

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

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

**“DISEÑO Y DESARROLLO DE UN RADIO DEFINIDO POR
SOFTWARE, PARA EL EJÉRCITO ECUATORIANO, MEDIANTE LA
UTILIZACIÓN DE UNA TARJETA USRP Y LA HERRAMIENTA
SIMULINK® DE MATLAB®”**

CAPT. DARWIN MANOLO PAREDES CALDERÓN

HUGO ANDRÉS ANGULO ORQUERA

SANGOLQUÍ - ECUADOR

2011

CERTIFICACIÓN

Certificamos que el presente proyecto de grado titulado: “**Diseño y Desarrollo de un Radio Definido por *Software*, para el Ejército Ecuatoriano, mediante la utilización de una tarjeta *USRP* y la herramienta *Simulink*[®] de *Matlab*[®]” fue realizado en su totalidad por los **Sres. Capt. Darwin Manolo Paredes Calderón** con C.I.: 180313800-5 y el **Sr. Hugo Andrés Angulo Orquera** con C.I.: 040103710-6, bajo nuestra dirección.**

Ing. Paúl Bernal

DIRECTOR

Ing. Román Lara

CODIRECTOR

“La razón es el manantial que reverdece al árbol de la ciencia y la sabiduría, sin una pequeña dosis, no somos más que animales”

Hugo Andrés Angulo O.

“La abundancia de palabras y la ignorancia predominan en la mayor parte de los hombres; si quieres sobresalir de la mayoría inútil, cultiva tu conocimiento y envuélvelo en nubes de silencio”.

Anónimo

RESUMEN

En la actualidad existe una gran demanda en el ámbito de la comunicación y por ende, la inherente solución a esta necesidad la exponen las telecomunicaciones. Sin embargo, es importante que todo tienda a la flexibilidad de los dispositivos, y en esta esfera, un radio definido en *software* es un modelo totalmente flexible, dinámico, versátil, que permite ser usado en todo el espectro electromagnético, mediante la utilización de un sistema periférico de radio, se pueden propagar ondas de radio en todos los rangos de frecuencia posibles.

La presente investigación presenta el diseño de un radio definido por *software* mediante el uso de la *USRP* y el *software Simulink* de *Matlab*. A partir de la digitalización de la voz, fue considerada necesaria la implementación del códec *G.726* de la *ITU*, en la versión de 32Kbps, a fin de realizar la respectiva comparación con el códec *G.711* de 64Kbps, ya desarrollado como bloque dentro de *Simulink*. A fin de proveer de robustez al sistema, se aprovechó la codificación de canal adecuada. Esto se lo consigue mediante el uso de los códigos de bloque y entrelazados o *Trellis*, como es el código Convolutacional.

Finalmente se propone ambos sistemas en modulación *QPSK* y en modulación en frecuencia, con la aspiración de proveer un aporte significativo y comparativo en el desarrollo de los sistemas *SDR*, mediante la valoración por medio del *BER*.

DEDICATORIA

A Dios por haberme proporcionado la sabiduría para alcanzar un logro más en mi vida. A mis padres que siempre lucharon sin miramientos por la superación de sus hijos, a mi Madre por su constante preocupación, a mi Padre por la fuerza que inspira para luchar, a mis hermanos por su forma de ser humildes y comprometidos el uno con el otro. A familiares por creer en mí. En especial a mi esposa Yolanda, mis hijas Paulette y Antonella, que supieron comprender mi ausencia en los momentos de sacrificio y dedicación, por el eterno amor y comprensión que me brindaron les agradezco y les llevo en mi corazón, por eso son y serán una eterna fuente de energía y motivo de superación por lo que lucharé siempre por ser su orgullo.

Capt. de Com. Paredes Calderón Darwin Manolo

A mis padres, Ximena y Carlos, que me han enseñado, con el ejemplo, que el esfuerzo y la dedicación son indispensables para merecer los laureles de la victoria. A mi hermana. A mis tías y a mi Abuela maternas, que en momentos difíciles supieron tenderme un puente para subir al siguiente peldaño. A mis familiares, que con su apoyo y respeto, he aprendido a forjarme como un individuo útil para la sociedad. A los amigos y amigas, lejanos y cercanos, que con sus mensajes de respeto y cordialidad, fui capaz de encontrar en ellos mucho más de lo que la dura coraza presenta. A todos aquellos que creen en mí y en mi capacidad. A todos aquellos verdaderos maestros que me moldearon para no conformarme con pequeñeces y seguir en busca de más y más respuestas. A los libros y a las letras de todos aquellos autores que con su genio construyeron en mí un ser ávido de conocimiento, que no terminará de saciarse ni con la muerte.

Hugo Andrés Angulo O.

AGRADECIMIENTO

Agradecemos a nuestros familiares por el aporte y soporte que representaron dentro del desarrollo del presente proyecto de tesis, ya que sin su apeo, llegar a la meta propuesta hubiese sido una tarea más compleja. Infinitas gracias para ustedes, sin su comprensión y paciencia, esto no hubiese llegado a buen puerto.

Además quisiéramos agradecer, tanto al Director, M.Sc. Ing. Paúl Bernal, y al Codirector, M.Sc. Ing. Román Lara, ya que gracias a su guía y apoyo, en el ámbito académico, nos fueron develados varios conocimientos realmente relevantes y válidos, que llegaron en momentos críticos dentro del desarrollo del proyecto y fueron una base fundamental a la hora del diseño y posterior implementación.

Queremos hacer una mención especial a la distancia, y retribuir su aporte a quienes colaboraron con nosotros de manera activa y precisa. A los investigadores Dipl.-Ing. Michael Schwall y al Dipl.-Ing. Stefan Nagel, pertenecientes al CEL (*Communications Engineering Lab*) de la ciudad de *Karlsruhe*, Alemania. Gracias a ellos, fue posible para nosotros entender el funcionamiento del *software* que ellos desarrollaron, conjuntamente aportando ideas y conocimientos, que ponen de manifiesto la importancia del apoyo entre investigadores, a pesar de la distancia.

Nos complace hacer otra mención especial para Marcus D. Leech del Departamento de Información y Ventas de *Ettus*, empresa fabricante de las tarjetas *USRP*, quien con sus conocimientos técnicos, nos puso al tanto del funcionamiento de la tarjeta de desarrollo mencionada y aportó con varios criterios técnicos en momentos críticos.

Sobre todo, quisiéramos corresponder a la ayuda que nos prestaron varios de los docentes de la institución, que gracias a su cúmulo de conocimientos, nos guiaron y recomendaron varias soluciones a los problemas que se presentaron en el arduo camino. Por último agradecemos a varios compañeros que creyeron en nosotros y en nuestras capacidades.

PRÓLOGO

La presente investigación ha procurado detallar de forma ordenada y precisa el tratamiento que requiera la información generada por una fuente determinada, para ser codificada y propagada al espectro electromagnético, para finalmente alcanzar el objetivo de llegar al receptor y que éste la decodifique y la comprenda, de la forma más parecida a como fue emitida.

Es así como inicialmente se detalla las características y generalidades de un radio tradicional, para luego poder observar la evolución que genera un radio definido en *software* y así comprender las potencialidades de estos diseños de radio.

El diseño en sí se explica en los capítulos dos y tres donde se plantea el fundamento teórico y el diseño propiamente dicho respectivamente. Una señal de audio para optimizar el uso del ancho de banda debe ser comprimida, esto se consiguió con la utilización del códec *G.726*, recomendado por la *ITU*, luego esta información requiere ser codificada mediante un algoritmo determinado, a fin de disminuir los errores que genera el ruido introducido en el canal, así como también de corregir estos errores cuando así sucedan, esto se lo implementó mediante la codificación convolucional y su respectiva decodificación usando el algoritmo de *Viterbi*. A modo de comparación, también se dotó a los modelos del código de *Hamming*, a fin de confirmar la diferencia en el uso de un código que detecta y corrige errores y otro, como el reciente mencionado, que únicamente detecta los errores.

La información deberá ser modulada, para esto se utilizó la modulación de fase en cuadratura, QPSK, y modulación FM; esto se usó a fin de suministrar

seguridad a la señal emitida por el transmisor; a pesar de que su costo computacional es elevado, es fundamental su utilización.

Finalmente esta señal se ha propagado en las frecuencias de HF y VHF. Todo esto se lo implemento sobre un periférico universal de radio, el cual procesa la información a través de una matriz de compuertas programables, que adecúa la información para poder ser emitida.

Consecuentemente, el proceso de decodificación se lo realizó utilizando los mismos algoritmos en el orden adecuado para obtener la señal semejante a la emitida por la fuente y conseguir un adecuado funcionamiento y rendimiento del radio.

INDICE DE CONTENIDO

RESUMEN	IV
DEDICATORIA	V
AGRADECIMIENTO	VI
PRÓLOGO	VIII
GLOSARIO	XVII
CAPÍTULO I: INTRODUCCIÓN	1
1.1. ANTECEDENTES.....	1
1.2. JUSTIFICACIÓN E IMPORTANCIA.....	3
1.3. ALCANCE DEL PROYECTO	4
1.4. OBJETIVOS.....	4
1.4.1. <i>Objetivo General</i>	4
1.4.2. <i>Objetivos Específicos</i>	4
CAPÍTULO II: FUNDAMENTO TEÓRICO	6
2.1. EQUIPOS DE RADIO CONVENCIONALES	6
2.1.1. <i>Historia y Evolución de los Equipos de Radio en el Ejército Ecuatoriano</i>	6
2.1.2. <i>Descripción de las Características de los Equipos de radio existentes</i>	7
2.1.3. <i>Arquitecturas</i>	13
2.2. INTRODUCCIÓN A LOS <i>SDR</i>	16
2.2.1. <i>Definición</i>	16
2.2.2. <i>Características Técnicas</i>	19
2.2.3. <i>Servicios</i>	22
2.3. INTRODUCCIÓN A <i>USRP</i>	25
2.3.1. <i>Definición y Desarrollo</i>	25
2.3.2. <i>Características Técnicas</i>	26
2.3.3. <i>Interfaces</i>	31
2.3.4. <i>Conversión Análoga Digital ADC</i>	31
2.3.5. <i>Conversión Digital Analógica DAC</i>	32
2.3.6. <i>Entradas y Salidas Analógicas</i>	32
2.3.7. <i>Entradas y Salidas Digitales</i>	33
2.3.8. <i>Potencia</i>	34
2.4. INSTALACIÓN DE SOFTWARE Y HARDWARE REQUERIDO PARA MANEJO DE <i>USRP</i>	34
2.4.1. <i>Identificación de Interfaces</i>	34
2.4.2. <i>Instalación de software para manejo de USRP</i>	37
2.5. EL PROCESO DE LA COMUNICACIÓN	44

2.6.	CODIFICACIÓN Y DECODIFICACIÓN DE FUENTE.....	45
2.6.1.	<i>Definición</i>	45
2.6.2.	<i>Compresión de la voz</i>	46
2.7.	CODIFICACIÓN DE CANAL.....	51
2.7.1.	<i>Definición</i>	51
2.7.2.	<i>Generalidades</i>	51
2.7.3.	<i>Códigos de Codificación de Canal</i>	52
2.8.	TÉCNICAS DE MODULACIÓN Y DEMODULACIÓN.....	65
2.8.1.	<i>Definición</i>	65
2.8.2.	<i>Importancia de la Modulación y Demodulación</i>	68
2.8.3.	<i>Tipos de Modulaciones</i>	69
CAPÍTULO III: DISEÑO Y DESARROLLO DEL SDR.....		80
3.1.	DISEÑO DEL SISTEMA DE COMUNICACIONES.....	80
3.1.1.	<i>Generalidades</i>	80
3.2.	CODIFICACIÓN DE FUENTE.....	82
3.2.1.	<i>Códec G.711</i>	82
3.2.2.	<i>Códec G.726</i>	84
3.3.	CODIFICACIÓN DE CANAL.....	96
3.3.1.	<i>Código Convolutacional</i>	96
3.3.2.	<i>Código de Hamming</i>	97
3.3.3.	<i>Decodificador Viterbi</i>	99
3.4.	MODULACIÓN.....	100
3.4.1.	<i>En Banda Base: QPSK</i>	100
3.4.2.	<i>En Frecuencia: FM</i>	101
3.5.	MANEJO Y CONFIGURACIÓN DE LOS BLOQUES DE <i>SIMULINK USRP</i>	103
3.5.1.	<i>Configuración módulo sink</i>	103
3.5.2.	<i>Configuración módulo source</i>	105
CAPÍTULO IV: SIMULACIONES, PRUEBAS Y RESULTADOS EXPERIMENTALES.....		109
4.1.	INTRODUCCIÓN.....	109
4.2.	SIMULACIONES.....	109
4.2.1.	<i>Modelo I</i>	109
4.2.2.	<i>Modelo II</i>	110
4.2.3.	<i>Modelo III</i>	111
4.2.4.	<i>Modelo IV</i>	112
4.2.5.	<i>Modelo V</i>	113
4.2.6.	<i>Modelo VI</i>	114
4.2.7.	<i>Modelo VII</i>	115

4.2.8.	<i>Modelo VIII</i>	116
4.3.	CÁLCULOS TEÓRICOS	117
4.3.1.	<i>Calculo de la Tasa de Error</i>	117
4.3.2.	<i>Cálculo del Ruido de Cuantificación</i>	119
4.3.3.	<i>Rendimiento del Error PSK</i>	119
4.3.4.	<i>Calculo de la relación señal Ruido (SNR)</i>	120
4.4.	COMPARACIÓN DEL BER DE LOS MODELOS CON MODELACIÓN DIGITAL.....	122
4.4.1.	<i>Sistemas con Códigos Convolucionales y modulación QPSK</i>	122
4.4.2.	<i>Sistemas con Códigos de Hamming y modulación QPSK</i>	124
4.4.3.	<i>Comparación entre los cuatro modelos que usan modulación QPSK</i>	125
4.5.	IMPLEMENTACIÓN DE LOS MODELOS DISEÑADOS	126
4.5.1.	<i>Materiales</i>	126
4.5.2.	<i>Diseños Implementados con el uso de los módulos USRP de Simulink</i>	127
4.6.	PRUEBAS EXPERIMENTALES.....	129
4.6.1.	<i>Resultados de las pruebas Experimentales en el Analizador de Espectros Agilent N1996A-506</i>	129
4.7.	COMPARACIÓN DE RESULTADOS.....	131
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES.....		133
5.1.	CONCLUSIONES	133
5.2.	RECOMENDACIONES.....	137
BIBLIOGRAFÍA.....		138

INDICE DE FIGURAS

FIGURA 2. 1. EQUIPO DE RADIO TRA 931.....	8
FIGURA 2. 2. EQUIPO DE RADIO PRM 4031 HF SSB.....	9
FIGURA 2. 3. EQUIPO DE RADIO DATRON PRC2100V.....	10
FIGURA 2.4. RADIO HARRIS RF-5800-V FALCON II.....	12
FIGURA 2. 5. ARQUITECTURA DE RADIO RACAL.....	14
FIGURA 2. 6. EQUIPOS DE RADIO TRADICIONALES.....	15
FIGURA 2. 7. RADIOS DEFINIDO POR SOFTWARE.....	15
FIGURA 2. 8. PRIMER SDR PRESENTADA AL MERCADO.....	15
FIGURA 2. 9. UNIVERSAL SOFTWARE RADIO PERIPHERAL.....	17
FIGURA 2. 10. ARQUITECTURA DE UN SDR.....	20
FIGURA 2. 11. ARQUITECTURA IDEAL DE UN SDR.....	22
FIGURA 2. 12. EVOLUCIÓN DEL SDR.....	24
FIGURA 2. 13. TARJETA MAIN BOARD DEL USRP.....	26
FIGURA 2. 14. DIAGRAMA DE BLOQUES DEL FPGA DEL USRP.....	29
FIGURA 2. 15. CIRCUITO E INTERFAZ USB.....	35
FIGURA 2.16. CONFIGURACIÓN DEL DRIVER USB.....	38
FIGURA 2. 17. ADMINISTRADOR DE DISPOSITIVOS DE WINDOWS.....	39
FIGURA 2. 18. BÚSQUEDA DEL SDK.....	41
FIGURA 2. 19. ACTIVACIÓN DEL SDK.....	41
FIGURA 2. 20. SELECCIÓN DE COMPILADOR EN MATLAB.....	42
FIGURA 2. 21. AJUSTE DE DIRECTORIO DEL MÓDULO SIMULINK-USRP.....	42
FIGURA 2. 22. EJECUCIÓN DEL COMANDO USRPBUILDBINARIES.....	43
FIGURA 2. 23. MÓDULO SIMULINK-USRP.....	43
FIGURA 2.24. ELEMENTOS DE UNA COMUNICACIÓN.....	45
FIGURA 2. 25. ALFABETO FUENTE Y ALFABETO CÓDIGO.....	46
FIGURA 2. 26. ESQUEMA DEL CODIFICADOR - DECODIFICADOR ADPCM.....	48
FIGURA 2. 27. MUESTRO DEL CODIFICADOR ADPCM.....	49
FIGURA 2. 28. CODIFICADOR ADPCM.....	49
FIGURA 2. 29. PROCESO DE CODIFICACIÓN CONVOLUCIONAL.....	57
FIGURA 2. 30. CODIFICACIÓN CONVOLUCIONAL CON 2 BITS DE ENTRADA A LA VEZ.....	57
FIGURA 2. 31. DIAGRAMA DE BLOQUES DE CÓDIGO CONVOLUCIONAL 4, 3, 5.....	59
FIGURA 2. 32. EJEMPLO DE UN CODIFICADOR CONVOLUCIONAL (2, 1, 3).....	59
FIGURA 2. 33. DIAGRAMA CODIFICACIÓN CONVOLUCIONAL.....	61
FIGURA 2. 34. MÁQUINA SECUENCIAL DEL CÓDIGO CONVOLUCIONAL.....	62
FIGURA 2. 35. TRELISA.....	63
FIGURA 2. 36. ESPECTRO ELECTROMAGNÉTICO.....	67

FIGURA 2. 37. CLASIFICACIÓN DE MODULACIÓN CON PORTADORA ANALÓGICA.	70
FIGURA 2. 38. CLASIFICACIÓN DE MODULACIÓN CON PORTADORA DIGITAL.	70
FIGURA 2. 39. MÉTODO DIRECTO DE MODULACIÓN FM.	73
FIGURA 2. 40. MÉTODO INDIRECTO DE DEMODULACIÓN FM.	74
FIGURA 2. 41. MODULACIÓN PSK.	75
FIGURA 2. 42. DIAGRAMA FASORIAL Y ONDA SENOIDAL.	77
FIGURA 2. 43. DIAGRAMA DE CONSTELACIÓN BPSK.	78
FIGURA 2. 44. MODULACIÓN QPSK.	79
FIGURA 3. 1. SISTEMA DIGITAL DE COMUNICACIONES.	81
FIGURA 3. 2. DIAGRAMA GENERAL DE UN CODIFICADOR ADPCM.	84
FIGURA 3. 3. ESQUEMA DE BLOQUES DEL CODIFICADOR G.726.	86
FIGURA 3. 4. ESQUEMA DE BLOQUES DEL DECODIFICADOR G.726.	87
FIGURA 3. 5. BLOQUE DE CONVERSIÓN A PCM.	88
FIGURA 3. 6. BLOQUE DEL CÁLCULO DE LA DIFERENCIA DE LAS SEÑALES DE ENTRADA Y ESTIMADA.	88
FIGURA 3. 7. BLOQUE CUANTIZADOR ADAPTATIVO.	89
FIGURA 3. 8. BLOQUE CUANTIZADOR ADAPTATIVO INVERSO.	90
FIGURA 3. 9. BLOQUE DEL FACTOR DE ADAPTACIÓN DE CUANTIZACIÓN DE ESCALA.	90
FIGURA 3. 10. CONTROL DE VELOCIDAD DE ADAPTACIÓN.	91
FIGURA 3. 11. BLOQUE DE PREDICTOR ADAPTATIVO Y CALCULADOR DE LA SEÑAL RECONSTRUIDA.	92
FIGURA 3. 12. BLOQUE DE DETECCIÓN DE TONO Y TRANSICIÓN.	92
FIGURA 3. 13. ESQUEMA DEL CODIFICADOR G.726 IMPLEMENTADO.	93
FIGURA 3. 14. BLOQUE DE AJUSTE DE CODIFICACIÓN SÍNCRONA.	94
FIGURA 3. 15. ESQUEMA DEL DECODIFICADOR G.726 IMPLEMENTADO.	95
FIGURA 3. 16. DIAGRAMA DE REGISTROS DEL CÓDIGO CONVOLUCIONAL.	96
FIGURA 3. 17. CONFIGURACIÓN DE PARÁMETROS DEL CODIFICADOR CONVOLUCIONAL.	97
FIGURA 3. 18. MENÚ DE SIMULINK.	98
FIGURA 3. 19. CONFIGURACIÓN DEL BLOQUE CODIFICADOR Y DECODIFICADOR DE HAMMING.	99
FIGURA 3. 20. CONFIGURACIÓN DEL DECODIFICADOR DE VITERBI.	100
FIGURA 3. 21. CONSTELACIÓN DEL BLOQUE DE MODULACIÓN QPSK DE SIMULINK.	101
FIGURA 3. 22. MÉTODO DE DEMODULACIÓN FM DE RICHARD G. LYON.	102
FIGURA 3. 23. IMPLEMENTACIÓN MODULADOR FM.	102
FIGURA 3. 24. IMPLEMENTACIÓN DEMODULADOR FM INDIRECTO.	103
FIGURA 3. 25. INTERFAZ DE CONFIGURACIÓN DEL USRP SINK EN SIMULINK.	104
FIGURA 3. 26. CONFIGURACIÓN DE FRECUENCIA DE TRABAJO DEL MÓDULO USRP SINK DE SIMULINK.	105
FIGURA 3. 27. INTERFAZ DE CONFIGURACIÓN DEL USRP SOURCE EN SIMULINK.	106
FIGURA 3. 28. CONFIGURACIÓN DE FRECUENCIA DE TRABAJO DEL MÓDULO USRP SOURCE DE SIMULINK.	107
FIGURA 4. 1. SISTEMA DE COMUNICACIONES PROPUESTO MODELO I.	110
FIGURA 4. 2. SISTEMA DE COMUNICACIONES PROPUESTO MODELO II.	111

FIGURA 4. 3. SISTEMA DE COMUNICACIONES PROPUESTO MODELO III.	112
FIGURA 4. 4. SISTEMA DE COMUNICACIONES PROPUESTO MODELO IV.	113
FIGURA 4. 5. SISTEMA DE COMUNICACIONES PROPUESTO MODELO V.	114
FIGURA 4. 6. SISTEMA DE COMUNICACIONES PROPUESTO MODELO VI.	115
FIGURA 4. 7. SISTEMA DE COMUNICACIONES PROPUESTO MODELO VII.	116
FIGURA 4. 8. SISTEMA DE COMUNICACIONES PROPUESTO MODELO VIII.	117
FIGURA 4. 9. CÁLCULO DE LA TASA DE BITS ERRÓNEOS.	118
FIGURA 4. 10. GRÁFICA COMPARATIVA DEL BER DE LOS MODELOS I Y III, CON CÓDIGO CONVOLUCIONAL.	123
FIGURA 4. 11. GRÁFICA COMPARATIVA DEL BER DE LOS MODELOS II Y IV, CON CÓDIGO DE HAMMING.	125
FIGURA 4. 12. GRÁFICA COMPARATIVA DEL BER DE LOS SISTEMAS DISEÑADOS.	126
FIGURA 4. 13. MONTAJE DEL SISTEMA DE COMUNICACIONES.	128
FIGURA 4. 14. IMPLEMENTACIÓN DEL TRANSMISOR DEL MODELO I.	128
FIGURA 4. 15. IMPLEMENTACIÓN DEL RECEPTOR DEL MODELO I.	129
FIGURA 4. 16. POTENCIA DE TRANSMISIÓN CON MODELO I.	129
FIGURA 4. 17. POTENCIA DE TRANSMISIÓN CON MODELO III.	130
FIGURA 4. 18. POTENCIA DE TRANSMISIÓN CON MODELO V.	130
FIGURA 4. 19. POTENCIA DE TRANSMISIÓN CON MODELO VII.	131

INDICE DE TABLAS

TABLA 2. 1. DATOS TÉCNICOS DE LA RADIO RACAL TRA 931.	8
TABLA 2. 2. DATOS TÉCNICOS DE LA RADIO RACAL PRM 4031.....	9
TABLA 2. 3. DATOS TÉCNICOS DE LA RADIO DATRON.....	10
TABLA 2. 4. DATOS TÉCNICOS DE LA RADIO HARRIS.	12
TABLA 2. 5. PINES DE FUNCIONAMIENTO DEL PUERTO USB.	31
TABLA 2. 6. CUADRO COMPARATIVO DE LOS COMPRESORES DE AUDIO.	50
TABLA 2. 7. TABLA DE POLINOMIOS GENERADORES.....	54
TABLA 2. 8. TABLA DE ESTADOS DE LA CODIFICACIÓN CONVOLUCIONAL.....	60
TABLA 2. 9. TABLA DE ENTRADAS - SALIDAS DEL CÓDIGO CONVOLUCIONAL.	62
TABLA 4. 1. TABLA COMPARATIVA DE LOS MODELOS PROPUESTOS.....	132

GLOSARIO

ADC	Conversión analógica-Digital
ADPCM	Codificación Diferencial Adaptativa por Modulación de Pulso.
ALIASING	Es la pérdida de información debido al traslape de la señal, generalmente ocurre en la recepción, debido a la presencia de copias periódicas de la señal en el espectro.
AM	Amplitud Modulada
ASK	Modulación por desplazamiento de Amplitud.
ASIC	Circuito Integrado de Aplicación Específica.
ATC	Código Programable de transformación.
BER	Tasa de error de bit.
BPSK	Modulación bipolar por desplazamiento en fase.
CELP	Código de Predicción Lineal por estructuras Computacionales.
CDMA	Acceso múltiple por división de código.
CÓDEC	Codificador-Decodificador.
DAC	Conversión Digital-Analógica.
DDC	Convertidor Digital de bajada.
DUC	Convertidor Digital de subida.
DSP	Procesamiento Digital de Señales
Eb/No	Relación Energía del bit versus ruido.
Firmware	Conjunto de instrucciones de propósito específico.

