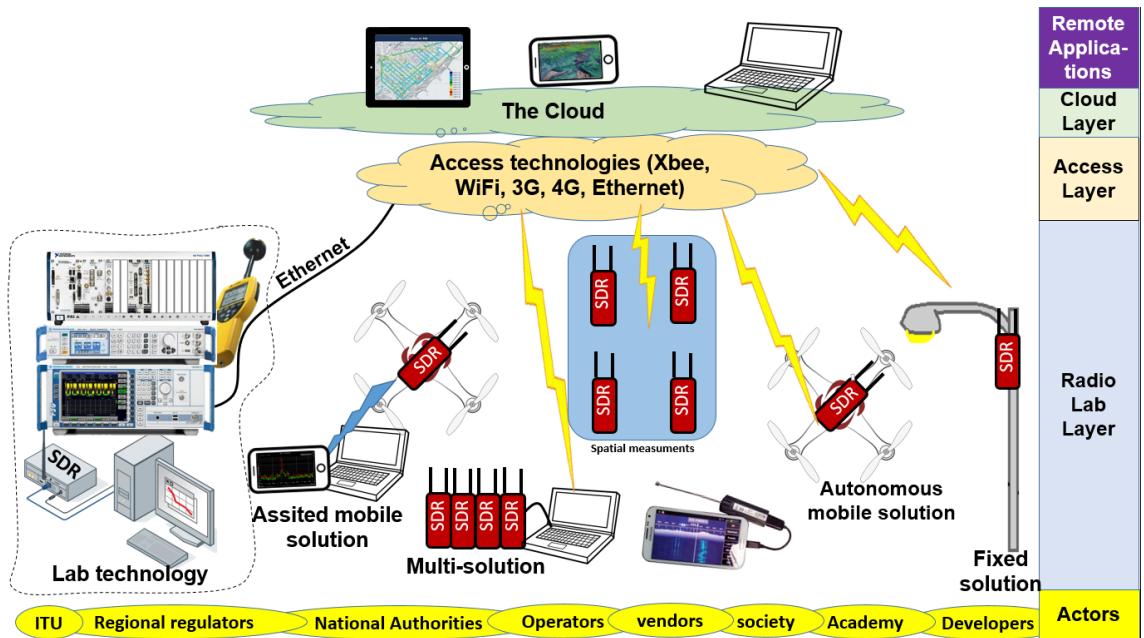


Comunicaciones Digitales basadas en radio definida por software.



ESCUELA DE INGENIERÍAS ELÉCTRICA ELECTRÓNICA Y TELECOMUNICACIONES–
E3T UNIVERSIDAD INDUSTRIAL DE SANTANDER – UIS



Homero Ortega Boada
Universidad Industrial de Santander
Escuela de Ingeniería Electrónica Eléctrica y de Telecomunicaciones
Bucaramanga
Teléfono: + 57 7 6344000 ext 2373
hortegab@uis.edu.co

Óscar Mauricio Reyes Torres
Universidad Industrial de Santander
Escuela de Ingeniería Electrónica Eléctrica y de Telecomunicaciones
Bucaramanga
Teléfono: + 57 7 6344000 ext
omreyes@uis.edu.co

Ilustraciones
Editorial UIS

Asesora Editorial
Publicaciones UIS

Primera edición Octubre 2019

Editorial UIS

Todos los derechos reservados

Comunicaciones Digitales basadas en radio definida por software.

Por
Homero Ortega Boada
Óscar Mauricio Reyes Torres

Página Web del libro:
<https://sites.google.com/saber.uis.edu.co/comdig>

Comunicaciones Digitales basadas en radio definida por software.

El conocimiento de las comunicaciones digitales desde la perspectiva más moderna

Homero Ortega Boada es Ingeniero Electrónico del Instituto de Aviación Civil de Kiev con Maestría y Doctorado en Ciencias de la Ingeniería de la Universidad Internacional de Aviación Civil de Kiev y conjuga un cúmulo de experiencias que se derivan de su paso de dos años por la empresa Ericsson , 9.5 en la Universidad Industrial de Santander donde ha sido líder de proyectos, docente investigador y director del Grupo de Investigación RadioGis, 2.5 años en la Agencia Nacional del Espectro, donde se desempeño como Asesor de Dirección General de la Agencia Nacional del Espectro, así como Director Interino de esa entidad y como Subdirector Interino de la Subdirección de Planeación y Gestión Técnica del Espectro. Allí apoyo el inicio de las actividades y la consolidación de la Agencia Nacional del Espectro, así como proyectos de socialización del conocimiento en temas de Gestión del Espectro Radioeléctrico, Estudios para asignación de espectro a proveedores de redes y servicios para las comunicaciones móviles e internet de banda ancha, el reordenamiento de las bandas de frecuencias para la entrada de la televisión digital y el aprovechamiento del Dividendo Digital, representando a Colombia en temas de Gestión Internacional con su participación en la Conferencia Mundial de Radiocomunicaciones del 2012 y en los Grupo de Radio de la UIT. Es autor de numerosas publicaciones científicas en revistas y congresos y de una patente en Radiocomunicaciones en temas de OFDM.

Óscar Mauricio Reyes Torres, Doctorado en Ingeniería (c) Technische Universität Hamburg-Harburg, TUHH. Alemania, Magíster en Ingeniera Electrónica de la Universidad Industrial de Santander, Especialista en Docencia Universitaria de la Universidad Industrial de Santander, Ingeniero Electrónico de la Universidad Industrial de Santander.

Tabla de contenido

INTRODUCCIÓN	15
1. Introducción a la programación en tiempo real en SDR	16
1.1. Introducción a SDR como un sistema de tiempo real que combina software y hardware	16
1.2. Programación de sistemas de tiempo real	16
2. Señales aleatorias y promedios de tiempos	22
2.1. Planteamiento del problema	22
2.2. Promedios de tiempo en señales reales	22
2.3. La media de una señal $x(t)$	23
2.4. La media cuadrática de una señal $x(t)$	23
2.5. Otros promedios de tiempo	23
2.6. El valor RMS	23
2.7. La potencia promedio de una señal $x(t)$	23
2.8. La función de Autocorrelación	24
2.9. La Función de Densidad de Probabilidad	24
2.10. La Función de Distribución Acumulativa	26
2.11. Promedios de tiempo en señales complejas	26
2.12. La Densidad Espectral de Potencia	27
2.13. Relación de la Densidad Espectral de Potencia con la función de Autocorrelación	28
2.14. Señal binaria bipolar aleatoria	28
2.15. El ruido blanco	32
2.16. La voz humana en la telefonía	33
2.17. Implementaciones en GNU Radio	33
2.17.1. Método de Welch para obtener la PSD	33
2.17.2. Implementación en GNU Radio de un analizador de la PSD	34
2.17.3. Implementación de un generador vectorial de señales básicas	36
2.17.4. Implementación de un bloque de visualización de los principales promedios de tiempo	42
3. La Conversión RF-EC. Aplicación en la radiodifusión	44
3.1. la Envolvente Compleja y sus características	44
3.1.1. El problema a resolver	44
3.1.2. La solución: la Envolvente Compleja	45
3.2. Conversión RF-EC en el dominio del tiempo	46
3.2.1. Conversión EC-RF en el dominio del tiempo	47
3.2.2. La Conversión RF-EC en el dominio de las frecuencias	47
3.2.3. Ejercicio sobre la Conversión RF-EC	47
3.3. El Down Converter	47
3.3.1. Demostración del Down Converter	48
3.3.2. Demostración gráfica del Down Converter	48
3.4. El Up Converter	49
3.5. Simulación en banda base de un sistema de radiodifusión	50
3.5.1. Modelo de capas e identificación de requerimientos	51
3.5.2. El Canal AWGN banda base	52
3.5.3. La Modulación FM Banda Base	55
3.5.4. Implementación de un Modulador FM mediante GNU Radio	58
3.5.5. El PLL como Demodulador FM	59

3.5.6. Implementación en GNU Radio de un Demodulador PLL	62
3.5.7. Discriminador de frecuencias con circuitos de pendiente	64
3.5.8. Discriminador de frecuencias con derivadores	65
3.5.9. Pruebas en GNU Radio del protocolo de modulación	66
3.5.10. Pruebas en GNU Radio de un Sistema FM con señales de audio	68
3.5.11. Influencia del ruido en la Modulación FM	71
3.5.12. Implementación final	74
4. La Radio Definida por Software. Aplicación en sistemas de radiodifusión	78
4.1. Integración de GNU Radio y SDR	78
4.2. Radio Definida por Software en Modo de Red	79
4.3. Un equipo SDR parara Modo de Red. El NI USRP-2920	81
4.3.1. El receptor del NI USRP-2920	82
4.3.2. El transmisor del NI USRP-2920	87
4.3.3. El papel de GPS en un NI USRP.	88
4.3.4. Los filtros en un NI USRP.	88
4.3.5. En la parte transmisora	89
4.3.6. Actividades de Caracterización del equipo	90
4.4. Cuidados con el uso de los equipos SDR	91
4.5. El Teorema de Nyquist aplicado en SDR	91
4.6. Equipo SDR para soluciones en modo embebido	92
4.7. Uso del USRP E310 en modo embebido	94
4.8. Un sistema de radiodifusión	96
4.8.1. La Conversión Análogo Digital	98
4.8.2. La Conversión Digital Análoga	99
4.9. Implementación de un sistema funcional de Radiodifusión FM basado en SDR	99
5. Modulación OOK, ASK, BPSK	103
6. Modulación FSK	104
7. Un sistema de radio comunicación digital sobre un canal guiado	105
7.1. Modelo de capas a seguir para la implementación	105
7.2. La modulación digital pasobandas y su versión bandabase	106
7.3. La modulación BPSK	107
7.4. Diferentes tipos de modulación basadas en constelaciones	110
7.4.1. La Modulación QPSK	110
7.4.2. La Modulación M-PSK y sus características distintivas	114
7.4.3. La Modulación M-QAM	116
7.4.4. Implementación en GNU Radio	118
7.5. Curvas de BER	122
7.5.1. El problema a resolver	123
7.5.2. La solución al problema	125
7.5.3. Implementación en GNU Radio	127
7.6. Formación de Pulses y Acoplamiento en comunicaciones Digitales inalámbricas	131
7.7. Consecuencias de otros Fenómenos	136
7.7.1. Desviación angular	136
7.7.2. Desviación en frecuencia	138
7.7.3. Frequency Lock Loop (FLL) y Phase Local Loop (PLL)	139
7.7.4. La codificación Diferencial para atacar la ambigüedad de fase	141
7.8. Implementación en GNU Radio de la capa Pre-Canal	145
7.9. Problemas y preguntas de control	146
8. El canal inalámbrico	151
8.1. Consecuencias del Fenómeno de Multirayectoria	151
8.1.1. El Jitter	151
8.1.2. Desvanecimientos lentos y rápidos. Efecto de Rayleigh	152
8.1.3. Afectación por no linealidades	153
8.2. Simulación de un canal inalámbrico en GNU Radio	154
8.3. Técnicas para corregir los efectos del canal	155
8.3.1. Clock Recovery o Timing	155
8.3.2. Alineación de tiempo	158

8.3.3. El Control Automático de la Ganancia	162
8.3.4. El ecualizador	163
8.3.5. La ecualización en GNU Radio	170
8.4. Problemas y preguntas de control	171
9. Comunicaciones multiusuario	173
9.1. Introducción	173
9.2. Multiplexación por división de tiempo TDM. La revolución PCM	173
9.3. MULTIACCESO POR DIVISIÓN DE FRECUENCIAS (FDMA)	177
9.3.1. Problema que justifica esta tecnología:	177
9.4. El Multiacceso por División de Tiempo (TDMA)	178
9.5. Técnicas adicionales usadas en las comunicaciones móviles	178
9.5.1. Técnicas o esquemas de diversidad	178
9.5.2. Diversidad de tiempo. <i>Interleaving</i> , FEC	179
9.6. Diversidad de espacio	180
9.6.1. Diversidad de polarización	181
9.6.2. Diversidad de frecuencias	181
9.6.3. Diversidad de multi usuario	182
9.6.4. Diversidad cooperativa	182
9.6.5. Diversidad de códigos	182
9.7. La Duplexación	182
9.7.1. La celda y la radiobase	184
9.7.2. El reuso	184
9.7.3. El handoff	184
9.8. Spread Spectrum (SS) y el Multiacceso por División de Códigos (CDMA)	184
9.8.1. Tipos de Spread Spectrum	188
9.8.2. El Multiacceso por División de códigos (CDMA)	189
9.9. Multiacceso por División de Códigos de banda ancha (WCDMA)	189
9.9.1. Convergencia de las comunicaciones	189
9.9.2. El UMTS	190
9.9.3. IMT-2000	190
9.9.4. WCDMA:	190
9.9.5. El reúso en UMTS	194
9.9.6. Respiración de las celdas en UMTS	195
9.10. OFDM – OFDMA	195
9.10.1. Realización práctica de OFDM	196
10. Codificación	201
10.1. Introducción	201
10.2. Conceptos previos	201
10.2.1. Polinomio generador de código	201
10.3. Aritmética de módulo 2	201
10.3.1. Campos de Galois	202
10.4. Fundamentos de codificación del canal	202
10.4.1. Teorema de Shannon	202
10.4.2. El proceso de codificación y decodificación digital del canal	202
10.4.3. Los retos de la teoría de codificación del canal	203
10.5. Códigos de Repetición	203
10.5.1. El Código de Repetición (3,1)	203
10.5.2. El Código de repetición basado en una matriz 2x3	204
10.6. Principales parámetros de la codificación	204
10.6.1. Distancia de Hamming entre dos palabras	204
10.6.2. Distancia de Hamming de un código	205
10.6.3. La rata del código.	206
10.6.4. La ganancia del código	207
10.7. Clasificación de los códigos según sus características	207
10.7.1. Códigos lineales	207
10.7.2. Los códigos de bloques	207
10.7.3. Los códigos convolucionales	207
10.7.4. Códigos binarios versus no binarios.	208
10.8. Técnicas de detección de errores	208

10.8.1. Chequeo de repeticiones.	208
10.8.2. Chequeo de paridad (Parity Check).	208
10.8.3. Chequeo de polaridad	209
10.9. El mecanismo de corrección de Errores	209
10.9.1. Solicitud Automática de Repetición (ARQ)	209
10.9.2. Corrección de Errores hacia adelante (FEC)	209
10.10 Propiedades para la detección y corrección de Errores	209
10.11 Códigos de redundancia cíclica (CRC)	210
10.11.1. Proceso de codificación usando CRC	210
10.11.2. Proceso de detección de errores usando Check sum	211
10.11.3. Cualidades del CRC	211
10.12 La codificación de Reed Solomon	211
10.13 Reed Solomon RS(255,223)en la práctica	213
10.14 Modelo de capas de DVB-T	214
10.15 Scrambling en DVB-T	214
10.16 FEC en DVB-T	215
11. Variables comúnmente usadas en los flujogramas que acompañan a este libro	216
Bibliografía	219

Índice de figuras

1.	Señal tipo stream.	17
2.	Ejemplo de la lista de OOT que se ofrece en el GRC.	18
3.	Diagrama para evaluar el bloque de promedios de tiempos.	21
1.	Señal truncada.	23
2.	Ejemplo de la Función de Densidad de Probabilidad.	25
3.	Ejemplo de un histograma.	26
4.	Ejemplo de una señal binaria aleatoria bipolar $x(t)$	29
5.	El desplazamiento de la señal binaria aleatoria bipolar que resulta en la señal $\tau < T_b$	29
6.	Multiplicación de la señal $x(t)$ con su versión desplazada $x(t + \tau)$ cuando $0 < \tau < T_b$	30
7.	Multiplicación de la señal $x(t)$ con su versión desplazada $x(t + \tau)$ cuando $\tau = T_b$	30
8.	La función de Autocorrelación de una señal binaria aleatoria bipolar.	31
9.	Forma de $R_{X_1}(\tau)$. Error: En el eje del tiempo: 1) debe mostrar flecha con la letra τ ; 2) a la izquierda del rectángulo debe aparecer $-T_b/2$, a la derecha $T_b/2$	31
10.	$R_{X_1}(\tau)$ y su Transformada de Fourier.	31
11.	PSD de la forma forma de $R_{X_1}(\tau)$	32
12.	PSD del ruido blanco y su función de Autocorrelación.	33
13.	Ejemplo de periodograma.	34
14.	Comparación del bloque FFT con <i>QT GUI Frequency Sink</i> para observar la PSD de una señal binaria bipolar aleatoria	34
15.	PSD obtenida por dos métodos	35
16.	Flujograma del bloque jerárquico “QT GUI b_PSD”	35
17.	Flujograma de un generador vectorial básico.	36
18.	Flujograma para una herramienta de visualización de promedios de tiempo.	37
19.	Promedios de tiempo. Señal de ruido blanco gausiano.	38
20.	Promedios de tiempo. Señal binaria aleatoria con código de línea tipo RZ.	39
21.	Promedios de tiempo. Señal binaria aleatoria con pulsos de forma sinc.	40
22.	Promedios de tiempo. Señal binaria aleatoria con pulsos de forma diente de sierra.	41
1.	Analizador de espectros basado en computador	45
2.	Espectro de la señal pasobandas de interés	45
3.	Espectro de la Envoltorio Compleja	46
4.	La Conversión RF-EC en el dominio de las frecuencias	47
5.	El Down Converter	48
6.	Resultado del uso del Down Converter	49
7.	Up converter	50
8.	Modelo de capas para la simulación de un sistema de radiodifusión	51
9.	Simulación en GNU Radio de la Envoltorio Compleja de ruido blanco	53
10.	Ruido Blanco Bandabase en el Diagrama Polar	54
11.	Ruido Blanco Banda Base, su PSD y representación Polar	54
12.	Implementación en GNU Radio de un Canal de ruido blanco aditivo	55
13.	Modelo de capas simplificado para implementar la modulación	55
14.	Relación entre las Funciones de Bessel y el espectro de una señal FM con mensaje sinusoidal	57
15.	Flujograma para comprobar la Modulación FM	58
16.	PSD de una señal con Modulación FM banda base y con mensaje sinusoidal	59
17.	El Bucle de Seguimiento de Fase - PLL	60
18.	Términos relacionados con ancho de banda en los PLL	61

19.	El Bucle de Seguimiento de Fase - PLL en tiempo discreto	61
20.	El error de fase	62
21.	Flujograma que muestra la capa de Modulación FM con un PLL como Demodulador	63
22.	Resultados de la Demodulación FM con PLL	63
23.	Respuesta en Frecuencia de un Circuito de pendiente	64
24.	Flujograma del bloque jerárquico “b_discriminador_frec_cf”	65
25.	Discriminador de frecuencias con derivadores	66
26.	Flujograma para comprobar la capa de modulación con diferentes métodos de demodulación	67
27.	Resultados comparativos de la demodulación con tres métodos	68
28.	Flujograma para un sistema de comunicaciones FM con señales de audio	69
29.	Resultados de un sistema de comunicaciones sonora FM	70
30.	Esquema para el análisis del efecto del ruido en un sistema de comunicaciones FM	71
31.	Una señal FM afectada por ruido blanco	72
32.	Ánalysis en el plano polar sobre el efecto del ruido en FM	72
33.	Análisis de ruido del receptor de FM	74
34.	Modelo de capas para ubicar la operación de énfasis de-énfasis	75
35.	Flujograma de un sistema completo de comunicaciones FM	75
36.	Resultados del sistema de comunicación FM	76
1.	Modelo de GNU Radio integrado con SDR	78
2.	Un sistema de comunicaciones punto a punto Full Duplex.	79
3.	Un transceptor (en inglés Transceiver).	80
4.	Modelo de capas de un sistema de comunicación con USRP en Modo Network.	80
5.	Foto del NI USRP-2920.	81
6.	Modelo de capas del NI USRP 29xx visto por capas. En color naranja: bloques de software embebido. En color verde: Hardware con parámetros controlables por computador. Blanco: Hardware no controlable.	82
7.	Señal a la entrada del cuantizador.	83
8.	Señal a la salida del cuantizador.	83
9.	Ancho de banda real que puede alcanzar el NI USRP 2920 con respecto a las capacidades de la tarjeta hija	84
10.	Ánalysis en banda base de frecuencias de muestreo y anchos de banda que se manejan en el USRP 2920 en el modo de recepción.	85
11.	Tabla sobre un análisis de valores de frecuencia de muestreo que acepta el USRP 2920	86
12.	Tabla de valores de frecuencia de muestreo que acepta el USRP 2920	87
13.	Elementos internos del USRP para la transmisión.	87
14.	Banco de filtros de la parte receptora del NI USRP E-310.	88
15.	Banco de filtros de la parte transmisora del NI USRP E-310.	90
16.	Montaje experimental para encontrar la relación entre la ganancia del UHD y la potencia de salida del dispositivo E310.	93
17.	Relación existente entre la ganancia del UHD y la potencia medida con el analizador de espectros.	93
18.	Modo sin interfaces gráficas.	94
19.	Flujograma en modo No GUI. (flujograma usado <i>flujograma_modo_embebido.grc</i>)	95
20.	192.168.0.100 dirección IP de ejemplo para el equipo SDR E310 luego de ser descubierta por arp-scan	95
21.	Lectura del espectro a través de UDP. (flujograma usado <i>flujograma_pc_remoto.grc</i>)	96
22.	Modelo de capas de un sistema de radiodifusión analógico implementado con tecnología digital	97
23.	Comparación entre una señal continua y una cuantificada a la razón de 3 bits/muestra	99
24.	Modelo para un sistema de radiodifusión FM	100
25.	Flujograma de un sistema de radiodifusión FM.	101
1.	Modelo de capas para una comunicación digital de voz por RF sobre un medio guiado.	105
2.	Diagrama de constelaciones BPSK.	108
3.	Esquema señal paso banda.	109
4.	PSD de una señal BPSK, con $R_s=1,5625$ kBaud, $R_b=1,5625$ Kbps.	110
5.	Constelación para la Modulación QPSK.	111
6.	Tablas de conversión de Binario a Gray.	112
7.	Constelación y Tabla de Verdad para la Modulación QPSK.	112
8.	Constelación y Tabla de Verdad para la Modulación QPSK con giro.	113
9.	Ejemplo de la Modulación QPSK en el dominio del tiempo.	113

10.	PSD para una señal QPSK.	114
11.	Constelación y Tabla de Verdad para la Modulación 8PSK.	114
12.	Modulación 8PSK. Ejemplo en el dominio del tiempo.	115
13.	PSD de para una señal 8PSK.	115
14.	Constelación y Tabla de Verdad para la Modulación 16QAM.	117
15.	Ejemplo de una señal 16QAM en el dominio del tiempo	117
16.	16-QAM en el dominio de las frecuencias	118
17.	Implementación en grc de GNU Radio de un modulador 16QAM.	119
18.	El bloque <code>Constellation Object</code>	121
19.	Ejemplo de la modulación 16QAM implementada en GNU Radio.	122
20.	PSD de una señal BPSK.	123
21.	PSD de una señal QPSK.	123
22.	PSD de una señal 8PSK.	123
23.	Comparación de BPSK, QPSK, 8PSK y 16QAM ante el ruido. Caso1	124
24.	Comparación de BPSK, QPSK, 8PSK y 16QAM ante el ruido. Caso2	125
25.	Curvas de BPSK, QPSK y 8PSK y 16PSK	126
26.	Curvas de BER Diversas Modulaciones.	127
27.	Flujograma para medir símbolos perdidos y SER	128
28.	Flujograma para obtener la Curva de BER para cualquier tipo de modulación.	130
29.	Curvas de BER obtenidas con GNU Radio.	131
30.	El Formador de Pulsos en una solución SDR.	131
31.	Ejemplo de la señal que entrega el modulador BPSK.	132
32.	El <code>Wave Forming</code> como intermediario entre el Modulador y el DAC.	132
33.	En línea azul la componente real y en rojo la señal a la salida del bloque <code>Wave Forming</code> . En negro la salida del DAC. <code>Sps=4</code>	132
34.	Envolvente Compleja con forma rectangular para la Modulación 8PSK.	133
35.	Flujograma con Modulación Digital y Formador de Pulsos.	133
36.	Resultados de <code>Wave Forming</code> aplicado a la modulación QPSK, con Filtro RC y <code>rolloff=0.5</code> .	134
37.	Resultados de <code>Wave Forming</code> aplicado a la modulación 8PSK, con Filtro RC y <code>rolloff = 1</code> .	135
38.	Diagrama de Ojo de una señal con Modulación 8PSK, con Filtro RC, <code>rolloff=1</code>	135
39.	Señal BPSK y su versión desfasada en 45 grados.	136
40.	Señal BPSK y su versión desfasada en 90 grados.	137
41.	Constelación de la Modulación 16QAM y su versión con desviación de fase.	138
42.	Desviación Frecuencias entre dos señales BPSK.	139
43.	Frequency Lock Loop.	139
44.	Resultados del sistema de comunicación FM.	140
45.	PLL para cuando la entrada es la Envoltura Compleja.	140
46.	Antes y después de aplicar FLL.	141
47.	Flujograma para implementar la capa base que incluye codificación diferencial y modulación con EC.	143
48.	PSD de para una señal super super QPSK	144
49.	Flujograma para implementar las técnicas pre-canal.	145
50.	Constelación desconocida	146
1.	Red de frecuencias únicas en TDT.	151
2.	El Jitter.	152
3.	Consecuencias en la constelación del desvanecimiento rápido.	152
4.	Consecuencias en la constelación del desvanecimiento lento.	153
5.	Variaciones espectrales debido a no linealidades del canal.	153
6.	QPSK impactada por no linealidades del canal.	154
7.	Waterfall para una señal QPSK afectada por no linealidades variables del canal.	154
8.	Flujograma para el canal inalámbrico.	155
9.	Timing.	156
10.	Punto óptimo.	157
11.	Punto óptimo no aparece dentro de las muestras recibidas.	157
12.	Búsqueda del punto óptimo.	158
13.	Constelación antes y después del timing.	158
14.	Flujograma de un sistema con modulación PCM sin Alineación de tiempo.	159
15.	Observación de las señales cuando no se presentan retardos en el canal.	160
16.	Flujograma de un sistema con modulación PCM y con Alineación de tiempo.	161
17.	Trama de Alineación de tiempo.	161

18.	Observación de las señales cuando se aplica Alineación de Tiempo.	162
19.	Elemento de área donde puede localizarse el receptor.	162
20.	AGC según Tom Rondau.	163
21.	Variaciones espectrales del canal (rojo) versus respuesta en frecuencias del ecualizador.	163
22.	Sistema LIT.	164
23.	Respuesta al impulso de un ecualizador.	164
24.	Proceso de entrenamiento del ecualizador.	165
25.	El ecualizador funcionando con pesos obtenidos durante el entrenamiento.	165
26.	Ecualización adaptativo.	166
27.	Respuesta variante al Impulso de un ecualizador.	166
28.	Filtro adaptativo.	167
29.	El mínimo de la media cuadrática.	168
30.	Resultado del la Ecualización. A la izquierda sin ecualización. A la derecha con ecualización.	171
1.	Central local	173
2.	Central de transito	174
3.	Ejemplo de TDM	175
4.	Teléfono digital	176
5.	Modelo con las redes telefónicas	176
6.	Red SDH	177
7.	Ejemplo de multiacceso	177
8.	Ejemplo de multiacceso	178
9.	Ejemplos de la diversidad de tiempo utilizando el intercalado.	179
10.	Diversidad de espacio	180
11.	Diversidad de espacio	181
12.	Ejemplo de diversidad de polarización	181
13.	Múltiples tuberías de capa física (PLP)	182
14.	Banda 1900 como posibilidad de ser aprovechada por duplexación por división de tiempo TDD.	183
15.	Resumen de FDMA-FD, TDMA-FDD, FDMA-TDD.	183
16.	La celda y la radiobase	184
17.	Spread Spectrum	185
18.	Sistema equivalente en banda base.	185
19.	Señales de ejemplo.	186
20.	PSD de las señales.	187
21.	Peor de los casos de la señal $i(t)$	188
22.	Señal $b(t)$	188
23.	Distribucion del espectro con CDMA.	189
24.	Redes clásicas y redes de nueva generación.	190
25.	Funciones de Walsh.	191
26.	Ejercicios funciones de Walsh.	192
27.	Sistemas de comunicaciones.	192
28.	Ejemplo $m_2(t)$	193
29.	Distribución del espectro combinando TDMA con CDMA.	194
30.	Cluster de N celdas.	194
31.	El reúso en UMTS	195
32.	OFDM	196
33.	sistema LTE (<i>Long Term Evolution</i>)	196
34.	Paso del primer símbolo por los bloques del multiplexor.	197
35.	Paso del segundo símbolo por los bloques del multiplexor.	198
36.	Paso del N símbolo hasta llegar al bloque “Conversor Discreto/Continuo”.	198
37.	Efecto de Aliasing.	199
38.	Efecto aliasing.	200
1.	Ruta de recepción de un mensaje.	202
2.	Proceso de codificación	203
3.	Comparación entre dos palabras.	205
4.	El espacio F_2^6 Los puntos negros representan las posibles palabras que pueden resultar en el campo y los círculos representan las palabras del código	206
5.	Rata del código.	206
6.	Código de bloques.	207

7.	Codificación de Reed Solomon.	212
8.	Modelo de capas de la capa física de dvb-t.	214
9.	Esquema de scrambling	215

Índice de tablas

4.1. Características generales del equipo SDR.	92
7.1. Plantilla para la Tabla de Verdad.	107
7.2. Tabla de Verdad de la Modulación BPSK	108
7.3. Tabla de Verdad de la Modulación BPSK. versión 2	108
7.4. Tabla de Verdad 16QAM.	116
7.5. Tabla de Verdad	147
8.1. Secuencia a la salida del transmisor.	159
8.2. Secuencia interpretada de manera errónea debido al retardo introducido por el canal.	159
10.1. Tabla de verdad. Código de repetición (3,1)	203
10.2. Tabla de verdad. Código de repetición (3,1) desde la perspectiva del receptor	204
10.3. Ejemplo de código de redundancia cíclica.	210

Nomenclatura de uso común en este libro

Abreviaturas

<i>SDR</i>	Software define radio
<i>UHD</i>	USRP Hardware Driver
<i>USRP</i>	Universal Software Radio Peripheral

Acrónimos

<i>SDR</i>	Software define radio
<i>UHD</i>	USRP Hardware Driver
<i>USRP</i>	Universal Software Radio Peripheral

Letras Griegas

λ	Longitud de onda
-----------	------------------

Siglas

<i>SDR</i>	Software define radio
<i>UHD</i>	USRP Hardware Driver
<i>USRP</i>	Universal Software Radio Peripheral

Símbolos

<i>SDR</i>	Software define radio
<i>UHD</i>	USRP Hardware Driver
<i>USRP</i>	Universal Software Radio Peripheral

INTRODUCCIÓN

Las comunicaciones han sido durante todos los tiempos la clave para el desarrollo social de la humanidad. Desde las señales de humo como medio de comunicación, pasando por la invención de la escritura, la imprenta, hasta llegar al teléfono, la televisión y el internet. Cada uno de esos avances ha impulsado un nuevo paso del hombre por diferentes etapas de desarrollo como la sociedad agraria, la sociedad industrial y la sociedad de la información. Por eso, varios autores aseguran que la historia de las comunicaciones es la historia de las revoluciones sociales. El nuevo paradigma es hoy la convergencia de las comunicaciones, es decir, la unión de: los medios de comunicación; la robótica; las redes de sensores para medir diferentes variables y fenómenos; el procesamiento inteligente de la información para la toma de decisiones; la informática; elementos normativos, de negocios y sociales. Todo lo anterior al servicio de las actividades humanas, como por ejemplo: la medicina, la industria y la conservación del medio ambiente. El concepto que aglutina todos esos componentes es hoy “*Tecnologías de la Información y las Comunicaciones*” (TIC). La ola de fusiones estratégicas a nivel empresarial así como los acontecimientos mundiales que se relacionan con la globalización del mundo son consecuencia de las TIC.

El auge de los sistemas inalámbricos y el desarrollo de más y mejores servicios de última generación en comunicaciones exige a los profesionales del área estar a la vanguardia de estos avances científicos y tecnológicos. Ante esta realidad, este curso busca crear las competencias necesarias para que los ingenieros electrónicos egresados de la UIS puedan fácilmente comprender las telecomunicaciones no sólo al nivel que requieren los operadores y proveedores de tecnologías, sino también al nivel científico. Abordar todos los temas de las comunicaciones en este curso es prácticamente imposible, pero sí se enfatiza en las competencias clave para lograr que los ingenieros de la UIS tengan una ventaja teórico-práctica frente a otros egresados para facilitar su participación en la demanda nacional e internacional en el campo de las comunicaciones. Son temas de estudio de este curso: los sistemas de comunicaciones digitales, analizando diferentes sistemas de modulación digital pasobanda y bandabase actuales y profundizando en las técnicas de acceso al medio más importantes hoy en día. De igual forma se estudian diversas aplicaciones de uso actual, sus principios y fundamentos necesarios para comprenderlas y analizarlas. Se hace uso de la programación de sistemas basados en SDR usando GRC y USRP como enfoque práctico de estos temas. Aunque se abordan muy débilmente otros temas relevantes como planeación de redes, mercados de las TIC, normativa y gestión del espectro, puede decirse que con este curso el estudiante habrá desarrollado las competencias que mayor esfuerzo intelectual y mayor acompañamiento requieren, lo cual habrá pavimentado el escabroso camino teórico-práctico para poder avanzar de manera más cómoda en la construcción de las competencias que demanda el mercado.

Si nos preguntaran en qué se diferencia este libro de todos los demás, la respuesta estaría en los siguientes puntos:

- Combina la enseñanza de las comunicaciones con la de GNU Radio.
- Ha sido escrito como parte de un proceso real de enseñanza, de modo que está probado en varias generaciones de estudiantes, ha crecido con ellas.
- Está orientado al desarrollo de competencias mediante prácticas de laboratorio.
- Se orienta hacia una enseñanza problematizada, de modo que para cada tema se propone un problema a resolver, se analizan sus causas y se presenta el tema como la solución a ese problema.
- Se usa una enseñanza siempre basada en modelos de capas.

Capítulo 1

Introducción a la programación en tiempo real en SDR

1.1. Introducción a SDR como un sistema de tiempo real que combina software y hardware

1.2. Programación de sistemas de tiempo real

En esta introducción del libro se ha querido hablar de la programación con GNU Radio y que mejor momento para hablar de programación de sistemas de tiempo real. Se trata de un área especial de la programación donde se trabaja con señales que tienen sentido dentro de un corto periodo de tiempo. Es precisamente el caso de los sistemas de comunicaciones. Cualquiera que sea el lenguaje de programación, cuando se trabaja con sistemas de tiempo real, la programación debe ser suficientemente eficiente como para que el sistema de comunicación no se congestione de operación que hagan colapsar el sistema. Es quizás por esa razón que existen herramientas especializadas en este tipo de programación como son: GNU Radio, labview, las herramientas de proveedores de tecnología como es el caso de KeySight.

En la programación en tiempo real es importante aprender a trabajar con señales, pues son las que necesitan desplazarse entre los bloques en tiempo real. El término señal lo usamos aquí en un sentido similar al que se usa en simulink. Se refiere aquello que un bloque de GNU Radio transfiere a otro bloque GNU Radio para entregarle información. Entonces, se trata de la información pero también la manera en que ella se entrega. Desde este punto de vista hay dos tipos de señales:

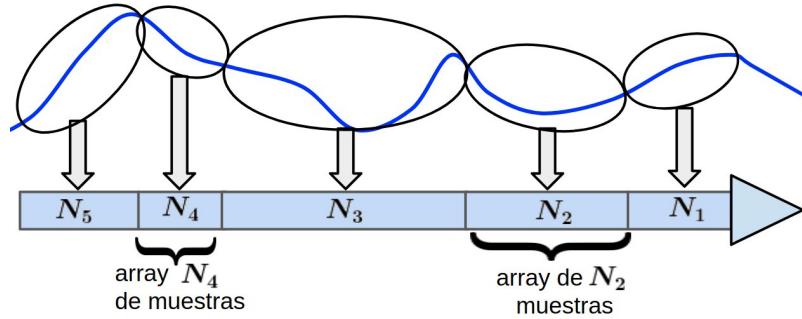
- **Las tipo stream:** La idea con este tipo de señales, es que el usuario de un bloque, tenga la ilusión de estar trabajando con un bloque con el que no debe preocuparse por empaquetar las muestras y consecuentemente no debe pensar en definir un tamaño para esos paquetes.
- **Las tipo vector:** Supongamos que un bloque no puede funcionar si no tiene N valores en la entrada. Es el caso de la FFT, si no tiene N muestras listas para ser procesadas no puede hacer nada. En el mundo físico esto equivale a los bloques que necesitan varias señales de entrada en paralelo. Entonces, el usuario debe preocuparse por definir el tamaño N de cada trama que desea sea procesada de esa manera. Lo resulta es una señal vectorial

GNU Radio impone un formato orientado a cumplir los fines de programación de sistemas de tiempo real. En ese formato, las señales nunca llegan a un bloque muestra por muestra, más bien por arrays. En este sentido las reglas son:

- Una señal de tipo stream puede ser vista como un array de una dimensión, solo que el tamaño del array puede ir cambiando como se muestra en la figura 1
- En un instante, la señal que entra a un bloque con entrada tipo stream estará definida por:

$$in = [x_0, x_1, x_{N_k-1}]$$

Figura 1: Señal tipo stream.



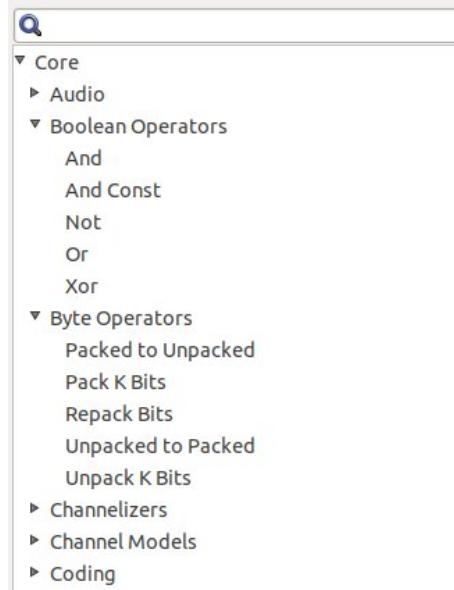
- Otro importante aspecto a destacar es la frecuencia de muestreo. En GNU Radio no sucede como en otros lenguajes como el que se ofrece en la herramienta de Simulink de Matlab, donde cada señal lleva información más completa sobre esa señal, por ejemplo lleva la información sobre la frecuencia de muestreo. En GNU Radio, cuando la señal se relaciona con el mundo real, por ejemplo cuando entra de un micrófono, cuando sale por una bocina, cuando se entrega o se recibe de un USRP, es impuesta por esos equipos que mantienen la relación con el mundo real. Claro está, que es posible configurar esos equipos para que reciban una u otra frecuencia de muestreo. Si no se tiene uno de esos equipos, entonces la velocidad con que corren las señales las impone el sistema y se desconoce a menos que incluyamos un bloque que se llama Throttle para que marque el ritmo en función de nuestras necesidades que pueden ser las del mundo real. Pueden haber diferentes frecuencias de muestreo en un flujo de datos, por ejemplo la señal que entrega un micrófono, luego de pasar por bloques moduladores y demás puede resultar en una señal con una mayor o menor frecuencia de muestreo. La recomendación para conocer la frecuencia de muestreo en cualquier parte de un flujo de datos y configurar correctamente los equipos de visualización es la siguiente: si el flujo usa un equipo físico, como un USRP, asigne a este el parámetro *samp_rate* u otro que sea de su preferencia, para configurar la frecuencia de muestreo de ese equipo. A partir de ese equipo vaya desplazándose bloque a bloque hacia el punto de interés, en cada bloque deténgase a analizar si este realiza una de estas dos operaciones y en qué grado: la interpolación o la diezma. Con estos datos realice cálculos para conocer la frecuencia de muestreo en cualquier punto. Los bloques que incluyen esas operaciones pueden ser muchos, pero usualmente ellos incluyen un parámetro que se llama “decimation” o “interpolation”, si ambos parámetros son igual a 1, significa que el bloque no aplica esas operaciones. El bloque favorito para convertir la frecuencia de muestreo de una señal de un valor a otro es el “Rational Resampler”.

Se resalta que principalmente se usarán señales tipo stream en las implementaciones de GNU Radio por ello no hemos entrado en tantos detalles con las señales tipo vector.

1.2.0.1. Ejemplos de implementación de bloques tipo stream y tipo vector en un OOT

GNU Radio es un kit de desarrollo que provee bloques de procesamiento listos para ser usados de manera amigable, ellos se organizan en una especie de sublibrerías que tienen bloques comunes. Por ejemplo, en la figura 2 se tienen las OOT “Boolean Operators” y la “Byte Operators” con sus bloques.

Figura 2: Ejemplo de la lista de OOT que se ofrece en el GRC.



Es posible trabajar con GRC de GNU Radio usando todos esos recursos sin necesidad de entrar a la programación a nivel de código. Pero entrar a programar puede resultar una tarea gratificante: es muy sencillo, ya que se programa por bloques, es decir, el programador concentra su atención solo al bloque particular, lo cual se puede realizar con unas pocas líneas de código. Una vez hecho esto, el bloque puede pasar a ser parte de cualquier sistema de tiempo real que lo quiera usar una y otra vez. Pero además, haciendo código se tiene la posibilidad de abrir la mente más allá de los recursos que trae GNU Radio por defecto, incluso puede aportar sus desarrollos a la comunidad internacional.

En este capítulo no vamos a entrar en los detalles de lo que significa crear un nuevo módulo OOT, ni como se programa ya que existen muy buenos tutoriales para ello en la red, por ejemplo el que ofrece en el sitio web de este libro [1]. Solo vamos a detenernos en el grado de dificultad que surge al llevar a la práctica de la programación de bloques los conceptos de señales tipo stream y tipo vector. El problema consiste en que un bloque puede tener una o más señales de entrada o salida, pero solo hay un parámetro que representa a las entradas con todo lo asociado a ellas y uno que representa a las salidas. Esos dos parámetros se usan así:

- **input_items:** el nombre de la variable que por defecto tiene cada bloque de GNU Radio para reunir en el todas las señales de entrada al bloque. Entonces:
 - `input.items[0]` corresponde a la primer señal, incluso cuando hay solo una señal.
 - `input.items[1]` a la segunda y así sucesivamente.
- **output_items:** igual que el anterior, pero para las señales que salen del bloque.

Recomendaciones para la programación con señales:

Las siguientes son recomendaciones a seguir cuando se trabaja a nivel de código con señales en un bloque de GNU Radio:

- Pase las señales a parámetros que usted entienda mejor, los siguientes son ejemplos:
- Caso de un bloque con señales tipo stream y con una sola señal de entrada y una de salida:
 - `in=input.items[0]`.
 - `out=out.items[0]`.
 - De esta manera, usted sabe que “`in`” es el array de una dimensión que representa a su señal, osea que es un array de una sola fila de N elementos.
- Caso de un bloque con señales tipo stream y con dos señales de entrada y una de salida:
 - `in0=input.items[0]`.

- in1=input_items[1].
- out=out_items[0].
- De esta manera, usted sabe que in0 es el array de una dimensión que representa a su primera señal e in1 a la segunda, mientras out es de una dimensión y representa a la salida.

En los capítulos siguientes se tienen ejemplos donde se usan estos bloques, de manera que solo vamos a ocuparnos de analizar el código implementado en los siguientes casos.

Bloque acumulador:

```

import numpy as np
from gnuradio import gr

class blk(gr.sync_block):

    def __init__(self): # only default arguments here
        gr.sync_block.__init__(
            self,
            name='e_Acum', # will show up in GRC
            in_sig=[np.float32],
            out_sig=[np.float32]
        )

    def work(self, input_items, output_items):
        x = input_items[0] #Senial de entrada.
        y0 = output_items[0] #Senial acumulada

        y0[:] = np.cumsum(x)

        return len(y)

```

Bloque diferenciador:

```

import numpy as np
from gnuradio import gr

class blk(gr.sync_block):

    def __init__(self): # only default arguments here
        gr.sync_block.__init__(
            self,
            name='e_Diff', # will show up in GRC
            in_sig=[np.float32],
            out_sig=[np.float32]

            self.acum_anterior = 0
        )

    def work(self, input_items, output_items):
        x = input_items[0] #Senial de entrada.
        y0 = output_items[0] #Senial acumulada diferencial

        N = len(x)
        diff = np.cumsum(x) - self.acum_anterior
        self.acum_anterior = diff[N-1]
        y0[:] = diff
        return len(y)

```

Bloque para hallar promedios de tiempo:

Vamos a revisar el código de un Python Block desarrollado para las prácticas de acompañamiento de la asignatura de comunicaciones digitales, el objetivo de este Python Block es entregar al mismo tiempo

varios promedios de tiempo tales como: la media, la media cuadrática, el valor RMS, la potencia promedio, la desviación estándar. Los requerimientos son :

1. El bloque tiene una entrada y 5 salidas, una para cada parámetro: la media, la media cuadrática, el valor RMS, la potencia promedio y la desviación estándar de una señal de entrada.
2. Todo el código se implementa dentro de un solo Python Block
3. Ten en cuenta que si hay 5 salidas, la variable output.items, puede verse como:

```
y0 = output.items[0]
y1 = output.items[1]
y2 = output.items[2]
y3 = output.items[3]
y4 = output.items[4]
```

Ahora si teniendo esto en cuenta vamos con la programación.

```
import numpy as np
from gnuradio import gr

class blk(gr.sync_block):

    def __init__(self): # only default arguments here
        gr.sync_block.__init__(
            self,
            name='Promedios_de_tiempos', # will show up in GRC
            in_sig=[np.float32],
            out_sig=[np.float32,np.float32,np.float32,np.float32,np.float32]
        )
        self.acum_anterior = 0
        self.Ntotales = 0
        self.acum_anterior1 = 0
        self.acum_anterior2 = 0

    def work(self, input_items, output_items):
        x = input_items[0] #Senial de entrada.
        y0 = output_items[0] #Promedio de la senial
        y1 = output_items[1] #Media de la senial
        y2 = output_items[2] #RMS de la senial
        y3 = output_items[3] #Potencia promedio de la senial
        y4 = output_items[4] #Desviacion estandar de la senial

        #Calculo del promedio
        N = len(x)
        self.Ntotales = self.Ntotales + N
        acumulado = self.acum_anterior + np.cumsum(x)
        self.acum_anterior = acumulado[N-1]
        y0[:] = acumulado/self.Ntotales

        #Calculo de la media cuadratica
        x2=np.multiply(x,x)
        acumulado1=self.acum_anterior1 + np.cumsum(x2)
        self.acum_anterior1 = acumulado1[N-1]
        y1[:] = acumulado1/self.Ntotales

        #Calculo de la RMS
        y2[:] = np.sqrt(y1)

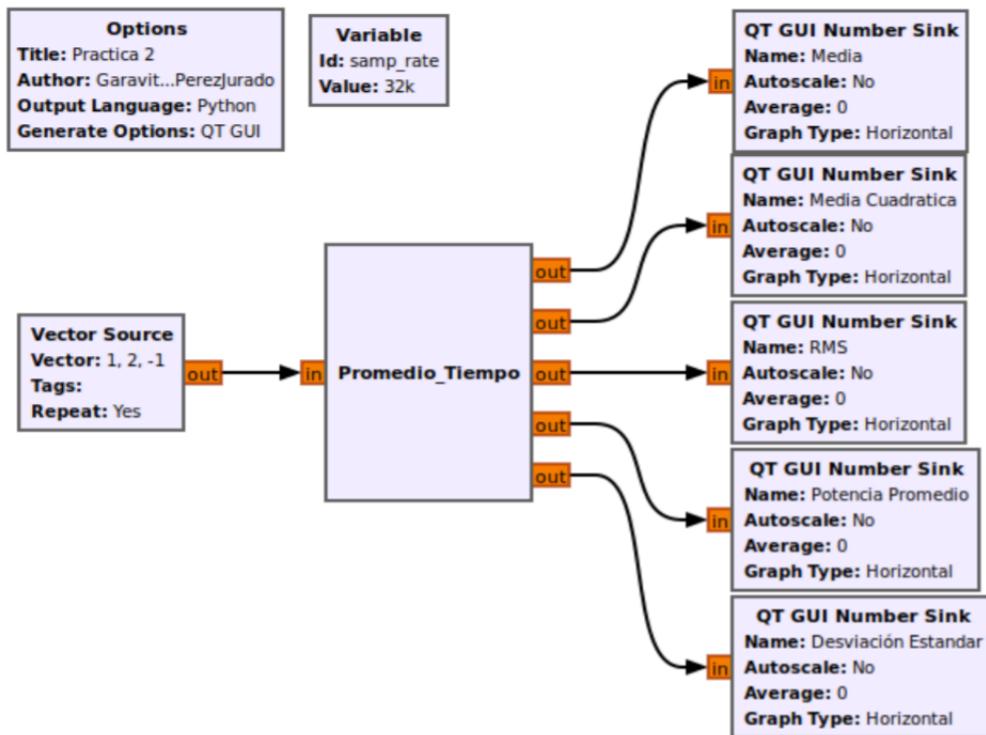
        #Calculo de la potencia promedio
        y3[:] = np.multiply(y2,y2)

        #Calculo de la desviacion estandar
        x3 = np.multiply(x-y0,x-y0)
        acumulado2 = self.acum_anterior2 + np.cumsum(x3)
        self.acum_anterior2 = acumulado2[N-1]
        y4[:] = np.sqrt(acumulado2/self.Ntotales)

    return len(x)
```

Para comprobar el correcto funcionamiento del anterior código de Python Block, construiremos el siguiente diagrama de bloques y lo ejecutaremos en GNU Radio.

Figura 3: Diagrama para evaluar el bloque de promedios de tiempos.



Capítulo 2

Señales aleatorias y promedios de tiempos

2.1. Planteamiento del problema

Hasta el momento, se han visto señales teóricas o determinísticas que pueden ser presentadas por fórmulas matemáticas u otras maneras. En el mundo real las señales son casi siempre aleatorias y eso hace que los conceptos vistos anteriormente no se puedan aplicar de manera directa. Por ejemplo, si pretendemos usar el concepto estricto de la TF para observar, en el dominio de las frecuencias la voz de una persona, deberíamos esperar a que esa persona hable durante toda su vida antes de aplicarle la TF, pues esta, por definición analiza la señal en el tiempo que va desde $t = -\infty$ hasta $t = \infty$. Es claro que esto no es viable en la mayoría de las aplicaciones prácticas debido a la necesidad de resultados en tiempos más reales.

Del análisis anterior surge la pregunta: ¿Es posible usar herramientas de análisis como la Transformada de Fourier, los filtros, los sistemas LIT, etc, que han tenido su origen en señales determinísticas, para analizar señales del mundo real donde predomina más bien la aleatoriedad y el caos? ¿Cómo hacerlo?.

2.2. Promedios de tiempo en señales reales

La respuesta está en la necesidad de adaptar esos conceptos a dos posibles campos:

- La teoría de señales aleatorias, la idea consiste en que si bien las señales son aleatorias, varios parámetros pueden ser vistos como determinísticos, al menos dentro de ciertos límites. Por ejemplo la media de una señal, la desviación estándar, la potencia promedio, la distribución de la potencia promedio en el dominio de las frecuencias, los cuales se pueden calcular de manera aproximada al observar el comportamiento de la señal durante un tiempo tan corto o tan grande como tan pequeña o grande sea la exactitud deseada.
- La teoría de procesos estocásticos, puede ser visto como una mayor generalización de la teoría de señales aleatorias, cuando la cuestión no se trata de analizar una sola señal sino un grupo de ellas. Por ejemplo, supongamos que hemos reunido las señales de voz de muchas personas diferentes para probar cómo se comportan al pasar por un sistema. El promedio de esas señales es otra señal que varía en el tiempo, pero que tiene un comportamiento más predecible.

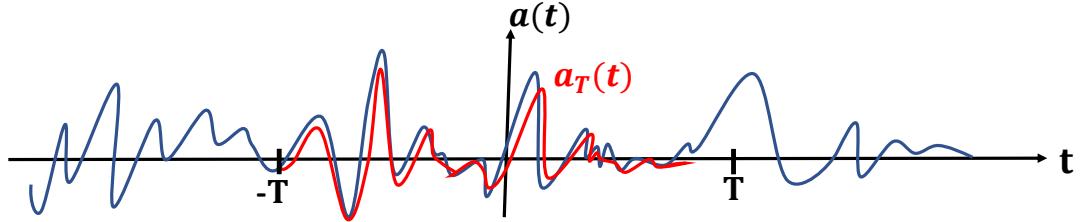
En este capítulo solo hablaremos de la teoría de señales aleatorias. Supongamos que en el problema en que usted trabaja solo se cuenta con una señal aleatoria, debido a las especificaciones de ese problema. Es claro que esa señal tiene una media y por lo tanto todos esos parámetros que se derivan de la media, como la media cuadrática, la varianza, la desviación estándar, etc. Incluso la función de distribución de probabilidad. En este caso, el concepto que se usa se conoce como promedios de tiempo. Los cálculos son similares a los que se realizan con una variable aleatoria, la diferencia está en que la variable tiene un resultado en cada instante de tiempo y que hay infinitos instantes de tiempo, ya que el tiempo es continuo.

El promedio de tiempo para una señal $a(t)$ se representa como $\langle a(t) \rangle$ y se halla así.

$$\langle a(t) \rangle = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T a(t) dt = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T a_T(t) dt \quad (2.1)$$

Donde $a_T(t)$ es una versión truncada de $a(t)$ como se muestra en la siguiente figura.

Figura 1: Señal truncada.



Como puede observarse, entre más grande sea T , más alta es la exactitud en el cálculo del promedio de una señal.

2.3. La media de una señal $x(t)$

La media de una señal $x(t)$ es:

$$X_m = \langle x(t) \rangle \quad (2.2)$$

2.4. La media cuadrática de una señal $x(t)$

La media cuadrática de una señal $x(t)$ es:

$$X_c = \langle x^2(t) \rangle \quad (2.3)$$

2.5. Otros promedios de tiempo

La Varianza

$$\sigma_x^2 = \langle [x(t) - X_m]^2 \rangle \quad (2.4)$$

La Desviación Estándar

$$\sigma_x = \sqrt{\langle [x(t) - X_m]^2 \rangle} \quad (2.5)$$

2.6. El valor RMS

El valor RMS (del inglés, *Root Mean Square* que se traduce como Raíz del Valor Medio Cuadrático) de una señal $x(t)$:

$$X_{RMS} = \sqrt{\langle |x(t)|^2 \rangle} \quad (2.6)$$

Como podemos ver, hemos usado $|x(t)|$ en lugar de $x(t)$ debido a que $x(t)$ puede ser una señal compleja en cuyo caso el Valor RMS se refiere a la magnitud media de señal.

2.7. La potencia promedio de una señal $x(t)$

La potencia promedio normalizada de una señal

$$P = X_{RMS}^2 = \langle |x(t)|^2 \rangle \quad (2.7)$$

Esta manera de concebir la potencia suscita cierta controversia debido a que nuestras señales usualmente tienen una procedencia eléctrica, de manera que la potencia debería estar relacionada con un voltaje y una corriente o la impedancia por la cual viaja esa corriente. En este aspecto es importante aclarar lo siguiente:

- no todas las señales son de naturaleza eléctrica, pero sí puede ser necesario conocer su potencia

- una señal puede cambiar su naturaleza al pasar por varia etapas de un sistema de comunicaciones. Por ejemplo, en un receptor, la señal que llega a la antena es una señal electromagnética; la antena la transforma en una señal que en la mayoría de los casos es vista como una señal de voltaje, pero, dependiendo del bloque que la trate también podría ser vista como una señal de corriente o algo diferente; la señal pasa luego por un ADC y lo que se tiene ahora es una señal discreta representada en forma de números que guardan relación con la amplitud de la señal antes del ADC, también podemos decir que es una señal digital.
- Las técnicas SDR realizan mayormente un procesamiento de señales digitales, en las cuales ya se ha perdido la relación que pueda existir entre la corriente y el voltaje. Con esto no estamos descartando que pueda existir un sistema que mantenga esa relación, pero no ocurre así en los sistemas conocidos de comunicaciones conocidos hoy.

Por lo anteriormente explicado, cuando se trata de procesar señales con técnicas SDR conviene usar un concepto genérico de potencia, que para los propósitos de este libro estamos llamando “potencia normalizada”. Pero no ocurre lo mismo cuando entramos a considerar las señales electromagnéticas, por ejemplo cuando se requiere conocer la potencia con que debe ser radiada una señal para que, al recorrer una una distancia d [metros] llegue con una potencia P_r [Vatios]. En cualquier caso la potencia puede ser medida en Vatios, pero una cosa son los Vatios del mundo físico (señales eléctricas) y otra cosa los que se usan en el procesamiento digital. Con este concepto no tendremos mayores problemas debido a que en las comunicaciones es la relación señal a ruido (SNR, del inglés Signal to Noise Relationship) la que conserva la relación entre el mundo físico y el del procesamiento digital.

2.8. La función de Autocorrelación

La función de Autocorrelación de $x(t)$:

$$R_X(\tau) = \langle x(t)x(t + \tau) \rangle = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t)x(t + \tau) d\tau \quad (2.8)$$

En el caso de las señales complejas la definición cambia debido a que la Función de Autocorrelación es función de la potencia

$$R_X(\tau) = \langle |x(t)x(t + \tau)| \rangle = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T |x(t)x(t + \tau)| d\tau \quad (2.9)$$

Es importante notar que:

- La función de Autocorrelación evaluada en $\tau = 0$, es igual a la media cuadrática de la señal.
- Por lo anterior, si la media de la señal es cero, entonces la función de Autocorrelación evaluada en $\tau = 0$, es igual a la potencia promedio de la señal.

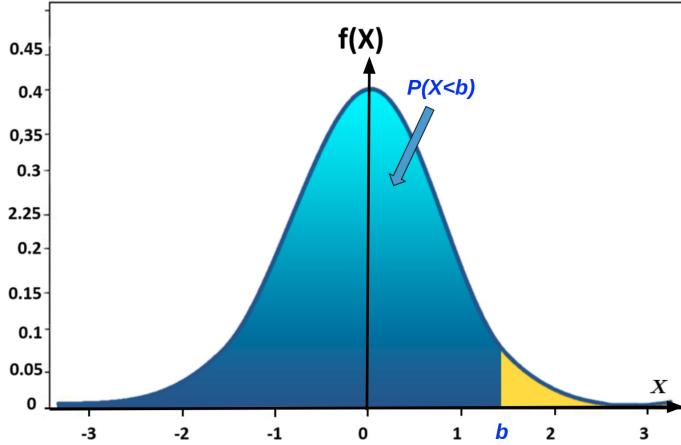
Error en la siguiente expresión porque no se debe usar el valor absoluto sino que una de las partes debe ser el conjugado. Conclusion: la F de Autocorrelación es siempre real

Debe haber error en la fórmula 1.51, porque si la señal es compleja no se usa el módulo, sino que una de las partes es el conjugado

2.9. La Función de Densidad de Probabilidad

La Función de Densidad de Probabilidad (FDP) describe la probabilidad de cada valor que puede tomar una variable aleatoria. También permite conocer la probabilidad de que una variable aleatoria caiga dentro de un rango de posibles valores. Para los propósitos de este resumen, resulta suficiente comprender lo que representa la curva de la FDP en un ejemplo como el que se presenta en la figura 2.

Figura 2: Ejemplo de la Función de Densidad de Probabilidad.

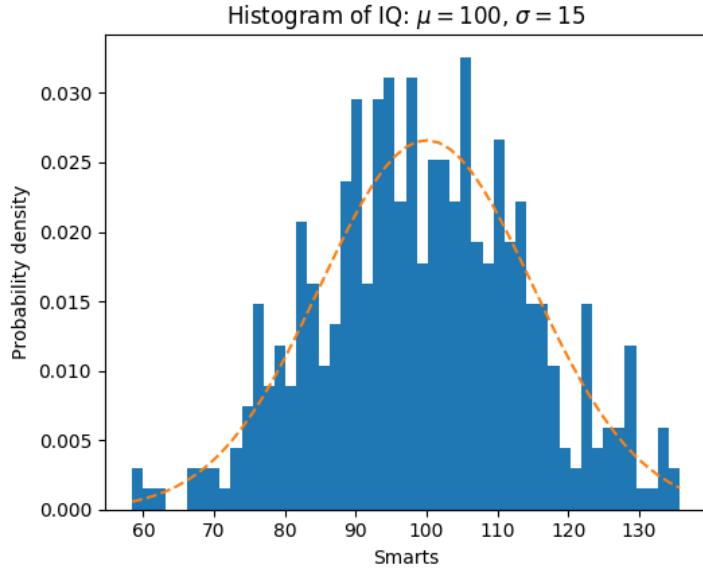


Para el ejemplo que representa esa figura, podemos observar que la probabilidad de que la variable aleatoria X tome el valor b es aproximadamente 0,1, que si se expresa en porcentaje, equivale al 10 %. El área bajo la curva de la FDP es siempre igual a 1. La probabilidad de que la variable aleatoria tome un valor menor que b , es decir $P(X < b)$ es igual al área bajo la curva desde $-\infty$ hasta b y para el ejemplo dado en la figura 2 se encuentra sombreada en color azul. De manera similar, buscando siempre el área bajo la curva entre intervalos diversos es posible encontrar la probabilidad de que X tome cualquier otro rango de valores.

Un aspecto muy importante a tener siempre en cuenta es que la FDP caracteriza completamente a una variable aleatoria. Es decir, si se conoce la FDP de una variable aleatoria, ya se tiene toda la información sobre esa variable.

En GNU Radio es común aproximarse a la FDP mediante lo que se conoce como un histograma, mientras que el uso de una forma continua de la FDP es más común en los análisis teóricos. En los histogramas usualmente se considera que la variable aleatoria puede tomar un número finito de valores discretos, como se muestra en el ejemplo de la figura 3. Solo hay que tener en cuenta que los histogramas no siempre se refieren a la FDP de una señal aleatoria, es más bien una manera gráfica de mostrar resultados de mediciones o de observaciones estadísticas, de manera que en el eje de las coordenadas podría estar la probabilidad, pero también el precio del dolar, datos de habitantes o valores de voltaje medidos a una señal. En el caso específico de GNU Radio se tiene el bloque “QT GUI Histogram Sink” que tiene como entrada una señal, realiza un conteo de las veces que se repiten los valores de la señal para mostrar una distribución, pero no calcula la probabilidad, como en el ejemplo que se muestra en la figura 19.

Figura 3: Ejemplo de un histograma.



fuente: Matplotlib

2.10. La Función de Distribución Acumulativa

La Función de Distribución Acumulativa (FDA) es otra manera de caracterizar completamente una variable aleatoria, pero no es nada nuevo con respecto a la FDP, sino más bien una deducción de ella, ya que la FDA se puede expresar como:

$$F(x) = \int_{-\infty}^x f(u)du, \quad (2.10)$$

Donde $f(u)$ es la FDP de una variable aleatoria U .

2.11. Promedios de tiempo en señales complejas

En el caso de las señales complejas es especial, ya que una señal es una especie de matrimonio entre una señal real y una imaginaria, de modo que es una pareja inseparable de señales. En este caso, es importante tener en cuenta lo siguiente:

- Una señal compleja, aunque es un apareamiento entre dos señales, puede ser vista como una sola cosa si se representa como un vector en el plano complejo, el cual puede variar diversos parámetros como su velocidad, su magnitud, su fase.
- Hablar del promedio de tiempo en una señal compleja es hablar del comportamiento promedio del vector.
- El promedio de tiempo se puede definir como el valor complejo, donde la parte real y la imaginaria resultan de sacar el promedio de tiempo de la parte real de la señal y de la parte imaginaria respectivamente. Lo cual, se traduce en el punto del plano complejo donde hay mayor probabilidad de encontrar al vector.
- En el caso del valor RMS es un tanto diferente ya que guarda relación con la potencia de una señal que bien podría ser compleja; en vez de la media cuadrática, lo que se usa es la magnitud cuadrática media del vector en el plano complejo. Esto se traduce en la longitud al cuadrado más probable para el vector.
- La magnitud media, se deduce de lo dicho anteriormente, ya que es la raíz cuadrada de la magnitud cuadrática media.

Esta parte hay que revisarla, pues el valor RMS puede ser más bien la magnitud sobre $\sqrt{2}$

- En este caso, la potencia promedio de una señal es claramente el cuadrado de la magnitud media.

De acuerdo a las consideraciones anteriores, la media de una señal compleja aleatoria es un valor complejo y le corresponde esta fórmula:

$$X_m = \langle x_r(t) \rangle + j \langle x_i(t) \rangle \quad (2.11)$$

Donde $x_r(t)$ y $x_i(t)$ corresponden a la parte real y a la imaginaria respectivamente de la señal analizada.

La magnitud media cuadrática de una señal compleja aleatoria es un valor real y le corresponde la siguiente fórmula:

- hay varios errores a corregir para el caso en que una señal $x(t)$ es compleja
- la función de autocorrelación es

$$R_X(\tau) = \langle x(t)x^*(t+\tau) \rangle = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t)x^*(t+\tau)d\tau \quad (2.12)$$

donde * significa el conjugado de la señal

- Está claro que la función de autocorrelación evaluada en cero es igual a la media cuadrática, que es la potencia promedio potencia promedio

$$X_c = P = \langle x(t)x^*(t) \rangle \quad (2.13)$$

- por lo tanto $X_c = P = \langle x_r^2(t) + x_i^2(t) \rangle = \langle |x(t)|^2 \rangle$

$$|X_c| = \langle |x(t)|^2 \rangle \quad (2.14)$$

También es válido decir que:

$$|X_c| = X_{rc} + X_{ic} \quad (2.15)$$

Donde X_{rc} y X_{ic} son la media cuadrática de la parte real y de la parte imaginaria respectivamente de la señal analizada $x(t)$. A la función de Autocorrelación le corresponde la fórmula:

$$R_X(\tau) = \langle |x(t)x(t+\tau)| \rangle = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T |x(t)x(t+\tau)|d\tau \quad (2.16)$$

Potencia promedio normalizada es:

$$P = X_c \quad (2.17)$$

De igual manera, se puede deducir que el Valor RMS para las señales complejas es:

$$X_{RMS} = \sqrt{|X_c|} = \sqrt{\langle |x(t)|^2 \rangle} \quad (2.18)$$

En el caso particular, cuando la media de una señal es cero, la potencia promedio es la misma varianza

$$P = \sigma_x^2$$

cuando

$$X_m = 0$$

2.12. La Densidad Espectral de Potencia

La PSD como un promedio de tiempo: Ya se ha dicho que la PSD es la distribución de la potencia promedio, de la señal analizada en el espectro de frecuencias. La PSD de una señal $x(t)$ se representa como $S_X(f)$.

Debido a la anterior afirmación se obtiene que:

$$P = \int_{-\infty}^{\infty} S_X(f)df \quad (2.19)$$

Pero la potencia promedio también puede ser hallada en el dominio del tiempo, aunque de manera aproximada, recurriendo a una versión truncada $x_T(t)$ de la señal $x(t)$ así:

$$P = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-\infty}^{\infty} |x_T(t)|^2 dt \quad (2.20)$$

Aquí puede haber error en la expresión anterior, porque si $x(t)$ es complejo, la multiplicación debería ser más bien de $x(t)$ por su conjugado. Tampoco convence que la media de una señal compuesta sea la suma de las medias de sus componentes.

la siguiente expresión debe ser revisada, pues si $x(t)$ es complejo, P resultaría siendo complejo

Ahora recordemos que, de acuerdo a la relación de Parseval:

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |X(f)|^2 df \quad (2.21)$$

Por lo tanto, también es válido hallar la potencia promedio de una señal $x_T(t)$ a partir de su TF $X(f, T)$:

$$P = \int_{-\infty}^{\infty} |X(f, T)|^2 df \quad (2.22)$$

Donde $X(f, T)$ es la TF de $x_T(t)$:

$$X(f, T) = \int_{-\infty}^{\infty} x_T(t) e^{-j2\pi f t} dt = \int_{-T}^{T} x(t) e^{-j2\pi f t} dt \quad (2.23)$$

Ahora, igualemos 2.19 y 2.22 para obtener:

$$\int_{-\infty}^{\infty} S_X(f) df = \int_{-\infty}^{\infty} |X(f, T)|^2 df \quad (2.24)$$

Con lo cual se deduce que la PSD se puede calcular a partir de la TF de una versión truncada de la señal así:

$$S_X(f) = \lim_{T \rightarrow \infty} \frac{1}{2T} |X(f, T)|^2 \quad (2.25)$$

Como es bien sabido, las dimensiones de la TF de una señal de voltaje son en V/Hz, de modo que las de la TF en magnitud al cuadrado son en Watts/Hz². Por lo tanto, las de la PSD son de Watts/Hz.

En 1.64 hay error, falta poner el límite cuando T tiene a infinito d 1/2T

En 1.66 hay error, falta poner el límite cuando T tiene a infinito d 1/2T

2.13. Relación de la Densidad Espectral de Potencia con la función de Autocorrelación

Existe una importante relación entre PSD y Autocorrelación de una señal aleatoria la cual se expresa de la siguiente manera: $S_X(f)$ es también la TF de $R_X(\tau)$ entonces,

$$S_X(f) = \int_{-\infty}^{\infty} R_X(\tau) e^{-j2\pi f \tau} d\tau. \quad (2.26)$$

Consecuentemente la potencia promedio se puede hallar también como:

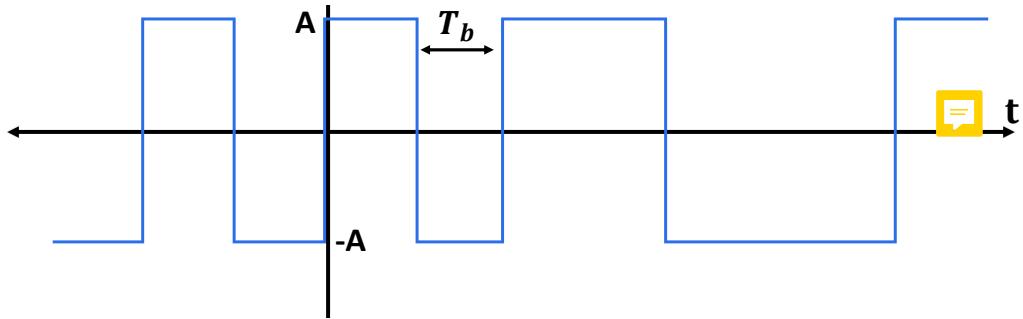
$$P = R_X(0) \quad (2.27)$$

2.14. Señal binaria bipolar aleatoria

Uno de los primeros pasos en el estudio de las comunicaciones consiste en conocer y analizar ciertas señales que son claves. Una de ellas son las señales binarias bipolares aleatorias.

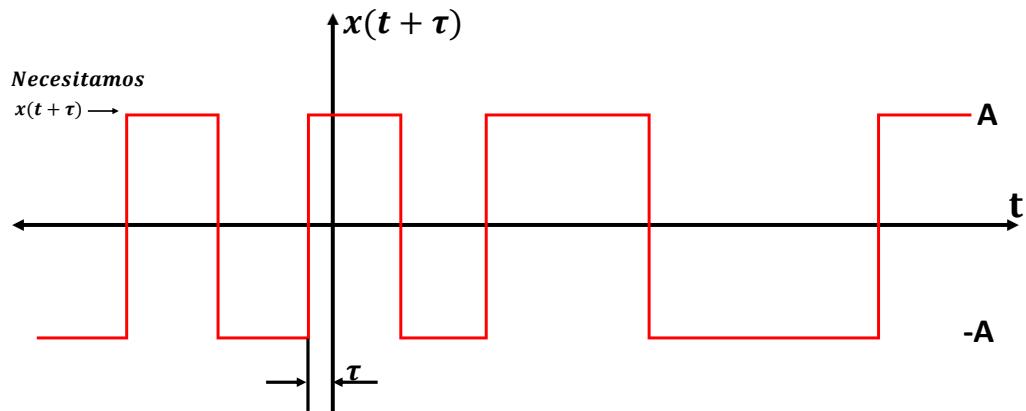
Obtenemos primero La función de Autocorrelación. Tomaremos como ejemplo la señal $x(t)$. de la figura 4 donde T_b es el ancho de cada pulso que en términos de señales digitales, puede ser equivalente a la duración de un bit.

Figura 4: Ejemplo de una señal binaria aleatoria bipolar $x(t)$.



Recordemos que $R_X(\tau) = \langle x(t)x(t + \tau) \rangle = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t)x(t + \tau) d\tau$. En la figura 5 se tiene la forma de la señal $x(t + \tau)$ cuando $\tau < T_b$

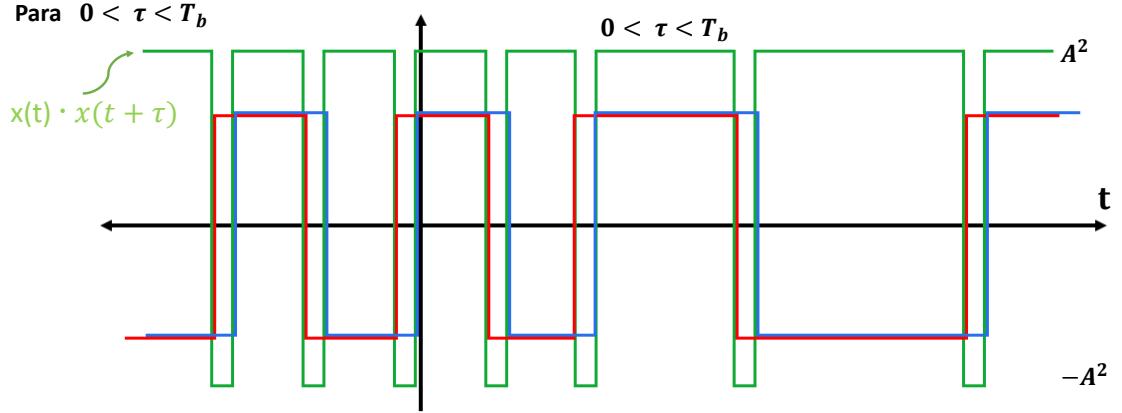
Figura 5: El desplazamiento de la señal binaria aleatoria bipolar que resulta en la señal $\tau < T_b$.



Vamos a revisar los siguientes casos:

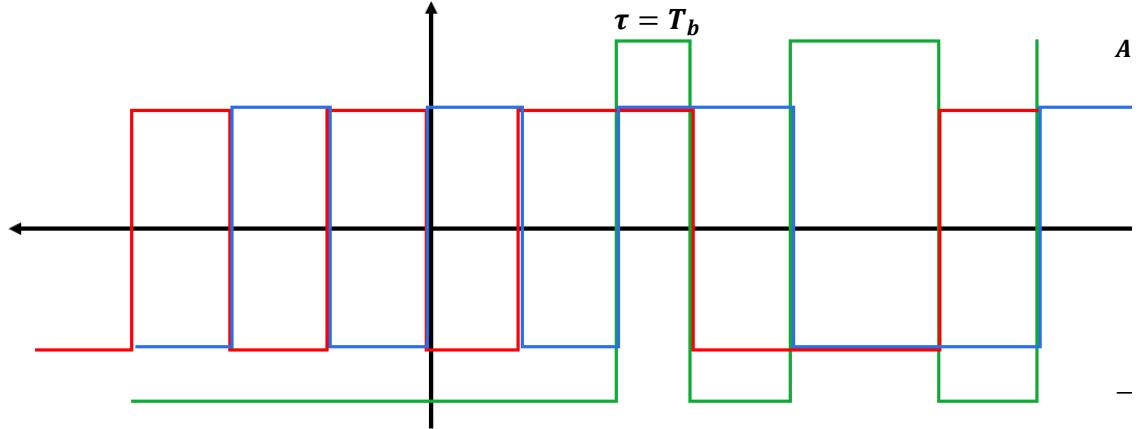
- Caso en que $\tau = 0$:
Tenemos que $\langle x(t)x(t + \tau) \rangle = \langle x^2(t) \rangle = \langle A^2 \rangle = A^2$
- Caso en que $0 < \tau < T_b$:
En este caso, la operación $x(t)x(t + \tau)$ tiene la forma que se presenta en la figura 6.

Figura 6: Multiplicación de la señal $x(t)$ con su versión desplazada $x(t + \tau)$ cuando $0 < \tau < T_b$.



- Caso en que $\tau = T_b$: Tenemos que la multiplicación $x(t)x(t + \tau)$ resulta siendo una nueva señal binaria aleatoria bipolar como se muestra en la figura 7.

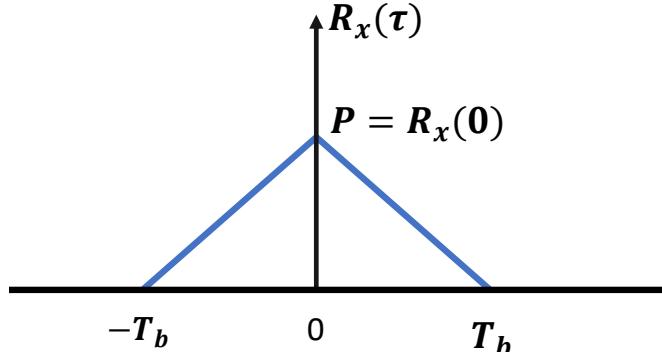
Figura 7: Multiplicación de la señal $x(t)$ con su versión desplazada $x(t + \tau)$ cuando $\tau = T_b$



Vemos que para $\tau = 0, x(t)x(t + \tau) = A^2$ y su promedio de tiempo es $\langle x(t)x(t + \tau) \rangle = A^2$ que es la misma potencia promedio. Cuando $0 < \tau < T_b$ el promedio de tiempo de $x(t)x(t + \tau)$ cae linealmente. Cuando $\tau = T_b$ ocurre que $x(t)x(t + \tau)$ se convierte en una nueva señal binaria bipolar aleatoria, cuyo promedio de tiempo es cero. Algo similar ocurre cuando $\tau > T_b$. Ahora, cuando $\tau < 0$ ocurre lo mismo que cuando $\tau > 0$.

Es importante tener en cuenta que la función de Autocorrelación evaluada en cero es la misma potencia promedio $P = R_X(0)$, solo que para el caso dado, $P = R_X(0) = A^2$. De modo que la función de Autocorrelación tiene la forma de la figura 8.

Figura 8: La función de Autocorrelación de una señal binaria aleatoria bipolar.

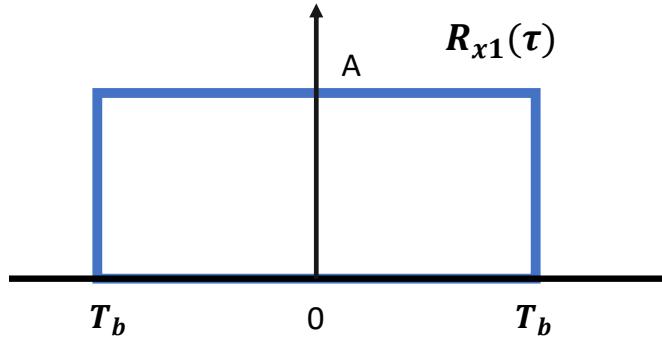


Recordemos que la PSD es la TF de la función de Autocorrelación. De manera que podemos usar la función de Autocorrelación obtenida para buscar la forma de la PSD. Para ello, expresamos la función de Autocorrelación como una Convolución así:

$$R_x(\tau) = \frac{1}{T_b} R_{X1}(\tau) * R_{X1}(\tau) \quad (2.28)$$

Donde $R_{X1}(\tau)$ tiene una forma rectangular como se muestra en la siguiente figura 9.

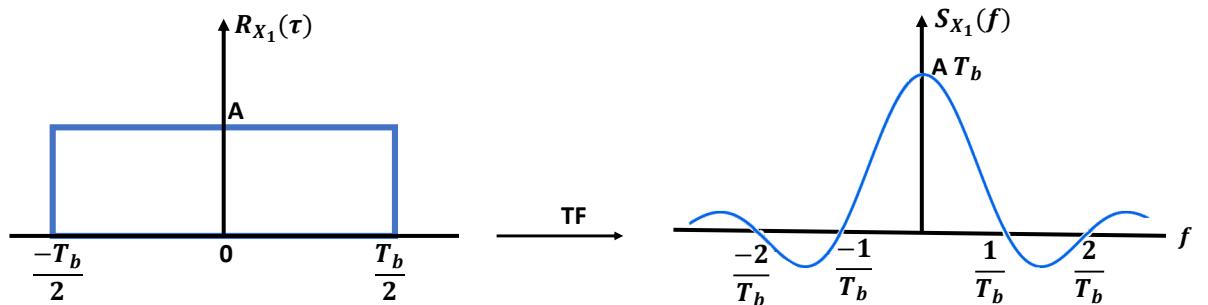
Figura 9: Forma de $R_{X1}(\tau)$. Error: En el eje del tiempo: 1) debe mostrar flecha con la letra τ ; 2) a la izquierda del rectángulo debe aparecer $-T_b/2$, a la derecha $T_b/2$



De esta manera podemos usar la TF de una señal de forma rectangular, la cual es bien conocida y que se muestra en la figura 10. Error en la gráfica ya que $R_{x1}(\tau)$ debe estar entre $-T_b/2$ y $T_b/2$.

debe haber un error en la figura, porque la función debe estar definida entre $-T_b/2$ y $T_b/2$

Figura 10: $R_{X1}(\tau)$ y su Transformada de Fourier.



Nótese que $S_{X1}(f)$ estaría dado en V/Hz tal como se conoce la TF de una señal cuadrada.

También aprovechamos el Teorema de la Convolución, también conocido como Teorema de Wiener-Khitchine que dice que si se tiene:

$$x_1(t) \xrightarrow{T_F} X_1(f) \quad (2.29)$$

y

$$x_2(t) \xrightarrow{T_F} X_2(f) \quad (2.30)$$

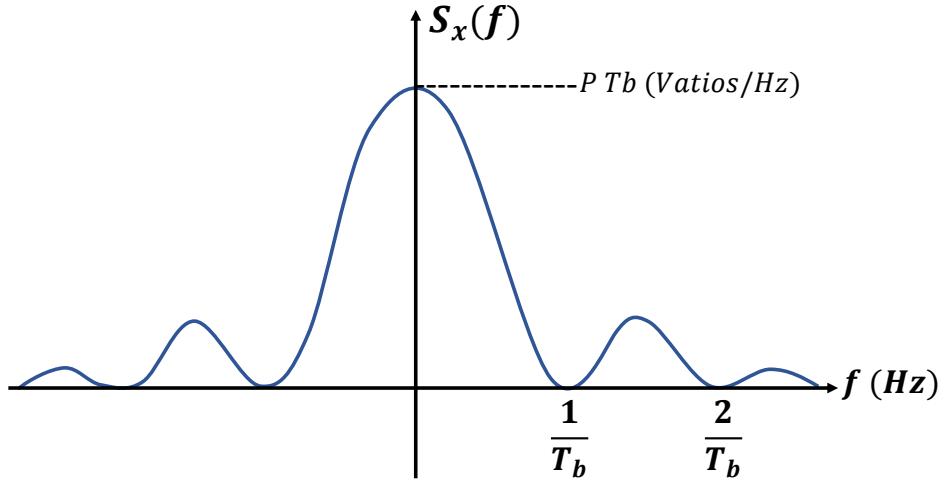
Entonces

$$x_1(t) * x_2(t) \xrightarrow{T_F} X_1(f)X_2(f) \quad (2.31)$$

Luego,

$$\frac{1}{T_b} R_{X_1}(\tau) * R_{X_2}(\tau) \xrightarrow{T_F} \frac{1}{T_b} S_{X_1}^2(f) \quad (2.32)$$

Figura 11: PSD de la forma forma de $R_{X_1}(\tau)$.



De esta manera se obtiene que la forma de la PSD de una señal binaria aleatoria bipolar corresponde a la función sinc cuadrática:

$$S_X(f) = PT_b \text{sinc}^2(f/R_b) \quad (2.33)$$

donde $R_b = 1/T_b$ es la rata de bits. La gráfica de la PSD de una Función Binaria Aleatoria Bipolar se presenta en la figura 11.

2.15. El ruido blanco

El ruido blanco (WN, del inglés *White Noise*) siempre está presente en cualquier medio de propagación, pero no deja de ser una abstracción matemática, debido a ciertos supuestos como:

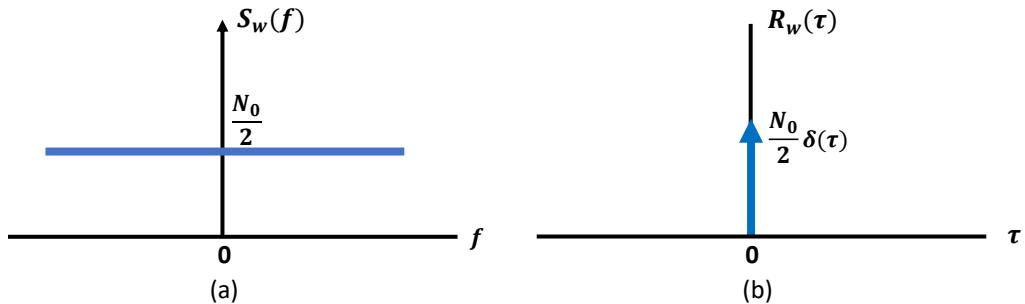
- Tienen un ancho de banda infinito. De allí viene el término “blanco”, pues se supone que el color blanco resulta de combinar homogéneamente todos los colores (las frecuencias).
- La Densidad Espectral de Potencia es constante e igual a $N_0/2$, donde:

$$N_0 = kT_e \quad (2.34)$$

Donde k es la constante de Boltzmann y T_e es la temperatura equivalente del ruido en el receptor. La idea de relacionar la potencia del ruido con un valor de temperatura equivalente proviene de experimentos realizados con sistemas electrónicos, donde se ha observado que a medida que aumenta la temperatura en estos sistemas, aumenta el ruido, de manera que la temperatura resulta en este caso directamente proporcional con la potencia. En los sistemas de comunicación, el ruido N_0 puede tener una componente relacionada con las características del receptor, pero es también muy relevante tener en cuenta que la señal que llega a la antena de un receptor tiene el ruido natural del medio inalámbrico.

- La función de Autocorrelación es una función delta centrada en cero, como se muestra en la figura 12, lo cual significa que dos muestras del ruido, por muy cercanas que sean no están correlacionadas, de modo que una muestra aparece correlacionada solo con sí misma. En este sentido, el ruido blanco es lo último en aleatoriedad.

Figura 12: PSD del ruido blanco y su función de Autocorrelación.



- Conocer el desempeño de un sistema de comunicaciones frente al ruido blanco que puede presentarse en el canal, no deja de ser una idealización, pero sirve de referencia para caracterizar el sistema y los métodos usados en el procesamiento de la información.

2.16. La voz humana en la telefonía

El sonido es uno de los principales tipos de mensajes que se emiten por los medios de comunicación y se usa intensamente en ejemplos y prácticas de este libro, sobre todo los sonidos que el ser humano emite para comunicar. La voz es toda una ciencia, sin embargo, para los propósitos del presente libro los siguientes apuntes son los de mayor relevancia:

- El sonido son ondas mecánicas longitudinales, se desplazan de un lugar a otro mediante perturbaciones de un medio elástico (sólido, líquido, gaseoso).
- Las ondas sonoras se reducen a los límites de frecuencia que pueden estimular el oído humano para ser percibidas en el cerebro.
- Los límites de frecuencia de las ondas sonoras se extienden de aproximadamente 20 Hz a cerca de 20 [[kHz]] y se llaman límites de audición.
- La voz está constituida por un conjunto de sonidos generados por el aparato fonador, de manera que tiene un menor rango de frecuencias que el sonido.
- En aplicaciones reales de comunicaciones, una señal de voz es una señal aleatoria, desde este punto de vista sus características son de tipo aleatorio, pero además depende de cada persona.
- De acuerdo a los estudios tenidos en cuenta por la UIT, la mayor energía de la voz se encuentra concentrada por debajo de los 4 [[kHz]]. De modo que la UIT ha establecido que para los propósitos de la telefonía, la señal de la voz puede estar limitada a 4 [[kHz]]. Es por eso que, al implementarse las tecnologías digitales en la frecuencia fija, se estableció que la frecuencia de muestreo es de $Samprateaudio = 8$ [[kHz]]. También se estableció que los niveles de cuantificación son 256, lo cual equivale a 8 bits por muestra ya que $Nbps = 256 = 2^8$.
- De lo anterior se deduce que la tasa de bits en la telefonía es de $Rb = Samprateaudio \times Nbps = 64$ kbps.

Complementar con todo lo que hemos aprendido de radiotecnología, para que el lector entienda que en el mundo real lo que se recibe de manera natural es el ruido que proviene del universo y no es lo mismo que el ruido blanco, sino información de la historia del universo

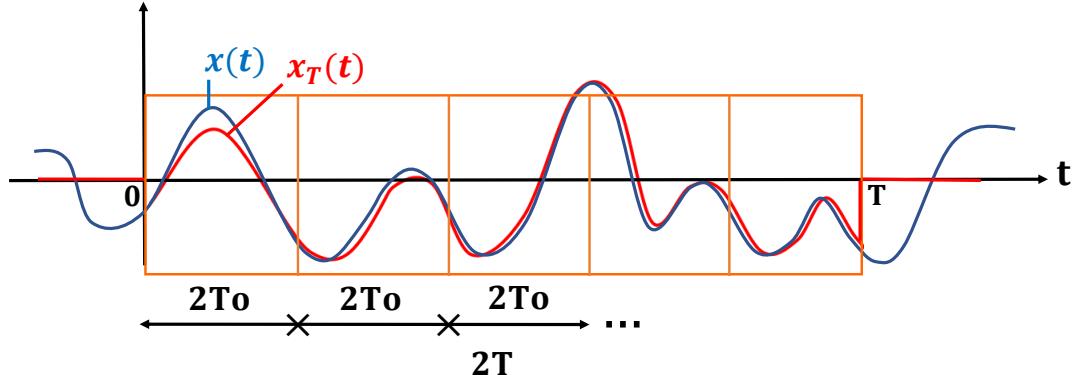
2.17. Implementaciones en GNU Radio

2.17.1. Método de Welch para obtener la PSD

La idea es caer en cuenta que esta fórmula 2.25 expresa un promedio. Para llevar esa expresión a la práctica y calcular la PSD por medios computacionales, P.D. Welch [2] introduce el término “periodograma” para referirse a la expresión $\frac{1}{2T} |X(f, T)|^2$ y a su gráfica.

En simulink o en GNU Radio se puede ir mostrando cómo se va calculando ese promedio a medida que pasa el tiempo. De modo que no es necesario esperar a que transcurra todo el tiempo T para visualizar la PSD.

Figura 13: Ejemplo de periodograma.

