 1000 bps y la señal modulada una amplitud pico de 1 V.
  
2. Para la generación de una señal ASK:
  - Se genera una secuencia de 60 datos binarios cuyos primeros 6 dígitos sean [1 1 0 0 1 0] usando la función **binary** para los 54 datos restantes.
  - Para obtener la señal ASK, con una frecuencia de portadora de 6 KHz, se puede generar una secuencia unipolar NRZ de la secuencia de datos binarios generada anteriormente.
  - Se mezcla o multiplica esta secuencia NRZ con la salida de un oscilador que opera a 6 KHz, habiendo uso de la función **mixer (entrada, freq\_portadora)**.
  - Debido a que se tienen 40 muestras por bit, se grafican las 400 primeras muestras para obtener los 6 bits conocidos de la secuencia de entrada y cuatro adicionales generados por **binary**.
  - Luego se visualizan la densidad espectral de potencia de ambas señales usando la función **psd (señal)**, siendo el argumento para ella, la señal antes y después de la modulación.
  
3. Para la generación de una señal PSK:
  - Para obtener una señal PSK con una portadora de 6 KHz, se lo puede tener tras generar una secuencia polar NRZ a partir de la secuencia de datos binarios determinados anteriormente.
  - Se multiplica la señal de la secuencia NRZ con la salida de un oscilador operando a 6 KHz, haciendo uso de la función **mixer**.
  - Debido a que se tienen 40 muestras por bit, se grafican las 240 primeras muestras para obtener los 6 bits conocidos de la secuencia de entrada.
  - Luego se visualizan la densidad espectral de potencia de ambas señales usando la función **psd (señal)**, siendo el argumento de esta función, la señal antes y después de la modulación.

#### 4. Para la generación de una señal FSK

- Para obtener una señal FSK de fase continua con frecuencia de marca y espacio de 4KHz y 8KHz respectivamente, se puede partir de una secuencia de datos codificados en polar NRZ de la secuencia de entrada ya conocida.
- Se aplica la forma de onda polar en la entrada de un oscilador controlado por voltaje, **VCO**, que para esta practica tiene una frecuencia de trabajo libre de 6 KHz y una sensibilidad de frecuencia de  $-2 \text{ KHz} / \text{V}$ .
- Debido a que se tienen 40 muestras por bit, se grafican las 400 primeras muestras para obtener los 6 bits conocidos de la secuencia de entrada más cuatro generados por la función **binary**.
- Luego se visualizan la densidad espectral de potencia de ambas señales usando la función **psd (señal)**, siendo el argumento de esta función, la señal antes y después de la modulación.

#### 3.1.1.3. Cuestionario

1. De la gráfica obtenida cuales son las principales características de la señal modulada mediante ASK?
2. De la gráfica obtenida cuales son las principales características de la señal modulada mediante PSK?
3. De la gráfica obtenida cuales son las principales características de la señal modulada mediante FSK?
4. Observando la densidad espectral de potencia de las señales originales vs. las moduladas, que se puede apreciar con respecto a sus lóbulo principal? Explique a que se debe?
5. Se producen los mismos efectos para ASK, PSK y FSK? Si no lo son, explique su diferencia y su justificación.
6. Repetir la simulación para un código Manchester, analizar los resultados.

## Código Mod\_Dig.m

```
% Modulación Digital,
% Generación de Señales ASK, PSK y FSK
% Densidad Espectral de Potencia.

clc;
clear all;

% *****

SAMPLING_CONSTANT = 40;
BINARY_DATA_RATE = 1000;
SAMPLING_FREQ = BINARY_DATA_RATE * SAMPLING_CONSTANT;

global SAMPLING_CONSTANT;

% *****

b=[1 1 0 0 1 0 binary(54)];

unr=wave_gen(b,'unipolar_nrz',1000);
ask=mixer(unr, osc(6000));

tt=[1:400];

figure(1)
subplot(2,1,1), waveplot(unr(tt))
subplot(2,1,2), waveplot(ask(tt))

figure(2)
subplot(2,1,1), psd(unr)
subplot(2,1,2), psd(ask)

pnr=wave_gen(b,'polar_nrz',1000);
psk=mixer(pnr, osc(6000));

tt=[1:240];

figure(3)
subplot(2,1,1), waveplot(pnr(tt))
subplot(2,1,2), waveplot(psk(tt))

figure(4)
subplot(2,1,1), psd(pnr)
subplot(2,1,2), psd(psk)

fsk= vco (pnr);

tt=[1:400];

figure(5)
subplot(2,1,1), waveplot(pnr(tt))
subplot(2,1,2), waveplot(fsk(tt))

figure(6)
subplot(2,1,1), psd(pnr)
subplot(2,1,2), psd(fsk)
clc;
```

### 3.1.2. DETECCIÓN DE SEÑALES PASABANDA

#### 3.1.2.1. Objetivos

- Comprobar mediante los resultados de la simulación las diferencias entre la detección coherente y la no coherente de señales moduladas digitalmente.
- Realizar la detección de señales moduladas digitales de forma Coherente y No Coherente utilizando las funciones que han sido desarrolladas para MATLAB.
- Observar los efectos de tener error en la fase del oscilador de recuperación en la recepción del sistema de comunicación.
- Observar la influencia y consecuencias de la existencia de desplazamientos de frecuencia en el oscilador de recuperación en el receptor del sistema.
- Apreciar el desempeño de sistemas de modulación digital pasabanda ante la presencia de ruido AWGN en el canal de transmisión contaminando las señales que pasan por él.

#### 3.1.2.2. Procedimiento

Para la realización de la presente práctica se utilizan funciones realizadas para MATLAB y cuyo código fuente se encuentran disponibles en los archivos del mismo nombre. Estos archivos se encuentran en el mismo directorio de la práctica y son compatibles con MATLAB7.

El código de esta práctica se encuentra en el archivo **Coh\_NonCo.m**.

1. Para realizar la detección Coherente de las señales ASK y PSK:
  - a) Se multiplica la señal modulada ASK o PSK por una portadora generada localmente en el receptor, a la misma frecuencia y fase de la portadora usada en el transmisor.
  - b) Se visualiza la señal obtenida a la salida del multiplicador (**mixer**), junto con su densidad espectral.

- c) Se aplica esa señal a un filtro casado mediante la función **match** ('código', **entrada**), donde código se refiere al código de línea usado en el transmisor y la entrada es la señal salida del multiplicador.
  - d) Se muestra la salida del filtro casado para ASK y PSK.
2. Para observar los efectos del error de fase, se demodula la señal ASK usando un oscilador que incluya este error de la forma **osc (freq\_portadora, error\_fase)**, con respecto a la fase de la portadora, por ejemplo 30°, 60° y 120°.
3. Para observar los efectos del error de fase, se demodula la señal PSK usando un oscilador que incluya este error de la forma **osc (freq\_portadora, error\_fase)**, con respecto a la fase de la portadora, por ejemplo 30°, 60° y 120°.
4. Para visualizar los efectos de la desviación de frecuencias en la demodulación de señales ASK, se demodula dicha señal con osciladores a 5800 y 5980 Hz, y se grafican los resultados.
5. Para la detección No Coherente de señales, se usa un filtro pasa banda que reduce el ruido fuera de banda y la interferencia, un detector de envolvente conformado por un rectificador y un filtro pasa bajos, cuyo ancho de banda es mayor ancho de banda de la señal y menor a la frecuencia de portadora.
6. Se selecciona por tanto un ancho de banda del LPF de 4000 Hz a ser usado en la función **envelope (entrada, BW)**. El resultado de esto se grafica junto con la señal ASK.
7. Finalmente para comprender mejor el desempeño del sistema pasabanda frente al ruido, se genera una señal binaria de 600 datos, siendo los primeros bits [1 1 0 0 1 0], la cual se modula como ASK usando la función **mixer**.
8. Esto se introduce en un canal con ganancia 1, ruido AWGN= 4 mW, y suficiente ancho de banda para que no se introduzcan distorsiones. Se grafican los primeros diez datos de la señal transmitida antes y después de pasar por el canal.

9. Se usa un detector Coherente para demodular la señal obtenida después de pasar por el canal. Se grafica este resultado con el diagrama de ojo correspondiente a la salida del filtro casado.

### 3.1.2.3. Cuestionario

1. De la gráfica de la salida del filtro casado, obtener cuál es la atenuación del filtro casado sobre la señal?
2. Explicar los efectos de la introducción de un error de fase en la recepción de una señal detectada coherentemente.
3. Explicar los efectos del desplazamiento de frecuencias en la recepción de una señal detectada coherentemente.
4. Modificar el ancho de banda del filtro LPF en la detección no coherente y explicar su respuesta.
5. Modificar el valor de ancho de banda disponible en el canal de transmisión y observar sus efectos sobre el espectro de la señal transmitida y su diagrama de ojo.
6. Determinar el mínimo ancho de banda, para que manteniendo el nivel de ruido, se puede obtener un diagrama de ojo razonable de la señal.
7. Variar el valor del ruido y determinar los efectos sobre la señal si se cuenta con un ancho de banda adecuado.