Flip-Flop	Es un circuito oscilador de onda cuadrada, capaz de permanecer en un estado determinado o en el contrario durante un tiempo indefinido.
FDD	Dúplex de la división de tiempo.
FDMA	Acceso múltiple por división de frecuencia.
FPGA	Compuerta de Arreglos Programados por Campo.
FM	Frecuencia Modulada.
FSK	Modulación por desplazamiento de Fase.
GPIF	Interface de Propósito General.
KSps	Miles de muestras por Segundo.
LPC	Código Predictivo Lineal.
LPAS	Código de Predicción lineal por síntesis y análisis.
LNA	Amplificador de Bajo Ruido.
PN	Ruido Pseudoaleatorio
PSK	Modulación por desplazamiento en Frecuencia.
PGA	Amplificador de ganancias programables.
QPSK	Modulación en cuadratura por desplazamiento en fase.
QAM	Modulación de Amplitud en cuadratura.
RF	Radio Frecuencia.
SDR	Radio Definida por <i>Software</i> .
SMA	(<i>Sub Miniature versión A</i>) a un tipo de conector roscado para cable coaxial utilizado en microondas, útil hasta una frecuencia de 33 GHz,

si bien suele dejar de utilizarse a partir de los 18 GHz, existen tipos diseñados para 26,5 GHz.

SNR	Relación Señal Ruido
SDK	Kit para desarrollo de <i>software</i>
TDD	Dúplex de la división de la frecuencia.
TDMA	Acceso Múltiple por división de tiempo.
USRP	Sistema Periférico Universal de Radio.
VCELP	Expresión Vectorial de los Códigos CELP.

CAPÍTULO I

INTRODUCCIÓN

1.1. Antecedentes

En la actualidad vivimos una era tecnológica, en la que la información es bombardeada hacia quienes la requieren por cualquier método o canal de comunicación, ya sea *internet*, la televisión, la radio, las redes sociales, entre otros.

Sin embargo para Instituciones importantes, Organizaciones Internacionales, para los Gobiernos seccionales y locales, es fundamental contar con un sistema de comunicaciones veraz y efectivo que les permita contar con la información precisa y en el momento oportuno, a fin de poder considerar las acciones a tomar, para enfrentar futuras condiciones en cualquier ámbito en las que éstas se presenten.

Es así como también el Comando y Control de una Fuerza Militar, se fundamenta en gran parte, en la capacidad operativa de sus medios de comunicación, los mismos que permiten un flujo de información en todos los niveles jerárquicos, que forman parte de una operación militar, es ahí donde resalta la importancia que tienen las comunicaciones en la ejecución de las operaciones militares.

La forma oportuna como se recibe la información, así como la veracidad de la misma, son herramientas que tiene un comandante para la adecuada toma de decisiones y sobre todo son los medios de comunicación los que permiten

informar estas decisiones a sus mandos superiores y subordinados para alcanzar el cabal cumplimiento de las misiones asignadas.

En los anales de la historia datan que la mayoría de inventos de telecomunicaciones, han nacido primero en una imperante necesidad de comunicación y control de los ejércitos en sus operaciones militares, es así como se inventó la radio, el teléfono, la comunicación satelital, etc.

La vigencia de los equipos robustos que ocupen grandes espacios físicos y que requieran de la composición y configuración de una gran cantidad de equipos, así sabemos que las radios son equipos receptores y transceptores de radiocomunicaciones y están constituidos por multitud de componentes electrónicos, los cuales forman circuitos sintonizadores, etapas de frecuencia intermedia, detectores, amplificadores de baja frecuencia, es decir, están constituidos por "*hardware*", esta constitución de los equipos de radio empezó una era decadente, ya que la tendencia mundial busca el confort, la portabilidad, la movilidad y sobre todo la seguridad en las comunicaciones; son éstas condiciones que en la actualidad se las encuentra definidas de forma más sencilla mediante un software o un FPGA (*Field Programmable Gate Array*), en los denominados SDR (*Radio Defined Software*).

La evolución inició con la inclusión de los microprocesadores en las radios, esto sucedió en los años ochenta, en tanto, en la década de los 1990's comenzó la introducción en los modernos equipos de radio de los chips DSP (*Digital Signal Processor*), los mismos que permiten mediante técnicas digitales realizar filtros de paso de banda y de supresión de ruidos, entre otras posibilidades, muy eficaces, mejor que los realizados tradicionalmente con circuitos analógicos.

Desde principios de la década del 2000 radioaficionados como *Gerald Youngblood*¹, *AC5OG*, que es gerente de marketing del canal *ADM* en todo el mundo, inició la investigación y desarrollo un nuevo concepto de equipos de radiocomunicaciones, los equipos de radio desarrollados por programa o "*radio*

¹ Disponible en: www.flex-radio.com/about.aspx?topic=team

software", en los que la parte hardware o circuitería es mínima, y la mayor parte de las funciones que definen un equipo de radio se definen por *software* en un ordenador, dotado indispensablemente de tarjeta de sonido, es por eso que la definición más precisa de un *SDR*, la da el *Forum* de Investigación de Comunicaciones Inalámbricas², que define así: Un *SDR* es una Radio en la que una o todas las funciones de la capa física están definidas por *software*.

Los llamados *SDR*, son fruto de la evolución de las tecnologías, que buscan confinar los antiguos equipos de radio con las nuevas necesidades de la humanidad. Esto es posible alcanzar mediante la utilización de tarjetas o módulos integrales de comunicación, como son los *USRP (Universal Software Radio Peripheral)*, el mismo que para su configuración utiliza el *software* GNU radio, Matlab, Simulink, éstos permiten implementar una plataforma flexible, capaz de cambiar de acuerdo a nuestras necesidades, el mismo equipo puede funcionar como un radio AM (Amplitud Modulada) , FM (Frecuencia Modulada), GPS (*Global Position System*), GSM (*Global System for Mobile Communications*), etc., todo en uno, evitando comprar un equipo propietario para cada tecnología.

1.2. Justificación e Importancia

El propósito de éste proyecto, es brindar una solución óptima y de bajo costo para cubrir las necesidades de radio que en la actualidad tiene el Ejército ecuatoriano, para lo cual la presente investigación pretende presentar una solución inicial y robusta, que constituya una base sólida para la implementación y fabricación de radios en grandes cantidades, así como también disponer de equipos terminales de radio flexibles que permitan satisfacer las necesidades existentes de equipos de comunicaciones en las Fuerzas Armadas y sobre todo de equipos de radio en los que la seguridad y codificación de la información sea manipulada por personas propias de la Institución y no por personas de empresas externas.

²Disponible en: <http://www.wirelessinnovation.org/mc/page.do?sitePageId=100379&orgId=sdf>

1.3. Alcance del Proyecto

El presente proyecto pretende configurar dos Radios Definidas por software, en la que la codificación y decodificación de fuente y de canal, así como la modulación, serán implementadas mediante *software (Matlab)*, y la parte de radio frecuencia estará implementada en un *USRP*, es decir la transmisión y recepción de la información utilizarán un Sistema Periférico Universal de Radio (*USRP*).

El *SDR* diseñado tendrá capacidad inicialmente de audio, la voz transmitida será digitalizada mediante un códec y la codificación y decodificación de canal se la realizará mediante un código de bloque, en tanto que la modulación se la implementará a través de los métodos existentes (QPSK, BPSK, FM, etc.), para lo cual se deberá realizar el análisis pertinente del rendimiento del *SDR* con cada una de las diferentes opciones de codificación existente, a fin de disponer un análisis comparativo del rendimiento, hasta finalmente presentar una solución óptima de *SDR*.

1.4. Objetivos

1.4.1. Objetivo General

Diseñar un Radio Definido por *Software*, mediante la utilización de una Tarjeta *USRP* y la herramienta *Simulink* de *Matlab*.

1.4.2. Objetivos Específicos

- Determinar las características y capacidades actuales de los equipos de radio existentes en el Ejército Ecuatoriano.
- Elaborar un tutorial de instalación y configuración tanto de software como de equipos utilizados.
- Implementar la codificación y decodificación de señales de audio, mediante los códecs adecuados.

-
- Determinar la modulación más eficiente y apropiada, que se adapte a las condiciones requeridas del SDR.
 - Determinar la codificación y decodificación de canal requerida, así como las técnicas de múltiple acceso para un rendimiento óptimo del SDR.
 - Realizar las pruebas de funcionamiento del diseño.

CAPÍTULO II

FUNDAMENTO TEÓRICO

2.1. Equipos de Radio Convencionales

2.1.1. Historia y Evolución de los Equipos de Radio en el Ejército Ecuatoriano

Desde sus inicios en la última década del siglo XVII, con los descubrimientos de Marconi, los equipos de radios se caracterizaban por ser dispositivos de gran tamaño, que estaban conformados por una gran cantidad de elementos electrónicos y que consecuentemente eran grandes y pesados, lo que complicaba su portabilidad, movilidad y confort para los usuarios.

Todo esto ha ido evolucionando a través del tiempo consiguiendo, cada vez, obtener equipos con capacidades ideales para las telecomunicaciones, es decir, gran alcance, bajo consumo de potencia, portabilidad, alta movilidad y sobre todo robustez, tanto en la operatividad como en durabilidad.

Los descubrimientos en la teoría de las telecomunicaciones han proporcionado los conocimientos adecuados y oportunos para cronológicamente ir creando equipos que utilicen diferentes tipos de codificación, modulación y principalmente el tratamiento mismo de las señales lo que ha generado un importante cambio de época en la historia de los equipos de radios, que es la era digital y la analógica.

Hoy en día la versatilidad que permite el tratamiento digital de los datos, ya sean estos audio, voz e imágenes, así como la optimización del uso del espectro y sobre todo la capacidad de corregir errores y librar eficientemente las interferencias, permite que la tendencia de los equipos de radio y sus fabricantes sea potencializar el crecimiento de los equipos de radio digitales.

El Ejército Ecuatoriano también forma parte de esta evolución, ya que desde sus inicios ha contado con equipos de radio que en su momento estén a la par de la tecnología, pasando desde los tradicionales equipos que operaban con claves morse, luego equipos analógicos modulados en AM o FM, hasta contar con equipos digitales que permiten hoy en día contar con radios digitales, que son parte fundamental del funcionamiento y operatividad de sus miembros e instituciones.

A continuación se procede a describir ciertos datos históricos y técnicos de los equipos de radio, más destacados y más utilizados, con los que cuenta el Ejército ecuatoriano.

2.1.2.Descripción de las Características de los Equipos de radio existentes

RADIOS RACAL

A continuación se presenta una breve descripción de los equipos de radio, que se usan aún en el Ejército Ecuatoriano.

- **Radio TRA 931**

Creada en 1971; además de las características indicadas, en la Tabla 2.1, el equipo cuenta con amplificador de potencia independiente, sistema de acoplamiento de antena y caja iniciadora de sintonía.

Este tipo de radio, son las más robustas en el sistema de comunicaciones HF del Ejército, debido a su capacidad de trabajar con grandes potencias y largas distancias de alcances. La Fig. 2.1 muestra

una imagen de una radio Racal, implementada como una estación de Radio Base, constituida por un amplificador de potencia y un acoplador de antena.



Figura 2. 1. Equipo de Radio TRA 931.

Tabla 2. 1. Datos Técnicos de la Radio Racal TRA 931.

Banda de Trabajo	1.6 a 29.9999 MHz
Modos de Operación	AM SSB: Banda Lateral única LSB: Banda lateral Inferior USB: Banda Lateral Superior CW: Onda Continua (Código MORSE)
Alcance	Hasta 100 Km dependiendo de la antena y potencia de salida.
Canales	284.000 canales a 100 KHz
Antena	<ul style="list-style-type: none"> ▪ Látigo ▪ Dipolo ▪ Helicoidal
Potencia de Salida	Alta Potencia: 10 W Baja Potencia: 2,5 W

- **Radio PRM 4031 HF SSB.**

Creada en los años ochenta, proviene de la misma familia Racal, pero diseñada para operar en el rango de frecuencias VHF, la Fig. 2.2, corresponde a un transceptor con capacidad de implementación como radio base o portátil, sus datos técnicos son mostrados en la Tabla 2.2.



Figura 2. 2. Equipo de Radio PRM 4031 HF SSB.

Tabla 2. 2. Datos Técnicos de la Radio Racal PRM 4031.

Banda de Trabajo	1.6 a 29.9999 MHz
Modos de Operación	USB : Banda lateral Superior LSB: Banda lateral Inferior CW: Onda Continua AM: Amplitud Modulada
Alcance	Hasta 100 km dependiendo de la antena y potencia de salida.
Canales	284.000 canales a 100 kHz
Antena	<ul style="list-style-type: none"> ▪ Látigo ▪ Dipolo ▪ Helicoidal

Potencia de Salida	Alta Potencia: 10 W Baja Potencia: 2,5 W
--------------------	---

RADIO DATRON

▪ Radio *Datron PRC2100V*

Es una radio digital, innovadora, creada en el año 2002, que incorpora aspectos importantes de seguridad, como salto de frecuencia, encriptación de datos y receptor GPS, como se muestra en la Fig. 2.3, sus datos técnicos se detallan en la Tabla 2.3:



Figura 2. 3. Equipo de Radio *Datron PRC2100V*.

Tabla 2. 3. Datos Técnicos de la Radio DATRON.

Banda de Trabajo	30 a 87.975 MHz
Modos de Operación	FF FM, Simplex o <i>Half Duplex</i> (FF Banda Completa de Frecuencia) FF FM COMSEC (Encriptado, con seguridad en las Comunicaciones) FH FM ECCM (FH Salto de Frecuencia) (Salto de frecuencia; resistencia a Contra Medidas Electrónicas) FH FM ECCM y COMSEC Incluye las dos anteriores.
Alcance	Hasta 30 Km dependiendo de la orografía del terreno y la potencia de salida.

Canales	4.640 con separación de canal de 12,5 kHz 2.320 con separación de canal de 25 kHz Capacidad de asociación de canales de hasta 10 grupos
Antena	<ul style="list-style-type: none"> ▪ Flexible de 1,2 m ▪ Plegable de 2 m
Potencia de Salida	10 W Portátil 75 W Vehicular

Radios HARRIS

Es importante señalar que las radios marca *Harris*, son las más robustas dentro del mercado, razón por la cual su costo mínimo se encuentra alrededor de los 20000 dólares americanos. A continuación se presenta una descripción de aquellas radios que están actualmente siendo usadas en el Ejército Ecuatoriano.

- **Radio Harris RF-5800-V Falcon II**

En la actualidad la marca Harris, se ha convertido en una empresa pionera en la fabricación y venta de equipos de radios tácticas, por lo que se hará una descripción más detallada de las características de éstos equipos, la Fig. 2.4 corresponde a una radio modelo *FALCON II*, producida a partir del año 2008. En la Tabla 2.4, además, se detallan las características técnicas del dispositivo en cuestión.



Figura 2.4. Radio Harris RF-5800-V Falcon II.

Tabla 2. 4. Datos Técnicos de la Radio HARRIS.

Banda de Trabajo	Multibanda 30 - 512 MHz <ul style="list-style-type: none"> • VHF /AM/FM • USF/AM
Modos de Operación	<ul style="list-style-type: none"> • FM <i>Analog Voice</i>, • AM <i>Voice</i> • ASK/FSK 16 Kbps CVSD <i>Voice</i> • ASK/FSK 16 Kbps <i>Data</i> • TCM 48/64 Kbps <i>Data</i>
Alcance	Varía según frecuencia y potencia de transmisión.
Canales	Espaciamiento de canal: 5, 6.25, 8.33, 12.5, 25 (KHz)
Antena	<ul style="list-style-type: none"> • Látigo • Dipolo • Helicoidal
Potencia de	<ul style="list-style-type: none"> • 50 W en transmisión FM

Salida	<ul style="list-style-type: none"> • 20 W en transmisión AM
Encriptación	CITADEL
Tasa de transmisión	64Kbps, ayuda a reducir el tiempo de transmisión
Contra Medidas Electrónicas	<ul style="list-style-type: none"> • Salto de Frecuencia • QUICKLOOH 1A • HAVEQUICK I/II • Rockwell-Collins TALON™.

2.1.3.Arquitecturas

Al diseñar un transceptor de radio tradicional, se debe pensar en una gran cantidad de tarjetas electrónicas, que reúnan las capacidades necesarias que requiera el equipo, es decir, se debe considerar una tarjeta para el transmisor, otra para la encriptación de los datos, otra para la codificación y decodificación, de tal forma que conforme se desearía incrementar las capacidades de la radio se debía pensar en el incremento del número de componentes de la misma, algo que sin lugar a dudas limitaba el incremento de ciertas capacidades de diseño, a más del costo de la fabricación de todo el conjunto.

Es así como por ejemplo en una radio Racal se encuentran varias tarjetas electrónicas, de gran tamaño como se aprecia en la Fig. 2.5, además de los elementos electrónicos, que obligaban a que el tamaño físico de la radio sea considerable y pesado.



Figura 2. 5. Arquitectura de radio Racal.

La arquitectura de radio Racal considera un módulo de antena, el cual se compone de la antena y una Unidad de Sintonización, un módulo de amplificación de potencia, módulo de transceiver, un sintetizador, módulos adicionales para filtrar la señal, además de todos los periféricos requeridos como parlantes, micro teléfono, etc.

En la actualidad debido a los avances tecnológicos, pasando de una simple memoria a los microprocesadores, luego los micro controladores y ahora los FPGA's, las arquitecturas de radio también han evolucionado, logrando reducir al mínimo el tamaño físico de la radio, sin necesidad de disminuir sus capacidades, por el contrario incrementándolas significativamente.

Es así como mediante el uso de un USRP y de sus tarjetas componentes la arquitectura de la radio se reduce a dos o tres tarjetas físicas y todo el tratamiento, codificación, corrección de errores y los filtros de señal se encuentran embebidos en la programación del FPGA, a todo esto se debe sumar la versatilidad del equipo ya que puede variar de forma rápida su utilidad, pasando de ser una radio HF a una VHF, a un celular, un receptor de televisión, etc., según sea la necesidad.

Con estos avances los equipos de radio pasaron de una arquitectura cerrada, robusta mostrada en la Fig. 2.6, con gran peso a equipos versátiles, livianos y eficientes como indican las Fig. 2.7 y 2.8.



Figura 2. 6. Equipos de radio Tradicionales.



Figura 2. 7. Radios definido por *Software*.

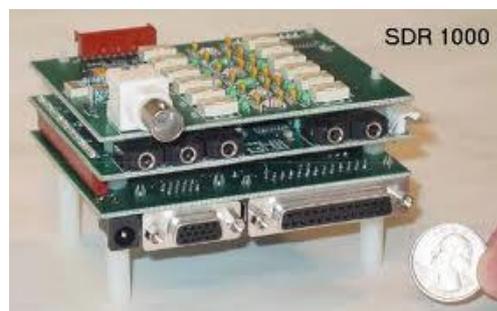


Figura 2. 8. Primer *SDR* presentada al mercado.

2.2.Introducción a los SDR

2.2.1.Definición

Históricamente los sistemas de radio han sido desarrollados para ajustarse a aplicaciones específicas, lo cual ha hecho que el hardware sea diseñado para cumplir con ciertas especificaciones. De esta manera surgen una serie de inconvenientes producidos por la incompatibilidad entre dispositivos de radio, tal es el caso del ejército norteamericano. Puesto el problema en escena, surge la necesidad de prescindir de sistemas de radio con arquitectura cerrada, debido a que no presentan la flexibilidad necesaria para adaptarse a los cambios vertiginosos que la tecnología presenta día a día.