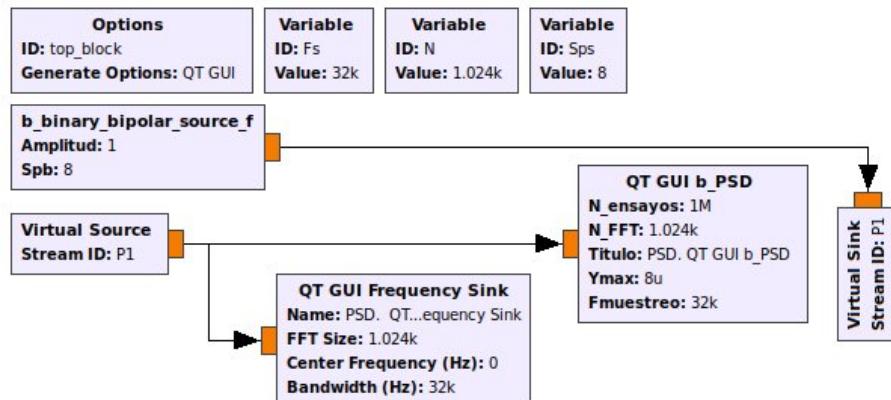


La idea es dividir el tiempo de medición T en pequeñas ventanas de duración T_0 . De esta manera, la señal truncada $X_T(t)$ resulta dividida en algo que vamos a llamar señales sub truncadas porque son versiones de más corta duración. Como resultado, tenemos las señales sub truncadas: $x_{T,1}(t), x_{T,2}(t), \dots, x_{T,N}(t)$. El siguiente paso es ir calculando, para cada señal subtruncada la TF para obtener: $|X_1(f, T)|^2, |X_2(f, T)|^2, \dots, |X_N(f, T)|^2$. Finalmente, se va recalculando la PSD resultante a medida que se procesa una nueva señal sub truncada, de manera que la PSD va pasando por diversas versiones $S_{X,1}(f, T), S_{X,2}(f, T), \dots, S_{X,N}(f, T)$, donde la última versión, que en este caso es $S_{X,N}(f, T)$ es la mejor aproximación de la PSD de la señal aleatoria. Lo que se logra es que el usuario no tenga que esperar mucho tiempo para ver la PSD, sino que en tiempo real pueda ver como la PSD va tomando una forma cada vez más definitiva. Para el ajuste mencionado solo hay que tener en cuenta que la PSD es un promedio, pero eso se explicará más abajo en una implementación que usa la FFT como bloque de cálculo de la Transformada de Fourier Truncada.

2.17.2. Implementación en GNU Radio de un analizador de la PSD

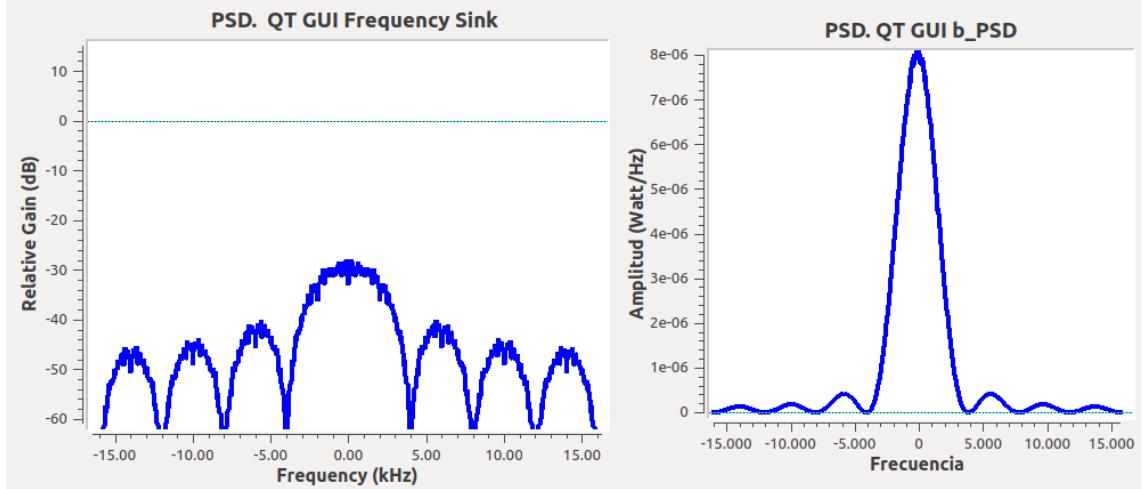
Analizaremos dos bloques que tiene GNU Radio y que usan la FFT: El bloque FFT y el bloque *QT GUI Frequency Sink* usando el flujo de la figura 14. En este ejemplo los parámetros usados son: frecuencia de muestreo $F_s = 32$ [[kHz]], el orden de la FFT es $N = 1024$, el número de muestras para darle a cada binaria rectangular es $Sps = 8$.

Figura 14: Comparación del bloque FFT con *QT GUI Frequency Sink* para observar la PSD de una señal binaria bipolar aleatoria



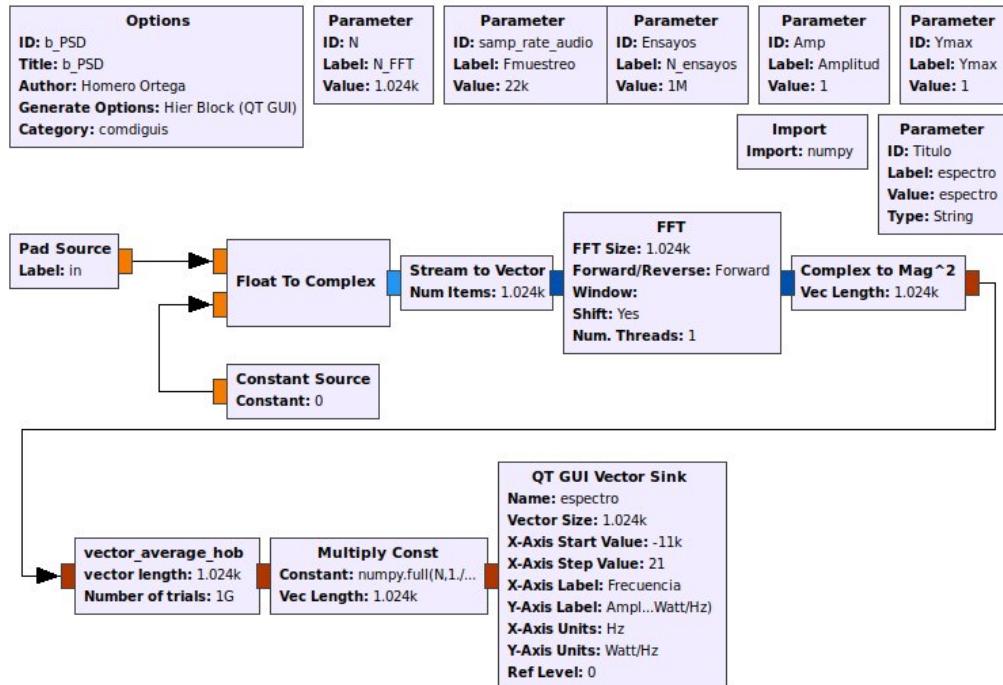
Flujo de datos: La-FFT.grc

Figura 15: PSD obtenida por dos métodos



En la figura 15 tenemos el resultado de la PSD que entrega el bloque *QT GUI Frequency Sink* comparado con el que entrega el bloque “*QT GUI b_PSD*”. Este último se basa en el algoritmo FFT y fue implementado en forma de un bloque jerárquico que se descubre en la figura 16

Figura 16: Flujograma del bloque jerarquico “QT GUI b_PSD”



Flujograma usado: *b_QT_PSD.grc*. Es un bloque jerárquico de la librería comdiguis

Conclusiones de las observaciones y recomendaciones para usar estos bloques:

- Obtener la PSD usando el bloque FFT requiere preparar el vector, pasar el resultado de la FFT a magnitud al cuadrado, realizar un promediado con los vectores previos, convertir a dB y graficar.
- Aunque parezca diferente, la imagen obtenida con el bloque FFT es equivalente a la del bloque *QT GUI Frequency Sink*, ya que la teoría de Fourier dice que la Transformada de Fourier Discreta (DFT) es periódica en $F_r = 2\pi \text{Rad/seg}$ y eso significa que es periódico en la frecuencia de muestreo F_s (Hz)

de la señal vista en el tiempo. Por eso, si el bloque FFT pudiese ser configurado para que muestre el espectro entre $\frac{-F_s}{2}$ y $\frac{F_s}{2}$ tendríamos imágenes similares. La imagen obtenida con el bloque FFT es más continua debido a que el bloque *vector_average.hob*, usado como complemento al bloque FFT realiza un promediado más prolongado.

- La frecuencia máxima que puede mostrar un analizador de espectro es igual a la frecuencia de muestreo sobre 2. También puede observarse que el ancho de banda B, que abarca tanto frecuencias negativas como positivas, es igual a la frecuencia de muestreo.
- N es el tamaño del vector y es igual al orden de la FFT, también es igual al número de puntos que son graficados en la ventana. Se deduce que la resolución espectral es $f_{Resol} = \frac{F_s}{N}$.
- Finalmente, es importante tener en cuenta que la PSD se calcula usualmente a señales que son parte del mundo continuo, por ejemplo la señal que se emite desde un transmisor, la que se recibe en un receptor, la que produce un micrófono o la que usa una bocina. Desde este punto de vista, no es usual buscar la PSD de señales que pueden encontrarse en bloques intermedios como por ejemplo un codificador o un modulador digital bandabase.

2.17.3. Implementación de un generador vectorial de señales básicas

El reto de los generadores vectoriales es generar en tiempo real y con parámetros reales, cualquier tipo de señal que pueda necesitar un investigador. En la figura 17 se muestra el flujograma propuesto.

Figura 17: Flujograma de un generador vectorial básico.

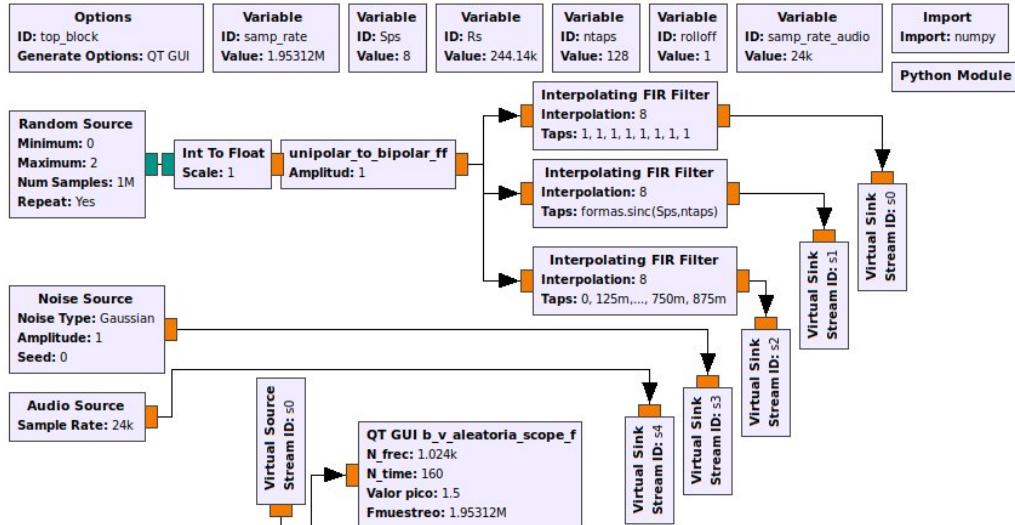
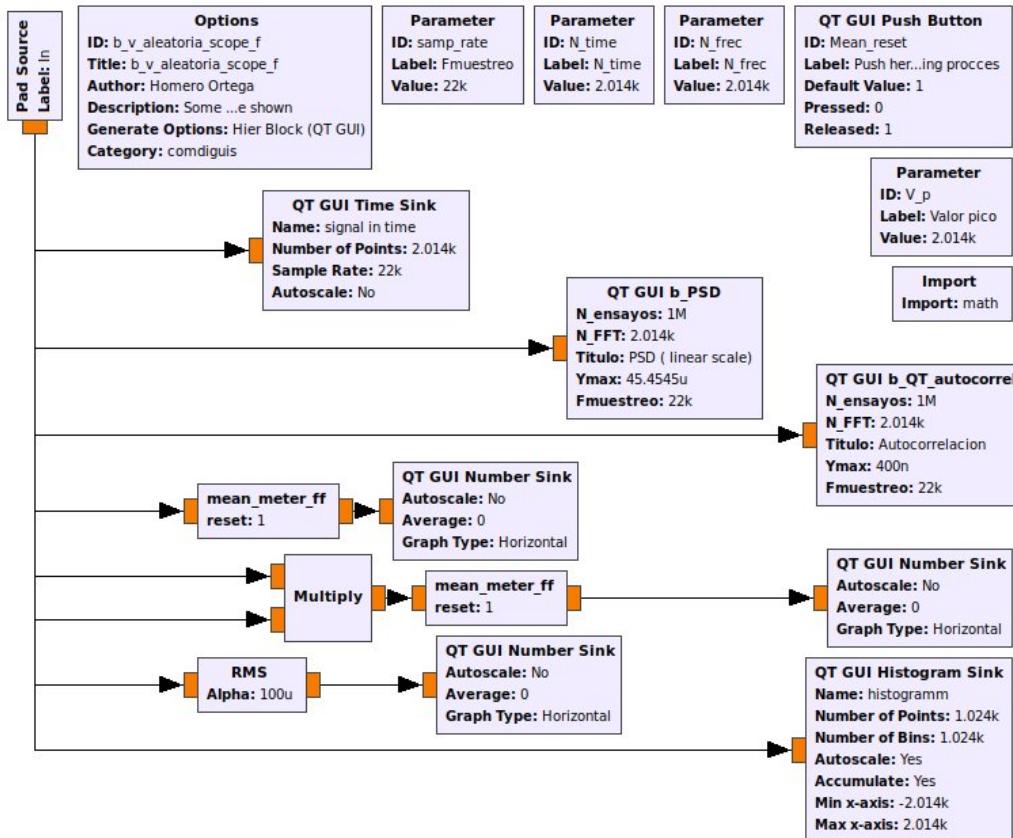


Figura 18: Flujograma para una herramienta de visualización de promedios de tiempo.



Flujograma usado: *b_QT-v_aleatoria_scope.grc* Es parte de la librería comdiguis

Figura 19: Promedios de tiempo. Señal de ruido blanco gausiano.

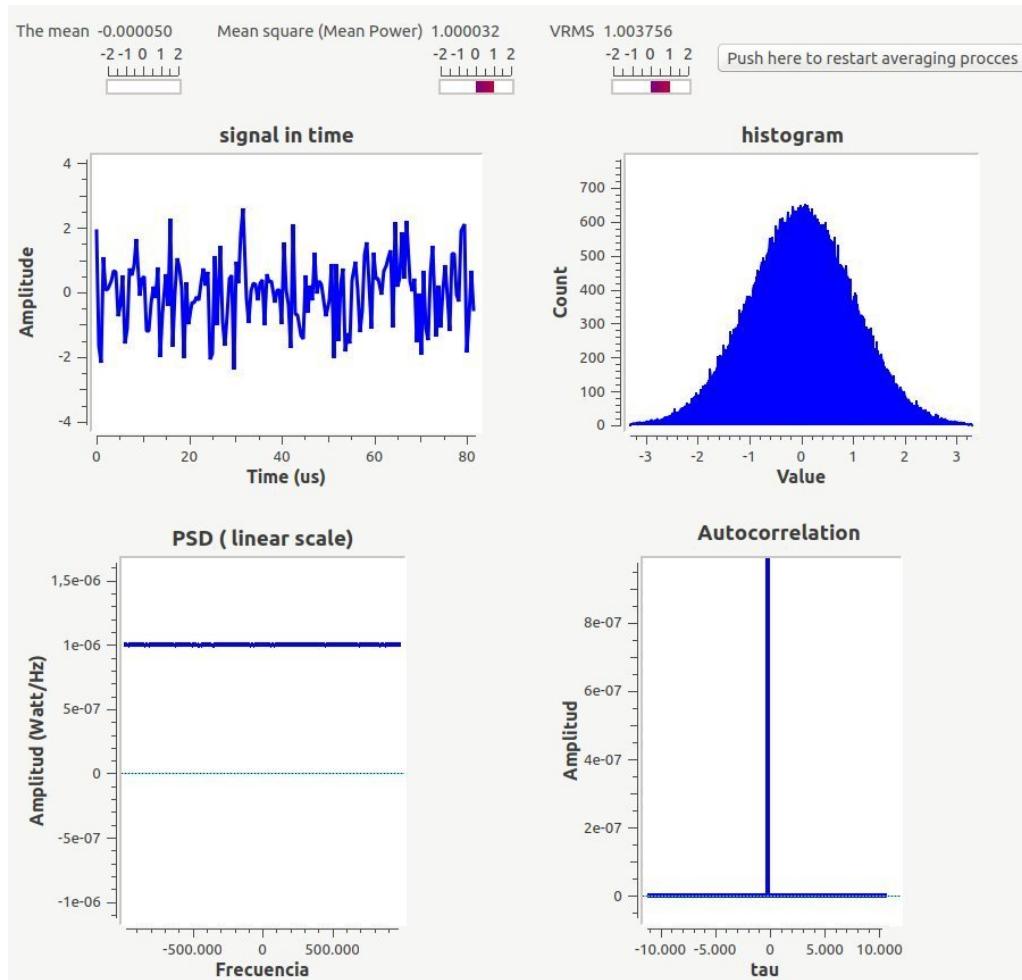


Figura 20: Promedios de tiempo. Señal binaria aleatoria con código de línea tipo RZ.

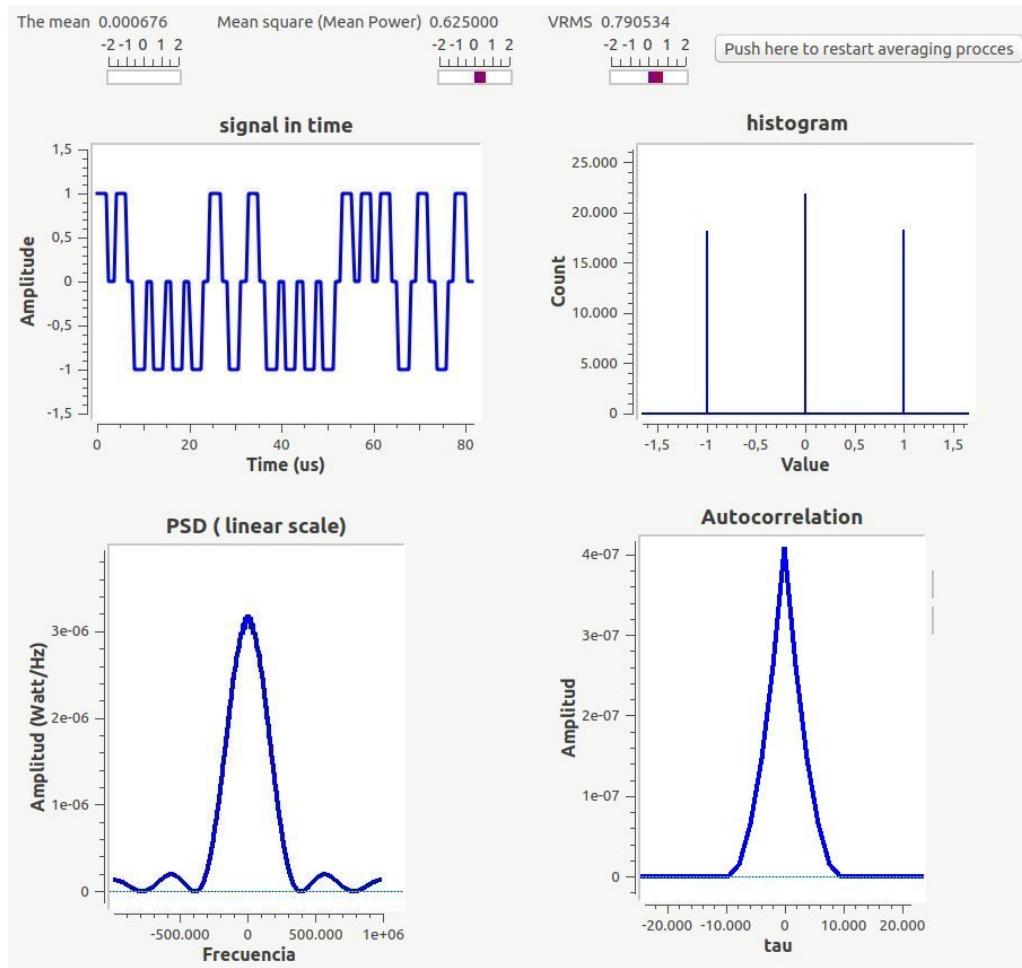


Figura 21: Promedios de tiempo. Señal binaria aleatoria con pulsos de forma sinc.

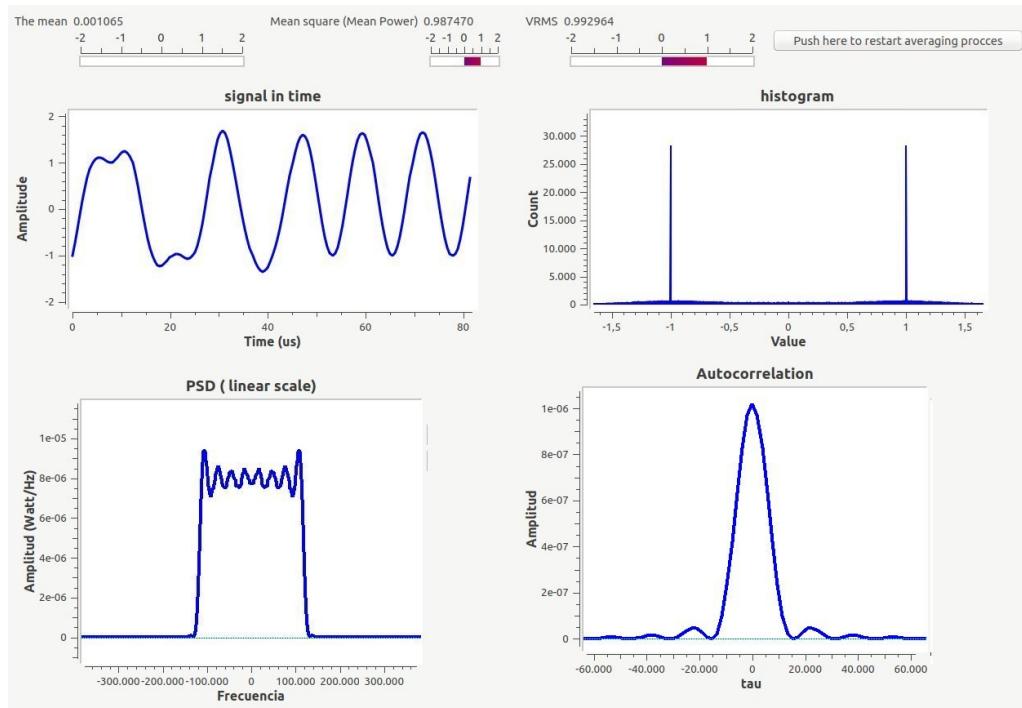
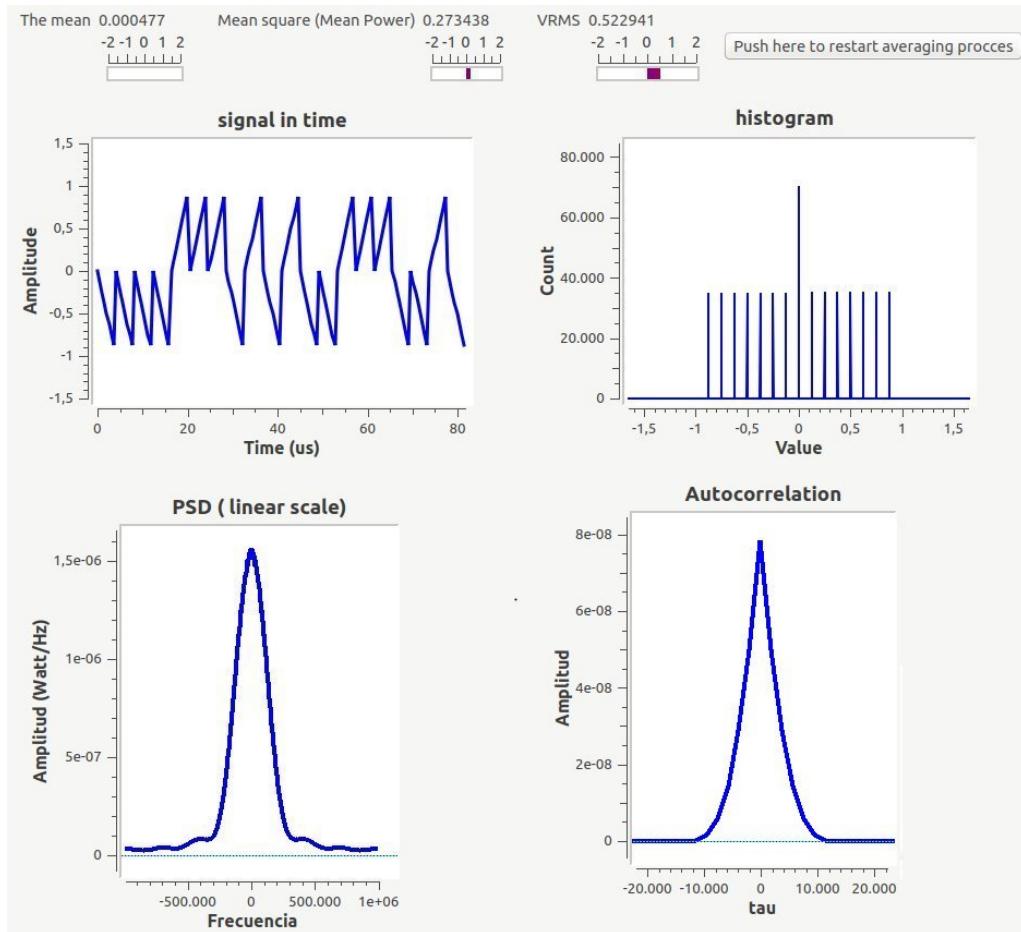


Figura 22: Promedios de tiempo. Señal binaria aleatoria con pulsos de forma diente de sierra.



A continuación se destacan las novedades incluidas:

- Se ha creado un nuevo “Python Module” en el cual se han programado varias formas deseadas para los pulsos de una señal binaria aleatoria. A continuación se presenta el código implementado, el cual contiene suficientes comentarios para comprenderlo.

```
# Nota : Este codigo esta guardado en forma de un python block en el bloque jerarquico b_help de la libreria de bloques jerarquicos comdiguis.

import math
import numpy as np
# Cada funcion siguiente es una forma de pulso para una senal digital
# Los parametros comunmente usados son:
# Sps (Samples per symbol): Duracion de un simbolo en tiempo discreto
# ntaps: es el numero de muestras a generar por simbolo. Esto debido a que,
# en la practica, un simbolo puede invadir a otros simbolos (ISI), asi que
# las muestras de un simbolo se pueden expandir mas alla de Sps

# Bipolar non return to zero level signal
def B_NRZ_L(Sps):
    return Sps*[1.,]
## Forma sinc
def sinc(Sps,ntaps):
    n=np.linspace(-int(ntaps/2), int(ntaps/2-1),ntaps)
    h=np.sinc(n/Sps)
    return h
# Forma diente de sierra
def saw(Sps):
    return np.linspace(0,Sps-1,Sps)
# Bipolar non return to zero signal
def RZ(Sps):
```

```

||| h=Sps*[1.,]
||| Sps_m=int(Sps/2)
||| h[Sps_m+1:Sps]=np.zeros(Sps-Sps_m)
||| return h

```

- Tenemos así, 3 tipos de señales implementadas por nosotros mismos:
 - Señal de forma NZ (del inglés *Return to Zero line code*).
 - Señal con pulsos en forma de señal *sinc()*.
 - Pulso en forma de señal dientes de sierra.
- La idea consiste en que con la ayuda del bloque “Random Source” se genera una señal discreta binaria aleatoria como las que se usan en software. Pero es claro que en software no se incurre en la insensatés de representar un bit mediante varias muestras, basta una muestra por bit. En el mundo físico en cambio los bits se representan mediante señales continuas, por ejemplo una señal de voltaje donde $5V$ es un “1” y $-5V$ es un “0”. Un generador vectorial busca generar señales del mundo físico, para ello debe producir varias muestras por símbolo, respetando el Teorema de Nyquist, esperando que sean vistas como algo parecido a las señales continuas o que pasen por un ADC que realmente las convierta en señales físicas. Para lograrlo, hemos estamos usando el mismo sistema LIT explicado anteriormente, representado en el bloque “Interpolating FIR Filter”, solo que esta vez, el parámetro “Interpolation” ya no es 1, sino Sps que es el número de muestras por símbolo (del inglés *Samples per symbol*).
- También se está usando una señal que se puede generar con un bloque que trae por defecto GNU Radio. La señal es ruido blanco gaussiano y el bloque es el “Noise Source”.
- Se incluye también el acceso a hardware, en este caso se trata de un micrófono para experimentar con las señales de voz y sonido en general. Esto mediante el bloque “Audio Source”.
- Finalmente se incluye un bloque para visualizar los promedios de tiempo de cualquier señal aleatoria. El bloque se llama “QT GUI b_v_aleatoria_scope” y su implementación se explica en el siguiente capítulo.

2.17.4. Implementación de un bloque de visualización de los principales promedios de tiempo

La idea es poder medir y visualizar diferentes promedios de tiempo a cualquier señal aleatoria. El flujo grama se presenta en la figura 18 y tiene las siguientes novedades:

- Se crea como un bloque jerárquico - una opción que ofrece GNU Radio para agrupar todo un flujo grama como un solo bloque.
- Se usa el bloque “mean_meter_ff” que es una mejora del ya explicado “e_mean_meter_ff” y es parte del OOT E3TRadio. Su objetivo es calcular la esperanza (la media) de una señal.
- Un bloque se pasa a un OOT, como lo hemos hecho con este bloque, cuando se considera que es un valioso aporte, que se va a usar continuamente en el futuro. Cualquier aporte en GNU Radio puede pasar a ser parte de una OOT. Podríamos decir que los bloques jerárquicos y los bloques embebidos que hemos implementado hasta ahora son de prueba, pues si se necesita para trabajos intensos, deberían ser traducidos para que sean parte de una librería OOT.

El caso de una señal de ruido blanco gaussiano

En la figura 19 se tienen los resultados para el caso de una señal de ruido blanco gaussiano. Como era de esperarse, la PSD tiende a ser una constante, el histograma tiende a ser una campana de Gaus, la potencia promedio en la figura 20 se tienen los resultado para el caso de una señal aleatoria binaria en la que los pulsos toman la forma del código de línea conocido como RZ.

El caso de una señal con código RZ Los resultados se presentan en la figura 20 es un caso interesante ya que es un código muy usado en las comunicaciones. Vemos una PSD muy similar a la obtenida para una señal binaria aleatoria bipolar.

El caso de una señal con pulsos de forma sinc Es un tipo de señal que también se usa mucho en las comunicaciones, consiste en que cada pulso se representa mediante una función sinc, pero resulta traslapándose con símbolos adyacentes en lo que se conoce como Interferencia Intersímbolo (ISI). Lo interesante es que aún así la información se puede recuperar, pues como vemos en el histograma, la amplitud 1 o -1 perseveran en cada símbolo. Pero lo más relevante es que gracias a esta forma, el ancho de banda

de la señal resultante es mucho menor que cuando se usan otras formas para las señales digitales. En este tema se profundiza más adelante.

El caso de una señal aleatoria con pulsos en forma de dientes de sierra: En la figura 22 se tienen los resultados. Hemos incluido esta señal por curiosidad y para demostrar que es posible seguir esta metodología para implementar un generador de cualquier tipo de señal aleatoria.

Capítulo 3

La Conversión RF-EC. Aplicación en la radiodifusión

Imaginemos que un atleta acaba de ganar el primer lugar en una competencia de atletismo y que al entrevistarlo nos asombra con una revelación, él es ingeniero, trabaja para una compañía y entrena sin salir de su oficina. Para ello usa una banda caminadora de manera que, mientras corre, está disponible para atender a cualquier persona que se acerca caminando. Cuando hablamos de señales RF nos referimos a las señales que usan el espectro radioeléctrico como medio de transporte, de manera más específica se habla de una onda portadora que sirve a ese propósito, pero la característica más importante de esa onda es su frecuencia y por eso se habla de la frecuencia portadora. Ese medio de transporte viaja a la velocidad de la luz cargando información de manera inalámbrica de un punto a otro. El uso de la EC equivale a cambiar la portadora por una de frecuencia cero sin que se afecte el el espectro que corresponde a la información, lo cual llamaremos en este libro - la Conversión RF-EC.

La Conversión RF-EC es un término que se está introduciendo en este libro, de manera que no es un término que pueda consultarse en otras fuentes, pero en realidad se refiere a un proceso bien conocido que consiste en convertir una señal RF a su correspondiente Envolvente Compleja (EC) y viceversa. La razón para introducir el concepto de Conversión RF-EC es meramente pedagógica, pero la Envolvente compleja es más que una teoría, es realmente parte de la tecnología SDR, en la cual, la componente de software solo distingue la EC de las señales que en el mundo físico viajan a altas frecuencias representadas en señales RF o señales paso bandas. La manera en que se define aquí la Conversión RF-EC ayuda a que el interesado pueda realizar una traducción muy sencilla entre lo que viaja físicamente entre las antenas y lo que realmente se puede ver usando técnicas de programación como GNU Radio.

En este capítulo también se avanza en la formación de competencias en GNU Radio para lo cual se plantea el reto de simular un sistema completo de radiodifusión FM, pero esta vez basado netamente en la EC.

3.1. la Envolvente Compleja y sus características

3.1.1. El problema a resolver

A continuación se enuncia un problema como ejemplo que justifica el uso de este concepto.

Deseamos implementar un analizador de espectros en un computador portátil mediante una aplicación basada en la FFT. Observar el espectro de la voz usando la FFT no representa un gran problema. Necesitamos conectar un micrófono al computador, el cual tiene un ancho de banda, por ejemplo 30 [kHz] que finalmente establece una frecuencia máxima para la voz. Cumpliendo el Teorema de Nyquist, la señal de voz deberá ser muestreada a una frecuencia

$$f_s \geq 2f_{max} = 30[\text{kHz}] \quad (3.1)$$

El problema se complica cuando se desea analizar una señal que proviene de una radio base de un operador móvil. Será necesario conectar un receptor, el cual convierte la señal electromagnética a una señal eléctrica amplificada de manera suficiente.

cambiar el ejemplo por el de ciclismo, usando una bicicleta sobre rodillos. Esto porque el giro de las ruedas puede parecer a la envolvente compleja

Falta explicar que la Envolvente compleja también se conoce como señal analítica. También sería bueno cerrar el capítulo con algo que permita integrar la explicación ofrecida aquí con la que aparece en otras fuentes, donde se habla del pre-envolvente de Transformada de Hilbert, de manera que quede muy claro que nuestra explicación es perfectamente coherente con las demás. En la industria y en las empresas de telecomunicaciones, el término más usado quizás no es la Envolvente Compleja, sino lo que se conoce como señal I-Q y esto también falta dejarlo claro en este libro y que a muchas personas les parece más sencillo entender que una señal de radio podría ser representada mediante dos señales, una I y una Q, en vez de pensar en una señal que compleja

Figura 1: Analizador de espectros basado en computador



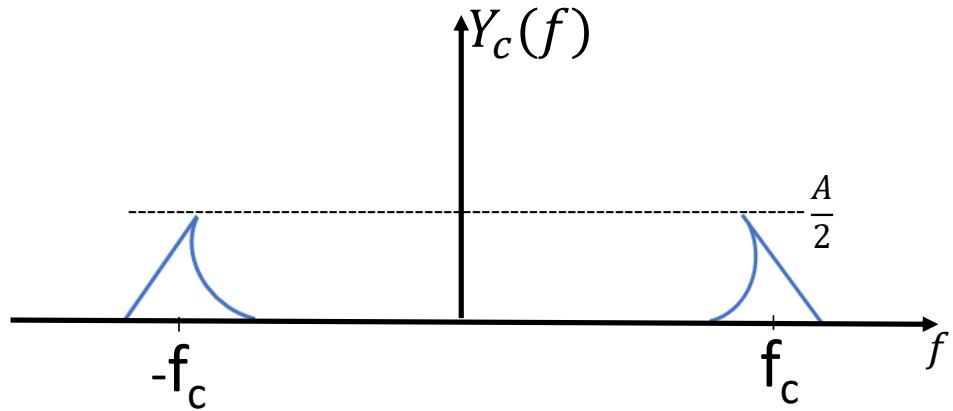
Pero ahora, la señal que queremos analizar tiene una frecuencia máxima muchísimo mayor, por ejemplo 1900000000 [Hz] ya que, en nuestro caso estamos recibiendo una señal RF. En este caso el muestreo sí que resulta extremadamente costoso y casi imposible de implementar, pues estamos hablando de una frecuencia de muestreo de al menos 3,8 [GHz]. Desde otro punto de vista, la FFT debe desarrollar tantas operaciones que el espectro no podrá ser visto en tiempo real, incluso quizás el software que tengamos disponible no acepte ese valor de muestreo y si lo hace también es posible que el computador se bloquee tan pronto arranque la aplicación.

3.1.2. La solución: la Envolvente Compleja

La solución lógica consiste en incluir dentro del receptor un dispositivo que permita desplazar el espectro de la señal de interés para centrarlo en una frecuencia mucho más baja a la que estaba centrado originalmente, lo cual se conoce como una operación de heterodinado. El caso más extremo se tiene cuando el receptor logra centrar el espectro en $f_c = 0$ [Hz].

A manera de ejemplo, en la figura 2 se tiene una forma hipotética de la Transformada de Fourier de una señal paso bandas, como es el caso de una señal RF.

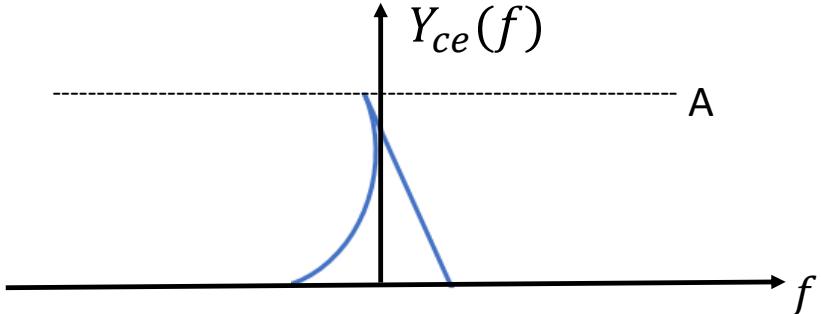
Figura 2: Espectro de la señal pasobandas de interés



En la figura 2 el espectro es simétrico alrededor de la frecuencia $f = 0$ [Hz], por lo tanto pertenece a una señal real, pues es bien sabido que la TF de una señal real es simétricamente conjugada. Sin embargo, alrededor de la frecuencia f_c el espectro no es simétrico, lo cual es usual en las comunicaciones. En la figura 3 se muestra el resultado al que queremos llegar, donde el espectro aparece centrado en $f_c = 0$ [Hz]. Pero ahora el espectro tiene doble altura con respecto al de la figura 2 con el fin de que conserve la misma energía. Además, vemos que ahora el espectro no es simétrico, lo que significa que esa operación ha

afectado a la señal en el tiempo y ahora lo que tenemos es una señal compleja que llamaremos Envolvente Compleja .

Figura 3: Espectro de la Envolvente Compleja



Puede decirse que la Envolvente Compleja es la señal que resulta en el dominio del tiempo cuando tomamos el espectro de una señal paso bandas de interés, que está centrado en una frecuencia $f_c > 0$, y lo desplazamos de forma que quede centrado en la frecuencia $f_c = 0$ Hz, sin que se afecte la forma del espectro y por lo tanto sin que se afecte la información que está contenida en él.

3.2. Conversión RF-EC en el dominio del tiempo

En varias partes del libro estaremos usando el concepto de Conversión RF-EC que facilita enormemente el proceso de buscar la versión banda base, es decir, la Envolvente Compleja de una señal pasobandas. Para lograr esta conversión se ha tenido en cuenta que:

- La búsqueda de la Envolvente Compleja solo es de interés de señales pasobandas, y una señal pasobandas siempre está representada por una onda senoidal. En contraste, la versión bandabase siempre se expresa como una función Euler compleja.
- En el capítulo anterior se analizaron las tres posibles opciones más generales en que una onda senoidal puede ser modulada: modificando su amplitud, su frecuencia o su fase.
- También se demostró, que una modificación a la frecuencia se traduce necesariamente en una modificación a la fase, como se explicó al obtener la ecuación (??) para la modulación FM.

De todo lo anterior, se deduce que, la expresión más genérica para una onda senoidal modulada en versión pasobandas y para su correspondiente versión bandabase son las que se muestran a la izquierda y a la derecha respectivamente de la siguiente expresión:

$$A(t) \cos [2\pi f_{ct} t + Q(t)] \xrightarrow{\text{Conversion RF-EC}} A(t)e^{jQ(t)} \quad (3.2)$$

La clave para usar esta conversión consiste en identificar cual es el parámetro que está siendo modificado por la señal de información para lograr mantener ese parámetro pero dentro de una función Euler. Puede darse el caso en que el parámetro modificado es la frecuencia, pero este también está implícito en la expresión anterior ya que una modificación a la frecuencia puede ser expresada en una modificación a la fase, como se muestra a continuación.

$$f(t) = d\phi/dt \quad (3.3)$$

Por lo tanto

$$\phi(t) = \int_0^t f(\tau)d\tau \quad (3.4)$$

De manera que si se observa que la información modifica (modula) la frecuencia, es necesario traducir ese efecto a un cambio en la fase. para luego aplicar la conversión.

3.2.1. Conversión EC-RF en el dominio del tiempo

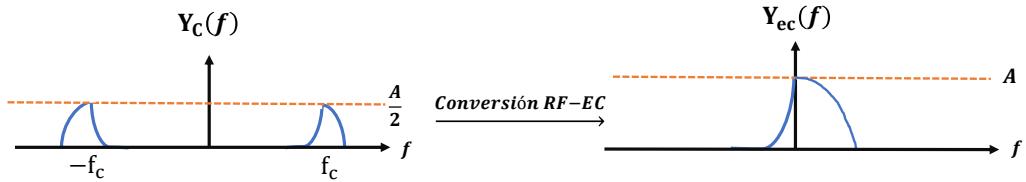
La Conversión inversa no es más que el proceso contrario de la Conversión Directa y puede ser expresada así:

$$A(t)e^{j[Q(t)]} \xrightarrow{\text{Conversion EC-RF}} A(t) \cos [2\pi f_c t + Q(t)] \quad (3.5)$$

3.2.2. La Conversión RF-EC en el dominio de las frecuencias

Aunque ya quedó explicado este caso, hemos querido formalizar la Conversión RF-EC en el dominio de las frecuencias de la manera que se presenta en la figura 4.

Figura 4: La Conversión RF-EC en el dominio de las frecuencias



3.2.3. Ejercicio sobre la Conversión RF-EC

El siguiente ejercicio ayuda a reforzar aún más los conceptos vistos hasta el momento. Se requiere encontrar una expresión para la Envolvente Compleja de una señal con Modulación FM.

Solución: Ya conocemos por la ecuación ?? que la expresión matemática de una señal pasobandas FM es:

$$s(t) = A_c \cos [2\pi f_c t + 2\pi K_f \int_0^t m(\tau) d\tau] \quad (3.6)$$

Donde, A_c es la amplitud de la portadora, f_c es la frecuencia de la portadora, K_f es el coeficiente de modulación, $m(t)$ es el mensaje y t es la variable de tiempo.

Comparando con la definición de la Conversión RF-EC en 3.4 tenemos que:

$$A(t) = A_c$$

y la fase de una señal FM varía de acuerdo a la siguiente ecuación

$$Q(t) = 2\pi K_f \int_0^t m(\tau) d\tau \quad (3.7)$$

Por lo tanto, aplicando la Conversión RF-EC obtenemos la expresión para la Envolvente Compleja de una señal con Modulación FM:

$$S_{ec}(t) = A_c e^{j2\pi K_f \int_0^t m(\tau) d\tau} \quad (3.8)$$

que es lo mismo que:

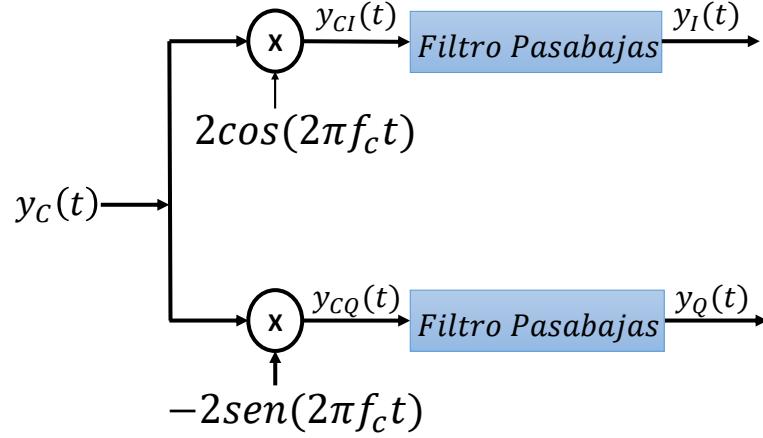
$$S_{ec}(t) = A_c \cos [2\pi K_f \int_0^t m(\tau) d\tau] + j A_c \sin [2\pi K_f \int_0^t m(\tau) d\tau] \quad (3.9)$$

Podemos concluir que la Envolvente Compleja de una señal FM es **un vector rotante**.

3.3. El Down Converter

Para obtener la Envolvente Compleja de una señal pasobandas, se usa un dispositivo que llamaremos “Down converter” y que se presenta en la figura 5.

Figura 5: El Down Converter



la Envolvente Compleja es la señal $y_{ec}(t) = y_I(t) + jy_Q(t)$, $y_I(t)$ es conocida como la componente en fase, $y_Q(t)$ como la componente en cuadratura de la Envolvente Compleja .

3.3.1. Demostración del Down Converter

La Transformada de Fourier es una herramienta útil para diseñar y demostrar la validez de este tipo de soluciones.

$$y(f) = \frac{1}{2}[y_{ec}(f - f_c) + y_{ec}(-f - f_c)] \quad (3.10)$$

$$y_{CI}(t) = y_c(t)2\cos(2\pi f_c t) \implies y_{CI}(t) = y_c(f - f_c) + y_c(f + f_c) \quad (3.11)$$

$$y_{CI}(f) = \frac{1}{2}[y_{ec}(f - 2f_c) + y_{ec}(-f) + y_{ec}(f) + y_{ec}(-f - 2f_c)] \quad (3.12)$$

$$y_I(f) = \frac{1}{2}[y_{ec}(-f) + y_{ec}(f)] \quad (3.13)$$

$$y_{CQ}(t) = -2y_c(t)\sin(2\pi f_c t) \implies TF y_{CQ}(f) = \frac{-1}{2j}[y_c(f - f_c) - y_c(f + f_c)] \quad (3.14)$$

$$y_{CQ}(f) = \frac{-1}{2j}[y_{ec}(f - 2f_c) + y_{ec}(-f) - y_{ec}(f) - y_{ec}(-f - 2f_c)] \quad (3.15)$$

$$y_Q(f) = \frac{1}{2j}[y_{ec}(f) - y_{ec}(-f)] \quad (3.16)$$

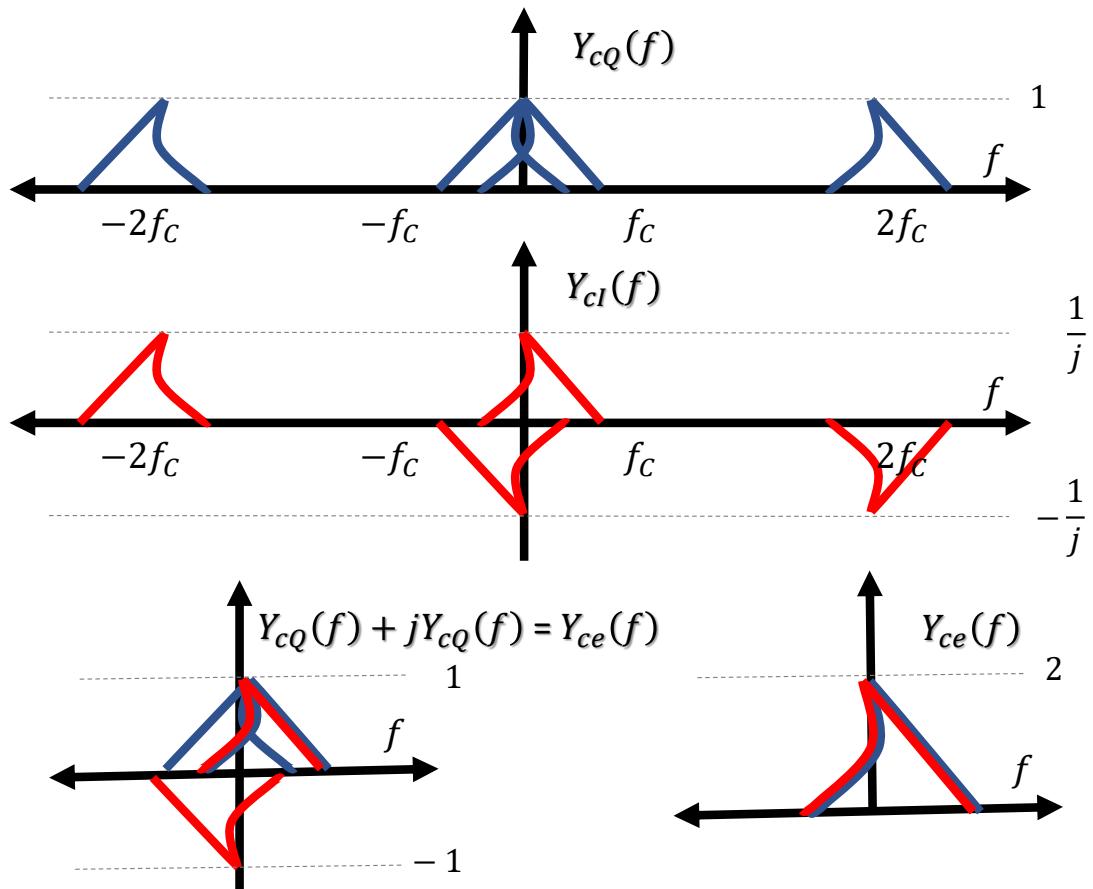
De modo que, el sistema de la figura 5 permite obtener la componente en fase y cuadratura de la Envolvente Compleja $y_{ec}(t) \implies y_{ec}(f)$.

Parece haber error en la fórmula anterior, cuando aparece TF, no queda claro que sea la Trasformada de Fourier de algo

3.3.2. Demostración gráfica del Down Converter

Aplicaremos el Down Converter de la figura 5 a la señal que tiene la TF de la figura 2. Deberíamos obtener la TF de la Envolvente Compleja que es la que aparece en la figura 3.

Figura 6: Resultado del uso del Down Converter



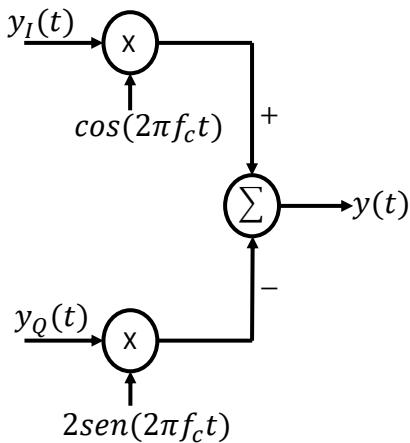
En la figura anterior, falta correr f_c un poco a la derecha para que quede claro que está en medio de cero y $2f_c$

3.4. El Up Converter

La Conversión RF-EC es un concepto teórico. El Up Converter es el Dispositivo que aprovecha este concepto para realizar la Conversión EC-RF y se presenta en la figura 7

[Hay un error en la Figura, porque en la parte inferior la portadora aparece multiplicada por 2, pero no en la parte superior. La corrección que hay que hacer es que en la parte de abajo no aparezca el 2]

Figura 7: Up converter



En la figura anterior hay un error porque la portadora en la parte inferior está escalada en 2 (dicho 2 sen(...)), debe ser como la parte superior donde cos(...) no está escalado en 2

3.5. Simulación en banda base de un sistema de radiodifusión

HOB: tenemos la idea de convertir este subcapítulo en un nuevo capítulo que tendrá los siguientes fines:

- tendrá dos partes: la primera consiste en proponer una metodología de programación en gnuradio; la segunda consiste en usar esa metodología en la implementación de un sistema completo de radiodifusión simulada, por ahora.
- sobre la metodología (queremos tambien publicarla en un artículo), la idea es incluir varios items como:
 - El estudio de requerimientos: la idea es que no se comience la implementación de un sistema de comunicaciones a la loca, sino haciendo ingeniería de software, sea esta muy sencilla o compleja. En este sentido, todo comienza por la identificación de los requerimientos. Estos pueden ser de usuarios, pero tambiéñ técnicos y puede ir acompañada de bastante teoría pues es la teoría la permite identificar parámetros a usar y prever el uso de recursos, como el tipo de modulación, el canal, etc.
 - Diseño: Todo comienza por distinguir las diferentes partes de la implementación como:
 - seguir una metodología que haga que la solución pueda ser entendida por ingenieros de sistemas para adaptarla a soluciones más profesionales. En este sentido, podemos dar una visión de hasta donde pueden llegar las soluciones, por ejemplo las de radiocognitiva donde hay similitud con las soluciones IoT: hay dispositivos que vigilan el espectro, bases de datos por consultar, sistemas de comunicación que funcionan no sin antes consultar esas bases de datos o incluso reportarse a ellas.
 - el flujo de datos del sistema de comunicación puro. Para esta parte nos apoyaremos en modelo de capas donde definiremos una serie de criterios como: uso de tipos de líneas (para mostrar los elementos que son hardware, los que son software), colores (para diferenciar señales reales de complejas o incluso lógicas (unos y ceros) de las señales de amplitud (señales continuas o de origen continuo o analógico)), colores de flechas (para lo mismo anterior); el uso de unas capas básicas: canal, capa base, aplicación
 - las componentes de visualización, que pueden estar en el mismo flujo de datos, pero debemos prever la posibilidad de que esa parte sea remota como en la tendencia en la vida real: instrumentos son una cosa: sistema es otra.
 - posibles bases de datos. Nuestras soluciones pueden comenzar a usar información de archivos, pero debemos prever la posibilidad futura de ver las soluciones de SDR como parte de las IoT, sin entrar en detalle en ellas, lo cual nos permite prever una buena metodología de desarrollo.

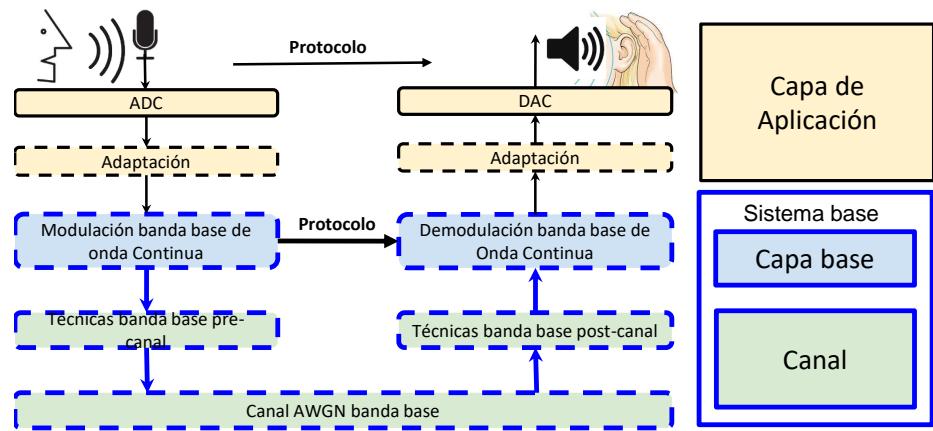
- sobre el sistema completo de radiodifusión, está claro que lo que queremos es lograr dos cosas a la vez: implementar la metodología y aprender sobre el sistema de radiodifusión.

Ya se comprobó que no tiene sentido implementar la simulación de un sistema de comunicaciones pasobandas debido a que resultaría irrealizable si se usan parámetros de frecuencias del mundo real, ya que el principal obstáculo está en la necesidad de un muestreo que superaría las capacidades de cómputo y de muestreo de los computadores de precio razonable. Lo que sí tiene sentido es implementar la simulación de una versión banda-base, la cual permite lograr experimentar cualquier técnica que se desee implementar en versión pasobandas.

3.5.1. Modelo de capas e identificación de requerimientos

En calidad de ejemplo, vamos a implementar la simulación de un sistema de radiodifusión lo más completo posible, para lo cual nos apoyaremos en el modelo de capas de la figura 8.

Figura 8: Modelo de capas para la simulación de un sistema de radiodifusión



La idea consiste en implementar un sistema base de radiodifusión, sobre el cual se puede montar una capa de aplicación para prestar el servicio a diferentes necesidades como por ejemplo conducir información de voz, o cualquier otra que sea de naturaleza analógica.

Un modelo como estos, define unos requerimientos que facilita el desarrollo que deseamos implementar, los analizaremos capa por capa, desde abajo hacia arriba. Es importante destacar que es completamente necesario comenzar con la implementación de un sistema simulado antes de pasar a implementar un sistema real basado en SDR, ya que en un sistema simulado es posible probar diversas técnicas para enfrentar los desafíos que un canal inalámbrico puede presentar. Por ejemplo, en un sistema simulado es posible probar de manera aislada una solución para enfrentar el ruido blanco, otra para enfrentar el Efecto Doppler y así sucesivamente hasta contar con una solución que puede enfrentar todos los principales fenómenos de propagación.

Los aspectos generales del modelo propuesto en la figura 8:

- Los bloques que aparecen con líneas punteadas son bloques que se implementan mediante software, mientras que los de líneas continuas son componentes de hardware, que en nuestro caso, son elementos del hardware del computador en el cual vamos a implementar la simulación.
- Los bloques y las flechas que están resaltadas en color azul manejan el concepto de Envolvente Compleja , por lo tanto trabajan con señales complejas. El caso de la modulación, la entrada es real y la salida es compleja, en el de la demodulación la entrada es compleja y la salida es real. Aquí es donde está la gran diferencia con la figura ??, donde no habían bloques con señales complejas.
- La simulación deberá correr en tiempo real, es decir, podremos estar hablando por un micrófono del computador, mientras al mismo tiempo podemos estar escuchando fluidamente lo que llega a la bocina.

Para la capa del canal tenemos que:

- En el mundo real, el canal es algo físico de naturaleza continua. En nuestro caso no porque estamos implementando un simulador que nos permite tener en cuenta solo el efecto del ruido blanco en el canal.
- Al tener en cuenta el efecto del ruido blanco del canal, estamos atacando principalmente el fenómeno de desvanecimiento que sufren las ondas en sus propagación, ya que en la medida en que las ondas se atenúan, se acentúa la influencia del ruido.
- Deberemos implementar una versión banda base del ruido blanco, ya que la señal que vamos a transmitir por ese canal está también en versión bandabase, es decir en términos de la Envolvente Compleja . Eso significa que debemos pensar en la Envolvente Compleja del ruido blanco.
- Se deduce entonces, que la sub-capa que hemos llamado Técnicas pre-canal/post-canal se aplican a señales que están convertidas a su Envolvente Compleja y se enfocan principalmente en el ruido blanco.
- Lo anterior no limita las posibilidades de implementar elementos adicionales al canal y a las técnicas pre-canal para comprobar cómo puede afectar por ejemplo las desviaciones de fase, frecuencia, las no linealidades y cómo corregirlas.

Para la capa de Modulación tenemos que:

- El modelo se presta para que implementemos un sistema de radiodifusión en el que podamos cambiar fácilmente el método de modulación, para tener por ejemplo un sistema de radiodifusión AM, DSB, VSB, FM, PM, etc.
- El problema de usar diferentes tipos de modulación está en que tenemos dos grandes tipos de modulación: las que son de tipo lineal como AM, DSB, VSB; y las que no son lineales como FM y PM. En este caso, las técnicas pre-canal pueden ser diferentes para esos dos casos.

Para la capa de aplicación tenemos que:

- Se trata del sistema de información que aprovecha a nuestro sistema de comunicación.
- En este caso nos planteamos el reto de enviar señales de audio que captura el micrófono del computador y recibirlas, para escucharlas en la bocina del mismo.
- El ADC y el DAC se refieren a las tarjetas que el computador trae para realizar el proceso de conversión analógica digital y digital analógica.
- La capa de adaptación está allí por varias razones, en primer lugar, como estamos implementando la simulación de un sistema que por naturaleza es analógico, la señal que entrega el ADC la veremos como una señal muestreada, no como una señal convertida a bits. Además, es muy probable que necesitemos realizar unas operaciones previas a la señal del audio bien sea porque las limitaciones del canal así lo impongan o porque queramos adaptarnos a lo que establece algún estándar de radiodifusión.

3.5.2. El Canal AWGN banda base

Como podemos ver en la figura 8 lo que se entrega al canal es una EC, de manera que si vamos a simular el canal, este debe interactuar con nuestra EC en condiciones similares, por lo tanto, el ruido que afecta a la EC también debe estar en versión compleja. Aun cuando esto no sea un simulador, tenemos que operar con la EC del ruido ya que el USRP captura ruido en pasobandas, pero su Down Converter lo entrega en forma de EC.

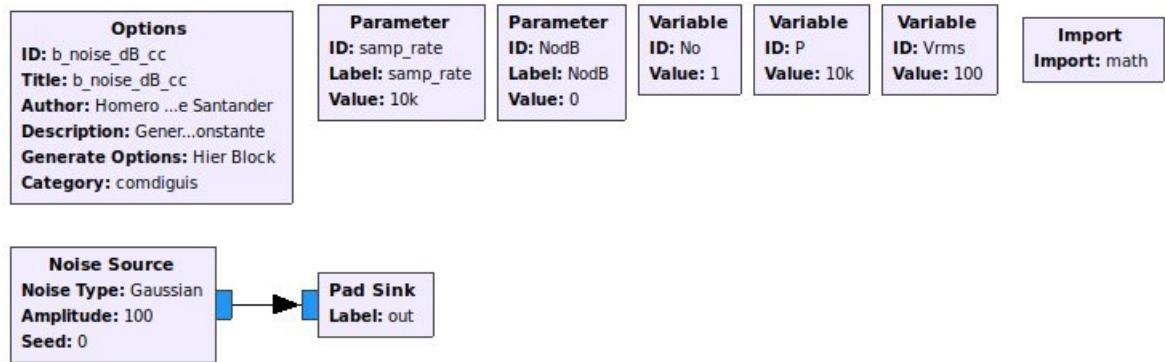
3.5.2.1. El ruido blanco en versión banda base

En el capítulo ?? se mostró una implementación en GNU Radio de un flujo gráfico que permite generar y comparar el ruido blanco de banda ancha con el ruido blanco de banda angosta o NBWN. Varios detalles de lo que implica este tipo de implementación fueron analizados allí. El reto ahora, es implementar algo similar pero el objeto de interés es la Envolvente Compleja del ruido blanco. El Ruido Blanco tiene la particularidad de estar presente en todo el espectro, por lo tanto es al mismo tiempo de naturaleza pasobandas y banda base. Sin embargo, cuando hablamos de versión banda base, nos referimos a la señal que se obtiene cuando el ruido blanco pasa por un Down Converter. La señal que resulta debe estar compuesta por una parte real y una imaginaria y es esencial para poder implementar un canal AWGN banda base. Los aspectos a tener en cuenta en la simulación de la Envolvente Compleja del ruido blanco son los siguientes:

- Cualquier canal y cualquier elemento usado en los sistemas de comunicación, como las antenas, los filtros, etc, tienen un ancho de banda limitado. De modo que, aunque el ruido blanco se supone que tiene un ancho de banda infinito, solo debe interesarnos simularlo en el ancho de banda de interés. En el fondo, lo que afecta a un sistema de comunicaciones es el NBWN, pero, a diferencia del NBWN visto en el capítulo ??, aquí estamos buscando la versión banda base del NBWN, o sea, la Envolvente Compleja del NBWN.
- Lo que nos interesa generar es equivalente a hacer pasar una señal de ruido blanco por un Down Converter y no hay que olvidar que este dispositivo incluye por defecto Filtros pasobajas de banda angosta.

la Envolvente Compleja del NBWN puede ser vista como aquello que resulta al aplicar la Conversión RF-EC al ruido, en el dominio de las frecuencias, de acuerdo a lo explicado en la figura 4. Para analizar esto en GNU Radio, hemos implementado el flujograma de la figura 9, en la cual podemos apreciar lo siguiente:

Figura 9: Simulación en GNU Radio de la Envolvente Compleja de ruido blanco



Flujograma usado: *b_noise_dB_cc.grc*

- GNU Radio ya incorpora un bloque que genera el ruido blanco en versión de Envolvente Compleja , se trata del bloque “Noise Source” y es el mismo usado para generar ruido blanco, solo que debe ser configurado en modo de salida compleja, lo cual hace que la salida del bloque cambie a color azul.
- El problema de ese bloque es que el único parámetro de configuración no dice mucho sobre las características requeridas en una aplicación más práctica, por esa razón, en la figura 9 se presenta el flujograma de un bloque jerárquico llamado “*b_noise_dB_cc*” que genera el ruido en forma de EC en función de unas necesidades concretas de la frecuencia de muestreo F_s y del valor $N_0 dB$. Los demás parámetros se deducen de la siguiente manera:

Ancho de banda del ruido:

$$BW = F_s/2 \quad (3.17)$$

El valor N_0 :

$$N_0 = 10^{N_0-dB/10} \quad (3.18)$$

La Potencia promedio:

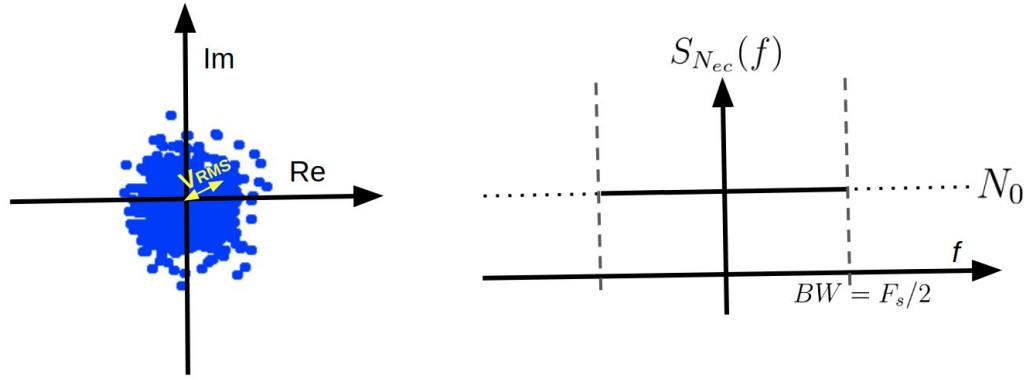
$$P = N_0 * 2 * BW \quad (3.19)$$

El Valor RMS:

$$V_{RMS} = \sqrt{P} \quad (3.20)$$

- De manera similar, los demás bloques que acompañan el procesamiento o la visualización deben ser configurados como de tipo complejo. Por eso sus entradas y/o salidas están en color azul en la figura 9.

Figura 10: Ruido Blanco Bandabase en el Diagrama Polar

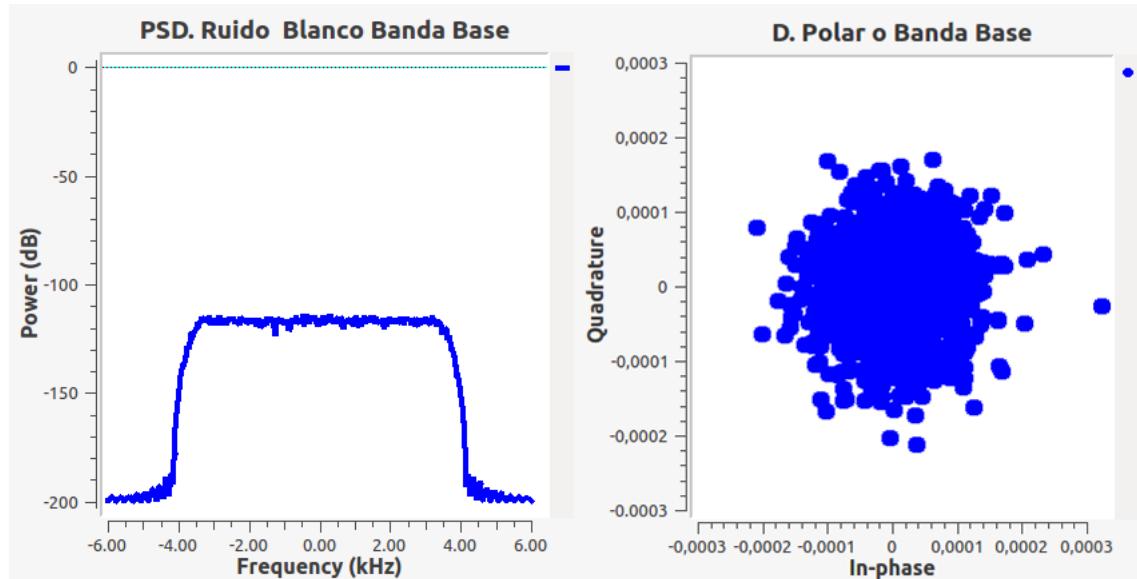


- Como se muestra en la figura 10 el ruido blanco banda base puede ser visto en el diagrama polar como una nube centrada en cero y con una magnitud promedio igual al Valor RMS V_{RMS} . Como resultado de la aplicación de la Conversión RF-EC en el dominio de las frecuencias la PSD se eleva dos veces y su altura es N_0 como se muestra en la parte derecha de la figura 10
- El parámetro Amplitud, en el bloque **Noise Source** corresponde al valor RMS v_{rms} del ruido
- El bloque Rational Resampler se agregó como parte de los equipos de visualización, solo con el fin de observar qué hay más allá del ancho de banda bandabase, que en este caso es $BW = 4$ [[kHz]]. Hay que entender, que el bloque **Rational Resampler** es en realidad un Filtro con capacidades de interpolación, aunque dependiendo de la configuración también puede decimar.

Los resultados de esta implementación aparecen en la figura 11. A la izquierda tenemos la PSD de la Envolvente Compleja del ruido blanco de banda angosta y a la derecha la gráfica en el dominio polar.

Los resultados muestran que, con el filtrado, la señal en el tiempo sufre una caída en amplitud y su forma se vuelve más suave, pero se conserva la altura de la PSD.

Figura 11: Ruido Blanco Banda Base, su PSD y representación Polar



3.5.2.2. Un canal de ruido blanco gaussiano aditivo en versión bandabase

El canal de ruido blanco gaussiano aditivo es mejor conocido en inglés como Additive White Gaussian Noise Channel (AWGN Channel). Es bien sabido que una señal en su viaje desde una antena transmisora hasta una receptora atraviesa una gran cantidad de fenómenos. Es como si la señal entrara a un sistema

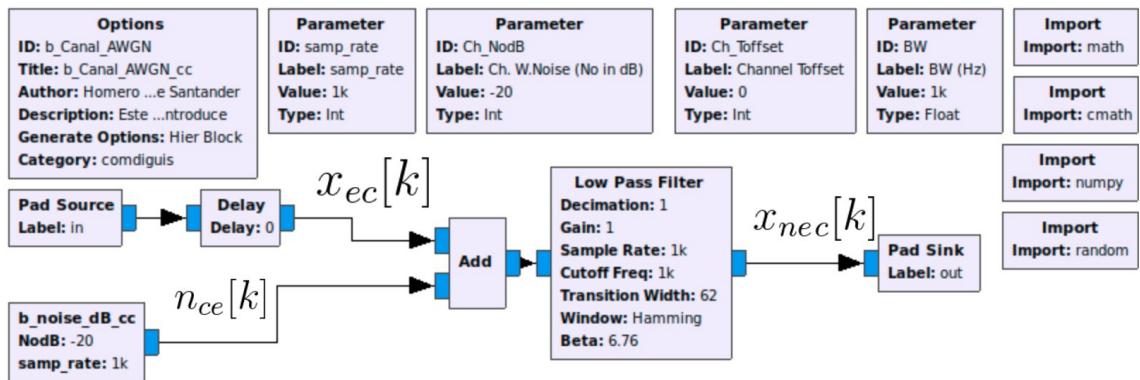
que la deforma de varias maneras y ese sistema es lo que llamamos canal de propagación. Uno de ellos es el ruido y precisamente nos preparamos para revisar cómo luchar contra el efecto que este produce. Es muy claro que si vamos a probar un método para atenuar el ruido blanco, resulta importante poder aislar todos los demás fenómenos que pueden existir en el canal de propagación, de manera que el canal solo tenga la posibilidad de injectar a la señal ruido blanco. En el capítulo ?? ya se presentó una implementación de este canal de propagación, lo que se busca ahora es la implementación de la versión banda base.

Para la implementación es importante tener en cuenta que la salida $x_{nec}[k]$ de un canal AWGN en versión banda base ante una entrada $x_{ec}[k]$ es $x_{nec}[k] = x_{ec}[k] + n_{ce}[k]$.

Donde $n_{ce}[k]$ es la Envoltura Compleja en versión discreta del ruido blanco de banda angosta $n(t)$.

En la figura 12 se tiene la implementación de un bloque jerárquico que es parte de la librería comdiguis para un canal de ruido blanco gaussiano aditivo.

Figura 12: Implementación en GNU Radio de un Canal de ruido blanco aditivo

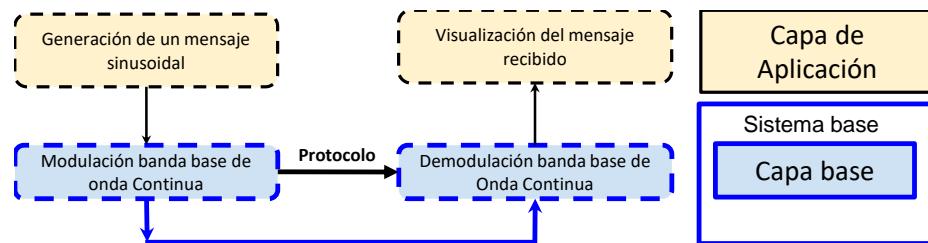


Flujograma usado: `b_Canal_AWGN.cc.grc` Nota: es parte de la librería comdiguis

3.5.3. La Modulación FM Banda Base

La implementación de la capa de Modulación FM es relativamente sencilla, sin embargo vale la pena dedicar esfuerzo a su análisis por tratarse de una Modulación que no es lineal, de modo que el espectro se ensancha sustancialmente con respecto al espectro del mensaje. Para realizar la implementación es conveniente comenzar con un modelo de capas simplificado como el propuesto en la figura 13.

Figura 13: Modelo de capas simplificado para implementar la modulación



A continuación, se realiza un resumen sobre los aspectos conocidos de la Modulación FM que resultan coherentes para una implementación basada en SDR. El siguiente resumen se apoya principalmente en el libro de Haykin [3]: ya se ha dicho que un Modulador FM es un caso especial de un Modulador PM y no es un sistema lineal, la consecuencia es que el espectro de una señal con Modulación FM es mucho más ancho que el espectro del mensaje. En este sentido, lo relevante es que una emisión FM ocupa un ancho de banda mucho mayor que una emisión AM.

Varios parámetros caracterizan una señal con Modulación FM:

- El coeficiente de sensibilidad, usualmente representado como K_f el cual equivale al número de veces que es necesario amplificar la señal mensaje para lograr el índice de Modulación requerido.
- La desviación de frecuencias, usualmente representada como Δf . Es una variable que toma un valor cercano al ancho de banda que la señal Modulada alcanza y está dada por:

$$\Delta f = K_f A_m \quad (3.21)$$

donde A_m es el valor máximo que puede alcanzar la amplitud del mensaje o señal modulante.

- Otros parámetros son el Ancho de Banda y el Índice de Modulación pero su forma de calcular se explica más adelante.

3.5.3.1. Caso en que el mensaje es reemplazado por una señal sinusoidal

La respuesta de cualquier Modulador de Onda continua cuando en calidad de mensaje se usa una señal sinusoidal sirve como una forma de caracterización sobre todo en el dominio de las frecuencias, pues a partir de esa respuesta es posible conocer cómo puede responder el modulador ante una auténtica señal mensaje. En este sentido, vale la pena considerar que la señal mensaje es la siguiente:

$$m(t) = A_m \cos(2\pi f_m t) \quad (3.22)$$

En este sentido, la Modulación FM tiene las siguientes características adicionales:

- **El índice de Modulación:** Corresponde a la relación entre la desviación de frecuencias Δf y la frecuencia de modulación f_m

$$\beta = \frac{\Delta f}{f_m} = \frac{K_f A_m}{f_m} \quad (3.23)$$

- Con la introducción del parámetro β es posible hablar de dos tipos de Modulación FM: La modulación FM de Banda Angosta, cuando β es pequeño en comparación a un radian; La Modulación FM de banda ancha, cuando β es grande comparado con un radian. Sin embargo la Modulación FM de Banda Angosta no tiene mucho sentido cuando se usa tecnología SDR ya que ella es más bien parte de las estrategias usadas para construir Moduladores FM mediante tecnología analógica, mientras que en las técnicas de SDR se usan métodos digitales.
- Para conocer el espectro que en este caso resulta, es necesario realizar el siguiente análisis. Tenemos que

$$S_{ec}(t) = A_c e^{j2\pi K_f A_m \int_0^t \cos(2\pi f_m \tau) d\tau} \quad (3.24)$$

de modo que

$$S_{ec}(t) = A_c e^{j\beta \sin(2\pi f_m t)} \quad (3.25)$$

Por lo tanto, $S_{ce}(t)$ resulta siendo una señal periódica en el tiempo y con una frecuencia fundamental igual a f_m . Recordemos que, de acuerdo a la RSF (??), toda señal periódica puede ser representada mediante la Ecuación de Síntesis de la RSF, por lo tanto:

$$S_{ec}(t) = \sum_{-\infty}^{\infty} C_k e^{j(2\pi k f_m t)} \quad (3.26)$$

donde C_k son los Coeficientes de Fourier y se hallan mediante la Ecuación de Análisis de la RSF (??), por lo tanto, tenemos que:

$$\begin{aligned} C_k &= \frac{1}{T_m} \int_{-T_m/2}^{T_m/2} S_{ec}(t) e^{-j2\pi k f_m t} dt \\ &= \frac{1}{T_m} A_c \int_{-T_m/2}^{T_m/2} e^{j[\beta \sin(2\pi f_m t) - 2\pi k f_m t]} dt \end{aligned} \quad (3.27)$$

Reemplazando por una nueva variable:

$$x = \frac{2\pi t}{T_m}$$

Llegamos a la siguiente expresión

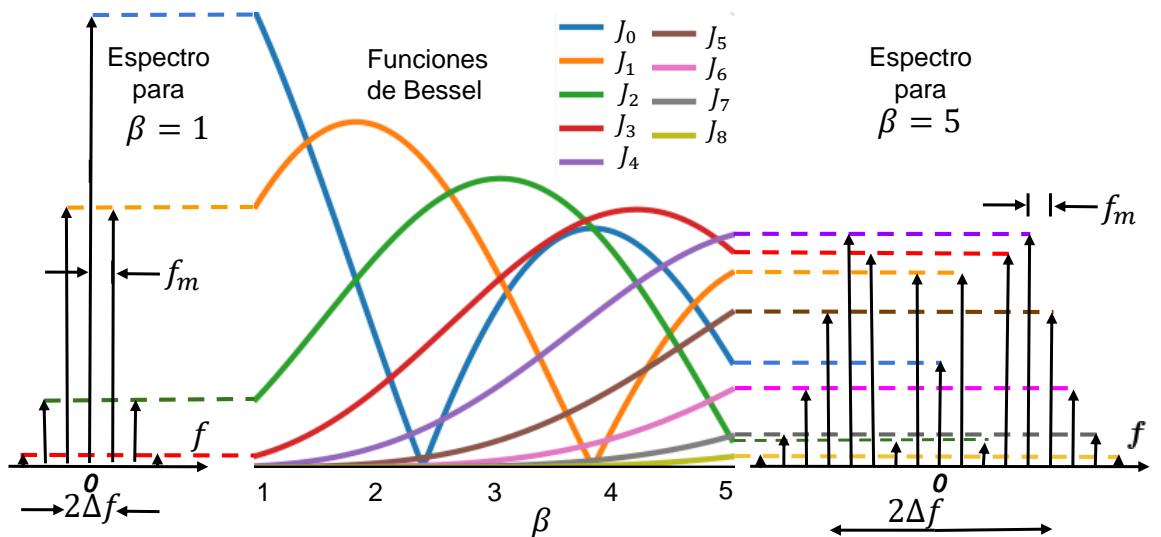
$$C_k = A_c \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{j[\beta \sin x - kx]} dx$$

La parte derecha de la ecuación anterior coincide con la definición de la Función de Bessel de primera especie orden k , por lo tanto:

$$C_k = A_c J_k(\beta) \quad (3.28)$$

- Todo el análisis anterior sirve para afirmar que una señal FM obtenida usando un coeficiente de sensibilidad K_f y de un mensaje sinusoidal de amplitud A_m y frecuencia f_m tiene un índice de modulación β que determina la forma del espectro a partir de las Funciones de Bessel, se obtiene el espectro de una señal FM con mensaje sinusoidal, para dos casos: cuando $\beta = 1$ y cuando $\beta = 5$.

Figura 14: Relación entre las Funciones de Bessel y el espectro de una señal FM con mensaje sinusoidal



Aplicación usada: F_Bessel_Mag.py

- De la figura 14 puede deducirse que en cualquier caso, el Ancho de Banda de una señal FM es infinito, pero fuera de cierto ancho de banda el espectro cae casi a cero. De allí se deduce la Regla de Carson para calcular, de manera aproximada el Ancho de Banda que puede usarse en la práctica para la EC de una señal FM cuando el mensaje es sinusoidal:

$$BW = \Delta F + f_m = \Delta F(1 + \frac{1}{\beta}) \quad (3.29)$$

de manera que el Ancho de banda de la versión pasobandas es:

$$B = 2\Delta F + 2f_m = 2\Delta F(1 + \frac{1}{\beta}) \quad (3.30)$$

3.5.3.2. Caso en que el mensaje tiene forma arbitraria

Después de todo ese análisis, solo resta identificar la regla para calcular, al menos de manera aproximada, el Ancho de banda de una señal FM en un caso más cercano a la práctica, cuando en vez de un mensaje sinusoidal se de un mensaje $m(t)$. En este caso, es importante establecer que:

- En cualquier caso, el mensaje $m(t)$ tiene un Ancho de Banda BW_m .
- El mensaje $m(t)$ también tiene una amplitud pico A_p .
- Entonces, la Señal FM estará caracterizada por la Relación:

$$D = \frac{K_f A_p}{BW_m} \quad (3.31)$$

D es conocida como Relación de Desviación, pero es el equivalente a β que es el índice de modulación en el caso en que el mensaje sea una señal sinusoidal.

- También estará caracterizada por la Desviación de frecuencias ΔF :

$$\Delta F = K_f A_p \quad (3.32)$$

- De esta manera, para hallar el Ancho de banda de una señal FM con una señal mensaje arbitraria $m(t)$, es posible usar una forma modificada de la Regla de Carson, de manera que para la EC de la señal FM tenemos un Ancho de Banda igual a:

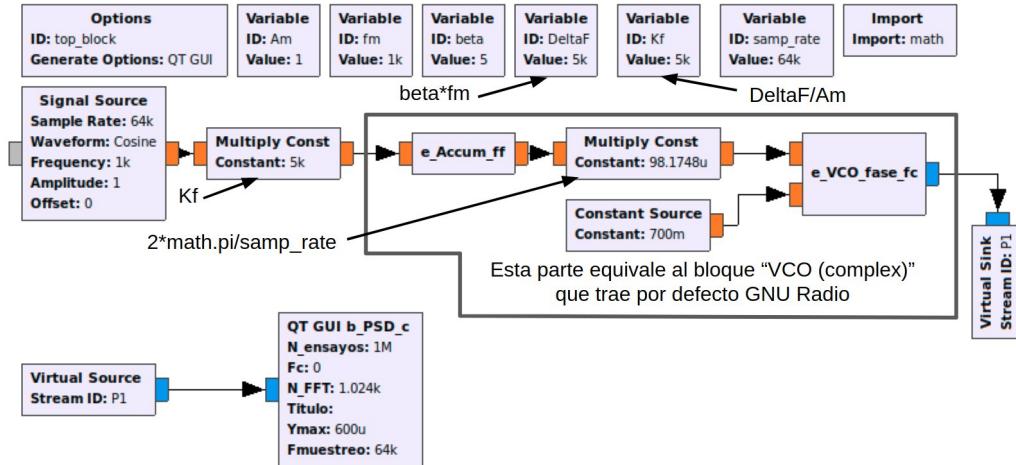
$$BW = \Delta F + BW_m = \Delta F \left(1 + \frac{1}{D}\right) \quad (3.33)$$

De manera que para una señal FM pasobandas, el Ancho de banda es aproximadamente:

$$B = 2\Delta F + 2BW_m = 2\Delta F \left(1 + \frac{1}{D}\right) \quad (3.34)$$

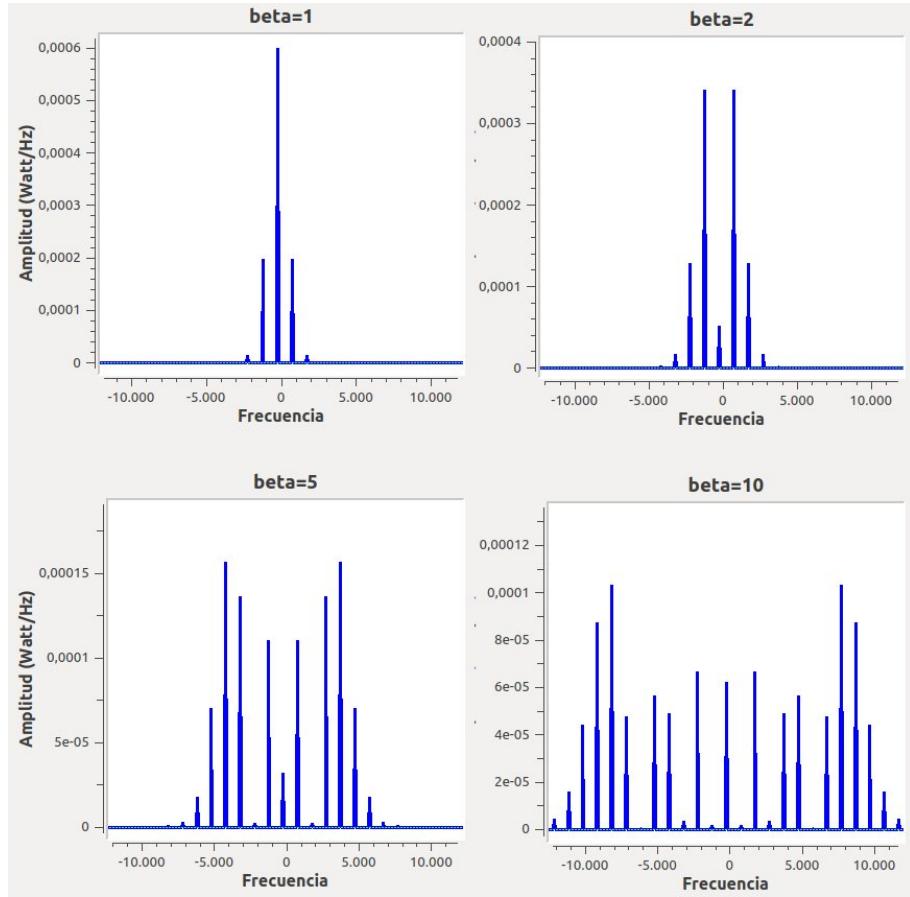
3.5.4. Implementación de un Modulador FM mediante GNU Radio

Figura 15: Flujograma para comprobar la Modulación FM



Flujograma usado: FM_banda_base_vco-fase.grc

Figura 16: PSD de una señal con Modulación FM banda base y con mensaje sinusoidal



Flujograma usado: FM_spectrum_analys.grc

La gran variedad de métodos reseñados para implementar un modulador FM puede resultar abrumador [3], por ejemplo existen métodos directos e indirectos que pasan por una implementación previa de un modulador FM de banda angosta. La existencia de esa gran variedad de métodos es debido a los grandes retos que plantean las implementaciones en hardware. En el caso de una implementación en software, como es el caso de SDR, ese problema es un tanto diferente, basta con usar un modelo matemático que funcione bien. Por ejemplo un VCO puede estar basado en la expresión matemática más genérica para la EC y puede ser usado como modulador FM, como en la implementación que se muestra en la figura 15 brindando resultados fantásticos para cualquier caso de aplicación práctica.

En la figura 16 se presentan los resultados de este modulador cuando se usa un mensaje sinusoidal. Como podemos ver, se obtiene la PSD correspondiente a la esperada del análisis teórico visto para diferentes valores de “ β ” .

3.5.5. El PLL como Demodulador FM

Revisaremos las tres técnicas más populares usadas en la demodulación de forma suficiente para implementarlas en GNU Radio:

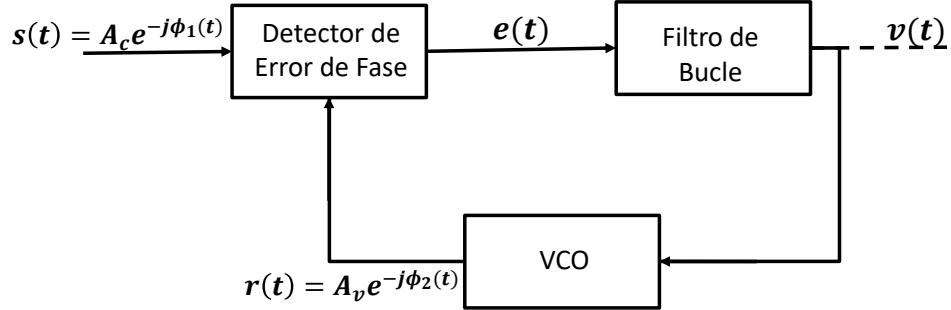
- Un PLL usado como detector de frecuencias.
- El Discriminador de frecuencias.
- El demodulador homodino.

Un PLL visto en tiempo continuo y en banda base:

El bucle de seguimiento es uno de los dispositivos más usados en los sistemas de control. Particularmente nos referiremos al bucle de Seguimiento de Fase (PLL del inglés Phase Locked Loop) como el dispositivo

que permite realizar un seguimiento a la fase y consecuentemente a la frecuencia de la onda que recibe como entrada. Existe mucha literatura sobre este tema, sin embargo poco se discute de la implementación en forma de Envolvente Compleja para una implementación útil en SDR. GNU Radio tiene por si misma facilidades para este tipo de implementación, sin embargo es importante contar con suficientes bases teóricas para poder configurar adecuadamente las herramientas de GNU Radio.