### Código Coh\_NonCo.m

```
% Detección Coherente y No Coherente
% Efectos de Desplazamiento de Fase
% Efecto de Diferentes Frecuencia de Oscilador de Recuperacion
% Respuesta al Canal

clc;
clear all;

% *****

SAMPLING_CONSTANT = 40;
BINARY_DATA_RATE = 1000;
SAMPLING_FREQ = BINARY_DATA_RATE * SAMPLING_CONSTANT;

global SAMPLING_CONSTANT;

% *****
```

```
b=[1 1 0 0 1 0 binary(54)];

unr=wave_gen(b,'unipolar_nrz',1000);
ask=mixer(unr, osc(6000));

pnr=wave_gen(b,'polar_nrz',1000);
psk=mixer(pnr, osc(6000));

fsk= vco (pnr);

c_ask=mixer(ask, osc(6000));

tt=[1:400];

figure(1)
subplot(2,1,1), waveplot(c_ask(tt))
subplot(2,1,2), psd(c_ask(tt))

c_psk=mixer(psk, osc(6000));

tt=[1:240];

figure(2)
subplot(2,1,1), waveplot(c_psk(tt))
subplot(2,1,2), psd(c_psk(tt))

ask_mat=match('unipolar_nrz',c_ask);
psk_mat=match('polar_nrz',c_psk);

figure(3)
subplot(2,1,1), waveplot(ask_mat(1:400))
subplot(2,1,2), waveplot(psk_mat(1:240))

c_ask1=mixer(ask, osc(6000,30));
c_ask2=mixer(ask, osc(6000,60));
c_ask3=mixer(ask, osc(6000,120));

ask_mat1=match('unipolar_nrz',c_ask1);
ask_mat2=match('unipolar_nrz',c_ask2);
ask_mat3=match('unipolar_nrz',c_ask3);

c_psk1=mixer(psk, osc(6000,30));
c_psk2=mixer(psk, osc(6000,60));
c_psk3=mixer(psk, osc(6000,120));

psk_mat1=match('polar_nrz',c_psk1);
psk_mat2=match('polar_nrz',c_psk2);
psk_mat3=match('polar_nrz',c_psk3);

tt=[1:400];

figure(4)
subplot(3,1,1), waveplot(ask_mat1(tt))
subplot(3,1,2), waveplot(ask_mat2(tt))
subplot(3,1,3), waveplot(ask_mat3(tt))

tt=[1:240];

figure(5)
```



```
subplot(3,1,1), waveplot(psk_mat1(tt))
subplot(3,1,2), waveplot(psk_mat2(tt))
subplot(3,1,3), waveplot(psk_mat3(tt))

ask_dev1=match('unipolar_nrz',mixer(ask,osc(5800)));
ask_dev2=match('unipolar_nrz',mixer(ask,osc(5980)));

tt=[1:400];

figure(6)
subplot(3,1,1), waveplot(ask_mat(tt))
subplot(3,1,2), waveplot(ask_dev1(tt))
subplot(3,1,3), waveplot(ask_dev2(tt))

nc_ask=envelope(ask,3000);

figure(7)
subplot(2,1,1), waveplot(ask(tt))
subplot(2,1,2), waveplot(nc_ask(tt))
clc;

% *****

bn=[1 1 0 0 1 0 binary(594)];

unrn=wave_gen(bn,'unipolar_nrz',1000);
askn=mixer(unrn,osc(6000));

ch_ask=channel(askn,1,0.004,10000);

tt=[1:400];

figure(8)
subplot(2,1,1), waveplot(askn(tt))
subplot(2,1,2), waveplot(ch_ask(tt))

ask_zm=match('unipolar_nrz', mixer(ch_ask, osc(6000)));

figure(9)
eye_diag(ask_zm);

clc;
```

## 4.1. CAPÍTULO IV

### 4.1.1. CÓDIGOS DE CONTROL DE ERRORES

#### 4.1.1.1. Objetivos

- Revisar de forma experimental el funcionamiento de algunos de los algoritmos de codificación de canal estudiados.
- Observar por medio de la simulación la respuesta de los códigos de bloques ante un error.
- Comprender de mejor manera la generación de códigos cíclicos y determinar si son o no sistemáticos.
- Realizar la codificación convolucional de una secuencia de datos de longitud variable y su respectiva decodificación mediante el algoritmo de Viterbi
- Corregir una secuencia de código con bits erróneos por medio del algoritmo de Viterbi.

#### 4.1.1.2. Procedimiento

Para la realización de la presente práctica se utilizan funciones realizadas para MATLAB y cuyo código fuente se encuentran disponibles en los archivos del mismo nombre. Estos archivos se encuentran en el mismo directorio de la práctica y son compatibles con MATLAB7.

1. Códigos de Bloques (7,4). Se ejecuta el archivo **bloques.m**, en el cual se pueden cambiar el vector de datos (`dat_vec`) o el vector de código errado (`code_err`).
  - Como se tiene  $n=7$  y  $k=4$ , quiere decir que  $m=3$ . Es decir, que la palabra de datos es de 4 bits, los bits redundantes son 3 y el código final es de 7 bits.
  - Se ingresa un vector de datos de 4 bits.

- Se ingresa la matriz generadora de la codificación.
- Por medio de la función **mult\_mod2 (vect\_datos, matr\_generadora)**, que realiza el producto de las matrices ingresadas como parámetros, se realiza la codificación del vector  $x$  con la matriz  $G$ , dando un vector de código.
- Se obtiene la matriz de chequeo de paridad por medio de la función **check\_matrix (mat\_generadora)** a partir de la matriz  $G$ .
- Para verificar la presencia de errores se utiliza nuevamente la función **mult\_mod2 ( )**, pero los parámetros para este caso son el vector de código y la transpuesta de la matriz de chequeo de paridad.
- Se realiza nuevamente el paso anterior pero se cambia el vector de código de manera que se conserve su longitud y se comparan los resultados.

2. Códigos cíclicos (7,4). Se ejecuta el archivo **ciclico.m**, en el cual se pueden cambiar el vector de datos (`dat_vec`) o el polinomio generador (`pol_g`).

- Como tenemos que  $n=7$  y  $k=4$ , quiere decir que  $m=3$ . Es decir, que la palabra de datos es de 4 bits, los bits redundantes son 3 y el código final es de 7 bits. Se considera el ejemplo revisado en la teoría de códigos cíclicos.
- Se ingresa un vector de datos de 4 bits.
- Se ingresa el polinomio generador de la codificación, que puede ser cualquiera de las tres opciones:  $x + 1$ ,  $x^3 + x + 1$ ,  $x^3 + x^2 + 1$ .
- Por medio de la función **cyclic\_matrix (pol\_g, longitud)**, que obtiene la matriz generadora a partir del polinomio generador y la longitud es la misma que la del vector de datos, en este caso cuatro.

- Se genera la forma sistemática de la matriz generadora por medio de la función **syst\_matrix ( )**, cuyo argumento es precisamente la matriz obtenida en el paso anterior.
  - Finalmente se obtiene la palabra de código por medio de la función **mult\_mod2( )**, cuyos parámetros son el vector de datos y la matriz generadora para código no sistemático, o el vector de datos y la forma sistemática de la generadora, para códigos sistemáticos.
  - Las respuestas obtenidas demuestran el ejemplo considerado y las diferencias propias de ellas.
3. Códigos Convolucionales (3,1, 3). Se ejecuta el archivo **convoluc.m**, en el cual se puede cambiar el vector de datos (**dat\_vec**) al que se desee.
- Como se tiene  $n=3$  y  $k=1$  y  $m=3$ , quiere decir que los datos entran bit a bit, el código generado es de 3 bits por cada bit que ingresa y que la memoria de código es de 3 bloques (actual y dos anteriores).
  - Se ingresa un vector de datos cuya longitud puede ser determinada por el usuario; un vector más largo incrementará el código generado y el cálculo computacional.
  - Se ingresa la matriz generadora, que viene a ser la matriz de conexiones de los bloques con los sumadores en binario, tal como se muestra en las gráficas de códigos convolucionales revisadas en la parte teórica.
  - Se define el número de desplazamientos por ciclo,  $k=1$ .
  - Se genera el código convolucional del vector de datos ingresado por medio de la función **conv\_encode (Gen, dat\_vec, k)**, cuyos parámetros son la

matriz generadora, el vector de datos ingresado y el número de desplazamientos.

- Una vez obtenida la secuencia codificada, se genera un código errado a partir de la misma. Se realiza de forma que exista un error en cada tercio del código generado.
- Para decodificar este tipo de códigos se utiliza generalmente el algoritmo de Viterbi, el cual se puede simular a través de la función **viterbi(Gen, Cod\_Bip)**, cuyos argumentos son la matriz generadora y la secuencia de código pero de forma bipolar por lo que se la multiplica por dos y se le resta uno, obteniendo valores  $\pm 1$ .
- Se aplica el mismo procedimiento pero ahora con la secuencia de código modificada para que contenga errores. El algoritmo de Viterbi debe detectar y corregir dichos errores. Se compara las respuestas.