De esta manera en 1992, el *Department of Defense* (DoD) de los Estados Unidos [2], decide empezar el proyecto de desarrollo de un Radio Definido por *Software*, que tuvo sus inicios en 1970, tanto en Europa como en dicho país. El proyecto fue bautizado con el nombre de “*SpeakEasy*” y su objetivo inicial fue el de operar en la banda de 2 a 2000 MHz. A mediados de 1995 el proyecto fue un verdadero éxito por conseguir todas las metas planteadas, lo cual fue un gran argumento para el desarrollo de la segunda etapa que iniciaría el mismo año y finalizaría en el año 2000. La segunda etapa consistía en darle la funcionalidad de un sistema de radio completo, que a complemento de la primera etapa, tenía una funcionalidad de *modem* [7].

Radio Definido por *Software* es un campo relativamente reciente tanto en el ámbito industrial como en el inalámbrico. Es por esta razón que en el año 2000, varios radioaficionados empiezan a generar proyectos enfocados hacia el desarrollo de esta reciente tecnología. Uno de estos es *SDR-1000* mostrado en la Fig. 2.8, que fue lanzado al mercado en el año 2003 por parte de *FlexRadioSystems*. Para el año 2001, un grupo liderado por *Eric Blossom*, denominado *GNU Radio*, empieza el desarrollo de una herramienta para el procesamiento de señales [6]. Poco tiempo después *Matt Ettus* desarrolla una tarjeta capaz de proveer la sección *front-end* de la radiocomunicación. Así

nace la denominada *USRP*, que se detalla en al Fig. 2.9; dispositivo a ser utilizado en el diseño y posterior desarrollo de la presente tesis.



Figura 2. 9. Universal Software Radio Peripheral.

Pero en definitiva, cómo puede definirse a una Radio Definida por Software. Para responder a esta pregunta se juntan dos grandes entidades como son el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) y el grupo encargado del estudio de esta tecnología, denominado *SDR Forum* (ahora *Wireless Innovation Forum*³). Ambas entidades ven la necesidad de describir de forma clara y concisa el concepto de SDR, por lo cual es posible describirlo como:

“Es una radio en la cual alguna o todas las funcionalidades de la capa física están definidas por software” [1]

Para aclarar lo mencionado anteriormente, se debe decir que, un radio es todo dispositivo capaz de transmitir y/o recibir información de forma remota mediante señales RF.

Los dispositivos de radio son muy utilizados en la actualidad, ya sea en: celulares, computadores, vehículos, televisión, dispositivos de posicionamiento global, etc., los mismos que prestan grandes

³ Disponible en: <http://www.wirelessinnovation.org>

funcionalidades, sin embargo en un mundo tan cambiante, tecnológicamente hablando, estos dispositivos se ven relegados por nuevos dispositivos con mejores prestaciones.

Tradicionalmente la gran mayoría de los dispositivos de radio tienen limitaciones de funcionalidad cruzada, por esta razón a la hora de tratar de actualizarlos, el usuario se enfrenta contra un inconveniente muy común, que es el de hacer la modificación física, es decir, en el hardware, lo cual hace que sea un proceso complejo y costoso a la vez. Entonces surge la tecnología SDR, la misma que permite erradicar esta limitación por medio de una actualización del software o del firmware, lo cual evidencia un gran beneficio, no solo económico, sino además ambiental [2].

La tecnología SDR, representa una revolución en la forma en que los radios son diseñados, fabricados, desarrollados y usados.

La evidente y creciente necesidad de los usuarios de comunicarse de un lado del mundo a otro, genera el vertiginoso incremento de sistemas de comunicaciones, lo cual establece un gran inconveniente debido a la necesidad de modificar e incrementar la capacidad de estos sistemas, lo cual convierte a las comunicaciones en un negocio ciertamente crítico. Así es como se considera importante incluir la tecnología SDR debido a sus grandes prestaciones, como son:

- Gran flexibilidad, le permite un mayor tiempo de vida en lo referente al hardware, por lo tanto un bajo costo.
- Reconfiguración por medio de *software*.
- Mayor eficiencia.
- Usa Dispositivos Digitales Programables como *Digital Signal Processor* (DSP) o *Field Programmable Gate Array* (FPGA).
- Inteligencia manipulada por *software*.

- Reduce el tiempo y costo del mantenimiento.
- Capacidad de escalabilidad.
- Gran capacidad de desarrollo en varias áreas de comunicación, etc.
- Según Paul Burns, el mayor reto que tienen hoy por hoy los *SDR* es igualar la flexibilidad e inteligencia que proporciona el *software*, en las soluciones que se dan al *hardware* [6].
- Esto hace suponer que en respuesta a las necesidades del usuario final, se necesita un gran desarrollo en la flexibilidad que prestan los *chips* de silicio. Además en cuanto a los desarrolladores, o programadores, se refiere, es necesario un entorno simple para el manejo de un lenguaje de alto nivel que permita un desarrollo enfocado en la reconfiguración inteligente de estos dispositivos.

En lo que cabe al crecimiento en la potencia computacional de los Circuitos Integrados de Silicio, dentro del siglo XXI, se puede vislumbrar un futuro muy prometedor en la expansión en el campo de las comunicaciones de radio, ya que el *SDR* desempeñará un papel fundamental dentro del mismo.

2.2.2. Características Técnicas

Al igual que cualquier dispositivo de comunicaciones, un *SDR* posee un grupo específico de características que lo definen como tal, y es así que la arquitectura general del mismo es la que permite la visualización de dichas características.

De manera general, un *SDR* se divide en dos grupos, el primero hace referencia a la sección digital y el segundo a la sección analógica, como se observa en la Fig. 2.10 [6].

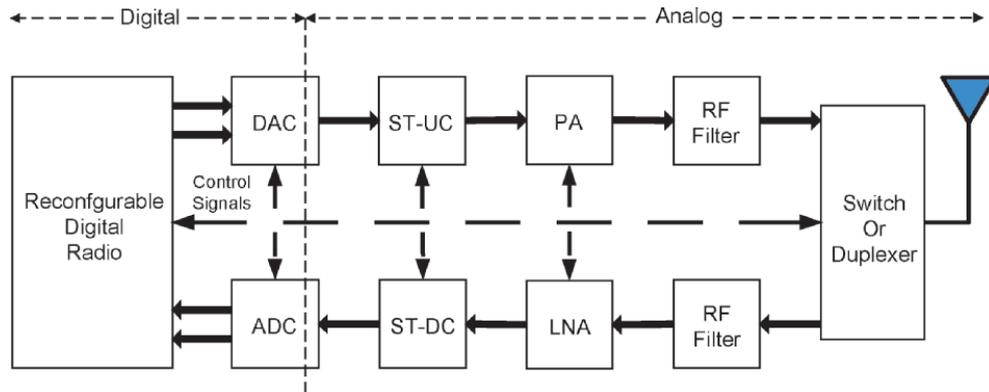


Figura 2. 10. Arquitectura de un SDR.

Las características de la sección digital son:

- Radio Digital Reconfigurable, que puede usar una de las tecnologías usadas comúnmente por el SDR, como un Procesador Digital de Señales o también un *FPGA*, en los cuales se realizará el procesamiento y control de la información recibida y también de aquella a ser transmitida.
- Conversor Análogo/Digital y Digital/Análogo (*DCA & DAC*), que se encargan de la transducción tanto para recepción como para transmisión. Además realizan la valoración respectiva sobre las señales. Es importante destacar que mientras estos tengan un mayor bus de datos, capacidad computacional y muy bajo tiempo de conversión, ayudarán a un menor número de errores y un ahorro significativo de energía.

En cuanto al manejo de la sección analógica, es importante destacar las siguientes características:

- Software Sintonizador Up Converter (*Software Tunable Up Converter, ST-UC*), se encarga de sintonizar la frecuencia digital intermedia (IF) de la señal proveniente del conversor Digital/Análogo y elevar dicha frecuencia a una de radio frecuencia. Estos generalmente consisten de un *software* atenuador sintonizable y un reloj sintetizador.
- Amplificador de Potencia (*Power Amplifier, PA*), en esta sección, la señal RF es amplificada, comúnmente por un atenuador sintonizable por

software que se del control adaptativo del nivel de energía a ser transmitido.

- Filtro RF Transmisor (*RF Filter*), una vez amplificada la señal, la misma es filtrada a fin de ser radiada correctamente por la antena a través del aire.
- Switch o Duplexador (*Switch or Duplexing*), depende mucho del método de duplexación escogido que puede ser en tiempo (TDD) o en frecuencia (FDD). Para el caso de la duplexación en tiempo, transmisión y recepción tendrán ciertos tiempos dedicados a cada uno, mientras que en la duplexación por frecuencia, es posible transmitir y recibir a la vez.
- Filtro RF Receptor (RF Filter), de igual manera que en la transmisión, un filtro de radio frecuencia se hace necesario en la recepción, con la finalidad de evitar todas aquellas señales que no son de interés.
- Amplificador de Bajo Ruido (Low Noise Amplifier, LNA), es un amplificador encargado de amplificar únicamente la señal en la frecuencia deseada, y que el ruido adherido a la misma sea eliminado por medio del uso de atenuadores digitales.
- Software Sintonizador Down Converter (*Software Tunable Down Converter, ST-DC*), de forma opuesta a la mencionada con en ST-UC, realiza una conversión de la señal proveniente en RF a la frecuencia intermedia. Estos generalmente consisten de un *software* atenuador sintonizable y un reloj sintetizador.

Lo descrito recientemente representa la arquitectura característica de un SDR en la actualidad, que como se puede observar es reconfigurable en todos los aspectos ya que todos sus bloques están controlados por medio de *software*, lo cual le hace merecedor de una gran flexibilidad. Sin embargo, debido a las limitaciones que presentan los Circuitos Integrados utilizados en el diseño, hacen que el mismo tenga varias limitaciones principalmente en frecuencia y potencia.

Entonces surge la incógnita sobre un SDR ideal. La respuesta a esta interrogante deberá mantener como principio fundamental la capacidad reconfigurable del software y deberá consistir básicamente de tres etapas. La primera es el Radio Digital Reconfigurable, la segunda será un Radio Análogo sintonizable por Software con un Sintetizador de Impedancia embebido y por último un Sistema de Antena sintonizable por Software, como se detalla en la Fig. 2.11.

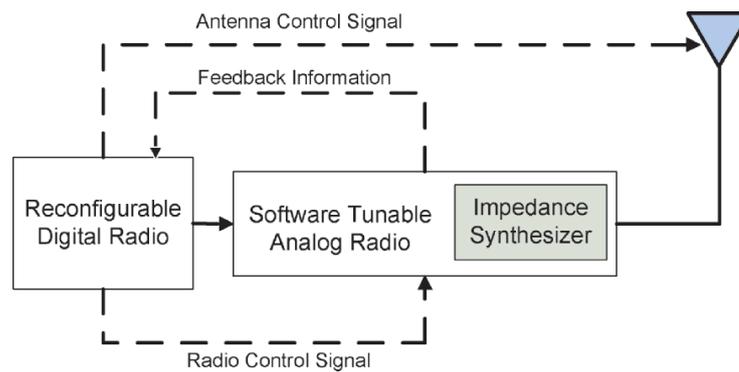


Figura 2. 11. Arquitectura ideal de un SDR.

Sin embargo, en la actualidad, el diseño de un *SDR* ideal, representa un aumento en el costo significativo, ya que incluye un bloque continuo de retroalimentación, el mismo que infiere en la creación de nuevos chips capaces de realizar dicha tarea. A pesar de ello, el desarrollo en *software* y paralelamente de algoritmos que optimicen el funcionamiento de los bloques, ayudarán a conseguir nuevas y mejores metas que beneficiarán a muchos sectores que han puesto sus ojos en esta tecnología.

2.2.3.Servicios

Debido a la gran flexibilidad que presentan los *SDR*, se convierten en un paso estratégico a dar por parte de los proveedores de servicios, ya que gracias a su bajo costo, es posible añadirlos a su infraestructura y con ello mejorar las redes, sin la necesidad de una gran inversión de capital.

Uno de los servicios prestados por esta tecnología es la de actualización de los sistemas vía software, es por esta razón que el beneficio económico

que esto representa hace posible que los usuarios finales, no necesariamente deban adquirir nuevos dispositivos, ya que como se menciona, es una gran ventaja mantener la información y además realizar una actualización de la tecnología que están utilizando.

En cuanto al ámbito militar se refiere, es posible que en batalla, los soldados tengan la capacidad de comunicarse inalámbricamente. Además de esto, como se sabe, históricamente las comunicaciones militares han sido y serán un punto muy frágil y la columna vertebral de muchas civilizaciones, esto se debe al hecho de que se necesita protegerlas de intrusos no deseados que puedan interferirlas y usarlas en contra. Debido a este inconveniente de magnitudes catastróficas, los *SDR* presentan la robustez necesaria para proteger las comunicaciones, ya que gracias a su flexibilidad en cuanto al control de la frecuencia de la señal por medio del uso del *software*, sea hace viable el cambio paulatino de la misma, a fin de tener una comunicación segura [24].

En cuanto al ámbito inalámbrico, los *SDR* se presentan como la tecnología emergente para el desarrollo de las comunicaciones en este campo. Es así que es una tecnología que ha tenido gran éxito en proyectos enfocados a prestar servicios *WiFi* y *WiMAX*. Un ejemplo de este último se tiene a Motorola, que desarrollo un proyecto denominado: “*Single RAN for WiMAX Evolution*”⁴.

⁴ White paper disponible en:

http://www.motorola.com/web/Business/Solutions/Industry%20Solutions/Service%20Providers/Network%20Operators/Wireless%20Broadband/wi4%20WiMAX/_Document/StaticFile/Single%20RAN%20White%20Paper-FINAL.pdf?localeId=252

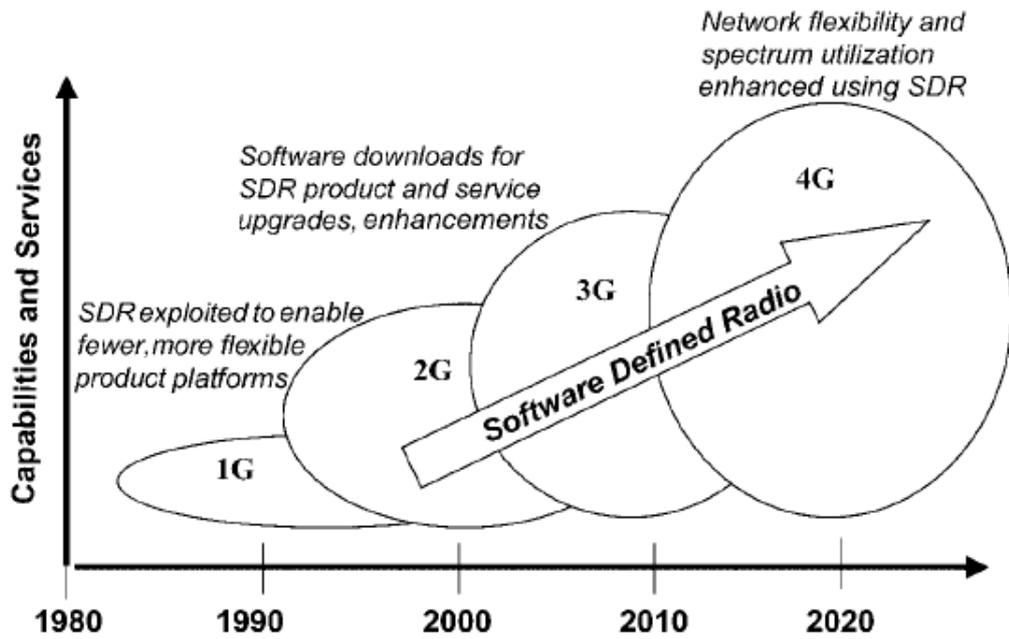


Figura 2. 12. Evolución del SDR.

En cuanto a las redes celulares, la tecnología *SDR* representa el salto inminentemente necesario, ya que gracias a las características ya citadas, representan un ahorro económico significativo para los proveedores de servicios y en consecuencia para el usuario final. Esta tecnología es usada en las redes de tercera generación (3G) y actualmente ya se está usando en las redes de cuarta generación (4G), ver Fig. 2.12, debido a la gran cantidad de servicios a ser ofrecidos al usuario. Esto representa un gran potencial en cuanto al soporte de múltiples interfaces aéreas para múltiples estándares; todo esto gracias a los grandes beneficios que representan los radios reconfigurables.

Es evidente que todos estos servicios y aplicaciones generan nuevos retos para los diseñadores tanto de *software* como de *hardware*. En el futuro inmediato, el mercado abrirá la posibilidad de usuarios cada vez más exigentes, lo cual hará que sea necesario el diseño de nuevas aplicaciones e interfaces más sencillas y fáciles para el manejo y mantenimiento de los usuarios.

2.3. Introducción a *USRP*

2.3.1. Definición y Desarrollo

Un *USRP* [5], es un Sistema Periférico Universal de Radio, que se define de este modo ya que desde su creación la idea general fue permitir a un equipo computacional, funcionar como un software de radio de alto rendimiento y con capacidad de operación de banda ancha, donde el computador sirve como una banda base digital y como una sección de frecuencia intermedia del sistema de radio comunicaciones.

El *USRP* está perfectamente diseñada para permitir un acople ideal eléctrica y mecánicamente con otras tarjetas adicionales como son: transmisores, receptores, receptores de televisión, etc., consiguiendo un traslado ideal de la información entre la sección banda base con la de IF y la de RF con la frecuencia a la que se desea transmitir.

En sí lo que se pretende mediante el acoplamiento PC-*USRP* es procesar integralmente toda forma de onda, lo que implica que la modulación y demodulación, estarán en el computador y las operaciones que requieran alta velocidad, digitalización, conversión, decimación e interpolación se realizan en el *FPGA*.

La parte neurálgica del *USRP* está en el *FPGA*, por lo que los diseñadores han considerado que los usuarios pueden trabajar ilimitadamente con la configuración inicial y estándar del *FPGA*, sin embargo, si la situación amerita y se requiere cambiar dicha configuración, esto es posible ya que la interface está diseñada, definida y documentada para ejecutar dichas acciones [5].

Es posible afirmar que el *USRP* posee un diseño modular basada en una tarjeta madre con cuatro ranuras de expansión, cada una de las ranuras se encuentra etiquetada como *TXA*, *RXA*, *TXB*, *RXB*, respectivamente y la organización de los buses *Serial Peripheral Interface* (*SPI*) es realizada de tal manera que si se ocupa las cuatro ranuras de expansión, las tarjetas se

observarán, una invertida con respecto a la otra. Esto nos permite realizar múltiples configuraciones, es decir, podemos conectar dos tarjetas con capacidad de transmisión y dos tarjetas para recepción o también se puede conectar dos tarjetas *tranceiver*.

2.3.2. Características Técnicas

Un *USRP* mostrado en la Fig.2.13 está conformado por las siguientes partes:

- *FPGA* ALTERA
- Conversor Análogo Digital
- Conversor Digital Analógico
- *Socket* de alimentación a corriente continua
- Controlador *USB*

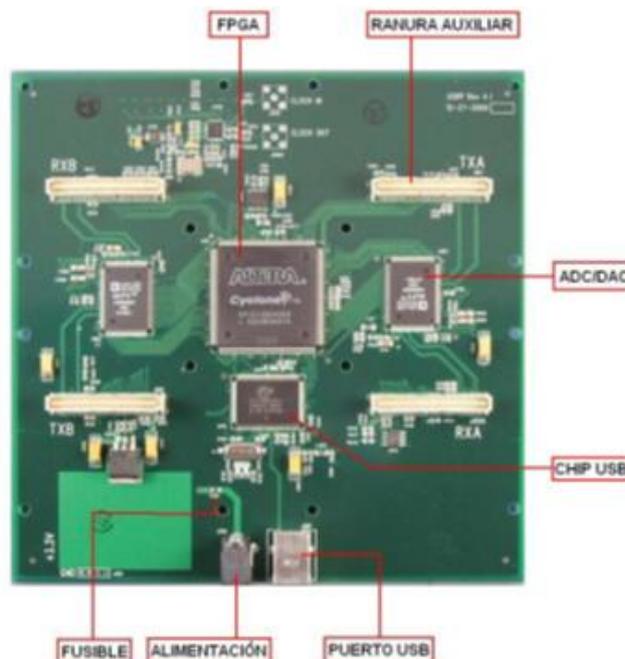


Figura 2. 13. Tarjeta Main Board del USRP.

Además tiene la capacidad de incrementar sus características de funcionamiento mediante el acoplamiento de tarjetas hijas, en los *Sockets* predestinados para las mismas.

Cada *daughterboard* o módulo adicional para el *USRP* cuenta con una memoria *EEPROM* (24LC024 ó 24LC025) que le permite identificarse automáticamente el momento que el *USRP* arranca, en caso de que la memoria no se encuentre grabada, el *USRP* emite un mensaje de advertencia, así mismo cada una de estas tarjetas tiene acceso a dos de los cuatro conversores AD/DA [4]. Los módulos adicionales que se pueden conectar son:

Basic TX/RX

Opera en el rango de frecuencias de 1 MHz a 250 MHz, además se puede usar como un externo para un sintonizador de subida o de bajada, además que también se puede usar como generador de señales, pero es importante considerar que para operar estas tarjetas en algunos casos se requiere un amplificador de radio frecuencia externo.

Este módulo permite el acceso a todas las señales del *USRP* incluyendo señales de 16 bits o entradas/Salidas digitales de alta velocidad, SPI y buses I2C, además de baja velocidad para los *ADC's* y *DAC's* [24].

LFTX/LFRX

Opera en la banda de hasta 30 MHz y son tarjetas similares a las Basic, sus diferencias puntuales son dos: la primera que LFTX y LFRX cuentan con amplificadores diferenciales, en lugar de transformadores y además cuenta con filtros pasa-bajos para evitar aliasing.

TVRX

Opera en la banda de 50 MHz a 860 MHz, es decir, en las bandas VHF y UHF; su uso es específicamente para el tratamiento de señales de televisión

únicamente en recepción y para canales que tengan un ancho de banda de 6 MHz. No puede operar con arreglos de antena tipo MIMO.

DBSRX

Es muy similar a TVRX, ya que es una tarjeta que sirve únicamente para recibir señales en el rango de 800 MHz a 2.4 GHz, con capacidad MIMO, además cuenta con un canal controlable que puede actuar como un filtro tan estrecho de 1 MHz hasta un amplio de 60 MHz [4].

RFX900

Su rango de operación va desde los 750 hasta los 1050Mhz, aunque posee un filtro para la banda *ISM* 902-928Mhz que puede ser desactivado para trabajar en las frecuencias de las operadoras celulares entregando una potencia de transmisión de 200mW.

RFX2400

Opera en el rango de frecuencias de 2.3GHz a 2.9Ghz, como el modelo anterior mediante filtros se puede disminuir el ancho de banda para seleccionar la banda *ISM* 2400 a 2483Mhz con una potencia de 500mW, ideal para experimentar en las frecuencias de trabajo correspondientes a la familia de estándares *802.11 b/g*.