Figura 17: El Bucle de Seguimiento de Fase - PLL



En la figura 17 se tiene el diagrama de bloques de un PLL, donde la línea punteada indica lo que se agrega al PLL para adaptarlo como Detector de Frecuencias que no es otra cosa que el Demodulador FM que buscamos. Los siguientes son los aspectos a tener en cuenta:

- $s(t)$ es la señal FM recibida expresada como la EC.
- $v(t)$ es la señal mensaje que se logra demodular, de modo que es una versión cercana, pero escalada de $m(t)$ que es el mensaje o la señal modulante.
- Comparando lo que entra y lo que sale del VCO con lo mostrado en la figura 15 podemos deducir que la señal $r(t)$ es una señal con modulación FM.
- El reto del PLL consiste en lograr que la fase de la señal $r(t)$ sea lo más cercana posible a la fase de $s(t)$, excepto por el signo. Por eso se dice que está enganchado a la fase.
- A menudo El Detector de Error de Fase es un multiplicador, que realiza la siguiente operación:

$$y(t) = s(t)r(t) = e^{j(\phi_1 - \phi_2)}$$

Lo anterior hace que ese detector sea un sistema no lineal. Necesita entregar una señal proporcional a la fase, pero esta es un parámetro interno de otra función. Sin embargo, es posible establecer unas condiciones, muy cercanas a la realidad, para que el sistema pueda ser visto como un sistema lineal. La idea consiste en que, en la práctica, el error de fase es muy pequeño, décimas de π radianes, de modo que el error de fase se puede obtener a partir de la parte imaginaria de $y(t)$ ya que:

$$\text{Im}[e^{-j(\phi_1 - \phi_2)}] \approx \frac{1}{2}(\phi_1 - \phi_2)$$

De manera que no resulta difícil hacer que el Detector de Error de Fase, una vez encuentra $s(t)r(t)$, entrega en realidad una señal $e(t)$ que es proporcional al error de fase:

$$e(t) = K_p(\phi_1 - \phi_2)$$

Incluso, se supone que es posible hallar $e(t)$ como la arco tangente de $y(t)$, ya que por ser pequeño el ángulo no hay problema con una posible ambigüedad de fase.

Incluso cuando se cuenta con un bloque PLL listo para usar, es importante conocer muy bien algunos parámetros necesarios para la configuración, como:

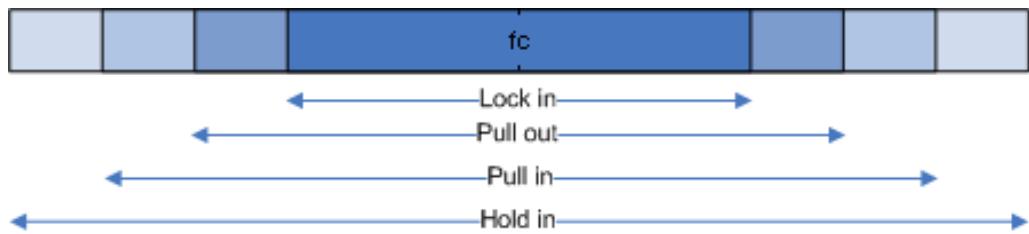
- **Ganancia de Bucle o “Loop Gain” (Hz):** Es usado más bien cuando el PLL se implementa en forma de hardware, de modo que no se tiene mucho control sobre el coeficiente de sensibilidad del VCO K_v y la amplificación K_m que pueda imprimir el multiplicador, de modo que el Loop Gain es $K_o = K_m K_v A_c A_v$
- **El Filtro de Bucle:** Es de alguna forma un Filtro pasobajas con un ancho de banda limitada y con posibles parámetros adicionales como: **la frecuencia natural, el factor de amortiguamiento.**

- **Loop Bandwidth:** Es una frecuencia de centésimas de radianes por segundo, pero influye en rango de frecuencias que el dispositivo puede identificar. En el ejemplo que se presenta en el capítulo de abajo se ofrece una forma de calcularlo para el caso de un PPL que es parte de GNU Radio.

- En la figura 18 se presentan varios términos relacionados con el ancho de banda a saber:

- **Hold in ó Rango de Bloqueo:** Siempre y cuando el PLL esté ya enganchado, con este margen de frecuencias es capaz de seguir variaciones muy lentas de frecuencias.
- **Pull out:** Es el margen de frecuencias para que, ante un salto brusco de la frecuencia de la señal entrante, el PLL se mantiene siempre enganchado.
- **Lock in:** Si el PLL está desenganchado, dentro de este margen de frecuencias es capaz de recuperar el enganche en menos de un periodo de la frecuencia de la señal de salida.
- **Pull in ó Rango de Captura:** Es el margen de frecuencias en el que el PLL es capaz de reengancharse fácilmente en caso de perder el enganche por alguna razón, aunque le puede tomar un tiempo mayor al periodo de la señal de salida.

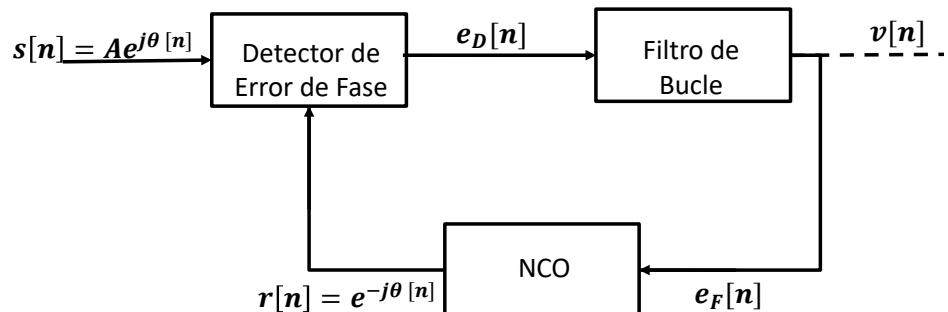
Figura 18: Términos relacionados con ancho de banda en los PLL



Fuente: https://es.wikipedia.org/wiki/Lazo_de_seguimiento_de_fase

Un PLL visto en tiempo discreto y en banda base: La programación que se puede implementar en el tiempo discreto brinda algunas facilidades adicionales. Para ver este caso, usaremos la figura 19.

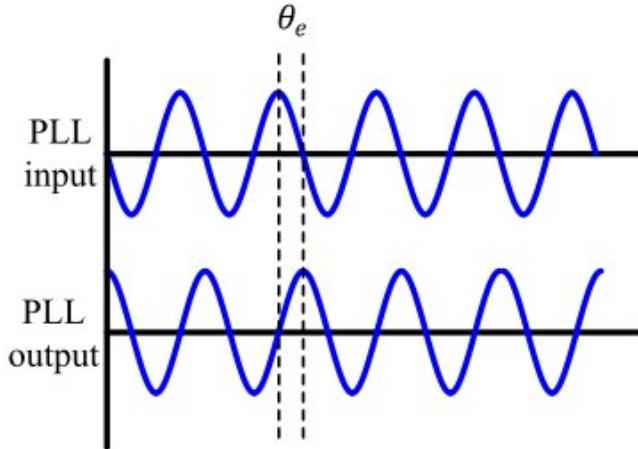
Figura 19: El Bucle de Seguimiento de Fase - PLL en tiempo discreto



El bloque Detector de Error de Fase Para este elemento aplican las siguientes observaciones.

- Es posible implementar un Bloque Detector del Error de Fase que calcule frecuentemente el error. $\theta_e[n] = \theta[n] - \hat{\theta}[n]$, como se muestra en la figura 20. Si esta diferencia se mantiene constante, las dos señales tienen la misma frecuencia y puede ser útil cuando por alguna razón la señal $s[n]$ llega afectada en frecuencia, por ejemplo, debido al Efecto Doppler, pero también en aplicaciones de FM. Se genera entonces la señal correspondiente $e_D[n]$.
- El bloque Detector de Error de Fase es el más versátil en un PLL y es el culpable de que existan muchas variantes de PLL.

Figura 20: El error de fase



- Como en el caso continuo, el Detector de Error de Fase no tiene acceso directo a la fase, por lo tanto se considera que el PLL en esta forma es un dispositivo no lineal, pero, en la mayoría de los casos puede ser aproximado a un sistema lineal debido a lo siguiente: cuando está en equilibrio, los ángulos $\theta[n]$ y $\hat{\theta}[n]$ son muy cercanos de manera que el error casi siempre tiende a cero y es posible identificar el ángulo de error mediante un proceso lineal, como se explicó para el caso continuo.

El bloque NCO Del inglés *Numerically Controlled Oscillator*. Es el equivalente al VCO, lo hemos llamado así, pues en términos discretos no se tienen niveles de voltaje sino números.

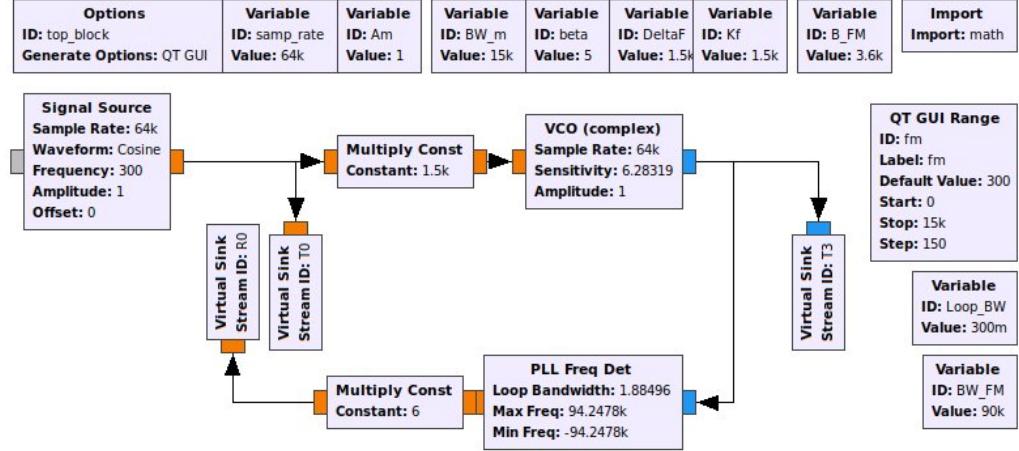
El Filtro de Bucle: El Filtro de Bucle es un filtro pasobajas, útil especialmente cuando la señal del PLL está implementado en versión pasobandas, ya que luego de una multiplicación entre senoidales es necesario filtrar las componentes de altas frecuencias. En el caso de una implementación, donde todo está en versión bandabase no resulta necesario el filtro para esa tarea, pero puede servir para otras, como por ejemplo para atenuar el ruido.

3.5.6. Implementación en GNU Radio de un Demodulador PLL

En el GRC de GNU Radio se dispone del bloque “PLL Freq Det” es básicamente un PLL adaptado como un medidor de frecuencias, que no es otra cosa que un Demodulador FM. En la figura 21 se ha implementado la capa FM de la siguiente manera:

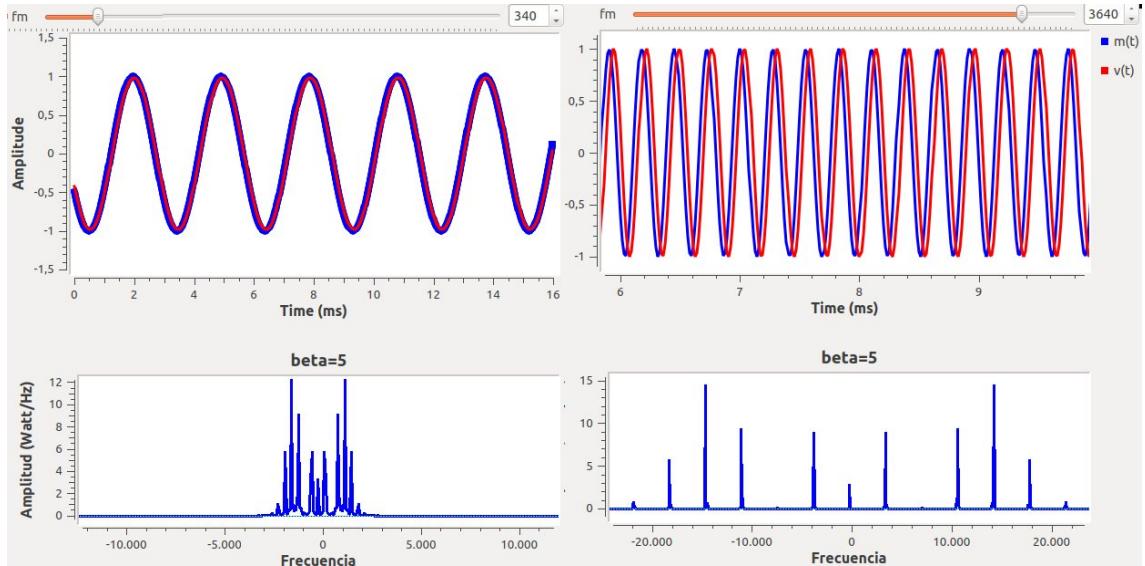
- En calidad de mensaje se tiene una fuente sinusoidal representada en el bloque “Signal Source”.
- Seguidamente encontramos el Modulador FM implementados mediante dos bloques: “Multiply Const” y “VCO (complex)”. El primero es para definir el Coeficiente de Sensibilidad de Modulación, el cual está asociado a su vez un Índice de Modulación β , representado en el parámetro *beta* y una desviación de frecuencias ΔF , representada en el parámetro ΔF .
- En calidad de DeModulador tenemos los bloques “PLL Freq Det” y “Multiply” el primero para detectar la frecuencia y el segundo para escalar la señal resultante. Este bloque tiene los siguientes parámetros de configuración:

Figura 21: Flujograma que muestra la capa de Modulación FM con un PLL como Demodulador



Flujograma usado: De_FM_PLL.grc

Figura 22: Resultados de la Demodulación FM con PLL



- Todos los parámetros relacionados con frecuencias deben ser configurados en términos de la frecuencia angular, es decir en radianes/segundo. Si se tiene el valor en Hz, simplemente se introduce ese valor multiplicado por 2π .
- **Loop Bandwidth:** Es una frecuencia que es cientos de veces menor a un Hertz, o a 2π radianes por segundo. Este parámetro debe ser seleccionado cuidadosamente para que el sistema funcione dentro de un cierto ancho de banda. En el caso del Flujograma propuesto, hemos creado la variable *Loop_BW* y la hemos calculado a partir del ancho de banda *NW_m* que puede llegar a tener el mensaje:

$$Loop_BW = (2BW_m/1000.)/100.$$

- *Max_Freq* y *Min_Freq*. Se supone que equivale al ancho de banda, sin embargo en las pruebas realizadas esto no queda tan claro ya que hemos experimentado con diversos valores y todos fueron aceptados sin afectar el funcionamiento del bloque para las condiciones dadas, excepto frecuencias por encima de 10^6 [Hz].

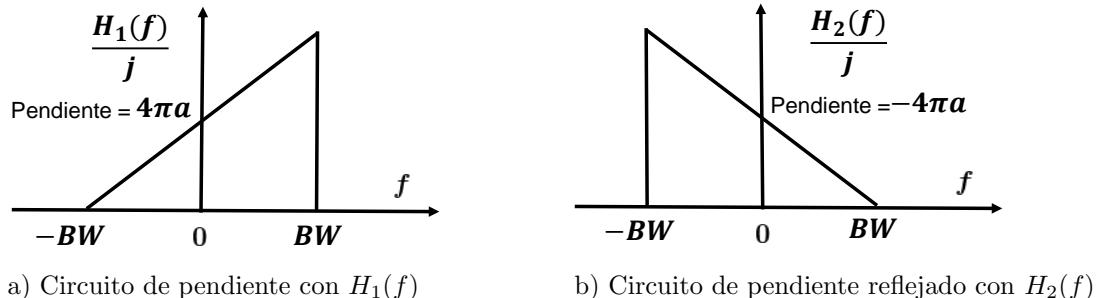
En la figura 22 se tienen los resultados de la simulación para dos casos: cuando la frecuencia del mensaje $m(t)$ es 340 Hz y cuando es 3640 [Hz], lo cual barre el rango necesario para el caso de la telefonía. Está claro

que $v(t)$ es la señal demodulada. En una implementación real de FM, el mensaje puede alzar los 15[[kHz]]. La reconfiguración del fluajograma en este caso parece consistir solo en cambiar el parámetro BW_m para que valga 15 [[kHz]]. En realidad, en cualquier implementación que se realice en términos de software requiere revisar cuidadosamente la frecuencia de muestreo con cualquier cambio que se realice al ancho de banda de las señales. En este caso particular, es necesario tener en cuenta que el ancho de banda de la señal FM se ensanchará cada vez más con el ancho de banda de la señal mensaje, como lo expresa la Regla de Carson (3.29), (3.33). Por esta razón, el fluajograma propuesto en la figura 21 incluye el parámetro BW_FM para calcular el ancho de banda que va a tener la señal FM. Usando la ecuación 3.29, reemplazando f_m por BW_m y expresándola en términos de β en lugar de ΔF tenemos que en el bloque “Variable” se configura la variable BW como $\beta * BW_m + BW_m$. Como resultado se obtiene que si $BW_m= 15$ [kHz], se obtiene que $BW_F=90$ [[kHz]], luego la frecuencia de muestreo samp_rate debe ser por lo menos 180 [[kHz]].

3.5.7. Discriminador de frecuencias con circuitos de pendiente

La idea parte de la teoría de señales y sistemas. Por un lado sabemos que una señal FM en banda base es una exponencial compleja expresada en la ecuación (3.8), de modo que si a esa señal le aplicamos la derivada, se obtiene la misma exponencial compleja pero el mensaje va a aparecer como parte de la amplitud, de donde lo podemos tomar de manera más sencilla. Por otro lado, sabemos que un derivador puede ser visto como un sistema LIT con una respuesta en frecuencia en forma de pendiente. Finalmente, debemos considerar que la señal FM banda base en el dominio de las frecuencias ocupa un ancho de banda entre $-BW$ y BW , que es lo mismo que decir entre $-B/2$ y $B/2$, si se tiene en cuenta que BW es el ancho de banda de la versión banda base y B es el de la versión pasobandas. Vamos a usar dos circuitos de pendiente. El primero lo representaremos con la respuesta en frecuencia $H_1(f)$, el segundo como $H_2(f)$. De momento nos concentraremos en el primero y su respuesta en frecuencia se presenta en la figura 23.

Figura 23: Respuesta en Frecuencia de un Circuito de pendiente



La expresión para la respuesta al impulso del primer circuito de pendiente es:

$$H_1(f) = \begin{cases} j4\pi a(f + BW), & -BW \leq f < BW \\ 0, & \text{others} \end{cases} \quad (3.35)$$

Si representamos la salida del discriminador como $s_1(t)$ tenemos que:

$$S_1(f) = \frac{1}{2} H_1(f) S(f) \quad (3.36)$$

$$\begin{cases} j2\pi a(f + BW)S(f), & -BW \leq f < BW \\ 0, & \text{others} \end{cases}$$

De la teoría de Tratamiento de Señales y Sistemas, sabemos que si en el dominio de las frecuencias la señal se multiplica por f , en el dominio del tiempo equivale a una derivación, de modo que:

$$s_1(t) = a \left[\frac{ds(t)}{dt} + j2\pi BW s(t) \right] \quad (3.37)$$

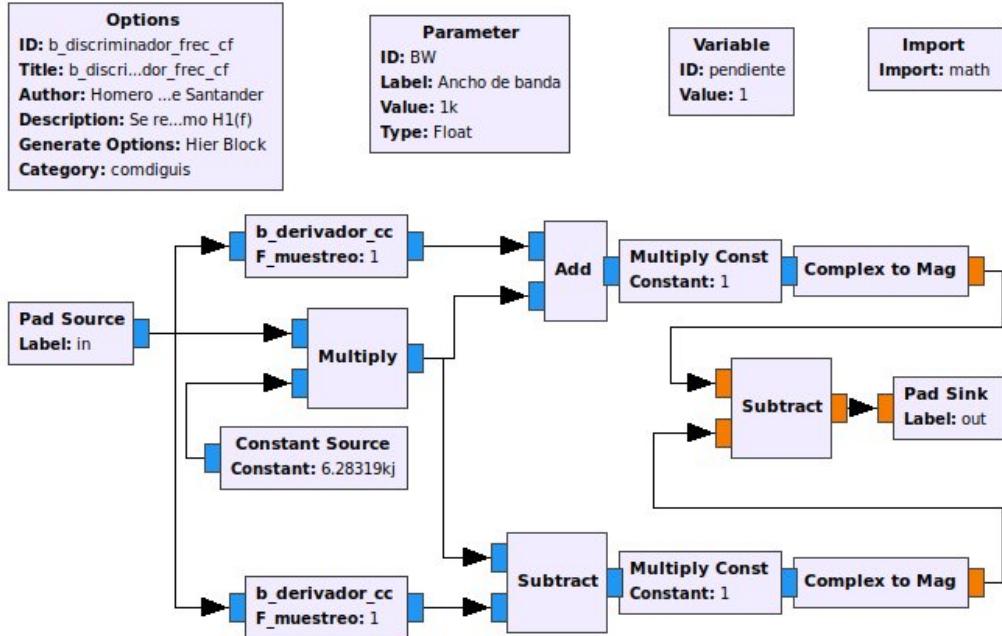
Teniendo en cuenta que nuestra señal de interés es una señal con modulación FM en banda base:

$$s_1(t) = A_c e^{j2\pi K_f \int_0^t m(\tau) d\tau} \quad (3.38)$$

sustituimos en (3.37) y obtenemos:

$$s_1(t) = j2\pi BW a A_c \left[1 + \frac{K_f}{BW} m(t) \right] e^{j2\pi K_f \int_0^t m(\tau) d\tau} \quad (3.39)$$

Figura 24: Flujograma del bloque jerárquico “b_discriminador_frec_cf”



Flujograma usado: Bloque “b_discriminador_frec_cf” de la librería “comdiguis”

De manera que si logramos obtener $s_1(t)$ es posible usar un sencillo algoritmo para obtener algo más cercano a nuestro mensaje $m(t)$, por ejemplo, dividiendo en j y aplicando una operación que entregue la amplitud de una señal compleja para obtener:

$$|s_1(t)| = 2\pi BW a A_c + 2\pi a A_c K_f m(t) \quad (3.40)$$

El problema que ahora surge es el término $2\pi BW a A_c$ que aparece a la izquierda de la ecuación anterior. Con la intención de eliminarlo es que, aparece la necesidad del segundo circuito de pendiente, representado en la respuesta en frecuencia $H_2(f)$:

$$H_2(f) = H_1(-f) \quad (3.41)$$

En la figura 24 se muestra la implementación en forma de un flujograma de un bloque jerárquico en GRC. Allí el circuito de pendiente se ha implementado mediante un bloque “b_derivador” que es otro bloque jerárquico mencionado ya en el capítulo ??, este bloque juega el papel de un derivador discreto. El discriminador propuesto está hecho para obtener la señal:

$$v(t) = \|s_1(t)\| - \|s_2(t)\| = 4\pi a A_c K_f m(t) \quad (3.42)$$

3.5.8. Discriminador de frecuencias con derivadores

Aunque el método anterior también se basa en derivadores, su diseño parte de los circuitos de pendiente, mientras que esta versión está más relacionada con una búsqueda directa del ángulo en la Envoltura Compleja para luego derivarlo y obtener la frecuencia a partir de la derivada del ángulo. En estudio en [4] con aplicación en Fibra Óptica y se describe de manera resumida en [5], comparándolo con otros métodos y señalando las ventajas que tiene en la práctica, aunque en esa fuente la aplicación es en óptica:

- Ofrece la mejor resolución y sensibilidad comparado con otros métodos. Se menciona una sensibilidad de 10^{-6} radianes.
- Alto rango dinámico.

y hemos decidido mencionarlo aquí por su sencillez y efectividad.
La idea consiste en operar con la componente en fase y en cuadratura de

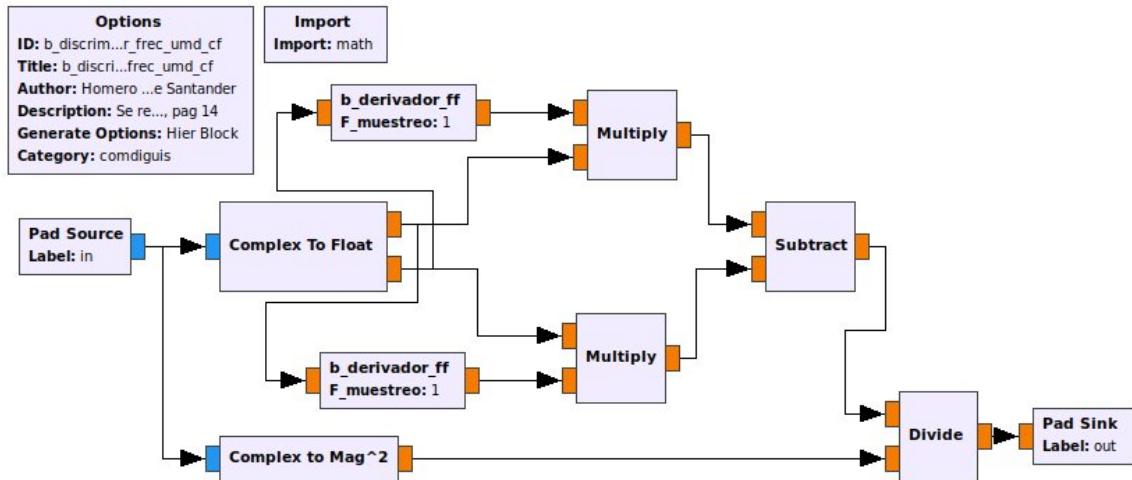
$$s(t) = s_I(t) + js_Q(t)$$

para obtener el ángulo a partir de la operación de arco tangente, para luego derivarla y obtener la frecuencia como:

$$v(t) = \frac{d\theta(t)}{dt} = \frac{s_I(t)\frac{ds_Q(t)}{dt} - s_Q(t)\frac{ds_I(t)}{dt}}{s_I^2(t) + s_Q^2(t)} = K_f m(t) \quad (3.43)$$

la implementación en GNU Radio se realizó en forma del bloque jerárquico “b_discriminador_frec_umd_cf” que es parte de la librería “comdiguis” disponible en la página web del libro y se representa en la figura 25.

Figura 25: Discriminador de frecuencias con derivadores



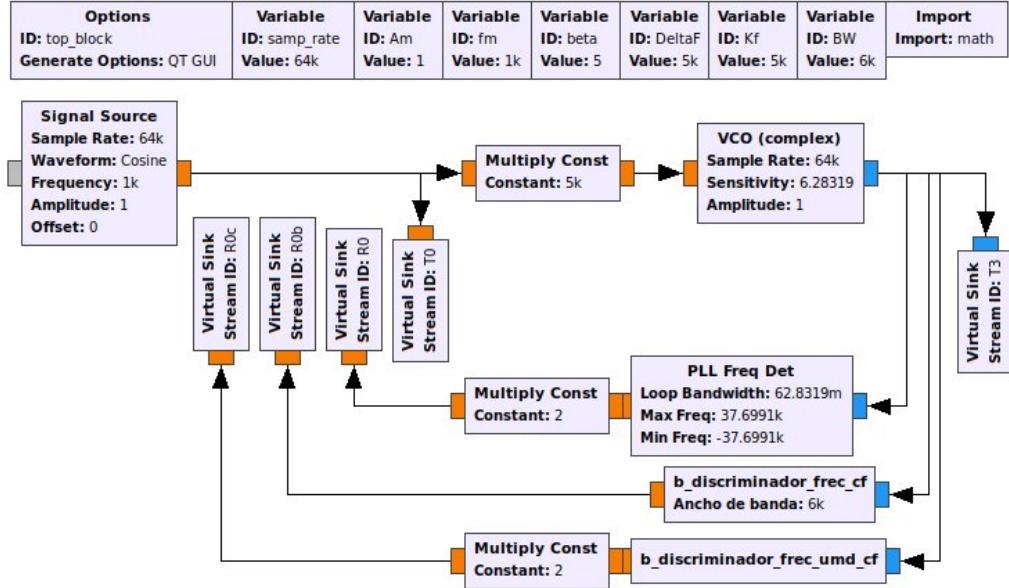
Flujograma usado: bloque “b_discriminador_frec_umd_cf” de la librería “comdiguis”

Es curioso que esta solución no pide ningún parámetro de configuración.

3.5.9. Pruebas en GNU Radio del protocolo de modulación

La idea es comprobar qué tanto se entiende el modulador FM con el demodulador FM. Es decir, si efectivamente lo que recibe el modulador FM es lo que puede entregar el demodulador.

Figura 26: Flujo de datos para comprobar la capa de modulación con diferentes métodos de demodulación

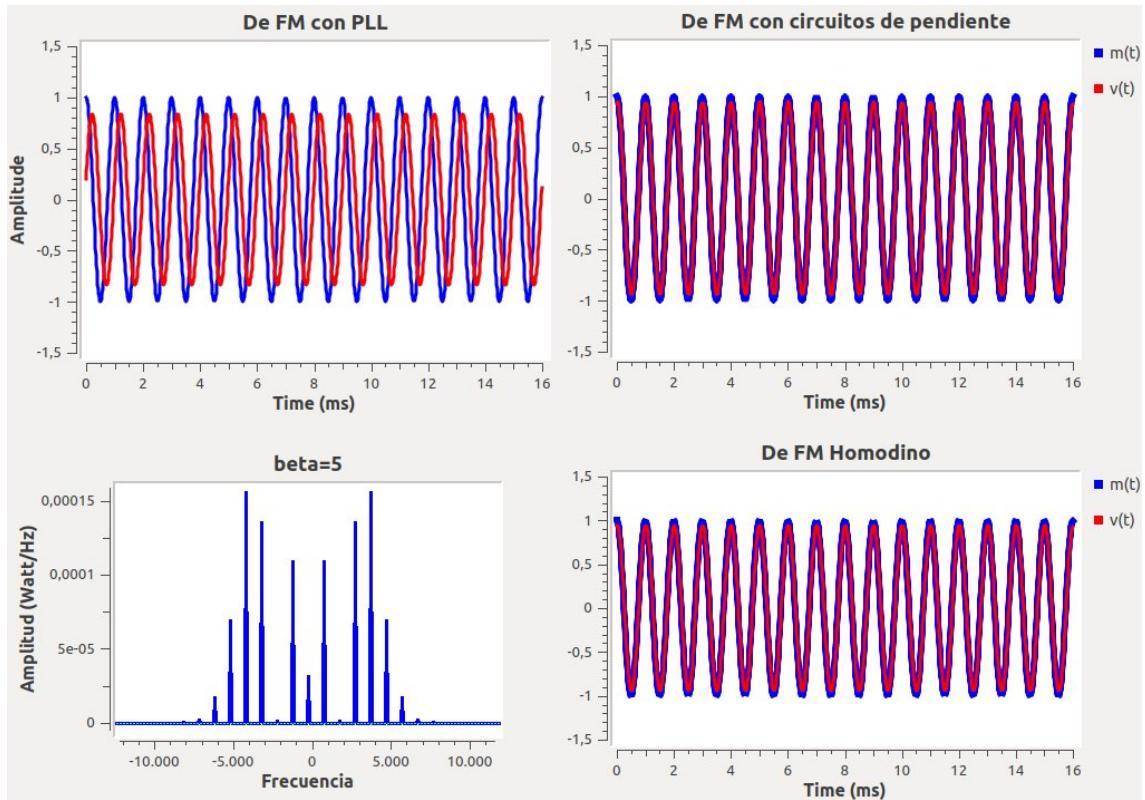


Flujo de datos: fm-mod-demod-comparacion.grc

En la figura 26 estamos presentando una implementación que combina la Modulación y la Demodulación, esta última con los tres métodos vistos.

En la figura 27 se presentan los resultados comparativos. Vemos que los métodos presentan buenos resultados, sin embargo, aún falta analizar el efecto del paso de la señal modulada por un canal para poder identificar mayores diferencias entre los métodos de demodulación.

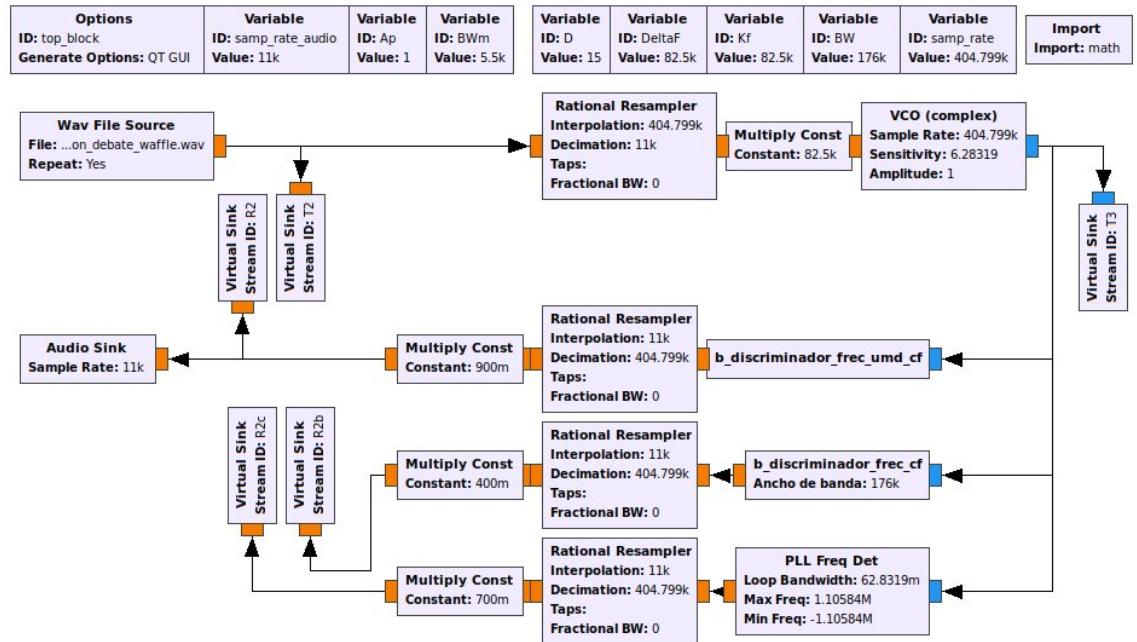
Figura 27: Resultados comparativos de la demodulación con tres métodos



3.5.10. Pruebas en GNU Radio de un Sistema FM con señales de audio

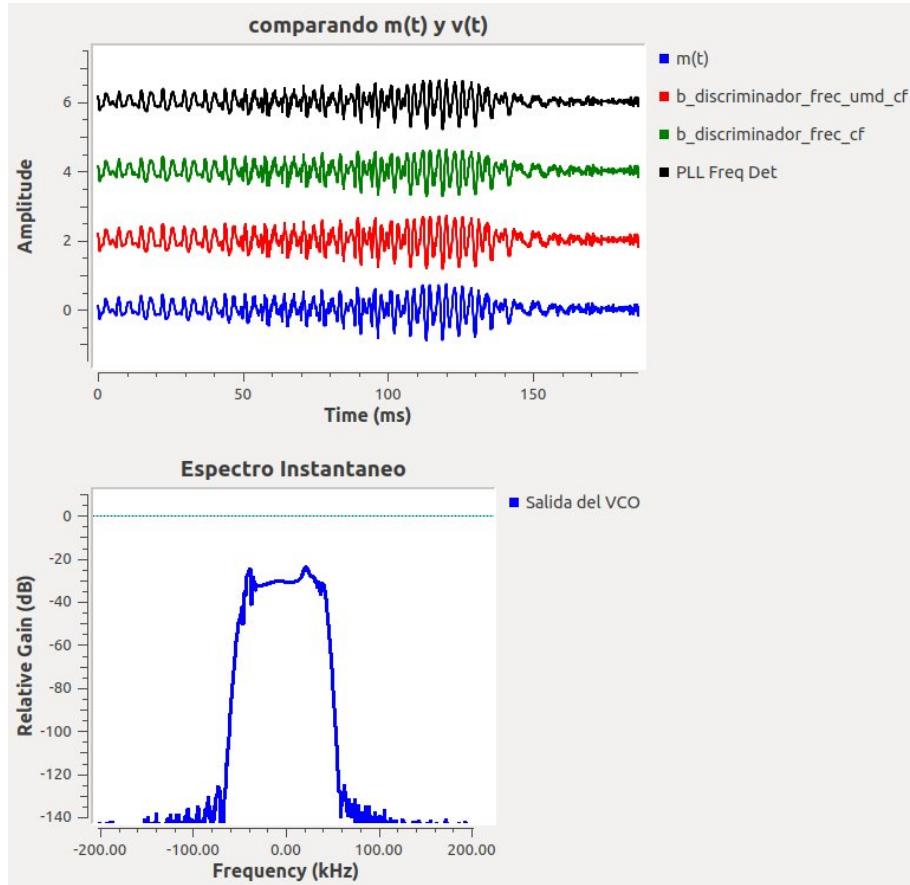
En la figura 28 se ha implementado el fluograma para pruebas en tiempo real de un sistema de comunicaciones FM con señales de audio. Se trata de una implementación real en el sentido de que el mensaje es captado del mundo real, aunque sea leído de un archivo, ya que tiene unos parámetros muy reales de tiempo y frecuencia, además porque a medida que la señal se modula y demodula, deberemos poderla escuchar en la bocina del computador en tiempo real, es decir, sin distorsiones de tiempo que hagan perder sentido a la comunicación.

Figura 28: Flujograma para un sistema de comunicaciones FM con señales de audio



Flujograma usado: sistema_comunicacion_FM_simulacion.grc

Figura 29: Resultados de un sistema de comunicaciones sonora FM



El flujograma tiene las siguientes características relevantes:

- El mensaje tiene su propia frecuencia de muestreo, que en este caso corresponde a

$$samp_rate_audio = 11[kHz].$$

Es un parámetro tomado del archivo de audio, que es muy propio de la manera en que la señal fue grabada y guardada en el archivo.

- Con la frecuencia de muestreo de la señal de audio viene asociado su ancho de banda que de acuerdo al Teorema de Nyquist es:

$$BW_m \leq samp_rate_audio/2$$

- Las mediciones muestran que la amplitud pico del mensaje es:

$$A_p = 1$$

Podría estar dada en voltios si queremos conservar una relación con el mundo físico, pero puede ser más sencillo comenzar a ver esa señal desde el punto de vista de la teoría de señales discretas, donde eso es simplemente un número que cuantifica una amplitud.

- Por otro lado, tenemos que la Modulación FM está configurada de la siguiente manera:

- En ese flujograma se ha decidido que la Relación de Desviación sea $D = 15$.
- De lo anterior se derivan los demás parámetros como $\Delta F = DBW_m = 82,5$ [kHz].
- También se deduce la sensibilidad de frecuencia: $K_f = \Delta F/A_p = 82,5$ [[kHz]].
- Para conocer finalmente una aproximación del ancho de banda de la señal modulada mediante la Regla de Carson: $BW = \Delta F + BW_m = 176$ [kHz].
- Finalmente seleccionamos una frecuencia de muestreo que sea mayor a dos veces el ancho de banda de la señal FM, en este caso hemos tomado $samp_rate = 2,3BW = 404,799$ [kHz].

- El reto que aparece ahora es el siguiente: La señal mensaje tiene una frecuencia de muestreo muy diferente a la frecuencia de muestreo de la señal FM: La primera vale $samp_rate_{audio} = 11$ [kHz], la segunda $samp_rate = 404,799$. Esto es un problema, ya que los bloques propuestos para la modulación y demodulación consideran que a la entrada tienen la misma frecuencia de muestreo que la de salida. La solución se encuentra en la inclusión del bloque “Rational Resampler”, hecho precisamente para poder realizar esa traducción entre frecuencias de muestreo, pero siempre respetando el Teorema de Nyquist, esto significa que:

- Siempre es posible pasar de una frecuencia de muestreo más baja a una más alta mediante una operación de interpolación. Solo hay que tener en cuenta que con esto el ancho de banda de la señal no se va a ampliar, lo único que se va a lograr es contar con una señal sobre muestreada, es decir, una señal con una frecuencia de muestreo mayor al mínimo que establece el Teorema de Nyquist.
- Siempre es posible pasar de una frecuencia de muestreo más alta a una más baja mediante una operación de decimación. Solo hay que tener en cuenta que la calidad de la señal no desmejora solo si su ancho de banda es al menos dos veces menor a la nueva frecuencia de muestreo. De no ser así, la operación de decimación va a actuar como un filtro pasobajas que comienza por limitar el ancho de banda de la señal para que sea igual a dos veces la nueva frecuencia de muestreo.
- La manera en que se recomienda configurar el bloque “Rational Resampler” es así: Al parámetro “Decimation” otorguele el valor de la frecuencia de muestreo de la señal entrante, al parámetro “Interpolation” otorguele el valor de la señal saliente.
- Se ha incluido un analizador de espectros que muestra el espectro instantáneo. Observamos que el espectro cambia todo el tiempo tanto su forma como el ancho de banda que está ocupando, siempre dentro de los límites previstos.

En la figura 29 se presentan parte de los resultados obtenidos al correr el flujograma que esta disponible en la página web del libro, que se menciona en la portada. De la experimentación con este flujograma podemos resaltar lo siguiente:

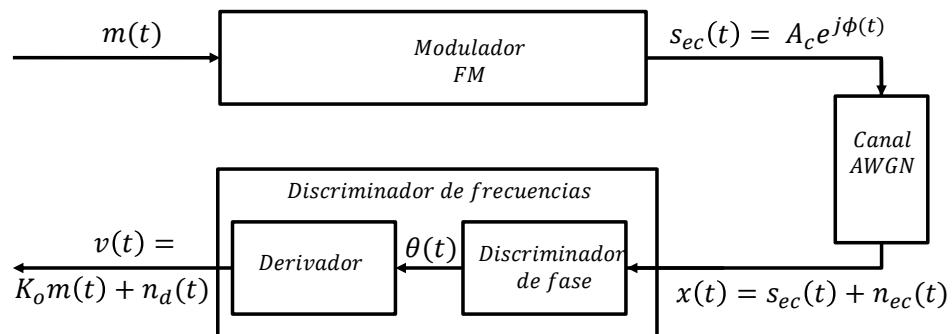
- Todos los métodos de demodulación FM funcionan muy bien para la señal con los parámetros normales de la radiodifusión sonora, al menos para la región de las américa.
- No se realizaron pruebas de desempeño, de modo que no podemos recomendar uno u otro método de demodulación. En todo caso, desde el punto de vista de configuración no hay nada mejor que el bloque “b_discriminador_fre_umd_cf” ya que no requiere ningún parámetro de configuración y en la literatura mencionada aparece bien recomendado por su sensibilidad y rango dinámico.

3.5.11. Influencia del ruido en la Modulación FM

Para este análisis nos basaremos en el esquema de la figura 30 donde podemos ver que el canal adiciona ruido blanco de banda angosta a la señal modulada $s_{ec}(t)$ para producir la señal $x(t)$. Hemos incluido un discriminador de fase teórico para contar con la fase $\theta(t)$ a partir de la cual, mediante un derivador, se traduce a frecuencia, lo cual no es otra cosa que una versión de la señal mensaje el cual trae agregado una componente del ruido $n_d(t)$ que produce el discriminador de frecuencias.

Entonces, tenemos los siguientes aspectos a considerar:

Figura 30: Esquema para el análisis del efecto del ruido en un sistema de comunicaciones FM



- Al usar técnicas SDR, todo el análisis se realiza en términos de la EC, de modo que toda señal y sistema que mencionemos aquí está dado en términos de EC. Además usamos letras en mayúsculas o subindicadas con ec cuando se trata de representar señales en versión de EC en el dominio del tiempo.
- La modulación FM en términos de RC está dada por la fórmula 3.8 que reproducimos así:

$$s_{ec}(t) = A_c e^{j2\pi K_f \int_0^t m(\tau) d\tau} = A_c e^{j\phi(t)}$$

- En realidad nos interesa el efecto del ruido blanco de banda angosta, que representamos como $n(t)$ ya que una señal modulada, en términos prácticos, está siempre limitada en un ancho de banda determinado.
- El ruido blanco en términos de su EC $n_{ec}(t)$ está dado por una componente en fase $n_I(t)$ y una en cuadratura $n_Q(t)$, por lo tanto, se puede representar como:

$$n_{ec}(t) = n_I(t) + jn_Q(t) \quad (3.44)$$

Entonces, podemos hablar de la magnitud del ruido blanco

$$r(t) = [n_I^2(t) + n_Q^2(t)]^{1/2} \quad (3.45)$$

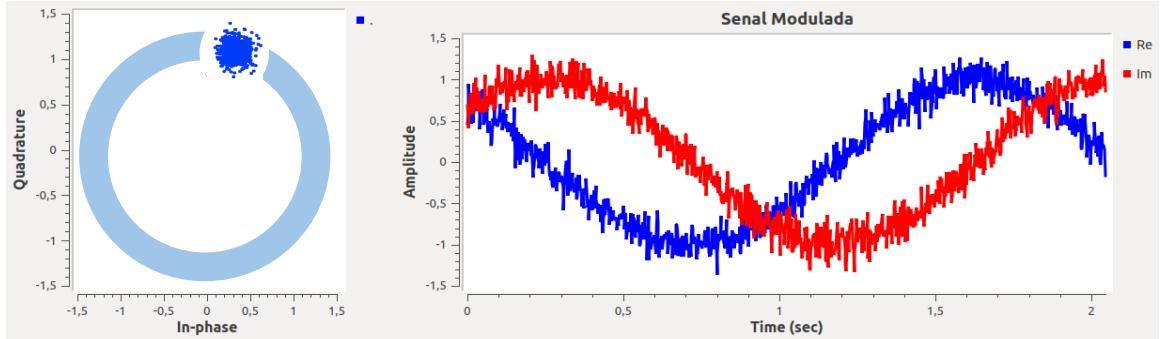
y de la fase del ruido blanco

$$\psi(t) = \tan^{-1} \left[\frac{n_Q(t)}{n_I(t)} \right] \quad (3.46)$$

- De manera que la señal afectada por el ruido, que representaremos como $x_{ec}(t)$ está dada por

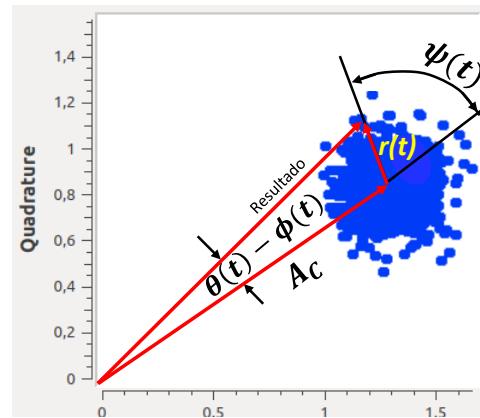
$$x_{ec}(t) = s_{ec}(t) + n_{ec}(t) = A_c e^{\phi(t)} + r(t) e^{j\psi(t)} \quad (3.47)$$

Figura 31: Una señal FM afectada por ruido blanco



Flujograma usado: `vector_rotante_full.grc`

Figura 32: Análisis en el plano polar sobre el efecto del ruido en FM



En la figura 31 presentamos los resultados de una simulación realizada con GNU Radio para obtener una señal FM afectada por ruido blanco aditivo gaussiano. Como podemos ver en la parte izquierda, el vector rotante que representa a la señal FM en términos de su EC, arrastra consigo una nube que representa a $n_{ec}(t)$. A la derecha se observa la componente en fase y en cuadratura de la $s_{ec}(t)$. En la figura 32 se analiza con más detenimiento el vector que representa a la señal $s_{ec}(t)$ en el plano complejo en un instante de tiempo.

- Se deduce entonces, que el ángulo de la señal resultante $x(t)$ cambia de la siguiente manera:

$$\theta(t) = \phi(t) + \tan^{-1} \left[\frac{r(t)\sin[\psi(t) - \phi(t)]}{A_c + r(t)\cos(\psi(t) - \phi(t))} \right] \quad (3.48)$$

Las variaciones que puedan ocurrir a la señal $r(t)$ pueden ser fácilmente eliminadas mediante un Limitador que es un dispositivo capaz de mantener estable la magnitud de la señal. Para simplificar la expresión anterior, vamos a suponer que la relación señal a ruido en $x(t)$ es grande comparada con la unidad y que podemos implementar un dispositivo ideal para medir el ángulo instantáneo, de modo que $r(t)$ en promedio es pequeña comparada con A_c y es entonces posible simplificar la expresión de θ y el ángulo instantáneo que se lograría medir es:

$$\theta(t) \approx \phi(t) + \frac{r(t)}{A_c} \sin[\psi(t) - \phi(t)] \quad (3.49)$$

donde:

$$\phi(t) = 2\pi K_f \int_0^t m(\tau) d\tau \quad (3.50)$$

- Como la expresión anterior corresponde al ángulo, la señal que sale de un discriminador es en realidad la derivada ya que este se ocupa de entregar las variaciones de las frecuencias:

$$\begin{aligned} v(t) &= \frac{1}{2\pi} \frac{d\theta(t)}{dt} \\ &= K_f m(t) + n_d(t) \end{aligned}$$

donde $n_d(t)$ corresponde al paso de la señal de ruido $n(t)$ a través del discriminador.

$$n_d(t) = \frac{1}{2\pi A_c} \frac{d}{dt} [r(t)\sin[\psi(t) - \phi(t)]] \quad (3.51)$$

A partir de todo ese análisis podemos concluir que el ruido blanco, al pasar junto con la señal modulada por el discriminador de frecuencias resulta siendo algo diferente a ruido blanco e incluso es algo que aparece no solo en función del ruido blanco sino también del mensaje a través de $\phi(t)$. En [3] capítulo 2.13 se describen algunas aproximaciones adicionales para llegar a expresar el ruido a la salida del discriminador como:

$$n_d(t) = \frac{1}{2\pi A_c} \frac{dn_Q(t)}{dt} \quad (3.52)$$

O sea que $n_d(t)$ se puede determinar de manera efectiva a partir de la componente en cuadratura $n_Q(t)$ el ruido blanco de banda angosta.

El objetivo ahora es conocer lo que ese nuevo ruido representa en el dominio de las frecuencias, de modo que pensando en términos del dominio de las frecuencias tenemos que:

- $n_d(t)$ puede ser visto como la señal $n_Q(t)$ luego de pasar por un filtro con respuesta al impulso

$$H(f) = j2\pi f \frac{1}{2\pi A_c} = \frac{jf}{A_c}$$

- por lo tanto, en términos de Densidad Espectral de Potencia tenemos que:

$$S_{Nd}(f) = \frac{f^2}{A_c^2} S_{NQ}(f)$$

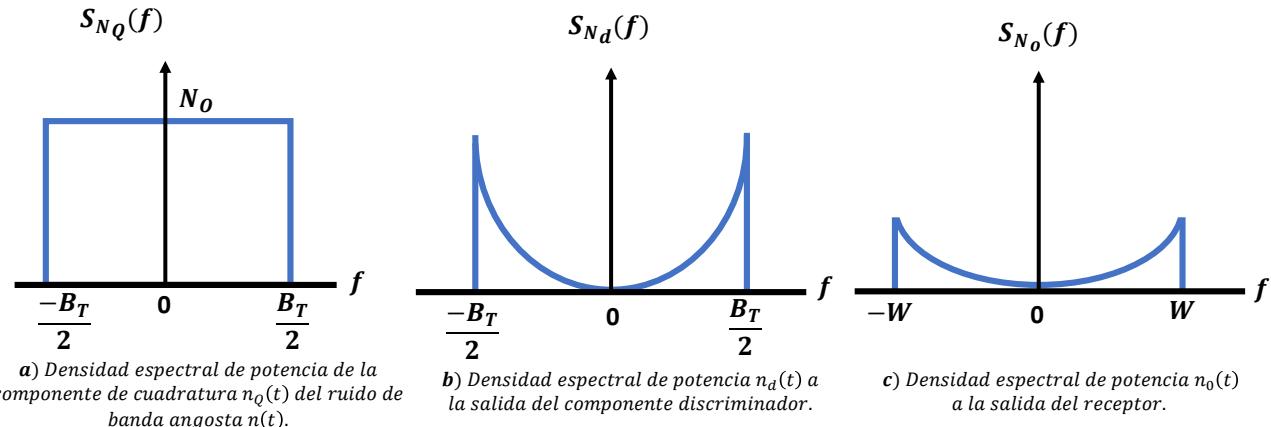
- de allí que

$$S_{Nd}(f) = \begin{cases} \frac{N_0 f^2}{A_c^2}, & |f| \leq BW \\ 0, & \text{Otros} \end{cases}$$

Una vez la señal pasa por filtro que se limita al ancho de banda del mensaje BW_m , tendremos que la afectación que el ruido causa en el dominio de la frecuencia es la adición de:

$$S_{Nd}(f) = \begin{cases} \frac{N_0 f^2}{A_c^2}, & |f| \leq BW \\ 0, & \text{Otros} \end{cases}$$

Figura 33: Análisis de ruido del receptor de FM



En la figura 33 podemos observar en forma gráfica lo que este análisis significa y enunciar las siguientes conclusiones de utilidad práctica:

- El mensaje estará más afectado por el ruido en las frecuencias más altas que en las bajas.
- Resulta fácil encontrar la potencia promedio del ruido que afecta a una señal FM ya es el área bajo la curva de la PSD, de modo que:

$$P_{Nd} = \frac{N_0}{A_c^2} \int_{-BW}^{BW} f^2 df = \frac{2N_0 W^3}{3A_c^2}$$

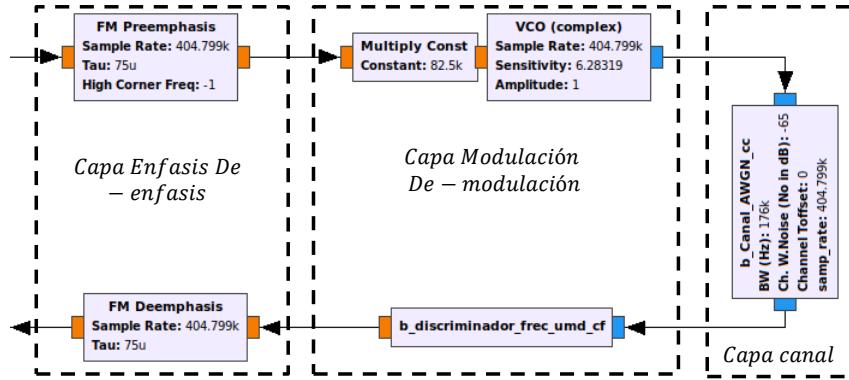
Gracias a esto podemos pensar en un análisis adicional para encontrar aspectos relacionados con la relación señal a ruido en sistemas de comunicación FM, pero esto está bien documentado en otras fuentes como [3] en el capítulo 2.13.

3.5.12. Implementación final

3.5.12.1. Énfasis y De-énfasis en FM

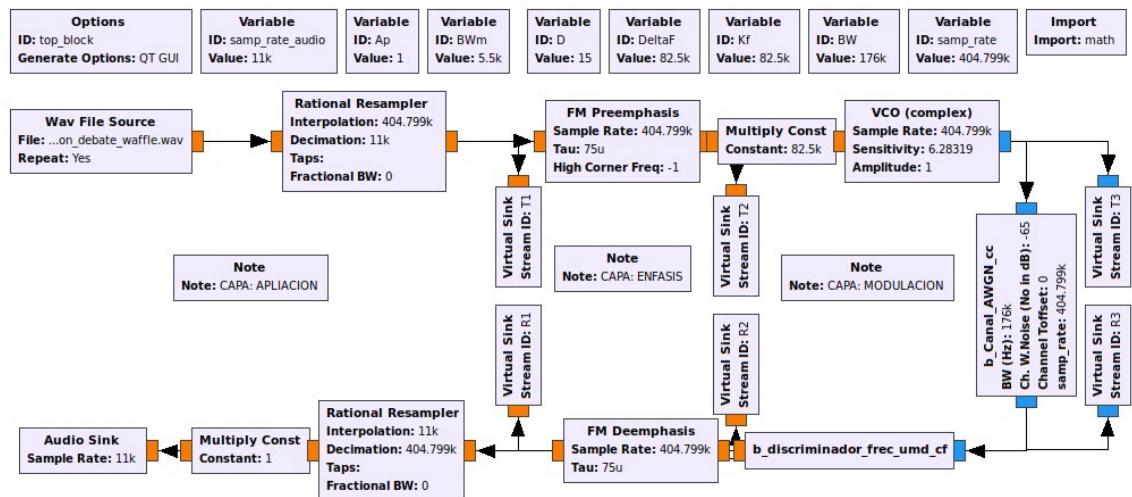
La operación de énfasis y deénfasis es muy común en los sistemas de comunicaciones con modulación FM. Desde el punto de vista de una implementación en GNU Radio es una capa que se le adiciona a un sistema de comunicaciones FM, como se muestra en la figura 34.

Figura 34: Modelo de capas para ubicar la operación de énfasis de-énfasis



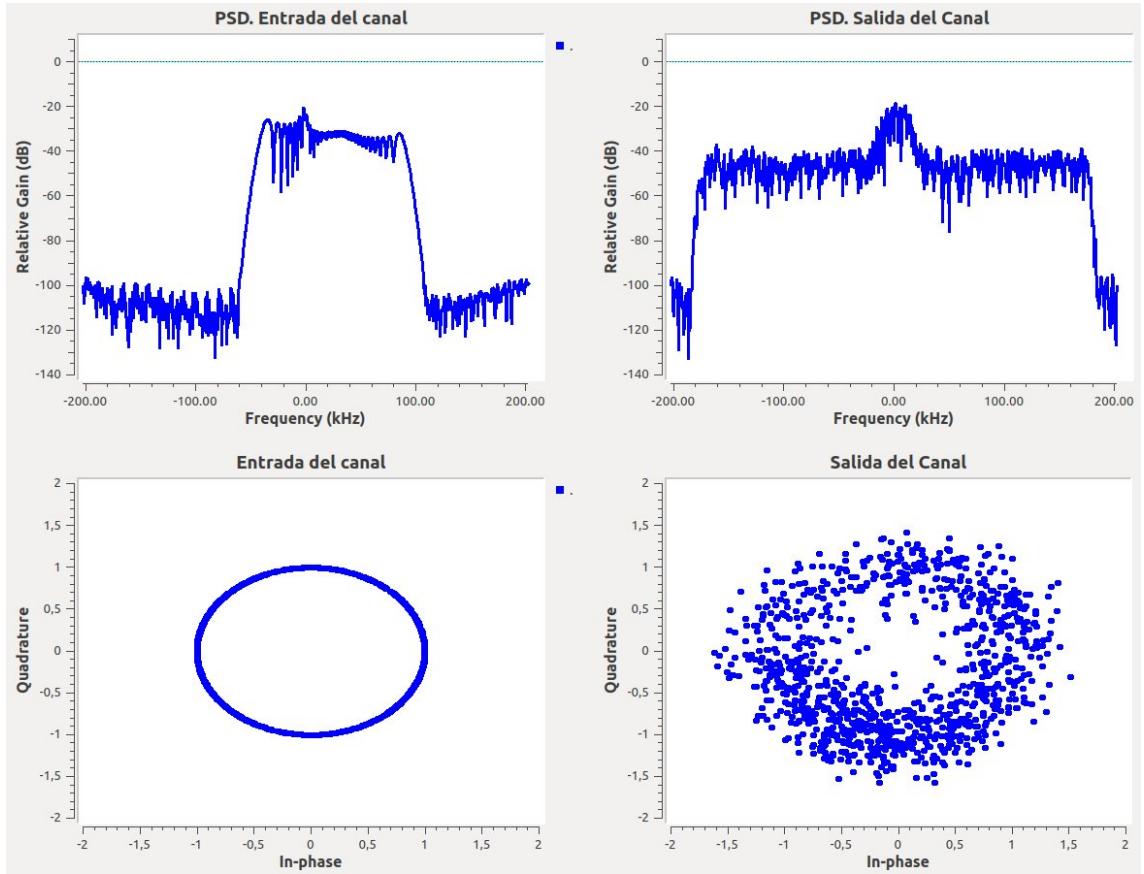
Flujograma usado: sistema_comunicacion_FM_enfasis_deenfasis.grc

Figura 35: Flujograma de un sistema completo de comunicaciones FM



Flujograma usado: sistema_comunicacion_FM_enfasis_deenfasis.grc

Figura 36: Resultados del sistema de comunicación FM



El problema que se busca resolver es el siguiente: De acuerdo a lo visto en el capítulo anterior, la señal que se obtiene luego de la de modulación FM resulta afectado por el ruido del canal de una manera más fuerte en las frecuencias altas que en las bajas. Como solución, se busca aplicar un procesamiento a la señal mensaje, antes de la modulación, para que, en el dominio de las frecuencias, la señal resulte más amplificada en las frecuencias altas que en las bajas.

Este proceso es sencillo y está ampliamente documentado en la literatura [3]. De manera que vamos a ocuparnos de estudiar los bloques que trae GNU radio para esta operación, en contraste con los métodos de demodulación propuestos.

GNU Radio viene acompañado con los bloques “FM Preemphasis” y “FM Deemphasis”. Estos bloques funcionan perfectamente con los parámetros que traen por defecto, siendo necesario ajustar solamente la frecuencia de muestreo de la señal a procesar. Sin embargo, en una aplicación práctica es necesario tener en cuenta que hay estándares que establecen los parámetros para esos bloques y dependen de las regiones definidas por la UIT. Esta información puede ser consultada en Internet ¹.

El sistema completo ha quedado implementado, ya que basta con unir el flujo de la figura 34 y el de la figura 28.

El cual se presenta en la figura 35. No aparece la capa de “Técnicas banda base pre/post canal” debido a que en los sistemas de comunicación FM las técnicas usadas para hacer frente a los retos que plantea un canal están antes de la capa de modulación, como es el caso de lo que podríamos llamar capa de énfasis/de-énfasis.

Una inquietud que el lector puede tener es la siguiente: debería haber un filtro después del demodulador o del bloque de de-énfasis con el fin de ajustar el ancho de banda justo a lo que ocupa el mensaje, teniendo en cuenta que el ancho de banda de la señal modulada es mucho más amplio en las técnicas de Modulación FM. Incluso, si observamos la figura 36 podemos comparar la PSD a la entrada del canal con la PSD a la salida del mismo y notamos que podemos estar admitiendo mayores niveles de ruido de lo estrictamente

¹ver más detalles en el enlace https://www.radiomuseum.org/forum/fm_pre_emphasis_and_de_emphasis.html

necesario ya que esta entrando ruido más allá del ancho de banda requerido por la señal. La respuesta a esa inquietud la resumimos en los siguientes puntos:

- No hay que olvidar que la modulación FM es un método no lineal, lo cual en el dominio de las frecuencias se traduce en un ancho de banda mucho mayor al que tiene el mensaje.
- El espectro que se observa en la figura, aunque se le llame PSD; es más bien un periodograma ya que el promedio no se realiza en el tiempo infinito. En otras palabras esa imagen espectral va a estar cambiando de forma que en cada instante ocupará mayor o menor ancho de banda. El verdadero ancho de banda se puede calcular de manera aproximada mediante la regla de Carson vista anteriormente.
- La señal que entrega el modulador está limitada adicionalmente por la frecuencia de muestreo de esa señal. Es necesario que la frecuencia de muestreo de la señal, representada en nuestro caso por la variable *samp_rate* sea mayor a 2 veces el ancho de banda de la señal modulada, representado en la variable *BW*. En el caso de la figura 35 se tienen una configuración de datos exactamente igual a la explicada en el capítulo 3.5.10.

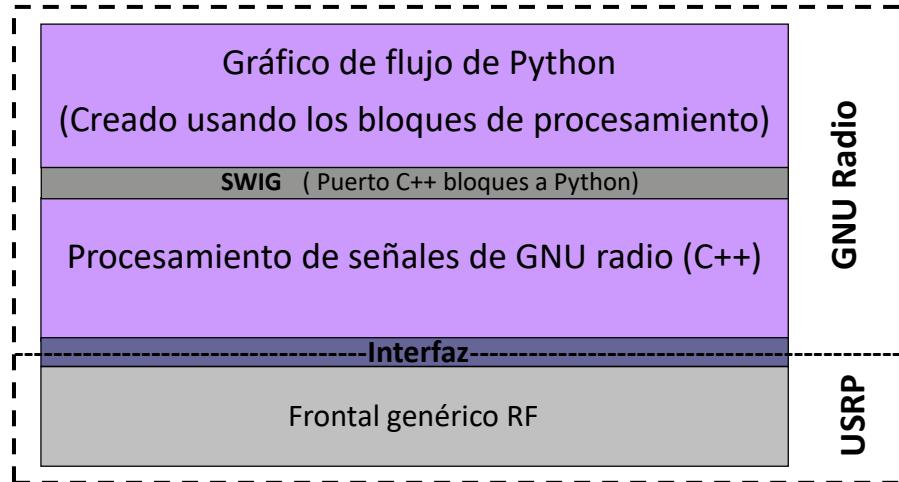
Capítulo 4

La Radio Definida por Software. Aplicación en sistemas de radiodifusión

Este capítulo tiene el fin de estudiar lo que significa Software Defined Radio (SDR) y aplicar estos conocimientos a un tipo específico de hardware que tendremos como referencia. Para ello se ha seleccionado el USRP-2920 de la empresa National Instruments. Otro objetivo es el de generalizar el conocimiento sobre el hardware usando modelos de capas, con lo cual abre un panorama más amplio al lector para comprender cualquier tipo de hardware SDR existente. El mayor énfasis de este capítulo es lograr que el lector conozca lo que el hardware representa en la tecnología SDR.

4.1. Integración de GNU Radio y SDR

Figura 1: Modelo de GNU Radio integrado con SDR



Fuente: El gráfico ha sido adaptado de Marwanto [6]

La idea detrás de GNU Radio quedó explicada en el capítulo anterior, donde quedó claro que se trata de una librería de recursos que pueden ser usados bien sea mediante programación en el lenguaje C++, en Python o inclusive mediante un lenguaje gráfico, basado en fluogramas. Para explicar la relación que se teje entre GNU Radio y SDR usaremos el modelo de la figura 1, del cual se puede deducir lo siguiente:

- Una solución SDR que combina GNU Radio y un USRP puede ser vista como un modelo de dos capas: la capa GNU Radio y la capa USRP.
- La capa USRP permite que la componente de software que se desarrolla, se conecte directamente con el hardware. Aunque en términos más genéricos, puede llamarse *Front End* Genérico de RF,

pues el término USRP es más bien un término de la compañía National Instruments para referirse al tipo de equipos SDR que ellos comercializan.

- En la práctica se requiere algo más: el hardware está conectado a un computador, el cual puede ser tan grande o tan pequeño como sea posible. Incluso el computador puede ser parte del hardware. Este hecho hace que en la práctica se tenga en realidad dos elementos por conectar: el computador y el hardware que para nuestro ejemplo es un USRP. Por esa razón, se requiere introducir un medio de comunicación para estos dos elementos. Ese medio usualmente está representado en los puertos de comunicación que estos dos elementos tengan como: USB, Ethernet u otros. En la actualidad, los equipos USRP optan por la opción de puerto Gigabit Ethernet, pero otros tipos de hardware como Realtek RTL2832U, optan por puerto USB. En todo caso, cuando hay un medio de comunicación de por medio, surge la necesidad de desarrollar una capa sobre este medio que se encarga de adaptar la información a este medio. En otras palabras, se requiere un software que llamaremos driver, el cual toma la información que entrega el software de la solución SDR, y la traduce para que pueda viajar por el protocolo de comunicación. Del lado del *Front End* se requiere también un driver similar para que la comunicación se dé. Para el caso de los USRP ese driver es conocido como UHD (*USRP Hardware Driver*)
- GNU Radio es realmente una librería de radiocomunicaciones para lenguaje C++, con una versión en el lenguaje Python. El GRC es un ambiente de programación gráfica que aprovecha esa librería y produce código en Python o en C++.
- Es de uso libre.
- Las aplicaciones hechas con GNU Radio son muy livianas y pueden llevarse a producción sin que dependan de una plataforma determinada.
- Es ampliamente usada por científicos y académicos.

4.2. Radio Definida por Software en Modo de Red

El término Modo de Red o “modo Network” es muy usado en los productos de la empresa Ettus, la cual a su vez pertenece a National Instruments, pero este modo corresponde la configuración más común en los laboratorios experimentales por ser la solución más económica y a la vez más potente. Consiste en lograr un complemento perfecto entre un hardware SDR y un computador con el software apropiado. El término network es usado debido a que el hardware se conecta al computador usualmente por un puerto Giga Ethernet de manera directa. En principio, no se descarta que esa conexión sea por Wi-Fi o que los componentes estén ubicados en lugares remotos distantes entre sí, unidos por Internet, pero esto no es usual pues entre el hardware y el computador se requiere enviar información muy pesada en tiempo real.

Figura 2: Un sistema de comunicaciones punto a punto Full Duplex.

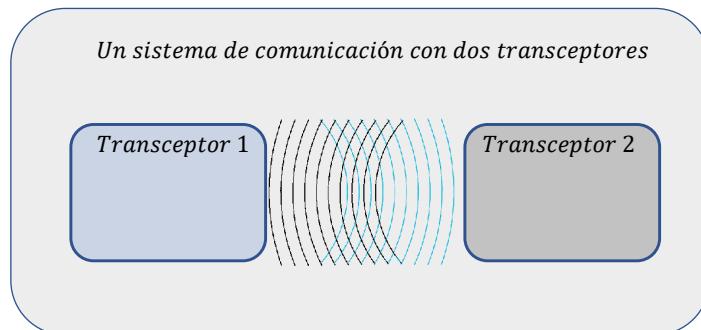
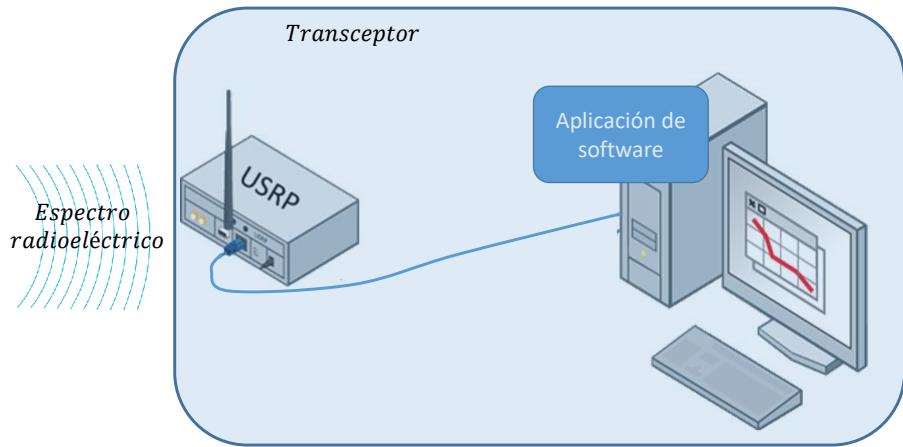


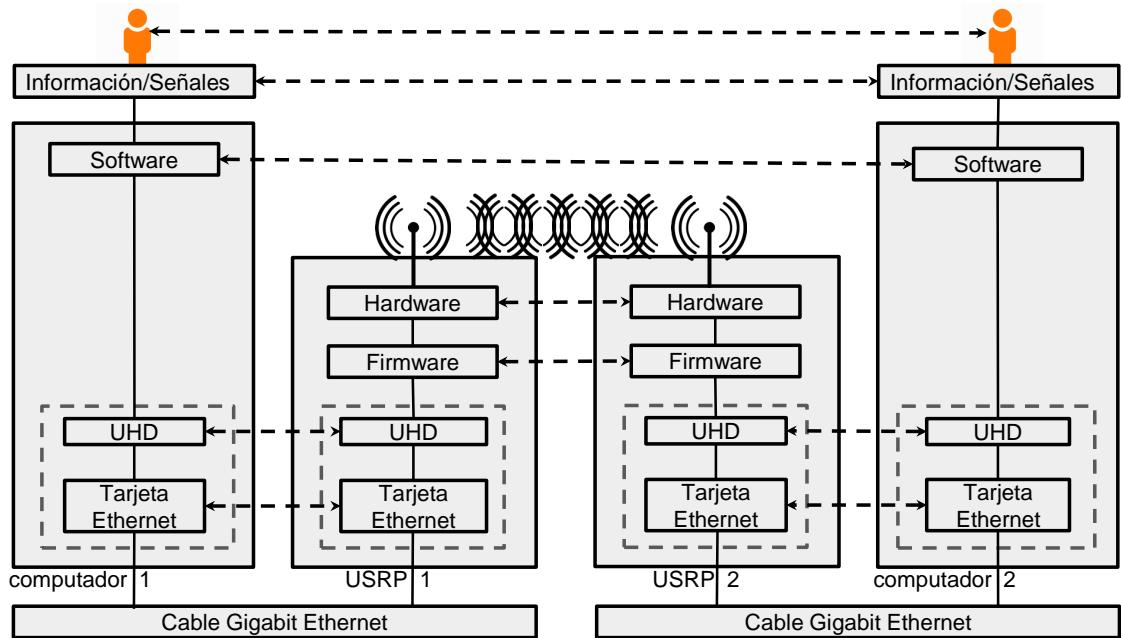
Figura 3: Un transceptor (en inglés Transceiver).



USRP (*Universal Software Radio Peripheral*) es el término usado por la empresa Ettus para denominar el hardware de una solución SDR. Como se muestra en la figura anterior, en el USRP se concentra la componente hardware del sistema de comunicación y en el computador la componente del software. Lo que viaja por el cable que une estos componentes es precisamente la Envoltura Compleja, no sin antes ser adaptada al protocolo Ethernet.

Para los USRP que solo funcionan en modo *Network*, se usa una nomenclatura especial, se considera que pertenece a la serie Nx, donde la N significa *Network* y la x unos números seriales. Algunos ejemplos son: USRP N200, USRP N210. Sin embargo, como la empresa Ettus fue adquirida por National Instruments (NI), esta empresa está usando otra nomenclatura como: NI USRP 2920, NI USRP 2921, NI USRP 2922, NI USRP 2930, NI USRP 2932. Ese cambio de nomenclatura es aplicado por NI para indicar que estos equipos funcionan con LabView que es la plataforma de NI que ahora incursiona en SDR. Una comunicación completa en Modo Network se presenta en la figura 4.

Figura 4: Modelo de capas de un sistema de comunicación con USRP en Modo Network.



4.3. Un equipo SDR para Modo de Red. El NI USRP-2920

El NI USRP-2920 es un ejemplo típico de un hardware pensado en una solución de comunicación basada en SDR en modo *network*. Vale la pena usarlo como ejemplo para conocer lo que este tipo de equipos usualmente contiene.

Figura 5: Foto del NI USRP-2920.

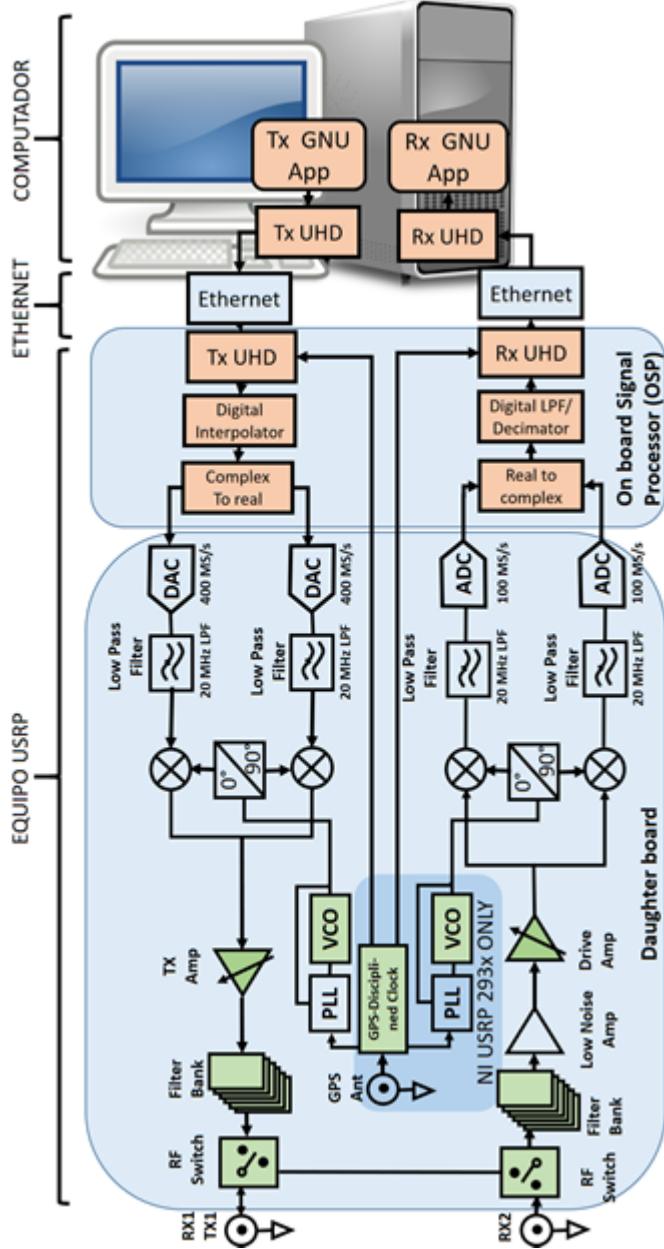


Es importante, tener claro desde un comienzo que estos USRP son construidos por la empresa [Ettus Research](#). Pero esta empresa fue adquirida en el 2015 por la empresa National Instruments (NI). De esta manera se entiende por qué un mismo hardware tiene dos nombres diferentes: uno que le da Ettus y otro que le da NI. **Es así como el NI USRP 2920 de National Instruments corresponde al USRP N210 de Ettus**, donde la N indica que se trata de un equipo que funciona en modo Network. El equipo viene con un software de NI que permite configurar el equipo con el driver que le permite actuar como NI USRP 2920 o bien con el driver que le permite actuar como USRP N210. El primer caso se usa cuando el equipo se usa con LabView sobre Windows, mientras el segundo, con GNU Radio sobre Ubuntu.

El manual de este equipo¹, es suficiente para comprender lo que el equipo representa externamente, así como sus especificaciones técnicas y cuidados de uso. Nos detendremos más bien en el esquema interno.

¹<http://www.ni.com/pdf/manuals/375839a.pdf>

Figura 6: Modelo de capas del NI USRP 29xx visto por capas. En color naranja: bloques de software embebido. En color verde: Hardware con parámetros controlables por computador. Blanco: Hardware no controlable.



En los siguientes capítulos, el término “Usuario P” se refiere a la persona que está sentada en el computador, desarrollando una aplicación en GNU Radio o usando una para contactar el USRP.

4.3.1. El receptor del NI USRP-2920

Como puede apreciarse, la parte receptora consiste en una o dos antenas que son conectadas por el RF switch. En muchos USRP la señal pasa inmediatamente por un banco de filtros que resultan sintonizados mediante las órdenes que da el Usuario P desde la aplicación ubicada en el computador. La señal es luego amplificada en el *Low Noise Amp*. La señal sigue a un amplificador que puede ser configurado por el usuario P desde la aplicación. El esquema que sigue es lo que en el primer capítulo hemos llamado *Down Converter*, en el cual la frecuencia del Oscilador controlado por voltaje (VCO) puede ser configurada desde la aplicación, pero no los parámetros del *Low Pass Filter*, que tiene un ancho de banda fijo, inmodificable,

de 20 MHz.

El Usuario P tampoco puede modificar los parámetros del Conversor Análogo Digital (ADC), que muestrea la señal a su entrada a una tasa fija, invariable de 100 MS/s y además la cuantiza a 14 bits/muestra. El Usuario P debe jugar con la ganancia o la atenuación para poder aprovechar todos los 14 bits por muestra, pues la señal entrante es muy débil y no alcanzaría a levantarse de manera suficiente. Pero también es necesario tener en cuenta que una sobre amplificación puede crear también un problema, el de saturación, como se muestra en las siguientes dos figuras.

Figura 7: Señal a la entrada del cuantizador.

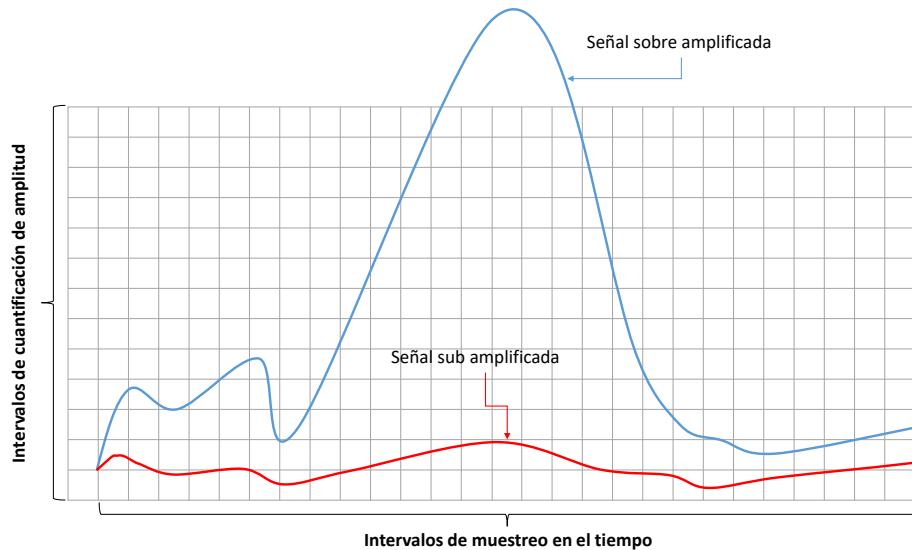
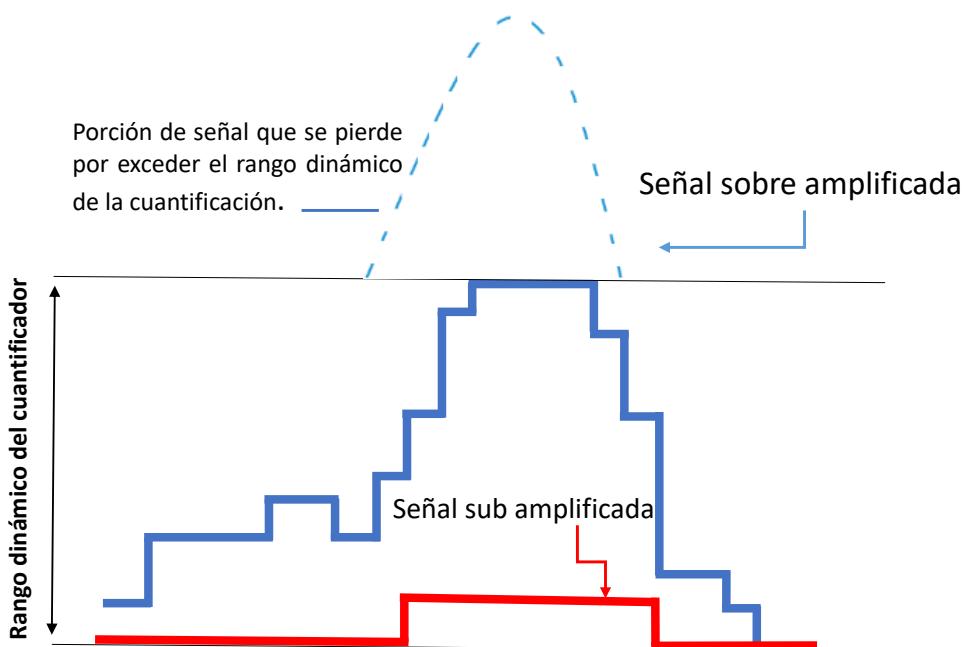


Figura 8: Señal a la salida del cuantizador.



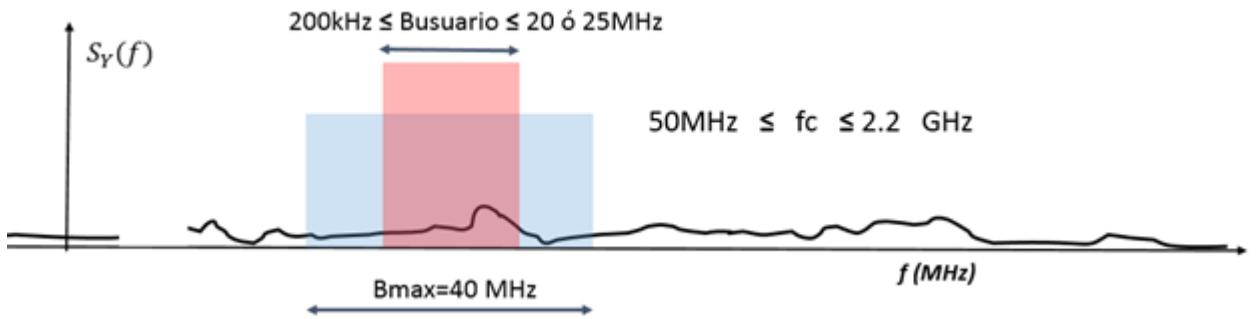
Por esta razón, es ideal que la señal resultante después de la amplificación quede con magnitud 1 o

incluso 0,8, ya que el rango dinámico del ADC es 1 Vp-p, para el caso del NI USRP 2920. Así, las amplitudes de la componente I y la Q se puede deducir de la magnitud $\sqrt{I_{max}^2 + Q_{max}^2} = 1$.

4.3.1.1. Cálculo de la frecuencia de muestreo para un NI USRP.

La señal pasa luego a la FPGA (la tarjeta madre), donde está el OSP (*Onboard signal processor*) que tiene algunos elementos de procesamiento digital, filtra la frecuencia central, de modo que no aparezca allí un delta que corresponde a la portadora. Pero también en el OSP se le aplica a la señal un diezmado, combinado con un filtrado digital, para bajar la tasa de muestreo de la señal *I/Q* para que cumpla con los deseos del programador que especificará un ancho de banda deseado que, en términos de frecuencia de muestreo, resultará en un valor que para el NI USRP 292x está entre 200 kS/s y 25 MS/s. Debido a las limitaciones del computador y la calidad del cable usado para unir el USRP con el computador, el Usuario P puede considerar que el ancho de banda pasobandas está entre 200 [kHz] y 20 ó 25 MHz, como se muestra en la siguiente figura.

Figura 9: Ancho de banda real que puede alcanzar el NI USRP 2920 con respecto a las capacidades de la tarjeta hija



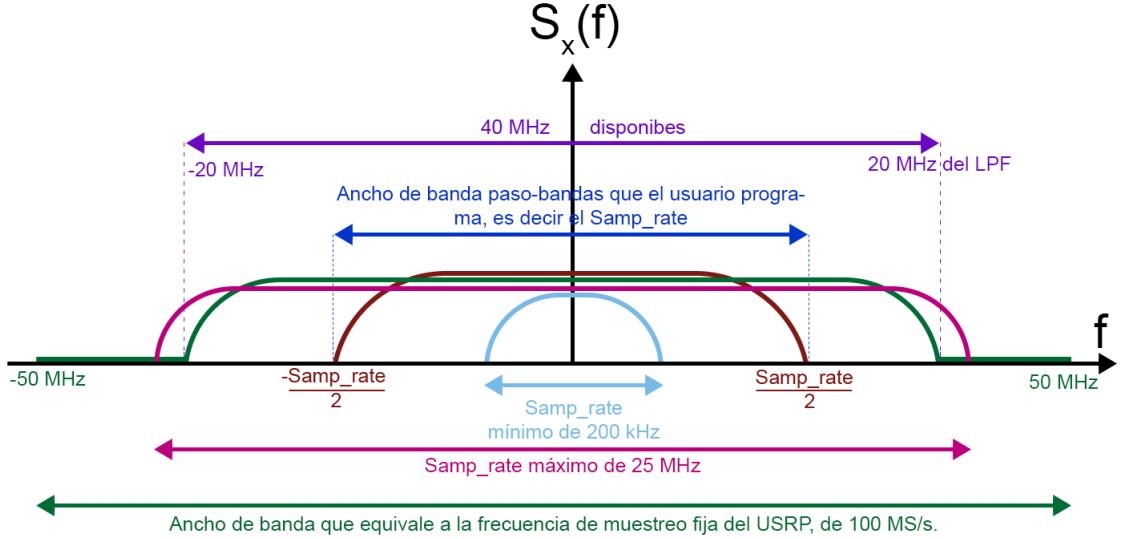
Ese ancho de banda se puede configurar en los bloques digitales alojados en el OSP. Esto ocurre de manera indirecta, cuando el Usuario P selecciona el Coeficiente de diezmado K_d para el bloque Digital LP-F/decimador con el fin de bajar la frecuencia de muestreo. Es importante tener en cuenta que, usualmente ese coeficiente es un número entero y a la vez potencia de 2, de modo que:

$$K_d = 2^m, \text{ donde } m \text{ es entero positivo.} \quad (4.1)$$

Los experimentos desarrollados por el autor, con el NI USRP 2920, muestran que, aunque el sistema acepta cualquier valor entero que sea par y positivo para ese coeficiente, si no se escoge como se señaló, se deforma el espectro de la señal que sale del decimador respecto a la que recibe el decimador. **El autor también demostró que con el NI USRP 2920, hay un tope máximo para ese coeficiente y es 512.** Eso significa que así como el equipo tiene un tope máximo para la frecuencia de muestreo de 100 MSps, también tiene un tope mínimo de 195,3125 kS/s, ósea $100 e^6 / 512$. Esto también significa que, el mínimo ancho de banda que el USRP captura en pasobandas es 195,3125 [kHz], que equivale a 197,65625 [[kHz]] en bandabase.

La señal resultante es luego enviada por el cable gigabit ethernet. En conclusión, el USRP captura una señal en toda la banda de 40 MHz, pero la procesa para enviar a la Rx GNU App el ancho de banda que el Usuario P configure en función del Coeficiente de diezmado. También es claro que el usuario está obligado a realizar el diezmado, ya que de otra suerte, estaría obligando al USRP a entregar una tasa de muestras que no es soportada por el sistema que une el USRP con el computador, o incluso por la capacidad del mismo computador.

Figura 10: Análisis en banda base de frecuencias de muestreo y anchos de banda que se manejan en el USRP 2920 en el modo de recepción.



En la figura anterior, vemos que la frecuencia de muestreo de 100 MS/s que por defecto maneja el DAC está sobredimensionada con respecto al ancho de banda del filtro analógico paso bandas usado como parte del down converter que es de 20 MHz, para capturar una señal de hasta 40 MHz. Pero esta a su vez está sobredimensionada con respecto a lo que puede realmente viajar por el cable Gigabit Ethernet, lo que puede procesar el computador y en general, por lo que en realidad desea el usuario P.

Los usuarios P que son principiantes cometan a menudo el error de usar un K_d no entero, el sistema lo puede aceptar, pero lo redondea hacia arriba, para obtener el valor entero, como consecuencia la frecuencia de muestreo que entrega el USRP puede ser un tanto mayor a la que el usuario cree que va a recibir, por lo tanto, se pueden producir una inconsistencia entre la frecuencia de muestreo usada por el Usuario P y la que realmente es, con lo cual se pueden producir distorsiones de señal en los siguientes bloques que conecte. Es importante que el Usuario P se asegure que la frecuencia de muestreo que él ha programado es la que el USRP le entrega, para ello, el GRC, como herramienta de programación tiene una ventana de texto donde aparece la advertencia: *Target sample rate:* tantos MHz, *Actual sample rate:* tantos MHz. El primer valor es lo que el usuario programó, el segundo, lo que el sistema pudo dar.

En resumen, en recepción, cuando se desea conocer los datos de configuración del USRP, tenemos los siguientes cálculos.

Los datos de entrada:

- El Ancho de banda designado al canal: B
- La frecuencia de muestreo interna del receptor del USRP: $samp_rate_usrp_rx = 100 \text{ MHz}$
- El máximo valor para el coeficiente de diezmado del USRP: $K_{dmax} = 512$

Los cálculos a realizar:

- Frecuencia de muestreo deseada: $samp_rate_d = B$
- Coeficiente de diezmado deseado:

$$Kd_d = \text{int}(samp_rate_usrp / samp_rate_d),$$

donde $\text{int}(x)$ significa parte entera de x . de manera que con lo anterior ocurre un redondeo hacia abajo.

- Nueva aproximación al coeficiente de diezmado:

$$K_{da} = 2^{int(\log_2 K_{dd})}$$

- Coefiente de diezmado final:

$$K_d = \min(K_{da}, K_{dmax}) ,$$

donde $\min(x, y)$ significa el menor valor entre x e y .