#### 4.1.1.3. Cuestionario

1. Analizar los programas proporcionados: bloques.m, ciclico.m y convoluc.m y entender su lógica de funcionamiento.
2. Una vez entendido esto, describir la secuencia de operación de estos códigos y de las funciones incluidas para su ejecución, cuyos archivos también se proporcionan.
3. Cambiar el vector de datos del archivo bloques.m y verificar su funcionamiento.
4. Modificar el vector de error cuatro veces y tabular los resultados obtenidos.
5. Comprobar por escrito la realidad de dichos resultados. De existir diferencias, explicar su origen.
6. Cambiar el vector de datos del archivo ciclico.m y verificar su funcionamiento.
7. Repetir la simulación con el nuevo vector de datos para los otros dos polinomios generadores.
8. Comparar los resultados obtenidos y determinar por escrito la veracidad de los mismos tanto para sistemáticos como para no sistemáticos.

9. Modificar el vector de datos del archivo convoluc.m y comprobar su funcionamiento.
10. De los resultados obtenidos, verificar la corrección de errores y tabular los códigos generado y erróneo, destacando los bits que han sido cambiados para ser considerados error.

### **Código bloques.m**

```
%Codificación en Bloques
```

```
clear all;  
clc;
```

```
dat_vec= [ 1 0 1 0 ];  
mat_gen= [ 1 0 0 0 1 0 1 ; 0 1 0 0 1 1 1 ; 0 0 1 0 1 1 0 ; 0 0 0 1 0 1 1];
```

```
code= mult_mod2(dat_vec,mat_gen)  
H= check_matrix(mat_gen)
```

```
mult_mod2(code,(H'))
```

```
code_err=[ 1 0 1 0 0 1 0]  
mult_mod2(code_err,(H'))
```

### **Código ciclico.m**

```
%Códigos Cíclicos
```

```
clear all;  
clc;
```

```
dat_vect=[1 0 1 0]
```

```
pol_g= [ 1 1 0 1 ];
```

```
Gen= cyclic_matrix(pol_g,4)
```

```
Gsis= syst_matrix(Gen)
```

```
code= mult_mod2(dat_vect,Gen)
```

```
sis_code= mult_mod2(dat_vect,Gsis)
```

### **Código convoluc.m**

```
%Códigos Convolucionales
```

%Decodificacion y Corrección de Errores Con Viterbi

clear all;

clc;

dat\_vec= [0 1 0 1]

Gen= [ 1 0 0 ; 1 0 1 ; 1 1 1 ];

k=1;

code= conv\_encode(Gen,dat\_vec,k)

[l,ln]=size(code);

code\_err=code;

for (i=1:ln/3:ln)

    code\_err(i)=~code\_err(i);

end

code\_err

dat=viterbi(Gen,2\*code-1)

dat\_corr=viterbi(Gen,2\*code\_err-1)

## 6.1. CAPÍTULO VI

### 6.1.1. SISTEMA DE COMUNICACIÓN CDMA CON SPREAD SPECTRUM

#### 6.1.1.1. Objetivos

- Implementar un sistema de comunicación con Acceso Múltiple por División de Código CDMA, usando la técnica de Espectro Ensanchado.
- Comprender la forma de utilización de los códigos Gold dentro de la técnica de espectro ensanchado.
- Analizar la forma de transmisión y recepción de las secuencias de datos, a las que se les aplica los códigos Gold.
- Verificar el desempeño de este tipo de sistemas frente al incremento de la señal de ruido en un canal de transmisiones.
- Aprender la influencia en el incremento del error de la comparación de un canal de comunicaciones por dos usuarios que utilizan la técnica de espectro ensanchado.
- Observar el desempeño del sistema frente al incremento del número de usuarios dentro de un mismo canal, interferencia multi usuario.

#### 6.1.1.2. Procedimiento

1. El código de esta práctica se encuentra en el archivo **CDMA\_SS.m**, junto con el cual se encuentran los archivos de las funciones necesarias para su ejecución y que han sido creadas para este caso particular.
2. Se parte de un conjunto de códigos Gold, compuesto por 33 códigos de longitud 31 (realmente se dispone de códigos de longitud 32, pero para efectos de la simulación, se descarta el primer chip de cada uno de ellos para conseguir la longitud deseada de 31 chips).
3. Para ello se realiza la lectura del archivo **gold32**, y se los limita a 31 chips.



4. Previamente se realiza la prueba de funcionamiento del sistema de comunicación CDMA con espectro ensanchado, para lo cual:
  - a. Se establece la secuencia de datos que se va a transmitir.
  - b. Se obtiene la secuencia que va a ingresar al canal por medio del comando  **$x0 = \text{cdma\_tx}(\text{datos0},1,\text{gold31}(1,:),0)$** ; el cual hace uso de la función `cdma_tx()` que requiere como parámetros la secuencia de datos y la secuencia de Gold.
  - c. Se obtiene la salida del detector de correlación por medio del comando  **$r0 = \text{cdma\_rx}(x0,\text{gold31}(1,:))$** ; en donde se usa la función `cdma_rx()`, en la que se ingresan como parámetros la secuencia ingresada al canal y la secuencia de Gold con la que va a correlacionarse.
  - d. Se obtiene la probabilidad de error del sistema.
  
5. En primer lugar se simula el funcionamiento del sistema para un usuario único, con un nivel de ruido creciente en el tiempo. Para ello se realizan algunas modificaciones al paso anterior como:
  - a. Se genera una secuencia aleatoria bipolar de datos con 100 elementos.
  - b. Se obtiene la secuencia de datos que ingresa al canal de manera análoga a lo revisado en el paso previo (4).
  - c. Se genera un vector de ruido de igual longitud que la secuencia que ingresa al canal y se los suma.
  - d. Se obtiene la secuencia de salida del detector de correlación de forma análoga a lo que se hizo en el paso anterior (4).
  - e. Se obtiene la probabilidad de error del sistema.
  
6. Ahora se simula el funcionamiento del sistema para dos usuarios, con un nivel de ruido creciente en el tiempo. Para lo cual:
  - a. Se genera una secuencia aleatoria bipolar de datos con 100 elementos para el nuevo usuario.
  - b. Se obtiene la secuencia de datos que ingresa al canal para el segundo usuario, de la misma manera como se realiza para un usuario.

- c. El nuevo usuario transmite con una potencia diez veces superior a la del primer usuario, por lo que se suma las secuencias de ambos usuarios más el ruido generado en el paso 5, y el resultado es lo que ingresa realmente al canal
  - d. Se obtiene la secuencia de salida del detector de correlación de forma análoga a lo que se hizo para un usuario.
  - e. Se obtiene la probabilidad de error del sistema
7. Otra simulación del funcionamiento del sistema se realiza manteniendo constante el nivel de ruido del canal y variando la amplitud de la señal del segundo usuario de forma creciente en el tiempo. Para lo cual:
  - a. Se modifica la señal del segundo usuario de forma que aumente su amplitud a lo largo del tiempo de simulación.
  - b. La señal del segundo usuario modificada se suma con la del primer usuario para determinar que es lo que ingresa realmente al canal, y que es el parámetro que debe utilizar el receptor CDMA.
  - c. Se obtiene la secuencia de salida del detector de correlación de forma análoga a lo que se hizo para un usuario.
  - d. Se obtiene la probabilidad de error del sistema.
8. La última simulación del funcionamiento del sistema se realiza aumentando el número de usuarios del sistema de comunicaciones, el nivel de ruido del canal es constante y las señales de los usuarios son hasta 100 veces mayores de la que se considero en un principio. Para esta parte final, se realiza el mismo proceso que para un usuario, con la diferencia que no se generan ni se obtienen vectores, sino matrices de 33 vectores, o usuarios.

### **6.1.1.3. Cuestionario**

1. Analizar los archivos entregados y entender su lógica de funcionamiento.
2. Explicar en sus propias palabras la forma como considera que esta funcionando la simulación para cada uno de los casos revisados.

3. Qué sucede con la probabilidad de error para cada uno de los casos expuestos, de que depende su comportamiento.
4. Cómo aprecia la relación entre la señal de ruido y la señal transmitida? Cumple con la definición de espectro ensanchado?
5. Realice un análisis y un comentario de la forma de onda recibida para los diferentes casos, frente a la forma de onda la señal enviada.
6. Qué sucede con la probabilidad de error de bit cuando se tienen los 33 usuarios (canal lleno)?

## 7.1. CAPÍTULO VII

### 7.1.1. PROYECTO DE DISEÑO DE UN SISTEMA DE COMUNICACIÓN CELULAR

Se presenta un proyecto de diseño de un sistema de comunicación celular, basado en una serie de simulaciones en MATLAB y con el objeto de aplicar lo aprendido en el desarrollo del curso de Comunicación Digital.

El proyecto consta de tres partes:

- (i) Diseño del codificador de voz con muestreo y cuantización.
- (ii) Diseño de la modulación y forma de pulso.
- (iii) Diseño del esquema de acceso múltiple y el presupuesto del enlace.

#### 7.1.1.1. Diseño del Codificador de Voz con Muestreo y Cuantización

En esta parte se tendrá que diseñar un cuantizador no uniforme que será usado para la transmisión de voz para el sistema celular. Se deberá escoger la situación óptima para estos niveles de cuantización. Aunque existe un diseño óptimo propuesto, se tendrá la mayor parte del crédito por encontrar un buen diseño alternativo. Se tiene libertad de consultar con otras personas, aunque la decisión final del diseño será únicamente la que se tome por uno mismo.