XCVR2450

Su rango de operación es en dos bandas la una va de 2.4GHz a 2.5Ghz y la otra va de 4.9GHz a 5.9Ghz completando así todas las frecuencias utilizadas por el estándar 802.11a y adicionalmente implementaciones *WiMAX* con frecuencia libre.

FPGA

El *FPGA* es una parte fundamental en el acoplamiento *FPGA-USRP*, ya que todos los convertidores tanto análogos digitales como digitales analógicos se conectan al *FPGA*, lo que el *FPGA* hace es reducir al mínimo al amplio

espectro de operaciones matemáticas para poder enviar o transmitir una tasa mínima de datos a través de un puerto serial *USB 2.0*.

A continuación se muestra una figura muy ilustrativa en lo referente a la estructura y operación del *FPGA* dentro del *USRP*.

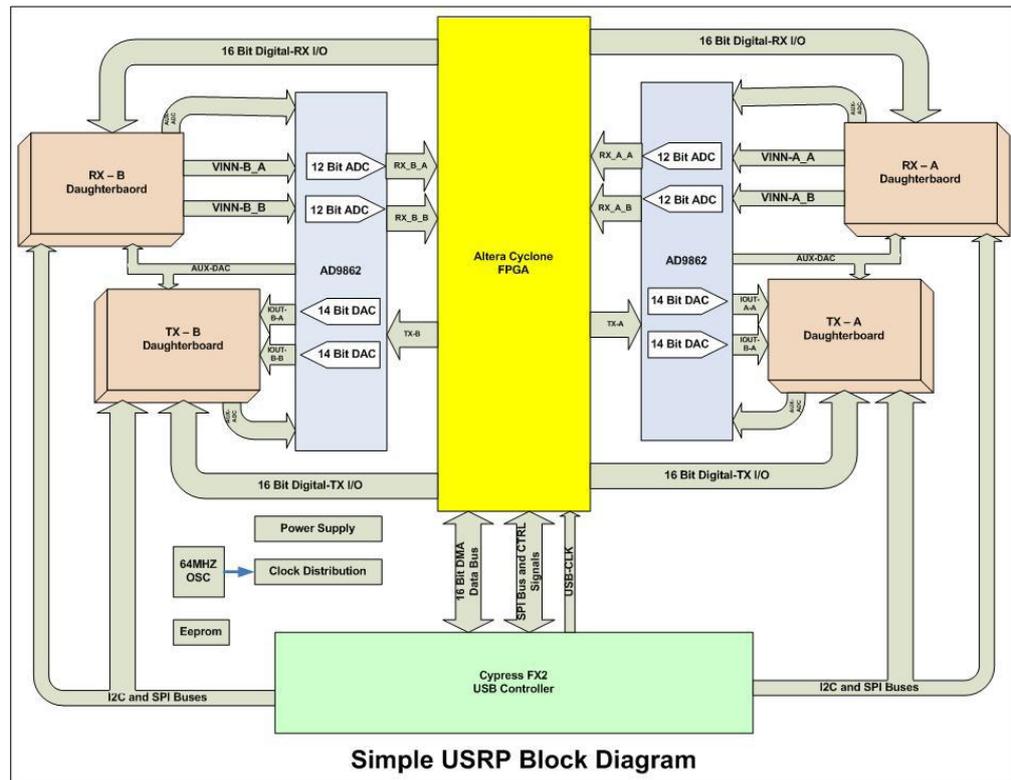


Figura 2. 14. Diagrama de Bloques del *FPGA* del *USRP*.

La configuración estándar del *FPGA* mostrada en la Fig. 2.14 [5], incluye digital *down converters* que permiten centrar la señal digitalizada de una frecuencia intermedia en una banda base, los cuales están implementados en cuatro estaciones en cascada y que son filtros integradores (*Cascade integrated comb CIC*), muy útiles para establecer retardos [4].

En la implementación de los cuatro *DDC* en la vía de recepción cuenta con 4 *ADC*'s y cuatro *DDC*'s cada uno con dos entradas I y Q, que permiten tener una amplia gama de canales de salida.

Dentro del *DDC* existe una etapa de decimación, en la cual el decimador es tratado como un filtro pasa bajos, seguido por un sub muestreo, por ejemplo si suponemos que el factor de decimación es N , la salida del filtro del espectro digital pasa bajos será $[-F_s/N, F_s/N]$, y el sub muestreo se extenderá desde $[-F_s, F_s]$ hasta $[-F_s/N, F_s/N]$, de hecho se cuenta con un ancho de banda muy estrecho para las señales digitales de interés.

Respecto al ancho de banda se puede sostener 32 MBps. a través del puerto *USB*, todas las muestras enviadas a la interface *USB* son de 16 *bits*, signadas como números enteros y en formato IQ, es decir 16 *bits* I y 16 *bits* Q formando un número complejo, lo que implica que la muestra es un número complejo de 4 bytes o lo que es lo mismo 4 muestras complejas de 4 bytes. Esto resulta 32 Mega bytes por segundo por cada 4 Bytes, que en síntesis resultan en 8 Mega muestras complejas por segundo enviadas por el puerto *USB*, lo que según el criterio de *Nyquist* implican 8 MHz de ancho de banda.

Para ejemplificar un caso práctico se puede afirmar que si el ancho de banda de una estación FM es 200 kHz, el factor de decimación a seleccionar puede ser de 250, entonces la tasa de datos a través del puerto *USB* es de 64 MHz/250 que es igual a 256 kHz, que pasan sin pérdidas en el espectro en un ancho de banda de 200 kHz. Finalmente se debe conocer que el factor de decimación está entre 8 y 256.

En el camino de transmisión la situación es muy parecida, pero ocurre de forma inversa, aquí no se usa el *DDC* sino el *DUC* digital *up converter* que permite pasar de banda base a frecuencia intermedia y enviarla a través de los *DAC's*.

Para la transmisión el dato para la tasa de transmisión se manipula a través de la interpolación, la cual permite que todos los canales de transmisión puedan tener la misma tasa de datos, pero es importante saber que la tasa de transmisión puede ser diferente a la tasa de recepción.

El *USRP* puede operar en modo full dúplex, en el cual los lados de transmisión y recepción son independientes el uno del otro y la única consideración que deben tener es que la tasa de datos debe ser de 32 MBps o menos.

2.3.3. Interfaces

La interfaz de conexión al *USRP* v.1 es únicamente un puerto *USB*, mientras que entre muchas de las modificaciones realizadas al *USRP2*, se consideró un puerto *Ethernet*.

Para el caso del *USRP1*, como se mencionó, se utiliza una interfaz *USB* que en la etapa de comunicación con el computador se encuentra un chip *Cypress FX2* que contiene un micro controlador *USB 2.0* no compatible con la versión 1.0 y por el lado del *FPGA* se conecta mediante una *GPFI* (*General Purpose Interface*). Para separar las diferentes operaciones realizadas en el bus *USB* se utilizan tan solo tres *endpoints*, donde las operaciones más comunes son transmisión, recepción y control, una descripción detallada la se observa en la Tabla 2.5.

La velocidad máxima soportada por el bus *USB* es de 32MBps y todas las muestras que se envían por este bus son de tipo *signed integer* de 16 bits en formato *IQ* es decir 16 bits *I*, 16 bits *Q* lo que significa un costo de 4 bytes por muestra en cuadratura.

Tabla 2. 5. Pines de Funcionamiento del puerto *USB*.

Endpoint	Descripción
0	Estado de Control
2	Host a <i>FPGA</i>
6	<i>FPGA</i> a host

2.3.4. Conversión Análoga Digital *ADC*

El *USRP* cuenta con 4 convertidores analógicos digitales de alta velocidad, de 12 bits, su tasa de muestreo es de 64 MS/s (Millones de

muestras por segundo), que permite iniciar digitalizando una banda ancha de 32 MHz, estos conversores pueden muestrear señales en bandas de aproximadamente 200 MHz [3].

Es importante considerar que si se toleran varios decibeles de pérdidas, entonces se pueden digitalizar hasta 500 MHz, pero si se muestrean frecuencias intermedias mayores que 32 MHz, entonces se introduce aliasing a la señal, por lo que es mejor trabajar únicamente entre - 32MHz y 32 MHz., con lo que incluso se podrían escuchar señales de estaciones FM sin utilizar equipos terminales de radiofrecuencia, por lo que finalmente se recomienda trabajar como máximo con señales de 100MHz, para evitar excesiva relación señal ruido.

El rango completo de convertidor es de 2(V) pico-pico, su voltaje de referencia VREF es de 1 (V) y las entradas diferenciales son de 50 ohmios, lo que implica que se opera con 40 mW o 16dBm, además se cuenta con un amplificador de ganancias programable (PGA), que actúa especialmente en las señales débiles.

2.3.5.Conversión Digital Analógica DAC

Son cuatro conversores digitales analógicos de alta velocidad, que operan con 14 bits, la frecuencia del reloj es de 128MS/s, así la frecuencia de Nyquist es 64 MHz, sin embargo para facilitar el filtrado es mejor trabajar por debajo de 64MHz, pero en la práctica se usa 44 MHz [6].

El *DAC* debe ser alimentado por 1 (V) pico-pico y cargado con impedancias de 50 (ohm), 10 (mW) o 10 (dBm), de igual forma que el *ADC* cuenta con un PGA que entrega hasta 20dBm de ganancia [3].

2.3.6.Entradas y Salidas Analógicas

Cuenta con 8 canales auxiliares de entrada analógica, conectadas a las entradas del conversor análogo digital a baja velocidad por medio de 10 bits, los cuales se pueden ser leídos a través de *software*.

Estos canales permiten convertir hasta 1,25 MSps. (millones de muestras por segundo), y tienen un ancho de banda de hasta 200 KHz, estos canales son útiles para medir señales RSSI (*Received Signal Strength Indication*: Indicador de la Intensidad de la señal recibida), temperaturas, inclinaciones, niveles, etc.

Adicionalmente estos son 8 canales analógicos de baja velocidad conectados a las salidas del *DAC* a través de 8 bits.

En resumen el *USRP* los conectores *RXA* y *RXB* comparten un sistema de 4 canales de salida analógica y cada uno tiene 2 entradas independientes de canales analógicos, de igual forma *TXB* y *RXB* operan de forma similar, pero además cuenta con una entrada adicional para un voltaje de referencia que se utiliza únicamente cuando las circunstancias lo requieran.

2.3.7. Entradas y Salidas Digitales

El *USRP* cuenta con 64 puertos en entrada y salida digital, divididos en 2 grupos de 32 bits tanto para entradas y salidas de transmisión, como para entradas y salidas de recepción. Estos pines están conectados a las interfaces de conexión de las tarjetas hijas del *USRP*.

Cada conector (*RXA*, *RXB*, *TXA*, *TXB*) tienen 16 bits digitales de entrada y salida.

Estos pines son controlados desde *software* y leídos y controlados por medio del *FPGA*, además cualquiera puede ser configurada como entrada o salida digital sin excepción.

Algunos pines tienen una función específica en las tarjetas hijas, como son la de controlar la recepción de RF, colocar en modo automático transmisión/recepción, controlar la potencia de alimentación de diferentes partes del *TX/RX*, detección de bloqueos del sintetizador, además son de mucha ayuda para depurar las implementaciones del *FPGA* que es conectado a analizadores lógicos.

2.3.8. Potencia

El *USRP* es alimentado por un convertidor de corriente alterna a corriente directa AC/DC que provee 6 [V] y 4 [A]. El convertidor es capaz de operar entre 90-260 [VAC], a una frecuencia de 50/60 Hz, lo que lo hace universal para funcionar en cualquier lugar del mundo.

La placa base *USRP* requiere únicamente 5V, pero debido a que se debe alimentar a las tarjetas hijas el *USRP* cuenta con una fuente de 6 [V]. En cuanto al consumo de corriente, está alrededor de 1.6 [A], considerando operar con 2 *daughterboards* fijas en el *USRP*.

La potencia de las computadoras actuales permiten modular y demodular señales complejas, así que es posible utilizar la versatilidad del *software* para ahorrar recursos en la parte hardware, lo que hace compatible a un computador con un *USRP*.

2.4. Instalación de *software* y hardware requerido para manejo de *USRP*

2.4.1. Identificación de Interfaces

A continuación se detallan la interfaz a ser utilizadas a fin de obtener una comunicación entre la tarjeta *USRP* y el usuario. Es así entonces, que el usuario utilizará una interfaz, con la finalidad de controlar dicha tarjeta. En este caso, la interfaz Gráfica está representada por el *software Simulink*[®] de *Matlab*[®], del cual más adelante se hablará con mayor profundidad, pero es importante destacar que una interfaz se define cómo:

“Conexión física y funcional entre dos aparatos o sistemas independientes.”

Por lo tanto, y teniendo en cuenta la definición, la interfaz a ser utilizada, entre el ordenador y la tarjeta *USRP*, será la interfaz *USB (Universal Serial Bus)*. Esto se debe a que la tarjeta posee dicha interfaz, como ya se mencionó anteriormente.

▪ Interface *USB*

Alrededor del mundo, la interfaz *USB* que se observa en la Fig. 2.15 no solo es una simple interfaz, sino que también es la más conocida. Utilizada para conectar dispositivos como: cámaras, teclados, mouse, pantallas, escáneres, impresoras, módems, teléfonos móviles, discos duros, unidades ópticas, unidades de almacenamiento, etc., la convierten hoy por hoy en la interfaz más reconocida, confiable, rápida, versátil, económica y por supuesto compatible con casi todos los sistemas operativos del mundo. Es así que, dada su importancia, la gran mayoría de dispositivos que se necesita comunicarlos con un computador, a fin de darle al usuario una interfaz de control, vienen provistos con esta interfaz. Los datos, relativamente actuales, manifiestan que cerca de dos billones de dispositivos, alrededor del planeta, contienen a esta interfaz [20]

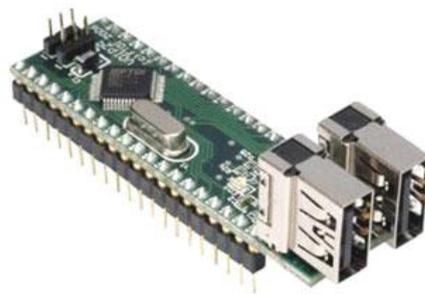


Figura 2. 15. Circuito e interfaz USB.

Gracias a la versatilidad que posee esta interfaz, y su fácil manejo, se puede comunicar con múltiples dispositivos, por lo cual al existir una gran demanda de ella, la hace cada vez más económica y fácil de usar, debido al gran desarrollo que ha tenido, pensando siempre en la facilidad que necesita el usuario para controlar diversos dispositivos. Debido a esta gran característica, es también usada en ciertos sectores industriales, a fin de comunicar a una máquina maestra con los dispositivos y máquinas esclavos.

- **Cypress EZ-USB**

La empresa *Cypress*, es una de los mayores fabricantes en el mundo de microcontroladores orientados al manejo de interfaces *USB*. Es así que la familia de semiconductores *EZ-USB* soportan las velocidades *High* y *Full Speed*. Poseen una gran variedad, en cuanto al *firmware* se trata, de arquitecturas. Algunos poseen la capacidad de almacenarlo, mientras que otros, como el caso actual, el *host* es el encargado de cargar el *firmware* sobre el microcontrolador [6].

- **Arquitectura**

La arquitectura del *Cypress EZ-USB* es muy similar a la del DS80C320 de *Maxim Integrated Products*. Este contiene un microcontrolador 8051 con un diseño mejorado para obtener un rendimiento óptimo en la transferencia de datos y además para que sea compatible con todo el conjunto de instrucciones. Posee una memoria RAM, memoria volátil, lo cual lo convierte en una interfaz de *High* y *FullSpeed*. Sin embargo, la gran desventaja que esto representa, se pone de manifiesto a la hora en que se desea que el *host* la reconozca, ya que al no disfrutar de una memoria no volátil, no es posible cargar directamente el *firmware*.

Esto no significa que no sea posible dotar al microcontrolador de una memoria no volátil. Es posible, mediante la conexión en paralelo con una memoria EEPROM.

- **Controlador USB**

Como ya se ha mencionado, el controlador puede estar conectado a una memoria externa o EEPROM a fin de localizar allí el *firmware*. Sin embargo, esta es una modificación a lo que comúnmente contiene el *EZ-USB*. El modo por defecto, es aquel en el cual el núcleo no detecta a la memoria externa, así entonces, es necesario que el dispositivo *host*

realice la enumeración, identificación y por último cargar el *firmware* sobre la *RAM*.

2.4.2. Instalación de *software* para manejo de *USRP*

Como se mencionó en el Capítulo I, el objetivo de este proyecto es el desarrollo de un *SDR* mediante el uso del hardware *USRP* y del *software Simulink*[®] de *Matlab*[®].

El proyecto *Simulink*[®]-*USRP* es un paquete de *software* de código abierto desarrollado por el *Communication Engineering Lab (CEL)*⁵ del *Karlsruhe Institute of Technology (KIT)*, de Alemania. El propósito que persigue este proyecto, es el de construir modelos en *Simulink* que interactúen en tiempo real con el hardware de la tarjeta *USRP*. La gama de desarrollo que permite este paquete de *software* es bastante amplia, desde procesamiento digital de señales hasta comunicaciones inalámbricas.

▪ Instalación del Controlador para *USRP*

En función de lo previamente mencionado, la tarjeta *USRP* contiene un microcontrolador *Cypress EZ-USB*, el mismo que no contiene una arquitectura suficiente para estar provisto del *firmware* necesario para su identificación en el *host*, en este caso una PC.

Por tal motivo se hace evidente la necesidad de instalar el *software* controlador para que la tarjeta *USRP*, pueda ser reconocida por el *host*. Se aclara que para este proyecto, se utilizará el Sistema Operativo *Windows*[®] 7 con arquitectura de *64 bits*.

Dicho *software* controlador se encuentra disponible en: http://www.cel.kit.edu/download/USRP_driver_1.1.zip de forma gratuita.

⁵ Disponible en: <http://www.cel.kit.edu/english/usrp.php>

Una vez descargado dicho controlador se procederán a seguir los siguientes pasos:

1. Descomprimir el paquete *USRP_driver_1.1.zip*, y ubicarlo en un directorio que sea fácil de recordar, como *C:\USRP_driver_1.1*
2. Conectar vía *USB* la tarjeta *USRP*. En este punto, el Sistema Operativo solicitará el controlador de la tarjeta. Entonces se deberá informar la ubicación del controlador que se lo debe elaborar primero obteniendo un archivo previo desde <http://sourceforge.net/projects/libusb-win32/files/>, seleccionando la opción *libusb-win32 releases*, para este caso se tomó la versión de abril-2011 que es el archivo *libusb/win32/bin 1.2.4.0.zip* de 848 KB, se debe tomar el archivo “*InfWizard*” ubicado en la carpeta *bin* dar clic en siguiente e instalarlos datos del *USRP* como muestra la Fig. 2.16, clic en siguiente y guardar el archivo creado en el directorio correspondiente. Finalmente ese será el driver a utilizar, cabe mencionar que esta complejidad se da para trabajar con sistemas operativos de 64 bits.

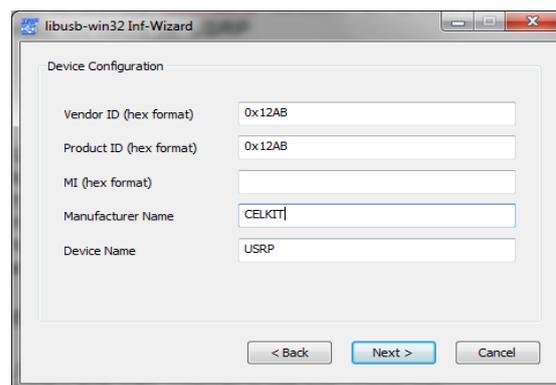


Figura 2.16. Configuración del driver USB.

3. Verificar en el Administrador de Dispositivos de *Windows* o *Device Manager*, que el dispositivo haya sido reconocido, tal como se observa en la Fig. 2.17.

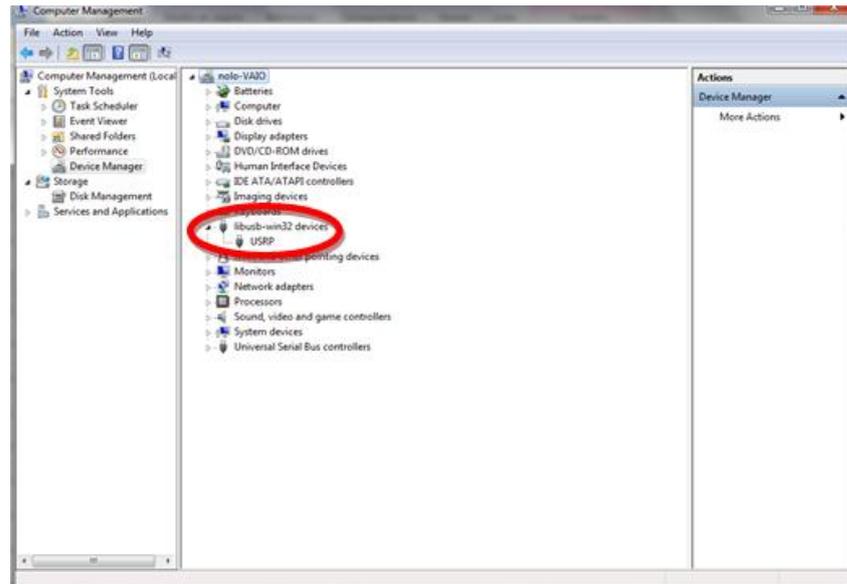


Figura 2. 17. Administrador de Dispositivos de *Windows*.

Es importante mencionar que este controlador es válido para *Windows 7*, *Windows Vista* y *Windows XP*.

▪ Instalación del Módulo *Simulink*[®] *USR*

Para esta sección, cabe destacar que se utilizará *Matlab R2010b* 64bits. Este fue el *software* escogido, debido a que no se desea tener inconvenientes de compatibilidad entre el Sistema Operativo y la plataforma de desarrollo de *Matlab*. Algo que se debe informar, es que el proceso que a continuación va a ser descrito, es compatible con cualquier versión de *Matlab* igual o superior a *R2010b*.

Como ya se manifestó en el inciso anterior, el módulo de desarrollo *Simulink-USRP*, es un *software* de código abierto que no fue desarrollado por *MathWorks*⁶, empresa encargada del desarrollo y comercialización del *software Matlab*, por lo tanto no viene integrado en la versión oficial de *Matlab*. Para lo cual es necesario descargarlo desde

⁶ Para mayor información disponible en: <http://www.mathworks.es/>

la web, disponible en: <http://www.cel.kit.edu/download/Simulink-USRP-2.1.3.zip> de manera gratuita.