- La frecuencia de muestreo a la cual queda configurado el USRP es:

$$samp_rate = samp_rate_usrp / K_d$$

- El ancho de banda resultante, con el cual queda configurado el USRP es:

$$B_r = samp_rate$$

En transmisión, cuando se desea emitir un ancho de banda B. El proceso es el mismo con las siguientes aclaraciones:

- $samp_rate_usrp_tx = 400 \text{ MHz}$
- K_{dtxmax} : No lo conocemos para el equipo en cuestión, quizá es 4 veces mayor al usado en recepción, ya que la frecuencia de muestreo en el transmisor es 4 veces mayor a la usada en recepción.
- Se supone que el transmisor realiza una interpolación en vez de un diezmado. Si se quiere ser muy estricto en esto, lo que hay que hacer es:
 - Calcular el coeficiente de interpolación como: $K_i = 1/K_{txd}$.
 - Luego obtener $samp_rate = samp_rate_usrp_tx * K_i$.
 - Pero la verdad, lo anterior se puede saltar, pues igual resulta que:

$$samp_rate = samp_rate_usrp_tx / K_{txd}$$

En la figura 11 se analizan los posibles valores de configuración que se pueden usar con el USRP 2920 y vemos que da lo mismo hacer el análisis con los datos del receptor o los de transmisor, ya que el parámetro de configuración que piden los bloques que controlan el USRP es solo la frecuencia de muestreo $samp_rate$, mientras que el coeficiente de diezmado es más bien algo que el equipo calcula de manera interna a partir de la frecuencia de muestreo que configuremos.

Figura 11: Tabla sobre un análisis de valores de frecuencia de muestreo que acepta el USRP 2920

En transmisión				En Recepcion				Comentario	
Frec. de muestreo de DAC (TX)	Ki=2^k dentro del USRP	frecuencias que acepta la señal que se entrega al USRP		Frec. de muestreo de DAC (RX)	Kd=2^k dentro del USRP	Frecuencias que nos puede entregar el USRP en recepción			
Hz		Hz	kHz	Hz		Hz	kHz		
400000000	1	400000000	400000	Fuera del rango de los filtros de 20 MHz en el USRP				Rango disponible	
400000000	2	200000000	200000						
400000000	4	100000000	100000						
400000000	6	66666666,67	66666,66667		100000000	1	100000000		
400000000	8	50000000	50000		100000000	2	50000000		
400000000	16	25000000	25000		100000000	4	25000000		
400000000	32	12500000	12500		100000000	8	12500000		
400000000	64	6250000	6250		100000000	16	6250000		
400000000	128	3125000	3125		100000000	32	3125000		
400000000	256	1562500	1562,5		100000000	64	1562500		
400000000	512	781250	781,25		100000000	128	781250		
400000000	1024	390625	390,625		100000000	256	390625		
400000000	2048	195312,5	195,3125		100000000	512	195312,5		
400000000	4096	97656,25	97,65625		100000000	1024	97656,25		
400000000	8192	48828,125	48,828125		100000000	2048	48828,125		
400000000	16384	24414,0625	24,4140625		100000000	4096	24414,0625		
400000000	32768	12207,03125	12,20703125		100000000	8192	12207,03125		

Del análisis anterior, se concluye que en términos generales, el USRP 2920 solo puede tomar 8 posibles valores para la frecuencia de muestreo y consecuentemente para los coeficientes de diezmado y de interpolación interna. Esos valores se muestran en la figura 12.

Figura 12: Tabla de valores de frecuencia de muestreo que acepta el USRP 2920

VALORES QUE PODEMOS CONFIGURAR EN USRP 2920			
#	Ki (TX)	Kd (RX)	Frec. de muestreo que podemos usar con los USRP (kHz)
1	16	4	25000
2	32	8	12500
3	64	16	6250
4	128	32	3125
5	256	64	1562,5
6	512	128	781,25
7	1024	256	390,625
8	2048	512	195,3125

Una alternativa interesante para configurar la frecuencia de muestreo en los USRP es crear un bloque que realice el cálculo, a continuación se muestra un ejemplo para el código realizado para un bloque ‘‘Python Module’’:

```
import math
def usrp2920(B):
    Kdmax=512
    samp_rate_usrp_rx=100e6
    Kdd=int(samp_rate_usrp_rx/B)
    Kda=pow(2,int(math.log(Kdd,2)))
    Kd=min(Kda,Kdmax)
    return samp_rate_usrp_rx/Kd
```

Con este bloque, la frecuencia de muestreo se calcula como:

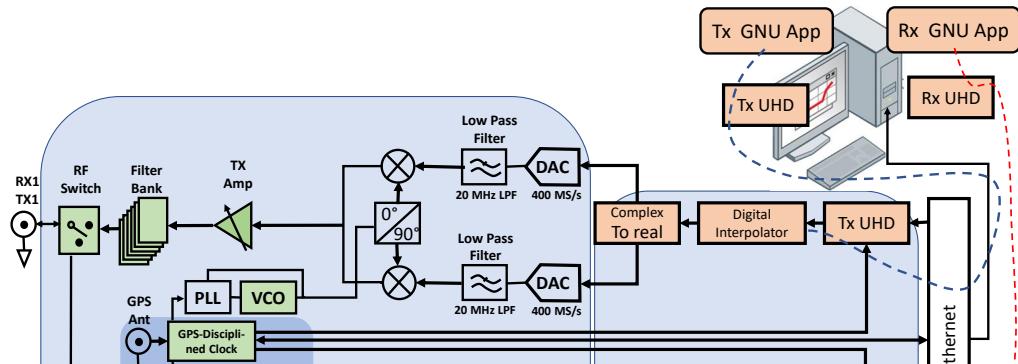
$$\text{samp_rate} = \text{samp_rate_usrp.usrp2920}(B)$$

4.3.2. El transmisor del NI USRP-2920

Puede entenderse de manera similar al receptor. En cualquier desarrollo que se realice con SDR es importante conocer en detalle las especificaciones del equipo. Para tener una idea de lo que esto representa, analizaremos a continuación las especificaciones del NI USRP-2920.

- Para una consulta más precisa, consultar el manual del equipo ²
- El ancho de banda analógico es de 20 MHz analógico, pero esto equivale en pasobandas a 40 MHz. Además entra en juego el DAC, que muestrea a una tasa fija, invariable de 400 MS/s la señal y la cuantifica a 16 bits/muestra.

Figura 13: Elementos internos del USRP para la transmisión.



²<http://www.ni.com/pdf/manuals/376358a.pdf>

En la figura 13 se observa que en modo transmisión, el USRP usa un DAC con una frecuencia de muestreo de 400 MS/s. También se observa que cuenta con el bloque Digital Interpolator para adaptar la frecuencia de muestreo de la señal que le llega. Pero al igual que ocurrió con el coeficiente de diezmado, en el modo recepción, aquí el coeficiente de interpolación también es obligatoriamente un número entero de la forma:

$$K_i = 2^m, \quad (4.2)$$

donde m es entero positivo.

4.3.3. El papel de GPS en un NI USRP.

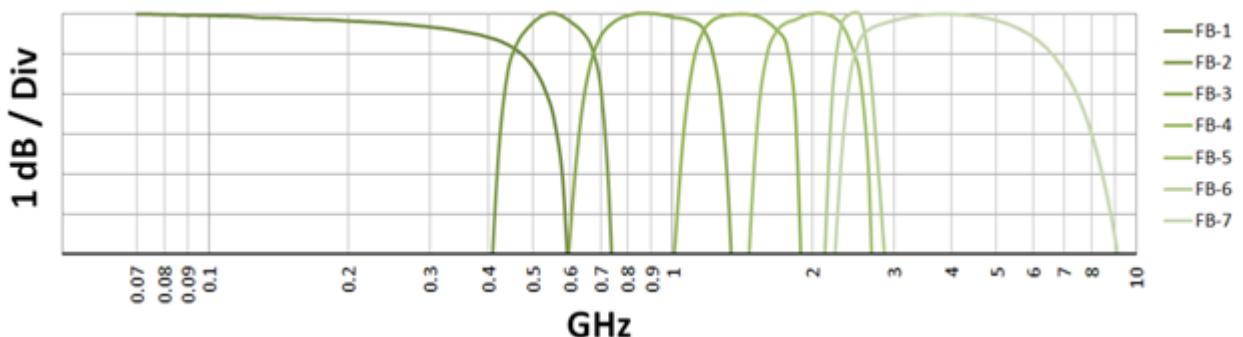
El papel del GPS consiste en usar una señal satelital, del sistema GPS para lograr que el VCO de cualquiera de los varios equipos genere una misma sinusoidal. Esto mejora enormemente la recepción de una señal transmitida con otro USRP similar, ya que no se presenta desfase entre la portadora usada en el transmisor y en el receptor. Pero esta opción solo la tienen los NI USRP-293x.

4.3.4. Los filtros en un NI USRP.

En la parte receptora

La siguiente figura presenta un ejemplo de la respuesta en frecuencia del banco de filtros, como no se tenía en el momento la información para el NI USRP 2920, se presenta la del NI USRP E-310, para que el lector tenga una idea de lo que ese banco de filtros significa.

Figura 14: Banco de filtros de la parte receptora del NI USRP E-310.



fuente: <https://www.ettus.com/content/files/USRP-E310-Product-Sheet.pdf>

Nota: no es claro que el NI USRP 2920 tenga un banco de filtros como estos, pues no aparecen en los manuales

El filtro a usar es seleccionado directamente por el Usuario P.A manera de resumen, las especificaciones del receptor del NI USRP-2920 son las siguientes:

- La información más confiable sobre las especificaciones es la del manual de equipo ³.
- Rango de frecuencias de 50 MHz a 2.2 GHz. (Las bandas de radio FM, GPS, GSM, radar, ISM).
- Pasos de frecuencia < 1 [kHz].
- Se usa un ADC de dos canales: 100 MS/s a 14 bits/muestra. Un canal es para la señal I y el otro para la señal Q, como se deduce de esta referencia ⁴.
- El rango dinámico del ADC es 1 Vp-p. La potencia de entrada al ADC es no lineal pues varía con la frecuencia.
- Rata máxima real de muestreo I/Q: 25 MS/s a 16 bits/muestra; 40 MS/s a 8 bits/muestra. El manual aclara que este valor puede verse limitado aún más por las limitaciones del computador usado y por la velocidad de la conexión entre el USRP y el computador. A diferencia de la frecuencia de muestreo del ADC, este valor se refiere a la rata de muestreo a la que realmente puede funcionar el sistema en

³<http://www.ni.com/pdf/manuals/376358a.pdf>

⁴<http://www.ni.com/white-paper/13881/en/>

tiempo real. En otras palabras, si se intentara aprovechar los 100 MS/s que entrega el ADC, veríamos que el sistema se bloquearía o produciría errores, debido a que esa velocidad no es soportada por los bloques que siguen.

- Ancho de banda máxima real: 20 MHz para 16 bits/muestra; 40 MHz para 8 bits/muestra.
- Nota 2: Se deduce también que el ancho de banda máxima instantánea en tiempo real, es el valor pasobandas que en términos reales se puede alcanzar, que representaremos como B. De modo que en bandabase, el ancho de banda es $BW=B/2$. Así toma sentido la frecuencia de muestreo que es $F_s = 2 \cdot BW = B$. Por eso, vemos que en el caso de 8 bits/muestra, con 40 MS/s se alcanza un ancho de banda de 40 MHz. Eso es lo máximo que el equipo permite. Pero el *Front-End* al que tiene acceso el usuario P, es decir, el GRC, puede no tener la opción para programar este valor de 8 bits/muestra.
- El usuario P usualmente necesita un menor valor para la frecuencia de muestreo que entrega el **USRP Source**. Para satisfacer esa necesidad, el **USRP Source** cuenta con un diezmador. El problema es que el factor de diezmado no es cualquiera, sino que se aproxima al valor más cercano entre los siguientes: 8, 16, 32, 64, 128, 256, 512.

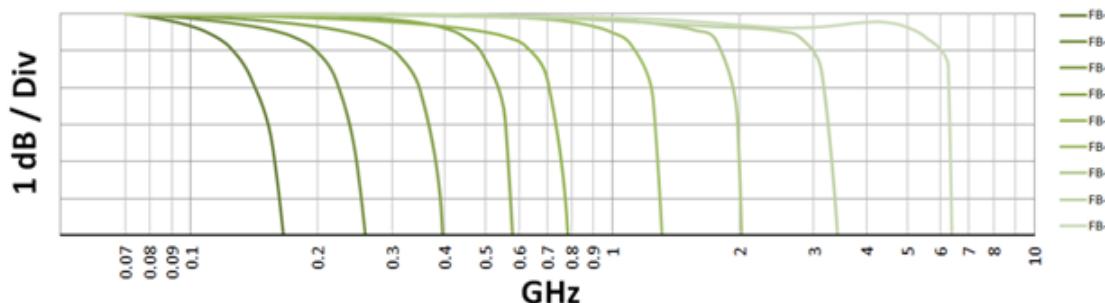
Otras notas de interés para configurar el bloque de GNU Radio conocido como **USRP Source**:

- Para los usuarios P de Simulink es un poco confuso que el muestreo sea realizado por el ADC. Esto se entiende mejor si imaginamos que el ADC del USRP junto con el procesado que realiza luego el OSP, equivalen en Simulink a la interconexión de los bloques: filtro paso bajas, muestreador (con el bloque **zero order hold**), cuantizador (en caso de que la salida del USRP source sea configurada para tipo entero). De modo que ADC+OSP lo que entregan es una señal, que usualmente veremos como una señal de valores decimales con un número finito de posibles valores. En otras palabras es la Envolvente Compleja de la señal que el usuario P desea capturar, pero en su versión muestreada y cuantizada, por lo tanto, esta señal pertenece al mundo físico.
- En Simulink de Matlab, el bloque **USRP Source** no pide frecuencia de muestreo sino coeficiente de diezmado .
- El valor máximo de frecuencia de muestreo que se pudo programar sin problemas con un computador Core i7 fue de 20 MSps, que equivale a un ancho de banda pasobandas de 20 MHz. Pero con advertencias puede ser de 25 MSps. Teóricamente, con computadores mejor dotados se podría elevar más, pero sin pasar nunca de 50 MSps.
- Aunque la documentación dice que la frecuencia de muestreo cambia en función de número de bits por muestra que se elija, vimos que al menos el bloque **USRP Source** no brinda esta opción.
- Si el parámetro Ch0: **Bandwidth** se deja en cero, el ancho de banda del receptor USRP se sintoniza por defecto a una frecuencia cercana a la frecuencia de muestreo, porque se aplica el Teorema de Nyquist en versión bandabase.
- Otros parámetros del receptor del USRP de la serie N2920 son:
 - La letra “N” se refiere a “*Network*”, debido a que usa puerto Gigabit Ethernet para la unión PC-USRP
 - La velocidad de los datos entre el PC y el USRP por el puerto Ethernet es de hasta 50 MS/s
 - Se pueden unir varios USRP como si fuesen uno solo para escalar en capacidad

4.3.5. En la parte transmisora

De manera similar, en la parte transmisora puede tenerse un banco de filtros como es el caso del NI USRP E-310, como se muestra en la figura 15

Figura 15: Banco de filtros de la parte transmisora del NI USRP E-310.



fuente: <https://www.ettus.com/content/files/USRP-E310-Product-Sheet.pdf>

4.3.6. Actividades de Caracterización del equipo

Las especificaciones anteriores son datos tomados de la documentación del equipo NI USRP-2920. Sin embargo, una verificación práctica brinda resultados más reales, pues muchas veces los manuales tienen inconsistencias. Los siguientes son verificaciones realizadas:

- Se programó el bloque **USRP Source** una tasa de muestreo de 20 MHz, para capturar una señal de ancho de banda pasobandas de 20 MHz, que puede equivaler a todo el espectro de FM. También se le programó una salida de tipo **Complex float 32**. y esto se observó:
 - Es algo perfectamente normal con un buen computador.
 - Al correr el flujobrama y observar los mensajes que aparecen en la ventana de texto inferior del GRC, dice lo siguiente: **decimation=dsp-rate/samp-rate: 5=100 MHz/20 MHz**. Eso significa que la frecuencia de muestreo fija usada por el USRP es 100 MS/s
 - Hay otro dato interesante en esa ventana de texto: *The requested decimation is odd; the user should expect CIC rolloff. Select even decimation to ensure that a halfband filter is enable* . O sea que se está pidiendo un diezmado impar y que eso tiene consecuencias en que el usuario P no reciba exactamente lo que espera, hay un filtro que no queda bien cuadrado, entonces sugiere solicitar siempre un diezmado par para asegurar que se activa un filtro pasobajas con un ancho de banda que es la mitad del ancho de banda pasobandas programado en el bloque **USRP Source**.
- Se probó la misma configuración del **USRP Source** pero con salida tipo **Complex Int16**. Se observó que:
 - Aparecen los mismos mensajes de texto, luego, la frecuencia de muestreo no se altera, es siempre 100 MS/s.
 - Se probó la misma configuración con frecuencia de muestreo 10 MHz y salida **Complex float32**. Funcionó a las mil maravillas.
 - Se probó la misma configuración con frecuencia de muestreo 25 MHz y salida **Complex float 32**, con lo cual se produce un diezmado igual a 4. Funcionó bien durante unos segundos, luego siguió funcionando, pero en la ventana de texto se podía observar una advertencia de Error: **error in pthread-set schedparam**. Se dice que sigue funcionando porque en el osciloscopio la señal continuaba apareciendo.
 - Se probó la misma configuración con frecuencia de muestreo 100 MHz y salida **Complex float32**, con lo cual se produce una diezmado igual a 1. No corre, la advertencia dice: *The hardware does not support the requested RX sample rate: Target sample rate: 100 MSps, Actual sample rate: 50 MSps.* o sea que el tope de frecuencia de muestreo que podemos pedir es de 50 MHz, si pedimos más, el sistema redondea para entregar 50 MSps, si es que el computador los soporta. Pero hay que tener en cuenta que hasta la misma velocidad del puerto Gigabit Ethernet puede ser un problema.
 - Se probó la misma configuración con frecuencia de muestreo 50 MHz y salida **Complex float32**, No corre, dice que: *Unable to set the thread priority. Performance may be negatively affected.* **Error in pthread-set schedparam**

- Se probó la misma configuración con frecuencia de muestreo 24 MHz y salida Complex float32. Funcionó con advertencias: Target sample rate: 24 MHz, Actual sample rate: 25 MHz. También mostró un error error in pthread-setschedparam

4.4. Cuidados con el uso de los equipos SDR

Es importante leer las instrucciones sobre los cuidados que se deben tener con los equipos USRP. Usualmente estos mencionan los siguientes:

- Previsión de cargas electrostáticas que pueden dañar su equipo.
- Póngase en conexión a tierra sosteniendo algo que esté aterrizado, como por ejemplo el chasis de su computador.
- Una conexión de transmisión/recepción en el mismo equipo debe realizarse con el cable suministrado y a través del atenuador de 30 dB suministrado. El atenuador debe conectarse directamente al puerto receptor.
- Se instala el software primero, antes de hardware.
- Si hace reinicio de PC, primero encienda el USRP, luego el PC.

Los siguientes son los cuidados que demandan más atención por el peligro de poder dañar las tarjetas hijas:

- El receptor es un dispositivo extremadamente sensible y puede dañarse o deteriorarse fácilmente por mal uso. Contrariamente, el transmisor es un equipo de alta potencia, si se compara con el receptor
- El uso natural del receptor sería el de recibir una señal inalámbrica que se ha propagado en la distancia y consecuentemente se ha atenuado sensiblemente. Una señal de este tipo no pone en peligro el receptor.
- Igualmente lo más natural sería usar el transmisor conectado a una antena para transmitir en la distancia.
- El problema es que en condiciones de laboratorio puede dársele otros usos al transmisor y al receptor, como estos:
 - El transmisor se conecta directamente al receptor a través de un cable. En este caso, el riesgo es para el receptor pues puede llegar a recibir una potencia más alta que la que puede soportar. La prevención consiste en conectarle previamente al puerto del receptor la carga del atenuador de 30 dB.
 - El transmisor se usa sin antena, por ejemplo para observar en un osciloscopio la señal que aparece en ese puerto. En este caso el transmisor está en riesgo, pues la energía que necesita ser emitida por la antena puede devolverse al transmisor y quemar el MOSFET. Ese fenómeno se conoce como Onda Reflejada y consiste en que la energía que no pueda ser emitida por una antena se devuelve al transmisor. Lo recomendable es que siempre se tenga una antena conectada al puerto del transmisor.

4.5. El Teorema de Nyquist aplicado en SDR

El título de este capítulo no significa que el Teorema de Nyquist cambia con el uso de GNU Radio. Solo está orientado a brindar las siguientes orientaciones claves para usar este concepto correctamente en GNU Radio:

- El Teorema de Nyquist dice que si una señal continua, banda base, tiene un ancho de banda BW, puede ser muestreada a una frecuencia de muestreo F_s que es mayor o igual a dos veces BW: $F_s \geq 2 \text{ BW}$.
- Una señal puede ser bandabase por naturaleza, por ejemplo la voz y la mayoría de los mensajes de información, pero también puede ser el resultado de hacer pasar una señal pasobandas, con un ancho de banda B, por un Down Converter. En este caso $BW = \frac{B}{2}$, por lo tanto el Teorema de Nyquist para este caso es así: $F_s \geq B$.

4.6. Equipo SDR para soluciones en modo embebido

De manera similar al SDR NI2920 los dispositivos de modo embebido como el USRP E310 se pueden caracterizar para poder encontrar los rangos de operación de manera que se pueda aprovechar eficientemente el equipo. Un dispositivo de SDR puede ser visto como como un sensor del espectro radioeléctrico (ERE) y a la vez como actuadores sobre el ERE. Además cuentan con conectividad IP mediante Ethernet y WiFi, así como con la posibilidad de convertirse en un terminal IoT.

4.6.0.1. Características de los dispositivos SDR

Como dispositivo SDR con modo embebido se tomará como ejemplo el USRP E310 que tiene sistema *OpenEmbedded* Linux con librerías GNU Radio y UHD instaladas que le permiten operar de forma autónoma. Las características del equipo E310 se resumen en la siguiente tabla:

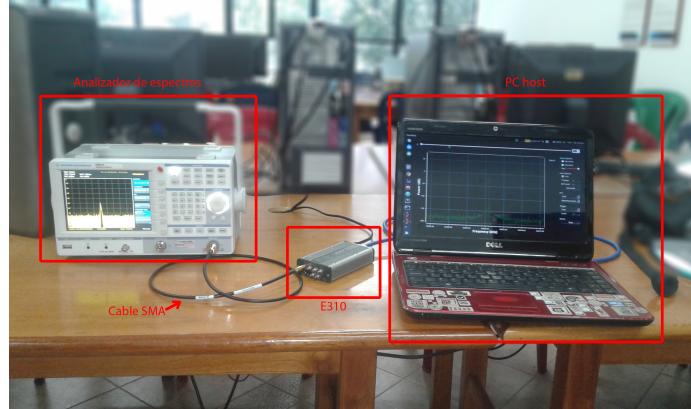
USRP E310	
Xilinx Zynq 7020 SoC	ARM Cortex A9 866 MHz con FPGA
Banco de filtros RF	Tx, Rx
UHD	V3.9.1
RFNoC	Si
Audio Jack	Si
2x2 MIMO transceiver	Si
Rango de frecuencias (Rx)	70 MHz - 6 GHz
Rango de frecuencias (Tx)	47 MHz - 6 GHz
Ancho de banda instantáneo (AD9631 RFIC)	56 MHz
SO	OpenEmbedded Linux, basado en el proyecto YOCTO Linux
Máxima potencia de entrada (Rx)	0 dBm
Máxima potencia de salida (Tx)	15 dBm

Tabla 4.1: Características generales del equipo SDR.

El transceptor AD9631 RFIC presente en el dispositivo E310 puede soportar un ancho de banda de 56 MHz, sin embargo, para procesar señales en ese ancho de banda se requiere implementar algoritmos en la FPGA usando herramientas como RFNoC, las cuales no se consideran para el desarrollo de este curso, sin embargo, este módulo se puede aplicar en sistemas digitales.

La tasa de muestreo máxima que se puede obtener de la FPGA a los núcleos ARM (procesador) es de 16 MS/s, lo cual se puede comprobar usando las herramientas del UHD `rx_samples_to_file` en el PC host, o corriendo un programa en modo embebido en el equipo E310 donde se varía la tasa de muestreo, de manera que cuando se superan los 16 MS/s se generan alertas en el UHD que indican que se ha superado la tasa de muestreo permitida. Otra característica importante es encontrar una relación entre los valores de ganancia Tx utilizados por software y los valores reales de ganancia Tx. Esto se puede realizar utilizando un E310, operando en modo transmisor, un computador portátil o de escritorio (PC host) y un analizador de espectros como se muestra en la Figura 16. La prueba de caracterización consiste en:

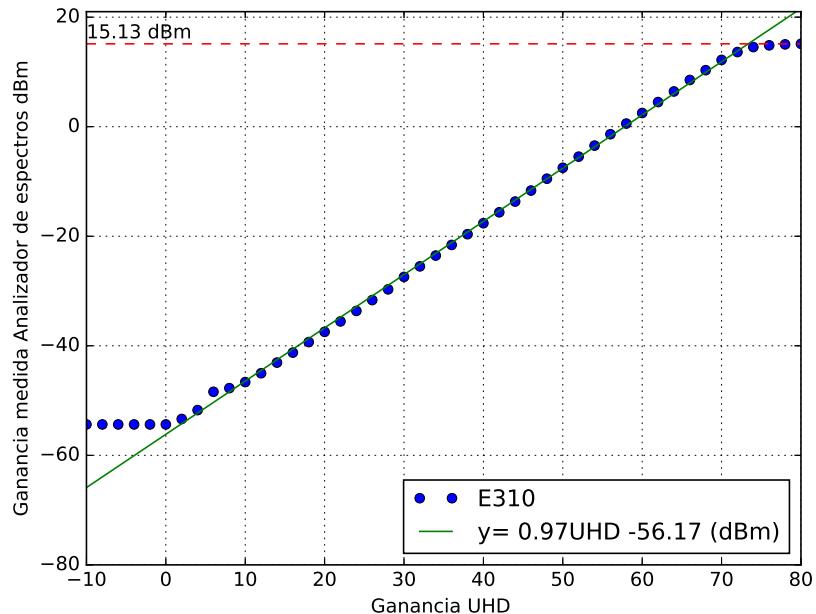
Figura 16: Montaje experimental para encontrar la relación entre la ganancia del UHD y la potencia de salida del dispositivo E310.



- Generar tonos desde el E310 a diferentes frecuencias (dentro de la banda 88-108 MHz) para verificar la linealidad del amplificador Tx presente en el equipo E310.
- Generar un tono desde el E310 a frecuencia central o frecuencia de portadora que esté en la banda VHF 88-108 MHz.
- Fijar una frecuencia central e ir variando la ganancia del UHD en GNU Radio, esto se puede hacer en modo embebido o en modo *network* a través de la interfaz GUI de GNU Radio.
- Tabular y graficar los datos obtenidos por el analizador de espectros para encontrar la zona lineal de operación.

La Figura 17 muestra la relación existente entre los valores de ganancia dados por software en el UHD y la potencia medida por el analizador de espectros, la potencia máxima de salida del dispositivo E310 es 15.13 dBm, la gráfica verde muestra la aproximación lineal entre la ganancia dada en UHD y el valor real medido.

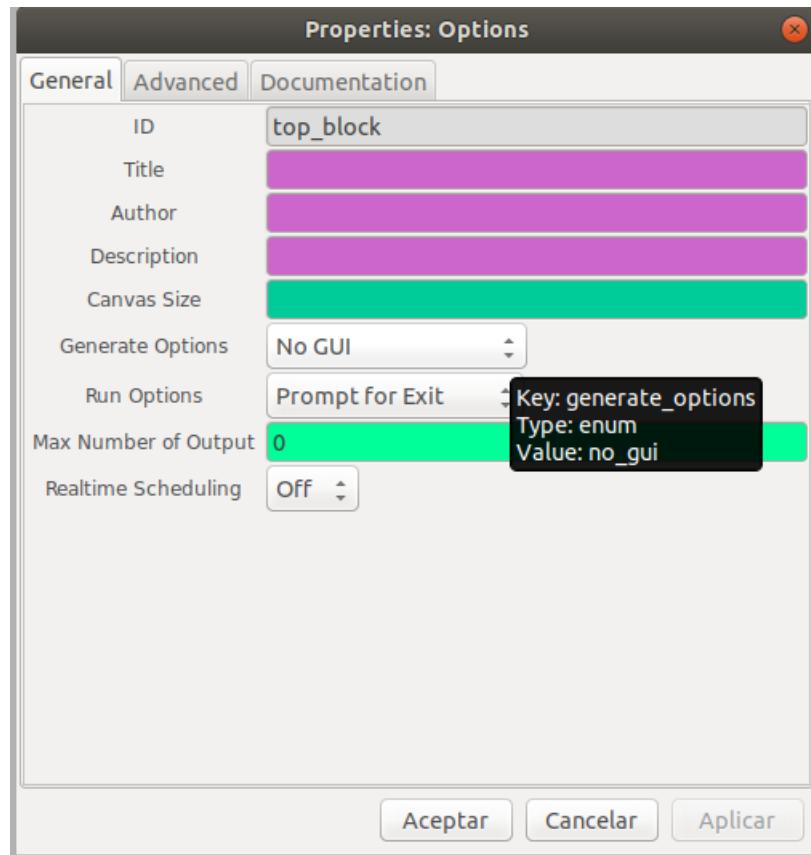
Figura 17: Relación existente entre la ganancia del UHD y la potencia medida con el analizador de espectros.



4.7. Uso del USRP E310 en modo embebido

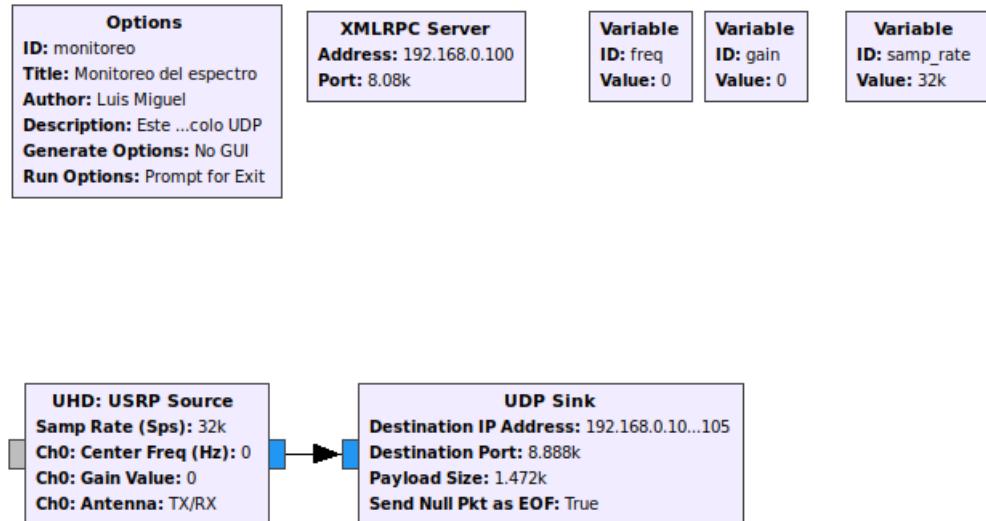
1. Diseñar el flujograma en GNURadio y eliminar todas las interfaces gráficas como se muestra en la figura 18.

Figura 18: Modo sin interfaces gráficas.



Para mostrar el funcionamiento del modo embebido se realizará el siguiente flujograma donde se realizará el control remoto del USRP E310 a través de un servidor XMLRPC que se ejecuta dentro del USRP y se transmitirán los datos del espectro a través de datagramas UDP como se muestra en la figura 19.

Figura 19: Flujograma en modo No GUI. (flujograma usado *flujograma_modo_embebido.grc*)



2. Generar el archivo de python correspondiente al flujograma realizado, por ejemplo: `monitoreo.py`.
 3. Descubrir la IP del E310 usando `arp-scan` de la siguiente forma:
- ```
sudo arp-scan --localnet
```
- La salida del comando retornará las direcciones IP correspondientes a la red del área local donde se encuentra conectado el computador como se muestra en la figura 20.

Figura 20: 192.168.0.100 dirección IP de ejemplo para el equipo SDR E310 luego de ser descubierta por `arp-scan`.

```

uis-e3t@uise3t-HP-ProDesk-600-G4-SFF: ~
Archivo Editar Ver Buscar Terminal Ayuda
uis-e3t@uise3t-HP-ProDesk-600-G4-SFF:~$ sudo arp-scan --localnet
[sudo] contraseña para uis-e3t:
Interface: eno1, datalink type: EN10MB (Ethernet)
Starting arp-scan 1.9 with 256 hosts (http://www.nta-monitor.com/tools/arp-scan/)
)
192.168.0.1 c0:4a:00:e4:4d:68 TP-LINK TECHNOLOGIES CO.,LTD.
192.168.0.100 00:80:2f:21:c8:5c NATIONAL INSTRUMENTS CORP.

2 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9: 256 hosts scanned in 2.417 seconds (105.92 hosts/sec). 2 responded
uis-e3t@uise3t-HP-ProDesk-600-G4-SFF:~$

```

4. Conectarse por ssh al E310:

```
ssh root@192.168.0.100
```

5. Copiar el archivo python al E310 mediante scp, por ejemplo:

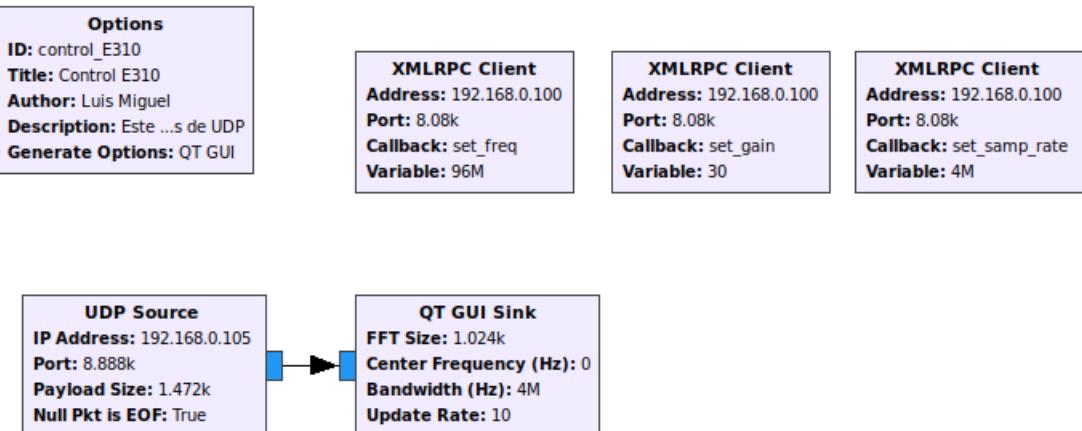
```
scp monitoreo.py root@192.168.0.100:/home/root/libro_comunicaciones
```

6. Ejecutar el programa de python copiado al radio, por ejemplo:

```
python /home/root/libro_comunicaciones/monitoreo.py
```

7. Ahora en el PC host realiza el flujograma que se encarga de establecer un cliente XMLRPC para poder controlar el radio con parámetros como frecuencia central, frecuencia de muestreo, y ganancia del receptor como se muestra en la figura 21.

Figura 21: Lectura del espectro a través de UDP. (flujo gráfico usado *flujo\_pc\_remoto.grc*)



Al ejecutar ambos fluojogramas se tendrá control del dispositivo SDR que está ejecutando un programa en modo embebido.

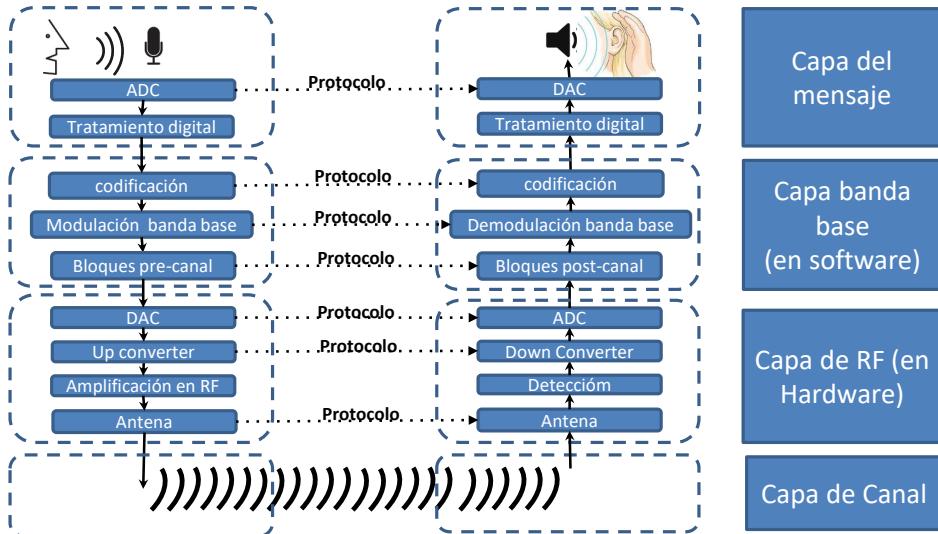
**Nota:** Se debe tener en cuenta que la dirección IP utilizada no es fija, es decir, el radio está configurado en modo DHCP por lo tanto cuando ejecute los fluojogramas deberá colocar la IP del radio luego de ejecutar `arp-scan` y la IP del PC host para poder escuchar por UDP.

#### 4.8. Un sistema de radiodifusión

Podría decirse que la Radio Definida por Software representa el uso de la teoría de señales y sistemas discretos llevada a la práctica. La aplicación práctica implica una relación entre el mundo físico que es por naturaleza continuo y el uso de técnicas de computación que es por naturaleza discreto.

Para entenderlo mejor, vamos a volver al sistema de radiodifusión expresado en el modelo de capas de la figura ???. Un sistema de radiodifusión es por su origen analógico tanto por la naturaleza del mensaje como de los equipos usados y la señal emitida. Sin embargo, con los avances tecnológicos, las soluciones que hoy se tienen en el mercado combinan tecnología digital y analógica, ya que no tiene lógica alguna usar hoy tecnología analógica para crear un sistema de comunicación por el solo hecho de que ese sistema es analógico. Es por esa razón que cuando hacemos una visita a los estudios de una emisora AM o FM encontramos que todos los equipos nuevos son digitales, aunque AM y FM son modulaciones analógicas. Para entenderlo mejor, en la figura 22 se presenta un modelo de capas para un sistema de radiodifusión analógico implementado con tecnología digital, que en realidad es una combinación de tecnologías analógicas y digitales. La combinación se realiza de la siguiente manera:

Figura 22: Modelo de capas de un sistema de radiodifusión analógico implementado con tecnología digital



- Se conserva la naturaleza analógica del mensaje y de la señal emitida.
- Tan pronto se tiene la señal del mensaje, se traduce a datos mediante un conversor Análogo Digital (ADC). Este proceso implica el uso de algunos conceptos como el Teorema de Nyquist para realizar un muestreo adecuado a la señal; la cuantificación; la codificación PCM (del inglés Pulse Code Modulation).
- Lo que sigue es el uso de métodos numéricos que se implementan en un equipo de cómputo tanto para realizar un tratamiento digital del sonido del mensaje pero también, realizar la modulación.
- En lugar de usar frecuencias intermedias, la modulación se realiza en banda base, que equivale a obtener una señal que tenga el mismo espectro de la señal modulada tradicionalmente, pero ahora, centrado en la frecuencia cero.
- Lo anterior implica nuevos retos teóricos y técnicos, pues la señal modulada en banda base resulta ser compleja y se conoce como la Envolvente Compleja.
- El reto teórico consiste en la necesidad de saber modular y demodular señales usando el concepto de Envolvente Compleja.
- El reto técnico aparece con la capa que hemos llamado Capa RF, donde predominan tareas con equipo analógico analógicas. Antes de entregar la señal a esta capa es necesario preparar la señal para su paso a un medio analógico, eso es lo que hemos llamado Bloques Pre-canal. Luego, la señal pasa al dominio continuo usando un Conversor Digital Analógico (DAC).
- El Up converter es un circuito que permite mover el espectro de la señal recibida para que quede centrado en la frecuencia RF.
- La antena simplemente traduce la señal eléctrica en señal obtenida en señal electromagnética.
- Es claro que en la parte receptora se tienen componentes que hacen pareja con los mencionados anteriormente, para recibir la señal, desplazar el espectro de esa señal a la frecuencia cero, usando un Down Converter y así sucesivamente.

Este modelo deja precisamente claro la tendencia natural para la implementación de los futuros sistemas de comunicaciones basados en Software Defined Radio (SDR). Básicamente, un sistema de SDR tiene las siguientes características:

- Tienen una componente de software y una de hardware.
- En la componente de hardware se realizan unas tareas que son comunes para cualquier sistema de comunicaciones: mover la señal en el dominio de las frecuencias para centrarla en la frecuencia RF, amplificarla de acuerdo a las necesidades, transmitirla en forma de ondas de radio. En fin, la componente de hardware corresponde a la Capa RF en la figura 22. Las prácticas propuestas en este libro se apoyan mayormente en un hardware SDR de National instruments conocido como USRP.

- La componente de software se aloja en un computador y es allí donde se implementan soluciones para sistemas específicos de comunicaciones. A diferencia de lo que ocurre en los sistemas analógicos, los métodos que aquí se implementan son muy cercanos a los teóricos, por ejemplo, una modulación FM puede estar dada por una o varias fórmulas o por uno o más algoritmos.
- En la componente de software la programación se realiza para señales banda base, para lo cual existen diversas herramientas de programación. La que usaremos en el presente libro se conoce como GNU Radio. Se trata principalmente de una librería que reúne las experiencias en el tema de una gran comunidad de desarrolladores.
- Al lado de GNU Radio, también se cuenta con una herramienta conocida como GRC, que permite realizar la programación gráfica, usando bloques, de manera similar a lo que se puede lograr usando Simulink de Matlab. De hecho, Simulink de Matlab también puede ser usada para implementar soluciones SDR.

#### 4.8.1. La Conversión Análogo Digital

Como se vió anteriormente, en la parte transmisora se requiere usar un dispositivo ADC y en la receptora un dispositivo DAC. Puede decirse que estos elementos sirven de compuerta entre el mundo continuo y el digital. Conocer muy las capacidades y las limitaciones que tienen estos dispositivos es clave para poder lograr un montaje funcional. A continuación se revisan los principales procesos y características de un ADC:

##### 4.8.1.1. El muestreo. La frecuencia de muestreo y el ancho de banda

El ADC puede ser visto como un medidor de la tensión de una señal, que mide en tiempos discretos separados entre sí en un tiempo  $T_s$ . Por lo tanto, las mediciones se presentan a la frecuencia de muestreo  $F_s = \frac{1}{T_s}$ .

Los siguientes son los aspectos que merecen mayor atención:

- De acuerdo al Teorema de Nyquist, la señal muestreada puede llegar a tener una frecuencia máxima igual a  $F_{max} = \frac{F_s}{2}$  se tiene el principal parámetro. Eso significa que un ADC está limitado en ancho de banda.
- El costo en dinero que puede tener un ADC se eleva exponencialmente con la frecuencia de muestreo que soporte.
- El muestreo es un proceso reversible, al menos desde el punto de vista teórico siempre y cuando la señal haya sido muestreada respetando el Teorema de Nyquist.

La puesta en práctica del Teorema de Nyquist es lo que produjo en los año 60 la revolución PCM, que es cuando la telefonía, siendo la red más grande del mundo adoptó la tecnología PCM en toda su dimensión. La idea consistía en tomar la señal de voz de un teléfono, limitar su ancho de banda hasta 4 [kHz], muestreala a una frecuencia de muestreo de 8 [kHz], con lo cual el periodo de muestreo o distancia de muestra y muestra sería de  $125\mu\text{seg}$ . Esa distancia de tiempo entre muestra y muestra se aprovechó para enviar allí muchas más señales de voz también muestreadas. Se elevó entonces la capacidad de las redes, pues ya no era necesario tener un par de hilos de cobre entre dos puntos para conducir cada llamada telefónica, pues muchas llamadas telefónicas podían ahora viajar sobre un mismo par de hilos de cobre.

Usualmente los estudiantes no se convencen del mensaje que envía el Teorema de Nyquist y que puede enunciarse así: si tu muestreas una señal usando una frecuencia de muestreo que sea igual o superior a dos veces la frecuencia máxima que esa señal pueda llegar a tener, siempre podrás recuperar, sin pérdida alguna, la forma continua de esa señal a partir de la versión discreta. Ó sea que el proceso de muestreo para dejar de transmitir la señal todo el tiempo y hacerlo solo por instantes, no implica que la señal continua haya dejado de existir.

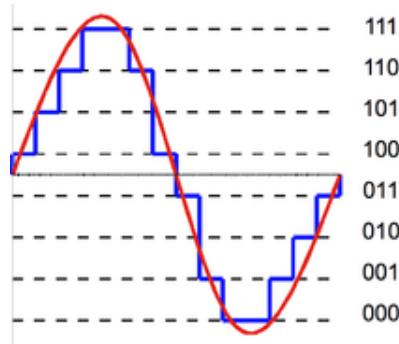
##### 4.8.1.2. La cuantificación. El Diapasón de cuantificación

En un ADC La cuantificación es inseparable del proceso de muestreo. Surge debido a que un ADC nunca puede llegar a entregar una señal muestreada como la que teóricamente esperamos. Esto es debido a las siguientes posibles razones:

- La salida del ADC es un número acotado en bits. Algunos pueden permitir una configuración, por ejemplo para usar 32 bits/muestra o 64 bits/muestra. Los usados en telefonía fija deben cumplir una recomendación de la UIT-T que establece que son 8 bits/muestra.
- Por lo anterior, el proceso de cuantificación equivale a una especie de muestreo en amplitud. Pero la cuantificación no es un proceso reversible como sí lo es el muestreo cuando se cumple el Teorema de Nyquist.

En la figura 23 se presenta una comparación entre una señal continua y su versión cuantificada a la razón de 3 bits/muestra.

Figura 23: Comparación entre una señal continua y una cuantificada a la razón de 3 bits/muestra



- Por lo anterior, un ADC tiene unos límites inferior y superior para la amplitud de la señal que puede entregar
- Pero algo no menos relevante es que un ADC también tiene unos límites inferior y superior para la señal que recibe. La diferencia entre esos dos límites es lo que se conoce como Diapasón de cuantificación.
- Los siguientes son los problemas más comunes con el uso de ADC:
  - La señal que entra al ADC puede ser tan débil que la cuantificación la deforma completamente porque queda expresada con muy pocos niveles de amplitud.
  - La señal que entra al ADC puede ser tan fuerte que se sale del diapasón que soporta, de modo que las amplitudes que se salen del diapasón desaparecen deformando la señal.
  - La señal entrante puede tener un ancho de banda superior al soportado por el ADC, lo cual hace que la señal resulte irrecuperable.

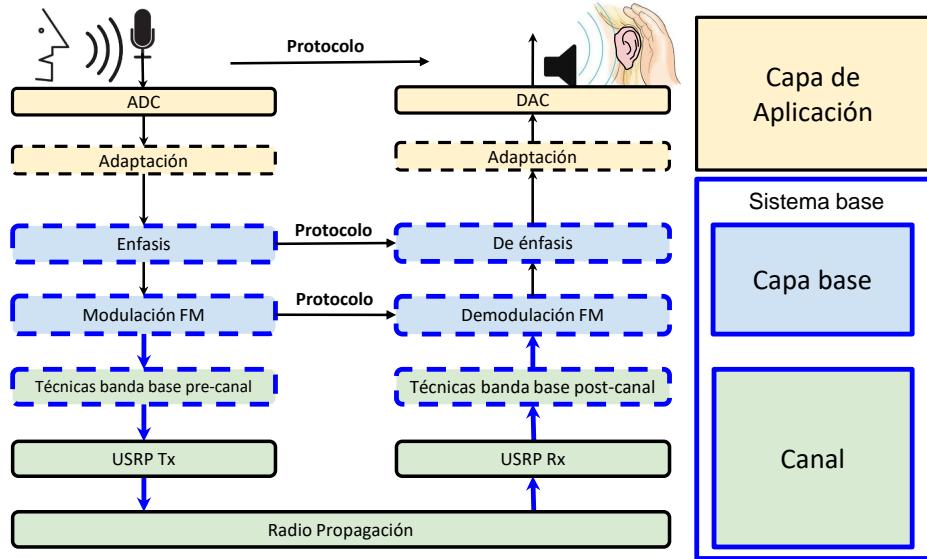
#### 4.8.2. La Conversión Digital Análoga

La Conversión Digital Análoga es el proceso inverso de la Conversión Análoga Digital. En este sentido, el DAC tiene parámetros similares al ADC.

### 4.9. Implementación de un sistema funcional de Radiodifusión FM basado en SDR

El propósito ahora es llegar a implementar un sistema de comunicaciones FM con USRP, completamente funcional aprovechando las implementaciones que ya se han realizado hasta el momento. Al parecer solo resta reemplazar el canal AWNG por los USRP, sin embargo esto trae consigo algunos retos que debemos superar.

Figura 24: Modelo para un sistema de radiodifusión FM



En la figura 24 tenemos el modelo sobre el cual nos vamos a basar. Si lo comparamos con el modelo que teníamos para una implementación simulada 8 tenemos varias diferencias:

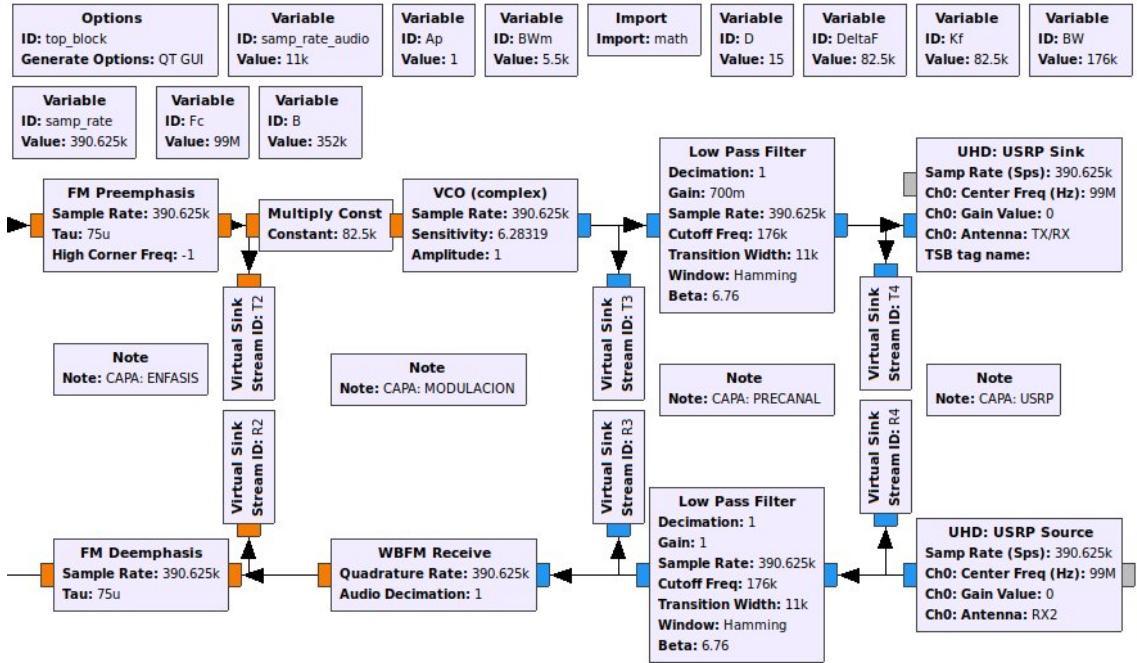
- Por primera vez aparece como parte del canal una sub capa que hemos llamado Radio propagación, la cual se refiere a que la señal modulada deberá viajar en forma de ondas electromagnéticas, afrontando los diversos fenómenos de propagación que se presenten.
- Tenemos una nueva sub capa que podemos llamar USRP Tx/USR Rx.
- Las dos sub capas anteriores pertenecen al mundo físico y por eso los recuadros están dados por líneas continuas y negras.
- Ahora tenemos de manera muy explícita que la modulación es FM.
- También hemos incluido la sub capa Enfasis/De enfasis. Le hemos asignado un protocolo ya que la pareja de bloques de esta capa se entienden por medio de un método que cada uno conoce.

#### Sobre la sub capa de radiopropagación.

Se trata de un medio donde tenemos un control muy limitado. Sin embargo, tenemos varios aspectos a tener en cuenta:

- Debemos asegurarnos de contar con los elementos RF necesarios y conocer sus características y limitaciones, incluyendo: las antenas, conectores, cables, el USRP y hasta el computador y cable que une USRP con el computador. Todos ellos pueden influir en el proceso de la comunicación.
- Debemos elegir sobre qué banda vamos a transmitir, previa revisión de que esa banda esté libre y que no estemos causando interferencias por fuera las normas establecidas.
- Debemos buscar un área de trabajo donde existan pocos obstáculos, teniendo en cuenta que los métodos implementados apenas si consideran la afectación por ruido blanco de banda angosta.

Figura 25: Flujograma de un sistema de radiodifusión FM.



#### Sobre la sub capa USRP-Tx/USRP-Rx.

Esta sub capa tiene varias novedades, para revisarlas nos remitimos al flujograma propuesto para la implementación del sistema de radiodifusión sonora FM presentado en la figura 25, donde podemos observar que:

- Lo más relevante en esta sub capa es conocer las características de la tecnología que se está usando. En este sentido, vale la pena destacar los principales errores que una persona comete aquí:
  - Se selecciona una frecuencia de muestreo que no soporta el hardware. Por ejemplo, ya se ha dicho que los USRP, a pesar de brindar un amplio rango de frecuencias para la comunicación, en términos de frecuencia de muestreo soporta solo ciertos valores que han sido especificados en la tabla 12.
  - Se viola el rango dinámico que el hardware soporta. Por ejemplo, en la mayoría de equipos USRP la magnitud de la señal a la entrada del USRP debe ser inferior a uno, podría ser por ejemplo 0,8. Es necesario asegurarnos de que así esté ocurriendo, de lo contrario, la señal que se enviará no será la que deseamos, estará distorsionada y eso se traducirá de una u otra manera en una disminución de la calidad. Se recomienda observar esa señal y de ser necesario, ajustar su magnitud, usando la variable que hemos incluido para ello: "Gain\_Tx\_precanal".
  - También hemos incluido la variable "Gain\_Tx\_USRP" para controlar la ganancia en Db de los amplificadores de RF con que cuenta el USRP para la transmisión, Ver "TX Amp" en la figura 6. La importancia de hacerlo depende de la distancia a la cual se encontrará del receptor, de modo que la señal que reciba no resulte sobre amplificada por encima de los niveles de señal que soporta el receptor, lo cual también se puede traducir en una baja calidad en la comunicación. Esto puede ocurrir cuando el transmisor y el receptor se encuentran ubicados a pocos cuantos metros, por ejemplo hasta 5 metros.
  - Hemos incluido también la variable "Gain\_Rx\_USRP": para controlar la ganancia en Db de los amplificadores de RF que se tienen dentro de un USRP, en la parte receptora, ver "Drive Amp" en la figura 6. Como ya se ha dicho, la parte receptora también tiene un rango dinámico y la falla más común es que la señal que se recibe sea tan débil que queda por fuera de ese rango y requiera una amplificación previa, antes de pasar por el Down Converter. Sin embargo, también es común el caso contrario: que la señal tenga demasiada amplificación, ya que el

receptor se ubica cerca al transmisor y también quede por fuera del rango, pero esta vez por sobre amplificación. Aunque no lo hemos puesto a prueba con FM, es posible que un bloque que viene con GNU Radio, conocido como AGC (del inglés Automatic Gain Control) ayude no solo a obtener la amplificación ideal para la señal recibida, sino que también es posible que sirva como limitador para el ruido que puede llegar en la amplitud de la portadora. Esto es debido a que en el capítulo 3.5.11 se habló de la necesidad de un limitador, pero nunca se propuso una solución para implementarlo, es posible que el bloque AGC pueda servir como solución.

- En la parte transmisora hemos incluido el Filtro paso bajas representado en el bloque “Low Pass Filter” con el fin de prevenir que ocurra algún tipo de emisión por fuera de la banda prevista. Es posible que este filtro no sea necesario ya que el USRP tiene su propio filtro, cuyo ancho de banda se puede configurar mediante el parámetro “Ch0:Bandwidth”. Lo mismo ocurre en la parte receptora, queremos evitar saturar al discriminador con el ruido que puede haber por fuera de la banda de interés, pero igualmente el USRP al parecer ofrece esa posibilidad.

## *Capítulo 5*

### *Modulación OOK, ASK, BPSK*

Este material aun no ha sido producido, pero si tenemos fuentes que hemos producido y usado en clase.  
A continuación se listan:

*Capítulo 6*  
*Modulación FSK*

# Capítulo 7

## *Un sistema de radio comunicación digital sobre un canal guiado*

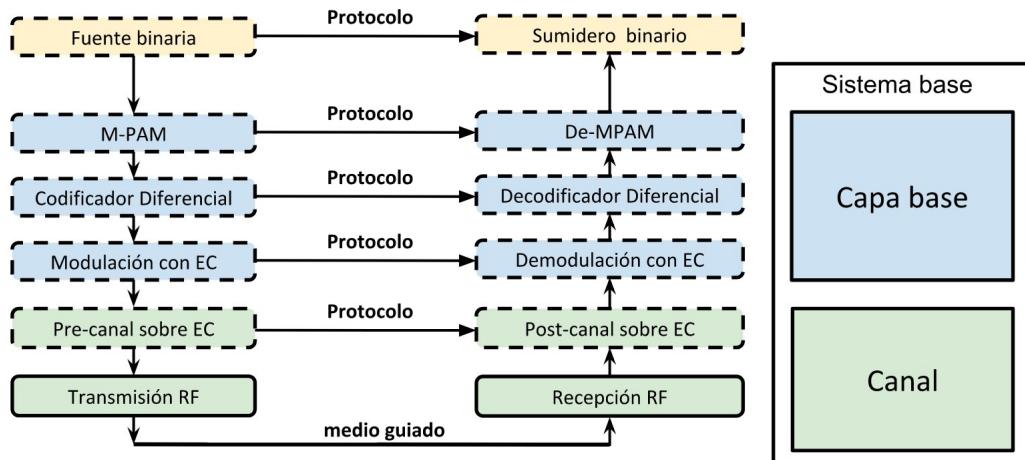
Está claro que GNU Radio está pensado en las comunicaciones inalámbricas principalmente, pero hemos decidido iniciar este estudio con un medio guiado, debido a que este tipo de medios tiene parámetros más fáciles de controlar que el caso del canal inalámbrico que es un medio no guiado, donde las señales digitales son altamente sensibles a los diversos fenómenos de propagación, que en el caso de las ondas electromagnéticas, son muy complejos. Cada fenómeno de propagación requiere un acercamiento y una solución particular, este acercamiento particular se torna difícil cuando todos los fenómenos se presentan al mismo tiempo sin poder ser aislados, esto lo que ocurre con el canal inalámbrico. Por ejemplo, si lo que se desea estudiar son los métodos para luchar contra el ruido, es necesario crear las condiciones para poder experimentar una transmisión sobre un canal que solo tenga ruido y donde los demás fenómenos estén ausentes.

Ahora, desde el punto de vista de aplicación práctica, usar medios guiados para enviar señales de RF no es algo tan descabellado. Años atrás los operadores de telecomunicaciones vieron como sus “redes de cobre se fueron transformando en oro” gracias a técnicas RF que les permitieron sacarle el máximo provecho a todo ancho de banda que tiene el cobre para ofrecer servicios convergentes combinando no solo telefonía, sino televisión e internet, en fin todos los servicios básicos de comunicación en un solo par de hilos. Hoy mismo estamos viviendo una agresiva expansión de la tecnología conocida como Fibra Óptica hasta el Hogar (FTTH, del inglés *Fiber to the Home*) y también está en boca del desarrollo de soluciones ópticas basadas en SDR incluyendo también la técnica conocida como RF sobre fibra óptica.

### 7.1. Modelo de capas a seguir para la implementación

El modelo que nos guiará en la discusión y en la implementación es el que se propone en la figura 1. De este modelo podemos identificar varios requerimientos como los siguientes:

Figura 1: Modelo de capas para una comunicación digital de voz por RF sobre un medio guiado.



Hay un dilema por resolver: timing y time alignment surgen como consecuencia de filtros de transmisión y recepción, pero se tratan en serie en capítulo 7. Pero timing se viene tratando, pero nada sobre time alignment. Quizá valga la pena tratarlos al menos desde el punto de vista de una solución manu

- El modelo consta de tres partes:
  - **El canal:** Nuestro canal sigue siendo un medio guiado, eso nos permite obviar por ahora varios fenómenos de propagación del canal inalámbrico, pero a diferencia del medio guiado ya visto, las señales viajan en RF, es decir, sobre una onda portadora a altas frecuencias, o lo que se conoce como pasobandas.
  - **Capa base:** Busca satisfacer las necesidades de comunicación para una aplicación superior que entrega información binaria que debe viajar a la mayor velocidad posible por un canal RF limitado en ancho de banda.
  - **Capa de aplicación:** Se trata de todo aquello que podamos enviar por ese canal para satisfacer necesidades de los usuarios, como archivos, voz, videos, imágenes. Solo que por ahora, esa aplicación está siendo simulada por una fuente binaria.
- **La modulación digital:** Está pensada para resolver el problema de acomodar, de alguna manera, un mensaje binario sobre una portadora de RF. Podemos decir que esa modulación consta de dos partes:
  - La que realiza la subcapa “Modulación con EC”, la cual genera la señal modulada, pero en versión bandabase, esto no es otra cosa que la Envoltoría Compleja de la señal modulada.
  - La que realiza el equipo SDR, que se ubica en la subcapa “Transmisión/Recepción RF”.
- **Las técnicas pre-canal:** Son similares a las tratadas en el capítulo anterior, solo que ahora deberán lidiar con señales moduladas en versión de su Envoltoría Compleja. Las técnicas pre-canal buscan preparar la señal para que sea entendida por el DAC que hay en el USRP; introducir información clave para luchar contra diversos fenómenos de propagación. En la parte receptora, las técnicas post canal, buscan descubrir entre la señal recibida aquello que pueda ser entendible por parte de la demodulación digital.
- **Subcapa Transmisión/Recepción RF:** Es donde está el equipo SDR. En el modo de transmisor se ocupa de convertir a continua la señal recibida y desplazar su espectro de manera que resulte centrado en la frecuencia de RF deseada. Es obvio que en el modo de recepción se realiza el proceso inverso.

## 7.2. La modulación digital pasobandas y su versión banda-base

Este capítulo explica la modulación digital paso bandas desde un enfoque poco común en la literatura, pero muy necesario para el estado tecnológico de hoy. Los siguientes son los conocimientos necesarios para poder llegar a realizar una implementación real basada en Software Defined Radio con un equipo como el USRP para realizar una transmisión real.

- La señal con modulación paso bandas es una onda senoidal de frecuencia  $f_c$  que lleva información codificada ya sea en la fase, la amplitud, la frecuencia, o una combinación de esos parámetros, los cuales, consecuentemente, varían en el tiempo, de acuerdo a la siguiente ecuación:

$$y(t) = B(t)\cos[2\pi f_c t + \varphi(t)] \quad (7.1)$$

- Los sistemas de comunicaciones modernos, como las comunicaciones móviles 4G o los enrutadores WiFi no usan un solo tipo de modulación, como en los tiempos de antes. Más bien escogen de manera inteligente la modulación que mejor se adapta a las condiciones del canal en el momento. Además, usan mayormente la fase y la amplitud de la portadora, lo cual se puede representar como en la ecuación anterior.
- Pero existen sistemas de comunicación que usan la frecuencia para llevar la información, lo cual también se basa en la ecuación anterior ya que fase y frecuencia conservan la siguiente relación:

$$f(t) = d\varphi(t)/dt \quad (7.2)$$

de manera que

$$\varphi(t) = \int_0^t f(\tau)d\tau \quad (7.3)$$

de manera que para estos casos la ecuación sería:

$$y(t) = B \cos[2\pi f_c t + \int_0^t f(\tau) d\tau] \quad (7.4)$$

- Como ya se ha visto, si este concepto se lleva a SDR con USRP, la señal paso bandas anteriormente descrita es la que entrega el USRP, mientras que la programación en SDR debe enfocarse en producir la señal que se le debe entregar al USRP que no es otra cosa que la Envolvente Compleja  $y_{CE}(t)$  de  $y(t)$ . Eso significa que, programar una modulación en SDR para un sistema de comunicación real, es programar la versión banda base de esa modulación paso bandas, que no es otra cosa que la Envolvente Compleja.

$$y_{CE}(t) = 2B(t)e^{j\varphi(t)} \quad (7.5)$$

- Desde el punto de vista de la Envolvente Compleja, el mensaje no modula parámetros de una portadora senoidal, sino a un parámetro de la Envolvente Compleja de esa portadora senoidal que es una señal exponencial compleja. En otras palabras, la portadora, para el caso de la Envolvente Compleja, es una señal exponencial compleja. También podemos decir que lo que se modula es una portadora en versión banda base o en versión de Envolvente Compleja.
- En los métodos de modulación más básicos, los parámetros de la señal exponencial compleja van cambiando a medida que llega nueva información que debe llevar.
- La regla que define como cambian los parámetros de la señal exponencial compleja puede ser expresada por medio de una tabla que es comúnmente conocida como tabla de verdad o mediante un diagrama de constelaciones.

Tabla 7.1: Plantilla para la Tabla de Verdad.

| Tabla de Verdad |                                |
|-----------------|--------------------------------|
| La clave        | valor del parámetro modificado |
|                 |                                |
|                 |                                |
|                 |                                |

La tabla 7.1 es una plantilla de la Tabla de Verdad y debe ser entendida de la siguiente manera: se supone que hay una clave digital que modula a uno o más parámetros de la Envolvente Compleja, de manera que la información pueda viajar como una variación de ese o esos parámetros. La clave usualmente está dada en forma de una combinación de  $m = \log_2(M)$  bits. O sea que, dependiendo de cómo se combine esos  $m$  bits se modificará de una u otra manera uno o más parámetros de la señal exponencial compleja. Pero, teniendo en cuenta que el bloque de modulación usado en GNU Radio exige que la entrada sea una señal M-PAM, entonces la clave puede estar dada en  $M$  posibles valores de amplitud. En el fondo, cada valor de amplitud equivale a una combinación de bits, porque la M-PAM puede ser vista como la traducción de cada combinación de bits a un valor de amplitud.

- Todos los tipos de modulación que estudiaremos a continuación pueden ser vistos como una modificación de la modulación M-PAM, ya que todos los moduladores que vamos a ver requieren que la entrada sea M-PAM. Pero además, la PSD que se logra obtener tiene una forma muy parecida a la PSD de la señal M-PAM. En otras palabras, la modulación M-PAM es la que convierte bits en símbolos, pero no es apropiada para ser enviada por un USRP, ya que no es estrictamente una Envolvente Compleja de algo.
- Todo lo anterior se explica mejor con ejemplos que se ilustran en las siguientes secciones.

### 7.3. La modulación BPSK

Esta modulación se distingue porque:

- La clave es de tipo: Binario, de modo que el código o clave que puede modificar los parámetros de la portadora puede tomar valores de 1 o 0.
- El parámetro de la portadora que es modificado es: La fase. Consecuentemente solo se tienen dos posibles estados para la fase. Usualmente la fase toma los valores 0 o  $\pi$ , pero también es BPSK si toma los valores  $-\pi/2$  y  $\pi/2$  o cualquier otros dos valores separados entre sí en un ángulo  $\pi$ .

- Para generar la Envolvente Compleja lo usual es considerar que la fase pueda tomar los valores 0 o  $\pi$ , por lo tanto, la Envolvente Compleja está dada por:

$$y_{CE}(t) = y_I(t) + jy_Q(t) \quad (7.6)$$

donde:

$$\begin{cases} y_I(t) = Ae^{j0}, \text{ para entrada 0} \\ y_Q(t) = Ae^{j\pi}, \text{ para entrada 1} \end{cases} \quad (7.7)$$

y donde A es la magnitud de los símbolos que son entregados al USRP. Lo anterior equivale a decir que:

$$\begin{cases} y_I(t) = A, \text{ para entrada 0} \\ y_Q(t) = -A, \text{ para entrada 1} \end{cases} \quad (7.8)$$

- La Tabla de Verdad 7.2 resume lo anterior así:

Tabla 7.2: Tabla de Verdad de la Modulación BPSK

| TABLA DE VERDAD BPSK |       |
|----------------------|-------|
| Clave                | Fase  |
| 0                    | 0     |
| 1                    | $\pi$ |

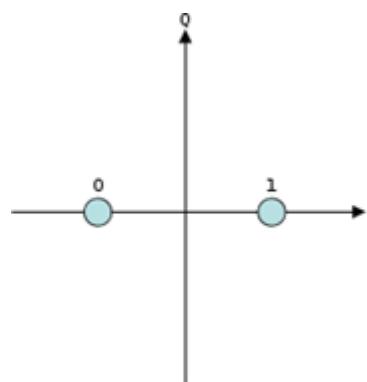
Otra manera de decir lo mismo es la siguiente.

Tabla 7.3: Tabla de Verdad de la Modulación BPSK. versión 2

| TABLA DE VERDAD BPSK |                |
|----------------------|----------------|
| Clave                | Valor complejo |
| 0                    | $Ae^{j0}$      |
| 1                    | $Ae^{j\pi}$    |

Otra manera de expresar lo mismo que la Tabla de verdad es usando un Diagrama de Constelaciones en el cual se muestra, de manera gráfica, en un diagrama polar, lo mismo que se quiere decir con la Tabla de Verdad. Para el caso de la Modulación BPSK el diagrama de constelaciones es el de la figura 2.

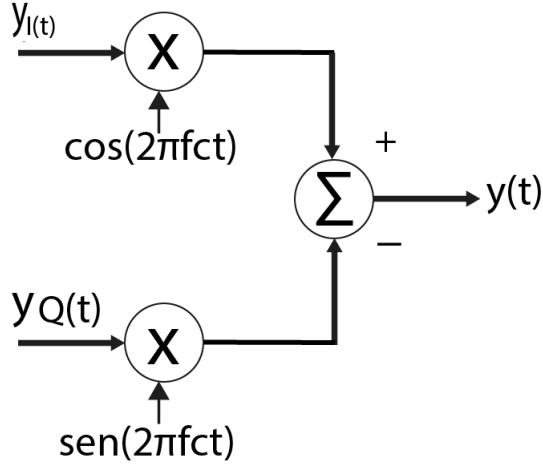
Figura 2: Diagrama de constelaciones BPSK.



Vale la pena recordar que si se tiene la constante pasobandas de señal modulada puede obtenerse apartir de su EC vale la pena recordar que se puedeLa señal pasa bandas se obtiene con el paso de esa señal por el Up Converter que se puede modelar con el esquema de la figura 3.

Hay que mejorar esta parte porque está saltando al uso del Upconverter sin haber siquiera mostrado la envolvente compleja, solo tiene como punto de partida la constelación. También es importante cambiar estilo, para que el lector sienta que hay un problema a resolver - se conoce la constelación, la EC, pero qué hay de la señal pasobandas? Luego entonces nos dedicamos a resolver este interrogante

Figura 3: Esquema señal paso banda.



La calidad de la señal enviada, respecto al ruido del canal está en función de la amplitud D de la señal s(t) que entrega el USRP. Varios elementos del USRP, como filtros y amplificadores, pueden alterar la amplitud:

$$s(t) = D \cos[2\pi f_c t + \varphi(t)] \quad (7.9)$$

donde

$$\varphi(t) = \begin{cases} 0, & \text{para entrada 0} \\ \pi, & \text{para entrada 1} \end{cases} \quad (7.10)$$

y donde  $D = AK_a$  y  $K_a$  representa la amplificación total que aplica el USRP .

Para llegar a conocer el desempeño de esta modulación ante el ruido, es necesario formular esta expresión en términos de la energía  $E_s$  que lleva cada símbolo, la cual para el caso de la BPSK es lo mismo que la energía  $E_b$  que lleva cada bit, pues cada símbolo contiene solo un bit en el caso de la BPSK. Esa energía es igual a la potencia promedio de la señal multiplicada por la duración del bit.

$$E_b = PT_b \quad (7.11)$$

A su vez, sabemos que:

$$V_{RMS} = \frac{D\sqrt{2}}{2} \quad (7.12)$$

Por lo tanto, la energía contenida en cada bit, en la señal modulada paso bandas, es en promedio:

$$E_b = \frac{D^2 T_b}{2} \quad (7.13)$$

De aquí, que.

$$D = \sqrt{\frac{2E_b}{T_b}} \quad (7.14)$$

Para obtener finalmente que:

$$s(t) = \sqrt{\frac{2E_b}{T_b}} \cos[2\pi f_c t + \varphi(t)] \quad (7.15)$$

donde

$$\varphi(t) = \begin{cases} 0, & \text{para entrada 0} \\ \pi, & \text{para entrada 1} \end{cases}$$

revisar bien esto, porque si  $D = AK_a$  abajo resulta que  $V_{RMS} = AK_a \sqrt{2}/2$ . El problema es que en el dominio de la constelación vemos que A es la magnitud y la magnitud debe coincidir con el valor RMS, os  $V_{RMS} = A$ . Puedo decir otras palabras la pregunta es el valor RMS de la EC debería ser el mismo de la señal pasobandas ya que la potencia promedio de la misma en las señales pasobandas que en la EC, cuales son las relaciones correctas?

De modo que cuando la señal pasobandas está antes de la antena transmisora del USRP se puede expresar como:

$$s(t) = K_a A \cos[2\pi f_c t + \varphi(t)] \quad (7.16)$$

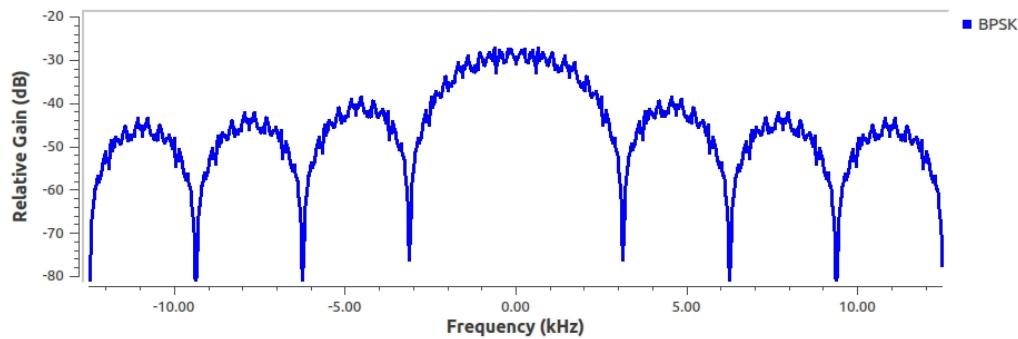
donde

$$\varphi(t) = \begin{cases} 0, & \text{para entrada 0} \\ \pi, & \text{para entrada 1} \end{cases}$$

Pero, más allá de la antena, el parámetro  $E_b$  se desvanece por el efecto de la propagación y aparece una componente adicional, el ruido.

En la figura 4 se presenta un ejemplo de la PSD de una señal BPSK para el caso específico en que la tasa de bits es  $R_b = 1,5625 \text{ kbps}$

Figura 4: PSD de una señal BPSK, con  $R_s=1,5625 \text{ kBaud}$ ,  $R_b=1,5625 \text{ Kbps}$ .



## 7.4. Diferentes tipos de modulación basadas en constelaciones

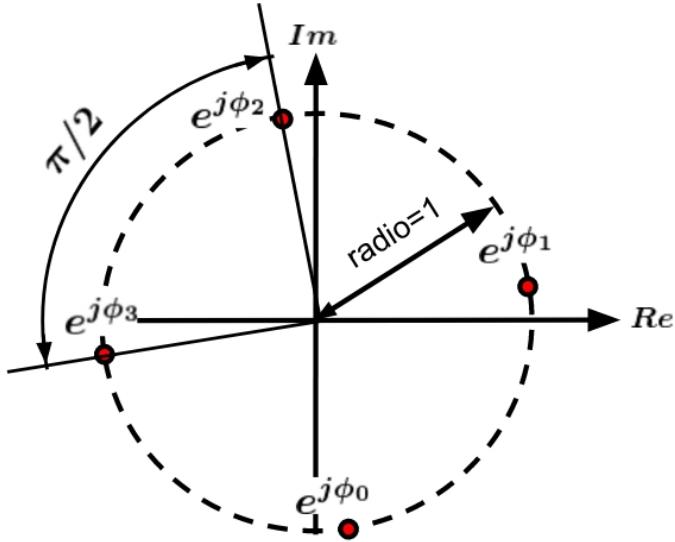
La modulación BPSK está basada en una constelación como la mostrada en la figura 2, pero bajo el mismo principio funcionan los tipos de modulación digital más conocidos como la Modulación QPSK (del inglés *Quadrature Phase Shift Keying*), la M-PSK (del inglés *M Phase Shift Keying*), la M-QAM (del inglés *M-Quadrature Amplitud Modulation*). Todas ellas funcionan bajo un mismo principio, como el explicado para BPSK. También vamos a comparar esas modulaciones con la M-PAM que se vió anteriormente por dos razones: La implementación en GNU Radio a menudo pasa por obtener primero la M-PAM; la PSD de una señal con una de las modulaciones a estudiar, es muy similar a la de la respectiva M-PAM.

### 7.4.1. La Modulación QPSK

Con este tipo de modulación iniciamos el estudio de varios métodos de modulación que incluyen  $M$  puntos de constelación.

En el caso de la Modulación QPSK o Modulación por desplazamiento de fase, (del inglés *Quadrature Phase Shift Keying*), se tienen  $M = 4$  puntos de constelación que están sobre un círculo y espaciados entre sí en un ángulo de  $\pi/2$ , como se muestra en la figura 5.

Figura 5: Constelación para la Modulación QPSK.



#### 7.4.1.1. El ordenamiento de la constelación

Para cualquiera de los métodos de modulación con  $M$  puntos de constelación se pueden realizar las siguientes observaciones para organizar la constelación y consecuentemente la Tabla de Verdad que la define:

- Cada punto de constelación puede ser activado mediante una clave, que es una combinación de  $bps$  bits, donde
$$bps = \log_2(M) \quad (7.17)$$
- Surge un importante reto: Lograr organizar los puntos de constelación de manera que no se eleve innecesariamente la probabilidad de pérdida de bits, que es lo mismo que la Rata de Errores de Bits (BER).
- Veamos un ejemplo de lo que pudiera ocurrir: En la parte receptora los puntos de constelación se ven influenciados por ruido, de manera que en ciertos casos, un punto de constelación puede llegar invadiendo el espacio de la constelación que corresponde a un punto vecino. Eso hace que el regenerador de símbolos tome una decisión errónea, con lo cual se habrá presentado un símbolo erróneo, que pueden significar hasta  $bps$  bits erróneos, ya que cada símbolo está compuesto de  $bps$  bits.
- Lo más natural es definir la constelación comenzando con la fase  $\phi = 0$ . En caso de requerirse una rotación de la constelación, se introduciría un factor tipo  $e^{j\phi_d}$ .

La manera de solucionar el problema de un innecesario aumento de la Probabilidad de Error de bits, consiste en lograr que los puntos de constelación resulten ordenados, de manera tal, que no tengan más de un bit diferente a los puntos adyacentes. De manera que si se presenta un error en un símbolo debido a que invade un símbolo vecino, solo se presentará un bit erróneo, pues de otra suerte se presentaría un símbolo erróneo es decir  $bps$  bits erróneos. La importancia de contar un menor número de bits erróneos se entiende si tenemos en cuenta que existen métodos para corregir errores, pero tienen un umbral de permisividad, es decir, funcionan bien hasta cierta probabilidad de pérdida de bits

Figura 6: Tablas de conversión de Binario a Gray.

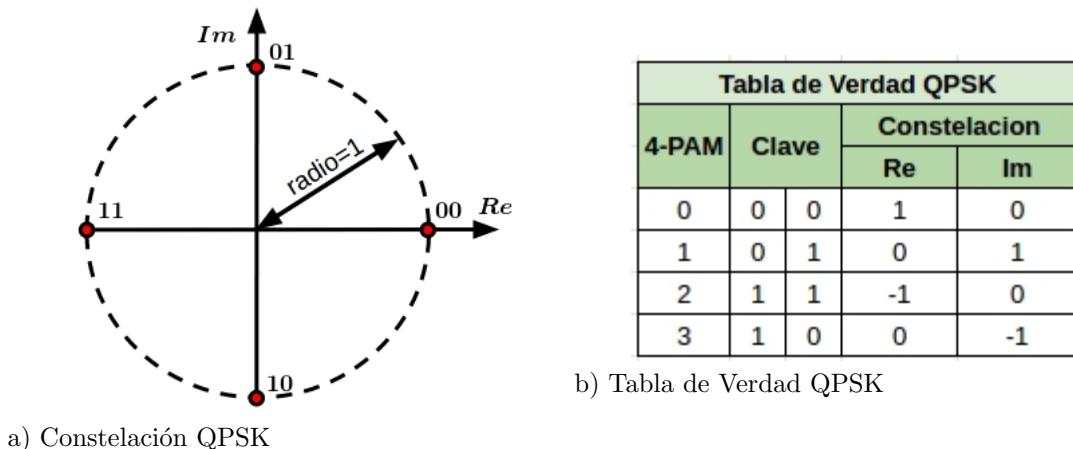
| Tabla Binario a Gray, M=4 |         |       | Tabla Binario a Gray, M=8 |         |         |       |           |           |       | Tabla Binario a Gray, M=16 |             |       |               |               |       |                 |                 |
|---------------------------|---------|-------|---------------------------|---------|---------|-------|-----------|-----------|-------|----------------------------|-------------|-------|---------------|---------------|-------|-----------------|-----------------|
| 4-PAM                     | Binario | Gray  | 4-PAM                     | Binario | Gray    | 4-PAM | Binario   | Gray      | 4-PAM | Binario                    | Gray        | 4-PAM | Binario       | Gray          | 4-PAM | Binario         | Gray            |
| 0                         | 0 0 0   | 0 0 0 | 0                         | 0 0 0 0 | 0 0 0 0 | 0     | 0 0 0 0 0 | 0 0 0 0 0 | 0     | 0 0 0 0 0 0                | 0 0 0 0 0 0 | 0     | 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 | 0     | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 |
| 1                         | 0 0 1   | 0 1 1 | 1                         | 0 0 1 0 | 0 0 1 0 | 1     | 0 0 1 0 0 | 0 0 1 0 0 | 1     | 0 0 1 0 0 0                | 0 0 1 0 0 0 | 1     | 0 0 1 0 0 0 0 | 0 0 1 0 0 0 0 | 1     | 0 0 1 0 0 0 0 0 | 0 0 1 0 0 0 0 0 |
| 2                         | 1 0 0   | 1 1 1 | 2                         | 0 1 0 0 | 0 1 0 0 | 1     | 0 1 0 0 1 | 0 1 0 0 1 | 1     | 0 1 0 0 1 0                | 0 1 0 0 1 0 | 1     | 0 1 0 0 1 0 0 | 0 1 0 0 1 0 0 | 1     | 0 1 0 0 1 0 0 0 | 0 1 0 0 1 0 0 0 |
| 3                         | 1 1 1   | 1 0 0 | 3                         | 0 1 1 0 | 0 1 1 0 | 0     | 0 1 1 0 0 | 0 1 1 0 0 | 0     | 0 1 1 0 0 0                | 0 1 1 0 0 0 | 0     | 0 1 1 0 0 0 0 | 0 1 1 0 0 0 0 | 0     | 0 1 1 0 0 0 0 0 | 0 1 1 0 0 0 0 0 |
|                           |         |       | 4                         | 1 0 0   | 1 1 0   | 4     | 0 1 0 0 1 | 0 1 0 0 1 | 0     | 0 1 0 0 1 0                | 0 1 0 0 1 0 | 0     | 0 1 0 0 1 0 0 | 0 1 0 0 1 0 0 | 0     | 0 1 0 0 1 0 0 0 | 0 1 0 0 1 0 0 0 |
|                           |         |       | 5                         | 1 0 1   | 1 1 1   | 5     | 0 1 0 1 0 | 0 1 0 1 0 | 1     | 0 1 0 1 0 0                | 0 1 0 1 0 0 | 1     | 0 1 0 1 0 0 0 | 0 1 0 1 0 0 0 | 1     | 0 1 0 1 0 0 0 0 | 0 1 0 1 0 0 0 0 |
|                           |         |       | 6                         | 1 1 0   | 0 1 0   | 6     | 1 1 0 1 0 | 1 0 1 0   | 1     | 1 1 0 1 0 0                | 1 0 1 0 0   | 1     | 1 1 0 1 0 0 0 | 1 0 1 0 0 0   | 1     | 1 1 0 1 0 0 0 0 | 1 0 1 0 0 0 0   |
|                           |         |       | 7                         | 1 1 1   | 1 1 0   | 7     | 1 1 1 1 0 | 1 1 0 0   | 0     | 1 1 1 1 0 0                | 1 1 0 0 0   | 0     | 1 1 1 1 0 0 0 | 1 1 0 0 0 0   | 0     | 1 1 1 1 0 0 0 0 | 1 1 0 0 0 0 0   |
|                           |         |       |                           |         |         | 8     | 1 0 0     | 0 0 1     | 1     | 1 0 0 0 1                  | 1 1 0 0 1   | 0     | 1 0 0 0 1 0   | 1 1 0 0 1 0   | 0     | 1 0 0 0 1 0 0   | 1 1 0 0 1 0 0   |
|                           |         |       |                           |         |         | 9     | 1 0 1     | 0 0 1     | 1     | 1 0 1 0 1                  | 1 1 0 1 1   | 0     | 1 0 1 0 1 0   | 1 1 0 1 1 0   | 1     | 1 0 1 0 1 0 0   | 1 1 0 1 1 0 0   |
|                           |         |       |                           |         |         | 10    | 1 0 0     | 1 0 1     | 0     | 1 0 0 1 0                  | 1 1 1 0 1   | 1     | 1 0 0 1 0 1   | 1 1 1 0 1 1   | 1     | 1 0 0 1 0 1 0   | 1 1 1 0 1 1 0   |
|                           |         |       |                           |         |         | 11    | 1 0 1     | 0 1 1     | 1     | 1 0 1 1 1                  | 1 1 1 1 1   | 1     | 1 0 1 1 1 1   | 1 1 1 1 1 1   | 0     | 1 0 1 1 1 1 0   | 1 1 1 1 1 1 0   |
|                           |         |       |                           |         |         | 12    | 1 1 0     | 0 0 1     | 0     | 1 1 0 0 1                  | 1 0 1 1 1   | 0     | 1 1 0 0 1 0   | 1 0 1 1 1 0   | 1     | 1 1 0 0 1 0 0   | 1 0 1 1 1 0 0   |
|                           |         |       |                           |         |         | 13    | 1 1 1     | 0 1 1     | 1     | 1 1 1 0 1                  | 1 0 1 1 1   | 0     | 1 1 1 0 1 0   | 1 0 1 1 1 0   | 1     | 1 1 1 0 1 0 0   | 1 0 1 1 1 0 0   |
|                           |         |       |                           |         |         | 14    | 1 1 0     | 1 0 1     | 0     | 1 1 0 1 0                  | 1 0 1 1 0   | 1     | 1 1 0 1 0 1   | 1 0 1 1 0 1   | 0     | 1 1 0 1 0 1 0   | 1 0 1 1 0 1 0   |
|                           |         |       |                           |         |         | 15    | 1 1 1     | 1 1 1     | 1     | 1 1 1 1 1                  | 1 0 1 1 0   | 0     | 1 1 1 1 1 0   | 1 0 1 1 0 0   | 0     | 1 1 1 1 1 0 0   | 1 0 1 1 0 0 0   |

Es allí donde toman importancia los códigos de Gray. En la figura 6 se presentan tres casos para los códigos de Gray: Cuando  $M = 4$ ,  $M = 8$ ,  $M = 16$ . Como podemos observar en todos los casos, cada código de Gray se diferencia de un vecino solo en un bit. También hemos conservado la relación con la modulación M-PAM por dos motivos: Los valores de la M-PAM sirven como identificador de cada punto de constelación, pero el principal motivo es porque los bloques que ofrece GNU Radio para los diferentes métodos de modulación digital piden como entrada no una señal binaria sino una M-PAM.

Ahora podemos obtener fácilmente la constelación para la Modulación QPSK y consecuentemente la Tabla de Verdad, como se muestra en la figura 7.

Debemos corregir todas las tablas de verdad de este capítulo para que muestren el proceso tal como se hace en GNURadio. La clave es que en GNURadio los símbolos PAM no son mapeados al código de Gray. Es la constelación lo que se mapea. Entonces en MPAM el 2 equivale a 10 no a 11. Pero nos olvidamos de la columna 4-PAM todo es correcto

Figura 7: Constelación y Tabla de Verdad para la Modulación QPSK.



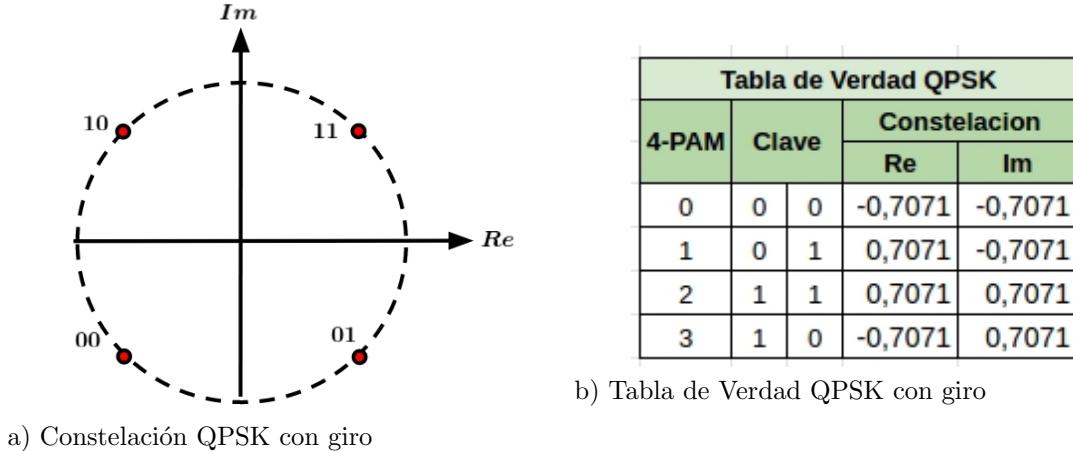
Podemos destacar en la Modulación QPSK los siguientes aspectos:

- En calidad de clave, los símbolos están compuestos de a dos bits, que es lo que se conoce como dubits.
- Al estar los puntos de constelación separados en  $\pi/2$ , la modulación QPSK puede ser vista como dos moduladores BPSK en paralelo: Un modulador BPSK que hace uso de la parte real de la Envoltura Compleja y otro que usa la parte imaginaria.
- Lo anterior nos induce a pensar que la probabilidad de pérdida de bits en la Modulación QPSK es similar a la BPSK, pero la QPSK tiene mayor eficiencia espectral ya que sobre un mínimo ancho de banda puede transportar dubits en vez de bits.

En la figura 8 podemos ver otra opción para la constelación y la Tabla de Verdad de la Modulación QPSK, la cual es la que usaremos con GNU Radio.