1) Una señal de voz humana tiene un ancho de banda aproximado de 3500 Hz. Se dan muestras de señal de voz tomadas a razón de 8000 muestras/segundo. Se deberá asumir que cada muestra  $x$  es una variable aleatoria Gaussiana con una pdf dada por

$f_x(x) = \frac{1}{\sqrt{2p}} e^{-\frac{x^2}{2}}$ . Se deberá determinar un conjunto de 16 niveles de cuantización para un

cuantizador no uniforme de tal manera que su SNR sea máximo. Se cuenta con un archivo

m-file **quant1.m** que se lo llama desde MATLAB colocando el archivo quant1.m en el directorio de trabajo y tecleando:

**quant1 ([level1 level2...level16], n\_trials)**

Donde level1, level2,..., level16 son los niveles de cuantización (en orden ascendente), y donde n\_trials es el número de iteraciones que se desea correr para determinar los niveles de cuantización.

La respuesta final deberá contar con:

- (a) Un listado de los niveles de cuantización.
- (b) El promedio de la SNR para la simulación.

#### **7.1.1.1.1. Observaciones:**

- (i) Se deberá usar el conocimiento de cuantizador no uniforme. Se debe concentrar los niveles de cuantización más pesados en áreas de densidad de probabilidad más grandes.
- (ii) El tamaño de la PDF será simétrico con respecto a cero, tiene sentido que un buen cuantizador probablemente debe ser también simétrico con respecto a cero.
- (iii) Se deberá correr la simulación del programa unas cuantas veces para conseguir una percepción de para cuántas muestras se necesita simular para conseguir consistencia razonable de resultados: n\_trials=100, son muy pocos ensayos, pero n\_trials=50000, toma demasiado el ejecutar muchos experimentos.

2) Usar el método analítico para calcular el SNR para el cuantizador del problema. Calcular la distorsión promedio necesaria en el ejercicio.

3) Se desea simular un cuantizador diferencial que tenga diferencias entre las muestra de dos señales consecutivas con un nivel de cuantización de 16 niveles.

Para el trabajo final, se debe proveer:

- (a) Una lista de los niveles de cuantización.
- (b) La relación SNR de la simulación.

**Observation:**

Debido a que el procedimiento de resolución de este problema será similar a la parte (1), la dispersión de valores para el cuantizador debe ser mucho más concentrada y la SNR significativamente mayor.

**7.1.1.2. Diseño de la Modulación y Forma de Pulso**

En la primera parte se realizó un esquema de cuantización diferencial para un sistema celular para transmisión de una señal de voz con SNR > 25 dB, tasa de datos de 32 Kbps. Ahora se realiza el diseño de la modulación y forma de pulso para transmitir una señal de voz sobre un enlace de comunicación celular.

**7.1.1.2.1. Requerimientos de diseño.**

Se tienen dos consideraciones que se deben satisfacer:

1. La señal transmitida debe ocupar un canal no mayor a 50 KHz. Todos los componentes espectrales fuera de esos 50 KHz de ancho de banda deben estar al menos 40 dB por debajo del valor de la densidad espectral de potencia.
2. Para poder tener una fidelidad aceptable, la señal debe ser transmitida con una BER no mayor a 0.001.

Una vez logrados estos requerimientos, el objetivo es minimizar el costo de manufactura de los equipos de comunicación.

**7.1.1.2.2. Condiciones de Diseño.**

**Modulación:** Se ha determinado que se usará modulación PSK, pero se debe determinar entre BPSK ( $M=2$ ), QPSK ( $M=4$ ), o 8-PSK ( $M=8$ ) para obtener una mayor eficiencia espectral. Mayores valores de  $M$  proporcionan mayor eficiencia espectral, aunque se ha comprobado que  $M=2$  o  $M=4$  son los mejores.

**Forma de Pulso:** Se usará la forma de pulso de coseno levantado, pero se debe determinar el factor de rolloff ( $0 < r < 1$ ). Valores pequeños de  $r$  proporcionan mejor eficiencia espectral pero es ligeramente más costoso y puede producir grandes lóbulos laterales cuando el pulso es truncado.

**Recepción Coherente / Diferencial:** Recepción coherente proporciona mejor desempeño, pero la recepción diferencial es mucho más barata de implementar.

**Filtro Casado / PasaBajos:** Un filtro casado tiene un desempeño óptimo. Como una alternativa más económica algunos receptores celulares usan un simple filtro pasabajos. En términos de desempeño nos cuesta aproximadamente 1 dB.

**Código de Corrección de Error:** Se puede usar un código de corrección de error en el sistema, similar al usado en IS-95. Esto añade complejidad (costos) al sistema y duplica el ancho de banda debido a los bits extra transmitidos. Sin embargo, esto representa también una reducción del  $E_b/N_0$  requerido en aproximadamente 1 a 5 dB, dependiendo de las condiciones de transmisión.

#### 7.1.1.2.3. Costos.

Cada uno de las condiciones elegidas son determinantes. De esta forma se determina el costo final del sistema de acuerdo a los siguientes criterios:

**Modulación:** Entre más compleja es la modulación, más cara resulta implementarla. BPSK cuesta \$2, QPSK cuesta \$4 y 8-PSK cuesta \$8.

***Eb/No:*** Mayor potencia de la señal requiere baterías más grandes, lo que representa mayor costo. Asumimos que cada dB adicional de energía cuesta \$1.

***Forma de Pulso:*** Implementar un factor de rolloff muy agudo puede añadir cierta complejidad. Asumimos que el costo del filtro de forma de onda esta dado por  $\$(1-r)^2$ .

***Recepción Coherente/Diferencial:*** Un receptor coherente cuesta \$5 más que un receptor diferencial.

***Filtro Casado/PasaBajos:*** Es \$ 1.50 más costoso implementar un filtro casado que un simple filtro pasabajos.

***Código de corrección de Error:*** Si se escoge usar un código de corrección de error para ahorrar energía, esto nos costará \$ 1.50.

#### 7.1.1.2.4. Herramienta de Simulación.

El archivo m (función) de MATLAB **mod\_sim.m** se usa para determinar la modulación y forma de pulso de acuerdo los resultados entregados con su ejecución. Una vez colocado en el directorio de trabajo adecuado, se utiliza MATLAB y se corre la simulación con el comando:

**mod\_sim (M, r, diff, matched\_f, code, Eb\_No\_dB, n\_trials)**

Donde:

**M** = el número de símbolos de modulación (Debe ser 2,4, o 8).  
**r** = es el factor de rolloff del filtro de coseno levantado ( $0 < r < 1$ ).  
**diff** = 1 indica que se está usando detección diferencial.  
0 indica que se usa detección coherente.  
**matched\_f** = 1 indica que se está utilizando filtro casado.  
0 indica que se utilice un filtro pasabajos simple.  
**code** = 1 indica que se está usando código de corrección de error.  
0 indica que no se usa código de corrección de error.  
**Eb\_No\_dB** = el valor de Eb/No en dB (usualmente,  $5\text{dB} < \text{Eb/No} < 15\text{dB}$ ).  
**n\_trials** = número de iteraciones para la simulación ( $1000 < n\_trials < 10e6$ ).



Así se tiene por ejemplo:

**mod\_sim (4,0.9, 0, 1, 1, 6.7, 20000);**

Lo que indica que se ha designado una modulación QPSK (4) con factor de rolloff de coseno levantado 0.9, usando filtro de detección coherente (0), un filtro casado receptor (1) y código de corrección de error (1). También se tiene para este ejemplo un  $E_b/N_0 = 6.7$  dB, y se ejecutan 20000 iteraciones de la simulación.

El programa retornará las siguientes salidas:

- Un dibujo de la forma de pulso y de la potencia espectral de potencia de la señal transmitida. La gráfica de la densidad espectral de potencia ayuda a comprobar si se verifican los requerimientos de ancho de banda.
- El valor simulado de BER para el sistema.
- Una estimativa de la fiabilidad de la simulación de la BER. En general una fiabilidad menor que 0.5 indica que se tienen que realizar más iteraciones, una fiabilidad entre 0.5 y 1 indica que la estimación es casi confiable, y una fiabilidad mayor que 1 nos indica que no se ganaría mucho más exactitud si se aumenta el número de iteraciones.
- El costo calculado de la implementación del sistema.

### **Observaciones:**

- Mientras que la simulación le ayudará en la estimación del desempeño final y costos del diseño escogido, debe usar los conocimientos adquiridos en este semestre para seleccionar los parámetros del sistema. No se debe realizar por simple asignación de valores a los parámetros, sino razonando el por qué de los mismos.
- Probablemente es más fácil encontrar en primera instancia un diseño que cumpla con los requerimientos y limitaciones de ancho de banda, y luego trabajar en minimizar los costos.
- Siempre se puede cumplir con los requerimientos de BER aumentando  $E_b/N_0$ , aunque esto represente un incremento en los costos.

**Resultados:**

- Los resultados deben incluir la selección de los cinco parámetros de diseño.
- Una breve justificación (un párrafo) del por qué de la selección de cada parámetro.
- Las gráficas de los resultados de la simulación del diseño elegido.

**7.1.1.3. Diseño del Esquema de Acceso Múltiple y Presupuesto del Enlace**

En la parte primera, se diseñó un esquema de cuantizador diferencial para el sistema celular que era capaz de transmitir una señal de voz con una SNR > 25 dB a una tasa de datos de 32 kbps = 8000 muestras/seg \* 4 bits/muestra. En la parte segunda se diseñó la modulación y forma de pulso para que la señal transmitida tenga un ancho de banda de 50 KHz y el BER no sea mayor a 0.001. En la parte tercera, se debe elegir un esquema de acceso múltiple para este sistema de comunicaciones.