Debido a que el módulo en cuestión no es propietario de *MathWorks*, es necesario compilarlo para que el mismo funcione sobre la versión de *Matlab* con la que se cuente. Para dicho propósito es necesario instalar el *software* Microsoft Visual C++ *Express Edition* 2008, que de manera gratuita está disponible en:

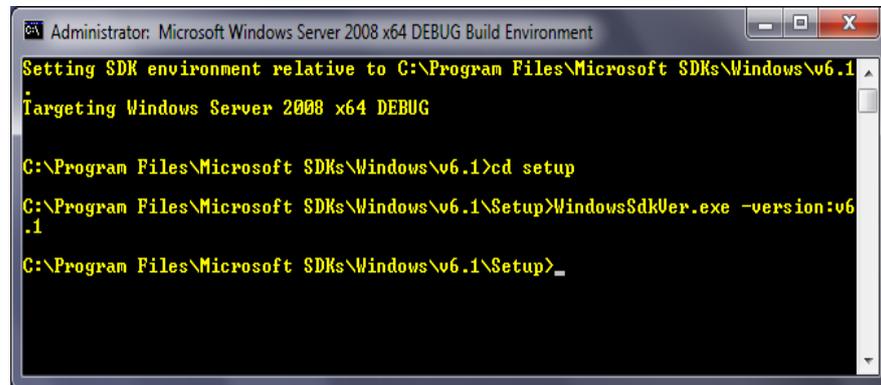
<http://www.microsoft.com/express/Downloads/#2008-Visual-CPP>.

Además de dicho *software*, es necesaria una herramienta de desarrollo como el SDK (*Software Development Kit*) de Microsoft para *Windows Server* 2008 y *.NET Framework* 3.5, disponible en: <http://www.microsoft.com/downloads/en/details.aspx?FamilyId=F26B1AA4-741A-433A-9BE5-FA919850BDBF&displaylang=en> de forma gratuita. El archivo a descargar es una imagen *.ISO*, la misma que puede ser grabada sobre un *DVD*.

A continuación se detallan los pasos a seguir para instalar el módulo:

1. Una vez descargados, tanto Microsoft Visual C++ *Express Edition* 2008 como SDK *Microsoft* para *Windows Server* 2008 y *.NET Framework* 3.5, proceder a la instalación respectiva de ambos programas, en este caso para 64 bits.
2. Es importante destacar que para versiones anteriores de *Matlab* se debía descargar *mexfiles* y copiarlos o reemplazarlos según sea el caso dentro de la carpeta *bin* de *Matlab*, sin embargo para la versión R2010a en adelante dicha acción no se la realiza, ni tampoco se verá crear variables de entorno adicionales; y lo que se debe hacer es activar el SDK instalado ingresando a la *CMD Shell* del sistema y digitar los siguiente como muestra la Fig. 2.18 y Fig. 2.19.

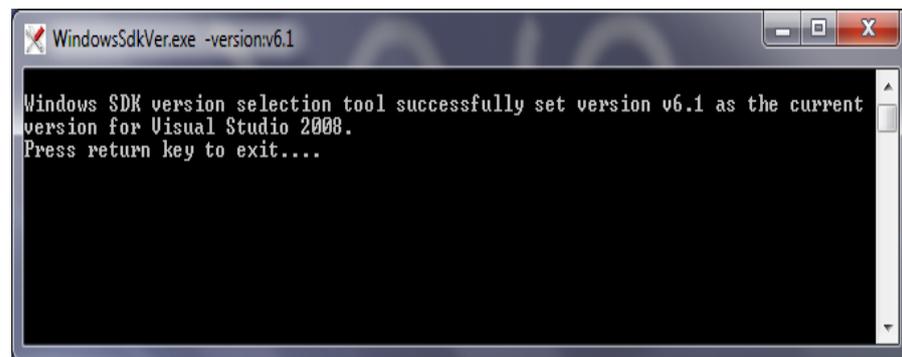
- `cd setup`
- `WindowsSdkVer.exe -version:v6.1`



```
Administrator: Microsoft Windows Server 2008 x64 DEBUG Build Environment
Setting SDK environment relative to C:\Program Files\Microsoft SDKs\Windows\v6.1
Targeting Windows Server 2008 x64 DEBUG

C:\Program Files\Microsoft SDKs\Windows\v6.1>cd setup
C:\Program Files\Microsoft SDKs\Windows\v6.1\Setup>WindowsSdkVer.exe -version:v6.1
C:\Program Files\Microsoft SDKs\Windows\v6.1\Setup>_
```

Figura 2. 18. Búsqueda del SDK.



```
WindowsSdkVer.exe -version:v6.1
Windows SDK version selection tool successfully set version v6.1 as the current
version for Visual Studio 2008.
Press return key to exit....
```

Figura 2. 19. Activación del SDK.

3. Posterior a esto, se deberá ingresar en *Matlab* y se procederá a digitar el comando `mex -setup`, y seleccionar el compilador, tal como se observa en la Fig. 2.20.

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

MATLAB desktop keyboard shortcuts, such as Ctrl+S, are now customizable.
In addition, many keyboard shortcuts have changed for improved consistency
across the desktop.

To customize keyboard shortcuts, use Preferences. From there, you can also
restore previous default settings by following the steps outlined in Help.

Click here if you do not want to see this message again.

>> mex -setup
Please choose your compiler for building external interface (MEX) files:

Would you like mex to locate installed compilers [y]/n? n

Select a compiler:
[1] Intel C++ 11.1 (with Microsoft Visual C++ 2008 SP1 linker)
[2] Intel Visual Fortran 11.1 (with Microsoft Visual C++ 2008 SP1 linker)
[3] Intel Visual Fortran 11.1 (with Microsoft Visual C++ 2008 Shell linker)
[4] Intel Visual Fortran 10.1 (with Microsoft Visual C++ 2005 SP1 linker)
[5] Microsoft Visual C++ 2005 SP1
[6] Microsoft Visual C++ 2008 Express Edition
[7] Microsoft Visual C++ 2008 SP1
[8] Microsoft Visual C++ 2010
[9] Microsoft Visual C++ 2010 Express

[0] None

Compiler: 6

```

Figura 2. 20. Selección de compilador en Matlab.

4. Descomprimir el módulo descargado y ubicarlo en: C:\Program Files\MATLAB\R2010b\Simulink-USRP_2.1.3.
5. Ahora es necesario direccionar a *Matlab* para que ubique el modulo a integrar en *Simulink*, por lo tanto es necesario fijar, tanto el *Blockset* como el *bin* del módulo, en el directorio donde éste se ha ubicado.

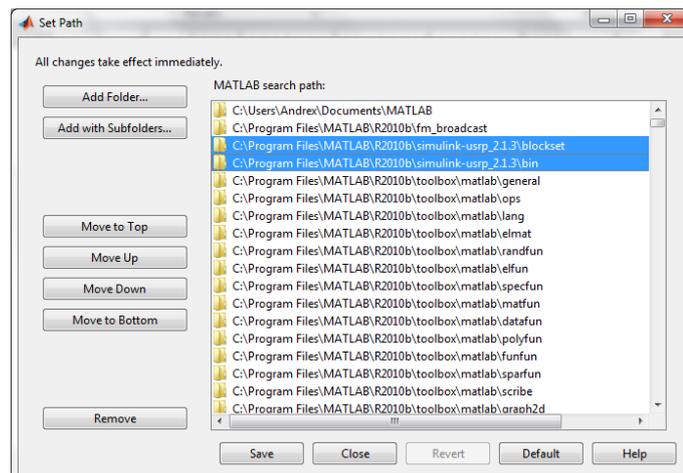
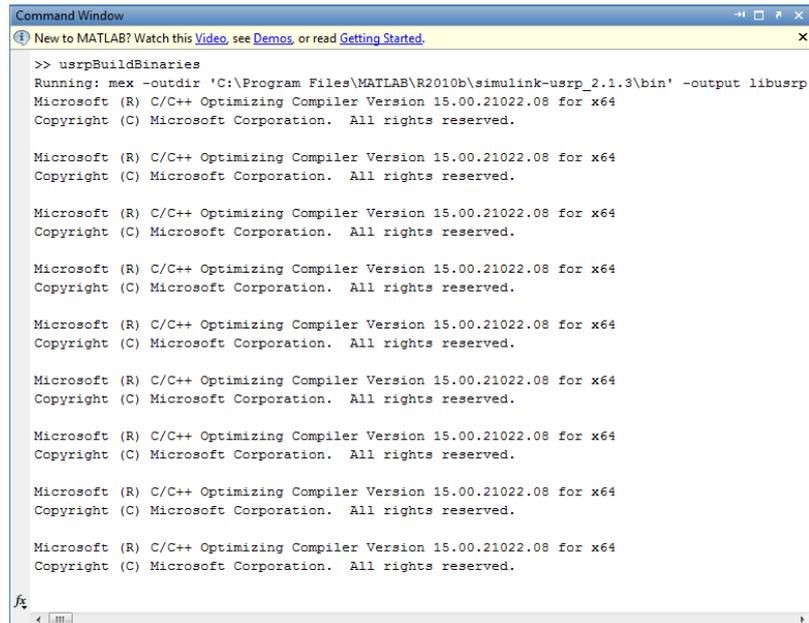


Figura 2. 21. Ajuste de directorio del módulo Simulink-USRP.



```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> usrpBuildBinaries
Running: mex -outdir 'C:\Program Files\MATLAB\R2010b\simulink-usrp_2.1.3\bin' -output libusrp
Microsoft (R) C/C++ Optimizing Compiler Version 15.00.21022.08 for x64
Copyright (C) Microsoft Corporation. All rights reserved.

Microsoft (R) C/C++ Optimizing Compiler Version 15.00.21022.08 for x64
Copyright (C) Microsoft Corporation. All rights reserved.

Microsoft (R) C/C++ Optimizing Compiler Version 15.00.21022.08 for x64
Copyright (C) Microsoft Corporation. All rights reserved.

Microsoft (R) C/C++ Optimizing Compiler Version 15.00.21022.08 for x64
Copyright (C) Microsoft Corporation. All rights reserved.

Microsoft (R) C/C++ Optimizing Compiler Version 15.00.21022.08 for x64
Copyright (C) Microsoft Corporation. All rights reserved.

Microsoft (R) C/C++ Optimizing Compiler Version 15.00.21022.08 for x64
Copyright (C) Microsoft Corporation. All rights reserved.

Microsoft (R) C/C++ Optimizing Compiler Version 15.00.21022.08 for x64
Copyright (C) Microsoft Corporation. All rights reserved.

Microsoft (R) C/C++ Optimizing Compiler Version 15.00.21022.08 for x64
Copyright (C) Microsoft Corporation. All rights reserved.

Microsoft (R) C/C++ Optimizing Compiler Version 15.00.21022.08 for x64
Copyright (C) Microsoft Corporation. All rights reserved.