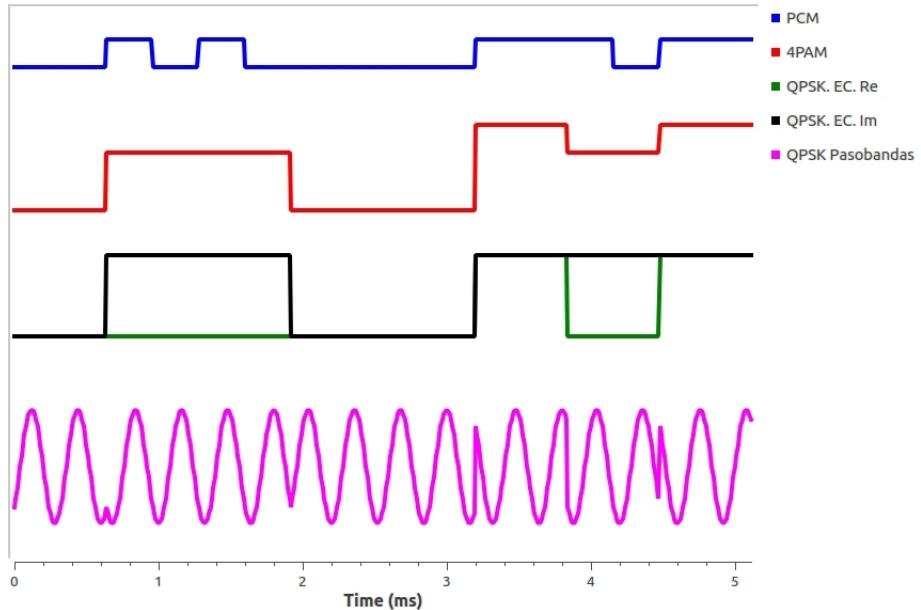
Debemos corregir todas las tablas de verdad de este capítulo para como se explicó en la Fig. 7

Figura 8: Constelación y Tabla de Verdad para la Modulación QPSK con giro.



Para esta segunda opción, en la figura 9, presentamos un ejemplo donde tenemos una señal binaria que es la señal modulante, la correspondiente señal M-PAM. Allí podemos ver también la Envolvente Compleja que se espera a la salida del modulador QPSK bandabase y en la parte inferior vemos la correspondiente señal pasobandas que se espera en la antena. Estas gráficas se han realizado para un caso ideal, en que las señales bandabase pudieran ser idealmente rectangulares y los saltos de fase pudieran ser realmente bruscos. Los casos reales se discuten más adelante, cuando hablemos de la implementación en GNU Radio. Podemos ver que a diferencia de la modulación BPSK la parte imaginaria no es cero sino una señal bipolar como lo es también la parte real:

Figura 9: Ejemplo de la Modulación QPSK en el dominio del tiempo.

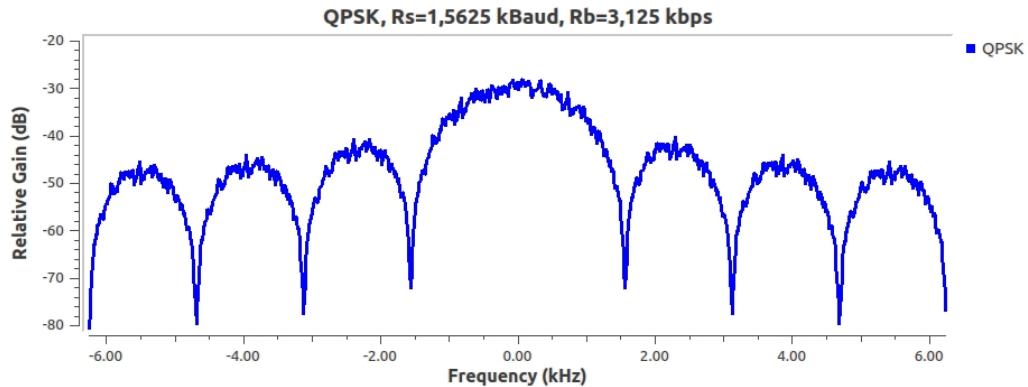


Flujograma usado: *graficas\_mod\_qpsk.grc*

Para comparar esta modulación con la BPSK cuando se aplican sobre un mismo ancho de banda, en la figura 10 presentamos la PSD de la Modulación QPSK en las mismas condiciones usadas para obtener

la PSD presentada en la figura 4 para una señal BPSK.

Figura 10: PSD para una señal QPSK.



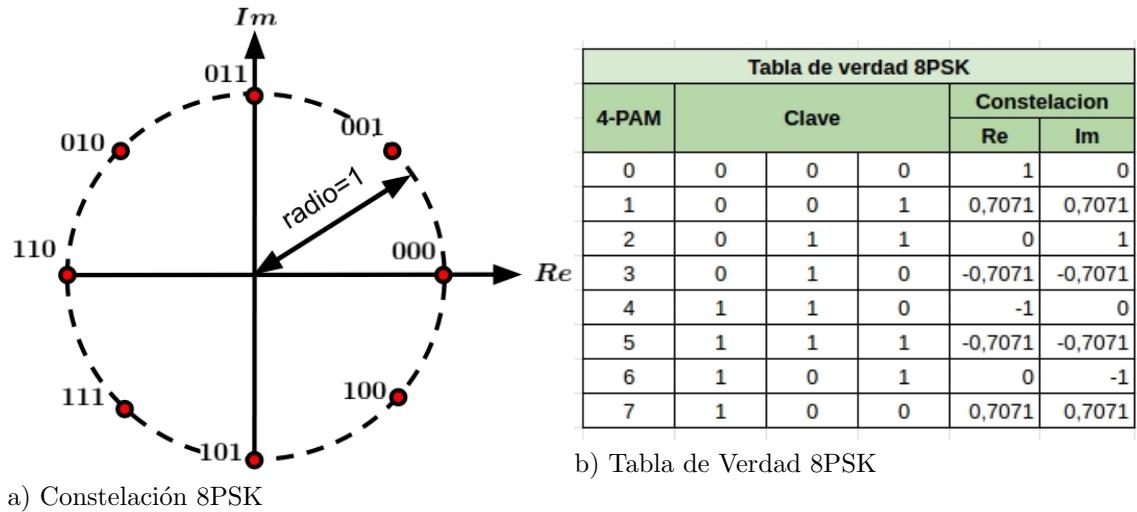
Flujograma usado: *graficas\_mod\_qpsk.grc*

Tenemos que: La forma de la PSD es la misma de la Modulación BPSK y la única diferencia está en que la QPSK lleva una tasa de bits que es el doble de la que lleva la BPSK. Esto significa que la modulación QPSK tiene una eficiencia espectral que es dos veces mayor que la modulación BPSK.

#### 7.4.2. La Modulación M-PSK y sus características distintivas

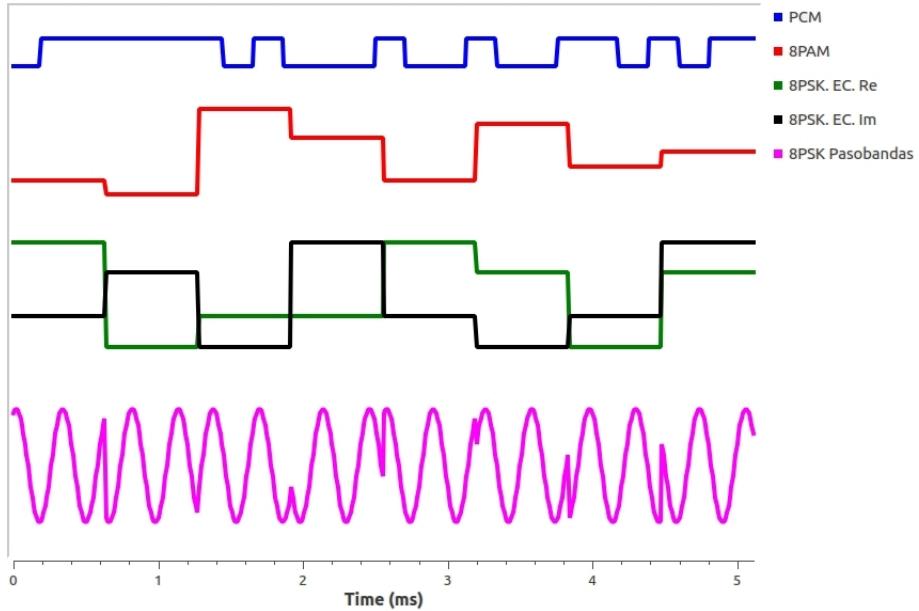
La Modulación M-PSK es similar a la QPSK, pero el hecho que  $M$  sea mayor a 4 marca fuertes diferencias, sobre todo que se aleja sensiblemente de la modulación BPSK. Analizaremos de manera específica el caso en que  $M = 8$ , es decir el caso de la Modulación 8-PSK, cuyo diagrama de constelaciones se presenta en la figura 11. Como puede verse, los puntos de constelación se han organizado en función de los códigos de Gray como se explicó en el capítulo anterior.

Figura 11: Constelación y Tabla de Verdad para la Modulación 8PSK.



Para observar las diferencias con respecto a las modulaciones vistas nos apoyaremos también en la figura 12.

Figura 12: Modulación 8PSK. Ejemplo en el dominio del tiempo.

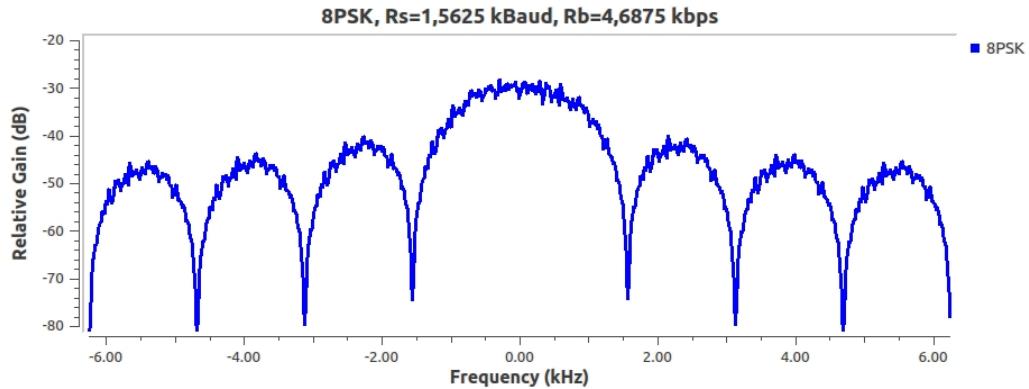


Flujograma usado: *graficas\_mod\_8psk.grc*

Podemos observar que:

- $M = 8$ , de manera que la constelación tiene 8 puntos o posibles diferentes símbolos.
- Consecuentemente el número de bits por símbolo es  $bps = 3$ .
- La modulación 8PSK no incluye cambios de amplitud, solo de fase, consecuentemente, la constelación incluye magnitud constante.
- Comparando esta modulación con las demás cuando se aplican sobre un mismo ancho de banda, tenemos que:

Figura 13: PSD de para una señal 8PSK.



Flujograma usado: *graficas\_mod\_8psk.grc*

- La forma de la PSD es la misma de la Modulación QPSK y de la BPSK y se presenta en la figura 13 donde la única diferencia que se aprecia en el aumento en la tasa de bits.
- La tasa de bits es igual a  $R_b = bps * R_s$ , o sea  $R_b = 3 * R_s$ .
- Lo anterior significa que la Modulación 8PSK tiene una eficiencia espectral tres veces mayor que la modulación B-PSK y 3/2 mayor que la QPSK. Pero también es previsible que esta modulación sea más sensible al ruido.

### 7.4.3. La Modulación M-QAM

Este es el caso más complejo de las modulaciones a tratar, pero en el fondo es el caso más genérico, del cual se pueden desprender todas las demás. Comenzamos por la Tabla de Verdad 7.4.

Tabla 7.4: Tabla de Verdad 16QAM.

| Tabla de Verdad Genérica de 16QAM |       |                         |
|-----------------------------------|-------|-------------------------|
| 16-PAM                            | Clave | Simbolo Constelacion    |
| 0                                 | 0000  | $A_0 e^{j\phi_0}$       |
| 1                                 | 0001  | $A_1 e^{j\phi_1}$       |
| 2                                 | 0011  | $A_2 e^{j\phi_2}$       |
| 3                                 | 0010  | $A_3 e^{j\phi_3}$       |
| 4                                 | 0110  | $A_4 e^{j\phi_4}$       |
| 5                                 | 0111  | $A_5 e^{j\phi_5}$       |
| 6                                 | 0101  | $A_6 e^{j\phi_6}$       |
| 7                                 | 0100  | $A_7 e^{j\phi_7}$       |
| 8                                 | 1100  | $A_8 e^{j\phi_8}$       |
| 9                                 | 1101  | $A_9 e^{j\phi_9}$       |
| 10                                | 1111  | $A_{10} e^{j\phi_{10}}$ |
| 11                                | 1110  | $A_{11} e^{j\phi_{11}}$ |
| 12                                | 1010  | $A_{12} e^{j\phi_{12}}$ |
| 13                                | 1011  | $A_{13} e^{j\phi_{13}}$ |
| 14                                | 1001  | $A_{14} e^{j\phi_{14}}$ |
| 15                                | 1000  | $A_{15} e^{j\phi_{15}}$ |

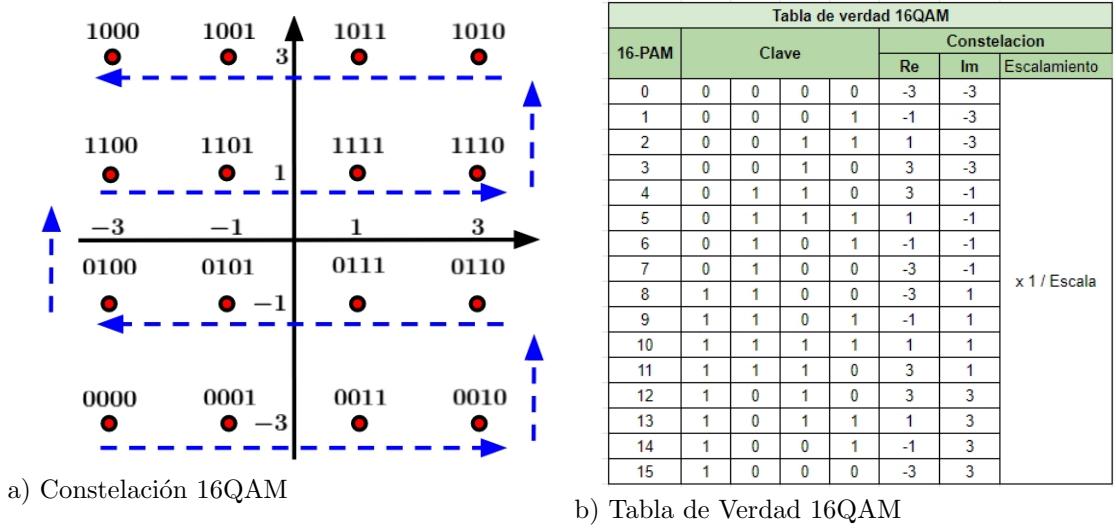
De esta tabla podemos ver algunos detalles:

- Los bits a enviar deberán previamente ser organizados en paquetes de  $bps = \log_2(M)$  bits, que para nuestro caso serán paquetes de cuatri-bits en código Gray.
- Vemos que puede haber una relación con la tabla de verdad de la modulación M-PAM.
- Vemos que la información puede modular la magnitud y la fase de la Envolvente Compleja, ya que para cada clave  $i$  le corresponde un valor  $A_i$  y  $\phi_i$ .

Encontrar los valores de  $A_i$  y  $\phi_i$  en la modulación M-QAM no es tan sencillo como en la modulación M-PAM. Pues van a aparecer más de un símbolo vecino ubicados a la misma distancia de cada punto de constelación. Aún así es necesario garantizar, como ya se dijo para todas las modulaciones digitales basadas en constelaciones, que los símbolos vecinos no se diferencien del nuestro en más de un bit. Para lograrlo, la constelación se debe armar como se ha hecho en la figura 14 donde el código de Gray se va ubicando en el orden que señalan las flechas azules. De esta manera, si tomamos por ejemplo el punto  $-1, 1$ , allí está ubicado el código 1101 y sus vecinos son 1111, 1100, 1001 y 0101 y vemos que todos los códigos vecinos se diferencian de nuestro código solo en un bit. En las coordenadas de la constelación se han usado los valores enteros  $-3, 1, 1$  y  $3$  para facilitar la ubicación de los puntos en la constelación y para que la constelación y la Tabla de Verdad parezcan más sencillas, pero se pueden dividir por un valor de escala, como se muestra en la columna escalamiento para lograr que la Envolvente Compleja que se genere tenga una potencia promedio deseada, por ejemplo, en el caso de los equipos USRP, es conveniente que los valores que toma la Envolvente Compleja estén por debajo del valor 1, pero si queremos comparar esta modulación con otras, es importante que todas se generen con la misma potencia promedio y para eso sirve el factor de escalamiento que hemos introducido.

Hay que corregir la tabla de verdad de la figura de 16QAM porque lo que está como columna  $I_m$  es lo real y lo que está como  $R_m$  es lo imaginario. Además aplica lo mismo por corregir dicho para la fig.7

Figura 14: Constelación y Tabla de Verdad para la Modulación 16QAM.

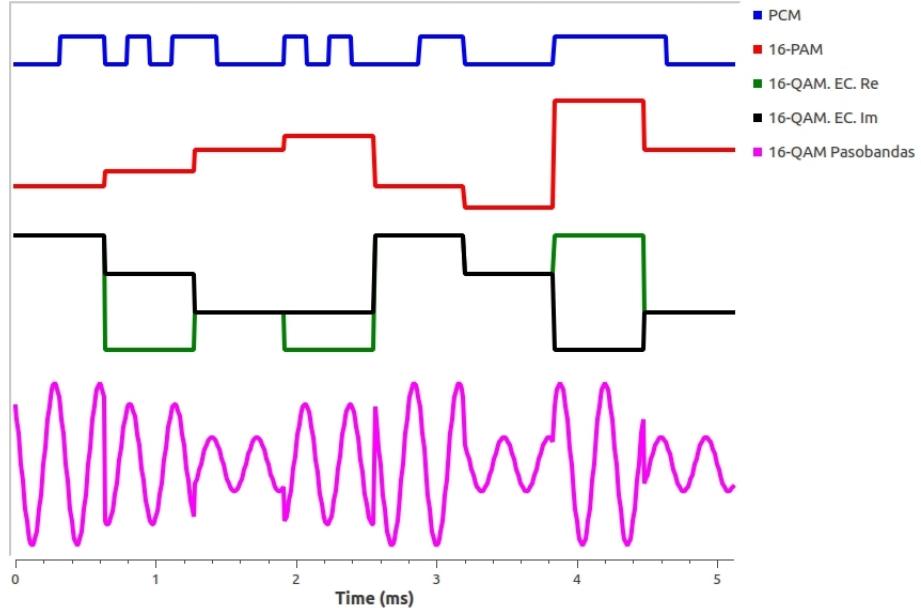


De esta constelación, podemos ver algunos detalles adicionales:

- Los valores  $A_i$  y  $\phi_i$  para cada símbolo de constelación  $i$  deben ser tomados de manera que los puntos de constelación resulten lo más separados posible y homogéneamente distribuidos en el área polar.
- Las flechas en la constelación muestran el orden en que se debe ir escribiendo el código de Gray en la constelación.
- Por lo anterior es más común hablar de 16QAM, 64QAM, 256QAM y no en cambio de un 8QAM.

El siguiente aspecto importante está en el dominio del tiempo, para lo cual hemos incluido el ejemplo de la figura 15

Figura 15: Ejemplo de una señal 16QAM en el dominio del tiempo



Flujograma usado: graficas\_mod\_16qam.grc

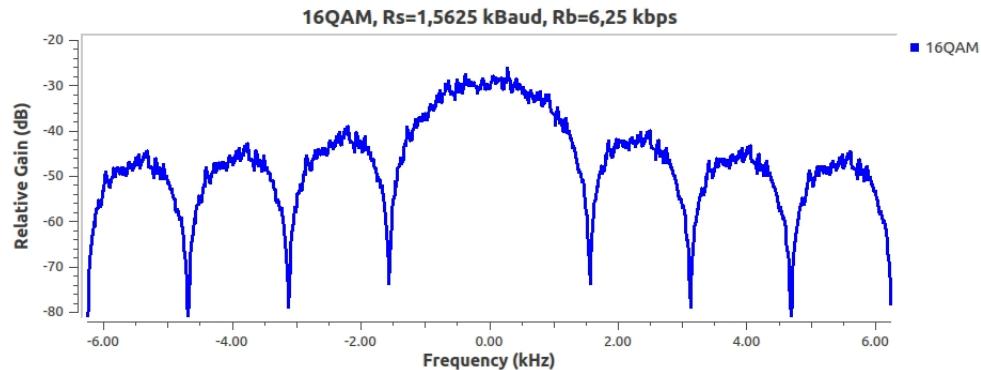
Podemos observar ahora que:

- Vemos que la señal pasobandas, que es la que realmente se emite, siendo una onda senoidal que cambia de amplitud y fase con cada nueva combinación de  $m = \log_2(M)$  bits a enviar.
- Igualmente, la Envolvente Compleja realiza un cambio con cada nueva combinación de  $m = \log_2(M)$  bits a enviar, pero lo que resultan son dos señales con símbolos rectangulares que corresponden a la parte real y a la imaginaria de la Envolvente Compleja.
- Por lo anterior es claro que la rata de símbolos está relacionada con la de bits así:

$$R_s = \frac{R_b}{\log_2(M)} \quad (7.18)$$

- Vemos que existe una cierta relación de la Envolvente Compleja con la señal con modulación M-PAM. Finalmente, es importante observar lo que ocurre en el dominio de las frecuencias, para lo cual nos remitimos a la figura 16.

Figura 16: 16-QAM en el dominio de las frecuencias



Flujograma usado: *graficas\_mod\_16qam.grc*

Lo que podemos observar ahora es lo siguiente:

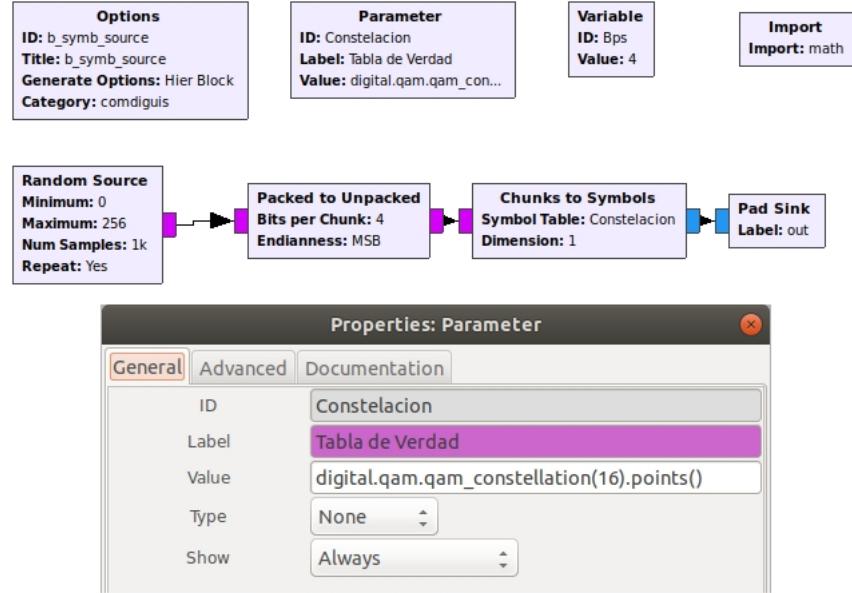
- La PSD tiene una forma similar a la de las señales binarias aleatorias bipolares, pero, al igual que los demás tipos de modulación vistos, los pasos por cero son en  $kR_s$ , donde  $k$  es un número entero y  $R_s$  es la rata de símbolos, que se calcula con la Ecuación 7.18.
- Es importante destacar que la única diferencia que se observa en la PSD con respecto a la PSD de otros tipos de modulación está en el valor de la rata de bits  $R_b$  que puede conducir esta modulación que es mayor a las demás modulaciones vistas, lo que significa que tiene la mayor eficiencia espectral.

#### 7.4.4. Implementación en GNU Radio

En la figura 17 se presenta un ejemplo de implementación en GNU Radio de la modulación 16QAM, pero sirve para cualquier otra modulación como se explica más adelante. Para esta implementación, se tienen las siguientes observaciones:

- La fuente de información debe ser de números enteros, como es el caso de las señales cuantificadas. Para este ejemplo hemos usado una fuente de amplitudes aleatorias que oscilan entre 0 y 255, pero puede oscilar entre cualquier otro valor.

Figura 17: Implementación en grc de GNU Radio de un modulador 16QAM.



Flujograma usado: *b\_symb\_source* Nota: es parte de la librería *comdiguis*

- En esta implementación no es necesario realizar el paso a señal PCM, simplemente bloque ‘‘Packed to Unpacked’’ lo realiza internamente, para luego producir una señal equivalente en información pero con amplitudes dadas en números enteros entre 0 y  $M - 1$ . En otras palabras lo que este bloque realiza es una modulación M-PAM.
- El bloque ‘‘Chunks to Symbols’’ aplica propiamente la modulación, que no es otra cosa que una traducción de los valores de amplitud recibidos a símbolos complejos de acuerdo a la tabla de verdad.
- La tabla de verdad se configura mediante el parámetro ‘‘Symbol Table’’ del bloque ‘‘Chunks to Symbols’’. Se trata de un vector que contiene todos los posibles símbolos de la constelación.
- Es posible asignarle al objeto nuestra propia constelación, ordenada de la manera consecutiva. A continuación se presentan varios ejemplos probados por consola pero que fácilmente se pueden llevar a un flujograma:

```
>>> import numpy
>>> # Caso QPSK
>>> M=4
>>> MPAM=numpy.linspace(0,M-1,M)
>>> Constelacion=numpy.exp(2.j*math.pi*MPAM/M)
>>> Constelacion
array([1.0000000e+00+0.0000000e+00j, 6.1232340e-17+1.0000000e+00j,
 -1.0000000e+00+1.2246468e-16j, -1.8369702e-16-1.0000000e+00j])

>>> # Caso 8PSK
>>> M=8
>>> MPAM=numpy.linspace(0,M-1,M)
>>> Constelacion=numpy.exp(2.j*math.pi*MPAM/M)
>>> Constelacion
array([1.0000000e+00+0.0000000e+00j, 7.07106781e-01+7.07106781e-01j,
 6.12323400e-17+1.0000000e+00j, -7.07106781e-01+7.07106781e-01j,
 -1.0000000e+00+1.22464680e-16j, -7.07106781e-01-7.07106781e-01j,
 -1.83697020e-16-1.0000000e+00j, 7.07106781e-01-7.07106781e-01j])
```

- Igualmente, con comandos de Python, fácilmente adaptables a GNU Radio, es posible obtener sus propios puntos de constelación ordenados con el código de Gray. A continuación se muestra un ejemplo para la constelación de la modulación 8PSK

```
>>> from gnuradio import digital
>>> import numpy
```

```

>>> import math

>>> M=8
>>> MPAM_Gray=numpy.array(digital.utils.gray_code.gray_code(M))
>>> MPAM_Gray
array([0, 1, 3, 2, 6, 7, 5, 4])
>>> Constelacion=numpy.exp(2.j*math.pi*MPAM_Gray/M)

>>> Constelacion
array([1.00000000e+00+0.00000000e+00j, 7.07106781e-01+7.07106781e-01j,
 -7.07106781e-01+7.07106781e-01j, 6.12323400e-17+1.00000000e+00j,
 -1.83697020e-16-1.00000000e+00j, 7.07106781e-01-7.07106781e-01j,
 -7.07106781e-01-7.07106781e-01j, -1.00000000e+00+1.22464680e-16j])

>>> # Si así lo prefieres puede girar la constelacion, por ejemplo en pi/8 rad
>>> Constelacion=Constelacion*numpy.exp(-1.j*math.pi/8)

>>> Constelacion
array([0.92387953-0.38268343j, 0.92387953+0.38268343j,
 -0.38268343+0.92387953j, 0.38268343+0.92387953j,
 -0.38268343-0.92387953j, 0.38268343-0.92387953j,
 -0.92387953-0.38268343j, -0.92387953+0.38268343j])

```

- GNU Radio también incluye varias funciones para calcular los puntos de constelación de diferentes tipos de modulación, pero ellos aparecen ordenados en una forma particular que puede tener ventajas para facilitar la prevención y corrección de errores, pero al mismo tiempo puede resultar confusa para la aplicación de técnicas adicionales de canal que requieren conocer el orden de los puntos de constelación como es el caso de la Codificación Diferencial que se trata más adelante. A continuación se muestran ejemplos de generación de puntos de constelación usando funciones que proporciona GNU Radio:

```

>>> from gnuradio import digital

>>> digital.constellation_bpsk().points()
((-1+0j), (1+0j))

>>> digital.constellation_qpsk().points()
((-0.7071070075035095-0.7071070075035095j),
 (0.7071070075035095-0.7071070075035095j),
 (-0.7071070075035095+0.7071070075035095j),
 (0.7071070075035095+0.7071070075035095j))

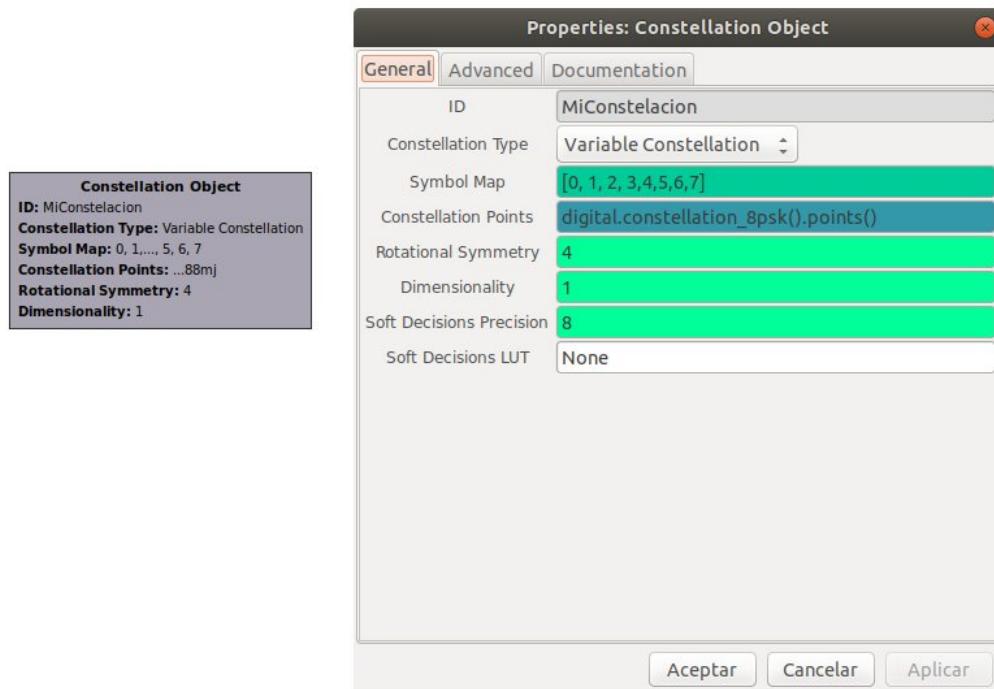
>>> digital.constellation_8psk().points()
((0.9238795042037964+0.3826834559440613j),
 (-0.9238796234130859+0.38268327713012695j),
 (0.9238795638084412-0.3826834261417389j),
 (-0.9238795042037964-0.3826834261417389j),
 (0.3826834261417389+0.9238795042037964j),
 (-0.38268351554870605+0.9238795042037964j),
 (0.3826836049556732-0.9238794445991516j),
 (-0.382683128118515-0.9238796830177307j))

>>> digital.constellation_16qam().points()
((0.3162277638912201-0.3162277638912201j),
 (-0.3162277638912201-0.3162277638912201j),
 (0.9486832618713379-0.9486832618713379j),
 (-0.9486832618713379-0.9486832618713379j),
 (-0.9486832618713379-0.3162277638912201j),
 (0.9486832618713379-0.3162277638912201j),
 (-0.3162277638912201-0.9486832618713379j),
 (0.3162277638912201-0.9486832618713379j),
 (-0.9486832618713379+0.9486832618713379j),
 (0.9486832618713379+0.9486832618713379j),
 (-0.3162277638912201+0.3162277638912201j),
 (0.3162277638912201+0.9486832618713379j),
 (-0.3162277638912201+0.9486832618713379j),
 (0.9486832618713379+0.3162277638912201j),
 (-0.9486832618713379+0.3162277638912201j))

```

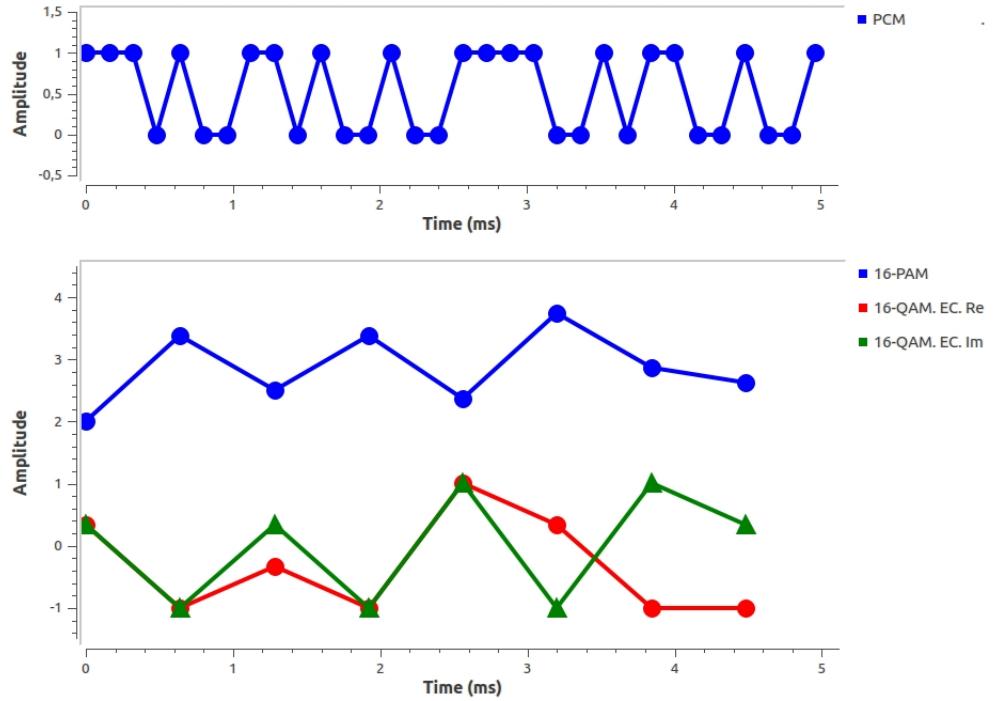
- GNU Radio incluye un bloque llamado “Constellation Object”, ver figura 18, que crea un objeto que requieren la mayoría de los bloques que guardan relación con las constelaciones. Ese objeto no es otra cosa que una especie de tabla que guarda todos los detalles de la constelación como: los puntos de constelación pero también otros detalles.
- varios bloques de GNU Radio relacionados con el procesamiento de las constelaciones reclaman la existencia del objeto de constelación mencionado anteriormente. Otros en cambio solo piden los puntos de constelación que también se pueden entregar por medio de ese objeto por ejemplo con un comando tipo: `MiConstelacion.points()`, donde “MiConstelacion” es el nombre dado al objeto.
- bloque “Constellation Object” tiene un parámetro algo confuso de entender llamado “Symbol Map”. Este parámetro y los demás que allí aparecen no son para nada de lo tratado hasta el momento. Está más bien relacionado con el uso de la codificación diferencial que se trata más adelante, pero no lo usaremos pues el mapeo a que hace referencia lo implementaremos de manera separada.

Figura 18: El bloque Constellation Object.



En la figura 19 se presenta un ejemplo del resultado de la Modulación 16 QAM implementada en GNU Radio.

Figura 19: Ejemplo de la modulación 16QAM implementada en GNU Radio.



Flujograma usado: *graficas\_mod\_16qam.grc*

De estos resultados podemos resaltar lo siguiente:

- A diferencia de los ejemplos mostrados hasta el momento, en GNU Radio se maneja, naturalmente, señales discretas, de manera que solo hay una muestra por bit, así como una muestra por símbolo.
- Es posible apreciar que los símbolos tienen una frecuencia de muestreo  $bps = \log_2(M)$  menor que los bits, lo cual es natural ya que cada símbolo representa a  $bps$  bits.

## 7.5. Curvas de BER

En una comunicación Digital, Errores de Bits o Bit Error es el número de bits en un volumen de datos transmitidos que llegan alterados al receptor debido a su paso por el canal, donde la señal es afectada por el ruido, interferencias y/o distorsiones. BER (*Bit Error Ratio*) es el número de bits con errores dividido entre el total de bits transmitidos en un intervalo de tiempo de interés. Es usualmente medido en forma de porcentaje. La probabilidad de error  $P_e$  o Probabilidad de Perdida de Bits es sinónimo de BER, pero es usualmente presentada como un porcentaje.

Veamos un ejemplo. Tenemos la siguiente secuencia a transmitir:

0110001011

Una vez transmitida, al receptor llega lo siguiente:

0010101001

En este ejemplo, podemos ver que se están presentando 3 Errores de bits entre 10 bits recibidos, luego se están presentando un Bit Error de 3. La BER se calcula como los 3 bits incorrectos divididos entre 10 bits transmitidos, lo cual resulta en una BER de 0.3 y equivale a una probabilidad de pérdida de bits del 30 %.

### 7.5.1. El problema a resolver

A continuación nos dedicaremos a descubrir cual es el problema que se busca resolver mediante las curvas de BER.

En este momento hemos estudiado varios tipos de modulación (BPSK, QPSK, M-PSK, M-QAM) que son comunes en las comunicaciones inalámbricas. Se ha comprobado que si el canal está limitado en una banda  $B$ , con cualquiera de esas modulaciones se conduce una misma tasa de símbolos  $R_s$  sobre esa banda fija  $B$ . De hecho, cualquiera de esas modulaciones, en esas condiciones, tiene la misma PSD. En la figura 16 se mostró la PSD de la modulación 16QAM bandabase. Ahora presentamos la PSD de la Modulación BPSK en la figura 20, la PSD de la Modulación QPSK en la figura 21 y la PSD de la Modulación 8PSK en la figura 22.

Figura 20: PSD de una señal BPSK.

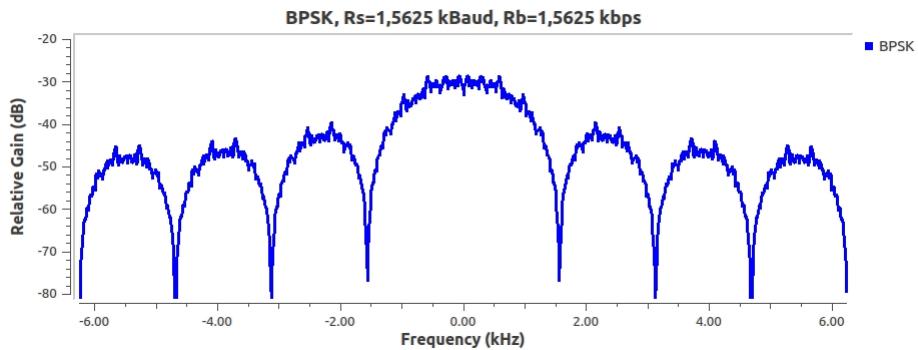


Figura 21: PSD de una señal QPSK.

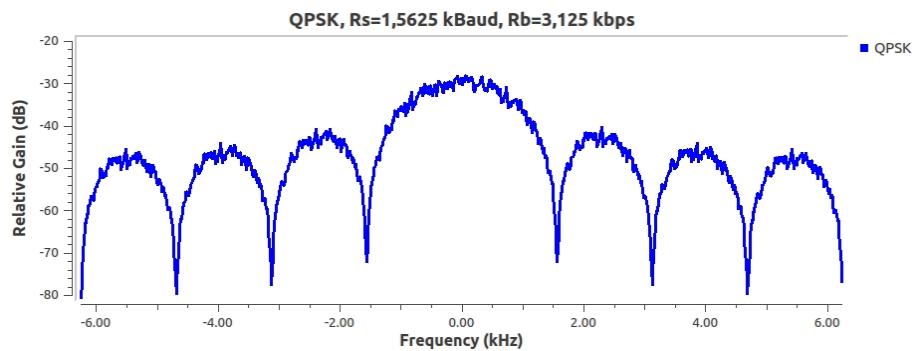
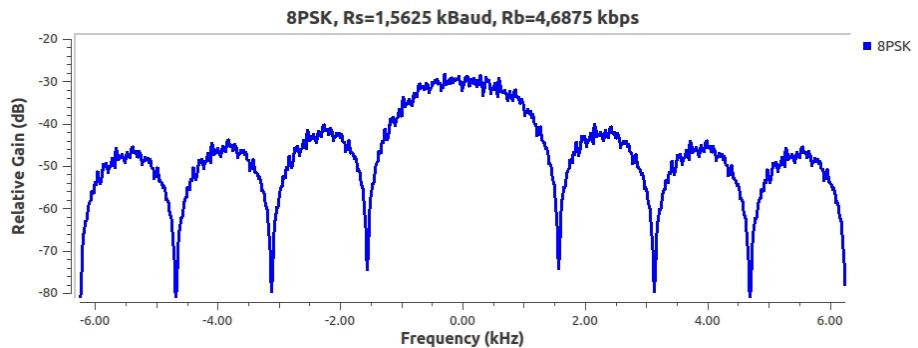


Figura 22: PSD de una señal 8PSK.

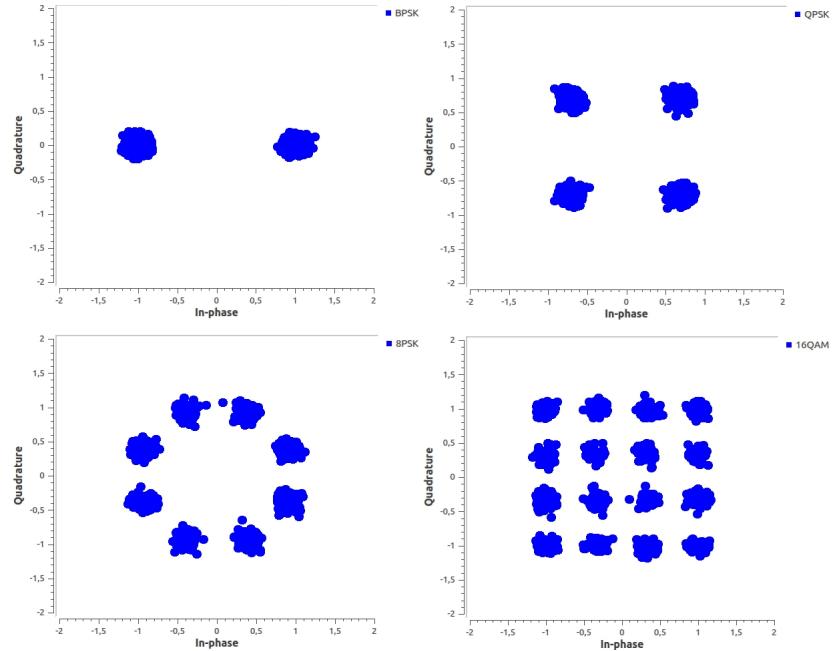


Vemos que para todas esas Modulaciones, la PSD es la misma y la diferencia entre esas modulaciones

está en la rata de bits, que es diferente para cada una de ellas, ya que  $R_b = R_s bps$ , donde  $bps = \log_2(M)$  y  $M$  es el número de puntos de constelación.

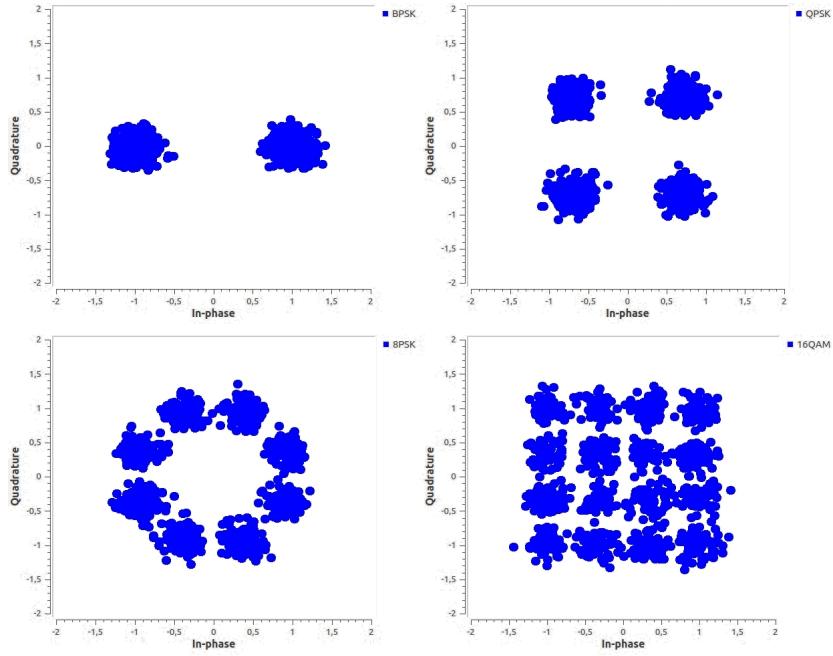
Pensaríamos entonces que lo recomendable es una Modulación con un valor infinitamente alto para  $M$ , ya que entre más alto sea  $M$  mayor es la eficiencia espectral de la Modulación dada. En realidad las cosas no son así tan sencillas, el problema para las modulaciones de mayor grado ( $M$  mayor) se ven más afectadas por el ruido. En la figura 23 se presenta un caso en que todas las constelaciones, a pesar del nivel de ruido conservan una forma que permite distinguir cada punto, de modo que para este caso la mejor modulación es la 16 QAM ya que además tiene mayor eficiencia espectral que las demás.

Figura 23: Comparación de BPSK, QPSK, 8PSK y 16QAM ante el ruido. Caso1



Sin embargo, en la figura 24 se tiene un segundo caso, en la que la relación señal a ruido es inferior, observamos cómo los símbolos en la modulación 16 QAM tienden a solaparse entre sí. Lo mismo ocurre con la 8PSK aunque en menor grado. Eso significa que se va a presentar una pérdida de símbolos muy alta y quizás inaceptable. Por lo tanto, en este caso, la mejor modulación es la QPSK.

Figura 24: Comparación de BPSK, QPSK, 8PSK y 16QAM ante el ruido. Caso2



En los sistemas de comunicaciones modernos, un sistema puede usar varios tipos de modulación, de manera que tiene la posibilidad de escoger la que más convenga de acuerdo a la relación señal a ruido que le brinde el canal. Entonces surge la pregunta: ¿Qué criterio de decisión puede servir para escoger una u otra modulación?

### 7.5.2. La solución al problema

La respuesta está en conocer previamente el desempeño de cada tipo de modulación ante un referente apropiado. Ese referente es el ruido blanco gaussiano aditivo. Es prácticamente imposible crear unas condiciones reales para probar un sistema de comunicaciones ante el ruido, ya que en un canal inalámbrico existen muchos otros fenómenos que el ruido blanco gaussiano aditivo. Por lo tanto, resulta importante hacer la experimentación mediante simulación, para lo cual es necesario implementar un Canal de Ruido Blanco Gaussiano Aditivo (AWGN, del inglés *Additive White Gaussian Noise*). La idea es poder medir la tasa de bits perdidos para cada tipo de modulación en condiciones similares para diferentes relaciones de señal a ruido.

En este caso, no es usual usar una relación señal a ruido como la que se usa en las comunicaciones analógicas sino más bien una relación  $E_b/N_o$ , donde  $E_b$  es la energía de cada bit, la cual se calcula como:

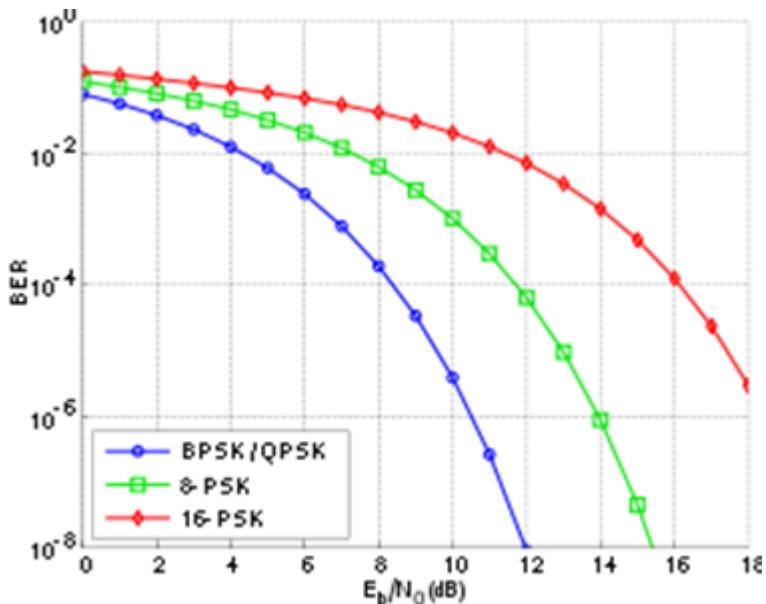
$$E_b = PT_b = \frac{P}{R_b} \quad (7.19)$$

donde  $P$  es la potencia promedio de la señal actual sin tener en cuenta ruido,  $T_b$  es la duración de cada bit y  $R_b$  es la tasa de bits. También es común usar la relación  $E_s/N_o$ , donde  $E_s$  es la energía de cada símbolo, la cual se calcula como:

$$E_s = PT_s = \frac{P}{R_s} \quad (7.20)$$

donde  $T_s$  es la duración de cada símbolo y  $R_s$  es la tasa de símbolos.

Figura 25: Curvas de BPSK, QPSK y 8PSK y 16PSK

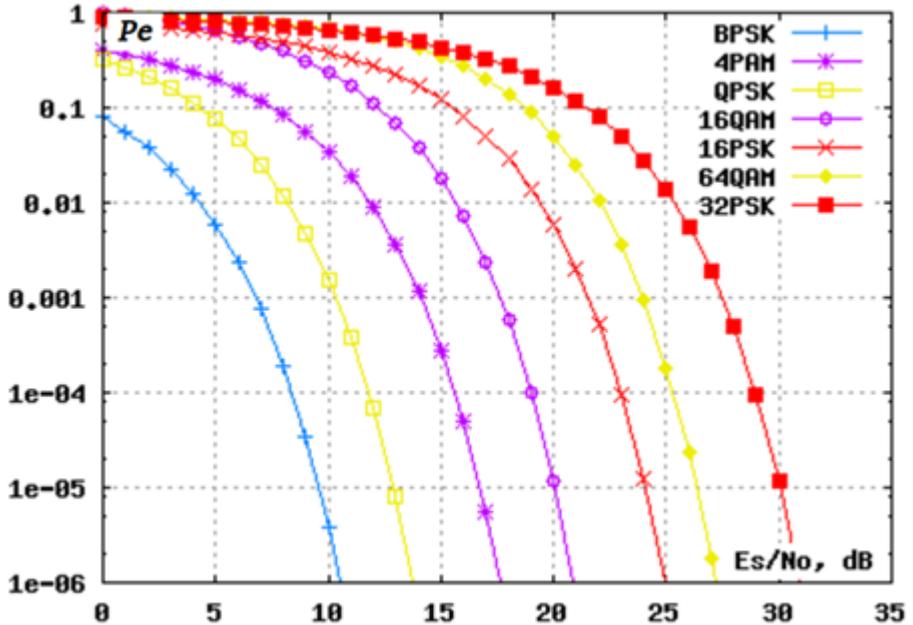


Al encontrar la probabilidad de pérdida de bits para cada valor de  $E_s/N_o$  o de  $E_b/N_o$ , se puede construir una gráfica que se conoce como la Curva de BER (del inglés *Bit Error Ratio*) o de SER (*Symbol Error Ratio*). En la figura 25 se tiene un ejemplo de las curvas de BER para las modulaciones BPSK, QPSK y 8PSK y 16PSK. De su observación podemos deducir lo siguiente:

- cuando la constelación está ordenada (mapeada) mediante Código de Gray la Curva de SER resulta ser prácticamente la misma curva de BER ya que el uso de los códigos de Gray garantiza que la pérdida de un símbolo se traduce en la pérdida de un bit.
- Entre más abajo esté la curva de BER para una Modulación, mejor desempeño muestra esa modulación ante el ruido blanco aditivo gaussiano.
- De acuerdo a la figura 25, las Modulaciones QPSK y la BPSK están coincidiendo, eso significa que la modulación QPSK es notablemente mejor, pues aunque tiene el mismo desempeño ante el ruido que la BPSK, es ya bien sabido que tiene mayor eficiencia espectral.
- Vemos que entre más alto sea el valor de M, para una modulación M-PSK menor es su desempeño ante el ruido.
- Supongamos que nuestro sistema debe soportar una pérdida de bits de al menos  $10^{-6}$  y en cierto momento de interés se presenta una relación  $E_b/N_0 = 16$ . Podemos ver que las siguientes modulaciones cumplen las condiciones dadas: BPSK, QPSK y 8PSK, entonces la modulación que más le conviene al sistema en esas condiciones es la 8-PSK, ya que aunque cumple las condiciones dadas y además tiene mayor eficiencia espectral que las demás.

Las curvas de BER también pueden ser dadas en función de Es/No como se muestra en la figura 26.

Figura 26: Curvas de BER Diversas Modulaciones.



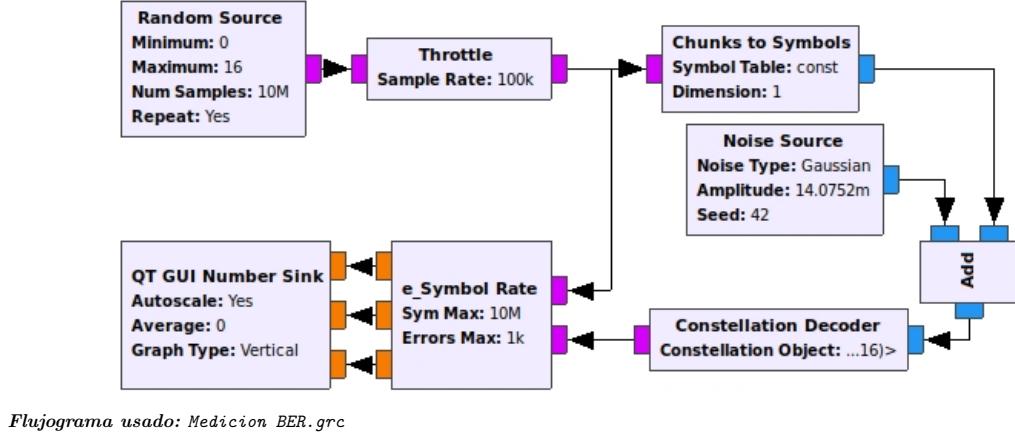
### 7.5.3. Implementación en GNU Radio

La BER puede ser evaluada usando simulación estocástica computarizada, lo cual se conoce mejor como el Método de Monte Carlo. Pero es bastante común también su cálculo analítico asumiendo un sencillo modelo del canal de transmisión y una fuente de datos, en este caso uno de los modelos es el de Bernoulli, el cual proporciona unas fórmulas ya listas para cada tipo de modulación.

A continuación se explica una implementación realizada por los autores para obtener las curvas de BER con el GRC de GNU Radio usando el método de Monte Carlo.

El primer paso consiste en contar con un medidor confiable de BER. En GRC de GNU Radio existe uno llamado “**Error Rate**”. Sin embargo, hemos optado por desarrollar uno propio ya que el anterior no brinda detalles necesarios para construir una Curva de BER como el número de bits procesados, el número de errores obtenidos. El bloque que hemos implementado lo hemos llamado “**e\_Symbol Rate**”. El mismo bloque puede servir para obtener mediciones de símbolos perdidos, por lo tanto para obtener Curvas de SER (Symbol Error Ratio) como se muestra en la figura 27, ya que a la entrada del bloque “**Chunk to Symbols**” y a la salida del bloque “**Constellation Decoder**” ya se tiene la señal en forma de símbolos M-PAM, lo cual es válido para diferentes tipos de modulación.

Figura 27: Flujograma para medir símbolos perdidos y SER



Con esta implementación podemos realizar las siguientes observaciones:

- Para obtener una comparación sobre el desempeño de un tipo de modulación con respecto a otra no es necesario recurrir a una implementación real.
- Lo único que hay que tener en cuenta es que la comparación se realice en condiciones similares
- También es importante que la implementación sea lo más sencilla posible ya que, dependiendo del grado de perfección que se desee para la medición de esos valores, el simulador tendrá que procesar un alto volumen de datos para estabilizar el promediado para obtener el valor de SER para un valor dado de  $E_b/N_0$ .
- En este sentido, solo hay que incluir, aparte de los elementos de modulación, un generador de ruido que permita variar el nivel de ruido en función del valor de la relación  $E_b/N_0$  o  $E_s/N_0$ . Eso lo hemos logrado con un sencillo canal compuesto por el bloque ‘‘Noise Source’’ y el ‘‘Add’’.
- El parámetro Amplitud del bloque ‘‘Noise Source’’ corresponde en realidad a la desviación estándar del ruido a generar, que para el caso dado, corresponde al valor RMS del ruido y a su vez corresponde al valor medio de los valores que toma la magnitud de la señal.
- Se sabe que la potencia promedio de una señal aleatoria es el valor RMS al cuadrado, de modo que la potencia promedio del ruido que entrega el ‘‘Noise Source’’ está relacionada con el valor RMS así:

$$V_{RMS} = \sqrt{P_r} \quad (7.21)$$

- En el capítulo 2.15 se estudió el ruido blanco gaussiano y se dijo que su PSD está distribuida en todas las frecuencias y tiene un valor constante igual a  $\frac{N_0}{2}$ .
- En el capítulo 3.5.2.1 quedó claro que cuando se trata de la Envoltura Compleja del ruido blanco, su PSD tiene un valor constante en todas las frecuencias igual a  $N_0$ .
- Aunque una señal como el ruido blanco ocupa todas las frecuencias, también es cierto que lo que se está generando el bloque ‘‘Noise Source’’ es una versión muestreada de ruido blanco en banda base
- También es importante tener en cuenta que, de lo que se trata en este caso, es de la Envoltura Compleja del ruido blanco discreto con una frecuencia de muestreo igual a la rata de símbolos.
- Del capítulo 2.12 se sabe que la potencia promedio de una señal es igual al área bajo la curva de su PSD.
- Por lo anterior, vale la pena recordar el Teorema de Muestreo (Teorema de Nyquist) explicado en el capítulo 4.8.1.1, el cual, para el caso del ruido blanco nos permite afirmar que, debido al muestreo, el bloque genera un ruido blanco pero limitado en su ancho de banda, entre la frecuencia  $F_{min} = -R_s/2$  y  $F_{max} = R_s/2$ , por lo tanto tenemos que la potencia promedio del ruido blanco simulado (Envoltura compleja del ruido blanco en versión discreta), cuando la frecuencia de muestreo es la misma rata de símbolos, es:

$$P_n = (F_{max} - F_{min})N_0 = R_s N_0 \quad (7.22)$$

- del análisis anterior se deduce que es más conveniente buscar la curva de SER que la de BER en función de  $E_s/N_0$ , pues el muestreo de las señales analizadas es igual a la rata de símbolos.

- Teniendo en cuenta la ecuación 7.20, puede decirse que la rata de símbolos está relacionada con la Energía y la Potencia de los símbolos recibidos de la siguiente manera:

$$R_s = \frac{P_s}{E_s} \quad (7.23)$$

Por lo tanto,

$$P_n = \frac{P_s}{E_s/N_0} \quad (7.24)$$

De aquí surge la siguiente importante deducción: La relación  $E_s/N_0$  es igual a la relación entre la potencia promedio de la señal útil con la del ruido blanco, que no es otra cosa que lo que se conoce como Relación Señal a Ruido o SNR (del inglés *Signal to Noise Relationship*) que es la relación entre la potencia de la señal y la potencia del ruido.

$$E_s/N_0 = \frac{P_s}{P_n} = SNR \quad (7.25)$$

Es importante dejar claro que esta igualdad se cumple cuando  $P_n$  ha sido medida a la entrada del demodulador, donde solo se tiene una muestra por cada símbolo. Si se mide en otra parte, el resultado puede ser diferente ya que el ruido afecta de manera diferente a la información. Por ejemplo, está claro que un filtro de acoplamiento atenúa el ruido blanco, entonces no es lo mismo el ruido antes de ese filtro que después.

Finalmente, encontramos que el bloque ‘‘Noise Source’’ debe ser configurado con el siguiente valor para el parámetro ‘‘Amplitude’’:

$$V_{RMS} = \sqrt{P_n} = \sqrt{\frac{P_s}{SNR}} = \sqrt{\frac{P_s}{E_b/N_0}} \quad (7.26)$$

- El bloque ‘‘e\_symbol Rate’’ puede también calcular bits perdidos si en sus dos entradas se conectan señales binarias en vez de M-arias, que es como está la configuración de la figura 27.
- El término SER y BER aparentan ser lo mismo, lo cual se puede ver en el siguiente ejemplo: Si entre  $10^6$  símbolos transmitidos se presentan 10 erróneos, se tiene que la probabilidad de pérdida de símbolos es:

$$SER = \frac{10}{10^6} = 10^{-5} \quad (7.27)$$

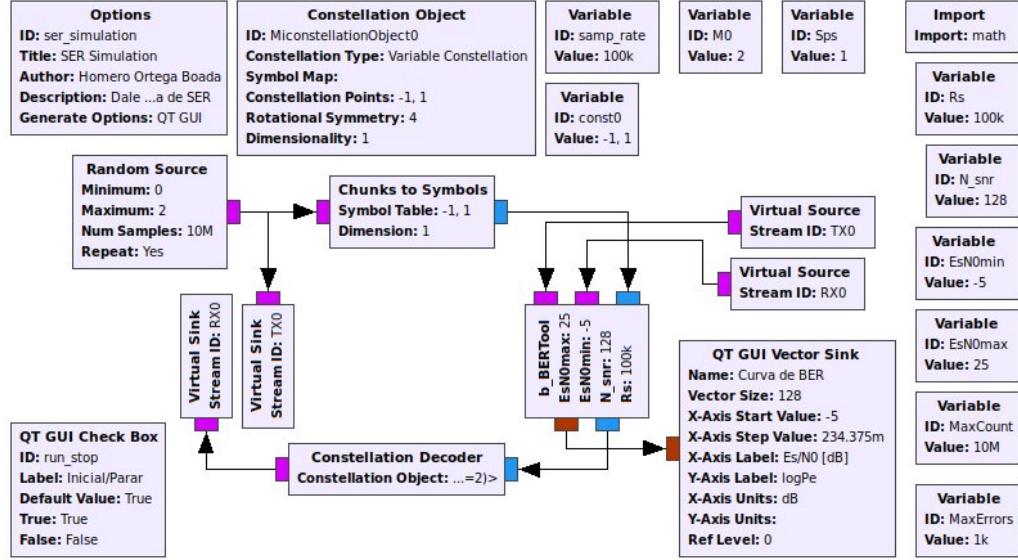
Supongamos ahora que se usa la modulación 8-PSK, la cual lleva  $bps = \log_2(8) = 3$  bits por símbolo. Entonces se habrán enviado en realidad  $3 \cdot 10^6$  y se habrán presentado 30 errores, de modo que:

$$BER = \frac{30}{3 \cdot 10^6} = 10^{-5} \quad (7.28)$$

- Pero en realidad no es tan así. Como vimos al hablar de los códigos de Gray, cuando se presenta un error, porque un símbolo ha sido confundido por uno vecino en la constelación, solo se ha perdido un bit, no los  $bps$  bits que hay en cada símbolo. Si no se aplica una codificación especial puede ocurrir lo contrario, que la pérdida de cada símbolo se traduzca en una pérdida más dramática de bits. Por lo tanto, dependiendo el grado de sofisticación del modulador y el demodulador, la curva de BER puede llegar a mostrar resultados más alagadores o más pesimistas que la de SER. En todo caso, la curva de SER revela resultados más genéricos que la BER.

ha mejorado el flujo de datos y ahora se llama sertool. Se ubica en el mismo repositorio

Figura 28: Flujograma para obtener la Curva de BER para cualquier tipo de modulación.



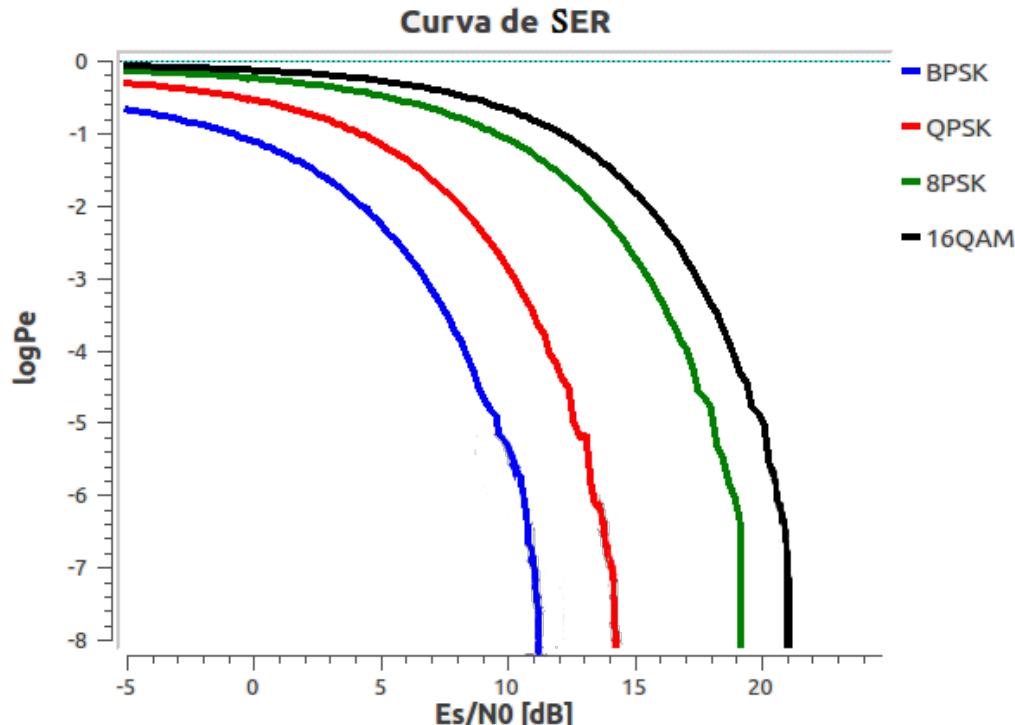
El bloque “e\_Symbol Rate”, usado apropiadamente, permite obtener valores de BER o de SER para distintos valores de  $E_b/N_0$  o  $E_s/N_0$ , pero no crea realmente una Curva de BER o de SER. GNU Radio y menos aún el GRC traen herramientas para obtener las Curvas de BER o de SER. En Matlab y en Simulink de Matlab se tiene una solución que consiste en crear un software para el sistema de comunicación y otro software para llamar al primero con diferentes valores de Es/No. Lo que proponemos en la figura 28 es algo mucho más sencillo y novedoso.

La solución consiste en la implementación del bloque ‘‘B\_BERTool’’, el cual tiene las siguientes cualidades:

- Funciona como un canal de ruido blanco Gaussiano Aditivo, es decir un canal AWGN. Pero tiene un comportamiento particular:
- Por un lado, va cambiando internamente la potencia del ruido de una manera conveniente para crear las más diversas condiciones para medir la BER o la SER.
- Además de pedir como entrada la señal modulada, también pide la señal de los bits o de los símbolos transmitidos y la de los bits o de los símbolos recibidos en el receptor.
- El bloque cuenta entonces con todos los datos para realizar los cálculos para obtener una curva de BER o de SER, la cual la entrega en forma de un vector de  $N_{snr}$  valores.
- Solicita un valor mínimo y uno máximo para la relación  $E_s/N_0$ . El bloque también pide el número de puntos para cada curva de BER o SER, lo cual se configura en el parámetro  $N_{snr}$ .
- El funcionamiento del bloque comienza por crear un vector de  $N_{snr}$  con valores tomados entre el valor mínimo y máximo dado para la relación  $E_s/N_0$ . Ese vector lo usa para generar el nivel de ruido apropiado para obtener un vector con los valores de la Curva de BER o de SER para poder entregar un vector de  $N_{snr}$  elementos de la Curva de BER o de SER. Esta curva se va ajustando todo el tiempo a medida que crece el número de bits o de símbolos que entran por las entradas `in_Tx` e `in_Rx`.
- La salida del bloque es un vector de  $N_{snr}$  valores de la Curva de BER o de SER que puede irse graficando en un bloque ‘‘QT GUI Vector Sink’’.

En la figura 29 se tienen las curvas obtenidas para los tipos de modulación estudiados.

Figura 29: Curvas de BER obtenidas con GNU Radio.

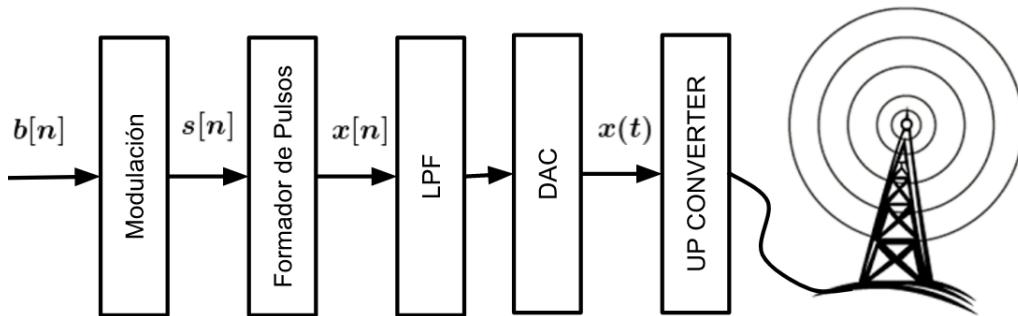


Flujograma usado: `bertool_plus_M.grc`

## 7.6. Formación de Pulses y Acoplamiento en comunicaciones Digitales inalámbricas

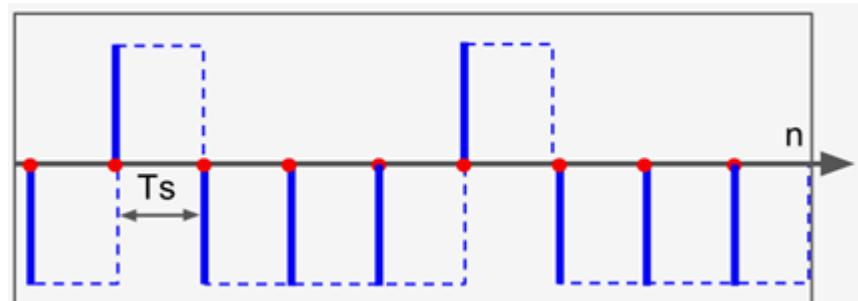
En GNU Radio el **Wave Forming** es también un paso necesario cuando se tiene una señal con modulación digital. La diferencia es que esa señal será compleja ya que se tratará de una señal que espera el *Up Converter*, el cual a su vez se encuentra en el Hardware de una solución SDR. Antes de llegar al *Up Converter* la señal debe pasar por un conversor digital analógico (DAC) como es el caso del hardware usado en el NI USRP 2920 visto en el capítulo 1 y que se resume en la figura 30.

Figura 30: El Formador de Pulses en una solución SDR.



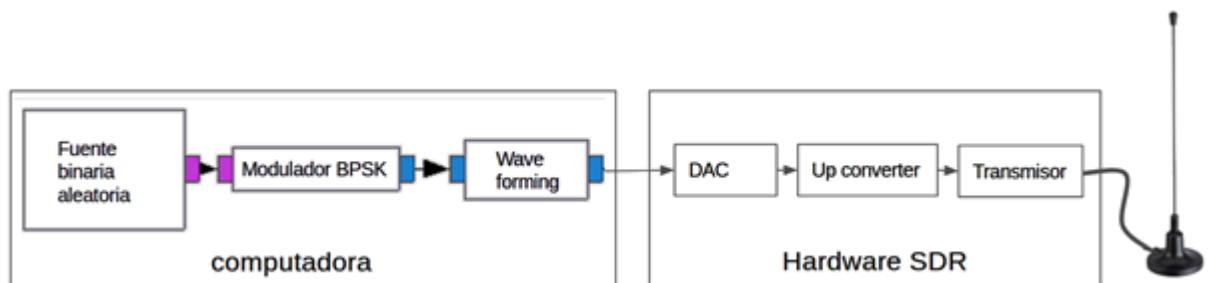
En la figura 31 se tiene un ejemplo de la señal que entrega un modulador digital BPSK en GNU Radio, la cual es compleja por tratarse de la versión bandabase. Como puede verse, esa señal está un tanto alejada de la correspondiente en el mundo real, que son los cuadros que aparecen punteados en la figura 31.

Figura 31: Ejemplo de la señal que entrega el modulador BPSK.



*Nota: En línea azul grueso la componente real, en rojo la imaginaria, en punteada la que corresponde en el mundo físico. Ts - duración del símbolo*

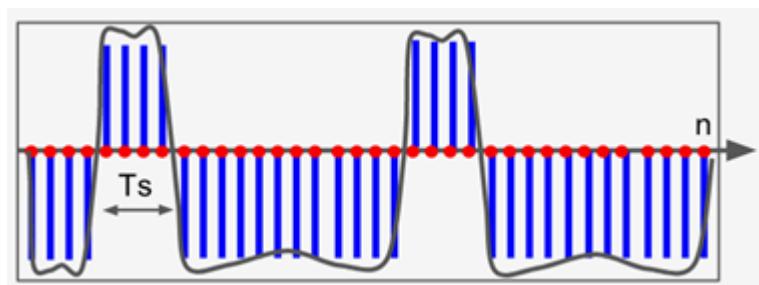
Figura 32: El Wave Forming como intermediario entre el Modulador y el DAC.



#### 7.6.0.1. Wave Forming basado en pulsos rectangulares

En la figura 33, en color azul muestra la señal que se podría tener a la salida del bloque Wave Forming si este se configura para que cada símbolo se repita cuatro veces en la misma duración  $T_s$ . Entonces, se obtiene una señal con una frecuencia de muestreo 4 veces mayor a la anterior la cual es más parecida a la esperada en el mundo real.

Figura 33: En línea azul la componente real y en rojo la señal a la salida del bloque Wave Forming. En negro la salida del DAC.  $Sps=4$ .



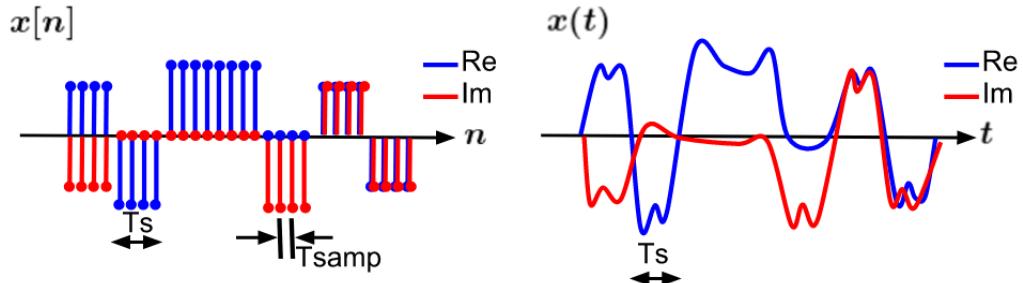
El DAC entrega como resultado la señal que aparece en color negro para la parte real y una de nivel cero para la parte imaginaria. Tampoco se ha logrado reproducir la señal continua en forma ideal como la mostrada en forma punteada en la figura 31, ya que esa señal tendría un espectro de ancho de banda infinito, lo que significa que es necesario que haya infinitas muestras por bit en la señal que se entrega al DAC, lo cual es imposible desde todo punto de vista. Se obtiene más bien una señal que tiene una forma similar a la esperada y que de paso tiene un ancho de banda finito igual a:

$$BW = \frac{\text{samp\_rate}}{2} = \frac{Sps}{2T_s} \quad (7.29)$$

Donde **samp-rate** es la frecuencia de muestreo de la señal que entra al ADC y **Sps** es el número de muestras por símbolo, (del inglés *samples per symbol*), lo cual es coherente con el Teorema de Nyquist.

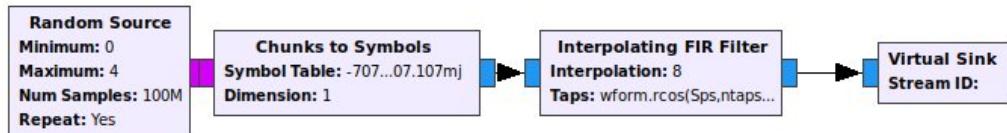
En la figura 34 se tiene otro ejemplo para el caso de la modulación 8PSK.

Figura 34: Envolvente Compleja con forma rectangular para la Modulación 8PSK.



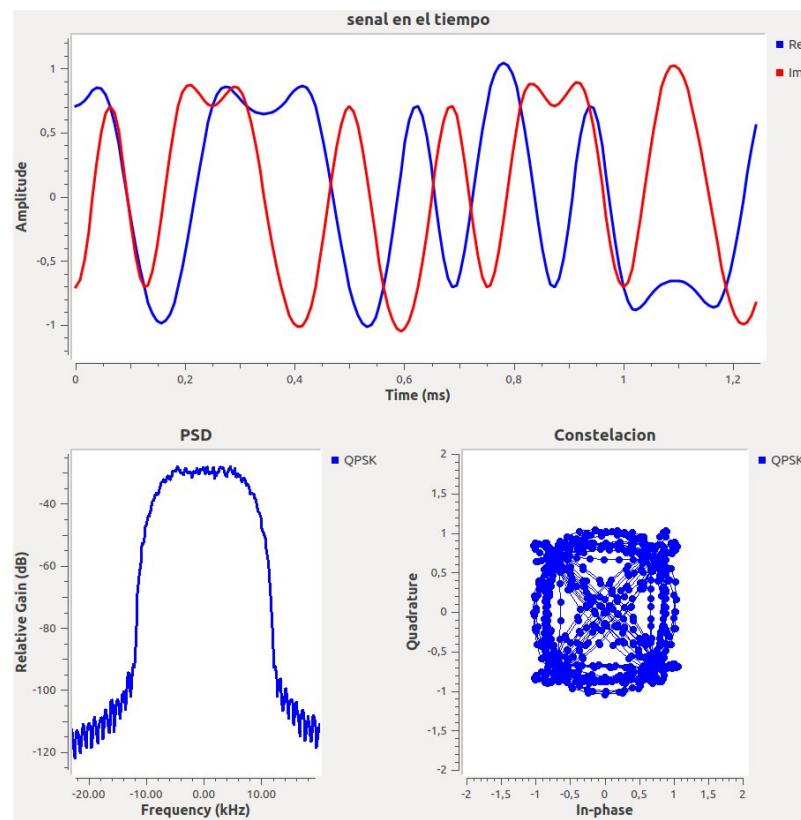
#### 7.6.0.2. El Filtro Coseno Alzado

Figura 35: Flujograma con Modulación Digital y Formador de Pulso.



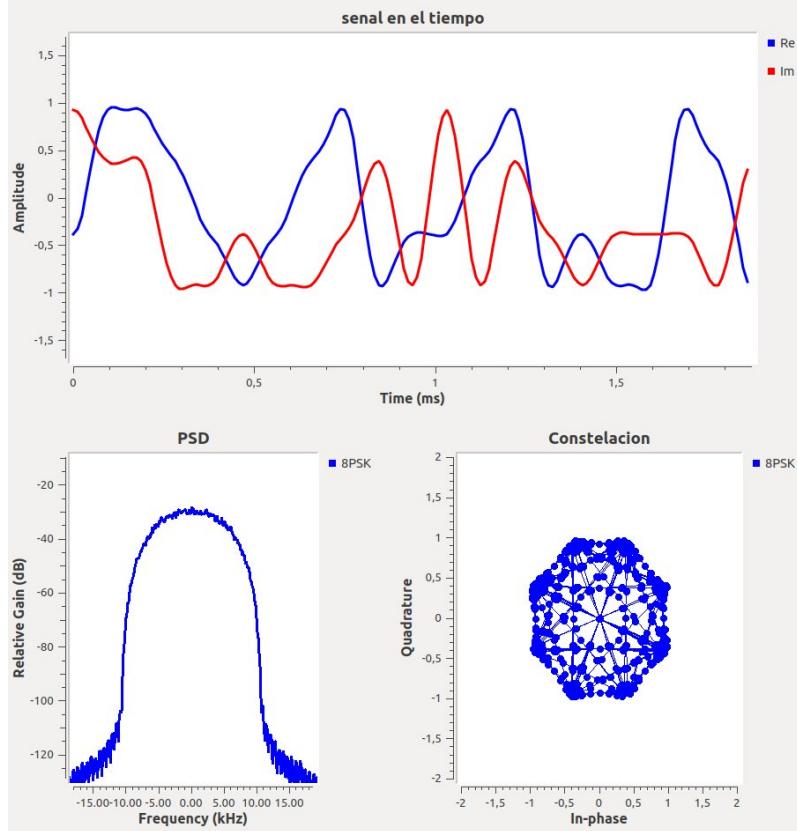
Flujograma usado: *WaveForming\_QPSK.grc*

Figura 36: Resultados de Wave Forming aplicado a la modulación QPSK, con Filtro RC y rolloff=0.5



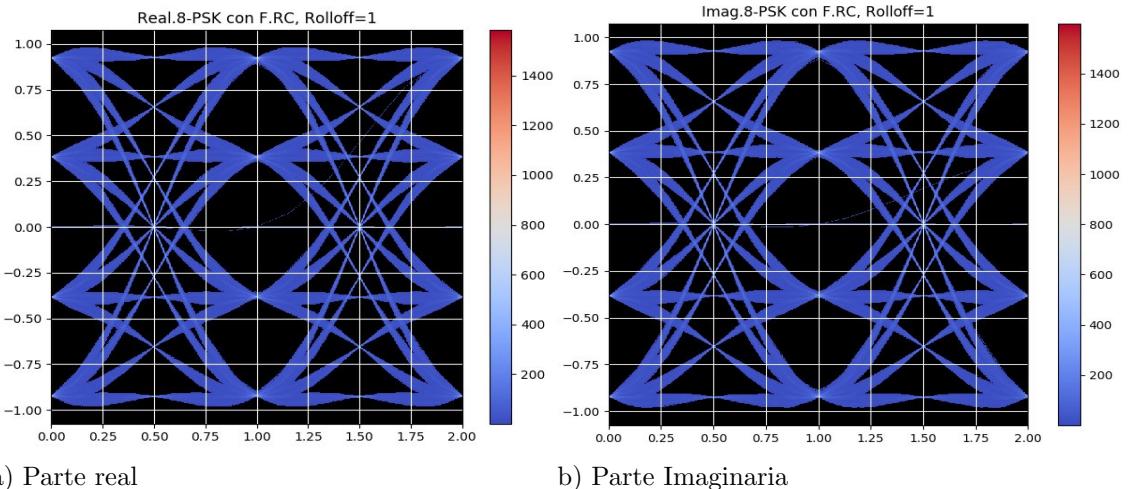
Flujograma usado: *WaveForming-QPSK.grc*

Figura 37: Resultados de Wave Forming aplicado a la modulación 8PSK, con Filtro RC y  $rolloff = 1$



Flujograma usado: *WaveForming\_8PSK.grc*

Figura 38: Diagrama de Ojo de una señal con Modulación 8PSK, con Filtro RC, rolloff=1



En la figura 35 se presenta el flujo de datos de un sistema básico que permite generar una señal M-PAM, que pasa a un Modulador basado en constelaciones implementado en el bloque ‘‘Chunks to Symbols’’ y finalmente pasa por el formador de pulsos implementado mediante el bloque ‘‘Interpolating FIR Filter’’. Sobre el Formador de pulsos podemos decir los siguientes:

- Para el formador de pulsos hemos usado exactamente el mismo bloque que hemos usado para el caso de señales reales, lo único diferente es su configuración que requiere lo siguiente: El parámetro ‘‘Type’’ se configura como *Complex* → *Complex(RealTaps)*, que significa que la entrada y la salida del bloque son complejas, pero la respuesta al impulso la introducimos como valores reales.
- Cualquier tipo de modulación puede ser configurada en el bloque ‘‘*Chunks to Symbol*’’.
- Cualquiera de los Formadores de pulsos vistos puede ser configurado en el bloque ‘‘*Interpolation FIR Filter*’’.

Si corremos este flujo de trabajo, para  $Sps = 8$ , Modulación QPSK, Filtro RC con  $rolloff = 0,5$  obtenemos los resultados presentados en la figura 36. En la figura 37 tenemos otro ejemplo para el caso de la modulación 8PSK, usando Filtro RC, con  $rolloff = 1$

De la misma manera, es posible encontrar el diagrama de ojo, pero claro está, para la parte real y la imaginaria como se muestra en la figura 38.

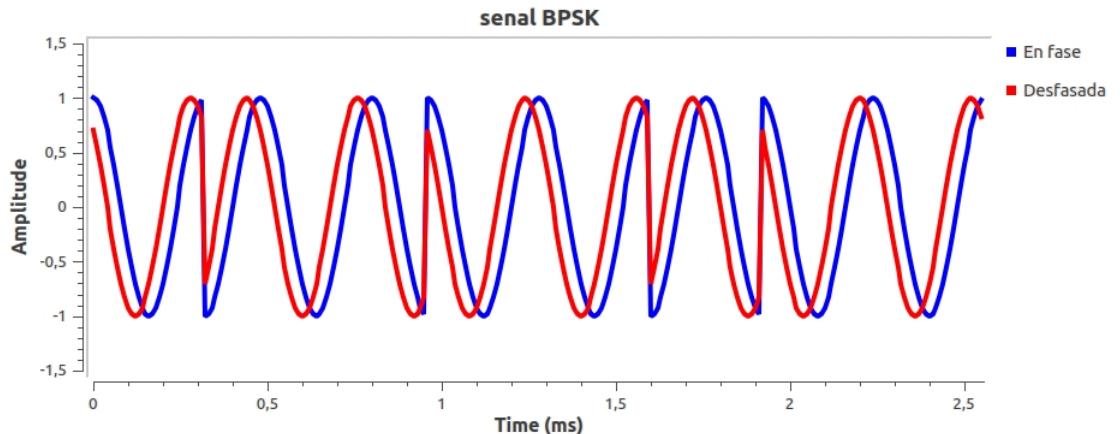
Igualmente, en este tipo de modulaciones aplican los filtros de acoplamiento para eliminar el ruido blanco, lo cual se deduce de lo ya explicado. Por lo tanto, también se usa ampliamente el Filtro RRC.

## 7.7. Consecuencias de otros Fenómenos

### 7.7.1. Desviación angular

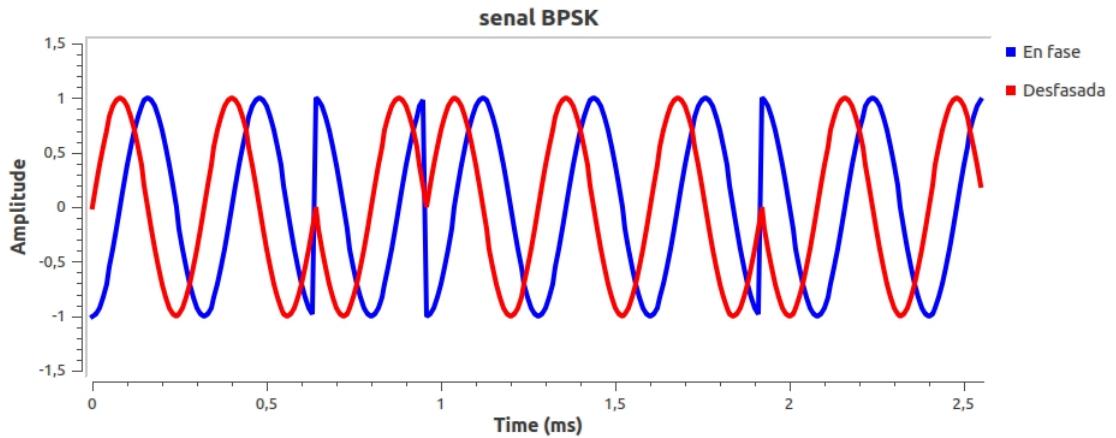
En términos de señal paso banda, se refiere a la desviación angular que existe entre dos señales. En la figura 39 se tiene un ejemplo para la modulación BPSK, donde la versión original está en color azul y la versión desfasada en 45 grados está en color rojo.

Figura 39: Señal BPSK y su versión desfasada en 45 grados.



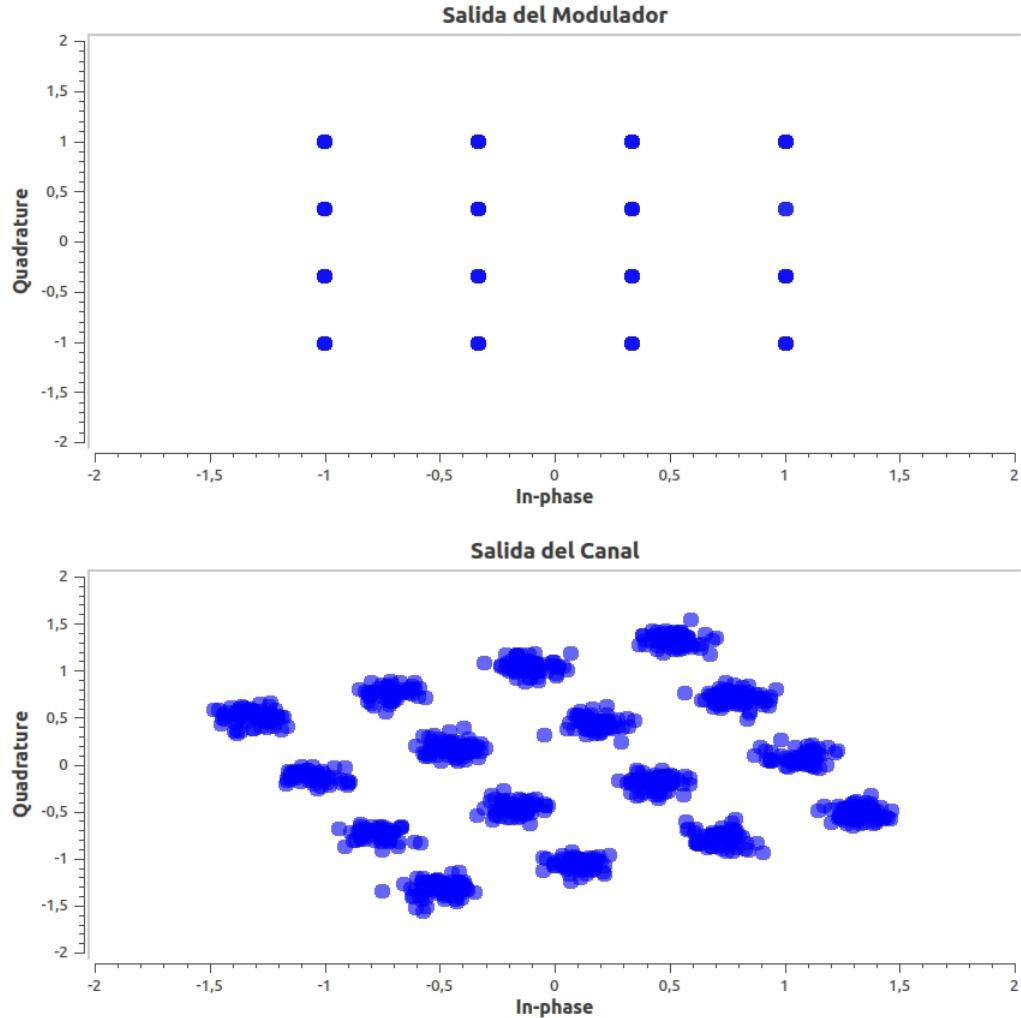
En la figura 40 se tiene otro ejemplo para el caso en que el desfase es de 90 grados. En los sistemas de comunicación, el desfase es prácticamente un fenómeno obligatorio dado principalmente por el paso de la señal a través de diferentes condiciones del medio de transmisión, que para nuestro caso es el canal inalámbrico. También se puede producir en el paso de la señal por diferentes circuitos de radio que se tienen tanto en los equipos de transmisión como de recepción. Incluso, aún en el caso hipotético en que la señal llegue sin desfase alguno, este puede aparecer al intentar bajar la señal a bandabase, ya que es poco probable que la fase del oscilador local del *Down converter* esté en fase con la señal recibida.