**7.1.1.3.1. Técnica de Acceso Múltiple.**

Cada uno de los enlaces requiere de un ancho de banda de 50 KHz. Se dispone de un total de 7 MHz de ancho de banda disponible para el sistema. Se puede escoger entre dos opciones para la técnica de Acceso Múltiple del sistema:

**FDMA/TDMA**

En este esquema se utilizará 4 usuarios por canal para crear 35 canales con un ancho de banda de 200 KHz cada uno. Esto tiene la ventaja de la simplicidad, pero solo 1/7 de los canales pueden ser usados en cada celda.

**CDMA**

Si se elige utilizar CDMA como técnica de acceso múltiple, se obtendrá una ganancia de procesamiento de 140. El número de usuarios por celda será determinado mediante la fórmula de aproximación Gaussiana para el BER de un sistema CDMA. Se debe asumir el

mismo  $E_b/N_0$  que calculó para la parte 2, y se debe asegurar que con la interferencia de acceso múltiple, el BER no ascienda a más de 0.002.

**Resultados:**

- Para la entrega de los resultados se debe de incluir el cálculo de la capacidad de cada uno de estos sistemas en términos del número de usuarios por célula, y una justificación escrita de las ventajas y desventajas de la selección del diseño.
- Esto último, no debe de ser mayor a dos páginas.

## **8.1. CAPÍTULO VIII**

### **8.1.1. ORTHOGONAL FREQUENCY-DIVISION MULTIPLEXING**

#### **8.1.1.1. Objetivos.**

- Comprender de mejor manera la forma como se realiza la Multiplexación por División de Frecuencia Ortogonal.
- Aprender a través del modelamiento en Simulink (MATLAB) la estructura de un sistema de comunicación que utiliza OFDM.
- Conocer algunos de los objetos disponibles en Simulink para efectuar simulaciones en MATLAB de forma gráfica, evitando realizar toda la codificación en scripts.
- Observar la respuesta en frecuencia del modelo de OFDM desarrollado, así como su diagrama de constelación para diferentes tipos de modulación pasabanda: BPSK, QPSK, 16-QAM y 64-QAM.

#### **8.1.1.2. Procedimiento**

El modelo de Simulink presenta un sistema simplificado de transmisión OFDM que introduce la aplicación de elementos IFFT / FFT en el proceso de modulación / demodulación de datos de acuerdo con el estándar IEEE802.11a.

El sistema de comunicaciones por tanto presenta tres etapas básicas:

- Transmisor.
- Canal.
- Receptor.

Dentro de cada una de estas etapas se encuentran diferentes procesos y subprocesos que se modelan mediante Simulink a través de la utilización de los objetos disponibles en las diferentes librerías. En el sistema también se añaden objetos que permiten realizar mediciones, observaciones de espectro y otro tipo de instrumentación virtual.

En este caso particular se usa objetos que permitan observar el espectro de la señal en diferentes puntos del sistema llamados **Spectrum Scope** y que se encuentran en la librería del Blockset de Procesamiento de Señales; y, objetos **Discrete Time Scatter Plot Scope** que permitan observar la constelación de la señal transmitida y se encuentran en la librería del Blockset de Comunicaciones.

Para realizar los modelos en MATLAB, se hace uso de enmascaramientos, los mismos que permiten esconder varios bloques u objetos bajo uno solo, simplificando la vista general del modelo y haciéndolo modular para sus modificaciones, ya que de ponerse todos los objetos en una sola ventana de modelo, la estructura se vuelve muy compleja.



#### **Vista del Modelo de Sistema de Comunicación con OFDM**

Cargar el modelo de la simulación OFDM en MATLAB. El archivo **simulación\_ofdm.mdl**, se proporciona para su análisis. Al llamarlo, se visualizará el bloque de la figura.

Para poder observar los bloques de procesos u objetos bajo la máscara, se hace clic derecho sobre el bloque de la figura y se selecciona **Look Under Mask** y aparece otra ventana conteniendo al subsistema correspondiente a este modelo.

Como se puede apreciar en el modelo, se incluyen otros objetos que permiten simular un sistema de comunicaciones. Es decir, no se trata solo del transmisor y receptor OFDM utilizando un canal AWGN.

Para poder observar la forma como se encuentran conectados todos los subsistemas, podemos recurrir al **model browser** en donde podemos observar las conexiones por subsistemas y por enlaces.

#### **8.1.1.2.1. Transmisión.**

En la parte de transmisión se incluye la generación de la señal, la modulación pasabanda, la normalización de la misma y el transmisor OFDM.

#### **8.1.1.2.2. Recepción.**

En la etapa de recepción se incluyen el receptor de OFDM. Este receptor trabaja de forma inversa al transmisor, encontrando en primera instancia un bloque que remueve los prefijos cíclicos, luego la FFT para demodular OFDM, conversor de tramas, removedor de **Zero Pad**, selector de filas y entrega los datos.

También se encuentran en la recepción un denormalizador que revierte el proceso de normalización, un demodulador de QAM y un bloque que detecta y corrige errores de acuerdo al algoritmo de Viterbi.

Los objetos usados en este modelamiento existen en las librería de **Simulink 6 (R14)** que es la versión que viene incluida en la instalación de **MATLAB 7 /R14)**. Esta aclaración es importante por cuanto existe, al igual que con las funciones vistas en otras prácticas, la posibilidad de crear librerías propias para un determinado modelamiento, así como de objetos y subsistemas de acuerdo con la flexibilidad propia de MATLAB.

### 8.1.1.2.3. Simulación.

Para ejecutar la simulación se debe seleccionar **start** en el menú **Simulation** o hacer clic en el botón **play** de **Simulink**, con lo cual se empezará a correr el modelo diseñado hasta que se decida hacer una pausa o detenerlo completamente.

Previo a la ejecución se deben de configurar dos parámetros del modelamiento analizado. Estos parámetros son:

- La señalización M-aria a utilizar para el mapeo de los datos.
- La SNR del canal de transmisión.

Esto se debe realizar en la ventana que se accede haciendo doble clic sobre el bloque del modelo. El bloque IFFT acepta solo señales complejas en su entrada, por ello la señal real M-aria debe ser convertida en una de tipo complejo, lo cual se logra a través del mapeo complejo de los datos.

De acuerdo con el estándar IEEE802.11a, al cual se refiere este modelamiento, se debe aplicar uno de los métodos de mapeo listados en la siguiente tabla para la transmisión OFDM. Inicialmente los valores de mapeo y SNR son 8 y 30 respectivamente, por lo que deben ser cambiados para cumplir con las recomendaciones del estándar que se nombran a continuación.

Así, el parámetro **M** es el que define el método a utilizar y no puede ser alterado mientras se ejecuta la simulación.

<b>M</b>	<b>Mapping Mode</b>
<b>2</b>	BPSK
<b>4</b>	QPSK
<b>16</b>	16-QAM
<b>64</b>	64-QAM

**Valores de M válidos para Mapeo de Acuerdo al estándar IEEE802.11a**

Para lograr transmisiones en OFDM con alto desempeño del error, la SNR debe ser mayor que 20 dB. Entre más alta sea el método de mapeo utilizado, más alta deberá ser la SNR necesaria para obtener el mejor desempeño de error posible. Este parámetro si puede ser alterado mientras se este ejecutando la simulación.

Para cumplir con el estándar IEEE802.11a, se deben de cumplir con otros parámetros, los cuales son configurados en la sección **initial commands** en el editor de máscara del sistema. Estos parámetros se listan a continuación:

<code>%Subcarriers related parameters according to ch.17.3.2.3 of IEEE Std802.11a</code>
<code>%</code>
<code><b>Nsd=48;</b> % Number of data subcarriers</code>
<code><b>Nsp= 4;</b> % Number of pilot subcarriers</code>
<code><b>Nst =Nsd+Nsp;</b> % Number of subcarriers, total (52)</code>
<code>%</code>
<code><i>% Parameters of names not following Std802.11a labeling</i></code>
<code>%</code>
<code><b>BW=20e6;</b> %Cumulative signal bandwidth (20Msamples/s)</code>
<code><b>ChTSamp=1/BW;</b> % The time duration between samples in the channel</code>
<code><b>SubchannelBW=BW/64;</b> %Subchannels signal bandwidth</code>
<code><b>Tbit=1/SubchannelBW;</b> %Subchannel signal sample time</code>

#### Parámetros Establecidos como Comandos Iniciales del Sistema

El estándar IEEE802.11a define un total de 52 subcanales sobre los cuales se deben transmitir los datos. Esto incluye 4 subcanales usados para señalización piloto de la transmisión.

### Resultados.

Las gráficas resultantes de la ejecución de este modelo permiten ver la evolución de la señal sobre el sistema de comunicación utilizando OFDM, así como la cantidad de datos y errores obtenidos al final de la cadena de transmisión.