Microsoft (R) C/C++ Optimizing Compiler Version 15.00.21022.08 for x64
Copyright (C) Microsoft Corporation. All rights reserved.

```

Figura 2. 22. Ejecución del comando `usrpBuildBinaries`.

- Para finalizar, se deberá digitar el comando `usrpBuildBinaries`, como se indica en la Fig. 2.22, para que *Matlab* integre el nuevo módulo en su herramienta de desarrollo *Simulink*, esto se observa de mejor forma en la Fig. 2.23.

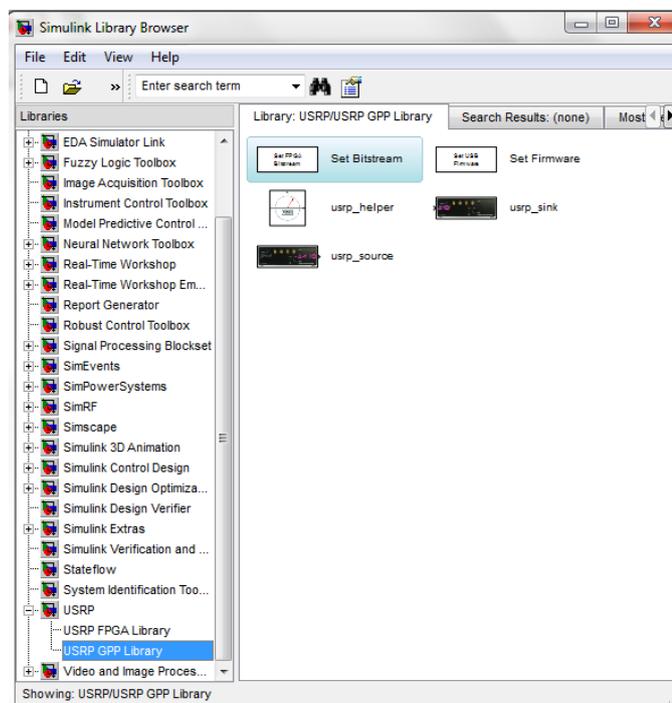


Figura 2. 23. Módulo Simulink-USRP.

Como muestra la Fig. 2.23, el módulo ha sido instalado satisfactoriamente. De no obtener este resultado, se recomienda verificar todos los pasos desde el principio.

2.5.El proceso de la Comunicación

El proceso cotidiano de comunicación, nunca es observado en su forma compleja e integral por parte de quienes hacen uso de la misma, en cualquiera de las formas en las que esta existe, así como también en el uso de las diferentes tecnologías que hacen posible la comunicación como son el teléfono, el celular, el internet o ya sea en una simple conversación; sin embargo, este proceso tiene sus requisitos definidos, sus etapas o fases muy bien identificadas, que debido a su larga lista de aplicaciones y su importancia en la vida diaria hacen que sus servicios sean fundamentales, casi interminables y en muchos casos imprescindibles para subsistir, dicho proceso se va a detallar más profundamente en la presente investigación.

Para que una comunicación se lleve a cabo, a más de existir la necesidad de comunicarse deben efectuarse los siguientes pasos: [11].

1. Generar la señal del mensaje, a través de voz, música, imagen, etc.
2. Describir el mensaje mediante símbolos sean: eléctricos auditivos o visuales.
3. Codificar adecuadamente los símbolos para transmitirlos por un medio determinado.
4. Transmitir los símbolos codificados al destino deseado.
5. Decodificar y reproducir los símbolos originales.
6. Recrear la señal del mensaje original.

En éste contexto la comunicación no engloba simplemente al ser humano, sino también otros equipos como los computacionales por ejemplo, donde el

hombre interviene únicamente para programarlos y configurarlos y estos pueden comunicarse luego sin la participación permanente del hombre.

En conclusión, para que exista comunicación, como indica la Fig. 2.24, se requieren mínimo tres elementos que son: Transmisor, Canal y Receptor; dónde el transmisor y el receptor están geográficamente distantes y el canal es el medio que permite la propagación del mensaje.

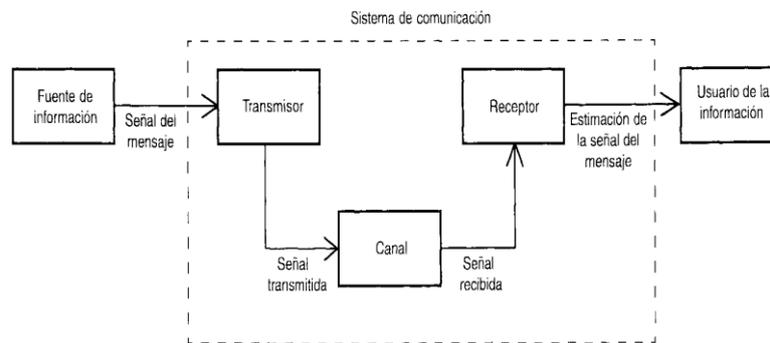


Figura 2.24. Elementos de una Comunicación.

2.6. Codificación y Decodificación de fuente

2.6.1. Definición

El codificador de fuente es el encargado de codificar la señal del mensaje producido por la fuente de la información, para luego transmitirlo por el canal respectivo, de forma más sencilla implica una transformación de la señal analógica en digital, en base a un proceso de muestreo y cuantificación, en fin el propósito de la codificación de voz es comprimir la señal para optimizar el uso del ancho de banda.

La información proporcionada por la fuente debe ser codificada de la mejor forma para que se haga posible su transmisión, por lo que el codificador de fuente deberá eliminar la señal redundante del mensaje, para optimizar el uso eficiente del canal, una vez que la información es codificada ésta se denomina la palabra código fuente y es ésta palabra la que finalmente se la envía a un siguiente proceso que es la compresión, cuyo proceso es opcional,

pero que es de gran ayuda para poder optimizar el uso del canal; luego de esto la información es enviada al proceso de codificación de canal.

La codificación de fuente consiste en establecer una correspondencia entre cada uno de los símbolos de un alfabeto fuente, tal como se indica en la Fig. 2.25, y una secuencia de símbolos de un alfabeto destino. Al alfabeto destino se le denomina alfabeto código y a cada una de las secuencias de símbolos de este alfabeto que se corresponda con un símbolo del alfabeto fuente se denomina palabra de código.

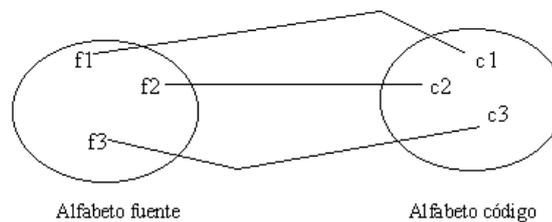


Figura 2. 25. Alfabeto fuente y alfabeto código.

Se debe precisar que la información puede estar definida como un archivo cualquiera de datos escritos o como un archivo de audio o de voz, por lo que para cada caso específico, se debe considerar las posibilidades existentes de codificación.

Para el caso de información no audible existen códigos de codificación como ASCII por ejemplo que es el código universal que utilizan los computadores, código *Baudot*, BCD, o un código de barras, etc.

De igual forma para el caso de que la información sea audible existen compresores de audio y sonido que permiten codificar esta información de forma adecuada para poder transmitirla eficientemente.

2.6.2. Compresión de la voz.

La tendencia de los servicios de telecomunicaciones, es siempre orientada proporcionar mayor movilidad, mayor confortabilidad, mayor velocidad, pero sin dejar de lado el aspecto fundamental que es la corrección

de errores, para obtener mayor velocidad se requiere una mayor tasa de transmisión de datos, pero en el caso del tratamiento de señales de voz esta velocidad no necesariamente debe ser alta para mejorar el proceso de la comunicación, por el contrario una alta tasa de transmisión genera problemas como la interferencia, el tráfico de datos en la red, o la misma pérdida de información durante la transmisión recepción, por lo que en la actualidad se usa compresores de voz que disminuyan la tasa de transferencia para evitar pérdida de información.

Esta evolución en el aspecto de la compresión es debido a que la voz cuenta con características muy importantes como son:

- Distribución de probabilidad de amplitud no-uniforme.
- Correlación no-cero entre muestras sucesivas.
- Existencia de ráfagas de muestras (períodos de silencio).
- Cuasi-periodicidad de la señal vocal.

Tipos de Codificadores de Voz

La compresión de la voz se la realiza por medio de algoritmos matemáticos, los cuales digitalizan la señal analógica proveniente de una fuente determinada, como la voz humana por ejemplo, consiguiendo una señal de audio con mejores cualidades y favorables al procesamiento y adecuación para la transmisión de los códigos a través del canal de comunicación elegido para tal efecto.

Los tipos de algoritmos de codificación vocal son:

- Los codificadores de forma de onda en el dominio del tiempo: PCM (Codificación por modulación de pulso), ADPCM (PCM diferencial Adaptativo), Delta, etc.
- Codificadores de forma de onda en el dominio de la frecuencia: ATC (*Adaptative Transform Coding*).

- Los codificadores *Vocoders* (LPC) y los híbridos (VSELP, CELP).

En un codificador PCM la forma de onda no es considerada en el momento de codificar la información, por lo que no es muy eficiente su uso para señales de voz, sin embargo es posible usarlo ya que se observa que presenta una alta correlación con muestras adyacentes, lo que permite reducir la tasa de error.

Un codificador PCM puede ser DPCM (PCM Diferencial) o ADPCM (PCM Adaptativo Diferencial), pero para el tratamiento con señales de voz es mejor utilizar un ADPCM, ya que en DPCM no varían en el tiempo ni el predictor ni el cuantificador, en tanto que en la codificación adaptativa se obtiene mayor eficiencia cuando el cuantificador se adapta a los cambios del residuo de predicción, también se puede conseguir que la predicción se adapte a la señal de voz, en la Fig. 2.26 se muestra un esquema del funcionamiento de un codificador ADPCM.

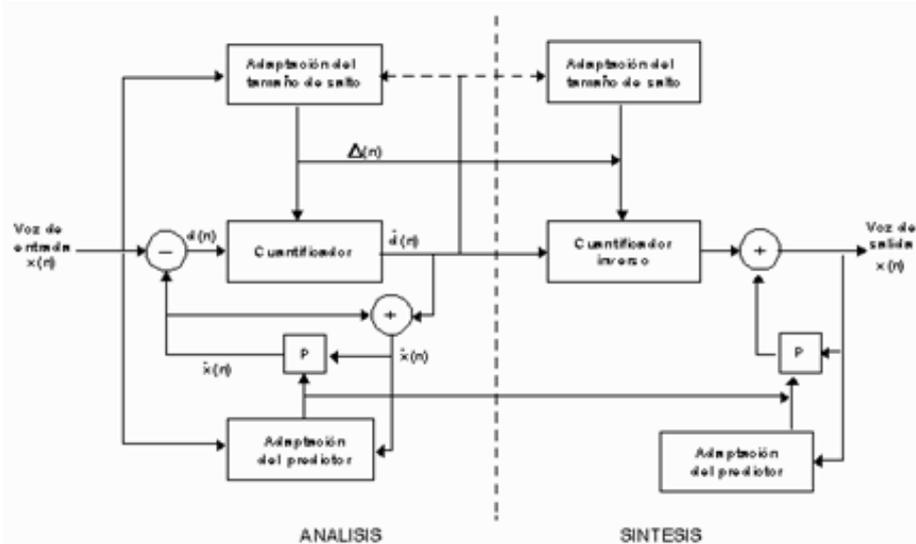


Figura 2. 26. Esquema del Codificador - Decodificador ADPCM.

Es debido a esta capacidad de adaptación que la *ITU* recomienda utilizar el estándar de codificación *G.721* que trabaja a 32 Kbps, perteneciente al grupo de codificadores ADPCM de manera específica para comunicación telefónica, debido a que usa un esquema de adaptación *feedbackward* tanto

para el cuantificador como para el predictor, lo que permite obtener un mejor rendimiento.

En la Fig. 2.27 se muestra el esquema del proceso de muestreo que sufre una señal codificada con ADPCM.

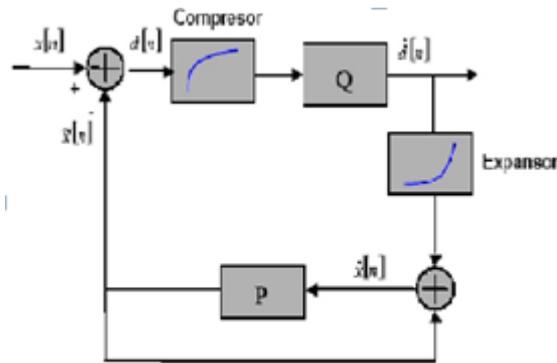


Figura 2. 27. Muestra del Codificador ADPCM.

Algo importante de la codificación ADPCM es la utilización de un cuantificador adaptativo que reduce el ruido de cuantificación debido a que el tamaño del escalón se adapta a varianza de la señal, así se detalla en la Fig. 2.28.

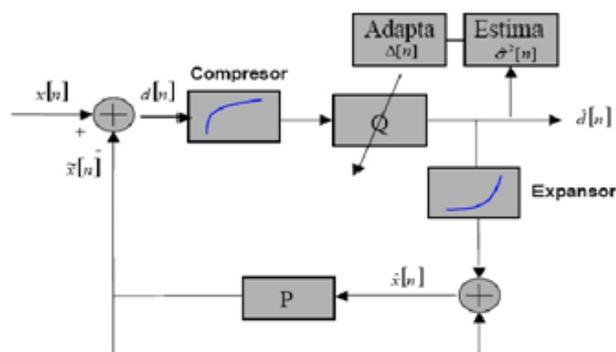


Figura 2. 28. Codificador ADPCM.

En la Tabla 2.6 que se detalla a continuación se hace constar un cuadro comparativo de los diferentes compresores de voz, haciendo referencia especialmente al nivel de compresión, a la Norma que los rige y el tamaño de la trama a ser transmitida.

Tabla 2. 6. Cuadro Comparativo de los Compresores de audio.

Codificación	Velocidad [Kbps]	Norma	Año	Mean Opinion Score (MOS)	Tamaño de Trama [mseg.]
PCM	64	G.711/ 712	1972	4,3	0.125
ADPCM	32	G.721	1984	4,1	0.13
ADPCM	16, 24, 32, 40	G.726	1990		0.13
LD-CELP	16	G.728	1992	4,0	0.63
CS-CELP	8	G.729	1995		10
VCELP	8	IS-54	1993	3,5	30
QCELP	1...8	IS-95	1993	3,4	20
CELP	4, 8	FS-1016	1991	3,2	30
LPC10	2, 4	FS-1015	1984	2,3	22,5

Una forma más eficiente de codificación es la predicción lineal LPC (*Linear Predictive Coders*).

LPAS (*Linear Prediction Analysis-by-Synthesis*). Es útil para señales que pueden modelarse como un sistema lineal. Mientras que PCM-ADPCM-Delta son técnicas de codificación de forma de onda, LPC se basa en la estimación lineal de la fuente.

CODEC VSELP (Vector Sum Excited Linear Predictive). Es una versión de códec desarrollado por *Motorola*. Se utilizan 3 fuentes de excitación al filtro de síntesis *LPC*. Uno de ellos utiliza un código de largo plazo adaptativo (*code-book*). Los restantes son conjuntos de códigos de 128 vectores cada uno. Es usado en el sistema celular dual americano *D-AMPS* (Sistema Dual Americano de Telefonía Móvil) (IS-54) y en *Iridium* a una velocidad de 7,95 kbps.

2.7. Codificación de Canal

2.7.1. Definición

La codificación de canal, llamada también codificación de control de errores, es un proceso que se realiza precisamente para detectar y corregir los errores producidos en el canal [15].

2.7.2. Generalidades

Como se mencionó anteriormente la codificación de canal tiene como objetivo la detección y corrección de errores por lo que para su estudio se la enfoca de dos formas [12]: codificación de la forma de onda, o bien codificación mediante secuencias estructuradas o de redundancia estructurada. La primera tiene por objeto transformar la forma de onda de la señal a fin de que el proceso de detección sea más inmune a los errores de transmisión. La segunda, transforma las secuencias de datos en "secuencias mejores" agregando redundancia, es decir más bits a la información de entrada para que el receptor pueda detectar y corregir errores en la información transmitida. La detección de errores, por sí sola, no es suficiente para proporcionar la protección necesaria y es necesario implementar algún medio para que una vez detectado el error pueda estimarse y reconstruirse la información perdida.

La codificación de forma de onda tiene por objeto transformar un conjunto de pulsos en otro conjunto mejorado, de modo que cada una de las

formas de onda así codificadas sea lo menos parecida posible a cualquier otra del conjunto.

2.7.3. Códigos de Codificación de Canal

▪ Códigos de Bloques lineales Sistemáticos

Son códigos capaces de detectar $X-1$ bits erróneos, donde X viene dado por la distancia de *Hamming* [15] mínima entre 2 palabras cualesquiera del código, y será capaz de corregir $(X-1)/2$ bits erróneos, por lo que su capacidad de corrección es limitada máximo 1 ó 2 símbolos erróneos por palabra código, por lo que se usan en canales con baja probabilidad de error.

▪ Código Hamming

Es un código de bloque lineal, que se representa mediante una combinación $p(n, k)$, donde n y k deberán cumplir las siguientes condiciones:

1. n es la longitud de la palabra código
2. k es el número de bits de datos de la palabra original sin codificar.
3. El número de bits de paridad es $m=n-k$, además deberá cumplir que:

$$n = 2^m - 1 \text{ con } m \geq 3$$

$$k = 2^m - m - 1$$

Como se observa a cada palabra codificada se le agregan bits de paridad, que son bits que permiten al receptor reconocer la veracidad de la palabra código recibido y así poder detectar errores.

Cada bit de paridad añadido a una palabra original va a afectar a unas determinadas posiciones de la nueva palabra de código, de forma que tomarán un valor adecuado para que se cumpla el criterio de paridad (par o impar) preestablecido en las sub combinaciones afectadas por cada uno de estos bits de paridad [19].

▪ Códigos Cíclicos

Estos códigos pertenecen a los códigos de bloque lineales y deberán cumplir las siguientes condicionantes:

- **Linealidad:** la suma módulo-2 de dos palabras del código es otra palabra del código [15].
- **Cíclicos:** cualquier desplazamiento cíclico de una palabra del código también pertenece al código [19].

El código se denota con el par (n, k) , donde n es la longitud de las palabras de código y k es la longitud de una palabra original [15].

Para el manejo de estos códigos se utiliza una notación polinómica, de forma que una palabra de código $C = (c_0, \dots, c_{n-1})$, de la forma indicada en la Ec. 2.1 se interpreta como un polinomio, y cada uno de los bits de la palabra de código será uno de los coeficientes de este polinomio:

$$C(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1} \quad \text{Ecuación 2. 1}$$

A su vez, una palabra original $m = (m_0, \dots, m_{k-1})$ como lo muestra la Ec. 2.2 se interpreta como el polinomio:

$$m(x) = m_0 + m_1x + \dots + m_{k-1}x^{k-1} \quad \text{Ecuación 2. 2}$$

Para generar $C(x)$ a partir de $m(x)$ se usa el polinomio generador $g(x)$ que es un factor de x^n+1 . Su grado es $n-k$. La obtención de la palabra codificada se hace aplicando la Ec. 2.3:

$$C(x) = m(x) \cdot g(x) \quad \text{Ecuación 2. 3}$$

Para realizar el control de errores se utiliza el polinomio de chequeo de paridad mostrado en la Ec. 2.4, que es un polinomio de grado k tal que:

$$g(x) \cdot H(x) = xn + 1 \quad \text{Ecuación 2. 4}$$

▪ Código de Redundancia Cíclica

Son un tipo de códigos cíclicos especialmente, de gran valor para la detección de errores:

- Se diseñan para detectar muchas combinaciones de errores.
- La implementación práctica es sencilla. Son los que se usan en la práctica.

Existen distintos polinomios generadores en base a pruebas que se muestran en la Tabla 2.7 [15]:

Tabla 2. 7. Tabla de Polinomios generadores.

	$g(x)$	$n - k$
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$	12
CRC-16	$x^{16} + x^{15} + x^2 + 1$	16
CRC-ITU	$x^{16} + x^{12} + x^5 + 1$	16

- Los tres contienen $H(x)$ como factor primo.
- CRC-12 se usa con palabras de datos de 6 bits.
- CRC-16 y CRC-ITU con palabras de datos de 8 bits.

▪ Códigos Convolutionales

Estos códigos se diferencian puntualmente de los códigos de bloques por su forma estructural y su capacidad de corregir errores, ya que los códigos Convolutionales son adecuados para trabajar con canales que tiene alta probabilidad de error y presencia de ruido.

Los códigos Convolutionales con códigos lineales, y su sistema tiene memoria debido a que la palabra actual depende de los datos enviados en el pasado y de los que se enviaron ahora [19].

La representación de un código convolutional no es par como los casos anteriores, si no que consta de 3 parámetros n , k , m dónde:

- n es el número de bits de la palabra codificada.
- k es el número de bits de la palabra de datos.
- m es la memoria del código o longitud restringida.

Para un caso específico se obtendrán las variables n , k , m de la siguiente forma:

Código (2, 1, 3)

- La palabra codificada tiene 2 bits de longitud.
- La entrada son bloques de 1 bit.
- La salida depende de los dos bloques anteriores y del actual.

Código (4, 2, 3)

- La palabra codificada tiene 4 bits de longitud.
- La entrada son bloques de 2 bit.
- La salida depende de los dos bloques anteriores y del actual.

Proceso de Codificación Convolutiva

Un codificador convolutivo genera bits redundantes utilizando convoluciones en módulo 2 de ahí el origen que da lugar a su nombre [15].

El codificador convolutivo binario con tasa $1/n$ medida en bits por símbolo puede considerarse como una máquina de estado finito, con un registro de corrimiento de M etapas, con conexiones preestablecidas a n sumadores módulo 2 y a un multiplexor que pone en serie las salidas de los sumadores.

Una secuencia de mensaje de L bits produce una secuencia de salida codificada de longitud $n(L+M)$, por lo que la tasa de código se calcula con la Ec. 2.5 así:

$$r = \frac{L}{n(L+M)} \text{ bits/símbolo}; \quad \text{Ecuación 2. 5}$$

pero debido a que L siempre será mucho mayor que M , la tasa de bits se simplifica a la Ec. 2.6:

$$r = \frac{1}{n} \text{ bits/símbolo} \quad \text{Ecuación 2. 6}$$