Figura 40: Señal BPSK y su versión desfasada en 90 grados.



Cuando se usan las técnicas de SDR la desviación de fase que llega al receptor no alcanza a ser corregida completamente en el hardware, de manera que se presenta en la Envolvente Compleja discreta que se recibe. En la figura 41 se presenta un ejemplo para el caso de la modulación 16QAM, en la parte superior se tiene la constelación generada en el modulador de la parte transmisora, en la parte inferior está la constelación que se obtiene en el receptor cuando se presenta la desviación angular.

Figura 41: Constelación de la Modulación 16QAM y su versión con desviación de fase.



La implementación de un simulador de un canal banda base con desviación de fase puede realizarse de la siguiente manera:

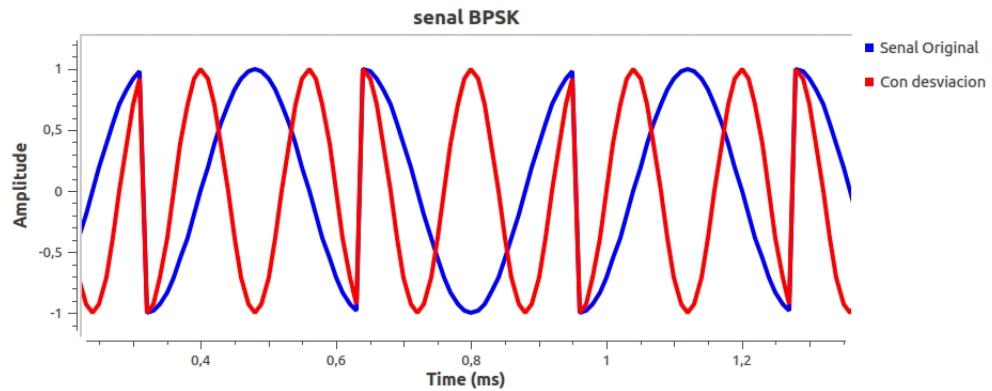
$$z(t) = s(t)e^{j\phi} \quad (7.30)$$

donde  $s(t)$  es la Envolvente Compleja de la señal que el transmisor entrega al canal;  $\phi$  es el desfase que introduce el canal; y  $z(t)$  es la salida del canal.

### 7.7.2. Desviación en frecuencia

Las desviaciones entre la frecuencia usada en la transmisión y la que llega a un receptor, luego de que la señal atraviesa un canal, son también un fenómeno casi obligado, sobre todo cuando el canal es inalámbrico. La causa principal es el Efecto Doppler, aunque pueden haber otras como por ejemplo las imperfecciones en los circuitos que generan la portadora tanto en el transmisor como en el receptor. En la figura 42 se presenta un ejemplo, donde la señal en color rojo aparece dos veces desviada de la original, sin embargo, en la práctica esta desviación suele ser de unos pocos Hertz.

Figura 42: Desviación Frecuencias entre dos señales BPSK.



Como en el caso de la desviación de fase, al usar técnicas de SDR, las desviaciones de frecuencias no alcanzan a ser corregidas completamente con el hardware, de modo que es un problema que se traslada a las componentes de software. En este sentido, nos ocuparemos ahora en conocer cómo es la Envolvente Compleja de una señal con desviación de frecuencias comparada con una sin ella. En este caso, la desviación está dada por la siguiente expresión:

$$z(t) = s(t)e^{j2\pi f_{desv}t} \quad (7.31)$$

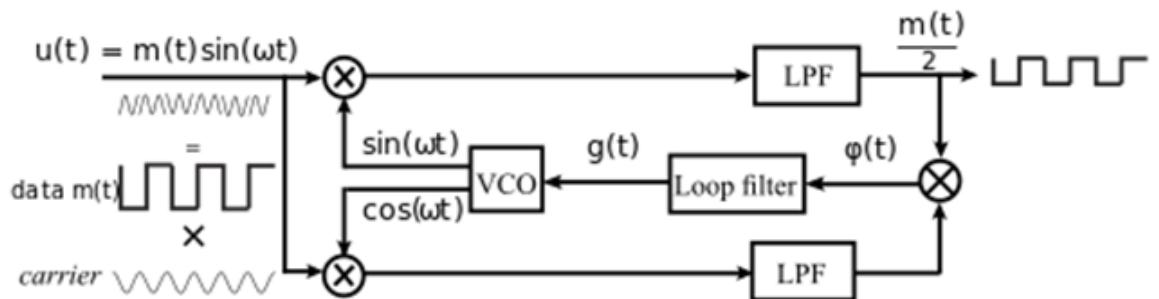
donde  $s(t)$  es la Envolvente Compleja de la señal que el transmisor entrega al canal;  $f_{desv}$  es la desviación de frecuencia que introduce el canal; y  $z(t)$  es la salida del canal.

Desde este punto de vista, es fácil deducir que una señal con desviación de frecuencias, puede ser observada a la salida del *Down converter* como una constelación similar a la mostrada en la figura 41 pero no estática, sino girando en contra de las manecillas del reloj si la desviación es positiva o en favor de las manecillas del reloj si es negativa.

### 7.7.3. Frequency Lock Loop (FLL) y Phase Local Loop (PLL)

Un método sencillo para estabilizar una constelación consiste en que el bloque de sincronización tenga la información sobre la constelación usada. Se basa en la implementación de un Bucle de Costas (Costas Loop) para medir la señal recibida en búsqueda del error que presenta con respecto al punto de constelación más cercano en la constelación que se tiene como modelo. La siguiente figura ha sido tomada de Wikipedia y refleja muy bien la idea: una señal  $m(t)$  puede llegar con una desviación de frecuencias  $\omega_d$ , se involucra un VCO para corregirla.

Figura 43: Frequency Lock Loop.



fuente: <https://en.wikipedia.org/wiki/Costas-loop>

De manera similar funciona el FLL (Frequency Lock loop), pero  $m(t)$  es una Envolvente Compleja. El bloque puede fallar cuando las desviaciones son exageradamente altas. El bloque se conecta usualmente después del ecualizador.

Figura 44: Resultados del sistema de comunicación FM.

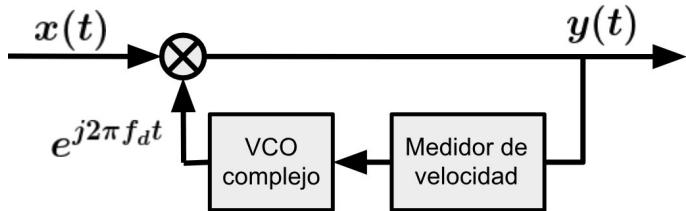
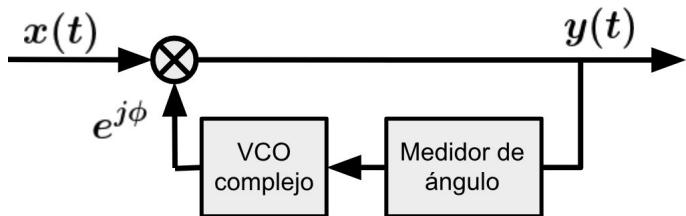


Figura 45: PLL para cuando la entrada es la Envolvente Compleja.



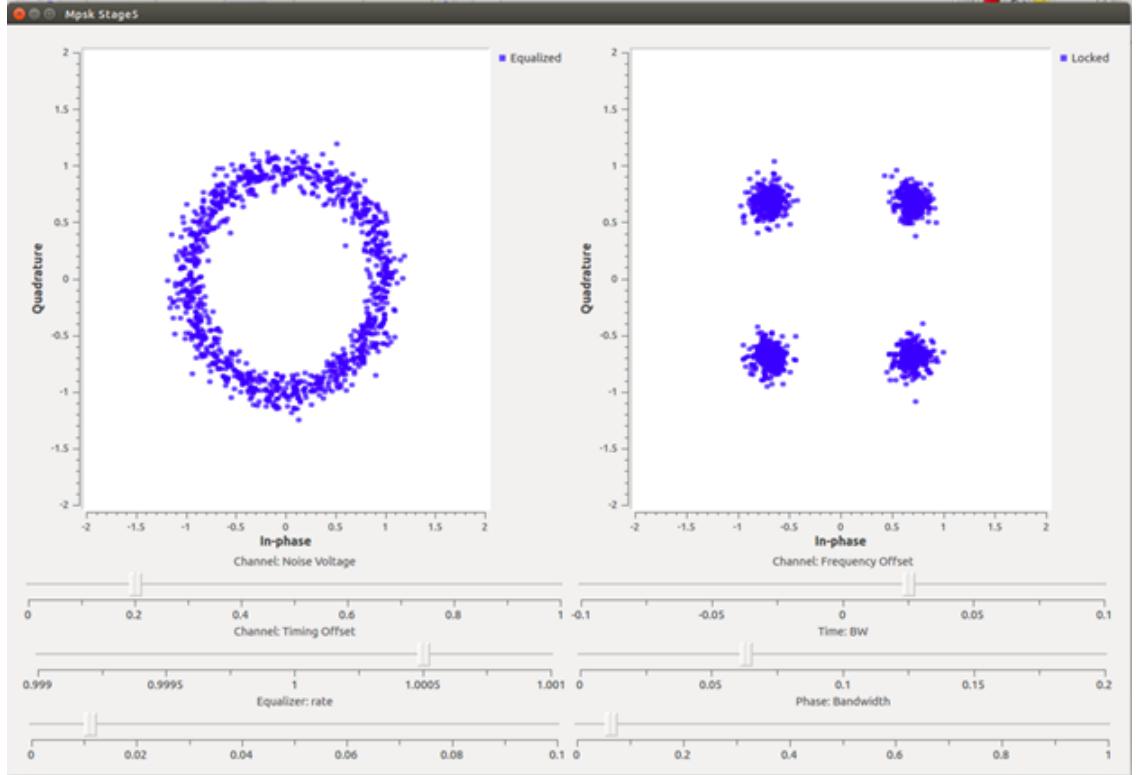
En la figura 44 se tiene una interconexión que puede ser usada para construir un bloque FLL en GNU Radio teniendo en cuenta que en GNU Radio no contamos con la señal pasobandas sino con la Envolvente Compleja. Por ello, lo que el generador debe suministrar es una onda compleja. El medidor de velocidad mide la velocidad del vector rotante para ordenar al VCO generar la onda compleja con la frecuencia apropiada. Esta solución aplica para aquellos tipos de modulación dados por constelaciones estáticas, de modo que una desviación de frecuencias  $f_d$  hace que la constelación representada en  $x(t)$  esté girando. Al multiplicar  $x(t)$  por  $e^{-j2\pi f_d t}$  se logra neutralizar el giro y  $y(t)$  aparece como una constelación estática. Lo mismo pasa para construir un bloque PLL, como se muestra en la figura 45, solo que ahora el VCO lo que entrega es un valor complejo  $e^{-j\phi}$ .

mejorar la gráfica para que la señal usada en la multiplicación sea  $e^{-j2\pi f_d t}$

mejorar la gráfica para que la constante usada en la multiplicación sea  $e^{-j\phi}$

Otro bloque es el “Costas Loop”. Este bloque está pensado para ser usados en modulaciones M-PSK, utiliza un bucle de segundo orden y por lo tanto se define con un parámetro de ancho de banda de bucle. La otra cosa que este bloque necesita es el orden de la modulación M-PSK, de modo que es 2 para BPSK, 4 para QPSK y 8 para 8PSK. En la 46 podemos ver que los símbolos a la entrada del bloque “Costas Loop” están todos en el círculo de unidad, pero girando debido a que aún no se está realizando la compensación de frecuencia ni la fase. A la salida del bloque de bucle Costas, podemos ver la constelación bloqueada más el ruido extra que no podemos suprimir.

Figura 46: Antes y después de aplicar FLL.



#### 7.7.4. La codificación Diferencial para atacar la ambigüedad de fase

Obsérvese que en nuestra discusión anterior la corrección de las desviaciones de frecuencia o de fase se basan en un conocimiento previo de la forma de la constelación, pero debido a que la constelación tiene una forma simétrica es posible que la constelación recibida y procesada resulte girada en un ángulo tal que tenga la misma apariencia de la constelación transmitida. Entonces ocurre que cada punto puede resultar expresando un símbolo diferente al transmitido. Para resolver este problema hay que aprovechar que, en todo caso, se mantiene el orden de la secuencia de los símbolos, es decir, los símbolos siguen llegando en el mismo orden en que fueron emitidos, solo que con fases diferentes.

##### 7.7.4.1. Proceso de codificación diferencial

La idea es no transmitir los símbolos tal y como se originan sino los que resulten de la diferencia entre ellos, de modo que la información útil no se asocia a un punto de constelación sino a la diferencia entre puntos de constelación. En la parte receptora se obtiene cada símbolo a partir de su relación con otro símbolo adyacente.

En la parte transmisora, la solución se implementa mediante un codificador diferencial. Para el caso más sencillo, supongamos que los símbolos transmitidos son simplemente bits y que en un instante determinado, a la entrada del codificador se tiene el bit  $x_i$ . La salida  $y_i$  se calcula así.

$$y_i = y_{i-1} \oplus x_i \quad (7.32)$$

Donde  $\oplus$  es la operación de suma binaria o suma por módulo 2.

En la parte receptora, se realiza la operación inversa

$$x_i = y_i \oplus y_{i-1} \quad (7.33)$$

De modo que en el receptor el valor de  $x_i$  no está en un determinado símbolo  $y_i$  sino en la diferencia con respecto al símbolo anterior  $y_{i-1}$ .

Para llevar a cabo esta idea en el caso en que se usan símbolos compuestos por  $k$  bits, es necesario aplicar la operación a paquetes de  $k$  bits, así, en el caso de BPSK,  $k = 1$ ; en QPSK,  $k = 2$ ; en 8PSK,  $k = 3$ ; en 16QAM,  $k = 4$ . Cabe notar que  $k = \log_2(M)$ , donde  $M$  es el número de puntos de constelación de la modulación de interés.

Veamos un ejemplo para la modulación 8PSK:

- El símbolo actual es  $x_i = 110$
- El símbolo anterior era  $y_{i-1} = 101$
- La salida del encoder es:  $y_i = y_{i-1} \oplus x_i = 101 \oplus 110 = 011$
- En el decoder se realiza la operación:  $x_i = y_i \oplus y_{i-1} = 011 \oplus 101 = 110$

Veamos un ejemplo más completo para la misma modulación:

Supongamos que la señal a la entrada del codificador es 110010011100 y que puede ser vista como símbolos de a 3 bits de la siguiente manera:

$$x = [110][010][011][100]$$

Supongamos que al iniciar el proceso el codificador está en estado:

$$y_0 = [111]$$

Con la llegada del primer símbolo al codificador se produce la salida de la siguiente manera:

$$y_1 = y_0 \oplus x_1 = 111 \oplus 110 = 001$$

Con el segundo símbolo

$$y_2 = y_1 \oplus x_2 = 001 \oplus 010 = 011$$

Con el tercer símbolo

$$y_3 = y_2 \oplus x_3 = 011 \oplus 011 = 000$$

Con el cuarto símbolo

$$y_4 = y_3 \oplus x_4 = 000 \oplus 100 = 100$$

Veamos ahora lo que ocurre en la parte receptora

Al decodificador llega la señal:

$$y = [111][001][011][000][100]$$

Con la llegada del primer símbolo al decodificador se produce la salida de la siguiente manera:

$$x_1 = y_0 \oplus y_1 = 111 \oplus 001 = 110$$

Con el segundo símbolo

$$x_2 = y_1 \oplus y_2 = 001 \oplus 011 = 010$$

Con el tercer símbolo

$$x_3 = y_2 \oplus y_3 = 011 \oplus 000 = 011$$

Con el cuarto símbolo

$$x_4 = y_3 \oplus y_4 = 000 \oplus 100 = 100$$

Como podemos ver, la salida del decodificador es: 110010011100 y es igual a la entrada del codificador.

#### 7.7.4.2. La codificación diferencial generalizada

La operación  $y_{i-1} \oplus x_i$  no es la única forma para una codificación diferencial. De manera más genérica, se puede usar una función  $u = F(x, y)$  siempre y cuando la ecuación  $u_0 = F(y_0, x)$  solo tenga una solución para cualquier  $y_0$  y  $u_0$ .

#### 7.7.4.3. Problemas de la codificación diferencial

La codificación diferencial tiene una desventaja significativa: Multiplica errores. Esto es, si un símbolo  $y_i$  es recibido incorrectamente, habrán dos símbolos de información  $x_i$  y  $x_{i+1}$  decodificados incorrectamente ya que ambos dependen de  $y_i$ , pues:  $x_i = y_i \oplus y_{i-1}$ , pero  $x_{i+1} = y_{i+1} \oplus y_i$ . Por esa razón, la tasa de símbolos perdidos se dobla.

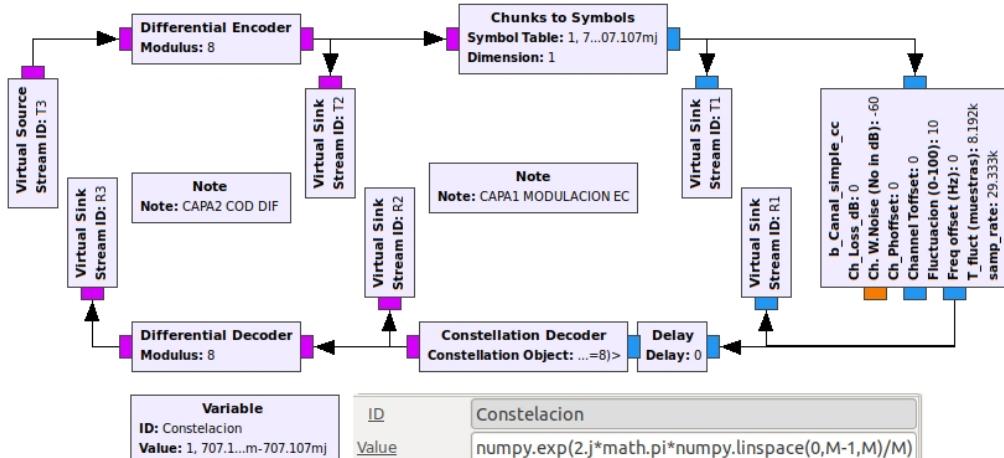
Pero hay otro problema: solo aplica para aquellos tipos de modulación donde la información va expresada solo con claves de fase como es el caso de BPSK, QPSK, 8PSK y en general M-PSK.

Por lo anterior, existen otras alternativas para resolver ambigüedad de fase. Por ejemplo, en la televisión digital satelital, con el sistema DVB-S, se usan *syncword* sobre una trama de sincronización, de manera que si en esa trama se detecta que las *syncword* están invertidas de alguna manera, todos los datos proceden a ser invertidos de manera consecuente.

#### 7.7.4.4. Implementación en GNU Radio de la Codificación Diferencial

GNU Radio trae una pareja de bloques dedicados a este tipo de codificación, son los bloques ‘‘Differential Encoder’’ y ‘‘Differential Decoder’’, como se muestra en la figura 47. Estos bloques trabajan con señales reales M-arias, que hasta el momento hemos llamado señales M-PAM.

Figura 47: Flujograma para implementar la capa base que incluye codificación diferencial y modulación con EC.



Flujograma usado: cod\_diff2.grc

En el ejemplo de la figura 47 hemos tenido que agregar un canal, el bloque ‘‘b\_Canal\_simple\_cc’’ de por medio que nos permitiera injectar desviaciones angulares controladas para comprobar el comportamiento del codificador diferencial.

#### 7.7.4.5. Implementación en GNU Radio de la codificación diferencial con mapeo

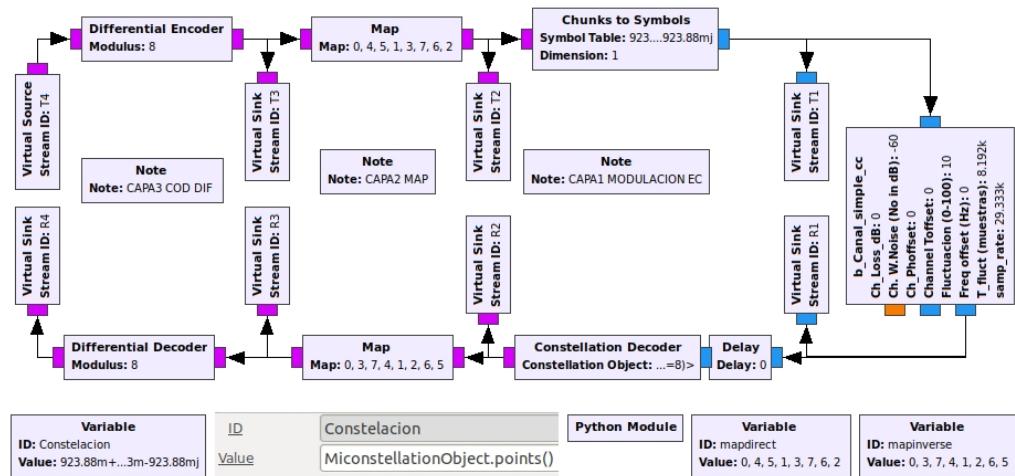
La solución implementada en el punto anterior tiene aún un gran problema: Solo funciona cuando los puntos de constelación están ordenados de manera consecutiva, lo cual no es el caso común pues la mayoría de las veces encontraremos esos puntos ordenados con el Código de Gray o incluso en un orden totalmente diferente.

La solución que se propone a continuación funciona igual de bien cuando los puntos de constelación están ordenados de acuerdo al código de Gray o con cualquier otro método. La idea es analizar previamente la constelación usada para identificar la manera de poder reordenar las muestras de la señal que entra al modulador de un modo tal que a la salida aparezca como puntos de constelación ordenados de manera decimal. Eso sin alterar los beneficios que tiene el orden usado internamente para los puntos de constelación ya que no se está modificando la Constelación, solo el orden en que las muestras entran.

En la Figura 48 se muestra el flujograma de la solución, donde se destacan como nuevos elementos: un bloque “Map” antes del modulador, otro después del Demodulador y unas nuevas variables.

La novedad está en el siguiente procedimiento:

Figura 48: PSD de para una señal super super QPSK



Flujograma usado: cod\_diff\_map.grc

- En un bloque “Python Module”, que para el ejemplo dado hemos llamado “coding” se crean las funciones que permiten encontrar, para cualquier constelación, el orden en que están organizados sus puntos de constelación. Como resultado obtendremos un código de mapeo directo y uno inverso. El siguiente es el código en Python:

```

import numpy
import math

def sum_modulus(a,b,modulus):
 # realiza una suma de a y b en modulo modulus
 return (a+b)%modulus

def angle_map(constellation_points):
 # esta funcion entrega todos los angulos positivos entre 0 y 2pi
 angles=numpy.angle(constellation_points)
 return sum_modulus(angles, 2*math.pi, 2*math.pi)

def inverse_map(constellation_points):
 M=len(constellation_points)
 angles=angle_map(constellation_points)
 angles=sum_modulus(angles,-angles[0], 2*math.pi) # giro a agulo cero
 delta_angle= 2*math.pi/M # angulo entre puntos
 cod_inv=numpy.round(angles/delta_angle)
 cod_inv=cod_inv.astype(numpy.int) #conversion a entero
 return cod_inv

```

```

def direct_map(constellation_points):
 M=len(constellation_points)
 codinv=inverse_map(constellation_points)
 coddirect=numpy.array([0]*M)
 for j in range(M):
 temp=numpy.where(codinv==j)
 coddirect[j]=temp[0][0]
 return coddirect

```

- No importa como esté definida la constelación, pero el bloque “Constellation Object” debe tenerla registrada en el parámetro “Constellation Points”, por ejemplo:

```
|| Constellation Points: digital.constellation_8psk().points()
```

- Para el ejemplo dado, el objeto de constelación se llama “MiconstellationObject” y los puntos de constelación los estamos invocando desde la variable “Constelacion” así:

```
|| Constelacion: MiconstellationObject.points()
```

- Se calcula el código de mapeo directo y el inverso para el caso particular. En el ejemplo dado se ha calculado dentro de la variable “mapdirect” configurada como

```
|| mapdirect: coding.direct_map(Constelacion)
```

y la variable “mapinverse” configurada como

```
|| mapinverse: coding.inverse_map(Constelacion)
```

- Se configura el bloque “Map”, en el transmisor, con “mapdirect”

```
|| Map: mapdirect
```

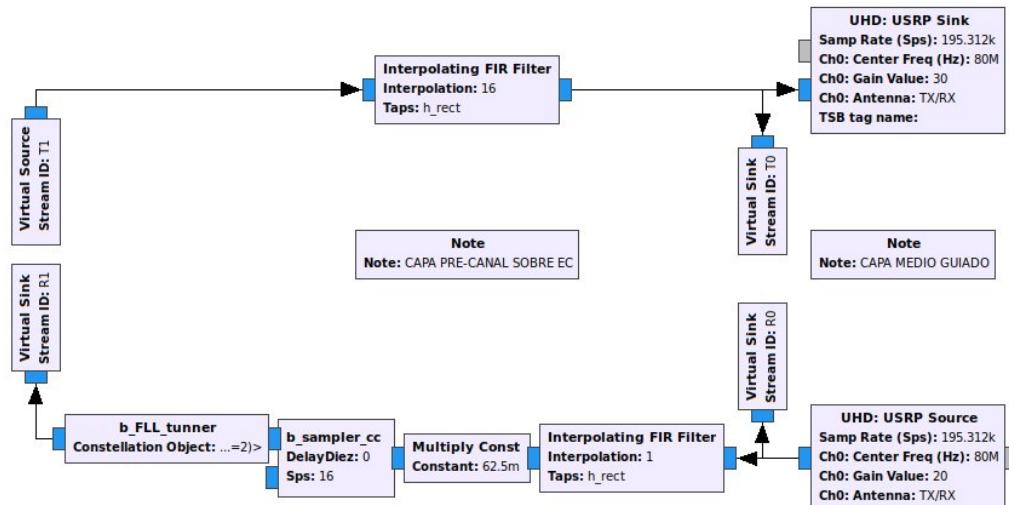
y el del receptor con “mapinverse”

```
|| Map: mapinverse
```

## 7.8. Implementación en GNU Radio de la capa Pre-Canal

Para concluir la implementación del sistema propuesto en el modelo de capas de la figura 1 lo que se propone es complementar el flujo presentado en la figura 47 con la solución que se propone para las técnicas pre-canal en la figura 49.

Figura 49: Flujo para implementar las técnicas pre-canal.



Flujo usado: RFoCable\_2.grc

Las siguientes son recomendaciones para lograr un sistema funcional:

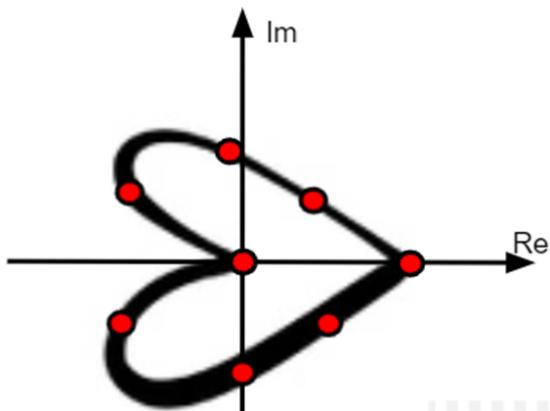
- El sistema propuesto hasta el momento es más bien de tipo pedagógico, por eso incluye varios elementos que han de ser ajustados manualmente, lo cual no debe ser problemático cuando el canal es guiado. En el siguiente capítulo, al enfrentarnos a un canal no guiado, buscaremos soluciones más automatizadas, más realistas que también pueden aplicar para el canal guiado, solo que hemos creído que no vale la pena complicar las cosas aún.
- No hemos tratado el tema de la sub capa M-PAM, pero ya lo hicimos en el capítulo anterior de modo que hemos estimado que no es necesario incluirlo. En todo caso el lector puede ir al sitio web de libro y encontrar los fluojogramas completos que incluyen toda la solución.
- Lo mismo podemos decir para la capa de aplicación, para contar con un sistema completamente funcional.

## 7.9. Problemas y preguntas de control

### Problema 1

Un organismo de seguridad trabaja en implementar un sistema de comunicaciones que sea diferente a los existentes por motivos de seguridad. Necesita revisar la viabilidad de implementar una modulación basada en una constelación que tiene la forma de la figura 50 y con los siguientes requerimientos:

Figura 50: Constelación desconocida



- Para la comunicación se tiene disponible la banda que está entre 100 MHz y 100,4 MHz
  - El Filtro transmisor tiene la respuesta es rectangular discreta con 4 pulsos
  - Para las pruebas preliminares se enviará la siguiente información: 111010010001011100
  - Los símbolos se organizan la constelación de acuerdo al código binario para los propósitos pedagógicos, si resulta necesario graficar la señal pasa bandas vamos a graficar 5000 veces menos ciclos.
1. Graficar la señal que produce el hardware de la tecnología SDR en la parte transmisora
  2. calcule la rata de bits para las condiciones dadas
  3. calcule la rata de bits que se obtiene si se usa un filtro coseno alzado con un rolloff de 0,35 para las mismas condiciones del canal

### Problema 2

Para un sistema de comunicaciones digitales basado en constelaciones, se tiene el siguiente conocimiento:

- no se tiene la capa de precanal aún
  - Se conocen las curvas de SER y aparecen en la figura 26
  - El sistema acepta hasta un símbolo perdido cada 100000 transmitidos
  - la constelación a la entrada del demodulador tiene un radio que en promedio vale 1
  - el radio de cada punto de constelación es 0,07943
1. ¿Qué tipo de modulación debe ser usada para lograr la máxima velocidad de bits posible?

2. ¿Qué velocidad de bits se alcanza si  $T_{sym} = 10^{-6}$  seg?

### Problema 3

Para un sistema de comunicaciones digitales basado en constelaciones, se tiene el siguiente conocimiento:

- Se conocen las curvas de SER y aparecen en la figura 26
- la señal portadora tiene una frecuencia 100 MHz
- En la ubicación del receptor se sabe que la señal transmitida ha tenido una pérdida de 30 dB de potencia
- El sistema acepta hasta un símbolo perdido cada 100000 transmitidos
- La relación señal a ruido en la parte transmisora es  $\text{SNR(dB)}=50$  dB
- Tenga en cuenta que, por la manera en que se han realizado las mediciones, la relación señal a ruido en el receptor es equivalente a  $\text{Pr/Pn}$  dado en dB pero también a la relación  $\text{Es/No}$  dada en dB
- Las variables usadas son:
  - Pr: Potencia de la señal recibida sin ruido
  - Pt: Potencia de la señal transmitida
  - Pn: potencia del ruido
  - $\text{SNR}=\text{Pt}/\text{Pn}$

1. Determine el tipo de modulación que el sistema deberá usar para lograr la máxima tasa de bits en las condiciones dadas
2. Calcule la distancia entre el transmisor y el receptor, suponiendo que las condiciones de propagación son ideales. Para ello debe involucrar la ecuación de Friis vista en volumen 1

### Problema 4

Para un sistema de comunicaciones digitales basado en constelaciones, se tiene el siguiente conocimiento:

- Se conocen las curvas de SER y aparecen en la figura 26.
  - la señal portadora tiene una frecuencia 40 MHz
  - La tasa de bits es 7,2 kbps
  - El sistema acepta hasta un símbolo perdido cada 10000 transmitidos
  - $\text{No}=10e-9$  Watts/Hz
1. Obtenga la Potencia reciba en Watts mínima para soportar la modulación 64QAM
  2. Determine la potencia del transmisor Pt en watts mínima a la cual debe ser transmitida la señal para lograr usar la modulación 64QAM con el receptor ubicado a una distancia de 100 metros y considerando condiciones ideales con antenas omnidireccionales

### Problema 5

En un complejo militar se desea implementar un sistema de modulación FSK diferente a los conocidos, para lo cual se tiene la tabla de verdad 7.5 para la envolvente compleja

Tabla 7.5: Tabla de Verdad

| TABLA |                                                                       |
|-------|-----------------------------------------------------------------------|
| Bits  | Comportamiento del vector rotante                                     |
| 0     | un giro en contra de las manecillas del reloj seguido de uno a favor  |
| 1     | un giro en favor de las manecillas del reloj seguido de uno en contra |

1. Grafique la envolvente compleja cuando se envía un grupo de 4 bits conocidos. Ud mismo decide los valores de esos bits.
2. Proponga una solución a nivel de bloques para producir la señal paso bandas.

**Problema 6**

Se tiene una solicitud de las fuerzas de seguridad del estado donde afirman que han detectado una señal de radio que puede ser una emisión clandestina o algo desconocido. Nos han solicitado crear una solución basada en SDR para estudiar si este tipo de señales lleva algún tipo de información que viaja en forma de cambios en la amplitud, en la fase o en la frecuencia. De manera más específica, nos han pedido crear una solución que permita medir todos los saltos de amplitud, de fase y de frecuencias que se dan en esa señal. El resultado de la medición se puede mostrar de cualquier forma, como por ejemplo: tablero de numérico, almacenarlos en un archivo. Un analizador de espectros indica que la señal está centrada en 90 MHz y esos saltos se producen cada 10 segundos.

1. grafique una solución en bloques para el medidor de las variaciones de amplitud.
2. grafique una solución en bloques para el medidor de los saltos de fase
3. grafique una solución en bloques para el medidor de las variaciones de frecuencia

**Problema 7**

Se desea experimentar un sistema de comunicaciones digitales donde la información modula a una portadora de 20 MHz de la siguiente manera: los unos son representados mediante dos saltos en la fase en la portadora así: inicia con fase  $\pi/4$ , pero a la mitad de bit cambia a  $-\pi/4$ . Los ceros en cambio son representados mediante dos saltos en la fase así: inicia con fase  $3\pi/4$ , pero a la mitad del bit cambia a  $5\pi/4$ .

1. Para el caso anterior grafique la señal que entrega la componente de software del transmisor cuando la señal de información lleva los siguientes bits: 001011
2. Si nos dan un ancho de banda disponible para la comunicación de 300 kHz, ¿cómo impacta esto la solución? Nota: lo que se espera es que el estudiante obtenga parámetros que resulten coherentes con la solución que ha propuesto, parámetros concretos como rata de bits o de símbolos entre otros
3. Grafique una solución en forma de bloques para obtener la señal de altas frecuencias que lleva la información.

**Problema 8**

Realice un diagrama de los bloques internos para un canal de radio que solo incluye el fenómeno de desvanecimiento por radiopropagación, cuando se conoce la distancia que recorre la señal y su frecuencia. Pero también se sabe que el sistema está hecho para transmitir y recibir una señal con modulación digital QPSK. Es necesario resaltar aspectos como:

- El canal propuesto debe ser implementable en GNU Radio con el fin de poder aislar este fenómeno de otros y poder comprobar soluciones a nivel del software SDR para luchar contra ese fenómeno
- El tipo de señal de entrada y salida.
- Los parámetros que el ese canal requiere, además de los mencionados que son distancia y frecuencia.
- La lógica necesaria. Es decir, si resulta necesario realizar operaciones en alguno de los bloques, debe explicar: que bloque las hace y cuales son esas operaciones.

**Problema 9**

Continuando con la pregunta anterior, realice ahora el diagrama de bloques que deben implementarse en el receptor para corregir los efectos que produce el canal propuesto en el punto 1

**Problema 10**

Tenemos un sistema de comunicación inalámbrica con modulación digital QPSK. Se ha observado que en el receptor, una vez que se obtiene la constelación, está girando de manera aleatoria en diferentes direcciones. A Ud le han dado la tarea de proponer un diagrama de bloques para un canal simulado que haga eso mismo. Es necesario resaltar aspectos como:

- El canal propuesto debe ser implementable en GNU Radio con el fin de poder aislar este fenómeno de otros y poder comprobar soluciones para luchar contra ese fenómeno
- El tipo de señal de entrada y salida.
- Los parámetros que el ese canal requiere, además de los mencionados.
- La lógica necesaria. Es decir, si resulta necesario realizar operaciones en alguno de los bloques, debe explicar: que bloque las hace y cuales son esas operaciones.

### **Problema 11**

Continuando con la pregunta anterior, realice ahora el diagrama de bloques que deben implementarse en el receptor para corregir los efectos que produce el canal propuesto en el punto anterior

### **Problema 12**

Tenemos un sistema de comunicación inalámbrica con modulación digital 8PSK. Se ha observado que en el receptor, una vez que se obtiene la PSD de la envolvente compleja se observa que la PSD sufre desplazamientos aleatorios alrededor de la frecuencia cero. A Ud le han dado la tarea de proponer un diagrama de bloques para un canal simulado que haga eso mismo. Es necesario resaltar aspectos como:

- El canal propuesto debe ser implementable en GNU Radio con el fin de poder aislar este fenómeno de otros y poder comprobar soluciones para luchar contra ese fenómeno
- El tipo de señal de entrada y salida.
- Los parámetros que el ese canal requiere, además de los mencionados.
- La lógica necesaria. Es decir, si resulta necesario realizar operaciones en alguno de los bloques, debe explicar: que bloque las hace y cuales son esas operaciones.

### **Preguntas de Falso/Verdadero**

1. La modulación 8PSK es mejor que la QPSK pero desde el punto de vista de la relación señal a ruido ya que el filtro de acoplamiento realiza un trabajo más eficiente.
  - Verdadero
  - Falso
2. Una señal pasa bandas en el tiempo que se obtiene mediante la Modulación OOK es el mismo mensaje pero con la mitad de la amplitud
  - Verdadero
  - Falso
3. El ruido blanco, como que entrega el hardware SDR en la parte receptora tiene una PSD que corta en samp\_rate/2 tiene altura de No
  - Verdadero
  - Falso
4. ¿ Son buenas opciones para luchar contra el ruido blanco el uso del filtro de acoplamiento y el demodulador?
  - Verdadero
  - Falso

### **Preguntas de Selección**

1. La energía de un bit ( $E_b$ ) se calcula haciendo:
  - $P_b * T_b$
  - $P_b * R_b$
  - $P_b * F_s$
  - $P_b * F_{max}$
2. En las curvas de SER cual es la modulación con mayor Probabilidad de error de transmisión de símbolo entre las que aparece a continuación:
  - 16-QAM
  - 16-PSK
3. ¿ Cuantas posibles fases y amplitudes puede presentar una señal 16-QAM, en paso banda en el dominio del tiempo ?
  - 12 fases y 3 amplitudes
  - 16 fases y 4 amplitudes
  - 8 fases y 2 amplitudes

4. Cuando se dice que tenemos la señal en cuadratura de una modulacion 16-QAM nos referimos a :
  - La parte real de la envolvente compleja.
  - La parte imaginaria de la envolvente compleja.
  - La magnitud de la envolvente compleja.
  - La fase de la envolvente compleja.
5. Cuando observamos que la constelación 16QAM en receptor gira en favor de las manecillas del reloj es porque la señal senoidal que llega a la antena ha sufrido un fenómeno de propagación que hace que:
  - la fase inicial sea menor a la de la senoidal transmitida
  - el número de ciclos por segundo no sea el mismo de la senoidal transmitida
  - haya resultado multiplicada por  $e^{-j\phi}$
  - haya resultado multiplicada por  $e^{-j2\pi f_d t}$
6. El codificador diferencial es una posible solución a
  - a) El problema de recepción conocido como ambigüedad de fase
  - b) La necesidad de introducir lo que se conoce como la alineación de tiempo (time alignment) para lograr conocer el inicio de la información útil
  - c) La necesidad de corregir las desviaciones de fase
  - d) La necesidad de introducir una capa de seguridad para la información

## *Capítulo 8*

### *El canal inalámbrico*

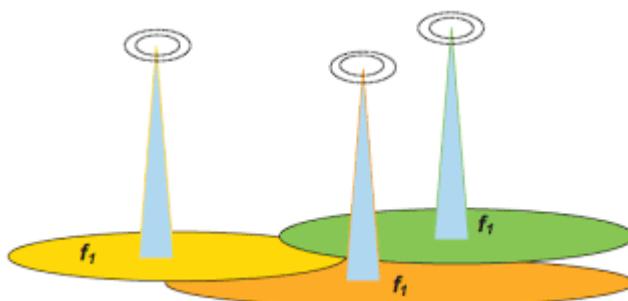
#### **8.1. Consecuencias del Fenómeno de Multitrayectoria**

##### **8.1.1. El Jitter**

La distorsión producida por el fenómeno de multitrayectoria puede ser intra-símbolo, cuando el efecto de las reflexiones es más corto que la duración de los símbolos, pero también se da el caso de que sea intersímbolo (ISI) debido al retardo que pueden tener ciertas reflexiones y a la deformación que ocurre en el dominio de las frecuencias, entonces los bits o los símbolos puede resultar superpuestos en el receptor, lo que también se conoce como Jitter.

El término Jitter es un poco más amplio, pues no es exclusivo para el fenómeno de multitrayectoria. Se denomina Jitter (término en inglés para referirse a fluctuaciones de los pulsos en el eje del tiempo) a la variabilidad temporal durante el proceso de envío-recepción de señales digitales. Una ligera desviación de la exactitud de la señal de reloj puede producirlo, pero también puede ser una de las consecuencias del fenómeno de multitrayectoria que siguen las ondas en su propagación desde el transmisor al receptor. En la televisión digital terrestre (TDT), con el standard DVB-T2, el Jitter es algo muy propio del sistema, pudiera decirse que es el precio que se paga para lograr que cada operador de TDT pueda usar una misma banda de frecuencias para cubrir todo el territorio asignado. Esto es lo que se conoce como Redes de Frecuencia única (SFN, en inglés *Single Frequency Network*). Esto representa un gran avance porque con el paradigma anterior, si se requería instalar varias estaciones de televisión para lograr ampliar la cobertura, era necesario que las estaciones vecinas usaran bandas de frecuencia diferentes. En el nuevo paradigma todas las estaciones deben radiar la misma señal en la misma banda de frecuencias, de manera que a un punto de interés llega no solo la señal de la antena más cercana sino aportes de otras antenas, con lo cual la señal se ve afectada por un Jitter que depende de la geografía y la distancia en que se encuentre el receptor, en otras palabras, depende de la configuración de la red de estaciones. El Jitter es algo no deseado, es inevitable, pero puede ser premeditado como en el caso de la TDT.

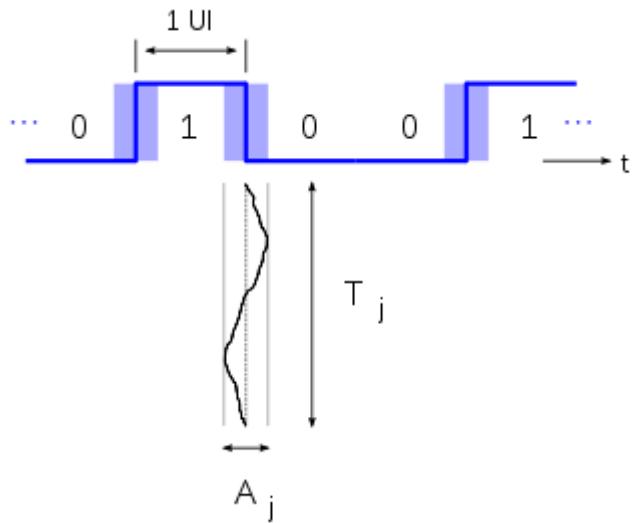
Figura 1: Red de frecuencias únicas en TDT.



En ese caso, cómo se tienen varias áreas o celdas cubiertas con la misma banda de frecuencias, radiando la misma señal, pueden llegar a interferir en ciertas zonas. Lo importante es que el Jitter no sobrepase unos valores que hagan que la señal útil sea irrecuperable. El Jitter puede estar variando, como se muestra

en la figura 2 donde  $A_j$  refleja esa variabilidad que se da cuando varía el canal, por ejemplo, porque el usuario o las fuentes reflectoras se mueven.

Figura 2: El Jitter.



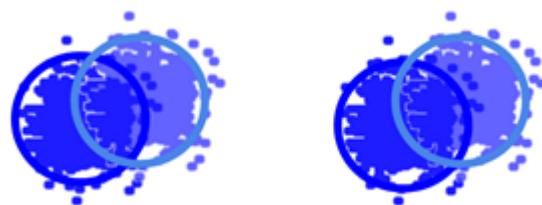
También es claro que el Jitter es una de las causas de lo que se conoce como Interferencia Intersímbolo (ISI), finalmente lo que se observa es que la energía de unos símbolos se solapan con la de otros, pero en muchos casos al Jitter se le puede dar un tratamiento especial, diferente a la ISI producida por las restricciones de ancho de banda del canal, cuando el Jitter puede ser visto como una inestabilidad en la frecuencia en que llegan los símbolos al receptor, lo cual se traduce en la necesidad de realizar un *timing or clock recovery*, pero de manera continuada o adaptativa.

### 8.1.2. Desvanecimientos lentos y rápidos. Efecto de Rayleigh

Consecuencias del desvanecimiento rápido:

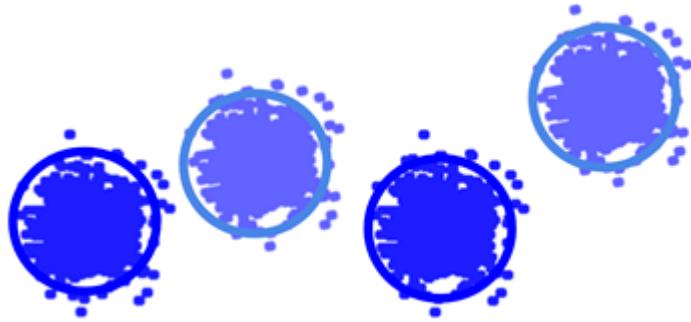
En un sistema de comunicaciones digitales basado en GNU Radio, este fenómeno se verá como una tembladera rápida de las constelaciones, pero con pocas desviaciones. En la siguiente figura hay un ejemplo con la constelación BPSK vista en dos instantes de tiempo (la que está en azul oscuro y la que está en azul más claro), donde se observa que la nube de puntos de una constelación se ha movido ligeramente de su posición. En una observación continua veremos entonces las constelaciones danzando alrededor de la posición original.

Figura 3: Consecuencias en la constelación del desvanecimiento rápido.



Consecuencias del desvanecimiento lento. En la figura 4 se presenta un ejemplo de lo que puede ocurrirle a una constelación BPSK.

Figura 4: Consecuencias en la constelación del desvanecimiento lento.

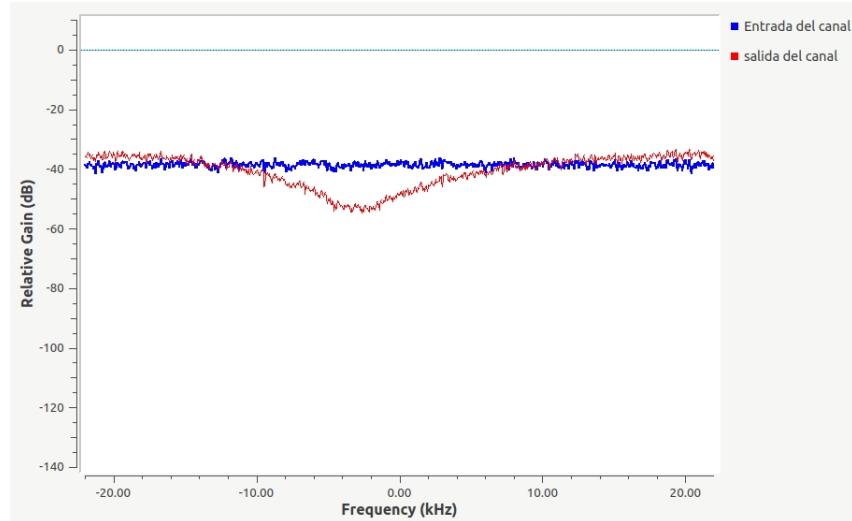


### 8.1.3. Afectación por no linealidades

La multirayectoria también trae consecuencias que son comunes en los sistemas no lineales, cuando una señal por un medio no lineal, lo cual se ve principalmente reflejado en el dominio de las frecuencias, donde algunas frecuencias o rangos de frecuencia de la señal resultan más o menos amplificadas que otras.

Para mostrar un ejemplo de esta afectación vamos al caso de SDR, con lo cual nos referiremos directamente a lo que ocurre con la Envoltura Compleja. El ejemplo se presenta en la figura 5, donde vemos que la señal que se origina en la parte transmisora, a la salida de modulador tiene una PSD que es plana en todas las frecuencias. Para lograrlo, hemos conectado el medidor de PSD a la salida de un modulador QPSK, antes del Formador de Pulses, de modo que solo se tiene una muestra por símbolo, con lo cual la señal modulada pueda ser vista, desde el punto de vista espectral, como un ruido blanco. Otro medidor de la PSD lo hemos conectado en el receptor, a la entrada del demodulador QPSK donde solo hay una muestra por símbolo. Vemos que la señal recibida, en color azul, tiene fuertes desviaciones de magnitud en el dominio de las frecuencias.

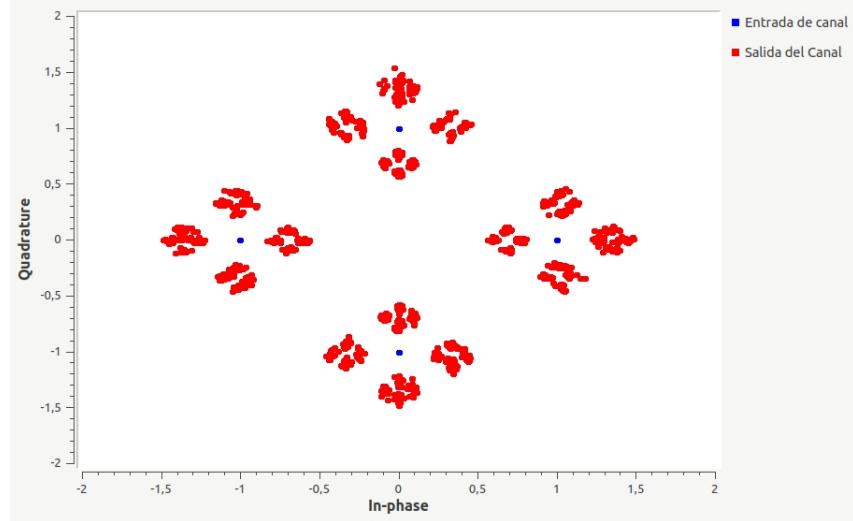
Figura 5: Variaciones espectrales debido a no linealidades del canal.



*Flujograma usado: Canal\_no\_linealidades.grc*

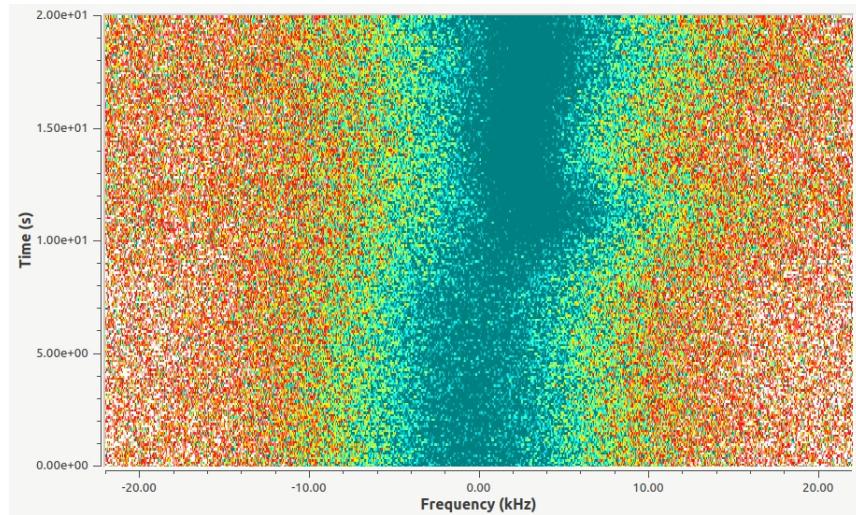
Es como si el canal estuviese jugando con las frecuencias como lo hace un disc-jockey con la música en una discoteca. En la figura 6 se muestra la constelación de la señal del mismo ejemplo con lo cual podemos apreciar que las no linealidades del canal impactan la forma de la constelación de una señal que para este ejemplo es QPSK. Como puede verse, el efecto resulta en una desfiguración que hace casi imposible reconocer directamente los símbolos.

Figura 6: QPSK impactada por no linealidades del canal.



*Flujograma usado: Canal\_no\_linealidades.grc*

Figura 7: Waterfall para una señal QPSK afectada por no linealidades variables del canal.



*Flujograma usado: Canal\_no\_linealidades.grc*

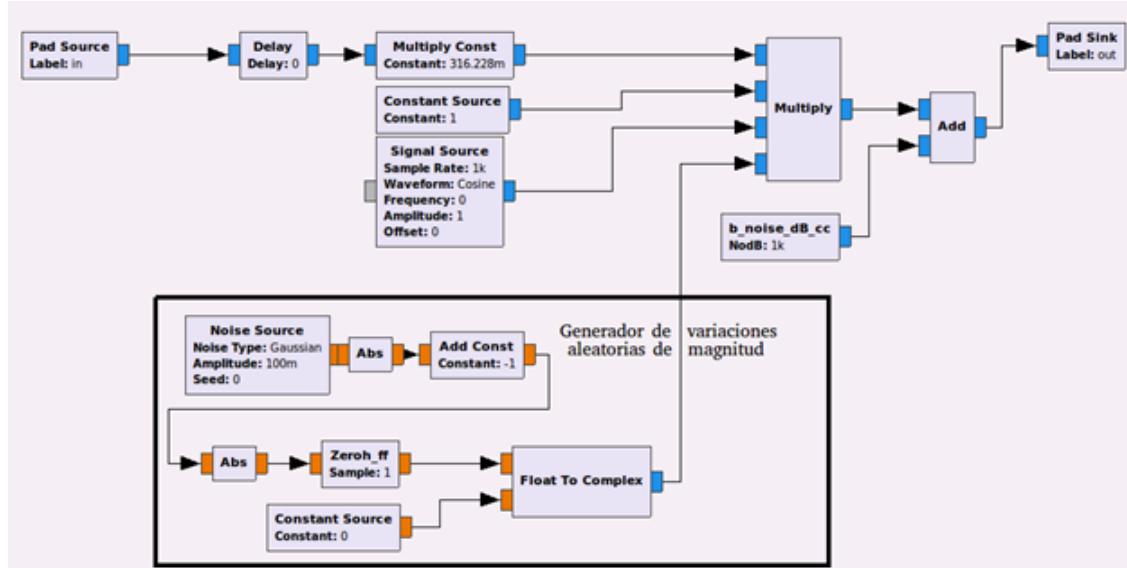
Pero lo peor de esta afectación, es que no es estática sino que varía con el tiempo, lo cual se puede apreciar mediante un analizador de espectros configurado en modo *waterfall* para mostrar como el espectro varía en el tiempo, como se muestra en la figura 7.

## 8.2. Simulación de un canal inalámbrico en GNU Radio

Simular el canal resulta tremadamente útil. Supongamos que vamos a implementar un método para corregir los estragos que produce la multitrayectoria. Para probar ese método es necesario contar con un canal en el cual se puedan aislar los fenómenos que no sean de multitrayectoria. Eso es posible cuando se cuenta con un canal simulado. Podemos ver el canal como un sistema que tiene una señal de entrada y una salida: la primera es la señal que entrega el equipo de transmisión y la segunda es la que entrega el equipo de recepción que obviamente estará afectada por los fenómenos de propagación. Es decir que, tanto el transmisor como el receptor operan con la Envoltura Compleja, resulta lógico que el canal sea

implementado en banda base para lo cual es necesario poder llevar a banda base los efectos que pueden ocurrir a la señal paso bandas. En la figura 8 se presenta el esquema de un canal realizado en GNU Radio.

Figura 8: Flujograma para el canal inalámbrico.



- **Bloques “Pad Source” y “Pad Sink”:** Son los puertos de entrada y salida del canal.
- **Bloque “Delay”:** Permite configurar el retardo que puede estar produciendo el canal a la señal recibida. Si este valor cambia en el tiempo puede simular en cierta manera la presencia de Jitter.
- **Bloque “Multiply Const”:** Permite configurar las pérdidas que el canal introduce, aunque este bloque no resulta muy necesario si lo que se requiere es lograr una cierta relación señal a ruido.
- **Bloque “Constant Source”:** Permite configurar una desviación de fase que se presenta en cualquier caso en un canal, principalmente por componentes no lineales, por retardos o por falta de sincronismo entre el transmisor y el receptor.
- **Bloque “Signal Source”:** Permite configurar una desviación de frecuencias, como la que puede presentarse por el Efecto Doppler o simplemente porque puede no coincidir exactamente la frecuencia usada en el transmisor con la usada en el receptor.
- **Bloque “Multiply”:** Reúne todas las desviaciones multiplicativas que se aplican a la señal de recibida.
- **Bloque “b-noise-dB-cc”:** Agrega ruido blanco aditivo a la señal que entrega el canal. Gracias a esto se puede configurar la relación señal a ruido que se puede producir con la propagación de la onda.
- En la parte inferior se tiene una interconexión para generar variaciones aleatorias de magnitud como las que se producen por el efecto de Rayleigh.

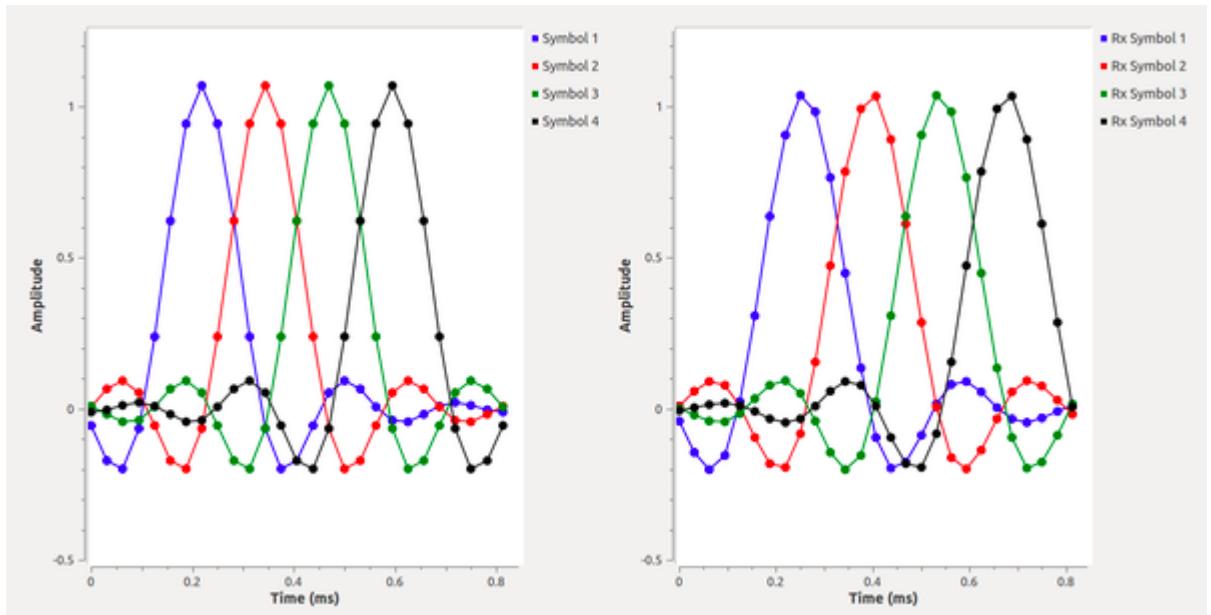
### 8.3. Técnicas para corregir los efectos del canal

#### 8.3.1. Clock Recovery o Timing

Ya sabemos que, para la recepción de un bit o un símbolo digital, usualmente existe un instante, dentro de ese bit o símbolo que es el más óptimo para analizar qué es lo que viene viajando allí. Eso se ha tratado en el tema de acoplamiento y regeneración de bits o símbolos. El problema que ahora surge consiste en lo siguiente: Los fenómenos de propagación dificultan enormemente el *timing* ya que el instante óptimo no será fijo en la señal de entrada, sino que estará variando en el tiempo. Podemos imaginar esto como una fila de hormigas que intentan llegar a su destino manteniendo una misma distancia entre ellas, pero por cuestiones imprevisibles esa distancia varía, pero es posible implementar una especie de peaje donde cada hormiga es separada para que entren a la cueva conservando una misma distancia.

La solución se conoce como *Timing*, *Timing Recovery* o *Clock Synchronization*, que básicamente consiste en lograr, de manera automática, seleccionar el mejor instante de muestreo de la señal digital recibida en un USRP, aún cuando se estén presentando variaciones de tiempo para ese instante. Esto equivale también a seleccionar la muestra donde el Diagrama de Ojo aparece más abierto. Pero la tarea va más allá de un Diagrama de Ojo, pues en la mayoría de los casos, el mejor instante para el muestreo no se encuentra ni siquiera entre las muestras de la señal discreta recibida. Es decir, en el proceso de muestreo, que se realiza previo al *timing*, es posible que el mejor instante de muestreo no sea captado por una de las muestras. Así, por ejemplo, en la siguiente figura, en la parte izquierda, el mejor instante de muestreo es parte de las muestras, pero en la parte derecha no lo es.

Figura 9: Timing.

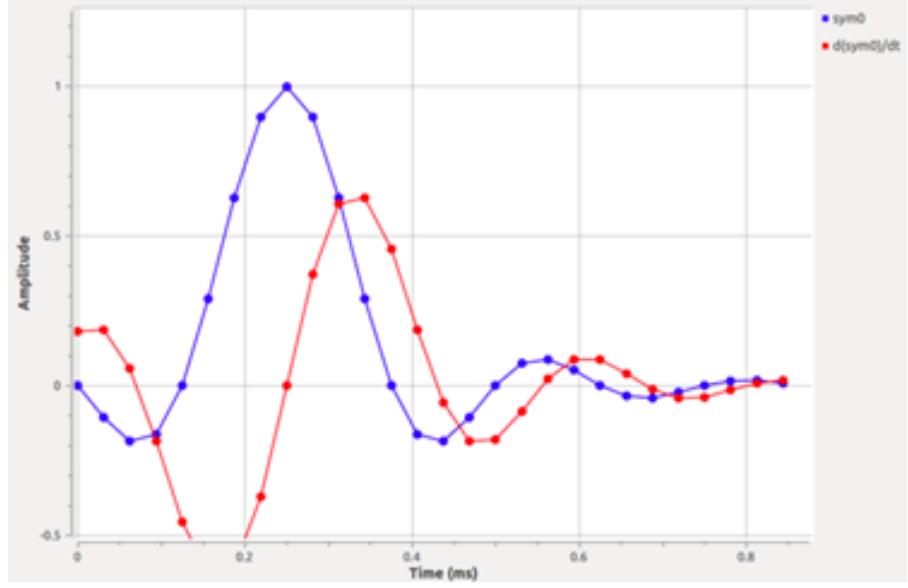


Hay varios algoritmos que podemos utilizar para recuperar el reloj en el receptor, y casi todos ellos implican algún tipo de bucle de control con retroalimentación. Otros se apoyan generalmente en una palabra conocida como un preámbulo.

El bloque **Polyphase Clock Sync**, usa una técnica de recuperación de reloj de filtrado polifásico que se describe en el libro de Fred Harris . Este bloque nos brinda dos cosas: En primer lugar, realiza la recuperación del reloj, en segundo lugar, realiza el acoplamiento entre señal transmitida y recibida para luchar con el problema de ISI.

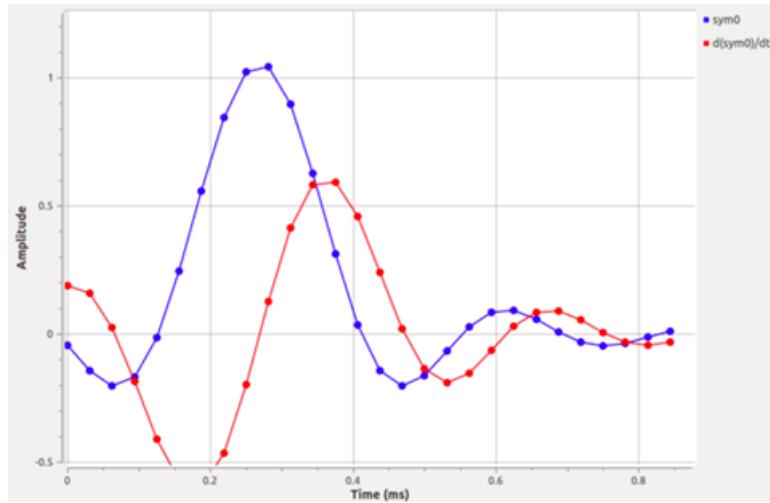
Para usar el bloque se supone que en el transmisor se ha empleado un Filtro Raíz del Coseno Alzado (Filtro RCC). El principio de funcionamiento consiste en hacer pasar la señal por otro Filtro Raíz de Coseno Alzado para realizar el acoplamiento y luego calcular la primera diferencia de la señal resultante para relacionarla con su desplazamiento del reloj. Como se muestra en la figura siguiente todo parece perfecto en este caso, pues el instante óptimo se encuentra dentro de las muestras recibidas. El filtro de diferencia  $([-1, 0, 1])$  genera el diferencial del símbolo, y como muestra la siguiente figura, la salida de este filtro en el punto de muestreo correcto es 0. Podemos entonces invertir esa afirmación y en su lugar decir que cuando la salida del filtro diferencial es 0, hemos encontrado el punto de muestreo óptimo.

Figura 10: Punto óptimo.



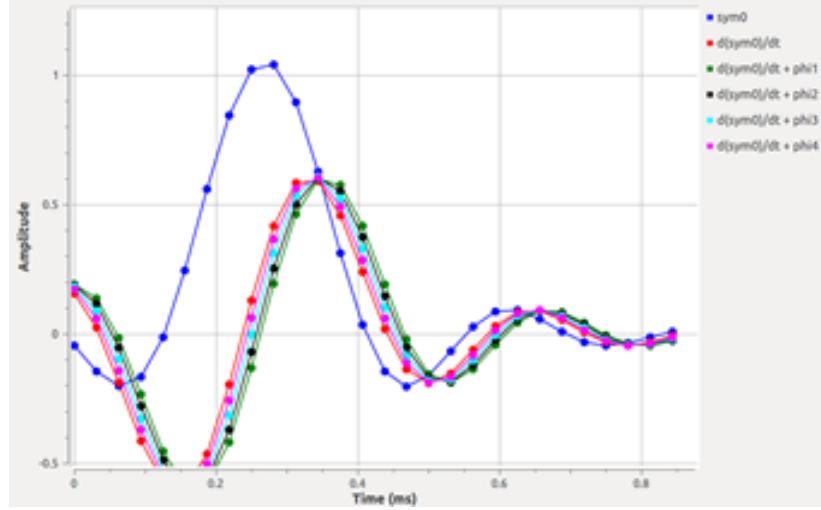
El problema es mayor cuando el punto óptimo no aparece dentro de las muestras recibidas, como se muestra en la siguiente figura.

Figura 11: Punto óptimo no aparece dentro de las muestras recibidas.



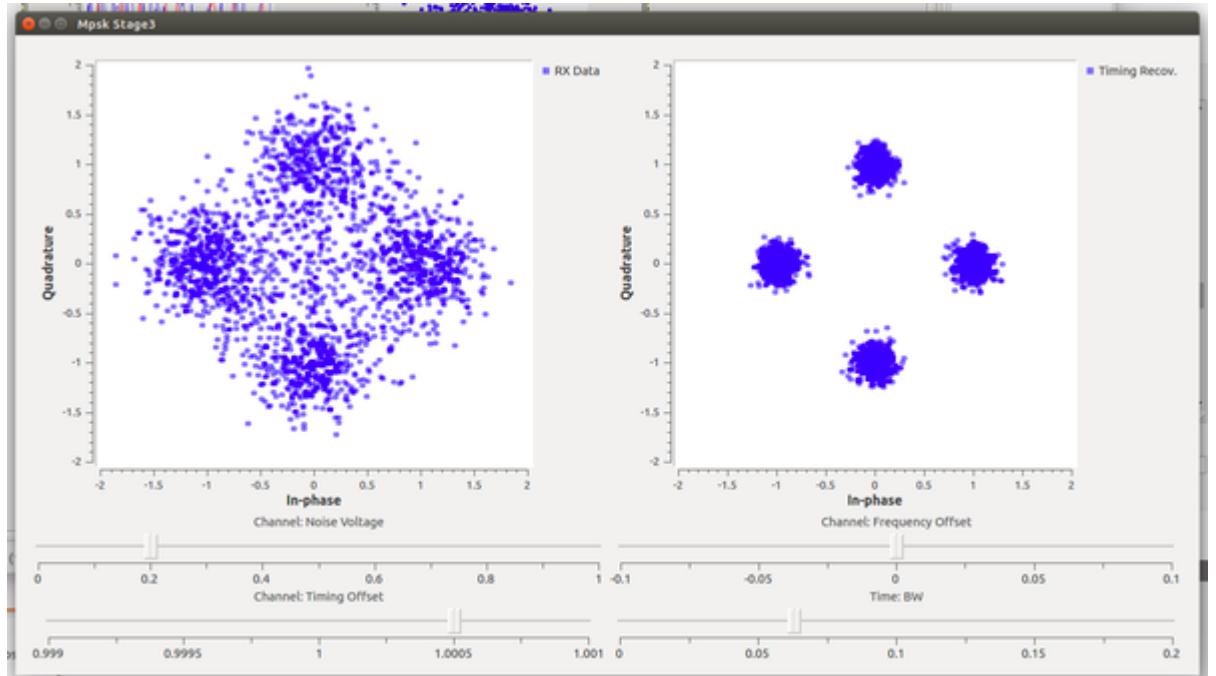
Por esta razón, el bloque **Polyphase Clock Sync** usa un número mayor de filtros cada uno con un desfase diferente. Si se tienen suficientes filtros con diferentes desfases, uno de ellos dará el valor del *timing* que buscamos. En la siguiente figura se muestra el resultado de incluir 5 filtros, lo que significa 5 desfases diferentes y podemos ver que la señal etiquetada como  $d(\text{sym}0) / dt + \phi_3$  tiene un punto de muestreo exactamente igual a 0. Esto nos indica que nuestro punto de muestreo ideal se produce con este desfase. Por lo tanto, si tomamos un filtro RRC de nuestro receptor y ajustamos su fase  $\phi_3$  (que es  $3 * 2\pi/5$ ), entonces podemos corregir la falta de sincronización y seleccionar el punto de muestreo ideal para este caso.

Figura 12: Búsqueda del punto óptimo.



El resultado del *timing* en una constelación se exemplifica en la siguiente figura en la cual se tiene en la parte izquierda la constelación de la señal a la entrada del bloque y en la parte derecha la de la salida.

Figura 13: Constelación antes y después del timing.



### 8.3.2. Alineación de tiempo

#### 8.3.2.1. Los aspectos teóricos

El problema consiste en que el instante en que comienza un Byte es desconocido para el receptor. Así, por ejemplo, en la siguiente tabla se representan los bytes y los bits que podría estar entregando un transmisor.

Tabla 8.1: Secuencia a la salida del transmisor.

| Byte | 23 |   |   |   |   |   |   |   | 56 |   |   |   |   |   |   |   | 1 |   |   |   |   |   |   |   | 99 |   |   |   |   |   |  |  |
|------|----|---|---|---|---|---|---|---|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|--|--|
| Bits | 0  | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0  | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1  | 0 | 0 | 0 | 1 | 1 |  |  |

Pero en la siguiente, debido a un retardo que es usualmente desconocido, aunque están llegando los mismos bits, son interpretados como bytes diferentes a los transmitidos.

Tabla 8.2: Secuencia interpretada de manera errónea debido al retardo introducido por el canal.

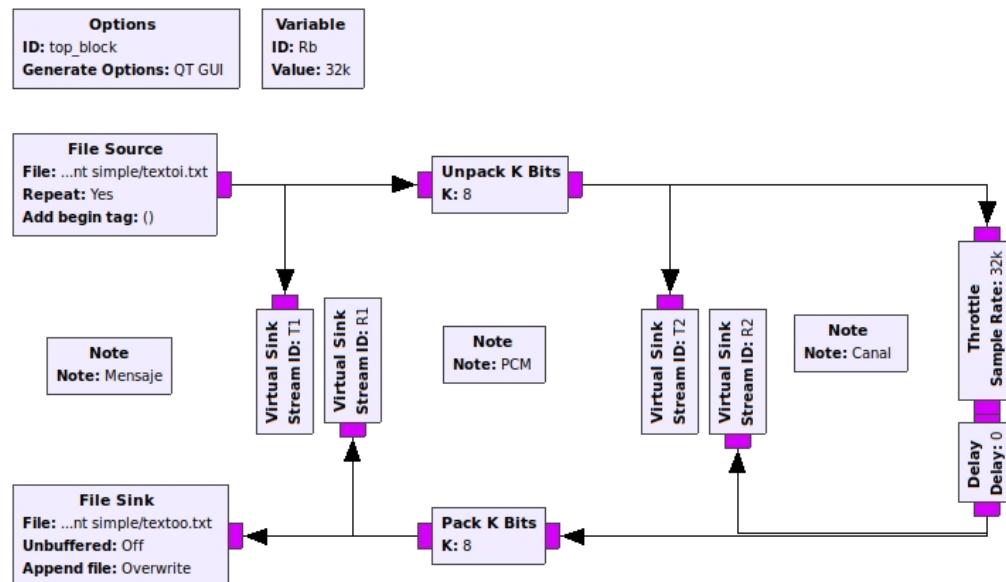
| Byte | 69 |   |   |   |   |   |   |   | 231 |   |   |   |   |   |   |   | 0 |   |   |   |   |   |   |   | 44 |   |   |   |   |   |   |  |
|------|----|---|---|---|---|---|---|---|-----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|--|
| Bits | 0  | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1   | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1 | 0 | 1 | 1 | 0 | 0 |  |

Como consecuencia, no se obtienen en el receptor los bytes transmitidos a partir de los bits que va recibiendo.

Para resolver el problema es posible usar un protocolo superior que permita marcar el comienzo de una emisión o de paquetes de emisión. Esto puede realizarse con un codificador y es lo que se conoce como Alineación de Tiempo, pero es mejor conocida por su nombre en inglés como Time Alignment.

### 8.3.2.2. Implementación en GNU Radio

Figura 14: Flujograma de un sistema con modulación PCM sin Alineación de tiempo.



Flujograma usado: *PCM\_simple.grc* Nota: es parte de la librería comdiguis

En la figura 14 se presenta un flujograma donde, por simplicidad, solo se incluye:

**La capa del mensaje** con las siguientes características:

- Se tiene una fuente (File Source) en la parte transmisora y un sumidero (File Sink) en la parte receptora.
- La fuente lee un archivo de texto y entrega una señal con valores enteros representados en Bytes, por lo tanto es una señal que varía entre -128 y 128.

**La capa PCM** con las siguientes características:

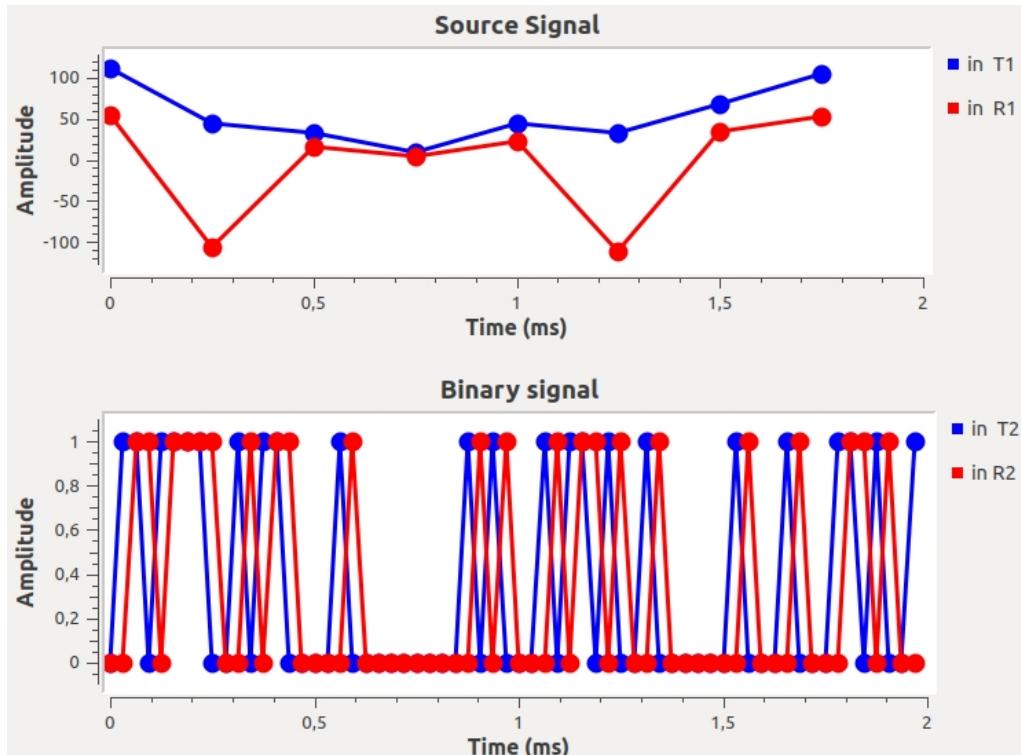
- Desempaquetar los bytes, para entregar la señal en forma de bits. El hecho de tener la señal representada ahora en un código binario es lo que se conoce como Modulación por Codificación de Pulso (PCM).

- La señal entregada se caracteriza principalmente por el parámetro  $R_b$  que es la rata de bits.

**La capa del canal** que se caracteriza en lo siguiente:

- El bloque “Delay” sirve para tener la posibilidad de introducir un retardo dado en muestras, que en este caso sería un retardo dado en bits.
- El bloque “Throttle” está allí solo debido a que así el programa que resulte del flujograma va a correr en tiempo real respetando la frecuencia de muestreo que se le programe a ese bloque. Sin este bloque ese programa va a correr con la máxima velocidad que le permita el sistema del computador.

Figura 15: Observación de las señales cuando no se presentan retardos en el canal.



Flujograma usado: PCM\_simple.grc Nota: es parte de la librería comdiguis

En la figura 15 se tienen los resultados cuando el canal presenta un pequeño retardo. Lo que podemos observar es que la señal binaria en el transmisor, en el punto T2, es la misma del receptor, en el punto R2, solo que aparece retrasada en lo que dura un bit. No ocurre lo mismo con la señal en la salida de la fuente, en el punto T1, con respecto a la que entra al sumidero, en el punto R1, lo cual es precisamente el problema que deseamos resolver con la Alineación de tiempo.

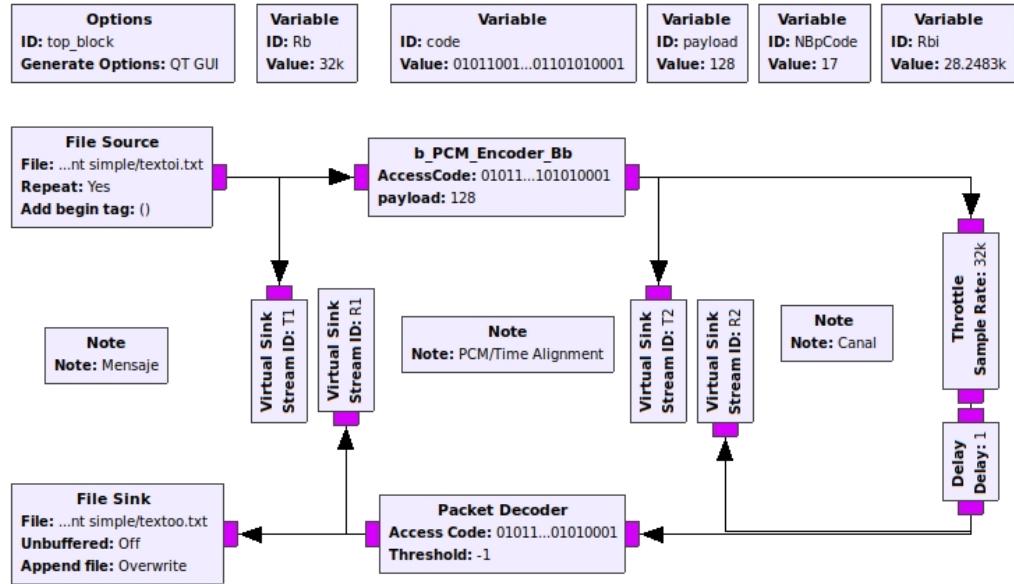
En la figura 16 se presenta el mismo flujograma pero con Alineación de tiempo. En esta solución podemos observar las siguientes novedades:

- En la parte transmisora se ha cambiado el bloque encargado de la modulación PCM por el bloque “b\_PCM\_Encoder\_Bb”.
- En la parte receptora tenemos el bloque “Packet Decoder”.

Sobre el bloque “b\_PCM\_Encoder\_Bb” tenemos las siguientes observaciones:

- Es un bloque de la librería “Lib\_Bloques\_Comdiguis” que acompaña el presente libro.
- Aplica la modulación PCM para convertir la señal entrante en binaria. En otras palabras, a cada byte entrante se le extraen los bits.
- Pero adicionalmente, organiza los bits en tramas, en una especie de tren que tiene un encabezado y una carga útil.

Figura 16: Flujograma de un sistema con modulación PCM y con Alineación de tiempo.



- El encabezado está dado por un código binario que se entrega mediante el parámetro *AccessCode* y debe ser un número de bits que sea múltiplo de 8, de modo que su tamaño en bytes *Bacode* sea un número entero.

$$Bacode = \text{len}(AccessCode)/8$$

Donde  $\text{len}(X)$  es la función que calcula el tamaño del vector  $X$ .

- La carga útil tiene un tamaño en bytes que entrega mediante el parámetro “*Bpayload*”.
- Se usa también un preámbulo de tamaño *Bprea* que, en el caso del bloque dado, son 10 bytes por defecto.
- El tamaño del encabezado *BpCode* hace que la tasa de bits de información útil a la entrada del bloque *R<sub>bi</sub>* sea menor que la tasa de bits a la salida *R<sub>b</sub>*.
- Resulta importante poder calcular *R<sub>bi</sub>* a partir de *R<sub>b</sub>*. Para ello es necesario calcular el número de bytes en el código así:

$$BpCode = Bacode + Bprea \quad (8.1)$$

*Bprea* es el tamaño del preámbulo dado en bytes, como se muestra en la figura 17. De manera que:

$$R_{bi} = \frac{R_b}{1 + \frac{BpCode}{Bpayload}} \quad (8.2)$$

Sobre el bloque “Packet Decoder” tenemos las siguientes observaciones:

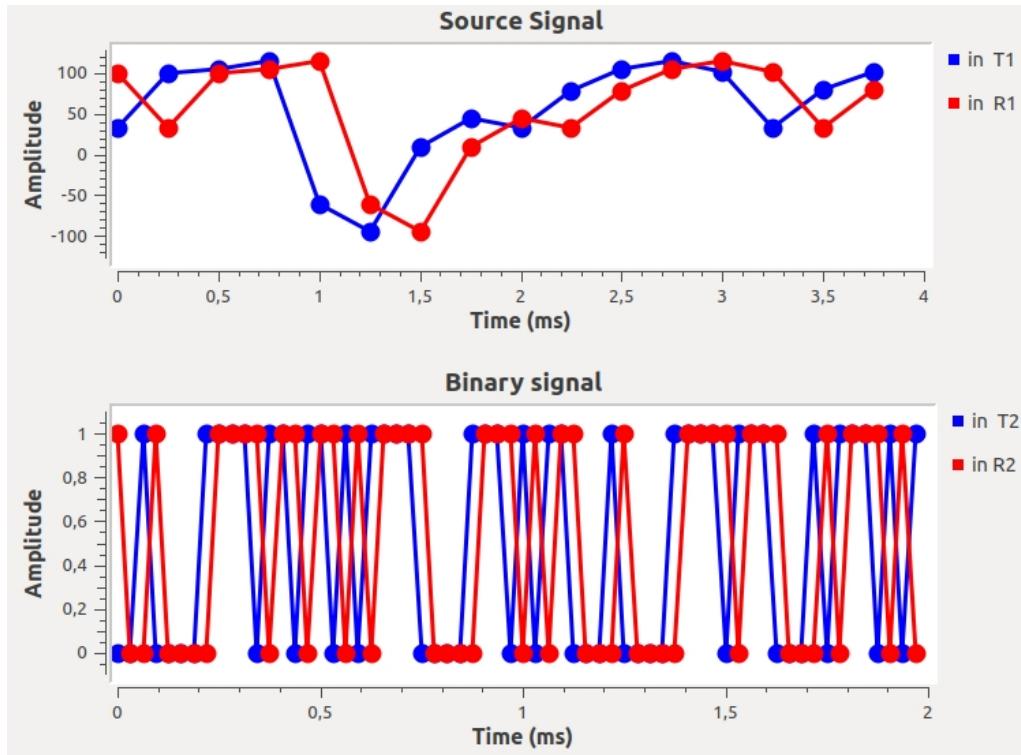
- Vigila y detecta el encabezado de la trama.
- Si no lo encuentra no entrega nada como salida.
- Si lo encuentra lo retira para dejar pasar solo la información útil.

En la figura 18 puede observar el resultado de aplicar Alineación de tiempo, como podemos ver, a pesar de que la señal binaria en el punto R2 esté atrasada con respecto al punto T2, la señal que llega al sumidero, en el receptor, en el punto R1, conserva la misma forma que en el punto T1.

Figura 17: Trama de Alineación de tiempo.



Figura 18: Observación de las señales cuando se aplica Alineación de Tiempo.

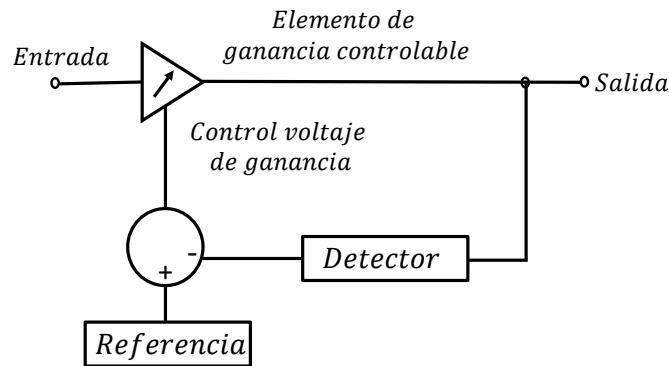


Flujograma usado: `PCM_time_alignment_simple.grc` Nota: es parte de la librería comdiguis

### 8.3.3. El Control Automático de la Ganancia

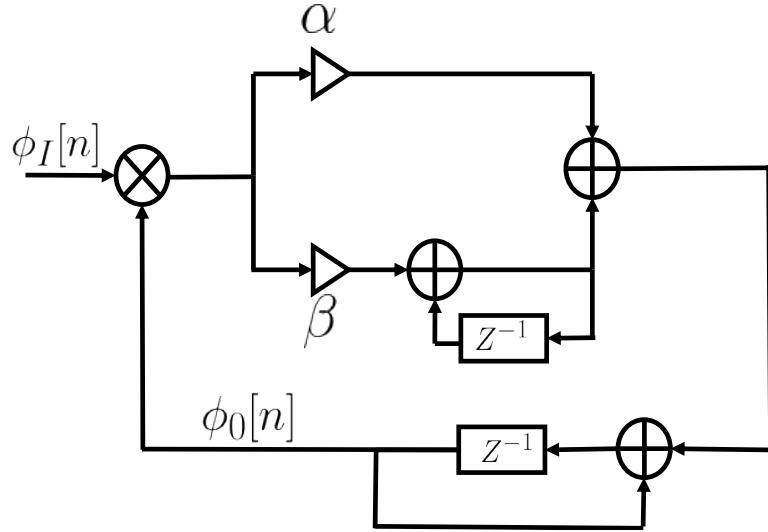
De acuerdo a lo visto, es inevitable que la señal a la salida de USRP receptor esté variando su magnitud de manera impredecible. Un AGC puede permitir estabilizar ese parámetro de una manera aceptable para su observación y procesamiento. El sistema puede ser implementado basado en el sistema retroalimentado de bucle cerrado que aparece en la figura 19.

Figura 19: Elemento de área donde puede localizarse el receptor.



Básicamente se cuenta con una referencia, que es el nivel deseado de señal. El detector está midiendo permanentemente el nivel de señal a la salida. La diferencia de los anteriores valores permite regular la ganancia del amplificador. Una descripción más detallada se encuentra en el trabajo de [7] quien desarrolló un AGC en GNU Radio que se resume en la figura 20.

Figura 20: AGC según Tom Rondau.

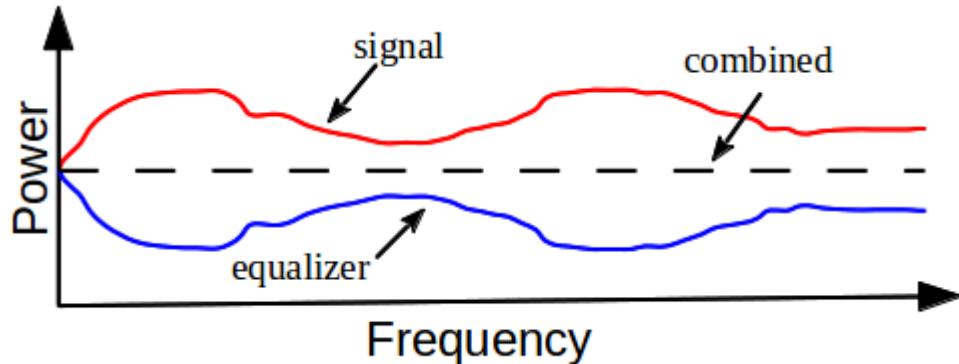


El sistema permite realizar un seguimiento continuo del nivel de señal, pero es claro que no contribuye a mejorar la relación señal a ruido. En realidad, todo radio receptor contiene un AGC.

#### 8.3.4. El ecualizador

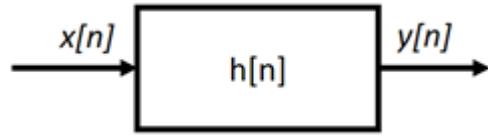
En muchos casos la señal que pasa por un canal sufre diversas alteraciones en función de la frecuencia. La ecualización del canal busca determinar cuáles son esas alteraciones, intentando descubrir la respuesta en frecuencias del canal para luego sintonizar un filtro con la respuesta en frecuencia inversa para corregir el efecto del canal.