### 8.1.1.3. Cuestionario

1. Identificar todos los subsistemas y máscaras que existen en el modelo `simulacion_ofdm.mdl` y enumerarlos, junto con una descripción breve de lo que realizan.
2. Analizar y comprender las funciones que realizan cada uno de los objetos usados dentro de cada subsistema y en el modelo en general.
3. De qué formas se puede utilizar el **model browser** y cuál es su diferencia?
4. Cuáles son los objetos que permiten obtener las gráficas de la simulación? Qué señal es entregada por cada uno de ellos?
5. Explicar como configurar estos objetos para poder visualizar de mejor manera los resultados de diversas simulaciones.
6. Simular el modelo para un valor fijo de SNR y variar los valores de mapeo de acuerdo al estándar 802.11a (ver tabla).
7. Analizar y comparar las gráficas obtenidas y concluir como influye o cuáles son los efectos de aumentar el mapeo de la señal de datos.
8. Simular el modelo para un valor de mapeo fijo y probar para al menos cinco valores de SNR entre 1 y 100 dB.
9. Comparar los resultados obtenidos y concluir como afecta la manipulación de este parámetro para un mismo método de mapeo.

## ÍNDICE DE FIGURAS

Figura. 1.1. Proceso de Compresión-Decompresión de la Información Digital.....	2
Figura. 1.2. Pasos del Proceso de Conversión Analoga - Digital.....	4
Figura. 1.3. Señal Sinusoidal Continua .....	5
Figura. 1.4. Señal Sinusoidal Discreta .....	6
Figura. 1.5. Señal Base $X(f)$ .....	6
Figura. 1.6. Esquema de Muestreo Ideal .....	7
Figura. 1.7. Señal Muestreada en el Dominio de Frecuencia .....	8
Figura. 1.8. Muestreo Natural con Pulsos.....	10
Figura. 1.9. Señal Muestreada en el Dominio de Frecuencia .....	10
Figura. 1.10. Muestreo de Tope Plano.....	11
Figura. 1.11. Señal Pasabanda .....	12
Figura. 1.12. Señal Muestreada en el Dominio de Frecuencia .....	12
Figura. 1.13. Paso de Cuantización “a”.....	14
Figura. 1.14. Error de Cuantizacion.....	15
Figura. 1.15. Cuantización Uniforme .....	16
Figura. 1.16. Características de Compresión, (a) Ley U, (b) Ley A.....	17
Figura. 1.17. Cuantización No Uniforme .....	18
Figura. 1.18. Cuantización Vectorial en Dos Dimensiones .....	19
Figura. 1.19. Muestras Naturales, Muestras Cuantizadas y PCM (Codificadas) .....	22
Figura. 1.20. Sistema DPCM a) Codificador, b) Decodificador .....	25
Figura. 1.21. Codificador / Decodificador ADPCM.....	26
Figura. 1.22. Codificador / Decodificador Sub Bandas .....	27
Figura. 1.23. Modelo de Producción de Voz.....	29
Figura. 1.24. Vocoder LPC – 10.....	30
Figura. 1.25. Emisor LPC – 10 .....	31
Figura. 1.26. Receptor LPC – 10 .....	32
Figura. 1.27. Codificación REL.....	34

Figura. 1.28. (a) Señal original. (b) Residuo del filtro LPC (aumentado en 10dB) (c) Residuo de los filtros LPC y de pitch en cascada (aumentado en 10dB) .....	35
Figura. 1.29. Transmisor CELP .....	36
Figura. 1.30. Receptor CELP .....	37
Figura. 1.31. Receptor .....	38
Figura. 1.32. Proceso RPE – LTP .....	39
Figura. 1.33. Codificador RPE – LTP .....	40
Figura. 1.34. Decimacion RPE y Selección de Grid .....	42
Figura. 1.35. Decodificador RPE – LTP .....	43
Figura. 1.36. Propiedades de Señal de Voz: Nueve .....	45
Figura. 1.37. Espectro de la Palabra Nueve .....	45
Figura. 1.38. Espectro del Fonema N .....	46
Figura. 1.39. Autocorrelación Letra N.....	47
Figura. 1.40. Espectro del Fonema U.....	48
Figura. 1.41. Autocorrelación Letra U.....	48
Figura. 1.42. Espectro del Fonema E.....	49
Figura. 1.43. Autocorrelación Letra E .....	50
Figura. 1.44. Espectro del Fonema V.....	50
Figura. 1.45. Autocorrelación Letra V.....	51
Figura. 1.46. Señal de Voz, 11025 Hz, 16 bits: Nueve .....	54
Figura. 1.47. Ventana de Hanning para 120 Muestras .....	55
Figura. 1.48. Bloque de 120 Muestras de la Señal de Voz para el Primer Bloque de Proceso.....	55
Figura. 1.49. Señal de Voz (Primer Bloque) Aplicada una Ventana de Hanning .....	56
Figura. 1.50. Señal Estimada del Primer Bloque de Proceso .....	56
Figura. 1.51. Señal de Error $e(n)$ para el Primer Bloque de Proceso.....	57
Figura. 1.52. Señal de Estimación de la Señal Original .....	58
Figura. 1.53. Señal $e(n)$ a Ser Transmitida. ....	58
Figura. 1.54. Señal de Voz, 11025 Hz, 16 bits: Nueve .....	60
Figura. 1.55. Variación de la Tasa de Cruce por Cero por Bloques .....	61
Figura. 1.56. Esquema de Cruce por Cero de la Señal de Voz.....	62
Figura. 1.57. Tasa de Cruce por Cero de la Señal de Voz.....	63
Figura. 2.1. Diagrama de Bloques de una Señal PAM Digital.....	66
Figura. 2.2. Formas de Onda de los Diferentes códigos de Línea .....	74

Figura. 2.3. Densidad Espectral de Potencia de los Diferentes códigos de Línea, donde $R = 1 / T$ es la Tasa de Bit .....	76
Figura. 2.4. Densidad Espectral de Potencia de los Diferentes códigos de Línea, donde $R = 1 / T$ es la Tasa de Bit .....	77
Figura. 2.5. Probabilidad de Error de Bit para los Diferentes códigos de Línea .....	78
Figura. 2.6. Señalización Duobinaria .....	81
Figura. 2.7. Interferencia InterSimbólica en recepción de un Sistema Binario .....	84
Figura. 2.8. Sistema de Transmisión de Pulsos en Banda Base .....	84
Figura. 2.9. Pulso Ideal y mínimo Ancho de Banda de Nyquist .....	85
Figura. 2.10. Características del Filtro de Coseno Levantado para varios factores de Roll-Off.....	87
Figura. 2.11. Diagramas de Ojo para una Señal: a) Ideal, b) Con ISI, c) Con ISI y Ruido.	89
Figura. 2.12. Características de un Diagrama de Ojo .....	90
Figura. 2.13. Formas de Onda de Diferentes códigos de Línea.....	95
Figura. 2.14. PSD para Diferentes códigos de línea, Igual Secuencia.....	97
Figura. 2.15. Respuesta al Canal para Polar NRZ y Señal Original.....	100
Figura. 2.16. Degradación de Señal causada por AWGN .....	101
Figura. 2.17. Degradación de Señal Causada por Restricción de Ancho de Banda .....	102
Figura. 2.18. Diagrama de Ojo para Señal Unipolar RZ y Señal Considerada .....	103
Figura. 2.19. Diagrama de Ojo para Señal Degradada por Ruido .....	104
Figura. 2.20. Diagrama de Ojo para Señal Degradada por Reducción Ancho de Banda ..	105
Figura. 2.21. Diagrama de Ojo para Diferentes códigos de línea .....	106
Figura. 3.1. Varios Tipos de modulación Digital Pasabanda .....	111
Figura. 3.2. Receptor de correlación Binario. a) Un Correlator, b) Dos Correlatores.....	115
Figura. 3.3. Demodulador de Señales MPSK .....	116
Figura. 3.4. Regiones de Decisión para MFSK, $M=3$ (ideal) .....	118
Figura. 3.5. Espacio Ortogonal para DPSK.....	119
Figura. 3.6. Detector No Coherente FSK en Cuadratura .....	120
Figura. 3.7. Detector No Coherente de FSK.....	121
Figura. 3.8. Curva Ideal PB versus $E_b/N_0$ .....	123
Figura. 3.9. Onda FSK, a) Ideal, b) Descompuesta. ....	124
Figura. 3.10. Modulador de FSK.....	125
Figura. 3.11. Demodulador FSK con PLL.....	126
Figura. 3.12. Receptor Óptimo para M-aria Ortogonal .....	127