Si un codificador tiene un registro de corrimiento de M etapas, implica que la memoria del codificador es igual a M bits de mensaje y se requiere $K= M+1$ corrimientos para que un bit entre al sistema de corrimientos y finalmente salga, lo que hace la longitud de restricción del codificador sea K , esto se puede comprender de mejor manera observando la Fig. 2.29 en la que el codificador tiene $n=2$ y $K=3$ y la secuencia de entrada es de un solo bit a la vez.

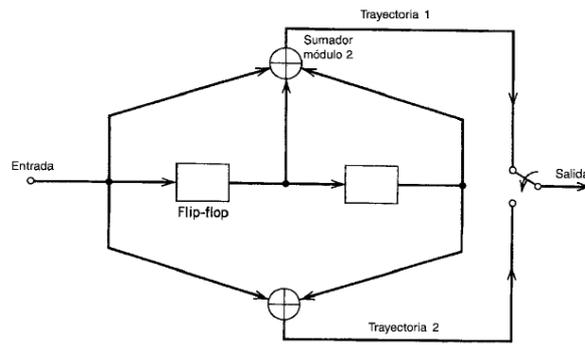


Figura 2. 29. Proceso de Codificación Convolucional.

La situación se complica un poco más si la secuencia de entrada es diferente a un bit a la vez y se cuenta con un multiplexor, la Fig. 2.30 muestra este caso, bajo las mismas consideraciones de $n=2$ y $K= 3$, debido a que se procesan 2 bits a la vez la tasa de código es de $2/3$.

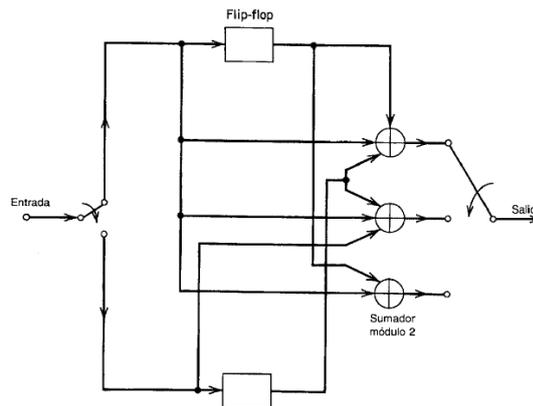


Figura 2. 30. Codificación Convolucional con 2 bits de entrada a la vez.

Este código se puede expresar como una respuesta al impulso, partiendo del caso que los *Flip Flops* están en estado cero; esto provoca la existencia de un polinomio generador mostrado en la Ec. 2.7, que se define como la transformada de retorno unitario de la respuesta al impulso, finalmente el polinomio generador tendrá la forma:

$$g^i(D) = g_0^i + g_1^i D + g_2^i D^2 + \dots + g_M^i D^M \quad \text{Ecuación 2. 7}$$

Si se considera que en la Fig. 2.29, la salida como respuesta al impulso de la trayectoria 1 es (1, 1,1) implica que el polinomio generador es:

$$g^1(D) = 1 + D + D^2$$

De igual forma si la respuesta al impulso de la trayectoria 2 es (1, 0,1), el polinomio generador correspondiente es:

$$g^2(D) = 1 + D^2$$

Si consideramos una secuencia de mensaje de entrada (1, 0, 0, 1,1) el polinomio generador aplicando la Ec. 2.2 sería:

$$m(D) = 1 + D^3 + D^4$$

Se considera que para la transformada de Fourier, la Convolución en el dominio del tiempo se transforma en una multiplicación en dominio de D, entonces el polinomio de salida de la trayectoria uno sería:

$$c^1(D) = g^1(D) * m(D)$$

$$c^1(D) = (1 + D + D^2)(1 + D^3 + D^4)$$

$$c^1(D) = 1 + D + D^2 + D^3 + D^6$$

De esto se puede afirmar que la secuencia de salida es (1111001), bajo las mismas consideraciones la secuencia de salida de la trayectoria 2 será (1011111) y al multiplexar las dos trayectorias en la salida tendremos:

$$c = (11, 10, 11, 11, 01, 01, 11).$$

Es importante tener en cuenta que la secuencia de L bits de mensaje produce una secuencia de salida de longitud n (L+k-1)= 14 bits, de igual forma para que la secuencia de corrimiento vuelva a su estado cero se debe agregar una secuencia de terminación K-1=2 ceros al

último del bit de entrada del mensaje, además se debe considerar que la secuencia de terminación $K-1$ ceros recibe el nombre de cola del mensaje.

El proceso se efectúa en base a las variables que representan el código así como muestra la figura siguiente

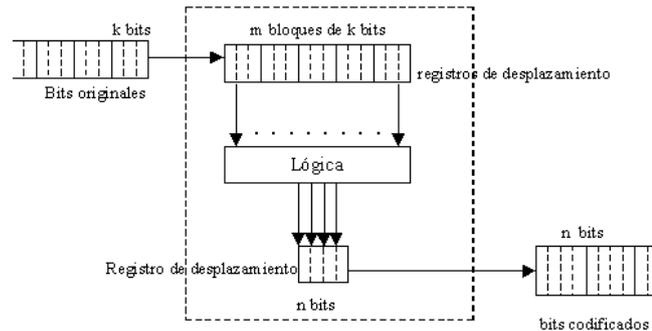


Figura 2. 31. Diagrama de bloques de Código Convolucional 4, 3, 5.

La palabra codificada se obtendría como el resultado de realizar una serie de operaciones lógicas entre determinados bits que están almacenados en los registros intermedios.

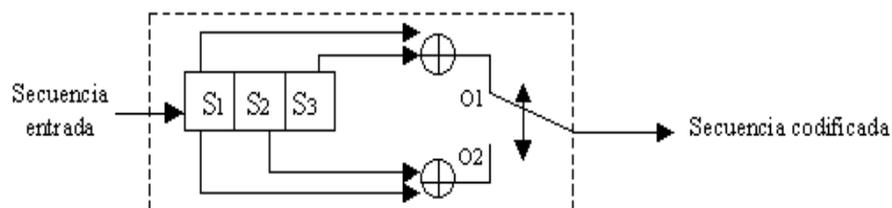


Figura 2. 32. Ejemplo de un Codificador convolucional (2, 1, 3).

- El conmutador con las dos entradas hace el papel de un registro de desplazamiento de dos estados.
- El código convolucional es generado introduciendo un bit de datos y dando una revolución completa al conmutador.
- Inicialmente se supone que los registros intermedios contienen ceros.

Tabla 2. 8. Tabla de Estados de la Codificación Convolutional.

Entradas	Salidas
S1, S2, S3	01, 02
0 0 0	0 0
0 0 1	1 1
0 1 0	0 1
0 1 1	1 0
1 0 0	1 0
1 0 1	0 1
1 1 0	1 1
1 1 1	0 0

En este ejemplo la palabra codificada se obtiene como resultado de sumas módulo-2 entre los bits indicados que están almacenados en los registros intermedios, el sistema actúa como una máquina secuencial.

Existen tres métodos gráficos para representar un codificador convolutional que son:

- **Diagrama árbol o árbol del código:** representación mediante un árbol binario de las distintas posibilidades.
- **Diagrama de estados:** es la forma menos utilizada.
- **Diagrama de *Trellisa* o enrejado:** es la forma más utilizada porque es la que permite realizarla decodificación de la forma más sencilla.

A continuación, en la Fig. 2.33, se observa un modelo del proceso de codificación convolutional:

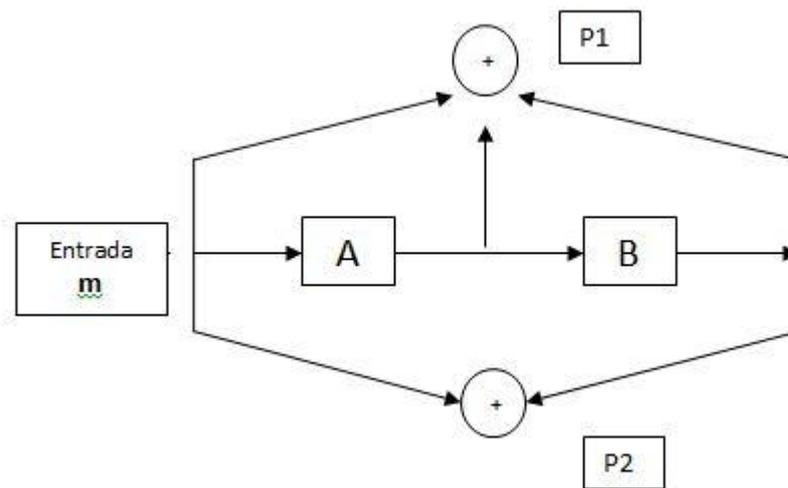


Figura 2. 33. Diagrama codificación Convolutional.

Donde P1 y P2 son bits de paridad.

La tasa código aplicando la Ec. 2.8 es:

$$r = \frac{\# \text{entradas}}{\# \text{salidas}} = \frac{1}{2} \quad \text{Ecuación 2. 8}$$

Igualmente se debe aclarar que A y B son las memorias, y además:

$$K=M+1= 2+1=3$$

▪ Diagrama de Estados

Para este punto, se deberá tomar en consideración lo siguiente, y como consecuencia de aquello, se obtiene los resultados tal como se indica en la Tabla 2.9:

$$A = m$$

$$B = A$$

$$P1 = m + A + B$$

$$P2 = m + B$$

Tabla 2. 9. Tabla de Entradas - Salidas del código Convolutional.

Entradas	A` B`	Salidas
A B m		P1 P2
0 0 0	0 0	0 0
0 0 1	1 0	1 1
0 1 0	0 0	1 1
0 1 1	1 0	0 0
1 0 0	0 1	1 0
1 0 1	1 1	0 1
1 1 0	0 1	0 1
1 1 1	1 1	1 0

La máquina secuencial funcionaría de la siguiente manera:

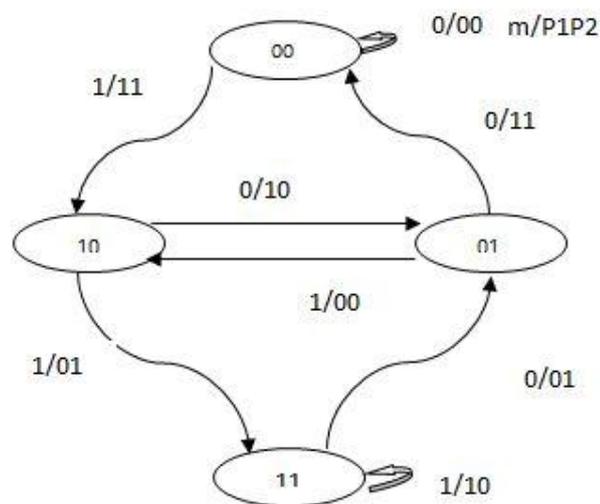


Figura 2. 34. Máquina Secuencial del código Convolutional.

En la Fig. 2.34, se puede apreciar el proceso realizado, usando la *Trellisa*, para el código convolutional. Asimismo, en la Fig. 2.35 se coteja lo expuesto en la máquina secuencial.

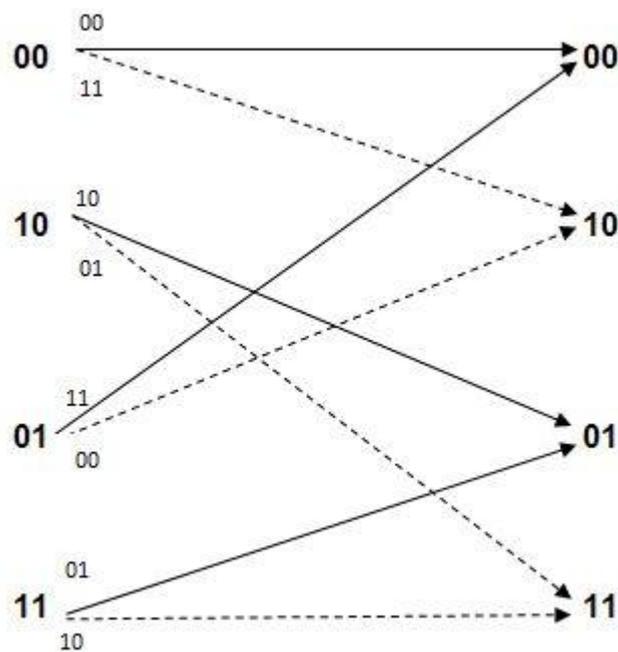


Figura 2. 35. Trellisa.

▪ Proceso de decodificación

Para que el receptor pueda decodificar la información deberá encontrar el camino correcto a través de la *Trellisa* o en el árbol del código o en el diagrama de estados, según sea el caso, a fin de encontrar la secuencia de bits más probable, si la secuencia no presenta errores implica que la palabra código recibida es exactamente la misma enviada.

El codificador convolucional añade una estructura a la secuencia de bits, incluso aunque la entrada sea totalmente aleatoria, se obliga a que la salida siga unas determinadas secuencias. Esta restricción es la que da la capacidad correctora a los códigos convolucionales.

De forma más específica la decodificación significa comparar la secuencia recibida con todas las posibles secuencias que pueden obtenerse con el correspondiente codificador y seleccionando la secuencia que está más próxima a la secuencia recibida.

Para realizar la decodificación se utiliza un algoritmo denominado Algoritmo de *Viterbi* o decodificadores con probabilidad a posteriori o APP. El fundamento, del algoritmo mencionado, está en que no se almacenan todas las secuencias a las que da lugar el codificador, *Viterbi* fundamenta la optimización, es decir, el mejor camino, el de menor distancia de Hamming a través del diagrama de *Trellisa* que pasa por un determinado nodo, necesariamente incluye el mejor camino desde el principio del diagrama de *Trellisa* hasta este nodo.

El principio anterior implica que para cada uno de los nodos de la *Trellisa* sólo es necesario guardar el mejor camino (secuencia) hasta ese nodo. De esta forma, como mucho se tendrán tantos caminos como estados diferentes (el número de estados es $2^{(m-1)*k}$).

▪ **Descripción del algoritmo de *Viterbi***

- **Paso 1:** en el nivel j (nivel determinado), calcular la distancia de Hamming de cada camino entrante en cada nodo (estado) desde el nodo del nivel $j-1$ hasta el nodo del nivel j a través del camino superviviente.
- **Paso 2:** para cada nodo (estado) del diagrama de *Trellisa* en el nivel j , descartar todos los caminos que entran en el nodo, excepto el de distancia mínima. Cuando a un nodo llegan dos caminos con la misma distancia se toma el superior, aunque no siempre se obtiene el resultado deseado, sin embargo de darse un error se deberá retomar el proceso desde este punto y tomar el otro nivel que tenía la misma distancia de hamming y avanzar por este camino.
- **Paso 3:** pasar al nivel $j+1$ y repetir los pasos 1 y 2.

Estos pasos se aplican para j mayor o igual que 2. Hasta ese valor se expanden los caminos.

La capacidad de corrección de errores de un código convolucional está en función de la distancia libre o distancia mínima de hamming, por lo que se puede corregir t errores únicamente si la distancia libre es mayor que $2t$, para esto se considera la siguiente relación, donde t es la capacidad de corrección de errores y se calcula aplicando la Ec. 2.9

$$t \leq \frac{d_{min}-1}{2} \quad \text{Ecuación 2.9}$$

2.8. Técnicas de Modulación y Demodulación

2.8.1. Definición

▪ Modulación

La modulación, es la acción y efecto de modular una señal. Esto quiere decir que al tener señal que ocupe un determinado *ancho de banda*⁷, es necesario que ocupe el menor espacio posible del espectro electromagnético a fin de optimizar el mismo. Modular quiere decir modificar, lo que significa que se modificarán las características esenciales de la señal original; dentro de éstas características asociadas a una señal, están su amplitud, su frecuencia o también su fase [12].

La modulación entiende básicamente dos tipos de señales, la señal moduladora que representa al mensaje original y la portadora que es la señal a transmitir por cualquier canal.

Un modulador, se encarga de alterar la onda portadora para que ésta se altere en función de las variaciones de la onda moduladora. Dicho sea de paso, preferentemente la señal portadora es de frecuencia más alta que la frecuencia más alta de la señal moduladora. La justificación de esto es porque, es necesario optimizar el espectro, ya que al aumentar la frecuencia, a cambio de un menor alcance, en

⁷ Ancho de Banda.- Conjunto de frecuencias usadas para la transmisión de información.

términos de distancia de cobertura, es necesario para el diseño de dispositivos de un tamaño no desmesurado, en especial lo que tiene que ver con las antenas.

- **Demodulación**

La demodulación nació como complemento del proceso de modulación. Como es evidente, se requiere que la modulación sea un proceso reversible, es decir que se pueda volver al mensaje original, de modo que el mensaje se recupere por medio de un proceso complementario, en este caso llamado demodulación.

Al igual que el modulador, el demodulador también transforma una señal, en este caso de una señal a una determinada frecuencia, a otra frecuencia, que será la frecuencia original del mensaje. Es decir, también realiza una traslación de frecuencias, las cuales se deben enfocar en base a la distribución de rangos de frecuencia detallados en la Fig. 2.36.

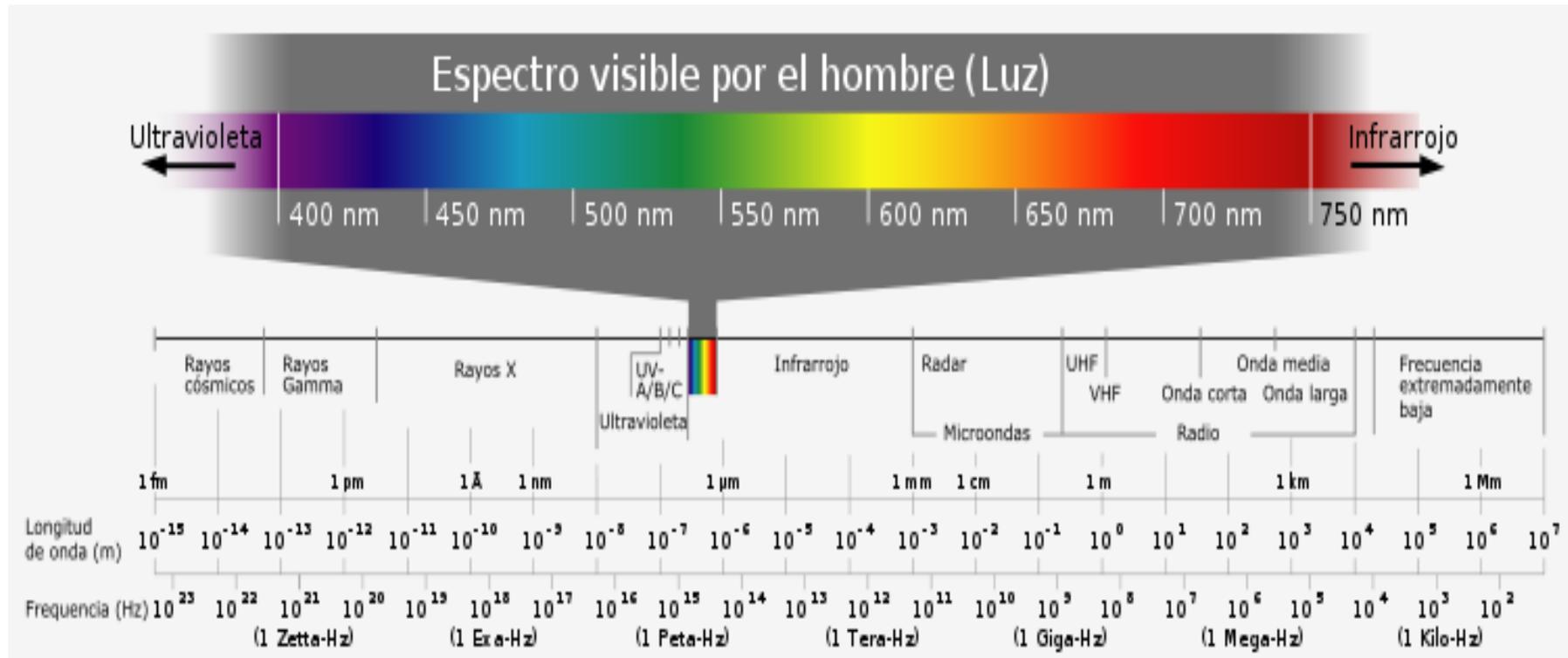


Figura 2. 36. Espectro Electromagnético.

2.8.2.Importancia de la Modulación y Demodulación.

Como ya se mencionó, el proceso de la modulación y sus efectos son significativos. A continuación se citan varias razones por las cuales es importante tanto la modulación como su complemento, la demodulación [12].

- i. **Eficiencia en la transmisión.-** La transmisión de señales implican una onda que viaja por un medio, sea guiado o no, y la eficiencia depende de la frecuencia a la que la señal es transmitida, por lo tanto, al valerse de la propiedad de traslación de frecuencias, el mensaje puede acoplarse a una onda portadora cuya frecuencia permite una propagación eficiente.
- ii. **Supera limitaciones en los equipos.-** Debido a la restricción que implica el hardware, debido a su costo y disponibilidad, y al rendimiento que presenta, esto en función de las frecuencias de trabajo, la modulación permite diseñar sistemas de comunicaciones lo suficientemente robustos para un intervalo de frecuencias. En el diseño, es inherente un tema que preocupa en cierta medida al diseñador, y este es el ancho de banda fraccionario, que se define como el ancho de banda absoluto dividido entre la frecuencia central [6]. Los costos y las complicaciones en cuanto al hardware, son mínimas si éste ancho de banda fraccionario se mantiene en el rango de 1% a 10%.
- iii. **Reducir el ruido y las interferencias.-** El ruido es algo que afecta a las señales y las modifica, las corrompe. Esto sucede debido a que las señales se atenúan con la distancia, es decir que a mayor distancia, la deformación de la señal será mayor y en consecuencia se confundirá con el ruido que puede tener diferentes fuentes de origen como la red eléctrica, ruido térmico, industrial, entre otros.

Afortunadamente, existen varios tipos de modulación, que tienen la propiedad de suprimir el ruido y la interferencia. Esta propiedad se denomina

reducción *del ruido de banda ancha*, que al igual que el ancho de banda fraccionario, se deriva de la ley de *Harley-Shannon*. Esta manifiesta que se requiere un ancho de banda mayor que el ancho de banda utilizado por la señal moduladora, o señal original.

- iv. **Multiplexación.-** La multiplexación es un proceso que se ayuda del proceso de modulación que mezcla varias señales para que toda la información pueda ser transmitida en un solo canal y al mismo instante. En el destino, un banco de filtros se encarga de separar las señales hacia los diversos destinos.

2.8.3. Tipos de Modulaciones

Con todos los antecedentes que se han mencionado, es importante destacar que la modulación, de manera muy general, tiene dos tipos, el primero de estos tiene que ver con las señales de tipo *analógicas*, que son señales que varían levemente en amplitud. Como un ejemplo de esta clase de señales se tiene a la voz humana. El otro camino, lo constituyen las señales del tipo *digital*, que como ya se mencionó anteriormente, lo conforman señales discretas en el tiempo, es decir que poseen un número finito de símbolos usados para representar la información, aquí se puede citar la comunicación entre computadores [12].

A continuación se presenta, en las Fig. 2.37 y Fig. 2.38, una clasificación generalizada de las diversas clases de modulaciones que existen:

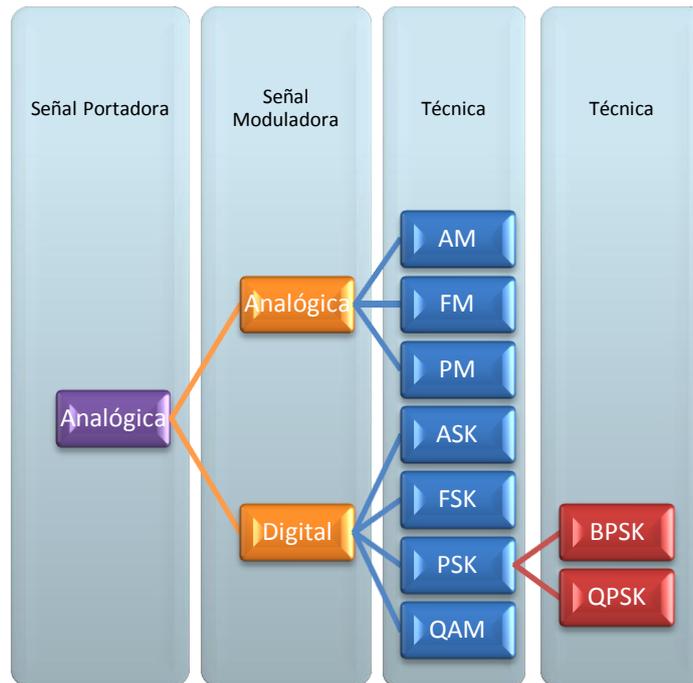


Figura 2. 37. Clasificación de modulación con portadora analógica.

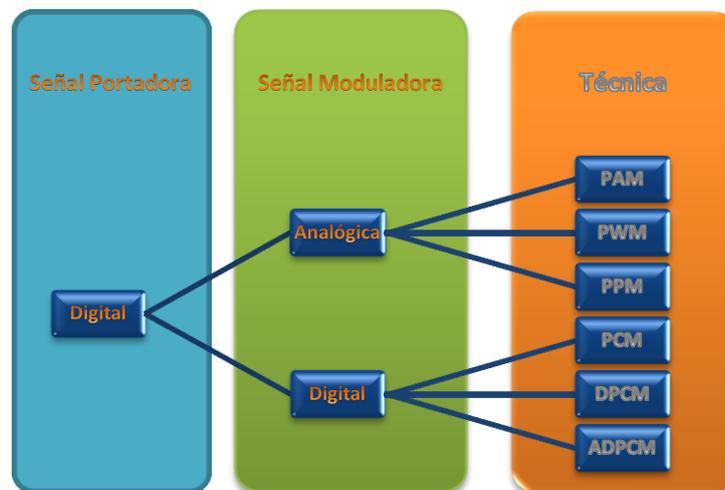


Figura 2. 38. Clasificación de modulación con Portadora Digital.

El estudio en el cual se enfoca la presente tesis, está direccionada a la sección que comprende portadora analógica con moduladora digital. Las razones del estudio más detallado en este vasto campo se demuestran por el hecho de que la información que es procesada por la tarjeta *USRP*, es esencialmente digital. Sin embargo, ésta no es la única razón. La justificación

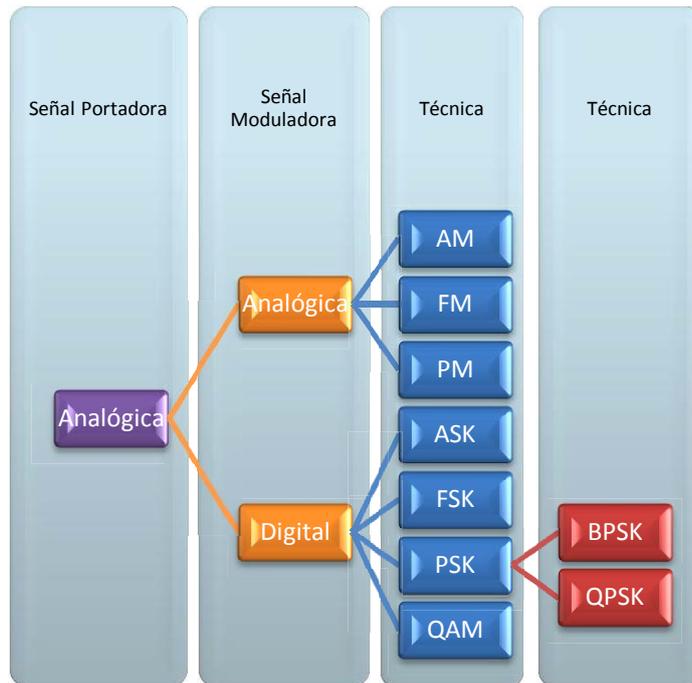


Figura 2. 37. Clasificación de modulación con portadora analógica.

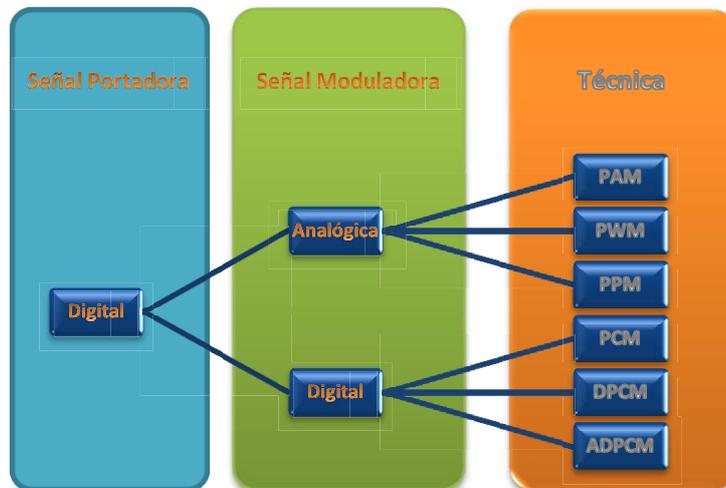


Figura 2. 38. Clasificación de modulación con Portadora Digital.

El estudio en el cual se enfoca la presente tesis, está direccionada a la sección que comprende portadora analógica con moduladora digital. Las razones del estudio más detallado en este vasto campo se demuestran por el hecho de que la información que es procesada por la tarjeta *USRP*, es esencialmente digital. Sin embargo, ésta no es la única razón. La justificación

de mayor peso, es aquella referente a la codificación de la información, y como es muy conocido en el ámbito de los sistemas de comunicaciones, es más fácil codificar digitalmente que análogamente. En consecuencia, la información a ser receptada en éste bloque, del sistema de comunicaciones será puramente digital.

Pero aún falta la explicación del por qué se va a usar una señal portadora analógica. Para responder a esta pregunta, es importante recordar algo que se mencionó en el inciso tercero de la sección 2.8.2. Allí se exhibió el problema que proviene de la comunicación a larga distancia, que se resume como una modificación, en amplitud especialmente, de la señal, razón por la cual un sistema con portadora digital podrían no ser muy viable en el desarrollo del presente proyecto. Si bien éste pretende demostrar que un sistema de comunicaciones puede implementarse sin inconvenientes sobre un *SDR*, también pretende cimentar las bases para el desarrollo de un prototipo robusto y a la vez confiable, que pueda cubrir ciertas distancias. Además de esto, en el rango de frecuencias que se ha propuesto trabajar, y por los equipos que se dispone, se justifica el uso de una portadora análoga.

▪ **Modulación en Frecuencia FM**

La modulación consiste en modificar una de las características de la señal portadora, pudiendo ser ésta la amplitud, la fase o la frecuencia; esta última consiste en modificar la frecuencia de la portadora en función de la señal moduladora, manteniendo su fase y amplitud de forma constante, esta modulación es considerada parte de la modulación angular, para lo cual se debe considerar la Ec. 2.10 como ecuación característica:

$$\phi(t) = A * \cos\{w_c * t + \int_0^t k_f * f(t)dt\} \quad \text{Ecuación 2. 10}$$

Dónde A es la amplitud de la señal portadora, w_c es la frecuencia angular de la portadora, k_f es una constante y $f(t)$ es la señal moduladora.

La demodulación por su parte básica, recuperar la información de la señal moduladora, la misma que se altera por los cambios en fase de la señal modulada, detectar dicho cambio de fase es la razón fundamental de la demodulación.

- **Métodos Utilizados para moduladores FM**