Figura 21: Variaciones espectrales del canal (rojo) versus respuesta en frecuencias del ecualizador.



- **Filtro digital:** El principal componente de un ecualizador es un simple filtro digital, que no es otra cosa que un sistema Lineal e Invariante en el Tiempo (LIT), bueno, al menos se comporta como un sistema LIT por momentos. Pues se supone que un sistema LIT tiene una respuesta al impulso única, nosotros en cambio la iremos variando en el tiempo, en función de las necesidades de ecualización que se presenten por el comportamiento del canal. En todo caso si podemos decir que un sistema que tiene una respuesta al impulso  $h[n]$ , una entrada  $x[n]$  y una salida  $y[n]$ , como el de la siguiente figura.

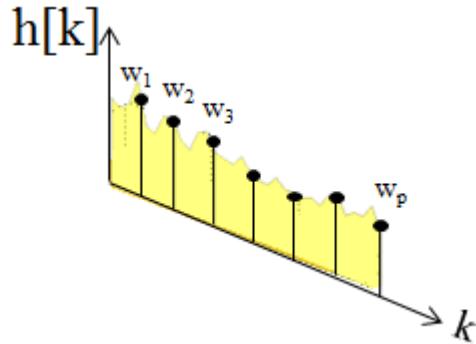
Figura 22: Sistema LIT.



$$y[n] = x[n] * h[n] \quad (8.3)$$

En los sistemas reales, como es el caso de GNU Radio,  $h[n]$  es un conjunto de valores  $W_k$  cada uno de los cuales representa el valor de amplitud de la muestra  $k$  de  $h[n]$ , como se muestra en la siguiente figura.

Figura 23: Respuesta al impulso de un ecualizador.



De modo que podemos, aplicando el concepto de convolución, decir que.

$$y[n] = w[n] * x[n] = \sum_{k=1}^p w_k x[n - k] \quad (8.4)$$

En GNU Radio se tienen bloques que juegan el papel de filtro digital, para lo cual hay que configurarle un vector  $w$  que contiene los pesos  $W_k$ . Esos pesos se conocen en GNU Radio como taps.

- **El ecualizador fijo:** Un filtro como el anterior, con una respuesta al impulso adecuada, podría resolver la necesidad de ecualización de un canal que es estable en el tiempo.

Figura 24: Proceso de entrenamiento del ecualizador.

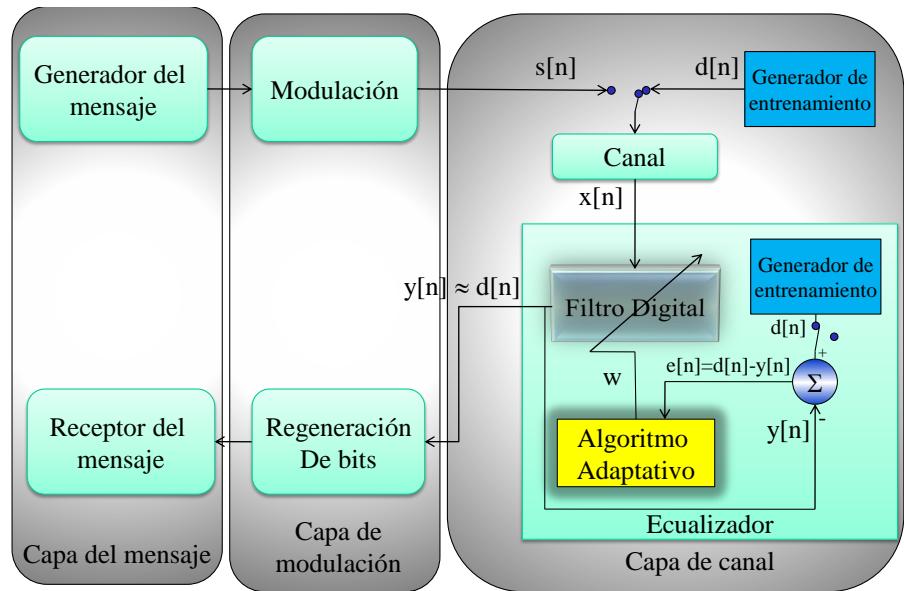
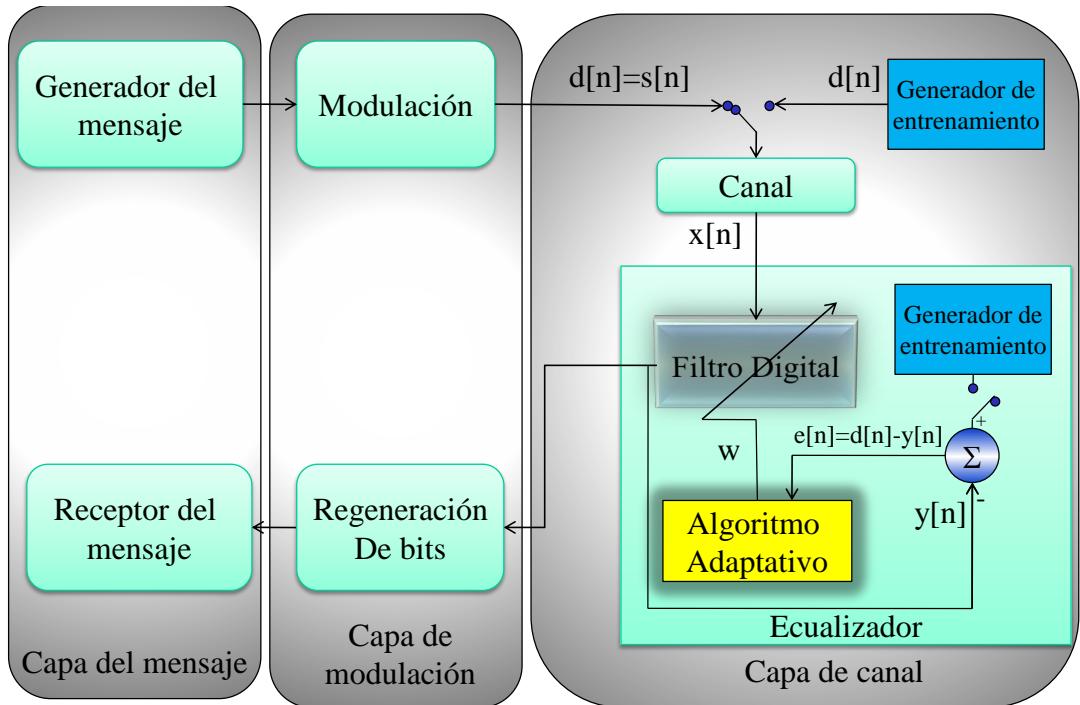


Figura 25: El ecualizador funcionando con pesos obtenidos durante el entrenamiento.



- **El ecualizador adaptativo:** Cuando el canal tiene cambios muy fuertes en el tiempo, la única solución puede ser un ecualizador que se adapte a esos cambios.

mejorar fig 26  
porque no pueden ser los bits regenerados los que se envían al ecualizador, sino símbolos regenerados

Figura 26: Ecualización adaptativo.

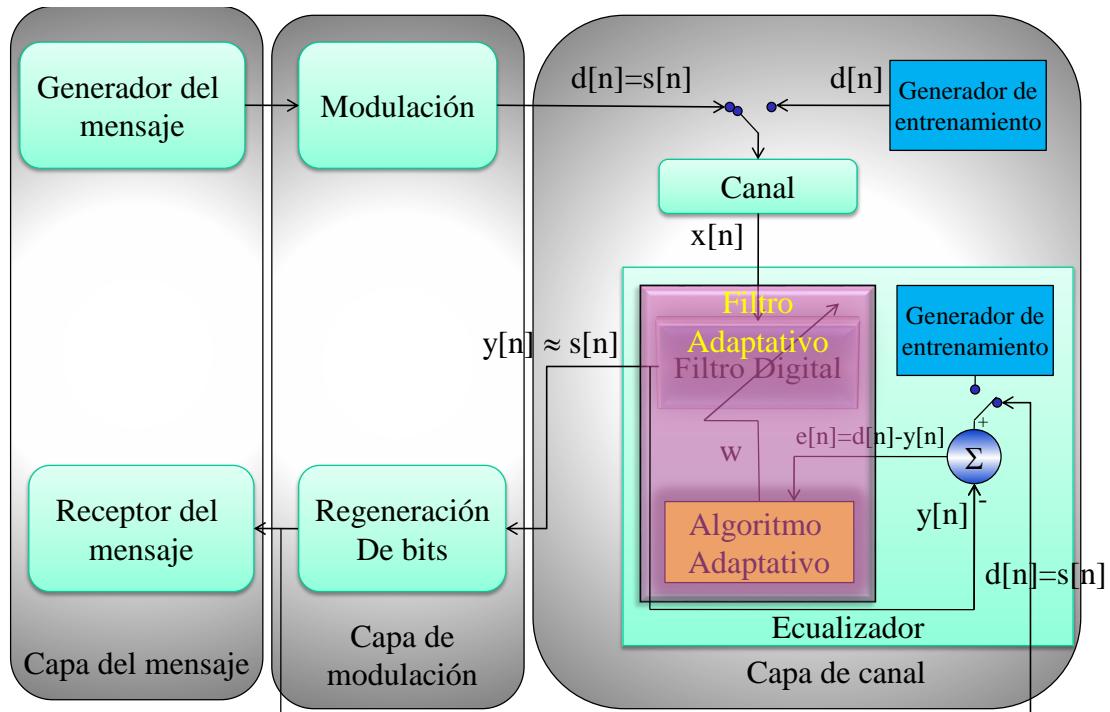
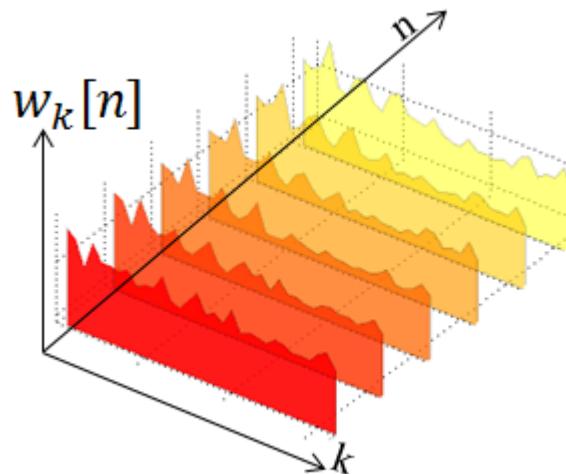


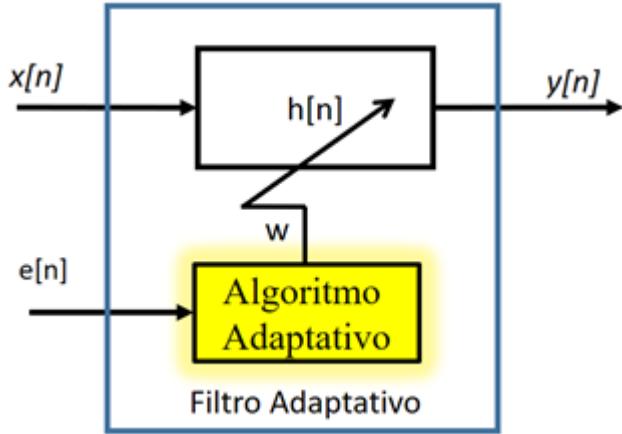
Figura 27: Respuesta variante al Impulso de un ecualizador.



- **Filtro basado en los mínimos cuadrados medios (LMS):** Se refiere en realizar a la búsqueda del mínimo error cuadrático medio que el sistema pueda producir. Filtro de *Wiener y Hopf* o Filtro LMS.

Los problemas a resolver son dos: El de obtener esa respuesta al impulso adecuadas y el de recalcularla cada vez que el canal cambie de estado. Esto se logra, si el filtro digital es complementado con un Algoritmo que estudie lo que se espera de él para recalcular los pesos o taps del filtro. Es necesario informar a este algoritmo lo que de él se espera y esto se puede realizar mediante una señal de error.

Figura 28: Filtro adaptativo.



Ahora es tiempo de preguntarnos qué operaciones realiza el Algoritmo Adaptativo. El algoritmo deberá seleccionar valores para el vector  $w$  y analizar cómo ellos van influyendo en el error  $e[n]$  hasta lograr que la media cuadrática del error sea la mínima posible. De esta manera, el índice de desempeño elegido es el valor cuadrático medio.

$$J = E[(d[n] - y[n])^2] \quad (8.5)$$

El criterio para encontrar los pesos consiste en lograr que ese índice de desempeño sea el mínimo, con lo cual llegamos a aplicar el criterio de aproximación conocido como el mínimo de la media cuadrada o LMS (en inglés *Least Mean Square*).

$$J = J_{min} \quad (8.6)$$

Como  $d[n]$  es la señal que se desea a la salida del filtro, entonces puede decirse que se desea que:

$$y[n] = \sum_{k=1}^p w_k x[n-k] \quad (8.7)$$

Por lo tanto se desea que:

$$J = E \left( [d[n] - \sum_{k=1}^p w_k x[n-k]] \right)^2 \quad (8.8)$$

Ahora debemos tener en cuenta la siguiente propiedad de las series:

$$\sum_{k=1}^2 a_k \sum_{j=1}^2 a_j = \sum_{k=1}^2 \sum_{j=1}^2 a_k a_j \quad (8.9)$$

Entonces obtenemos.

$$J = E[d^2[n]] - 2 \sum_{k=1}^p w_k E[d[n]x[n-k]] + \sum_{j=1}^p \sum_{k=1}^p w_j w_k E[x[n-j]x[n-k]] \quad (8.10)$$

Se pueden realizar algunas simplificaciones si se tiene en cuenta que:

- La señal  $x(t)$  es la función muestra de un proceso estacionario  $X(t)$  y su versión muestrada  $x[n]$  tiene media.

$$\mu_X[n] = \mu_X = 0 = E[x[n]] \quad (8.11)$$

- De lo anterior se deduce que la varianza.

$$E[(x[n] - \mu_x)^2] = E[x^2[n]] - \mu_x^2 = \sigma_x^2 \quad (8.12)$$

- Sabiendo que la función de autocorrelación está definida como:

$$E[x[n]x[n-k]] = R_X[k] \quad (8.13)$$

- Y que la función de correlación entre  $d[n]$  y  $x[n]$  es:

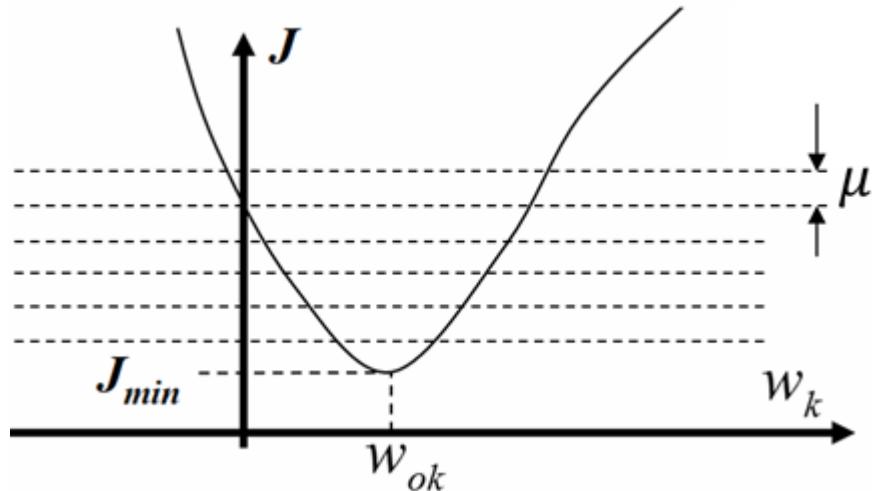
$$E[d[n]x[n-k]] = R_{DX}[k] \quad (8.14)$$

Se deduce que:

$$J = \sigma_x^2 - 2 \sum_{k=1}^p w_k R_{DX}[k] + \sum_{j=1}^p \sum_{k=1}^p w_j w_k R_X[k-j] \quad (8.15)$$

Lo que resta es identificar qué pasa con los pesos  $w$  cuando  $J = J_{min}$ , con lo cual podríamos llegar a despejar los pesos óptimos  $w_o$ , como se muestra en la siguiente figura.

Figura 29: El mínimo de la media cuadrática.



El mínimo error cuadrático medio  $J_{min}$  se presenta cuando el gradiente (la pendiente) del error cuadrático medio  $J$  es cero. De modo que si nos interesa hallar  $W_k = W_{0k}$  para ese caso, debemos despejar  $W_k$  cuando  $\frac{\delta J}{\delta W_k} = 0$ .

$$g_k = \frac{\delta J}{\delta W_k} = -2R_{DX}[k] + 2 \sum_{j=1}^p w_j R_X[k-j] \quad (8.16)$$

Igualando a cero el vector gradiente  $g_k$ , se obtiene la expresión de optimización de los *mínimos cuadrados de Wiener-Hopf*, también conocida como *ecuaciones de Wiener Hopf*:

$$R_{DX}[k] = \sum_{j=1}^p w_j R_X[k-j] \quad (8.17)$$

Veamos el conjunto de todos los valores que intervienen en esta ecuación. Ellos se pueden organizar en forma matricial.

$$r_{DX} = \begin{bmatrix} R_{DX}[1] \\ R_{DX}[2] \\ \vdots \\ R_{DX}[p] \end{bmatrix}; w_0 = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_p \end{bmatrix}; \quad (8.18)$$

$$R_X = \begin{bmatrix} R_X[0] & R_X[1] & \dots & R_X[p-1] \\ R_X[1] & R_X[0] & \dots & R_X[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ R_X[p-1] & R_X[p-2] & \dots & R_X[0] \end{bmatrix} \quad (8.19)$$

$R_X$  es conocida como la matriz simétrica de Toeplitz. Representando  $R_X$  y  $w_0$  de la forma vectorial siguiente:

$$w_0 = [w_1, w_2, \dots, w_p]^T r_{DX} = [R_{DX}[1], R_{DX}[2], \dots, R_{DX}[p]]^T \quad (8.20)$$

Se obtiene la forma matricial de la expresión de optimización.

$$r_{DX} = w_o R_X \quad (8.21)$$

Facilitando el despeje de los pesos así:

$$w_o = r_{DX} R_X^{-1} \quad (8.22)$$

$R_X^{-1}$  es la inversa de  $R_X$  y se supone que existe pues  $R_X$  no es singular.

- **Filtro Recursivo basado en los mínimos cuadrados medios (RLS)**

Se trata del mismo Filtro LMS solo que se usa un método conocido como el descenso más pronunciado (*deepest descent*) para obtener una ecuación donde los pesos pueden ser hallados de manera recurrente, realizando un ligero ajuste a los pesos previamente calculados.

$$w_k[n+1] = w_k[n] + \mu x[n-k] e[n], k = 0, 1, \dots, N \quad (8.23)$$

$\mu$  Es un paso que el algoritmo usa para ir buscando el valor  $J_{min\ ideal}$  a partir de un valor  $J_{min\ ya\ calculado}$ . Este es quizás el método más utilizado en ecualización adaptativa. A continuación algunos apuntes sobre experiencias de trabajo con este tipo de filtros:

- En cuanto mayor sea el parámetro del tamaño del escalón  $\mu$ , tanto más rápida la capacidad de rastreo del ecualizador adaptable.
- Un parámetro grande de  $\mu$  quizás resulte en un error cuadrático medio de exceso inaceptablemente alto, es decir, excede exageradamente a  $J_{min}$ .
- Entre más pequeño sea  $\mu$  encontramos que después de un gran número de interacciones el comportamiento del algoritmo de mínimos cuadrados es casi similar al del *Algoritmo de Descenso Pronunciado*, que utiliza el gradiente real en vez de la estimación de error para el cálculo de los pesos del filtro.
- La elección en la práctica de un valor adecuado para  $\mu$  implica establecer un compromiso entre el rastreo rápido y la reducción del error cuadrático medio de exceso.
- El bloque LMS de Simulink tiene una opción para usar una forma normalizada del algoritmo LMS para que resulte limitada así:  $0 < \mu < 2$ .

Proceso que realiza el filtro:

1. En cada paso se halla un nuevo  $J_{min\ ya\ calculado} = J_{min\ calculado} + \mu$ .
2. Luego analiza la diferencia con el  $J_{min\ ideal}$ .
3. Si no alcanza este  $J_{min-ideal}$ , entonces vuelve al punto 1.

Para el caso, cuando no se usa el algoritmo normalizado, el material “Ecualización adaptativa de un canal digital ” del profesor Lorenzo Díaz [8] es el más explícito encontrado hasta ahora. En esa obra, el profesor Lorenzo afirma que es muy difícil encontrar el valor óptimo de  $\mu$  en tiempo real, pero que hay una manera aproximada y rápida:

$$0 < \mu < \frac{2}{3LP_X} \quad (8.24)$$

Donde L: Es el tamaño del filtro, es decir el tamaño del vector de pesos y  $P_x$ . La potencia media de la señal.

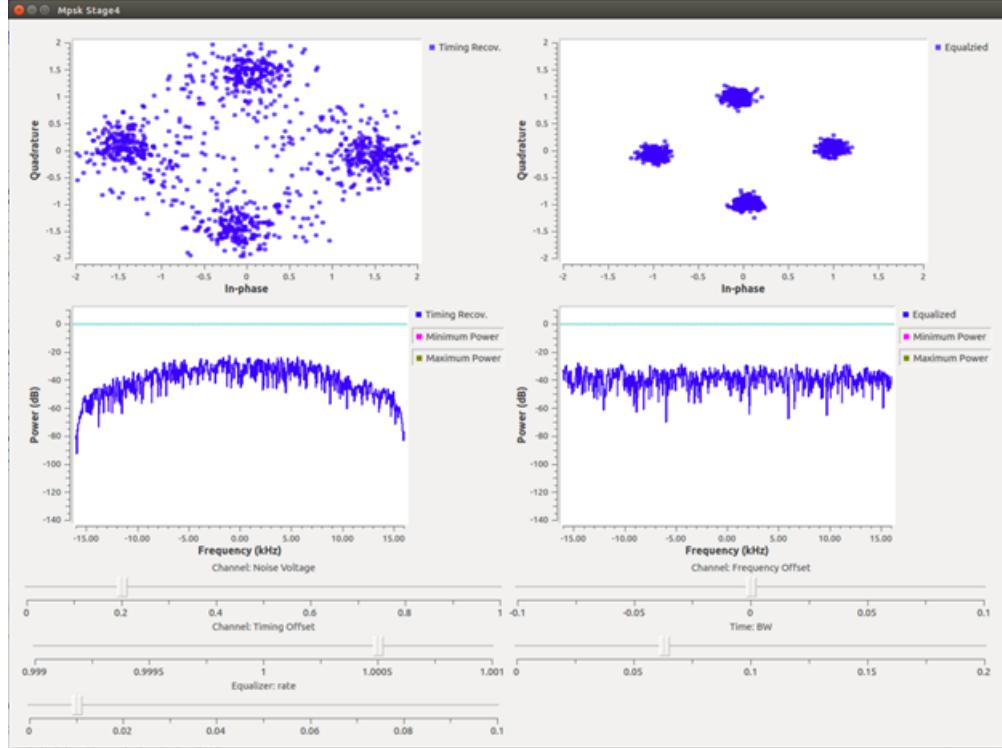
### 8.3.5. La ecualización en GNU Radio

GNU Radio viene con dos ecualizadores fácilmente utilizables. El ecualizador de *CMA Equalizer* y el *LMS DD Equalizer*. El CMA, o Algoritmo de Módulo Constante, es un ecualizador ciego, porque no necesita mucha información sobre la señal originalmente transmitida, pero sólo funciona con señales que tienen una amplitud constante, o módulo, como es el caso de la modulación M-PSK ya que los puntos de constelación están en un círculo. El algoritmo de CMA acepta el número de taps a utilizar en el ecualizador, que se basará en una combinación de una conjectura educada, las mejores prácticas conocidas y quizás algún conocimiento real del canal en sí.

La siguiente figura ha sido obtenida al conectar un bloque **CMA Equalizar** de 11 taps a la salida de un **Polyphase Clock Sync**. La constelación y el espectro a la izquierda corresponden a la entrada del ecualizador, las de la derecha a la salida.

El “*LMS DD Equalizer*” usa un filtro de predicción basado en el principio del mínimo error medio cuadrático (en inglés *Least Mean Square*) con decisión directa (DD) con respecto a una información que se conoce sobre la señal originalmente transmitida. En el caso del bloque usado en GNU Radio, lo que se conoce de la señal originalmente transmitida es la forma de la constelación. El filtro va recalculando los taps del filtro vigilando que la salida corresponda con la constelación que se tiene como modelo. Puede fallar cuando SNR es demasiado alta, arruinando el desempeño. Este bloque además tiene mayor complejidad computacional.

Figura 30: Resultado del la Ecualización. A la izquierda sin ecualización. A la derecha con ecualización.



*Flujograma implementado: Canal\_no\_linealidades.grc*

## 8.4. Problemas y preguntas de control

### Preguntas de Selección

1. Entre las siguientes soluciones la que mejor ayuda a luchar contra el fenómeno de desvanecimiento por propagación es:
  - El ecualizador
  - El automatic Gain Control
  - El filtro Receptor
  - El PLL
2. Entre las siguientes soluciones la que mejor ayuda a luchar contra las no linealidades del canal es:
  - El ecualizador
  - El automatic Gain Control
  - El filtro Receptor
  - PLL
3. Tenemos un receptor que consta, a nivel de software, de un filtro de acoplamiento seguido de un muestreador manual y un demodulador 16QAM. Hemos logrado hacer que la señal entrante solo esté afectada por el Jitter típico del fenómeno de multitrayectoria y muy poco ruido. El siguiente es el comportamiento de la constelación vista en la entrada del demodulador:
  - La constelación muestra variaciones en la dispersión de los puntos, mostrando que pasa por momentos de nitidez en su forma y momentos en que se dispersa tanto que se pierde la forma. Si hacemos variar el timing cuando se pierde la nitidez se logra recuperar nuevamente esa nitidez

- La constelación se atenúa por momentos afectando con ello la relación señal a ruido. Al variar el timing se logran estabilizar esas variaciones y la constelación toma una forma nítida que se mantiene mientras el jitter se mantenga constante.
  - La constelación se nota temblando con movimientos angulares, pero se estabiliza si la señal pasa previamente por un sistema retroalimentado conocido como PLL
  - La constelación permanece todo el tiempo dispersa pero se logra recuperar haciendo pasar la señal por un sistema retroalimentado de ecualización antes de entrar al muestreador.
4. Tenemos un receptor que consta, a nivel de software, de un filtro de acoplamiento seguido de un muestreador manual y un demodulador 16QAM. Hemos logrado hacer que la señal entrante solo esté afectada por Fast Fading y Slow Fading típico del fenómeno de multitrayectoria y muy poco ruido. El siguiente es el comportamiento de la constelación vista en la entrada del demodulador:
- La constelación se atenúa por momentos afectando con ello la relación señal a ruido. Al variar adecuadamente la amplificación se estabilizan esas atenuaciones aunque no mejora la relación señal a ruido
  - La constelación se nota temblando con movimientos angulares, pero se estabiliza si la señal pasa previamente por un sistema retroalimentado conocido como PLL
  - La constelación permanece todo el tiempo dispersa pero se logra recuperar haciendo pasar la señal por un sistema retroalimentado de ecualización antes de entrar al muestreador.
  - La constelación se observa en forma de líneas continuas
  - La constelación se dispersa por momentos afectando con ello la relación señal a ruido. Al variar el timing se logran estabilizar esa dispersión y toma la constelación su forma
5. El bloque usado en las prácticas realizadas en clase, con GNU Radio, para realizar el trabajo conocido como Clock Recovery actúa como:
- Filtro de acoplamiento y a la vez corrige el Jitter y hace el muestreo (timing)
  - una solución que solo corrige el Jitter y hace el muestreo (timing)
  - permite corregir la señal portadora recibida de manera que coincida en fase y frecuencia con la emitida
  - Una solución que permite recuperar el reloj, de manera que altera la velocidad de los símbolos para que coincida con la velocidad de los símbolos transmitidos

# *Capítulo 9*

## *Comunicaciones multiusuario*

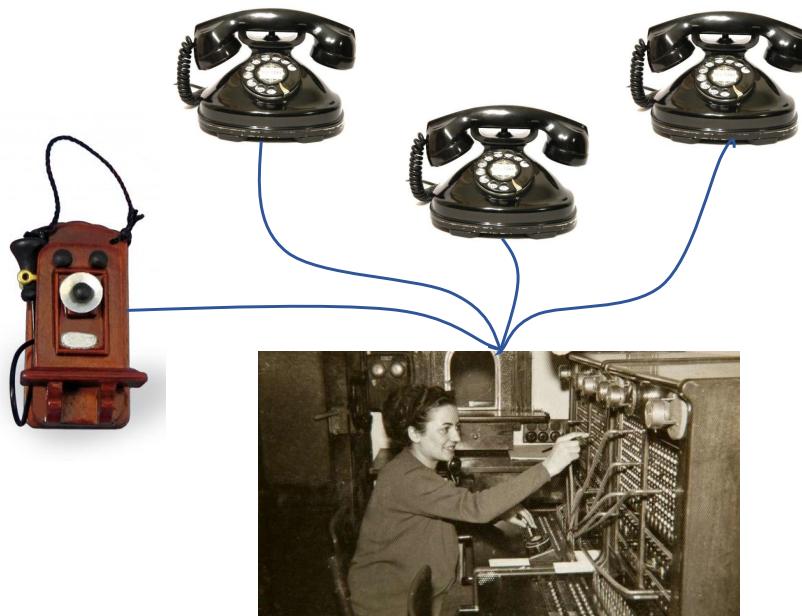
### **9.1. Introducción**

En este capítulo se abordan los problemas y métodos modernos para aprovechar al máximo el canal en comunicaciones multiusuarios.

### **9.2. Multiplexación por división de tiempo TDM. La revolución PCM.**

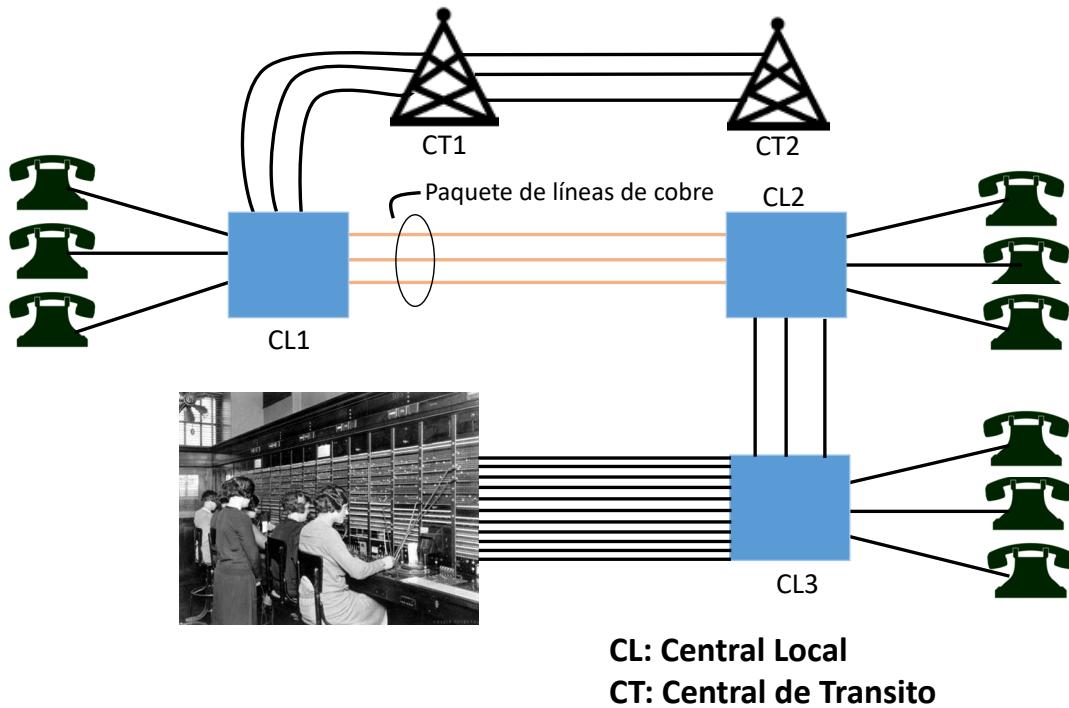
Veamos el problema que dió origen a aparición del Multiplexado por División de Tiempo (TDM) en los años 60: Todos los hogares soñaban con tener un teléfono, pero los costos del servicio no estaban al alcance de todos. La telefonía era analógica y establecer una llamada entre dos personas requería establecer una conexión física, alambrada entre ellas, como se muestra en la siguiente figura.

Figura 1: Central local



*Central Local (CL)*

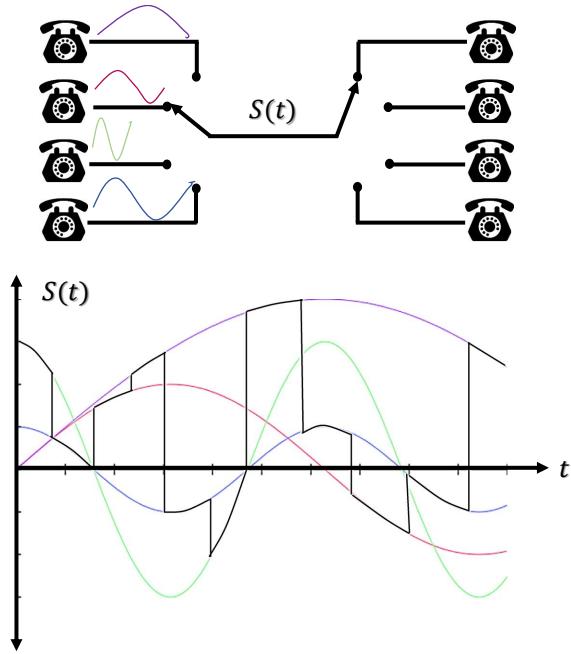
Figura 2: Central de transito



El problema: Se volvió insostenible el creciente número de operadoras en las centrales telefónicas, sin que bajaran sensiblemente los costos de conexión. La telefonía seguía siendo solo para los elites. La UIT se propuso impulsar una revolución para lograr que cada hogar tuviese un teléfono. Esa es conocida como la revolución PCM (por sus siglas en inglés *Pulse Code Modulation*). La idea es que la voz de cada llamada se convierte a PCM, las centrales telefónicas cambiarían la conmutación manual por la conmutación por circuitos que se podía realizar de manera automática, por medios computarizados. Pero el mayor avance consistió en lograr que cada línea que une dos centrales telefónicas, que antes podía llevar solo una llamada en un momento determinado, pudiera ahora llevar cientos de llamadas multiplexadas por división de tiempo.

La siguiente figura muestra resume el principio de TDM.

Figura 3: Ejemplo de TDM



Como puede verse, entre los dos puntos comunicantes viaja una sola señal, en cada uno de los teléfonos receptores llega una señal que es una versión PAM (por su significado en inglés *Pulse Amplitud Modulation*) de la emitida. Puede decirse que cada señal que llega al destino resulta muestreada, pues llega una pequeña duración de ella cada periodo  $T_s$ . Por lo tanto, este proceso no resulta en pérdida de información si se respeta el teorema de Nyquist. De modo que  $s(t)$  es la señal TDM. En este caso, surge el concepto de canal de tiempo, pues la manera en que un usuario distingue de otro en el medio de propagación es una ventana de tiempo.

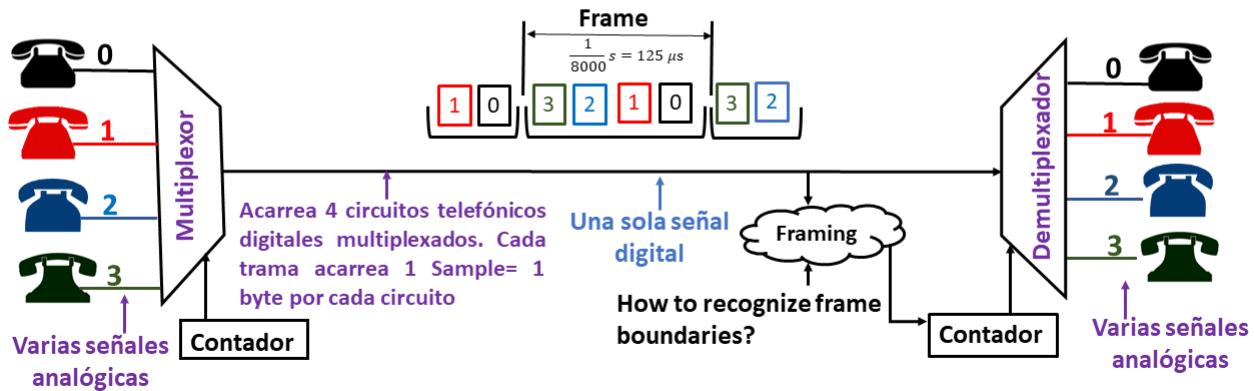
En la práctica, la telefonía usó una forma más avanzada de TDM. Consiste en que antes de emitir la señal  $s(t)$  esta se convierte a unos y ceros. Para ello, la señal  $s(t)$  es muestreada de manera que haya una muestra por cada ventana de duración  $T_s$ , luego se cuantizan y finalmente se traduce a una señal binaria que no es otra cosa que una señal PCM (por su significado en inglés *Pulse Code Modulation*) produce la señal PCM que no es un proceso Una versión más avanzada de él aunque esta figura no lo muestra, una vez la señal se multiplexa es convertida a PCM. De modo que la señal TDM usada hoy día es digital como se muestra en la siguiente figura.

Figura 4: Teléfono digital

## Telefonía Digital

Calidad de voz telefónica  $\leq 3.5$  [kHz]  $\rightarrow \cong 8000 \frac{\text{samples}}{\text{second}}$

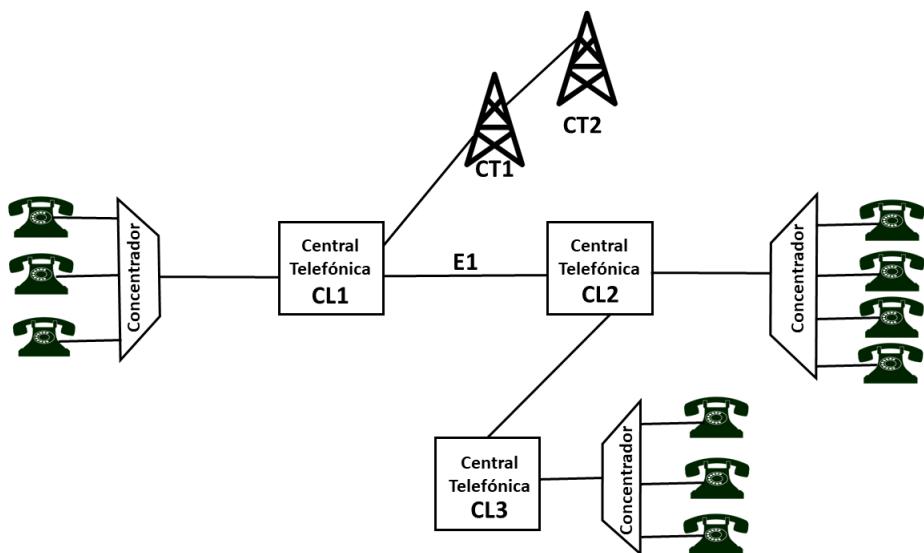
$$8000 \frac{\text{samples}}{\text{s}} \cdot 8 \frac{\text{bits}}{\text{samples}} = 64,000 \frac{\text{bits}}{\text{s}} \dots \text{Un circuito de telefonía digital}$$



En telefonía al multiplexor a menudo se le llamó concentrador, ya que recibe muchas líneas y entrega pocas. El multiplexor digital usado en la telefonía también incluía capacidades de conmutación. Era justamente la era de la conmutación de circuitos que consistía en cambiar la ventana de tiempo asignada a una llamada de modo que así cambiará el destinatario.

Una red más real requería usar centrales telefónicas para que una llamada pudiese llegar a cualquier destino.

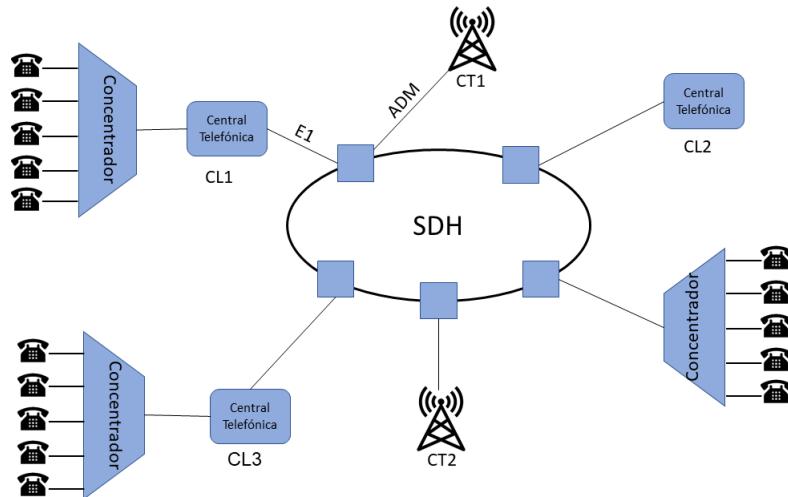
Figura 5: Modelo con las redes telefónicas



Al poco tiempo se complementó el modelo con las redes de transporte especializadas, inicialmente,

en interconectar centrales telefónicas. La tecnología usada en esas redes se llama SDH (por su significado en inglés *Synchronous Digital Hierarchy*) y hoy sirven para interconectar todas las redes de voz y datos, mediante enlaces de fibra óptica, incluyendo cable submarino, enlaces de microondas y enlaces satelitales.

Figura 6: Red SDH

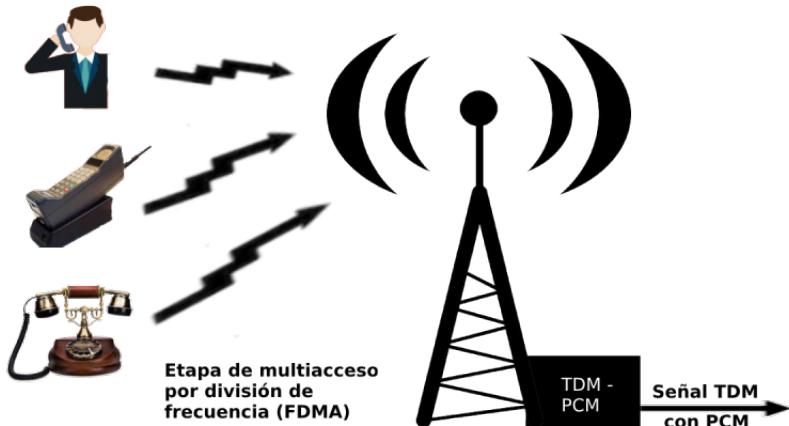


### 9.3. MULTIACCESO POR DIVISIÓN DE FRECUENCIAS (FDMA)

#### 9.3.1. Problema que justifica esta tecnología:

Al comenzar la era de las comunicaciones móviles surgen diversos sistemas de primera generación 1G como: NMT (*Nordic Mobile Telephone*), AMPS (en inglés *Advanced Mobile Phone System*), TACS (*Total Access Communication System*), C-450, Radicom, RTMI, NTT. El reto principal consistía en lograr de muchos usuarios pudieran conectarse a una misma radiobase. La solución consistió simplemente en asignar una banda de frecuencias a cada llamada.

Figura 7: Ejemplo de multiacceso

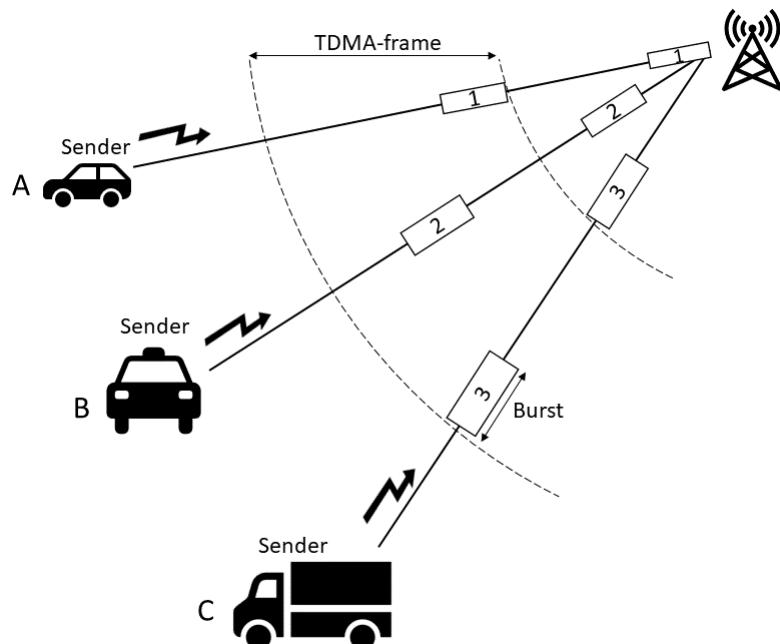


El multiacceso es la manera en que múltiples usuarios logran compartir un mismo recurso inalámbrico para resultar conectados a una misma radiobase.

## 9.4. El Multiacceso por División de Tiempo (TDMA)

Con FDMA solo las élites tenían acceso a las comunicaciones móviles, pues resultaban extremadamente costosas. Como ocurrió con la telefonía fija, la UIT ahora se propuso la meta de lograr que cada persona tuviese un celular y eso requería una nueva revolución tecnológica para hacer más eficiente y flexible el uso del espectro. Surgieron entonces las comunicaciones móviles de Segunda Generación (2G). La novedad era esencialmente la implementación de TDMA. Como se deduce de la siguiente figura, el terminal de usuario debe preparar la señal digital para ir enviándola por una ventana de tiempo que se repite periódicamente. Surge así un nuevo tipo de canal para enviar información- el **canal de tiempo**.

Figura 8: Ejemplo de multiacceso



En la figura vemos que al usuario A ocupa el canal 1, el B el 2 y el C el 3. Con TDMA surgen los sistemas móviles de segunda generación (2G) que se caracteriza por: ser digital, ser asequible, llevar voz y datos. Entre los sistemas 2G se tienen: GSM, D-AMPS (también conocido como sistema TDMA), *cdma-one* (también conocido como IS-95), PDC.

## 9.5. Técnicas adicionales usadas en las comunicaciones móviles

### 9.5.1. Técnicas o esquemas de diversidad

El problema a resolver con estas técnicas puede ser planteado de la siguiente manera:

El Teorema de Shannon nos plantea un reto importante, que puede ser resuelto mediante técnicas de codificación del canal. Pero en un sistema de comunicaciones es muy común que los errores aparezcan concentrados de manera indeseada. Por ejemplo, cuando vimos el comportamiento de los niveles de potencia de la señal RF cuando se presenta el fenómeno de multirayectoria, vimos que la señal puede presentar fuertes caídas, con respecto a la media. Supongamos que una caída de esas es debido al paso de carros entre el receptor y la radio base, el caso es que eso hace que se produzca una ráfaga de un volumen alto bits perdidos concentrados en un instante de tiempo. El volumen puede ser tan alto que los algoritmos de codificación del canal no tienen nada que hacer. Entonces diríamos que el Teorema de Shannon falla en estos casos?.

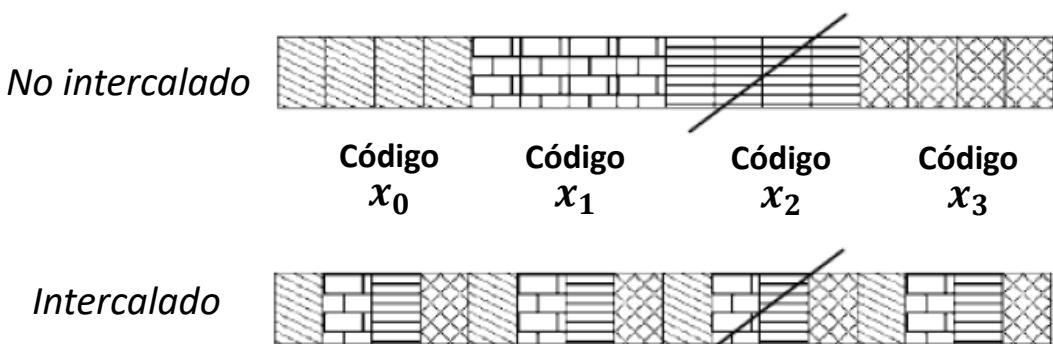
La respuesta es que se requieren técnicas adicionales, como los esquemas de diversidad para lograr que los errores queden de alguna manera distribuidos uniformemente y la codificación del canal funcione mejor.

Los esquemas de diversidad se tratan de métodos para mejorar la recepción de una señal mensaje usando dos o más canales con diferentes características, pero usualmente sin agregar otra redundancia que la que pueden imprimir los métodos de codificación del canal. Desde este punto de vista, puede decirse que esta técnica es complementaria a los métodos de codificación del canal, para lograr que la codificación del canal sea más efectiva. Es una técnica común para combatir el desvanecimiento variable y las interferencias cocanal que experimentan las señales en su paso por un canal que producen ráfagas de errores. Se basan en que canales individuales experimentan diferentes niveles de desvanecimiento y de interferencia. La idea es transmitir y/o recibir múltiples versiones de la misma señal para hacerles un tratamiento en el receptor que permita obtener una mejor versión del mensaje que cuando se usa un solo canal.

### 9.5.2. Diversidad de tiempo. *Interleaving, FEC*

Se usa principalmente cuando el sistema de comunicación usa canales de tiempo, como ocurre al usar TDMA en un sistema de comunicaciones móviles, donde a la comunicación de un usuario le corresponde solo una ventana en el tiempo o *Time-Slot* dentro de un canal de frecuencia. Es el caso de las comunicaciones móviles digitales en general, pero el caso más claro son los sistemas de comunicaciones móviles de segunda generación (2G), los cuales usan, en la interfaz aire, PCM y multiplexado por división de tiempo (TDM) como principal innovación con respecto al sistema de comunicaciones móviles de primera generación (1G) el cual usaba métodos analógicos en la interfaz aire. Una señal TDM tiene muchos canales de tiempo, y resulta lógico pensar que cada uno de ellos pasa por diferentes condiciones de propagación. Por ejemplo, si la señal entre una radiobase y un usuario es interrumpida por el paso de algún obstáculo en medio de ellos, esto puede impactar a algunos canales de tiempo pero no a todos. En este caso toma sentido una técnica conocida como *bit-interleaving* que consiste en mezclar los bits de todos los canales de tiempo de manera que, si se pierde un canal de tiempo, no se pierdan todos los bits de unos ciertos *time-slots* sino que esa pérdida se reparta homogéneamente entre todos los usuarios, de modo que cada usuario pierda solo unos pocos bits. Ahora es cuando surge el reto de poder recuperar los pocos bits perdidos y cuando toma sentido otra técnica adicional conocida como *Forward Error Correction (FEC)*. Esta consiste en agregar redundancia al mensaje, de manera previa o posterior al *interleaving*, para lograr reconocer en el receptor los bits perdidos y poder recuperarlos. El *interleaving* evita entonces las ráfagas de errores, mejorando el desempeño de las técnicas FEC que son efectivas cuando hay pocos bits perdidos en cada canal de tiempo. Es importante notar que el *interleaving* no significa enviar el mensaje de manera redundante, son las técnicas FEC las que introducen información que permite al receptor descubrir errores. Desde este punto de vista *interleaving* y FEC son capas diferentes en un sistema de comunicaciones: la capa de *interleaving* se encarga de distribuir bits, la FEC de mantener un sistema capaz de corregir errores para lo cual en el transmisor se tiene una especie de codificador y en el receptor un decodificador. La cuestión es que la capa de FEC es más efectiva con el *interleaving* y esta última tiene sentido sólo si se va a usar FEC, luego, van de la mano, pero queda claro que el interleaving no aplica redundancia, pero es el FEC quien lo hace.

Figura 9: Ejemplos de la diversidad de tiempo utilizando el intercalado.



Pero el bit *interleaving* también tiene sentido incluso cuando la comunicación no se basa en TDMA. Supongamos que se tiene una comunicación digital inalámbrica solo entre dos puntos. Está claro que FEC

va a funcionar mejor si los errores se concentran en ráfagas, va a ser más difícil corregir esos errores que cuando quedan repartidos de manera uniforme en el tiempo.

## 9.6. Diversidad de espacio

En los sistemas de comunicación cableada corresponde a la transmisión de una misma información por diferentes cables. En el caso de las comunicaciones inalámbricas es menos obvio pues no es la idea usar varios canales para la comunicación. Se trata más bien de una Diversidad de Antena, usando varias antenas para transmitir y/o recibir, lo cual debe ir seguido de técnicas de procesamiento digital para recuperar el mensaje a partir de esa diversidad que permite que cada antena aporte algo diferente que ayuda a la tarea de recuperación. Cuando el transmisor y el receptor usan varias antenas, se habla de técnicas MIMO (por su significado en inglés *Multiple Input Multiple Output*). Pero también se habla de *macrodiversity* o *site diversity*, cuando las antenas que participan en la comunicación están muy alejadas entre sí como es el caso de la existencia de varios puntos de acceso (*access point*) como se hace en las redes WLAN. Cuando esas distancias entre las antenas son menores a una longitud de onda se habla de microdiversidad y MIMO es un ejemplo de ella. Estas técnicas pueden combinarse adicionalmente con codificación para cada canal, con lo cual se habla de *space-time coding*(STC), así como con técnicas de *beamforming* para hacer que las antenas busquen a los usuarios, por ejemplo, cuando la concentración de usuarios cambia en el curso del día.

Figura 10: Diversidad de espacio

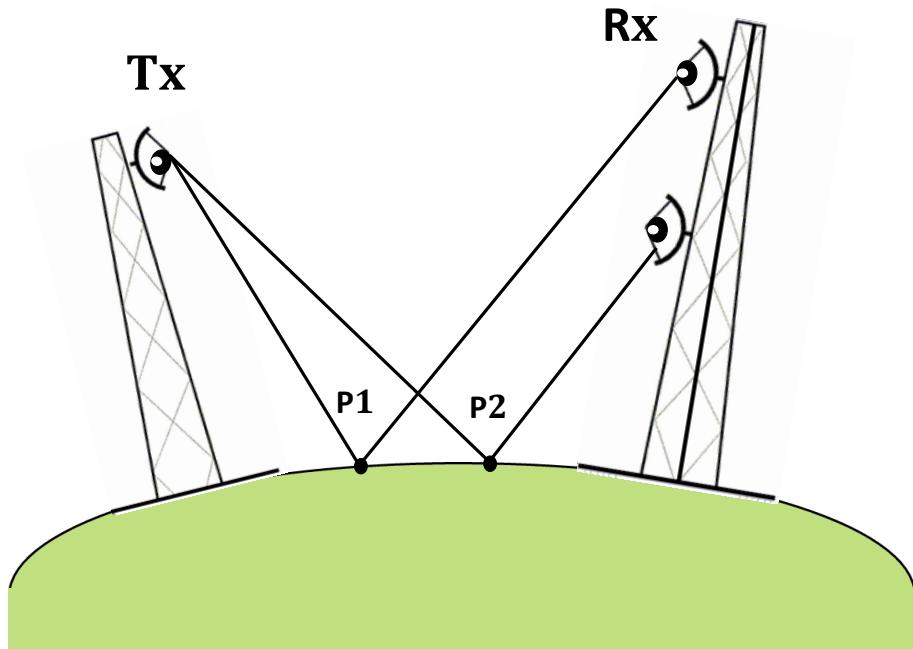
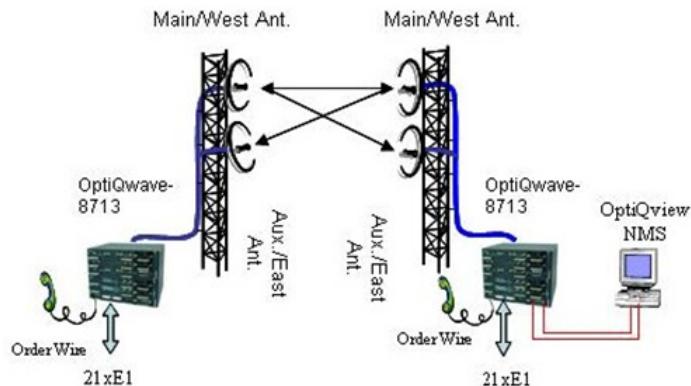


Figura 11: Diversidad de espacio

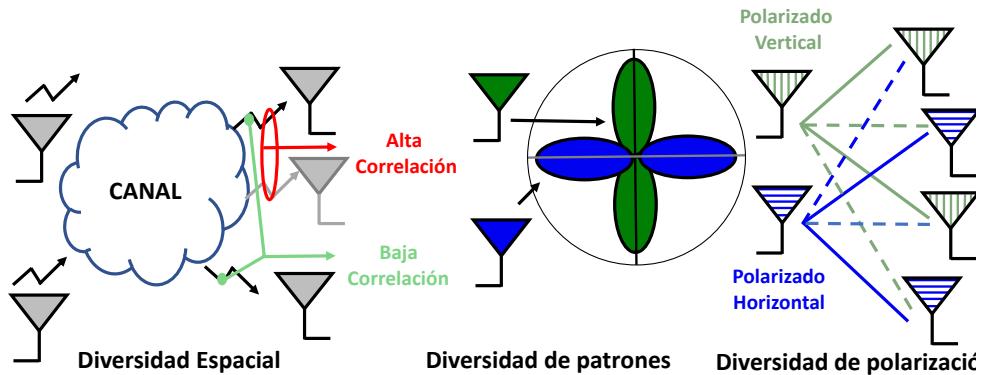
#### 1+0 Space Diversity Application diagram



#### 9.6.1. Diversidad de polarización

Consiste en enviar y/o recibir varias versiones de un mensaje usando diferentes polarizaciones de antena. Es en el fondo un caso especial de diversidad de espacio ya que usualmente se usan al menos dos antenas para darle a cada una polarización diferente.

Figura 12: Ejemplo de diversidad de polarización



#### 9.6.2. Diversidad de frecuencias

El mensaje se transmite y/o recibe usando varios canales de frecuencia. Como en los casos anteriores, no necesariamente debe haber transmisión redundante, pues también es muy común la distribución pre-meditada de la información en varios canales. Supongamos un sistema de comunicaciones móviles donde el espectro disponible se ha dividido en sub-bandas A,B,C,D. Supongamos también que a un usuario se le asigna la sub-banda A, pero en la práctica la información transmitida para todos los usuarios ha sido previamente mezclada, de modo que resulta repartida en todas las sub-bandas. Entonces, el usuario, en modo de recepción, deberá poder recibir todas las sub-bandas para poder pretender extraer lo que le corresponde. La técnica que más claramente se relaciona con este ejemplo es la que se conoce como OFDM usada en las comunicaciones móviles de Cuarta Generación (4G) y en la televisión digital.

Figura 13: Múltiples tuberías de capa física (PLP)



Otra posibilidad es el uso de la técnica de conocida como FHSS (*Frequency Hopping Spread Spectrum*), usada en las redes WIFI y Bluetooth.

#### 9.6.3. Diversidad de multi usuario

Consiste en que el transmisor o radiobase agenda de manera oportunista al usuario, estudiando las condiciones del canal entre ella y todos los usuarios para seleccionar al usuario que tiene mejores cualidades para la comunicación. La idea es elegir el mejor instante para atender a cada usuario de la manera más óptima posible.

#### 9.6.4. Diversidad cooperativa

Se trata de una especie de diversidad de ganancia de antena que se alcanza usando antenas de otros usuarios o nodos.

#### 9.6.5. Diversidad de códigos

Un ejemplo es lo que puede lograrse usando la técnica conocida como DS-SS (*Direct Sequence Spread Spectrum*) que predomina en las comunicaciones móviles de tercera generación (3G) que facilita el uso redundante del ancho de banda.

### 9.7. La Duplexación

Es la solución que permite a los usuarios conectados a una radiobase mantener una comunicación bidireccional.

Duplexación por División de Frecuencias (FDD) se usa una banda de frecuencias para la información que va desde el usuario a la radiobase, es decir para el **Enlace de subida o up-link** y otra banda para la información que va desde la radiobase al usuario. Es decir el **Enlace de bajada o Dowlink**.

FDD es el método de Duplexación más usado desde el comienzo de las comunicaciones móviles hasta el día de hoy. A manera de ejemplo, la figura 14 muestra la distribución de la banda conocida como 1900 MHz, entre los operadores móviles establecidas en Colombia. Allí vemos por ejemplo que a la banda A destinada para *up-link* le corresponde la banda A destinada para *Downlink*. Obsérvese también que en medio de esas grandes bandas existe una central que sirve de guarda, pero que hoy día busca ser aprovechada mediante otro tipo de Duplexación, **La Duplexación por División de Tiempo (TDD)**.

Figura 14: Banda 1900 como posibilidad de ser aprovechada por duplexación por división de tiempo TDD.

|        |   |        |   |   |        |               |        |    |        |    |    |        |
|--------|---|--------|---|---|--------|---------------|--------|----|--------|----|----|--------|
| A      | D | B      | E | F | C      | Banda Central | A'     | D' | B'     | E' | F' | C'     |
| 15 MHz | 5 | 15 MHz | 5 | 5 | 15 MHz | 20 MHz        | 15 MHz | 5  | 15 MHz | 5  | 5  | 15 MHz |

a) Canalización de la banda 1900 en Colombia

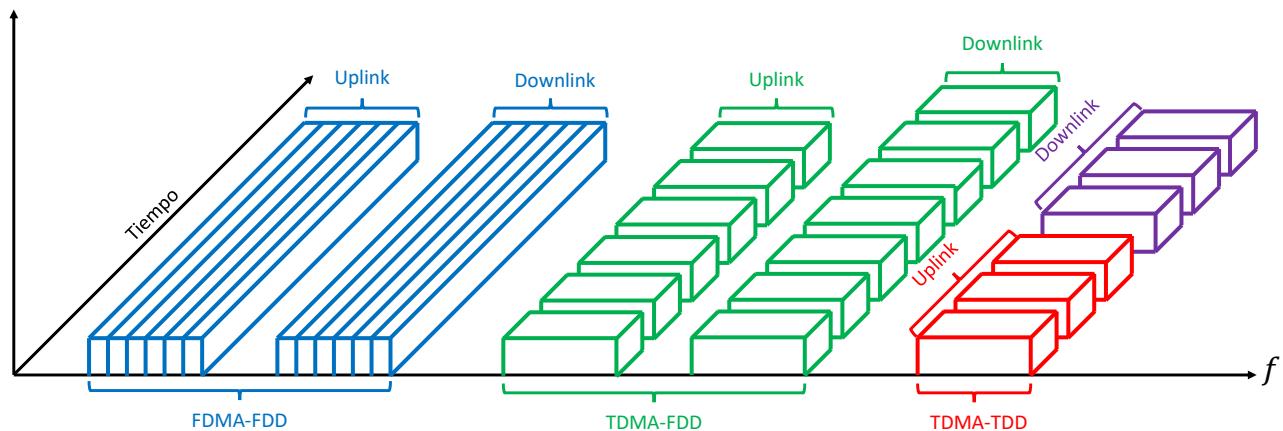
| DIS | T | C | O1 | M   | C   | O2 | T (PCS) | Banda Central | DIS | T | C | O1 | M   | C   | O2 | T (PCS) |
|-----|---|---|----|-----|-----|----|---------|---------------|-----|---|---|----|-----|-----|----|---------|
| 5   | 5 | 5 | 5  | 7,5 | 7,5 | 5  | 20 MHz  | 20 MHz        | 5   | 5 | 5 | 5  | 7,5 | 7,5 | 5  | 20 MHz  |

b) Ocupación de la banda en Colombia Tigo-Colombia Móvil (T). Movistar (M). Comcel (C). Espectro disponible (DIS)



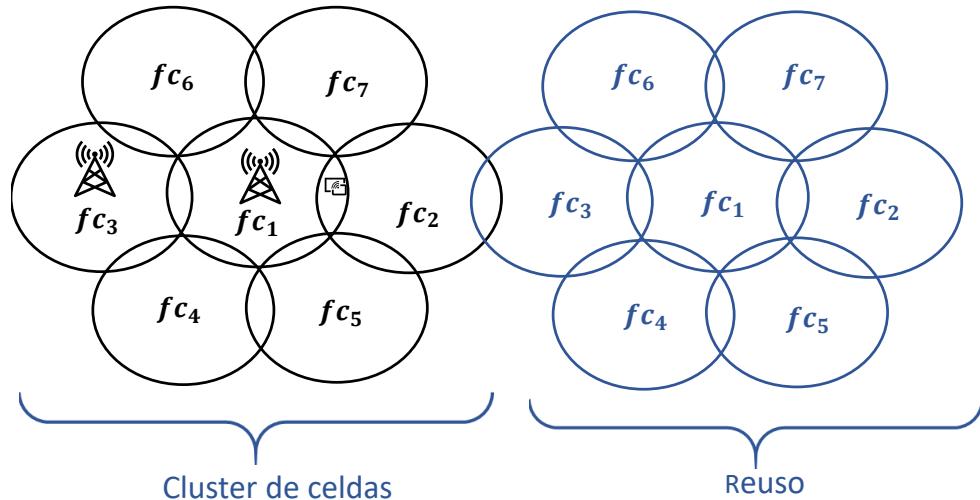
La figura 15 resume la distribución de bandas de frecuencia de los términos estudiados.

Figura 15: Resumen de FDMA-FD, TDMA-FDD, FDMA-TDD.



### 9.7.1. La celda y la radiobase

Figura 16: La celda y la radiobase



### 9.7.2. El reuso

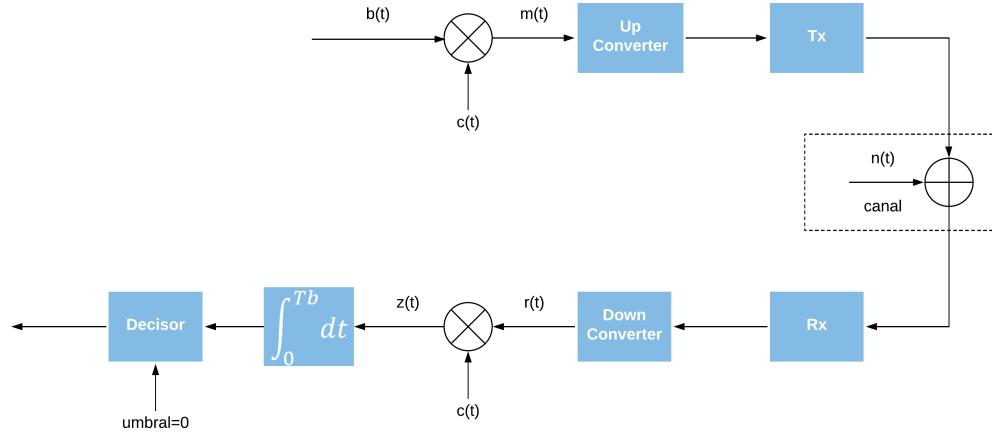
### 9.7.3. El handoff

## 9.8. Spread Spectrum (SS) y el Multiacceso por División de Códigos (CDMA)

Es la tecnología que hace posible el multiplexado por división de códigos (CDM) y consecuentemente el multiacceso por división de códigos (CDMA), que dio origen a un sistema 2G conocido como CDMA-ONE, que inicialmente competía con los sistemas GSM y D-AMPS. La batalla en 2G por el mercado mundial la ganó GSM debido a la gran expansión que logró al provenir de la unión de todos los países europeos. Pero CDMA-ONE logró demostrar que era la tecnología que permitía el uso más flexible del espectro.

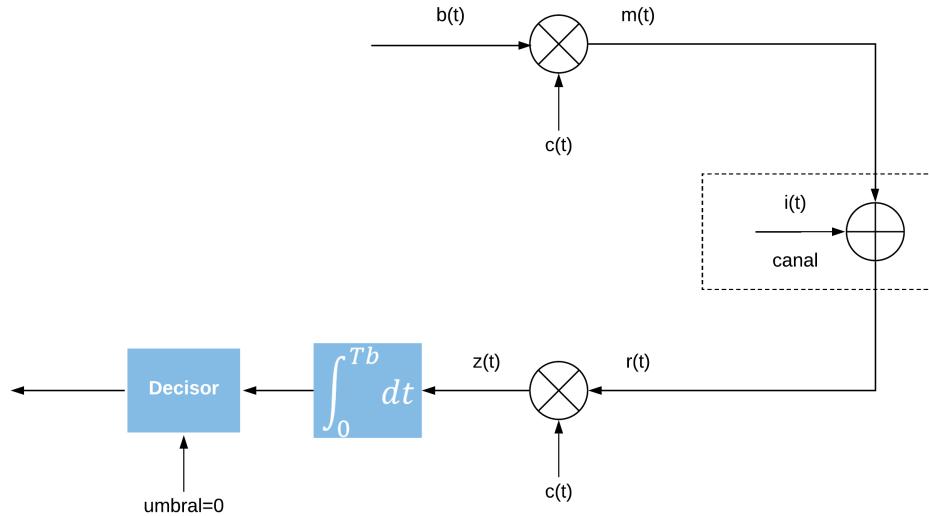
**Problema que impulsa la tecnología Spread Spectrum:** La necesidad no está originalmente en las comunicaciones móviles, sino simplemente en la comunicación entre dos puntos. Por ejemplo, entre el piloto de un avión militar y el comando en tierra. Durante la segunda guerra mundial era ideal poder emitir señales que el enemigo no pudiese detectar. La idea consiste en lograr distribuir la energía de la señal útil en una banda muy ancha, para que su PSD se parezca a la del ruido blanco. Así el enemigo solo detectará en sus equipos de medición la presencia de ruido blanco. Hoy esta tecnología es ampliamente usada para la implementación de enlaces de microondas en bandas de frecuencia de uso libre, las cuales son muy congestionadas. En este sentido, SS permite establecer enlaces incluso en bandas que ya están ocupadas sin interferirlas.

Figura 17: Spread Spectrum



Para su análisis resulta útil usar un sistema equivalente en banda base, como el siguiente:

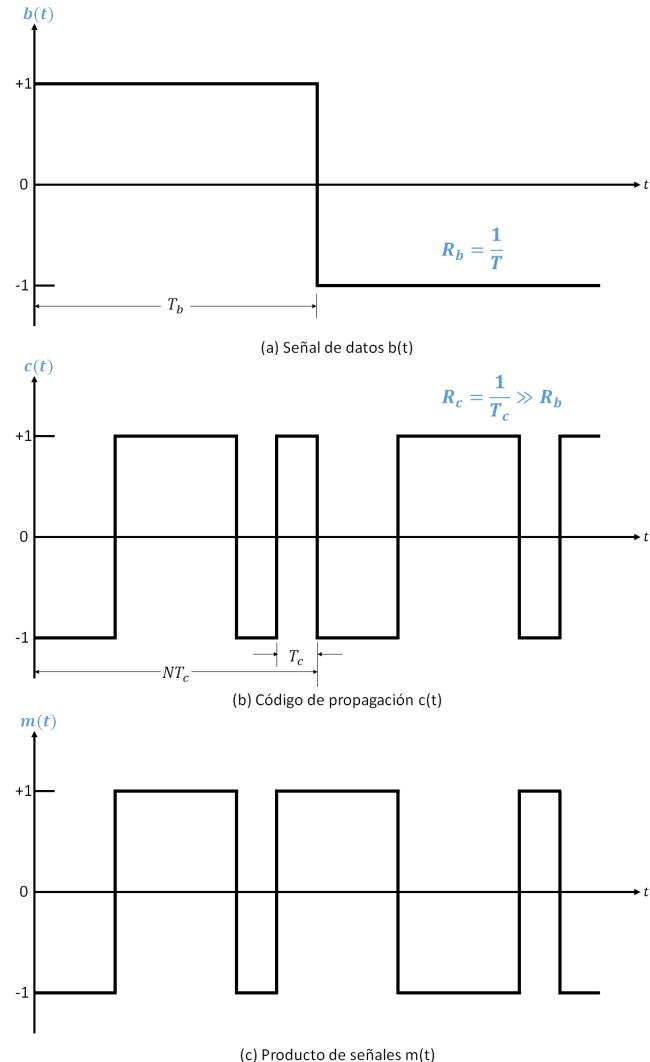
Figura 18: Sistema equivalente en banda base.



$b_t$  es usualmente una señal binaria con  $R_b = \frac{1}{T_b}$ , pero puede ser cualquier señal digital banda base, por ejemplo la envolvente compleja 8-PSK, con  $R_{sym} = 1/T_{sym}$ .

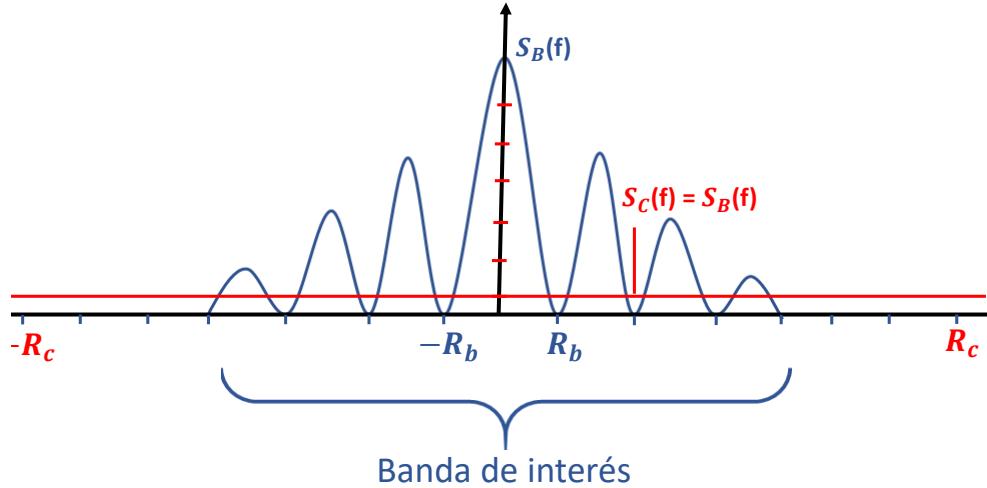
$c(t)$  es una señal de ruido que se caracteriza por tomar valores en 1 y -1 cada  $T_c$ , donde  $T_c \ll T_b$  o  $T_c \ll T_{sym}$ .

Figura 19: Señales de ejemplo.



Veamos la PSD de estas señales:

Figura 20: PSD de las señales.

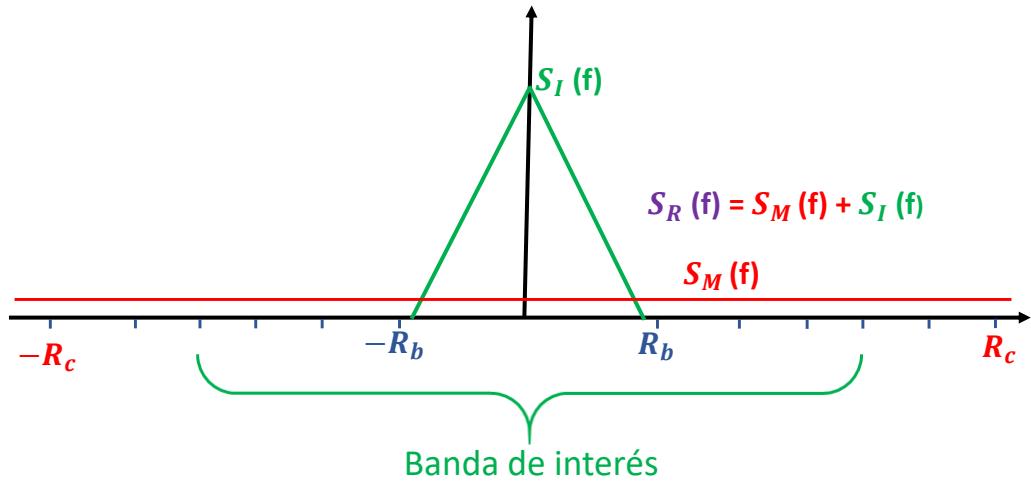


Podemos observar que:

- La PSD de la señal  $c(t)ym(t) = b(t) * c(t)$  se expande hasta una frecuencia  $N = \frac{R_c}{R_b}$  veces mayor, a la vez que cae en altura, hasta el punto que, dentro de una banda de interés para la transmisión/recepción  $S_c(f)$  y  $S_m(f)$  resultan siendo similares a la del ruido blanco. Por lo anterior,  $c(t)$  aunque es binaria bipolar es considerada una señal de ruido.
- Es ideal que  $c(t)$  sea totalmente aleatoria para que tenga la PSD mostrada, pero esto es difícil de lograr ya que  $c(t)$  debe ser conocida tanto en el transmisor como en el receptor.
- En la práctica,  $c(t)$  es generada por un dispositivo conocido como GENERADOR DE SECUENCIAS DE PSEUDORUIDO. La idea es lograr que exista un generador idéntico en la parte transmisora y en la receptora. Esta limitación hace que  $c(t)$  no tenga características completamente aleatorias, al punto que incluso  $c(t)$  resulta siendo una señal periódica, aunque el periodo puede ser tan grande que esta periodicidad puede ser ignorada en el análisis. Por esta razón,  $c(t)$  es en la práctica una señal o secuencia de pseudoruido.
- Entre más grande será el factor de dispersión (SF- del inglés *Spreading Factor*)  $N = \frac{R_c}{R_b}$  o  $N = \frac{R_c}{R_s y m}$ , menos se notará la señal emitida  $m(t)$  en receptores diferentes al nuestro, los cuales la verán más bien como ruido blanco.

**Análisis de la parte receptora:** Es importante resaltar que la señal  $i(t)$  que en el canal se suma de manera indeseada a nuestra señal útil  $m(t)$  puede ser, más que ruido blanco, potentes señales de otras emisiones indeseadas, es decir, interferencias que producen otros sistemas de comunicación de la banda de frecuencia de nuestro sistema. Por eso, revisaremos el peor caso de la PSD de una señal  $i(t)$  como se muestra en la figura 21

Figura 21: Peor de los casos de la señal  $i(t)$ .



$$r(t) = m(t) + i(t); z(t) = r(t) * c(t) \quad (9.1)$$

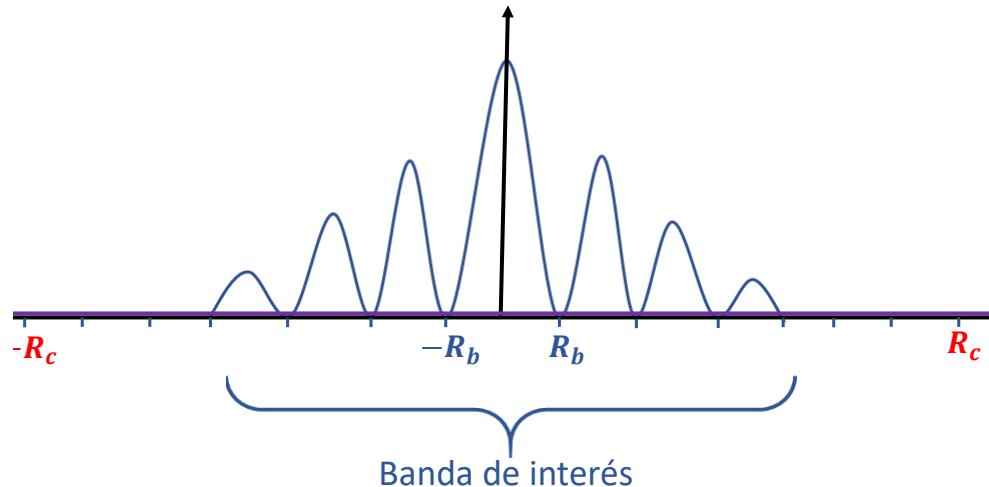
$$z(t) = [m(t) + i(t)] * c(t) = [b(t) * c(t) + i(t)] * c(t) \quad (9.2)$$

$$z(t) = b(t) * c^2(t) + i(t) * c(t) \quad (9.3)$$

Si  $c(t)$  es binaria bipolar de amplitud 1, entonces  $c^2(t) = 1$ .

$$z(t) = \underbrace{b(t) + i(t)}_{\text{Nuestro mensaje se recupera}} * \underbrace{c(t)}_{\text{La señal de interferencia se dispersa en las frecuencias}} \quad (9.4)$$

Figura 22: Señal  $b(t)$ .



Finalmente, un filtro de acoplamiento y un decisor de umbral permiten recuperar la señal  $b(t)$ .

### 9.8.1. Tipos de Spread Spectrum

Hoy se habla de dos grandes tipos de espectro disperso:

- El que hace uso de las secuencias de pseudoruido, que es justo el tema explicado anteriormente y se conoce mejor como DS - SS (*Direct Sequence Spread Spectrum*).

- El que hace uso de saltos de frecuencias y que se conoce mejor como FH – SS (*Frequency Hopping Spread Spectrum*).

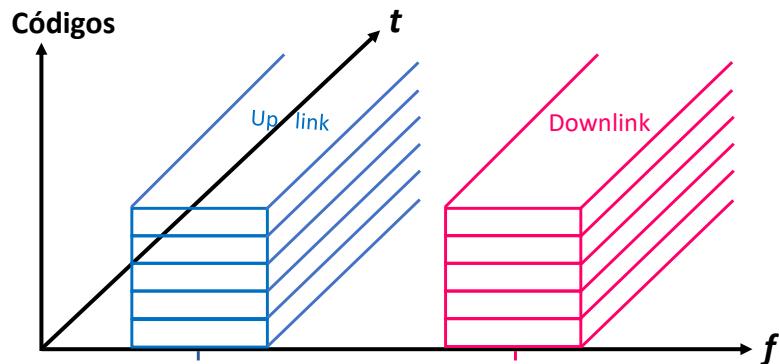
### 9.8.2. El Multiacceso por División de códigos (CDMA)

DS-CDMA (*Direct Sequence Code Division Multiaccess*).

Este tipo de multiacceso fue usado por la empresa estadounidense Qualcomm en el sistema móvil 2G conocido como CDMA - ONE. Este sistema pronto perdió la batalla en los mercados, pero demostró que esta forma de multiacceso era más flexible y más eficiente que TDMA en comunicaciones que combinan voz y datos.

CDMA usa tecnología DS-SS y a menudo se conoce también como DS – CDMA. La figura 23 muestra cómo se usa el espectro cuando se usa CDMA.

Figura 23: Distribución del espectro con CDMA.



## 9.9. Multiacceso por División de Códigos de banda ancha (WCDMA)

Finalizado el siglo XX la UIT se propuso comenzar el año 2000 con una nueva revolución en las comunicaciones: Lograr que cada persona, independientemente de su condición económica, tuviese la posibilidad de estar siempre conectado, desde cualquier lugar, en cualquier momento, mediante cualquier terminal y en cualquier situación.

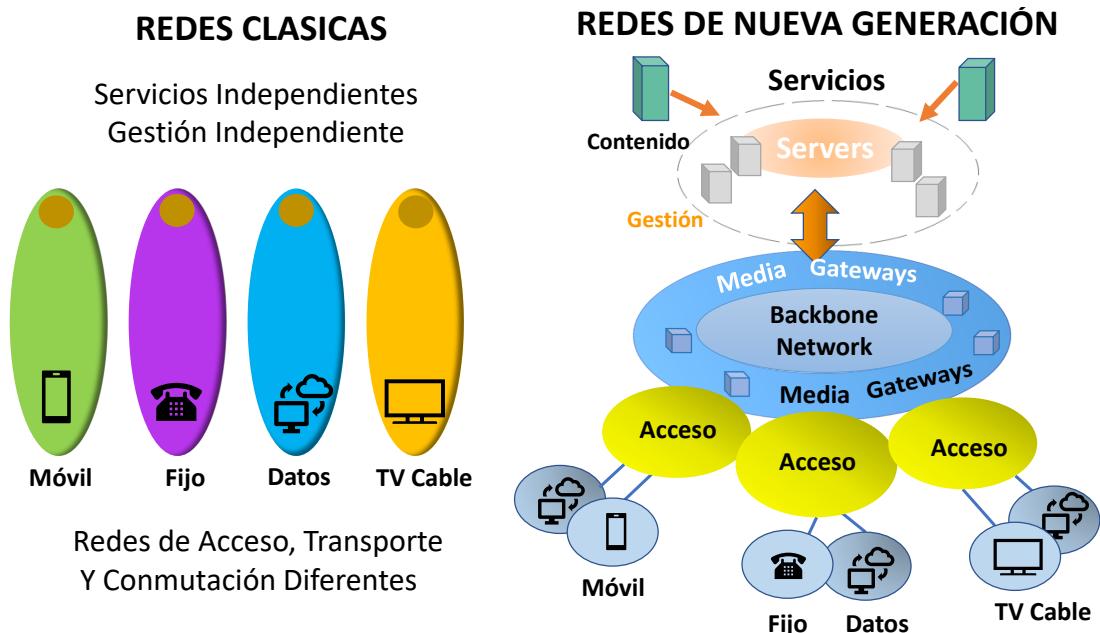
### 9.9.1. Convergencia de las comunicaciones

Hasta ese momento, las redes de telecomunicaciones se desarrollaban para prestar servicios independientes. Por ejemplo, la telefonía fija, permitía: Hacer llamadas, usar buzones de llamadas, llamadas bipartidas, etc. Pero solo entre usuarios de telefonía fija. Por supuesto se podía llamar desde un teléfono fijo a un móvil, pero los costos eran exorbitantemente mayores. Lo mismo ocurría con las redes de datos, tenían servicios propios, que si bien era posible acceder por medio de la telefonía fija o la móvil, se hacía con una muy baja calidad y unos costos absurdos.

Retos tecnológicos:

- Organizar el núcleo de todas las redes, ahora ese núcleo se llamaría *REDES DE NUEVA GENERACIÓN (NGN)*.

Figura 24: Redes clásicas y redes de nueva generación.



El impacto principal de las NGN consistía en posibilitar el desarrollo de servicios comunes a todas las posibilidades de los usuarios.

- Retos políticos: En Colombia, la política nacional se consolida en la *ley 1341 de 2009* ley 1341 de 2009 y en el *Plan de Gobierno Vive Digital*.
- Acuerdo internacional para producir un sistema único para las comunicaciones móviles, las cuales eran la clave para lograr la revolución trazada.

### 9.9.2. El UMTS

Hoy ese sistema *UMTS (Universal Mobile Telecommunication System)*, que corresponde a la *tercera generación de comunicaciones móviles (3G)*. Algunos también le llaman WCDMA ya que implementa la tecnología de multiacceso conocida como Multiacceso por División de Códigos de banda ancha, para lograr que los usuarios puedan establecer comunicaciones con requerimientos flexibles de uso de las bandas de frecuencia disponibles.

### 9.9.3. IMT-2000

Es la recomendación que emitió la UIT para impulsar las comunicaciones móviles 3G. Fue escrito con apoyo del 3GPP (*Third Generation Partnership Project*). Un conjunto de entidades interesadas en estos avances que incluyen: proveedores de tecnología, operadores, etc. El número 2000 hace referencia al esperanzador año 2000 y también a la necesidad de sumar bandas de frecuencias para el nuevo sistema, cercanas a los 2000 MHz. 2000 kbps también era la meta establecida para la velocidad de datos que podrían tener los usuarios desde sus terminales móviles usando la nueva Interfaz aire. IMT (*International Mobile Telecommunications*) significa la necesidad de un sistema único mundial.

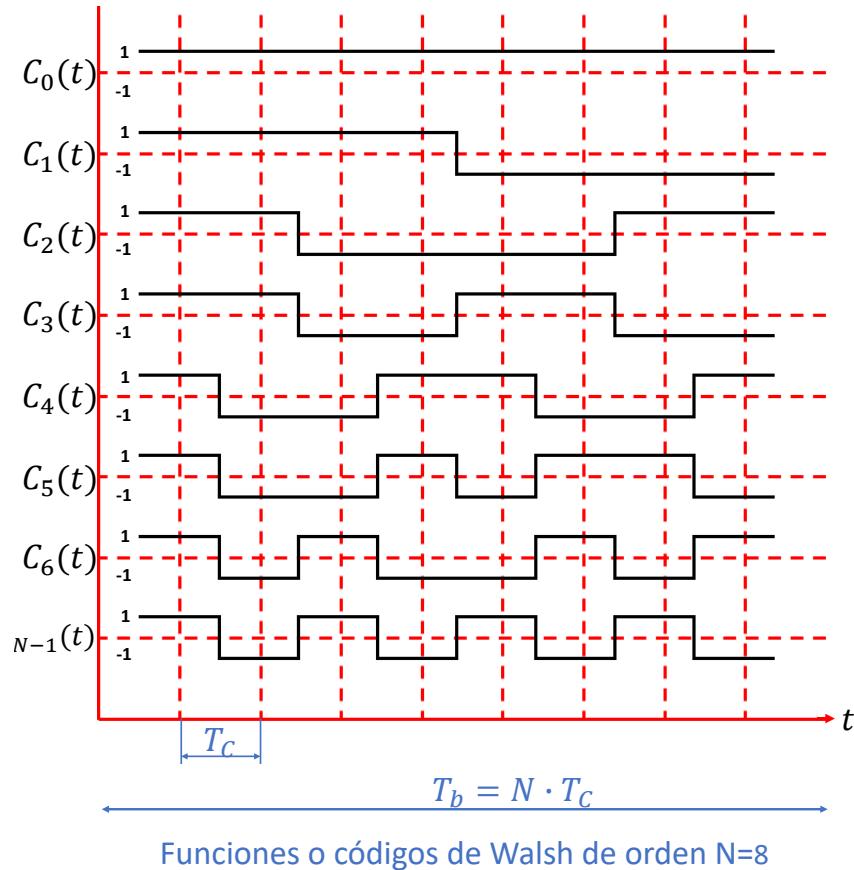
### 9.9.4. WCDMA:

El sistema de tercera generación UMTS usa CDMA de banda ancha (WCDMA) que se diferencia de CDMA en los siguientes aspectos:

Se usan bandas de frecuencia de ancho 5 MHz, que es notablemente mayor al usado en CDMA-ONE.

Se usan códigos de Walsh para distinguir los usuarios. Se trata de una familia de señales que tienen una forma similar a las secuencias de pseudoruido, pero son todas ortogonales entre sí. La siguiente figura muestra un ejemplo de ocho funciones de Walsh.

Figura 25: Funciones de Walsh.



$N$  es el número de diferentes códigos de Walsh. Coincide con el número máximo de usuarios simples. También coincide con el número de chips en 1 duración de un bit. Por tanto  $SF = NxN = 2^n$ , donde  $n$  es un entero.

El término “usuario simple” se ha usado porque en principio un usuario podría llegar a usar varios códigos en la práctica.

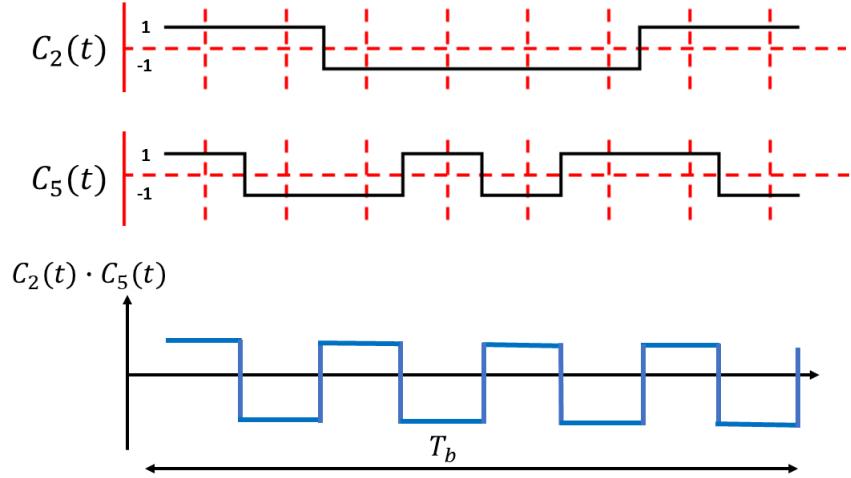
Debido a su ortogonalidad se cumple que:

$$\int_{T_b}^0 C_n(t) * C_k(t) dt = 0, \text{ para } n \neq k \quad (9.5)$$

Gracias a esto los usuarios no se interfieren entre sí.