Figura. 3.13. Modulador BPSK .....	128
Figura. 3.14. Tabla de Verdad y Diagrama Fasorial de BPSK.....	129
Figura. 3.15. Demodulador BPSK.....	129
Figura. 3.16. Modulador QPSK.....	130
Figura. 3.17. Tabla de Verdad (a) y Diagrama fasorial (b) de QPSK .....	131
Figura. 3.18. Posibles Señales de Salida QPSK .....	131
Figura. 3.19. Demodulador QPSK.....	132
Figura. 3.20. Modulador OQPSK.....	132
Figura. 3.21. Alineación de Bits (b) y Constelación (c) de OQPSK.....	133
Figura. 3.22. Modulador 8PSK.....	134
Figura. 3.23. Convertidores de Nivel 2 a 4, a) Tabla de Verdad Canal I, b) Tabla de Verdad Canal Q, c) Niveles PAM .....	134
Figura. 3.24. Constelación de 8PSK.....	135
Figura. 3.25. Posibles Señales de Salida de 8PSK. ....	135
Figura. 3.26. Demodulador 8PSK.....	135
Figura. 3.27. Tabla de Verdad y Constelación 16PSK .....	136
Figura. 3.28. Modulador y relación de Señales DBPSK .....	137
Figura. 3.29. Demodulador y Secuencia de Tiempo DBPSK.....	137
Figura. 3.30. Onda Binaria ASK .....	138
Figura. 3.31. Detección de Filtro Apareado de ASK.....	139
Figura. 3.32. Tabla de Verdad y Diagramas ASK .....	139
Figura. 3.33. Modulador 8 QAM.....	140
Figura. 3.34. a) Tabla de Verdad, b) Fasores, c) constelación 8 QAM .....	141
Figura. 3.35. Posibles Fases de Salida 8 QAM.....	141
Figura. 3.36. Modulador 16 QAM.....	142
Figura. 3.37. Tablas de Verdad Convertidores de Nivel, a) Canal I, b) Canal Q .....	142
Figura. 3.38. a) Tabla de Verdad, b) Fasores, c) Constelación 16 QAM .....	143
Figura. 3.39. Forma de Onda de MSK.....	144
Figura. 3.40. Trama de Exceso de Fase MSK .....	145
Figura. 3.41. PSD de Envolvente Compleja de MSK, QPSK, OQPSK.....	147
Figura. 3.42. Modulador MSK .....	147
Figura. 3.43. Curvas de Probabilidad de Error en Sistemas Binarios.....	152
Figura. 3.44. Probabilidad de Error de Bit Coherentemente Detectada a) Señalización M-aria Ortogonal (MFSK), b) Señalización Fase Múltiple (MPSK) .....	153

Figura. 3.45. Probabilidad de Error de Bit para QAM .....	156
Figura. 3.46. Señal Modulada ASK, portadora 6 Khz.....	159
Figura. 3.47. Densidad Espectral de Potencia: Secuencia Binaria y Señal ASK .....	160
Figura. 3.48. Señal Modulada PSK, Portadora 6 KHz. ....	161
Figura. 3.49. Densidad Espectral de Potencia: Secuencia Binaria y Señal PSK.....	161
Figura. 3.50. Señal Modulada FSK, Marca: 8KHz, Espacio: 4 KHz .....	162
Figura. 3.51. Densidad Espectral de Potencia: Secuencia Binaria y Señal FSK.....	163
Figura. 3.52. Señal ASK Multiplicada por Oscilador de Recuperación y su PSD .....	166
Figura. 3.53. Señal PSK Multiplicada por Oscilador de Recuperación y su PSD.....	166
Figura. 3.54. Señal ASK y PSK a la Salida del Filtro Casado del Receptor .....	167
Figura. 3.55. Señal ASK con Errores de Fase en el Oscilador del Receptor .....	168
Figura. 3.56. Señal PSK con Errores de Fase en el Oscilador del Receptor .....	169
Figura. 3.57. Señal ASK con Desviación de Frecuencia del Oscilador en Recepción.....	170
Figura. 3.58. Detección No Coherente de Señal ASK.....	171
Figura. 3.59. Respuesta al Canal para Señal ASK.....	171
Figura. 3.60. Diagrama de Ojo para Señal ASK.....	172
Figura. 4.1. Señales Antipodales .....	179
Figura. 4.2. ARQ, a) Parada-Espera, b) Continuo, c) Continuo Selectivo .....	185
Figura. 4.3. Código de Paridad Singular (Serial).....	186
Figura. 4.4. Código de Paridad Rectangular (Paralela).....	187
Figura. 4.5. Código Entrelazado .....	203
Figura. 4.6. Esquema código Concatenado .....	204
Figura. 4.7. Codificador Convolutacional (4, 3, 5).....	206
Figura. 4.8. Codificador Convolutacional $N=3, v=2$ .....	207
Figura. 4.9a. Codificador Convolutacional (2, 1, 3) secuencia 0101 .....	208
Figura. 4.9b. Codificador Convolutacional (2, 1, 3) secuencia 0101 .....	208
Figura. 4.9c. Codificador Convolutacional (2, 1, 3) secuencia 0101 .....	208
Figura. 4.9d. Codificador Convolutacional (2, 1, 3) secuencia 0101 .....	209
Figura. 4.10. Diagrama de Estados Codificador (2, 1, 2) secuencia 101.....	210
Figura. 4.11. Árbol de código para Codificador (2, 1, 3) secuencia 0101.....	212
Figura. 4.12. Diagrama de Trellis Codificador (2, 1, 3) .....	213
Figura. 4.13. Diagrama de Trellis Codificador (2, 1, 3) secuencia 0101.....	213
Figura. 4.14a. Algoritmo Viterbi Codificador (2, 1, 3) .....	215
Figura. 4.14b. Algoritmo Viterbi Codificador (2, 1, 3) .....	216

Figura. 4.14c. Algoritmo Viterbi Codificador (2, 1, 3) .....	216
Figura. 4.14d. Algoritmo Viterbi Codificador (2, 1, 3) .....	217
Figura. 4.14e. Algoritmo Viterbi Codificador (2, 1, 3) .....	217
Figura. 4.14f. Algoritmo Viterbi Codificador (2, 1, 3) .....	218
Figura. 4.14g. Algoritmo Viterbi Codificador (2, 1, 3) .....	218
Figura. 4.15. Distancia más Corta para Codificador (2, 1, 3) .....	219
Figura. 4.16. Ráfaga de Errores .....	222
Figura. 4.17. Diagrama General de un Turbo Codificador .....	224
Figura. 4.15. BER de Turbo Código para CDMA2000 Para Varias Tasas de código Palabra de entrada de 1530 bits, BPSK y canal AWGN .....	226
Figura. 5.1. Jerarquías PDH .....	243
Figura. 5.2. Trama PDH Europea .....	244
Figura. 5.3. Portadora T – 1 .....	246
Figura. 5.4. Trama PDH Americana .....	247
Figura. 5.5. Trama SONET .....	247
Figura. 5.6. Synchronous Payload Envelope .....	248
Figura. 5.7. Multiplexado SDH .....	251
Figura. 5.8. Trama SDH .....	251
Figura. 5.9. Falta de Visibilidad de PDH .....	254
Figura. 5.10. Especificación de Líneas de Salida .....	254
Figura. 5.11. Como Viaja WDM en una Fibra Óptica .....	256
Figura. 5.12. Division del Espectro en Ventanas de WDM .....	258
Figura. 5.13. Integración de canales para WDM. ....	258
Figura. 5.14. Sistema de Comunicación Normal y Usando WDM .....	259
Figura. 5.15. Separación de Canales Según Recomendación ITU-R G.692 .....	260
Figura. 6.1. Sistema de Comunicación Spread Spectrum .....	267
Figura. 6.2. Efecto de Ensanchamiento de una Señal .....	268
Figura. 6.3. Efecto de De-ensanchamiento de una Señal .....	268
Figura. 6.4. Rendimiento en usuarios del Espectro Ensanchado .....	268
Figura. 6.5. Formación de la Señal Ensanchada .....	269
Figura. 6.6. Generación de una Secuencia de Pseudo-Ruido .....	270
Figura. 6.7. Generador Pseudo-Ruido .....	271
Figura. 6.8. Generación de Secuencias Gold .....	274
Figura. 6.9. Comparación de Secuencia M y Walsh en Dominio de la Frecuencia .....	275

Figura. 6.10. Diferentes Técnicas de SS .....	276
Figura. 6.11. Espectro Ensanchado en Secuencia Directa .....	277
Figura. 6.12. Demoduladores de DSSS .....	278
Figura. 6.13. Esquema de Frecuencias para FHSS .....	280
Figura. 6.14. Espectro Ensanchado en Salto de Frecuencias FHSS .....	280
Figura. 6.15. Salidas de LFHSS para 8FSK .....	282
Figura. 6.16. Esquema de Espectro Ensanchado DS/FFH.....	283
Figura. 6.17. Esquema de Espectro Ensanchado THSS .....	283
Figura. 6.18. Secuencia de Prueba de Sistema de Comunicación CDMA-SS .....	285
Figura. 6.19. Secuencias de Datos Origen y Destino para un Usuario .....	286
Figura. 6.20. Ruido Creciente en el Canal y Probabilidad de Error para un Usuario.....	287
Figura. 6.21. Secuencias de Datos Origen y Destino para Dos Usuarios .....	288
Figura. 6.22. Ruido Creciente en el Canal y Probabilidad de Error para Dos Usuarios.....	288
Figura. 6.23. Secuencias Origen y Destino para Dos Usuarios, Ruido Constante .....	289
Figura. 6.24. Señal Creciente del Segundo Usuario y Probabilidad de Error.....	290
Figura. 6.25. Secuencias Origen y Destino para Múltiples Usuarios .....	290
Figura. 6.24. Número Creciente de Usuarios y Probabilidad de Error .....	291
Figura. 7.1. Acceso Múltiple por División de Frecuencia .....	297
Figura. 7.2. Distribución de Usuarios en FDMA .....	298
Figura. 7.3. Acceso Múltiple por División de Tiempo .....	299
Figura. 7.4. Distribución de Usuarios en TDMA .....	299
Figura. 7.5. Esquema CDMA en el Tiempo .....	301
Figura. 7.6. Acceso Múltiple por División de Código .....	306
Figura. 7.7. Acceso Múltiple por División de Espacio .....	307
Figura. 7.8. Acceso Múltiple por División de Polaridad .....	308
Figura. 7.9. Modulación BPSK, $a= 0.4$ , Coherente, Filtro Casado y Detección de Errores.....	320
Figura. 7.10. Modulación BPSK, $a= 0.4$ , Diferencial, Filtro Pasabajos y Detección de Errores.....	321
Figura. 7.11. Modulación QPSK, $a= 0.6$ , Coherente, Filtro Casado y Detección de Errores.....	322
Figura. 7.12. Modulación QPSK, $a= 0.8$ , Diferencial, Filtro Pasabajos y Detección de Errores.....	323



Figura. 7.13. Modulación 8PSK, $a= 0.9$ , Coherente, Filtro Casado y Detección de Errores.....	323
Figura. 7.14. Modulación 8PSK, $a= 0.9$ , Diferencial, Filtro Pasabajos y Detección de Errores.....	324
Figura. 7.15. Modulación BPSK, $a= 0.4$ , Coherente, Filtro Casado y Sin detección de Errores.....	325
Figura. 7.16. Modulación QPSK, $a= 0.6$ , Coherente, Filtro Casado y Sin detección de Errores.....	326
Figura. 7.17. Modulación 8PSK, $a= 0.9$ , Coherente, Filtro Casado y Sin detección de Errores.....	327
Figura. 8.1. Esquema de Discrete MultiTone DMT.....	338
Figura. 8.2. Modulación ADSL DMT con FDM.....	339
Figura. 8.3. Modulación ADSL DMT con Cancelación de Ecos .....	340
Figura. 8.4. CAP Divide la Línea en Tres Bandas Espaciadas .....	343
Figura. 8.5. Esquema de ADSL CAP .....	343
Figura. 8.6. Transmisor CAP Utilizando Desplazamiento de Fase .....	344
Figura. 8.7. Modulación CAP <sub>4</sub> .....	345
Figura. 8.8. Codificador / Modulador Genérico TCM.....	347
Figura. 8.9. Particionamiento de Conjuntos para 8PSK.....	348
Figura. 8.10. Particionamiento de Conjuntos para 16QAM .....	349
Figura. 8.11. Diagrama de Bloques de un Sistema de modulación Codificada.....	350
Figura. 8.12. Esquema TCM Simple: a) Codificador, b) Trellis de Cuatro Estados, c) Mapeo de los Bits Codificados en los puntos de señal. ....	351
Figura. 8.13. Codificador Trellis de Ocho Estados para modulación Codificada 8PSK...	352
Figura. 8.14. Esquema de Bandas de Transmisión OFDM .....	353
Figura. 8.15. Modulación con Portadoras Ortogonales y Técnica Multiportadora Convencional .....	355
Figura. 8.16. OFDM vs. FDM .....	355
Figura. 8.17. Diagrama de Bloques General de Transmisión con OFDM .....	356
Figura. 8.18. Aplicación de Windowing (Coseno Levantado) para OFDM.....	356
Figura. 8.19. Etapas de Transmisión, Canal y Recepción para OFDM.....	357
Figura. 8.20. Distribución de Símbolos en el Dominio del Tiempo .....	358
Figura. 8.21. Receptor OFDM con Detección Coherente .....	359
Figura. 8.22. Tranceptor Usado en HiperLAN OFDM .....	359

Figura. 8.23. Vista del Modelo de Sistema de Comunicación con OFDM.....	364
Figura. 8.24. Subsistema Modelo de Sistema de Comunicación con OFDM .....	365
Figura. 8.25. Subsistema Bajo la Máscara del Normalizador.....	366
Figura. 8.26. Subsistema Bajo la Máscara del Transmisor OFDM .....	366
Figura. 8.27. Subsistema Bajo la Máscara de Zero Pad del Transmisor OFDM.....	367
Figura. 8.28. Subsistema Bajo la Máscara del Receptor OFDM.....	367
Figura. 8.29. Subsistema Bajo la Máscara del Denormalizador .....	368
Figura. 8.30. Ventana de Configuración Principal de Parámetros. ....	369
Figura. 8.31. Constelación de Señal Recibida $M= 4$ . ....	371
Figura. 8.32. Salida del Generador de Datos $M= 4$ , $SNR= 22$ . ....	372
Figura. 8.33. Salida del Modulador $M= 4$ , $SNR= 22$ . ....	372
Figura. 8.34. Salida del Transmisor OFDM $M= 4$ , $SNR= 22$ . ....	373
Figura. 8.35. Salida del Canal AWGN $M= 4$ , $SNR= 22$ . ....	373
Figura. 8.36. Salida denormalizada del Receptor OFDM $M= 4$ , $SNR= 22$ .....	374
Figura. 8.37. Salida del Demodulador $M= 4$ , $SNR= 22$ . ....	374
Figura. 8.38. Indicadores de Datos y Errores del Sistema. ....	375
Figura. 8.39. Constelación de Señal Recibida $M= 64$ . ....	375
Figura. 8.40. Salida del Generador de Datos $M= 64$ , $SNR= 50$ . ....	376
Figura. 8.41. Salida del Modulador $M= 64$ , $SNR= 50$ . ....	376
Figura. 8.42. Salida del Transmisor OFDM $M= 64$ , $SNR= 50$ . ....	377
Figura. 8.43. Salida del Canal AWGN $M= 64$ , $SNR= 50$ . ....	377
Figura. 8.44. Salida denormalizada del Receptor OFDM $M= 64$ , $SNR= 50$ .....	378
Figura. 8.45. Salida del Demodulador $M= 64$ , $SNR= 50$ . ....	378
Figura. 8.46. Indicadores de Datos y Errores del Sistema. ....	379
Figura. 8.47. Constelación de Señal Recibida $M= 64$ . ....	379
Figura. 8.48. Salida del Generador de Datos $M= 64$ , $SNR= 5$ . ....	380
Figura. 8.49. Salida del Modulador $M= 64$ , $SNR= 5$ . ....	380
Figura. 8.50. Salida del Transmisor OFDM $M= 64$ , $SNR= 5$ . ....	381
Figura. 8.51. Salida del Canal AWGN $M= 64$ , $SNR= 5$ . ....	381
Figura. 8.52. Salida denormalizada del Receptor OFDM $M= 64$ , $SNR= 5$ .....	382
Figura. 8.53. Salida del Demodulador $M= 64$ , $SNR= 5$ . ....	382
Figura. 8.54. Indicadores de Datos y Errores del Sistema. ....	383

## ÍNDICE DE TABLAS

Tabla. 1.1. Códigos Formados con Tres Bits.....	23
Tabla. 1.2. Tabla Resumen de la Tasa de Cruce por Cero para Cada Fonema.....	62
Tabla. 2.1. Ejemplo Señalización Duobinaria .....	82
Tabla. 2.2. Ejemplo Precodificación para Señalización Duobinaria .....	83
Tabla. 3.1. Función de Bessel.....	126
Tabla. 3.2. Modulación Digital.- Eficiencia. ....	148
Tabla. 3.3. Probabilidad de Error en Sistemas Binarios. ....	152
Tabla. 4.1. Código de Bloques Lineales (6,3) .....	188
Tabla. 4.2. Generadores de código BHC Primitivo. ....	196
Tabla. 4.3. Polinomios Generadores CRC .....	201
Tabla. 4.4. Secuencias de Salida para Codificador (2, 1, 3) .....	207
Tabla. 4.5. Secuencias de Salida por Niveles .....	219
Tabla. 5.1. Jerarquía Europea .....	244
Tabla. 5.2. Jerarquía Americana .....	245
Tabla. 5.3. Jerarquía Japonesa .....	245
Tabla. 5.4. Características de SONET .....	249
Tabla. 5.5. Comparación de las Tasas de STS y STM .....	252
Tabla. 5.6. Multiplexado SDH.....	252
Tabla. 5.7. Estructura de Multiplexión basada en G707.....	253
Tabla. 5.8. Comparación de Sistemas PDH y SDH y Capacidades G707.....	255
Tabla. 6.1. Generación de Secuencia Pseudo-Ruido .....	271
Tabla. 7.1. Comparación de Sistemas de Acceso Múltiple .....	309
Tabla. 7.2. SNR para Diferentes Iteraciones .....	313
Cuantización No Uniforme .....	313
Tabla. 7.3. SNR para Diferentes Iteraciones .....	315
Cuantización Diferencial.....	315
Tabla. 8.1. Valores de M válidos para Mapeo de Acuerdo al estándar IEE802.11a .....	369
Tabla. 8.2. Parámetros Establecidos como Comandos Iniciales del Sistema.....	370

Sangolquí, Octubre de 2005

## **HOJA DE LEGALIZACIÓN**

**ELABORADO POR:**

---

**Sr. Alex Paúl Espinoza Gutiérrez**

**AUTORIDADES:**

---

**TCRN. DE E.M. Ing. Xavier Martínez C.**  
**Decano de la Facultad de Ingeniería Electrónica**

---

**Ab. Jorge Carvajal**  
**Secretario Académico de la Facultad de Ingeniería Electrónica**