Ejercicio: Comprobar si el código  $C_2(t)$  y el  $C_5(t)$  son ortogonales en el periodo  $T_b$ .

Figura 26: Ejercicios funciones de Walsh.

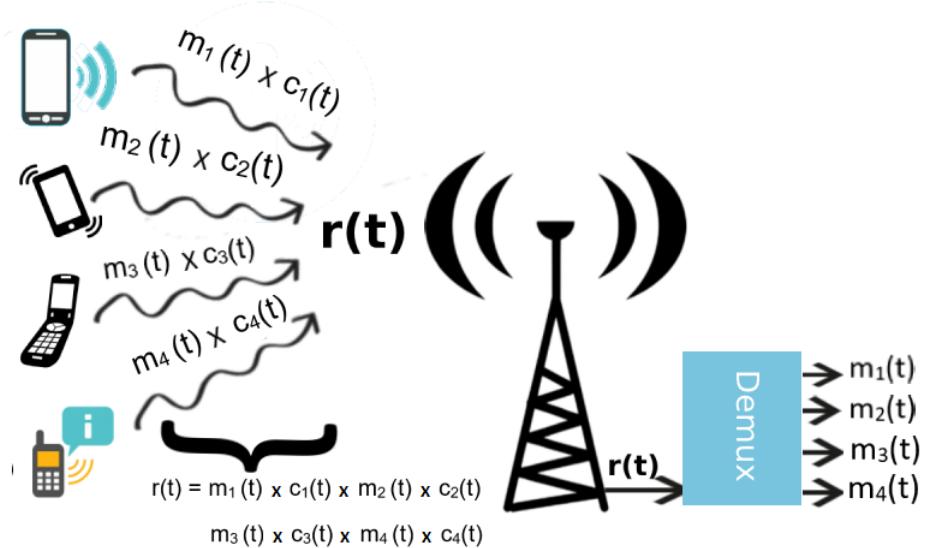


$$\int_{T_b}^0 C_2(t) * C_5(t) dt = 0 \quad (9.6)$$

Así que  $C_2(t)$  y  $C_5(t)$  son ortogonales en ese periodo.

**Ejercicio:** Tenemos el siguiente sistema de comunicación. 4 usuarios envían un mensaje cada uno a una radio base, donde esos 4 mensajes deben ser extraídos a partir de la señal  $s(t)$ .

Figura 27: Sistemas de comunicaciones.

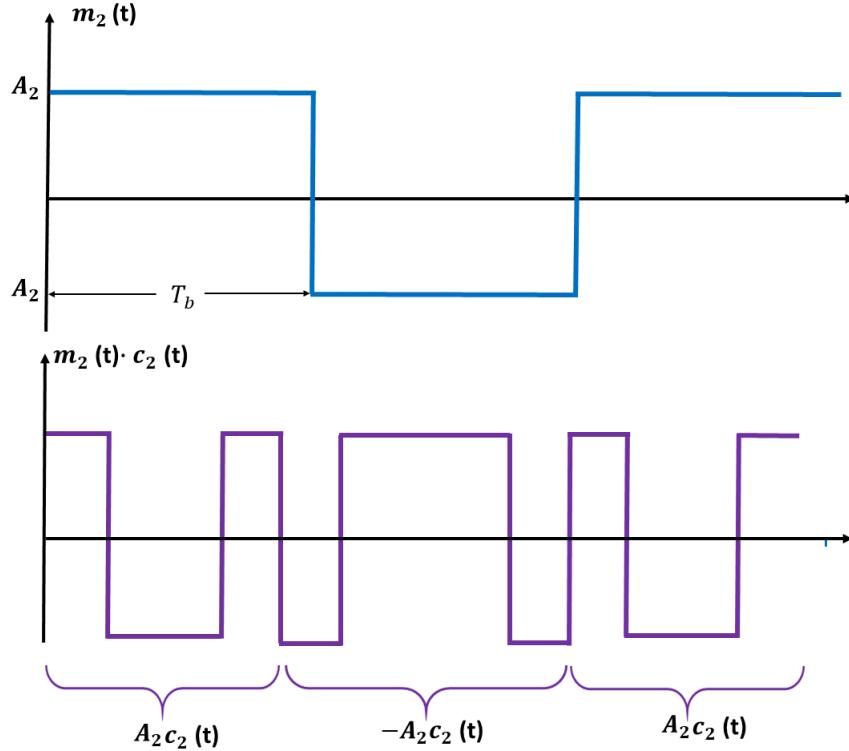


Para obtener cada señal  $m_k(t)$  en el demux se usa el esquema de la figura 27 para cada una de esas señales. Veamos el caso del usuario  $k=2$ . Dentro del demux se halla  $r(t) * c_2(t)$

$$z_2(t) = [m_1(t) * c_1(t) * c_2(t)] + [m_2(t) * c_2(t) * c_2(t)] + [m_3(t) * c_3(t) * c_2(t)] + [m_4(t) * c_4(t) * c_2(t)] \quad (9.7)$$

Es importante aclarar que durante el tiempo, cada una de las señales  $m_k(t)$  tiene un valor constante  $A_k$ . Veamos un ejemplo para  $m_2(t)$ . Si  $m_2(t)$  es la siguiente

Figura 28: Ejemplo  $m_2(t)$ .



$$\int_{T_b}^0 z(t)dt = \int_{T_b}^0 m_1(t)*c_1(t)*c_2(t)dt + \int_{T_b}^0 m_2(t)*c_2^2(t)dt + \int_{T_b}^0 m_1(t)*c_3(t)*c_2(t)dt + \int_{T_b}^0 m_4(t)*c_4(t)*c_2(t)dt \quad (9.8)$$

$$\int_{T_b}^0 z(t)dt = A1 \int_{T_b}^0 *c_1(t) * c_2(t)dt + A2 \int_{T_b}^0 + A3 \int_{T_b}^0 c_3(t) * c_2(t)dt + A4 \int_{T_b}^0 c_4(t) * c_2(t)dt \quad (9.9)$$

Debido a la ortogonalidad entre los códigos en el intervalo  $T_b$  tenemos que:

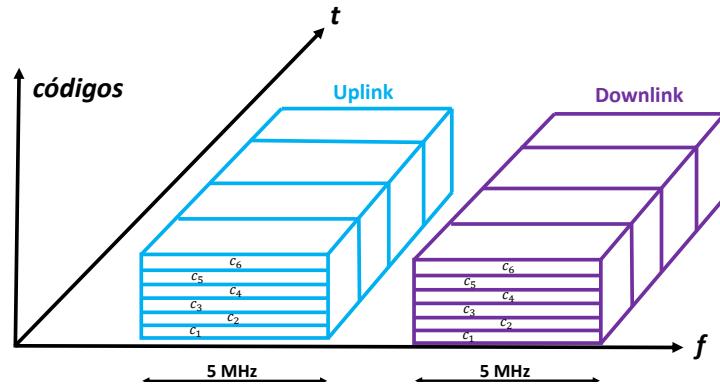
$$\int_{T_b}^0 z(t)dt = A2 * T_b \Rightarrow \frac{1}{T_b} \int_{T_b}^0 z(t)dt = A2 \quad (9.10)$$

*En conclusión:* Vemos que para cada ventana de tiempo de duración  $T_b$  es posible encontrar el valor de cada mensaje sin ningún tipo de afectación de parte de los otros usuarios.

Por el hecho de usar códigos ortogonales este tipo de técnica de espectro ensanchado se conoce como *Orthogonal Variable Spreading Factor* (OVSF) y es considerado como una forma de CDMA. Esto es debido a que cada código de Walsh tiene un ancho de espectro diferente y por lo tanto se produce un SF diferente a mensaje, pero nunca superior a  $SF=N$ .

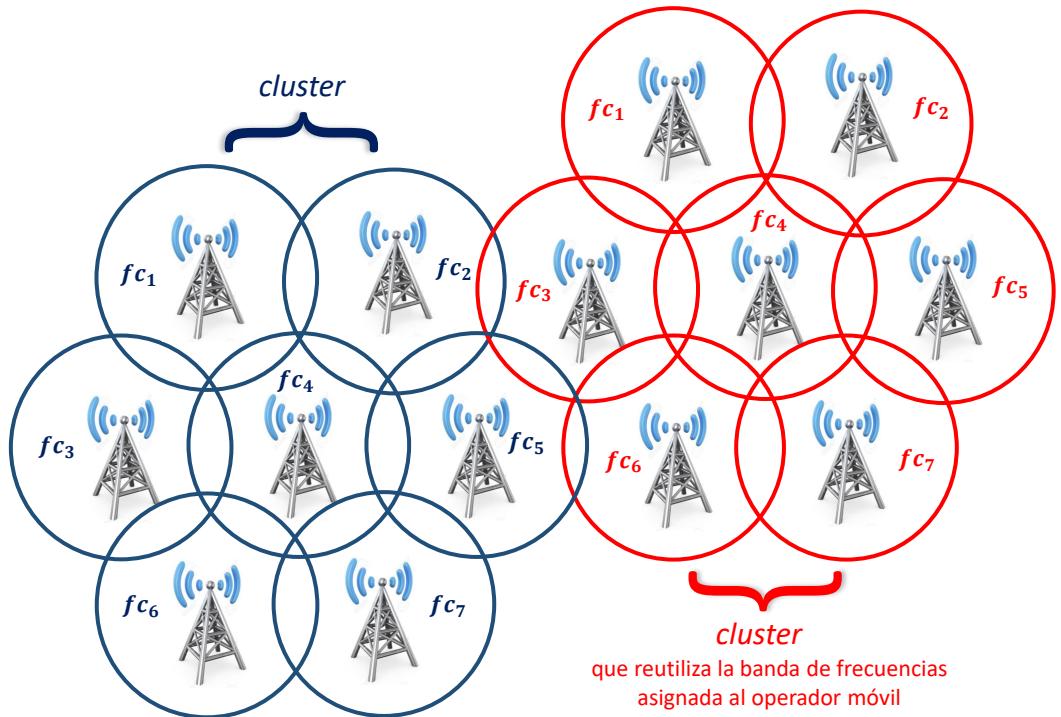
Se combina TDMA con CDMA.

Figura 29: Distribución del espectro combinando TDMA con CDMA.



La figura 29 muestra cómo se aprovecha la banda de frecuencias destinada a una radio base. En UMTS, esa banda es de 5 MHz. Surge la pregunta lógica ¿Cómo se podría representar en esa gráfica el uso que las diferentes radio bases hacen del espectro? Recordemos que en los sistemas basados en TDMA (como GSM y D-AMPS) la banda de frecuencias asignada a un operador es dividida en N subbandas centradas en las frecuencias de las portadoras  $f_{c1}, f_{c2}, f_{c3}, \dots, f_n$  que se asignan a N celdas aledañas de modo que no se presenten interferencias perceptibles por los usuarios de cualquier celda. Se conforma entonces, un clúster de N celdas, como se muestra en la figura 30:

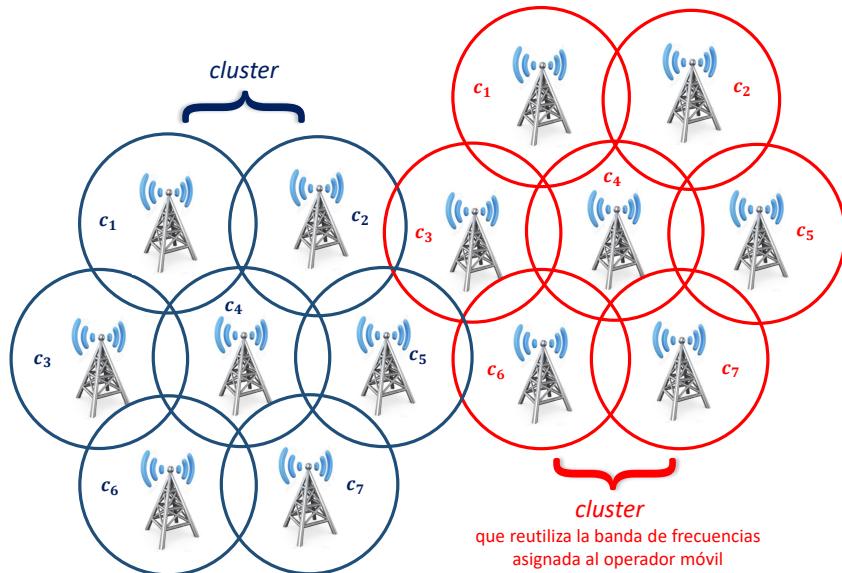
Figura 30: Cluster de N celdas.



#### 9.9.5. El reúso en UMTS

Se usan secuencias de pseudoruido (PN – del inglés *Pseudo - noise*) para distinguir las celdas.

Figura 31: El reúso en UMTS



De modo que, admirablemente, con UMTS un operador móvil podría contar mínimamente con 10 MHz (5 MHz para el *Uplink* y 5 para el *Downlink*, si usa duplexación FDD (la mayoría lo hace)). No necesita subbandas de frecuencias para conformar clusters de celdas, pues UMTS reutiliza una banda de 5MHz mediante códigos de Pseudoruido.

Cabe aclarar que una banda tiene en todo caso una capacidad limitada de tráfico dada por la limitación de potencia de los transmisores usados principalmente en las radio bases. Cada usuario conectado demanda parte de esa energía que emite la radio base de manera proporcional a la velocidad de datos del servicio que el usuario utiliza. Por ello, no es raro que los operadores tengan en la práctica varias bandas UMTS. Cuando esto ocurre, pudiera decirse que el multiacceso en UMTS combina CDMA con TDMA y con FDMA, aunque a la combinación de todo eso, en la práctica, se conoce simplemente como WCDMA.

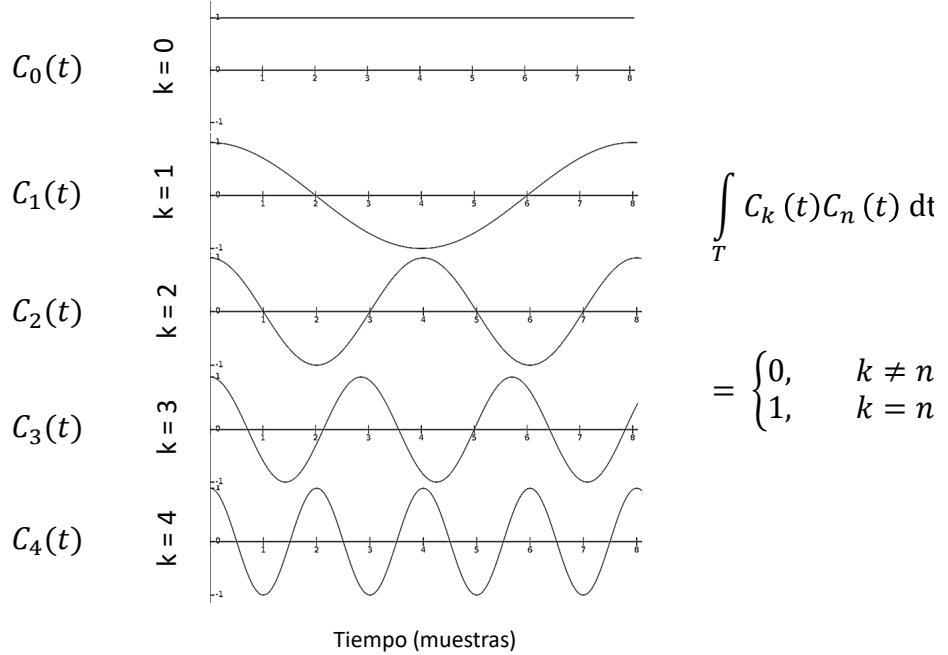
#### 9.9.6. Respiración de las celdas en UMTS

Este tema se ha dejado para que el estudiante lo investigue y ponga a prueba los conocimientos adquiridos para comprender fácilmente otros. Durante esta investigación no olvide que: una radio base y también un terminal tienen potencia limitada; a diferencia de otros sistemas. UMTS busca atender con calidad similar a un usuario que está cerca de la radio base y a otro que está lejos de ella.

### 9.10. OFDM – OFDMA

OFDM es un multiplexado por División de Frecuencias (FDM), que usa portadoras que son ortogonales entre sí. La figura 32 muestra 4 portadoras ortogonales.

Figura 32: OFDM

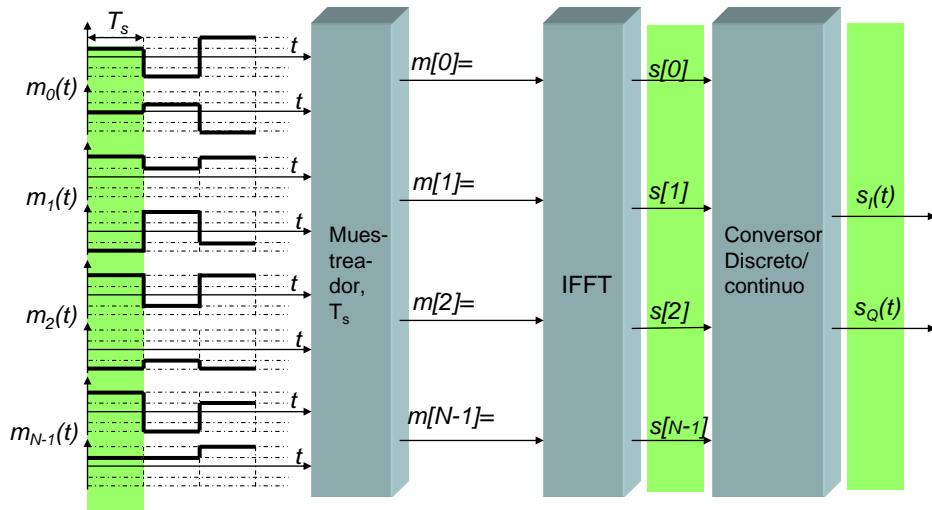


Todas las señales senoidales cuyas frecuencias sean múltiplo entero de una cierta frecuencia  $f_0$ , son ortogonales entre sí.

### 9.10.1. Realización práctica de OFDM

Haber descubierto que una señal OFDM se puede producir usando el algoritmo FFT (*Fast Fourier Transform*) es una de las causas del amplio uso que ha tenido en las comunicaciones de Cuarta Generación (4G), que se han materializado en el sistema LTE (*Long Term Evolution*).

Figura 33: sistema LTE (*Long Term Evolution*)



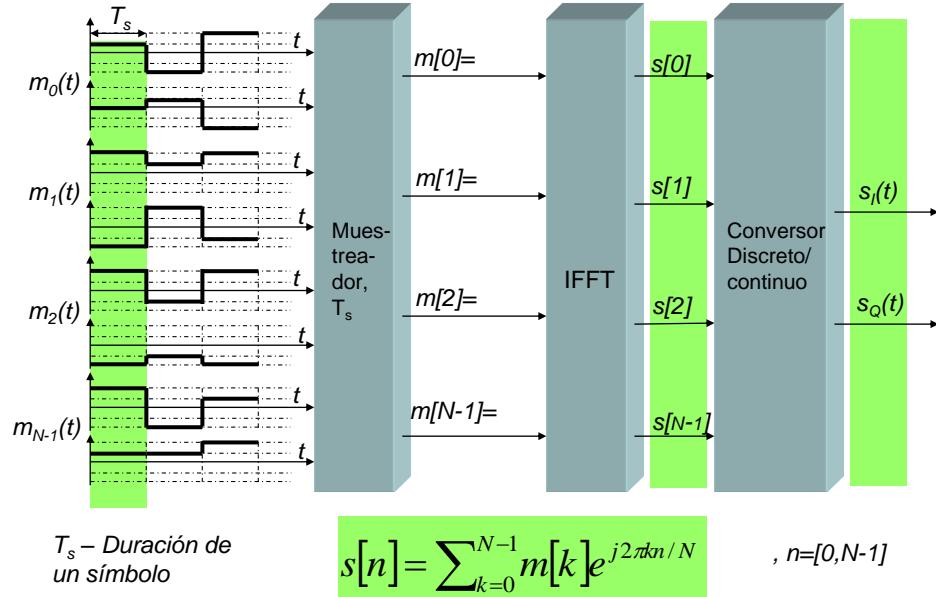
$m_0(t), m_1(t), \dots, m_{N-1}(t)$  son las señales a multiplexar, ellas pueden ser señales reales binarias, pero también pueden ser señales digitales moduladas en su versión de envolvente compleja, como se muestra en la figura, donde cada señal tiene una componente I y una Q y por lo tanto trae símbolos de duración  $T_s$ .

El muestreador hace que solo una muestra por cada símbolo pase al siguiente bloque llamado IFFT (*Inverse Fast Fourier Transform*). Este bloque ejecuta la operación inversa de Representación en Series de Fourier Discreta:

$$\sum_{k=0}^{N-1} m[k] e^{j \frac{2\pi k n}{N}}, n = [0, N-1] \quad (9.11)$$

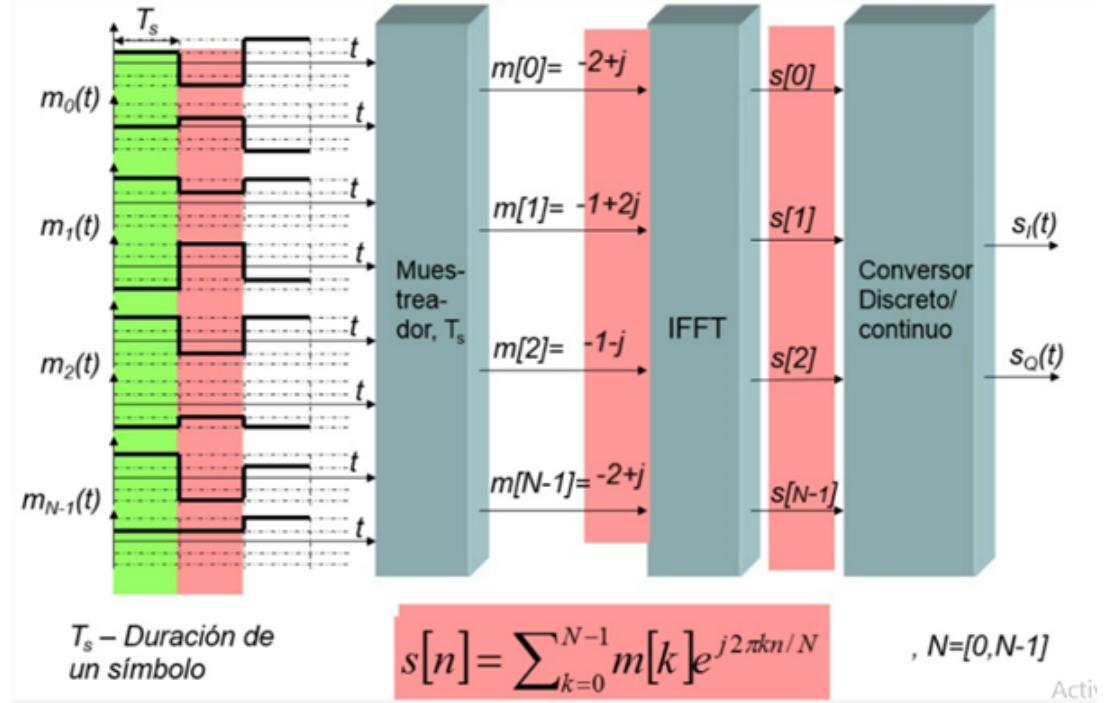
La figura 34 muestra el paso del primer símbolo de cada mensaje por los bloques del multiplexor. La IFFT recibe ese paquete y entrega N muestras de la señal multiplexada.

Figura 34: Paso del primer símbolo por los bloques del multiplexor.



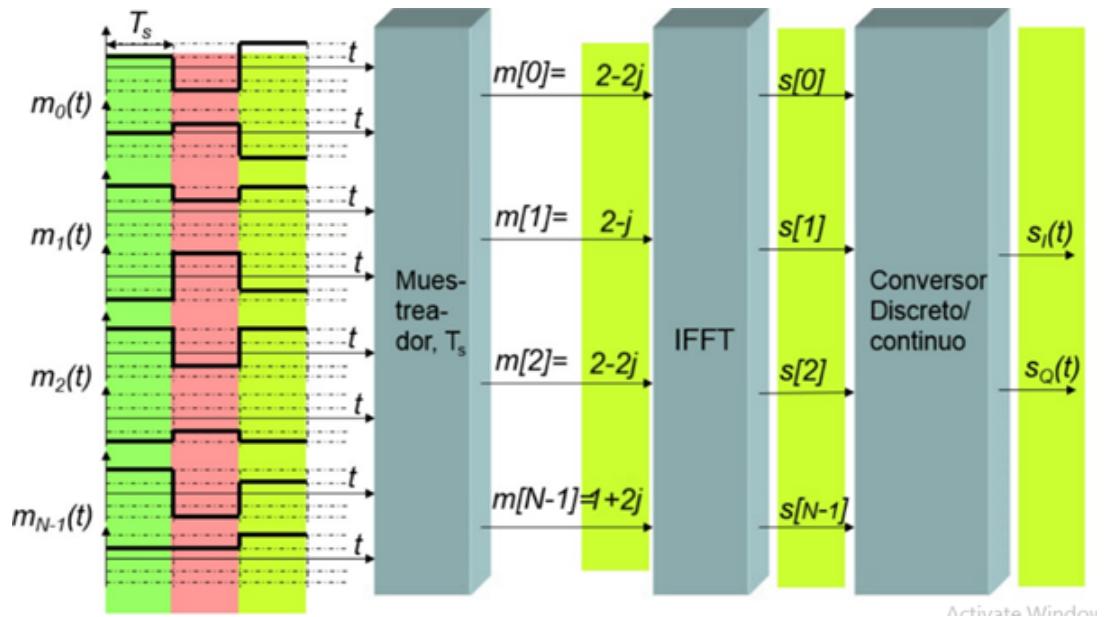
Lo mismo se repite para el segundo símbolo como lo muestra la figura 35.

Figura 35: Paso del segundo símbolo por los bloques del multiplexor.



Se produjo otro paquete de  $N$  muestras de la señal  $s[n]$ . Esas muestras van siendo organizadas en serie y convertidas en una señal continua por el bloque “Conversor Discreto/Continuo”.

Figura 36: Paso del  $N$  símbolo hasta llegar al bloque “Conversor Discreto/Continuo”.



Como la señal discreta es:

$$s[n] = \sum_{k=0}^{N-1} m[k] e^{j \frac{2\pi k n}{N}}, \quad n = [0, N-1] \quad (9.12)$$

Es importante notar que con un sola muestra  $m[k]$  de todos los usuarios se obtienen  $N$  muestras de la señal  $s[n]$ . Eso significa que durante un periodo  $N$ , la muestra  $m[k]$  se mantiene constante. Luego,  $s[n]$

varía N veces más rápido que  $m[k]$ . Sea  $m[n]$ , n una versión de  $m[k]$  sobre muestrada N veces, o sea que por cada valor  $m[k]$  se produce  $m[k,0] = m[k], , m[k,1] = m[k], \dots , m[k,N-1] = m[k]$ , entonces se justifica la siguiente expresión.

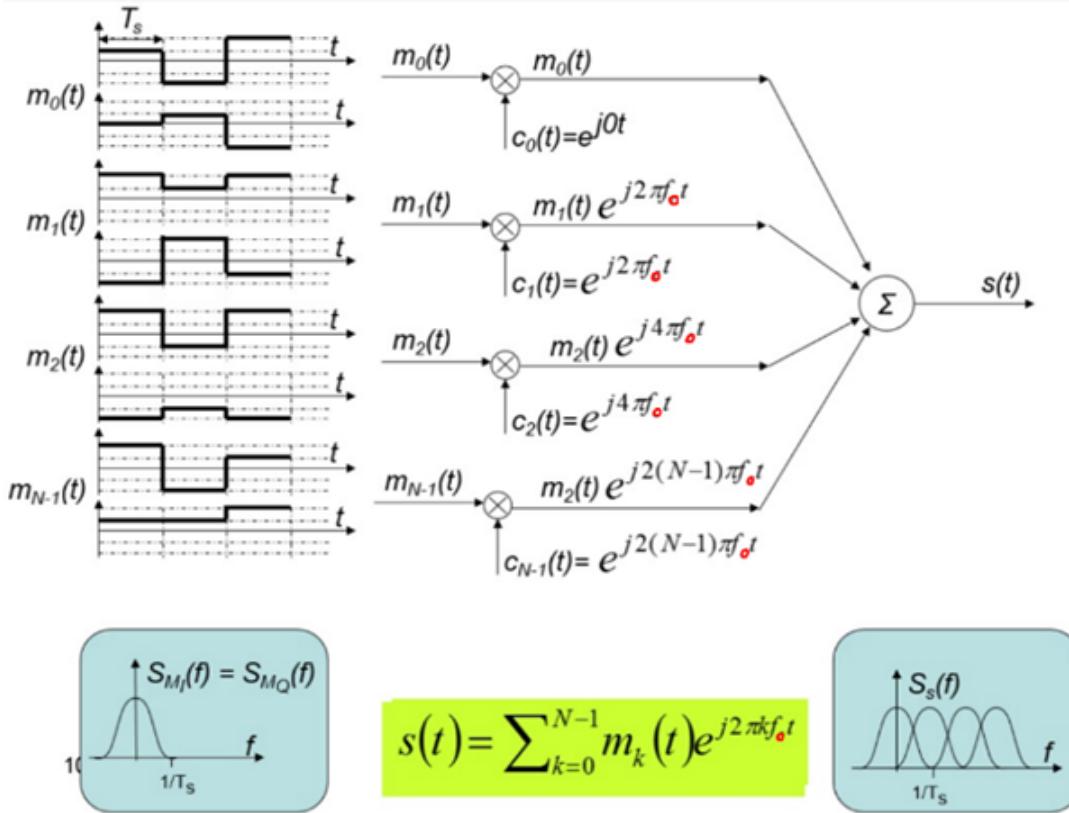
$$s[n] = \sum_{k=0}^{N-1} m[k,n] e^{j\frac{2\pi kn}{N}}, n = [0, N-1] \quad (9.13)$$

Entonces puede deducirse que si esa señal se hace pasar por un conversor discreto a continua, se obtiene:

$$s[n] = \sum_{k=0}^{N-1} m_k(t) e^{j2\pi k f_0 t}, f_0 = \frac{1}{T_s}, t = [0, \infty] \quad (9.14)$$

Esta fórmula nos revela que esta interconexión es equivalente a la figura 37.

Figura 37: Efecto de Aliasing.



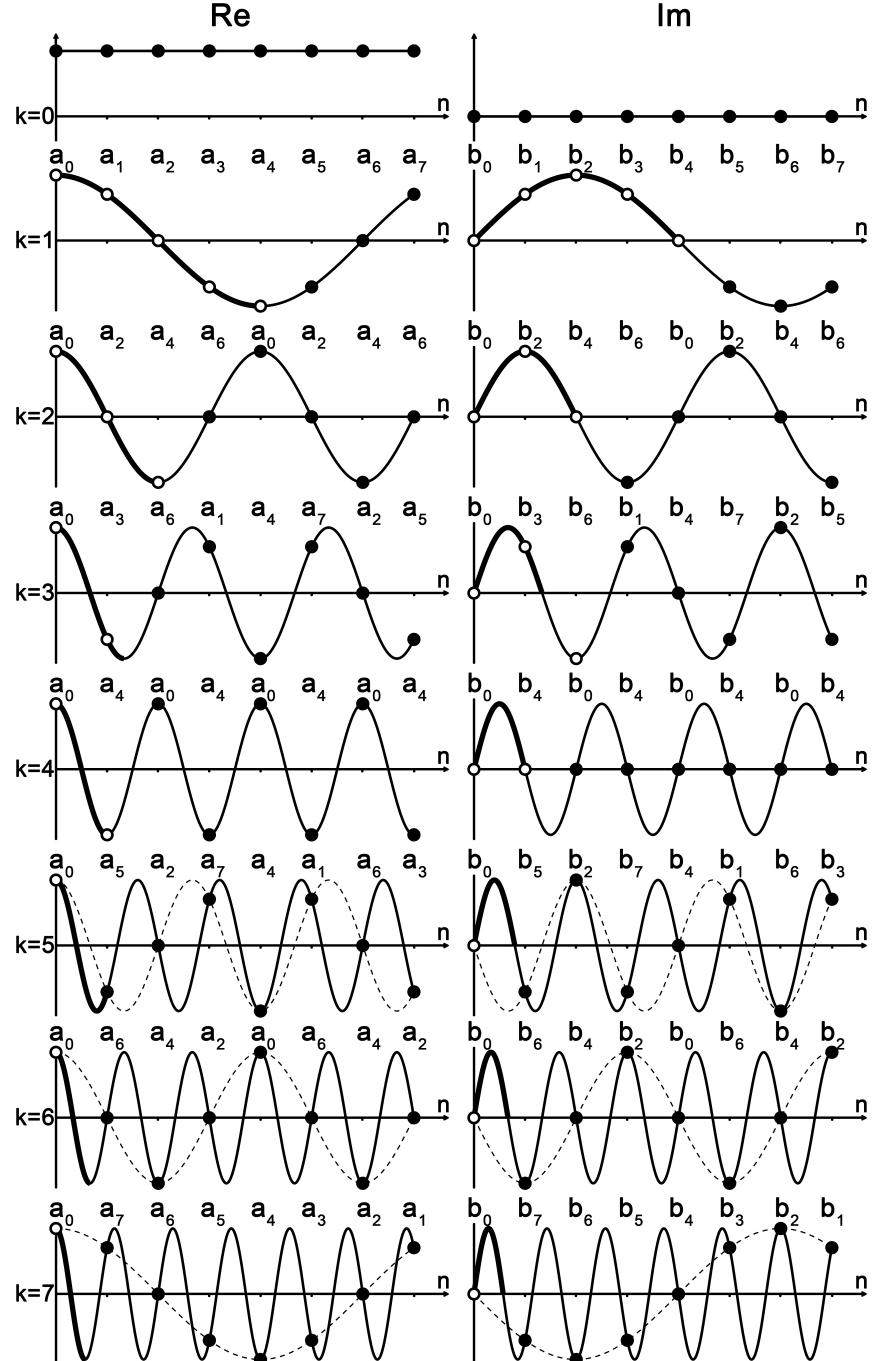
Efecto del aliasing.

Es importante tener en cuenta los efectos del aliasing que se presenta con las señales senoidales discretas, como se muestra en la siguiente figura, en la cual puede deducirse que  $c_7[n] = c_1[n]$ , donde \* se usa para señalar que es el conjugado. Si estas dos señales se miran en un plano complejo, veremos que son vectores rotantes que giran a una misma velocidad, pero en direcciones contrarias. Por lo tanto también puede decirse que:

$$c_7[n] = e^{j2\pi 7n/N} = C_{-1}[n] = e^{j2\pi n/N} \quad (9.15)$$

Lo mismo ocurre entre  $c_6[n]$  y  $c_2[n]$ , así como entre  $c_5[n]$  y  $c_3[n]$ .

Figura 38: Efecto aliasing.



Como el USRP no mira las frecuencias que están entre 0 Hz y la frecuencia de muestreo  $F_s$ Hz, sino entre  $-F_s/2$  y  $+F_s/2$ , equivale a decir que solo reconoce la existencia de las portadoras  $c_{-N/2}[n], \dots, c_{-1}[n], c_0[n], c_1[n], \dots, c_{N/2}[n]$ . Por esta razón, el espectro OFDM se obtiene centrado en la frecuencia cero como se muestra en la siguiente figura tomada de GNU Radio. **Homero: falta lo más importante: mostrar implementación en GNU radio y resultados**

# *Capítulo 10*

## *Codificación*

### 10.1. Introducción

Ideas a incluir en la introducción:

- En 1950 la televisión a blanco y negro analógica comenzó a extenderse por el mundo y convertirse en uno de los eventos más memorables. En ese momento ya se había extendido el uso del teléfono, pero resultaba muy costoso usarlo en llamadas de larga distancia.
- Pero ya desde 1948, Shannon ya estaba comenzando a usar la palabra bit para llegar a producir la revolución digital que estamos viviendo hoy, donde la información fluye en forma digital por cables o sin ellos.
- En ese momento, la tecnología digital parecía algo innecesario ya que por un lado significaba tecnología más costosa y por el otro requería mayores anchos de banda.
- En realidad, la tecnología digital tiene sus desventajas, pero las ventajas son mayores. El progreso que ha habido en la electrónica, pero también en las técnicas de codificación del canal y en algoritmos de procesamiento digital han sido claves para que las comunicaciones digitales se impongan.

### 10.2. Conceptos previos

#### 10.2.1. Polinomio generador de código

Cualquier código binario puede ser representado mediante un determinado polinomio  $G(x)$ , de cierto grado  $r$ . También puede decirse que cualquier polinomio de estos puede generar un código binario, por eso se habla de un polinomio generador. Por ejemplo, el Polinomio Generador podría ser.

$$G(x) = x^3 + 1, \text{ el cual es de grado es } r = 3 \quad (10.1)$$

$$G(x) = 1x^3 + 0x^2 + 0x + 1x^0 \quad (10.2)$$

El código binario corresponde a los coeficientes: 1001. Los signos de suma en los polinomios generadores son suma por módulo 2. Por eso podríamos decir que  $X^2 + X^2 = 0$ , ya que  $1 \oplus 1 \rightarrow 0$ .

### 10.3. Aritmética de módulo 2

Vale la pena recordar la aritmética de módulo 2, donde las sumas y las restas funcionan como compuertas XOR.

$$0 \oplus 0 \rightarrow 0 \quad (10.3)$$

$$1 \oplus 0 \rightarrow 1 \quad (10.4)$$

$$0 \oplus 1 \rightarrow 1 \quad (10.5)$$

$$1 \oplus 1 \rightarrow 0 \quad (10.6)$$

### 10.3.1. Campos de Galois

## 10.4. Fundamentos de codificación del canal

### 10.4.1. Teorema de Shannon

Ideas a incluir en este capítulo:

- El Teorema de Shannon o Teorema de codificación del canal ruidoso establece las fronteras para la máxima información teórica que se puede transferir a un canal con un cierto nivel de ruido.
- Increíblemente, los últimos límites en la teoría de la información y codificación fueron determinados justo cuando esta ciencia nació, con los primeros artículos de Shannon.
- El resultado más conocido de Shannon es el teorema de la capacidad de un canal. El teorema establece que para muchas clases comunes de canal existe un parámetro que ha sido llamado Capacidad de Canal C, sobre el cual viajan códigos a una tasa  $R < C$  que pueden alcanzar transmisiones con cualquier grado de confiabilidad, pero estos códigos no existen para  $R > C$ . así.
- El teorema también establece que para una canal de ancho de banda B (Hz) y afectado solo por ruido blanco aditivo gaussiano (AWGN del inglés Additive White Gaussian Noise), la capacidad C (en bps) solo depende de dos parámetros: el ancho de banda B y la relación señal a ruido SNR (del inglés Signal to Noise Relationship), así.

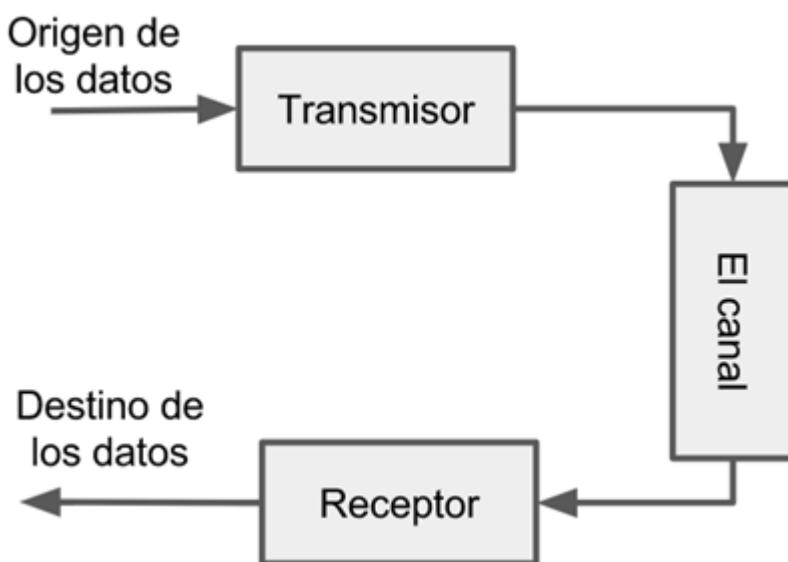
$$C = B \log_2(1 + SNR) \quad (10.7)$$

- Este teorema ha significado un reto para las siguientes generaciones de investigadores que se han enfrascado en lograr que sus algoritmos se acerquen al menos a ese límite establecido por Shannon. En los procesos de codificación y decodificación está en gran medida la clave para alcanzarlo.

### 10.4.2. El proceso de codificación y decodificación digital del canal

El problema de la codificación es planteado por el canal. Como ya se ha dicho en otros capítulos los efectos del canal en el proceso de la comunicación es la razón por la cual aumenta constantemente el número de elementos o nuevos métodos de tratamiento de la información. Pareciera que con lo visto en capítulos anteriores, ya se ha visto todo sobre esos métodos, pero la realidad es que por más soluciones que se implementen siempre existirá la posibilidad de que la información llegue al destino con errores.

Figura 1: Ruta de recepción de un mensaje.

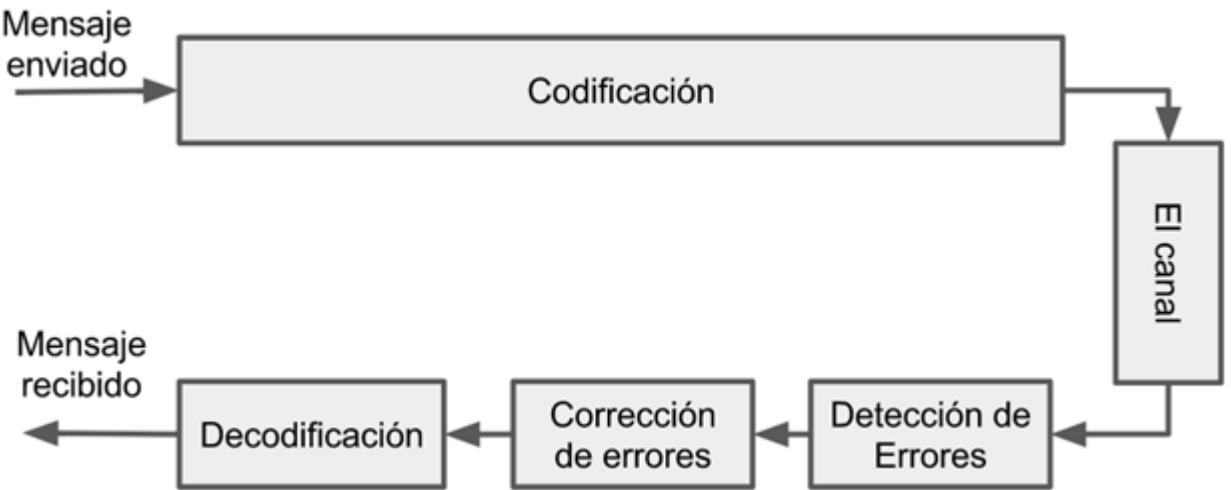


Entonces surgen varias preguntas:

- ¿Es posible reconocer en qué momento se están presentando los errores?
- ¿Qué acciones es posible tomar cuando se han detectado errores?

La solución a estas preguntas está en lo que se conoce como Codificación del Canal. La idea es enviar junto con la información ciertos datos adicionales que permitan detectar la pérdida de bits, lo cual llamaremos codificación y se realiza en el transmisor. Finalmente, en el receptor se debe realizar el proceso de la detección de los errores, su corrección y la decodificación para recuperar los datos enviados, como se muestra en la siguiente figura.

Figura 2: Proceso de codificación



#### 10.4.3. Los retos de la teoría de codificación del canal

El primer reto consiste en poder corregir el máximo número de errores mientras se usa la mínima redundancia (rata) posible. El segundo reto consiste en construir códigos que tengan procesos eficientes de codificación y decodificación.

### 10.5. Códigos de Repetición

#### 10.5.1. El Código de Repetición (3,1)

Para presentarle rápidamente un ejemplo de codificación al lector, a continuación se analiza el ejemplo más sencillo de codificación. Consiste en transmitir 3 veces cada bit. Entonces se tiene una tabla de verdad como la Tabla 10.1, a la cual también se refieren como un diccionario:

Tabla 10.1: Tabla de verdad. Código de repetición (3,1)

| Bit de información | Palabra asignada |
|--------------------|------------------|
| 0                  | 000              |
| 1                  | 111              |

En el receptor, para el ejemplo dado, se pueden recibir 8 posibles versiones de datos a interpretar. En el receptor podría usarse la tabla 10.2 como una especie de tabla de verdad para interpretar la información útil:

Tabla 10.2: Tabla de verdad. Código de repetición (3.1) desde la perspectiva del receptor

| Tribits recibidos | Bit Interpretado | Número de Errores identificados |
|-------------------|------------------|---------------------------------|
| 000               | 0                | 0                               |
| 001               | 0                | 1                               |
| 010               | 0                | 1                               |
| 100               | 0                | 1                               |
| 111               | 1                | 0                               |
| 110               | 1                | 1                               |
| 101               | 1                | 1                               |
| 011               | 1                | 1                               |

**Ventajas de esta codificación:** es muy fácil codificar y decodificar

**Desventaja:** Baja rata  $\frac{1}{3}$

En el sitio web del libro, sección Multimedia, sub sección codificación se tiene al menos un vídeo sobre este tipo de codificación

### 10.5.2. El Código de repetición basado en una matriz 2x3

El código se obtiene a partir de una matriz G Por ejemplo

$$G = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \quad (10.8)$$

Como la matriz es de dimensión 2x3, el el mensaje se organiza en bloques de a 2 bits o símbolos.

$$m = \mu_1, \mu_2 \in F^2 \quad (10.9)$$

Mediante una multiplicación matricial se obtiene la señal a transmitir.

$$T = m \cdot G \quad (10.10)$$

Para el ejemplo

$$T = (\mu_1, \mu_2) \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} (\mu_1, \mu_2, \mu_1 + \mu_2) \quad (10.11)$$

como el último componente es la suma de los dos primeros, este puede indicar si hay o no error, por ejemplo si  $u_1 = 1$ ,  $u_2 = 1$ , entonces para el último miembro debe ocurrir que  $u_1 \oplus u_2 = 0$ .

El código es el conjunto de todas las posibles combinaciones de símbolos a transmisiones. Para el ejemplo dado el código tiene 4 posibles palabras, pues al organizar la información por díbits, se tendrán  $2^2 = 4$  posibles díbits diferentes. Para este ejemplo, también se dice que T pertenece a un espacio  $F_2^3$ . Eso significa que en ese espacio pueden haber  $2^3 = 8$  posibles combinaciones, pero que solo 4 son de información válida.

La ventaja con respecto al código del capítulo anterior, es que por cada 2 componentes del mensaje se envía en 3 elementos, entonces la Rata es  $\frac{2}{3}$ . La desventaja: se puede detectar 1 error, pero no se puede corregir.

## 10.6. Principales parámetros de la codificación

### 10.6.1. Distancia de Hamming entre dos palabras

Se usa normalmente para comparar una palabra digital original con respecto a otra que puede ser una versión codificada o una versión recibida con errores. La distancia de Hamming es el número de bits que difieren en esas dos palabras. En la figura 3 se muestran dos palabras a comparar.

Figura 3: Comparación entre dos palabras.

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

En ellas, la distancia de Hamming es 4, lo cual significa que se necesitan 4 errores para transformar una palabra en la otra. La manera en que se simboliza matemáticamente la distancia de Hamming es la siguiente:

$$d(101101, 110100) = 3$$

Significa que la distancia entre los códigos 101101 y 110100 es 3.

### 10.6.2. Distancia de Hamming de un código

Supongamos que se desea emitir un mensaje que solo tiene las vocales abiertas: a,e,o. De manera que a cada vocal le asignamos un código, entonces tenemos por ejemplo las siguientes tres palabras 100, 111, 011. Para cada pareja de palabras de un código se pueden tener diferentes distancias, para este ejemplo tenemos:

$$d(100,111)=2$$

$$d(100,011)=3$$

$$d(111,011)=1$$

La Distancia de Hamming para el código dado, de varias palabras es el mínimo de todas las distancias. Para el ejemplo dado es 1. Es claro que entre más grande sea esta distancia, mejor será el código, pues es más grande la diferencia entre las palabras, de modo que los errores que puedan resultar son más fácilmente identificables.

Veamos un ejemplo de codificación usando una matriz 3x6.

- El mensaje binario se divide en bloques de 3 bits, de manera que uno de esos bloques podría representarse mediante un símbolo  $m$ , por ejemplo:

$$m = [110] \in F^3 \quad (10.12)$$

- La matriz de codificación a usar es la siguiente:

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad (10.13)$$

- La señal a transmitir sería entonces  $T = u \times G$ .

Ejemplo para los símbolos 110 y 101.

$$(1, 1, 0) \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = (1, 1, 0, 0, 1, 1) \quad (10.14)$$

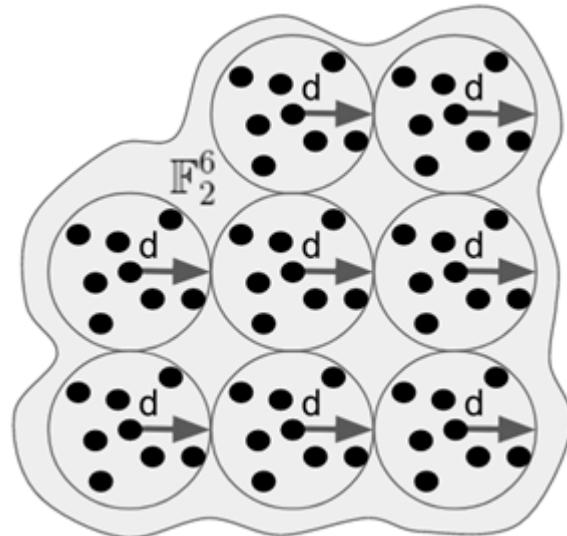
$$(1, 0, 1) \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = (1, 0, 1, 1, 0, 1) \quad (10.15)$$

Podríamos hacer lo mismo para  $2^3 = 8$  palabras, se dice que el código pertenece a un  $F_2^6$ , es decir un espacio que tiene  $2^6 = 64$  posibles elementos, pero sólo  $2^3 = 8$  son parte del código. Si realizamos el análisis para todas las posibles palabras, descubriremos que la distancia de hamming es igual a 3, pues al menos 3 bits serán diferentes entre cualquiera de las componentes del código. Si una palabra de un código C se representa como x y otra como y, se dice que la distancia de Hamming es:  $h(x,y) =$  el número de símbolos de x que difieren de y para todo un código C, la distancia es:

$$d(C) = \min h(x,y) \mid x, y \in C, x \neq y \quad (10.16)$$

Podemos imaginar que el campo  $F_2^6$  es el que se presenta en la siguiente figura.

Figura 4: El espacio  $F_2^6$  Los puntos negros representan las posibles palabras que pueden resultar en el campo y los círculos representan las palabras del código



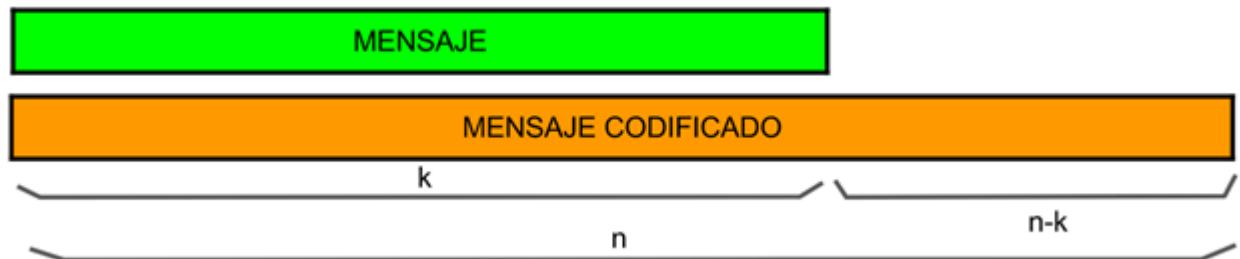
En la gráfica podemos observar que cada palabra del código tiene una cobertura de radio d. Quiere decir que si al receptor llega la señal con errores, en todo caso cada palabra recibida caerá dentro del área de cobertura de una palabra del código y es lo que permitirá corregir el error. Por eso es importante que el código cuente con coberturas grandes, que abarquen todo el espacio.

La gráfica anterior se refuerza si observamos que un bloque  $(n,k)$ , que es un código  $C(n,k)$  y proviene de una matriz G de  $n \times k$ , es un subespacio de un espacio vectorial  $F^n$  y las filas de G son la base de C.

### 10.6.3. La rata del código.

En cualquier caso de codificación se tiene un mensaje de  $k$  que luego de la codificación sale con  $n$  palabras.

Figura 5: Rata del código.



La rata de codificación  $R_c$  señala qué tan eficiente es el código en términos de la cantidad de información redundante que es necesario transmitir.

$$R_c = \frac{k}{n} \quad (10.17)$$

#### 10.6.4. La ganancia del código

Es la diferencia entre los niveles de relación señal a ruido (SNR) del sistema que no incluye codificación y el que la incluye.

### 10.7. Clasificación de los códigos según sus características

#### 10.7.1. Códigos lineales

Se dice que un código es lineal cuando se cumple lo siguiente:

$$\begin{aligned} &\text{si} \\ m_1 &\rightarrow c_1 \\ m_2 &\rightarrow c_2 \end{aligned}$$

Entonces se cumple que.

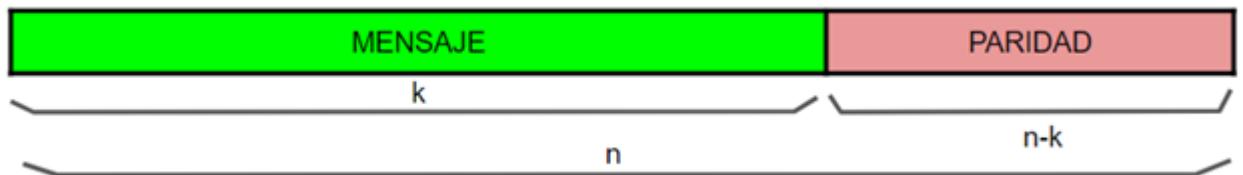
$$m_1 \oplus m_2 \rightarrow c_1 \oplus c_2$$

Donde  $m$  es una palabra del mensaje de  $k$  bits de información;  $c$  es una palabra de código de  $n$  bits de información;  $\oplus$  significa suma de módulo 2 bit a bit. De modo que la suma de cualquier dos palabras de código es una palabra de código que corresponde a la suma de las correspondientes palabras del mensaje.

#### 10.7.2. Los códigos de bloques

Los códigos de bloques son los vistos en los anteriores ejemplos, donde a cada palabra de un mensaje le corresponde exclusivamente una palabra de código que usualmente se agrega al final de la palabra del mensaje en forma de una palabra de paridad.

Figura 6: Código de bloques.



#### 10.7.3. Los códigos convolucionales

Pueden ser vistos como una variación de los códigos de bloques si se tiene en cuenta lo siguiente:

- Tanto en los códigos de bloques como en los convolucionales, un mensaje se divide en bloques de bits o palabras que se pueden representar como  $u_2, u_1, u_0$ .
- En los códigos de bloques se usa una matriz  $G$  que sirve de base para obtener la señal a transmitir, de modo que:

$$\dots u_2, u_1, u_0 \xrightarrow{G} (\dots V_2 = u_2G, v_1 = u_1G, v_0 = u_0G) \quad (10.18)$$

- En algunas fuentes se usa la representación polinomial que es la siguiente:

$$\dots + u_2D^2 + u_1D + u_0 \xrightarrow{G(D)} \dots + u_2GD^2 + u_1GD + u_0G \quad (10.19)$$

- Para el caso de los códigos convolucionales, lo que se tiene es una matriz para cada posición de la palabra de bits del mensaje  $G_0, G_1, \dots, G_s$ . También puede decirse que se tiene un polinomio de matrices:

$$G_0 + G_1 + \dots + G_s D^s \quad (10.20)$$

Donde D puede ser visto como un retardo unitario (del inglés Delay). o sea que  $G_0$  se aplica a la primera palabra del mensaje que llega,  $G_1$  a la que llega con un retraso, etc.

- De modo que, en la codificación convolucional, la señal a transmitir se calcula así:

$$(\dots, u_2, u_1, u_0) \xrightarrow{G(D)} (\dots, v_2 = u_2 G_0 + u_1 G_1 + u_0 G_2, v_1 = u_1 G_0 + u_0 G_1), v_0 = u_0 G_0 \quad (10.21)$$

- Como puede verse, cada palabra a transmitir, se obtiene usando varias palabras mientras se desplazan las matrices usadas, lo cual puede ser visto como una convolución
- La representación polinomial para decir lo mismo anterior es la siguiente:

$$\dots + u_2 D^2 + u_1 D + u_0 \xrightarrow{G(D)} \dots + (u_2 G_0 + u_1 G_1 + u_0 G_2) D^2 + (u_1 G_0 + u_0 G_1) D + u_0 G_0 \quad (10.22)$$

- Lo destacable de los códigos convolucionales es que una palabra a transmitir lleva información no solo de la palabra presente del mensaje sino también de palabras anteriores. Por ejemplo la palabra a transmitir  $v_2$  se obtiene no solo a partir de la palabra del mensaje  $u_2$  sino de  $u_1$  y de  $u_0$ . Por eso se dice que los códigos convolucionales tienen memoria, es decir, guardan información de palabras transmitidas anteriormente.

Ejemplo de una matriz para un código convolucional.

$$G(D) = \begin{bmatrix} D^2 & + & 1 \\ D^2 & + & D & + & 1 \end{bmatrix} \quad (10.23)$$

Implementación para la anterior matriz [Vídeo](#)

#### 10.7.4. Códigos binarios versus no binarios.

### 10.8. Técnicas de detección de errores

#### 10.8.1. Chequeo de repeticiones.

#### 10.8.2. Chequeo de paridad (Parity Check).

Ha sido uno de los primeros métodos usados para detectar errores de un mensaje enviado desde un transmisor hasta un receptor pasando por un canal que puede producir interferencia y consecuentemente pérdidas de bits. En mensaje es visto de manera binaria, el mensaje a transmitir es visto como un bloque de bits, usualmente 8 bits, pero algunos de ellos, usualmente solo uno, el último, se destina para el proceso de control de paridad.

El proceso de insertar los bits de paridad:

- Ocurre en el transmisor.
- contar el número de unos que hay en la información que se aloja en el bloque.
- Si el número de unos es impar, se agrega un uno en el espacio destinado al control de paridad, de lo contrario, en ese espacio se agrega un cero.

El proceso de control de errores:

- Ocurre en el receptor.
- A cada bloque recibido se le cuentan los unos y siempre deberá dar un número par, de lo contrario, se habrá detectado un error [Vídeo](#)

### 10.8.3. Chequeo de polaridad

## 10.9. El mecanismo de corrección de Errores

### 10.9.1. Solicitud Automática de Repetición (ARQ)

Solicitud automática de repetición (ARQ del inglés Automatic Repeat Request). Es la solución más obvia, donde la Corrección de errores se implementa como un sistema de solicitud de retransmisión parcial de la información.

### 10.9.2. Corrección de Errores hacia adelante (FEC)

Control de Errores hacia adelante (FEC del inglés Forward Error Correction). Consiste en la posibilidad de implementar la Corrección de errores como un proceso de recuperación de la información a partir de información redundante que ha sido emitida en el proceso de codificación realizado en el transmisor. La Corrección de Errores hacia adelante (FEC del inglés Forward Error Correction) es una técnica usada en las comunicaciones digitales para darle la posibilidad al receptor de identificar un cierto número de errores en la señal de información recibida y corregirlos sin necesidad de recurrir a una retransmisión de la información, aunque a costa de usar un mayor ancho de banda en el canal. Por lo tanto es un elemento esencial en los sistemas de comunicación digital de tiempo real de voz, vídeo y otros servicios donde no resulta conveniente usar métodos de transmisión.

La idea central consiste en que el transmisor codifica el mensaje de una manera especial, usando cierta redundancia, para lo cual usa un Código de Corrección de Errores (ECC- Error Correcting Code). La idea fue presentada por primera vez en 1940 por el matemático americano Richard Hamming cuando inventó el ECC conocido como Hamming Code. El ejemplo más sencillo es precisamente el Código de Repetición (3,1) explicado anteriormente.

Algunos codificadores FEC brindan la posibilidad de calcular la rata de errores de bit (BER del inglés Bit Error Ratio), lo cual resulta valioso para otros algoritmos de sintonización.

## 10.10. Propiedades para la detección y corrección de Errores

**Para la detección de errores:** Para detectar  $d$  errores de un bit entre dos palabras, es necesario un código con una distancia de Hamming de al menos  $d+1$ . O sea, que con una distancia de Hamming de  $d$  se pueden detectar  $d-1$  errores.

- Para corregir  $d$  errores de un bit entre dos palabras es necesario un código con una distancia de Hamming de al menos  $2d+1$ . O sea que con una distancia de Hamming  $d$  se pueden corregir  $(d-1)/2$  errores.
- Ejemplo  $C=000000000,000001111,1111100000$
- Distancia de Hamming =5
- Se pueden detectar  $d-1=4$  errores
- Se pueden corregir  $(d-1)/2 = 2$  errores

Teorema básico de corrección de errores:

- Un código  $C$  permite detectar hasta  $s$  errores si.

$$dist(C) \geq S + 1 \quad (10.24)$$

- Un código  $C$  permite corregir hasta  $t$  errores si.

$$dist(C) \geq 2t + 1 \quad (10.25)$$

## 10.11. Códigos de redundancia cíclica (CRC)

Código de chequeo de redundancia cíclica (CRC, del inglés - Cyclic and Redundancy Check).

- Originalmente se tiene una secuencia de datos Por ejemplo: 1111111000000011110000000111
- Esa secuencia se divide en bloques de n elementos. Por ejemplo si  $n=7$  y se continúa con el ejemplo anterior, se tienen 4 bloques cada uno con un espacio al final para agregar k elementos.

Tabla 10.3: Ejemplo de código de redundancia cíclica.

|         |  |         |  |         |  |         |  |
|---------|--|---------|--|---------|--|---------|--|
| 1111111 |  | 0000000 |  | 1111000 |  | 0000111 |  |
|---------|--|---------|--|---------|--|---------|--|

- Por eso se habla de un código tipo

$$C(n, k) \quad (10.26)$$

- El espacio en blanco a la derecha de cada bloque es para agregar información clave para facilitar la detección de errores. Esa información se deduce a partir un código binario que sea conocido tanto en el transmisor como en el receptor. A menudo no se habla de una palabra binaria en sí, sino de un polinomio generador de esa palabra.

### 10.11.1. Proceso de codificación usando CRC

- El Proceso de codificación de una secuencia de datos usando un polinomio generador  $G(x)$  en el transmisor es el siguiente:

- Se selecciona el polinomio generador Por ejemplo  $G(x) = x^3 + 1$
- Se prepara la secuencia de datos Por ejemplo:

1111111

Se supone que a ella le corresponde un polinomio que se puede representar como  $M(x)$ .

- Tanto  $M(x)$  como  $G(x)$  se expresan como secuencias de bits. Para el ejemplo usado tenemos  
 $M(x) \rightarrow 1111111$   
 $G(x) \rightarrow 1001$   
 $T(x)$  será el polinomio de la señal a transmitir

- Se agregan r ceros a la secuencia  $M(x)$ , con lo cual se obtiene  $Q(x)$ . Continuando con el ejemplo.  
 $Q(x) \rightarrow 1111111000$

- Se calcula el código CRC como el residuo que resulta al dividir  $G(x)$  por  $Q(x)$ . Para nuestro ejemplo tenemos que:

$G(x)/Q(x)$

1001 / 1111111000.

$$\begin{array}{r}
 \oplus 1001 \\
 0110111000 \\
 \oplus \underline{1001} \\
 010011000 \\
 \oplus \underline{1001} \\
 00001000 \\
 \oplus \underline{1001} \\
 0001
 \end{array}$$

El código CRC son las r últimas cifras del residuo, es decir 001 y se puede representar como  $R(x)$

- Se determina el resultado o señal a transmitir  $T(x)$  adicionando el residuo obtenido en el punto anterior a  $Q(x)$ .

Para nuestro ejemplo  $T(x) = Q(x) + R(x)$

$$\begin{array}{r}
 1111111000 \\
 001 \\
 \hline
 1111111\textcolor{red}{001}
 \end{array}$$

- Vemos que finalmente  $T(x)$  es el mismo mensaje complementado con el código CRC.

### 10.11.2. Proceso de detección de errores usando Check sum

Se realiza un proceso similar al visto en el proceso de codificación CRC. Si el residuo es 0 significa que no hay errores, de lo contrario hay errores. En calidad de ejemplo, veamos cómo se trataría la señal recibida del ejemplo anterior:

$$\begin{array}{r}
 1111111\textcolor{red}{001} \\
 \oplus \underline{1001} \\
 0110111001 \\
 \oplus \underline{1001} \\
 010011001 \\
 \oplus \underline{1001} \\
 00001001 \\
 \oplus \underline{1001} \\
 0000
 \end{array}$$

En este ejemplo vemos que el residuo es cero, cuando la señal transmitida es idéntica a la recibida, es decir, cuando no se presentan errores.

Veamos un ejemplo, cuando se pierde un bit, por ejemplo cuando al receptor llega la señal: 1011111001

$$\begin{array}{r}
 1011111001 \\
 \oplus \underline{1001} \\
 0010111001 \\
 \oplus \underline{1001} \\
 00101001 \\
 \oplus \underline{1001} \\
 001101 \\
 \oplus \underline{1001} \\
 0100
 \end{array}$$

Vemos que se obtiene un residuo diferente de cero cuando la señal recibida es diferente a la transmitida.

### 10.11.3. Cualidades del CRC

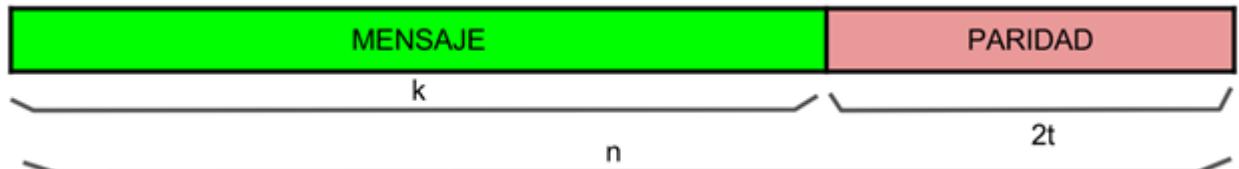
- Detecta todos los errores de 1 bit.
- Si se escoje de manera apropiada  $G(x)$  se detecta.
- Todo error de 2 bits.
- Todo error de bits impares.
- Todo error de ráfagas con una longitud que es menor al grado r
- Es fácil de implementar en hardware binario.
- Es fácil de analizar matemáticamente.
- Es bueno para detectar errores comunes, causados por ruido en el canal de transmisión.

## 10.12. La codificación de Reed Solomon

Es uno de los tipos de codificación más notables ya que su uso se extendió al disco compacto (CD), el DVD, los discos duros, Blue-rays, otros sistemas de almacenamiento, los códigos QR, la televisión digital, WiMax, Comunicaciones satelitales, entre otros. Fue introducido en 1960 por Irving S. Reed y Gustave Solomon. Estrictamente pertenece al tipo de codificación cíclica no binaria y lo que hace es agregar t símbolos de chequeo a los datos para detectar cualquier combinación de hasta t símbolos erróneos o corregir hasta t/s símbolos. Este código tiene las siguientes cualidades:

- Es un tipo de código lineal sistemáticamente cíclico y es no-binario, usada en FEC. Desde este punto de vista, el chorro de datos del mensaje se divide en bloques de datos, se le agrega a estos bloques una redundancia que solo depende de las entradas actuales.
- Se apoyan en un área de la matemática conocida como Campos de Galois o campos finitos. Estos campos se caracterizan por las operaciones aritméticas (+,-,x) siempre tienen un resultado dentro del campo. Precisamente, porque las herramientas de programación no soportan aritmética de campo de Galois, la implementación tradicional de este código es en hardware.
- Se representa como RS(n,k),

Figura 7: Codificación de Reed Solomon.



donde:  $n = 2^{s-1}$  es la longittud del bloque resultante, s: es el número de bits que tiene cada simbolo del mensaje, k: es el número simblos del mensaje en cada bloque, t es el número de simblos erroneos que es posible corregir en la palabra del código. n es un número entero entre 3 y  $2^s - 1$ ;  $k < n$ , s es entero entre 3 y 16.

- Este código es capaz de corregir tanto errores en ráfagas como datos perdidos (erasures). Se usa principalmente en la corrección de ráfagas de errores.
- La ganancia de codificación de este código: es muy alta, aunque menor que los códigos LDPC y los TURBO códigos. Por esta alta ganancia, el código de Reed Solomon se usa en muchas aplicaciones incluyendo almacenamiento y transmisión de información:
  - Almacenamiento (cintas de almacenamiento, discos compactos,DVD, etc).
  - Comunicaciones móviles.
  - Radioenlaces.
  - Comunicaciones satelitales.
  - Televisión digital
- La información puede estar dada en bits, pero también en bytes o en otro tipo de palabras. Por esa razón es ampliamente usada no solo en las comunicaciones sino también en sistemas de respaldo de información. Véase por ejemplo este [Vídeo](#)

El principio de la codificación consiste en:

- Se organiza la información en una matriz de dimensiones QxQ.

Por ejemplo, si la información se organiza en una matriz 4x4 y está dada en bytes, en cierto momento podrían ser letras del alfabeto como en el siguiente caso:

$$U = \begin{bmatrix} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ M & N & O & P \end{bmatrix}$$

- Contar con una matriz de dimensiones Qx(Q+k), donde el espacio QxQ de la matriz es una matriz de identidad y los demás espacios corresponden al código a usar.

Continuando con el ejemplo anterior, y teniendo en cuenta que la información está dada en bytes, la matriz puede ser la siguiente:

$$G = \begin{bmatrix} 01 & 00 & 00 & 00 \\ 00 & 01 & 00 & 00 \\ 00 & 00 & 01 & 00 \\ 00 & 00 & 00 & 01 \\ 1b & 1c & 12 & 14 \\ 1c & 1b & 14 & 12 \end{bmatrix}$$

- Se obtiene la matriz de la señal a transmitir como  $T=u \times G$ .

Continuando con el ejemplo, la información a transmitir está dada por la matriz siguiente:

$$T = u \times G = \begin{bmatrix} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ M & N & O & P \\ 51 & 52 & 53 & 49 \\ 55 & 56 & 57 & 25 \end{bmatrix}$$

- Se toma la matriz obtenida y se transmite fila por fila.

Continuando con el ejemplo anterior, la señal a transmitir es la siguiente:

|                |   |   |   |   |   |   |   |   |   |   |   |                |   |   |   |    |    |    |    |    |    |    |    |
|----------------|---|---|---|---|---|---|---|---|---|---|---|----------------|---|---|---|----|----|----|----|----|----|----|----|
| A              | B | C | D | E | F | G | H | I | J | K | L | M              | N | O | P | 51 | 52 | 53 | 49 | 55 | 56 | 57 | 25 |
| <i>mensaje</i> |   |   |   |   |   |   |   |   |   |   |   | <i>paridad</i> |   |   |   |    |    |    |    |    |    |    |    |

El principio de la de-codificación, cuando no hay errores consiste en:

- Contar con una matriz que sea el inverso a la matriz usada para la decodificación.

$$\text{Para el ejemplo dado anteriormente, esa matriz es. } G^{-1} = \begin{bmatrix} 01 & 00 & 00 & 00 \\ 00 & 01 & 00 & 00 \\ 00 & 00 & 01 & 00 \\ 00 & 00 & 00 & 01 \\ 8d & f6 & 7b & 01 \\ f6 & 8d & 01 & 7b \end{bmatrix}$$

- Se obtiene la señal recibida en forma matricial como  $r = T X G^{-1}$

Cuando hay errores, la decodificación se realiza en 5 niveles:

- Síndrome de cálculo: dice si ocurrió un error durante la transmisión de datos
- Localización del error: dice donde se presentó el error la magnitud del error
- Evaluación del error: corrige el error.

Tres son los posibles resultados de la decodificación:

- Si  $2s + r < 2t$ , donde s son símbolos con errores, r son símbolos borrados, entonces la palabra de código originalmente transmitida podrá ser siempre recuperada
- El decodificador detectará que no puede recuperar la palabra de código original e indicará este hecho.
- El decodificador decodificará erróneamente y recuperará una palabra de código de manera incorrecta y sin indicación alguna.

Otras aplicaciones de la codificación Reed Solomon.

Los métodos de codificación pueden encontrar una gran cantidad de aplicaciones.

Ejemplo aplicado a bases de datos. El mismo ejemplo anterior, aplicado al respaldo de bases de datos se encuentra en el siguiente video:  
[Codificador Reed Solomon aplicado para el respaldo de bases de datos](#)

Hay también una gran multitud de aplicaciones como es el caso de los QR Codes:  
[Reed Solomon aplicado a Códigos QR](#)

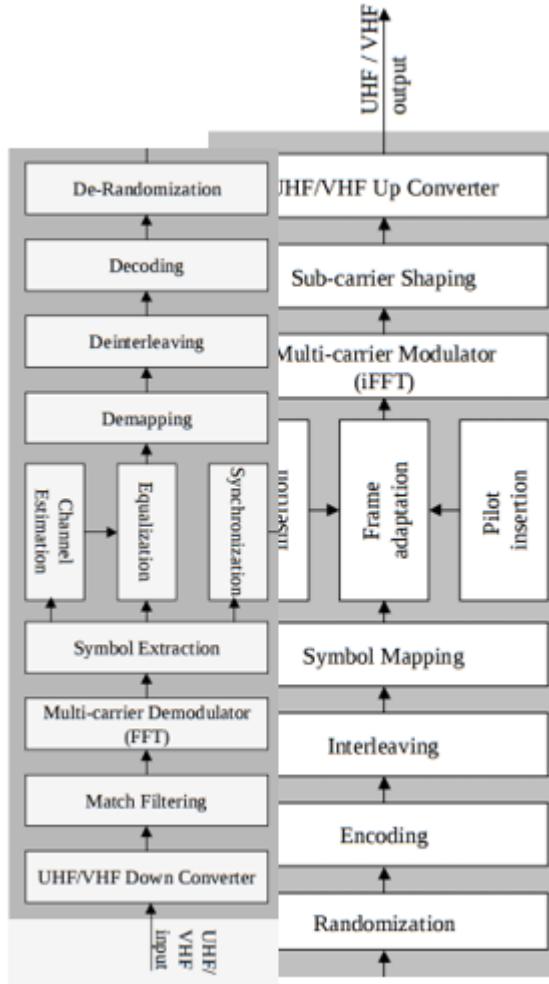
### 10.13. Reed Solomon RS(255,223)en la práctica

En la práctica es común encontrar una configuración del Reed Solomon, donde cada palabra de código que se produce es de 255 bytes, de los cuales k=223 son bytes del mensaje y 2t=32 son bytes de paridad: n=255, k=223, consecuentemente, s=8, 2t=32, t=16

Eso indica que este código permite corregir hasta 16 símbolos erróneos en la palabra de código, en cualquier parte del código y de manera automática.

## 10.14. Modelo de capas de la capa física de dvb-t

Figura 8: Modelo de capas de la capa física de dvb-t.



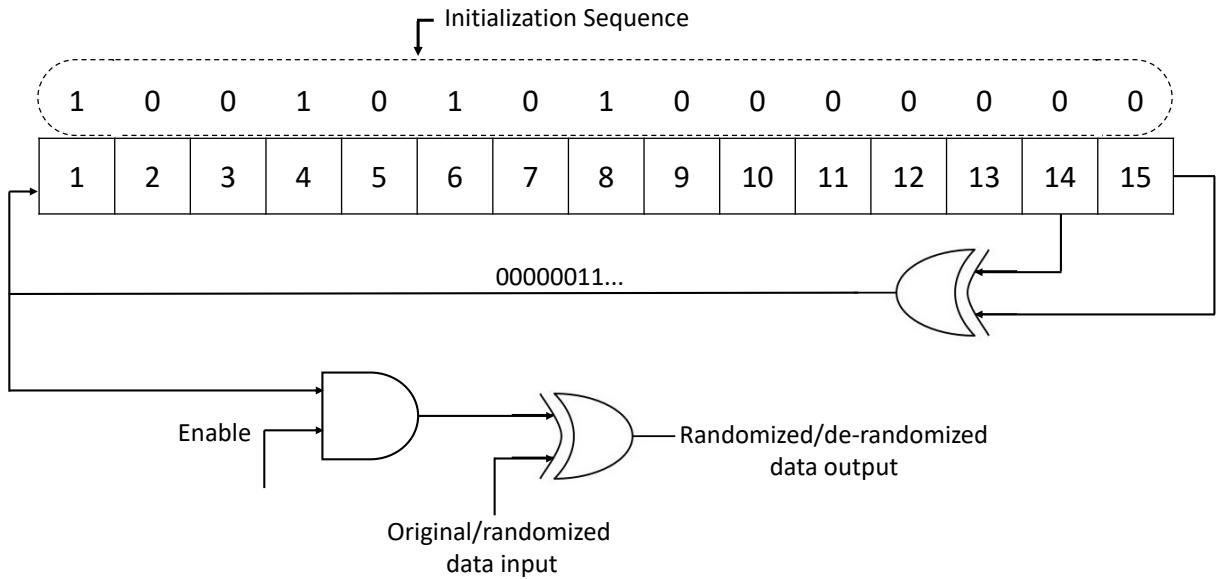
*fuente: ETSI EN 301 958*

## 10.15. Scrambling en DVB-T

El scrambling es un tema que ya ha sido explicado en el capítulo 1. Para el caso de la DVB-T el scrambling tiene las siguientes particularidades:

- La secuencia PN se genera mediante una interconexión de 15 registros, una compuerta XOR y una AND.

Figura 9: Esquema de scrambling



- . Esta interconexión puede entenderse con la teoría que se explica en el libro de Haykin, capítulo 7, sobre generación de señales PN.
- Los registros están conectados en serie y con cada pulso de reloj, el estado de un registro pasa al siguiente registro.
- Los estados de los registros son los de la secuencia de inicialización que aparece en la figura, cuyo polinomio generador es  $p(x) = 1 + x^{14} + x^{15}$
- El esquema es equivalente a la Fig. 1, pero si  $bs(t)$  se cambia de polaridad, o sea que se multiplica por -1.
- Los registros 14 y 15 entregan su estado a una compuerta XOR y su salida va pasando nuevamente al primer registro y al mismo tiempo a una compuerta AND.
- El scrambling ocurre en una compuerta XOR, a la cual entran el código PN y la señal útil.
- El periodo que resulta para la secuencia PN es de  $188 \times 8 = 12032$  bits.
- El primer byte de cada paquete de la señal de video es de sincronización y no debe ser aleatorizado. Sin embargo el generador PN sigue generando bits aunque no se usen en esos bytes específicos. Justamente, la entrada Enable es la que desactiva o activa el scrambling para estos casos. La entrada Enable puede ser vista como un vector cuyos bits cambian cuando es necesario detener el scrambling.

### 10.16. FEC en DVB-T

# *Capítulo 11*

## *Variables comúnmente usadas en los fluajogramas que acompañan a este libro*

- **b:** bit.
  - **B:** Byte (cuando se refiere a unidad de medición).
  - **S:** se usa la S en mayúscula para indicar “Sample” o muestra. Se dejó así debido a que esa práctica está extendida en los manuales de los USRP y en los materiales que se encuentran en Internet que usualmente están inglés.
  - **s:** Se usa la s en minúscula para indicar “symbol”.
- Para señales en el canal o bloques de la capa de precanal:
- **B:** Es el ancho de banda pasobandas del canal inalámbrico. Se refiere a la banda útil del espectro radioeléctrico (ERE) que se puede usar para una comunicación dada.
  - **B\_eFM:** El ancho de banda de una emisora FM.
  - **BW:** Ancho de banda bandabase. A diferencia de B, este valor aplica para la envolvente compleja de la señal útil que viaja por el ERE.
  - **Ch\_Freq\_Offset:** Permite configurar canal para que produzca desfases de frecuencia.
  - **Ch\_Jitter:** Sirve para variar el valor Epsilon del bloque Channel Model de modo que el canal produzca una inestabilidad del reloj o Jitter variable.
  - **DelayAcc:** Retardo a introducir al Filtro de Acoplamiento con acumulador para lograr el mejor desempeño posible
  - $F_c$  : frecuencia de la portadora de radiofrecuencias, la frecuencia central de la banda de frecuencias asignada para la comunicación.
  - **Kd:** Coeficiente de decimación que aplica el bloque decimador que hay internamente, en el USRP.
  - **Kd\_d:** Es el coeficiente de decimación deseado, es decir, el que resultaría si el USRP pudiese manejar cualquier coeficiente de decimación. Sirve como insumo para obtener Kd.
  - **Kd\_rx o Kd\_tx:** Se usan cuando se requiere diferenciar entre el coeficiente de decimación usado en recepción y el usado en transmisión.
  - **Ki:** Coeficiente de interpolación que aplica el bloque interpolador que hay internamente en el USRP.
  - **N\_Lob\_p\_B:** Número de lóbulos del espectro de una señal digital, en el ancho de banda B. El lóbulo del medio, por tener doble ancho, se cuenta por dos. Usualmente es igual a Sps.
  - **NdB:** es la altura de la PSD del ruido blanco, conocido como No.
  - **ntaps:** Es el número de componentes que tiene la respuesta al impulso del filtro FIR usado.
  - **rcc\_taps:** Son los taps, es decir las componentes de la respuesta al impulso, para un filtro Root Raid Cosine.
  - **Rolloff:** Es el parámetro esencial de un Filtro Coseno Alzado. También le llaman alpha o Excess Bandwidth.
  - **Samp\_rate:** Es la frecuencia de muestreo que maneja el USRP, luego es la frecuencia de muestreo de la envolvente compleja que viaja por el cable Ethernet.
  - **Samp\_rate\_adc:** Es la frecuencia de muestreo que usa el DAC ubicado internamente en la tarjeta hija de transmisión del USRP. . Es igual a 400 MSs para el NI-USRP 2920.

- **Samp\_rate\_d:** Es la frecuencia de muestreo de la señal de la envolvente compleja que desearíamos entregar al USRP, pues es la que se obtiene teóricamente.
- **Samp\_rate\_d:** Es la frecuencia de muestreo de la señal de la envolvente compleja que desearíamos entregar al USRP, pues es la que se obtiene teóricamente.
- **Samp\_rate\_dac:** Es la frecuencia de muestreo que usa el ADC ubicado internamente en la tarjeta hija de recepción del USRP. Es igual a 100 MSs para el NI-USRP 2920
- **Samp\_rate\_rx :** Es igual que samp\_rate, solo que se usa cuando queremos diferenciar este parámetro de la frecuencia de muestreo usada en transmisión. De modo que es la frecuencia de muestreo de la señal de la envolvente compleja que entregamos al USRP Sink.
- **Samp\_rate\_tx:** De manera similar a samp\_rate\_rx, esta es la frecuencia de muestreo de la señal de la envolvente compleja que recibimos del USRP Source.
- **Sps:** *samples per symbol* es el número de muestras que lleva un símbolo de la envolvente compleja que pasa al canal.
- **SymTune:** Sintonización de los símbolos que entran al demodulador M-PAM al momento de armar los paquetes de bits que componen cada muestra del mensaje. Lo que se hace es retrasar en el valor SymTune los símbolos hasta que sean reconocidos correctamente por el demodulador.
- **TimingDelay:** Es el tiempo (en número de muestras) de retardo que debe ser introducido a la señal que llega al muestreador del receptor con el fin de seleccionar el mejor instante de muestreo, aquel en el que el diagrama de ojo está más abierto.
- **W:** Ancho de banda de Nyquist o del criterio de Nyquist para ISI.

Para las señales fuente:

- **NbpS:** *Number of bits per sample* Es el número de bits que representan a cada muestra del mensaje.
- **Niveles Q:** Número de niveles de cuantificación de la señal mensaje cuantificada.
- **Samp\_rate\_audio:** Es la frecuencia de muestreo de la señal mensaje de audio, que en este caso es de audio. Esto para diferenciarla de samp\_rate que la estaremos usando para el USRP.
- **Vp:** Amplitud máxima que puede alcanzar la señal mensaje antes de ser cuantificada.

Para señales moduladas:

- **bps:** *Bits per symbol* número de bits que representan cada símbolo de la señal modulada.
- **Constelacion:** Representa la tabla de verdad de la modulación usada.
- **M:** Orden de la modulación usada.
- **MiconstellationObject:** Es una constelación, pero hecha bajo unas normas comunes para una familia de bloques GNU Radio, como por ejemplo el bloque Constellation Decoder.
- **Rs:** Es la tasa de símbolos útil que se entrega al canal. No incluye símbolos usados en señalización, sincronismo o alineamiento de tramas.
- **SymDelay:** Es el retraso que habría que introducir a los bits transmitidos para que puedan ser comparados en una misma escala de tiempo con los recibidos.
- **Tsym:** Duración de un símbolo.

Para señales codificadas:

- **AccessCode:** *Código de acceso.* Es el vector que contiene el código de acceso
- **Preamble:** *Preámbulo.* Es el vector que contiene el preámbulo del código
- **Code:** *El código.* Es el vector que contiene el código. A su vez, el código es el AccessCode junto con el Preamble
- **Bpayload:** *bytes en la carga útil.* Es el tamaño en bytes de la carga útil en un paquete con codificación
- **BaCode:** *Bytes en el código de acceso.* Corresponde al número de bytes que contiene el AccessCode
- **Bpre** *Bytes en el preámbulo del código.* Corresponde al número de bytes que contiene el preámbulo del código
- **BpCode:** *Bytes por código.* Corresponde al número de bytes que ocupa el código junto con su preámbulo.

Capas de multiplexado:

- **C\_PN:** Almacena un vector de un código de pseudo ruido.

- **ChipSysDelay:** Es el equivalente a SymSysDelay pero dado en número de chips, cuando se usa DS-SS. Es demasiado útil en el bloque b\_de.ds\_spreadspect\_cc ya que el código que se va a aplicar debe ser previamente retrasado en el valor para que funcione correctamente el de-ensanchamiento.
- **Nspt\_s:** número de símbolos por time slot. Es el número de símbolos que un usuario pone en cada ventana de tiempo (time-slot) que ocupa en la trama multiplexada.
- **Nu:** Número de usuarios.
- **Rbu:** Rata de bits por usuario, cuando hay más de un usuario en el sistema de comunicación, como por ejemplo cuando se usa multiplexado.
- **Rch:** Rata de chips.
- **Rs\_subchannel:** se refiere a la rata símbolos que lleva cada subcanal para el caso de señales multiplexadas, donde cada canal aporta una rata de símbolos, pero la señal multiplexada lleva la suma de las ratas de símbolos de todos los subcanales
- **Rsu:** Es la rata de símbolos de cada usuario que corre por el canal, cuando el canal se usa para enviar información de varios usuarios, como cuando se usa el multiplexado.
- **SF:** El Spreading Factor es número de veces que el espectro se ensancha. Es la relación entre el ancho de banda que ocupa la señal ensanchada con respecto al que ocuparía sin la técnica DSSS.
- **Spch:** Es samples per chip. el número de muestras asignadas a cada chip.
- **spch:** Samples per chip. Es el número de muestras por chip que se usa cuando una señal ensanchada con DS-SS se entrega a un USRP. Es el equivalente a sps, solo que en DS-SS no se envían símbolos, sino chips.
- **SymSysDelay o MuxDelayComp:** Es retardo que sufre la señal de la información al viajar por todo el sistema hasta llegar al punto de destino. Sirve para introducir un retardo a la señal transmitida a la hora de compararla con la recibida.
- **t-s:** time slot es una celda de tiempo usada en TDM.
- **Us:** Usuario

Para la instrumentación:

- **Nscope\_span:** Es el número de símbolos que se desean mostrar en un osciloscopio.
- **Tmax\_scope:** Es la duración de la ventana que se grafica en el osciloscopio.

## ***Bibliografía***

- [1] H. Ortega, “Creación y edición de módulos oot con sus bloques índice 1. ,” p. 54, 2017.
- [2] P. D. Welch, “The use of fast Fourier Transform for the Estimation of power Spectra: A method based on time averaging over short, modified periodograms.” *IEEE Transactions on audio and electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.
- [3] S. Haykin, *Communication systems*, 2001.
- [4] A. Dandridge and A. B. Tveten, “Homodyne Demodulation Scheme for Fiber Optic Sensors Using Phase Generated Carrier,” *IEEE Journal of Quantum electronics*, vol. QE-18, no. 10, pp. 1647–1653, 1982.
- [5] J. Lorenzo, C. Hernández, and A. R. Antonio, “Diseño de un modulador de FM mediante PLL para la interrogación de sensores interferométricos de fibra óptica.” vol. 39, no. 128, pp. 335–351, 2018.
- [6] A. Marwanto, M. A. Sarijari, N. Fisal, S. Kamilah, S. Yusof, and R. A. Rashid, “Experimental Study of OFDM Implementation Utilizing GNU Radio and USRP - SDR,” no. December, pp. 132–135, 2009.
- [7] C. B. Thomas Rondeau, *Artificial Intelligence in Wireless Communications*, 2009.
- [8] D. Lorenzo, “Ecualización adaptativa de un canal digital,” pp. 1–25, 2004. [Online]. Available: <https://drive.google.com/file/d/1XoqU-Hh8m18M3muvptJiY-LHh5bgdcwV/view>

## *Abreviaciones y definiciones*

**BER** *width*Bit Error Rate.. 127

**GNU Radio** *GNU Radio Companion* Herramienta de desarrollo libre y abierta que provee bloques de procesamiento de señal para implementar sistemas de radio definida por software. Puede utilizarse con hardware de RF de bajo costo para crear radios definidas por software, o sin hardware en un ambiente de simulación. Es utilizada extensivamente por ambientes académicos, aficionados y comerciales para dar soporte a la investigación en comunicaciones inalámbricas y en sistemas de radio en el mundo real.. 16–18, 25, 34, 42

**PCM** *Pulse Code Modulation* Así se conoce al método de transmisión de la información, cuando esta es representada por unos y ceros en el canal.. 159, 160

**SER** *width*Symbol Error Rate.. 127

**Time Alignment** *Alineación de tiempo.* Es una técnica usada en las transmisiones binarias que permite a la parte receptora distinguir el inicio de cada byte en medio de un chorro de unos y ceros. Consiste en insertar periódicamente un código que es conocido tanto por el transmisor como por el receptor, de manera que, cuando en el receptor se identifica ese código, se sabe que lo que sigue es el inicio de la carga útil. 159