- **Método Indirecto**

Se generan inicialmente señales FM [9] de banda angosta, luego se las transforma en señales de banda ancha, mediante el uso de multiplicadores de frecuencia.

- **Método Directo**

En éste método la frecuencia de la señal portadora es controlado por la señal moduladora, para esto se requiere disponer de un oscilador sintonizado, que oscila en función de una inductancia y una capacitancia que permiten modular la señal en frecuencia, siempre y cuando la inductancia o la capacitancia varíen en función de la señal de entrada, además también se puede generar estas señales por medio de un oscilador controlado por voltaje cuya frecuencia de oscilación dependerá del voltaje instantáneo de la señal de entrada

- **Métodos de Demoduladores FM**

- **Método Directo**

Este método se fundamenta en un discriminador en frecuencia, que obtiene un voltaje de salida proporcional a la frecuencia de la señal de entrada, un esquema de ésta demodulación se muestra en la Fig. 2.39, que muestra la existencia de un limitador, un diferenciador y un detector de envolvente [9].

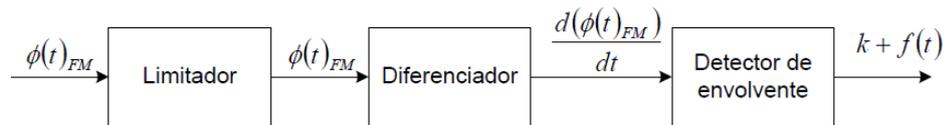


Figura 2. 39. Método Directo de modulación FM.

Este método utiliza un limitador para garantizar que la señal de entrada sea constante, mientras que el diferenciador permite a través de la derivada de la fase permite encontrar la frecuencia instantánea y tratar a la señal como una señal de amplitud modulada, finalmente con el detector de envolvente se extrae la moduladora de la señal equivalente de amplitud modulada.

▪ **Método Indirecto**

Este método como lo muestra la Fig. 2.40, se compone de cuatro fases; un detector de fase, un filtro de lazo, un amplificador y un oscilador controlado por voltaje, que es quien se sintoniza y permite que el filtro de lazo se enganche, esto se da cuando opera a la frecuencia central, fundamentalmente es el oscilador quien al existir un cambio de fase envía la información al detector, el cual encuentra la diferencia de fase entre la modulada y la salida del oscilador, esta diferencia representa una señal de voltaje AC alterna la misma es amplificada y filtrada para ser realimentada al oscilador.

Si esta diferencia de fase es igual a cero el nivel de tensión a la salida es constante y el oscilador se sintoniza correctamente permitiendo que el lazo se enganche, finalmente la tensión de salida es proporcional a la desviación en frecuencia por lo que se convierte en la señal demodulada.

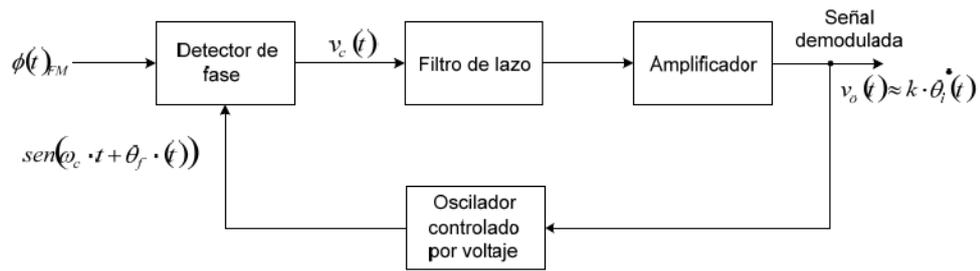


Figura 2. 40. Método Indirecto de demodulación FM.

Para obtener la derivada de la fase, existe el algoritmo de Richard G. Lyon el cual partiendo la siguiente Ec. 2.11:

$$\Delta\phi = \frac{d}{dt} \left[tg^{-1} \frac{q(t)}{i(t)} \right] \quad \text{Ecuación 2. 11}$$

Considera que $r(t) = \frac{q(t)}{i(t)}$, y q(t) la componente real e i(t) la

componente imaginaria de la señal, obtiene la derivada de la función inversa tangente que es:

$$\frac{d}{dx} tg^{-1} = \frac{1}{1 + [f(x)]^2} * Dxf(x)$$

Obtiene:

$$\Delta\phi = \frac{1}{1 + \left[\frac{q(t)}{i(t)} \right]^2} * \frac{i(t) \frac{dq(t)}{d(t)} - q(t) \frac{di(t)}{d(t)}}{i^2(t)}$$

Sustituyendo los valores de tiempo continuo a tiempo discreto, es decir cambiando (t) por (n) y operando se obtiene:

$$\Delta\phi = \frac{i(n) \frac{dq(n)}{d(n)} - q(n) \frac{di(n)}{d(n)}}{i^2(n) + q^2(n)}$$

▪ Modulación PSK

La modulación PSK que ilustra la Fig. 2.41, también conocida como *Modulación por Desplazamiento de Fase* (del inglés, *Phase-Shift Keying*), es un tipo de modulación angular, muy utilizada en varias ramas de los sistemas de comunicaciones [12].

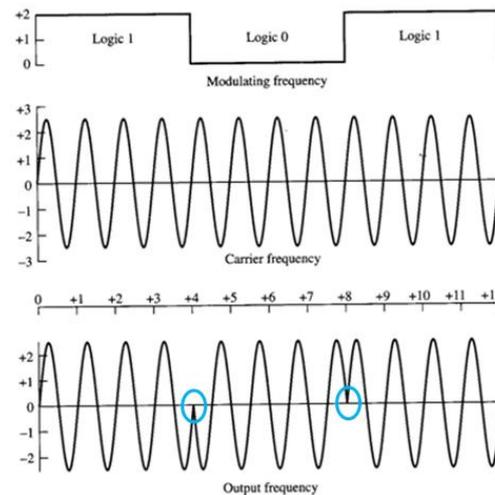


Figura 2 41. Modulación PSK.

Al igual que las anteriores técnicas ya mencionadas, ésta modifica una característica de la onda portadora. La propiedad que ésta técnica modifica es la fase de la portadora. Es posible percibir que la onda en este caso, también es del tipo senoidal. Esto quiere decir que la señal posee tres características muy marcadas que son: su amplitud A , la frecuencia f y por último su fase ϕ . En consecuencia, una onda senoidal está determinada por la Ec. 2.12 que indica:

$$s(t) = A \cdot \sin(\omega t + \phi) \quad \text{Ecuación 2. 12}$$

Tomando en cuenta que:

$$\omega = 2\pi \cdot f \left[\frac{\text{radianes}}{\text{seg.}} \right]$$

Retomando el camino de la onda resultante de la modulación PSK, y recordando la codificación PCM, entonces se tendrá dos valores, el primero del bit “0” y el segundo del bit “1”. Como se acaba de señalar, éste tipo de modulación modifica la fase de la portadora, que dicho sea de paso es del tipo sinusoidal. En conclusión, la fase de la señal resultante, la señal modulada, deberá cambiar de fase entre dos valores, por lo general entre 0 y π radianes, o si se prefiere, entre 0° y 180° [22].

En éste punto, es importante mencionar tres nuevos términos que se debe tener muy en cuenta, el primero es el llamado *símbolo* o *palabra*. Ambos son sinónimos del mismo concepto. Los dos hacen referencia al bit o grupo de bits que son usados para dar forma a la modulación. Dicho de otra manera, si se escogen dos símbolos, un sistema binario, “1” y “0”. Cada uno de ellos es un símbolo, y al conjunto total de símbolos se los denomina *alfabeto*. En éste caso, el alfabeto consta de dos símbolos. A esto a su vez se le conoce como el *orden* (M), que significa el número de símbolos usados. Es importante destacar que el orden, le da el nombre a la modulación, M -ario, así por ejemplo, si $M = 2$, significa que será una modulación binaria (BPSK: *Modulación por Desplazamiento de Fase Binario*). Si $M = 4$, la modulación será cuaternaria (QPSK), y así sucesivamente.

Imagínese ahora, que ya no únicamente se toma un bit para representar un símbolo, sino que se toman dos bits. En consecuencia, es posible que con dos bits por cada símbolo, se tenga un *alfabeto* de 4 palabras. Por lo tanto el orden M es igual a 4.

A continuación se detallan los dos subtipos más conocidos de la modulación PSK.

- **Modulación BPSK**

La técnica BPSK [10], también conocida como *Modulación por Desplazamiento de Fase Binario* (del inglés, *Binary Phase-Shift Keying*),

es una de las técnicas de modulación más conocidas en el ámbito de las comunicaciones, y además una de las más usadas. Como ya se citó, esta técnica, al ser derivada de PSK, tiene el mismo principio, cambiar la fase a fin de obtener una diferencia, para transmitir y codificar la información [21].

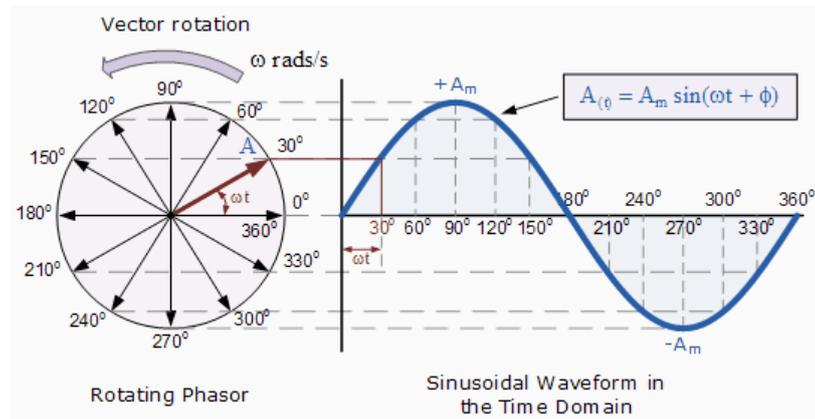


Figura 2.42. Diagrama Fasorial y Onda Senoidal.

Al ser un sistema binario, esto significa que $M = 2$, por lo tanto, tendrá dos fases, 0° y 180° . El receptor es capaz de medir el cambio de fase, tal como se observa en la Fig. 2.42. Allí se destaca en azul, los dos cambios de fase, al pasar de “1” a “0” y viceversa. La pregunta que surge es cómo el receptor analiza éste cambio de fase. Para responder esta pregunta, es necesario recordar que cualquier función senoidal puede ser representada en un plano bidimensional. A este tipo de representación también suele conocerse como *representación fasorial* [11], tal como se observa en la Fig. 2.42. Como se puede observar, la longitud del vector A_m , es la amplitud de la señal, ω es la frecuencia en radianes por segundo, también llamado *frecuencia angular*. Por último se tiene a ϕ que es la fase o desfase. En cuanto al diagrama fasorial, en el campo de la modulación, se lo denomina *diagrama de constelación*, que de igual manera es un plano complejo, el mismo que exterioriza el orden de la modulación y consecuentemente proyecta las posibles fases de la señal, mientras que a los ejes, real e imaginario, se los denomina *en fase (I)* y *en cuadratura (Q)* respectivamente; en el caso de la

modulación BPSK, el diagrama de constelación contiene dos valores, como se observa en la Fig. 2.43, análogamente como si fuesen vectores.

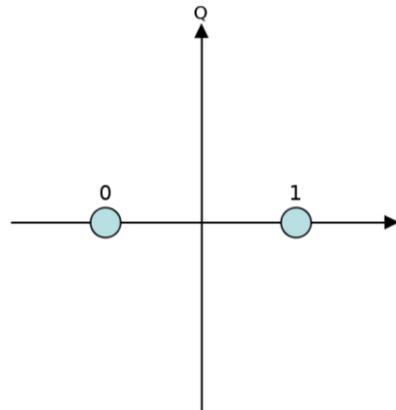


Figura 2. 43. Diagrama de Constelación BPSK.

La observación que se puede realizar en base a lo citado, el ancho de banda de ASK y BPSK son los mismos, debido a que usan una única frecuencia y dos valores, en el primer caso en amplitud, en el segundo dos fases [21].

▪ **Modulación QPSK y M-aria**

La modulación QPSK, que muestra la Fig. 2.44, *Modulación por Desplazamiento de Fase en Cuadratura* (del inglés, *Quadrature Phase-Shift Keying*), al igual que BPSK, es una técnica que usa un determinado número de símbolos para codificar en fase a la señal moduladora, que en este caso es una señal digital. Aquí se usan 2 bits por símbolo, es decir un alfabeto de 4 palabras. Por consiguiente $M = 4$.

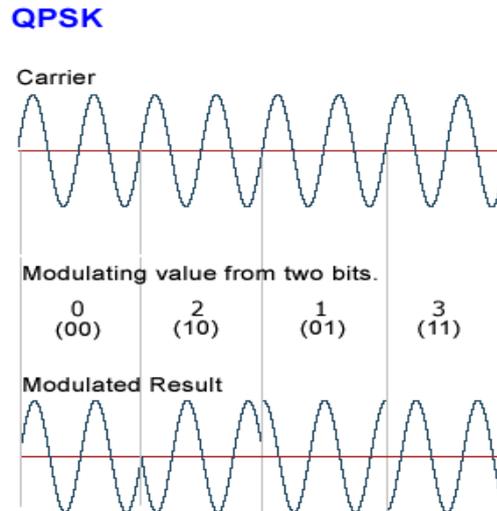


Figura 2. 44. Modulación QPSK.

Por lo general, el diagrama constelación que corresponde a ésta técnica, y a las de orden superior, como 8-PSK, 16-PSK, etc., así se tendrá amplitud constante y el número de fases de salida será ilimitado.

Es muy importante destacar que el ancho de banda ocupado por la modulación QPSK es la mitad de aquel ocupado por BPSK. Esto se debe esencialmente a que al aumentar al doble el número de símbolos ensamblados sobre la portadora modulada con QPSK, la tasa de transmisión se duplica y en consecuencia el sistema será más eficiente [12].

Tanto BPSK como QPSK son técnicas ampliamente utilizadas en los sistemas de Radio Comunicación, por su facilidad y su bajo coste.

CAPÍTULO III

DISEÑO Y DESARROLLO DEL SDR

3.1. Diseño del Sistema de Comunicaciones

3.1.1. Generalidades

Un sistema de comunicaciones cuenta con varios procesos indispensables y fundamentales para garantizar que la información llegue de forma oportuna y eficaz desde el emisor hasta el receptor, los mismos que para el presente diseño han sido analizados y seleccionados de acuerdo a su rendimiento. en las condiciones en las que los equipos de radio van a operar, esto implica, por ejemplo, considerar la banda de frecuencia; esto, a su vez, involucra la existencia de longitudes de onda en el orden de los metros, es decir HF y VHF, con frecuencias bajas que permiten tener una gran área de cobertura; de igual forma los canales con los que trabajan al estar modulados en fase, sufren mucha interferencia e inserción de ruido, debido a que los equipos deberán tener la capacidad de operar en condiciones climáticas adversas, en terrenos cuya orografía será complicada, además de otros factores que pueden sumarse de acuerdo al lugar y condiciones de operación [20].

En base a esto se ha considerado que el equipo de radio estará conformado de la manera que se detalla en el siguiente diagrama.

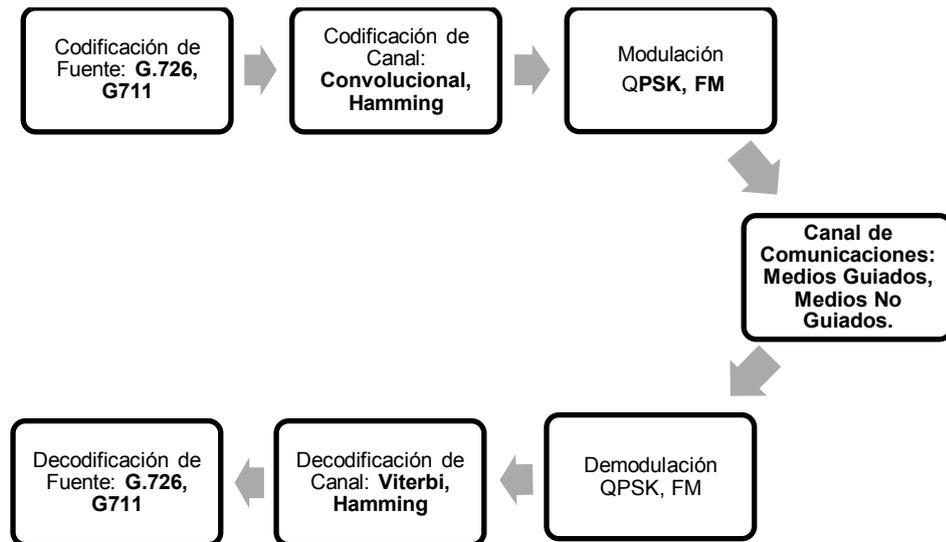


Figura 3.1. Sistema Digital de Comunicaciones.

En la Fig. 3.1, es posible observar un diagrama de un sistema básico de comunicaciones digital. Bajo ésta premisa, y tomando en consideración todo lo mencionado anteriormente, se procederá al diseño de cada uno de los bloques, a excepción del canal, del sistema de comunicaciones [8]. En consecuencia, el sistema constará de las siguientes etapas:

- Codificador – Decodificador de Fuente:
 - Códecs: Recomendaciones *ITU G.711* y *G.726*.
- Codificador – Decodificador de Canal:
 - Convolucional – Viterbi.
 - Hamming.
- Modulador – Demodulador:
 - QPSK.
 - FM.
- Dispositivo de Front – End:

- *USRP*, que será el dispositivo que realiza el enlace para hacer efectivo el proceso de comunicación.

Por lo tanto, se procederá a diseñar e implementar los modelos que serán propuestos, a fin de proveer el aporte comparativo que el presente proyecto pretende lograr.

3.2. Codificación de Fuente

3.2.1. Códec G.711

La Recomendación *ITU - G.711*⁸, es un códec de voz que utiliza 8 bits para codificar cada muestra de la señal vocal. Al ser uno de los primeros códecs desarrollados para optimizar la digitalización y posterior transporte de la voz, ayudó para que se mejoren los algoritmos desarrollados.

Tomando en consideración el teorema de *Nyquist*, que manifiesta que para la digitalización óptima de una señal analógica, como la voz, es necesario realizar el proceso de muestreo con el doble de la frecuencia más alta de dicha señal, inicialmente utilizada. Así y por lo tanto, las frecuencias más altas que normalmente se considera para la voz, es de 3.5 KHz. Siguiendo con ello, y además con el teorema de *Nyquist*, se aproxima que la frecuencia de muestreo seleccionada para una cuantización ideal de la voz, deberá ser de 8KHz. Esto quiere decir que se tomarán 8000 muestras por segundo, o dicho de otra forma, una muestra de la señal cada 125us. Por ello, la tasa de transferencia usada por éste códec es de 64Kbps. Esto se obtiene de multiplicar los 8 bits con los 8KHz [24].

La gran ventaja que éste códec representa es el “ahorro” en cantidad de información por muestra y la calidad de la señal. Además sirve de parámetro para la calificación cualitativa de los códecs que constantemente evolucionan y tienden a usar cada vez menos bits por muestra.

⁸Para más información, dirigirse a: <http://www.itu.int/rec/T-REC-G.711-198811-I>

Éste códec consta de tres etapas muy marcadas:

- Muestreo
- Cuantización
- Codificación

El proceso de muestreo, como ya se mencionó en el CAPÍTULO II, hace referencia al proceso de tomar valores instantáneos, de la señal analógica, en intervalos de tiempo iguales.

Por otro lado, la cuantización es el proceso de asignación de ciertos valores discretos, en función de los valores obtenidos de las muestras. Para éste caso existen dos tipos de cuantizadores, los Uniformes y los No Uniformes. En el primero, se asignan valores que se encuentran a distancias iguales, uno y su inmediato, a cada una de las muestras. En el segundo caso, la asignación de valores está regida por algoritmos complejos o leyes de codificación y compresión. Dentro de los No Uniformes, se encuentran:

- Ley A, usada en Europa y América Latina.
- Ley u, usada en Norteamérica y Japón.

Ambas leyes responden a la necesidad y exigencia de la señal de voz que cambia constante y erráticamente, por lo cual, dichas leyes definen grupos o tramos de valores para los cuales se asignan un dato determinado.

Por último, la sección de codificación, y posterior decodificación, hace referencia a la secuencia de *bits* que van a ser transmitidos hacia el receptor.

3.2.2. Códec G.726

▪ Generalidades

Un codificador *G.726*⁹ es un tipo de códec *ADPCM*, es decir que puede adaptarse al tipo de señal que esté recibiendo. Además tiene la capacidad de comprimir la señal de forma dinámica a 16 Kbps, 24Kbps, 32Kbps o 40 Kbps; en lugar de codificar la señal de forma directa, lo realiza mediante la diferencia que existe entre la señal y una predicción de la misma, hecha a partir de dicha señal, con el objetivo de mantener la calidad del sonido, en consecuencia, la señal no sufre mayores alteraciones.

Es importante señalar que la relación entre las frecuencias vocales y demás leyes de codificación y decodificación se especifican de forma detallada en la recomendación *G.711* de la *ITU-G*, ya que el códec *G.726* es una mejora tanto al códec *G.711* como el *G.721* y el *ADPCM*.

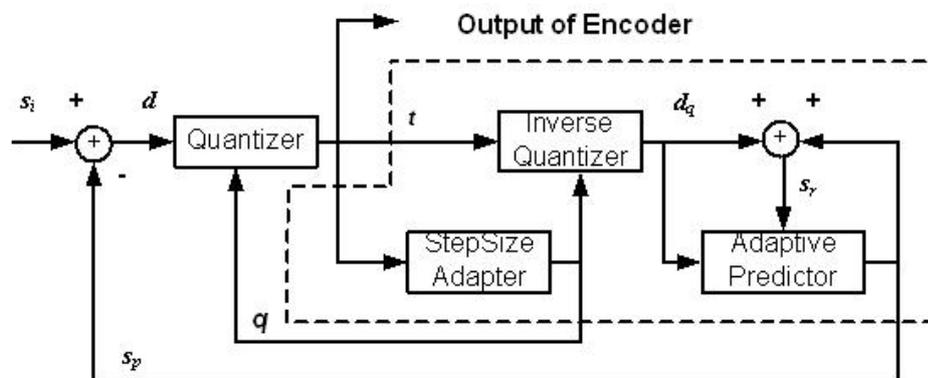


Figura 3. 2. Diagrama general de un codificador ADPCM.

La Fig. 3.2 muestra un diagrama general de un codificador *ADPCM*, que puede definirse como un Cuantizador Adaptativo Diferencial. En los esquemas de este tipo, el tamaño del paso del cuantizador es proporcional a la varianza de la entrada del cuantizador.

⁹ Para mayor información, dirigirse a: <http://www.itu.int/rec/T-REC-G.726-199012-1>

Sin embargo, como la señal diferencia $d(n)$ será proporcional a la entrada, es razonable controlar el tamaño del paso ya sea a partir de $d(n)$, o como se muestra en la Fig. 3.2, a partir de la entrada, $s_i(n)$. Los procesos de adaptación del tamaño del paso pueden proveer mejoras en el SNR de alrededor de 5dB sobre el PCM no adaptativo de ley μ estándar. Esta mejora junto con los 6dB que se pueden obtener de la configuración diferencial con predicción fija indica que ADPCM con predicción adaptativa de alimentación hacia delante proporcionaría un SNR de 10-11 dB mayor que el obtenido con un cuantizador fijo con el mismo número de niveles.

Cada uno de los bloques del codificador, desempeñan un papel fundamental dentro del proceso de adaptarse y predecir la señal, por ésta razón, más adelante se provee una explicación detallada de cada uno de ellos a fin de evidenciar la importancia en la implementación de éste códec dentro del presente proyecto de pregrado.

▪ Implementación del Codificador

El códec del presente diseño está configurado para trabajar únicamente a 32Kbps, el mismo que ha sido implementado en *Simulink* en base a las recomendaciones establecidas por la *ITU* en su recomendación G-REC G.726 publicada en diciembre de 1990. Conjuntamente, se han considerado el anexo publicado en noviembre de 1994, Anexo B publicado en julio del año 2003 y las correcciones finales publicadas en marzo 2005, las mismas que están disponibles en la página de la *ITU*.

Las Fig. 3.3 y Fig. 3.4 corresponden al codificador decodificador de la recomendación *ITU-G.726*.

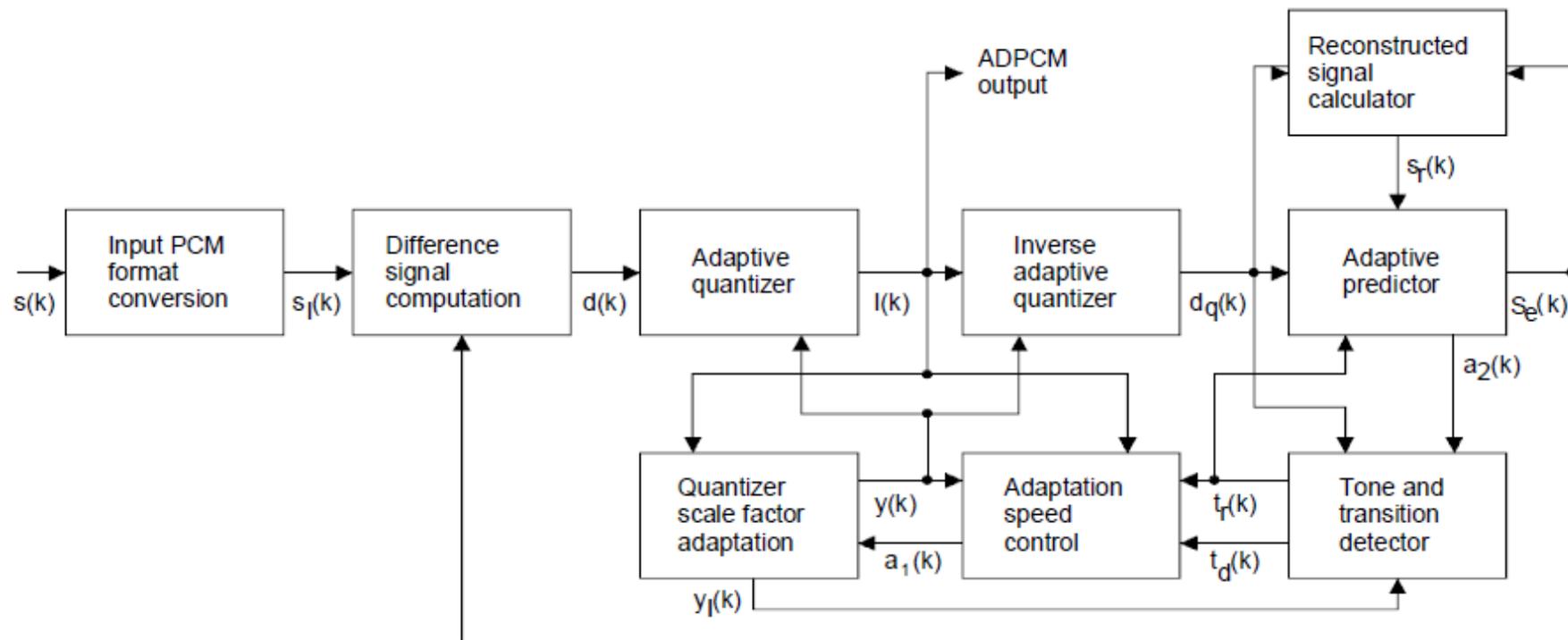


Figura 3 3. Esquema de bloques del Codificador G.726.

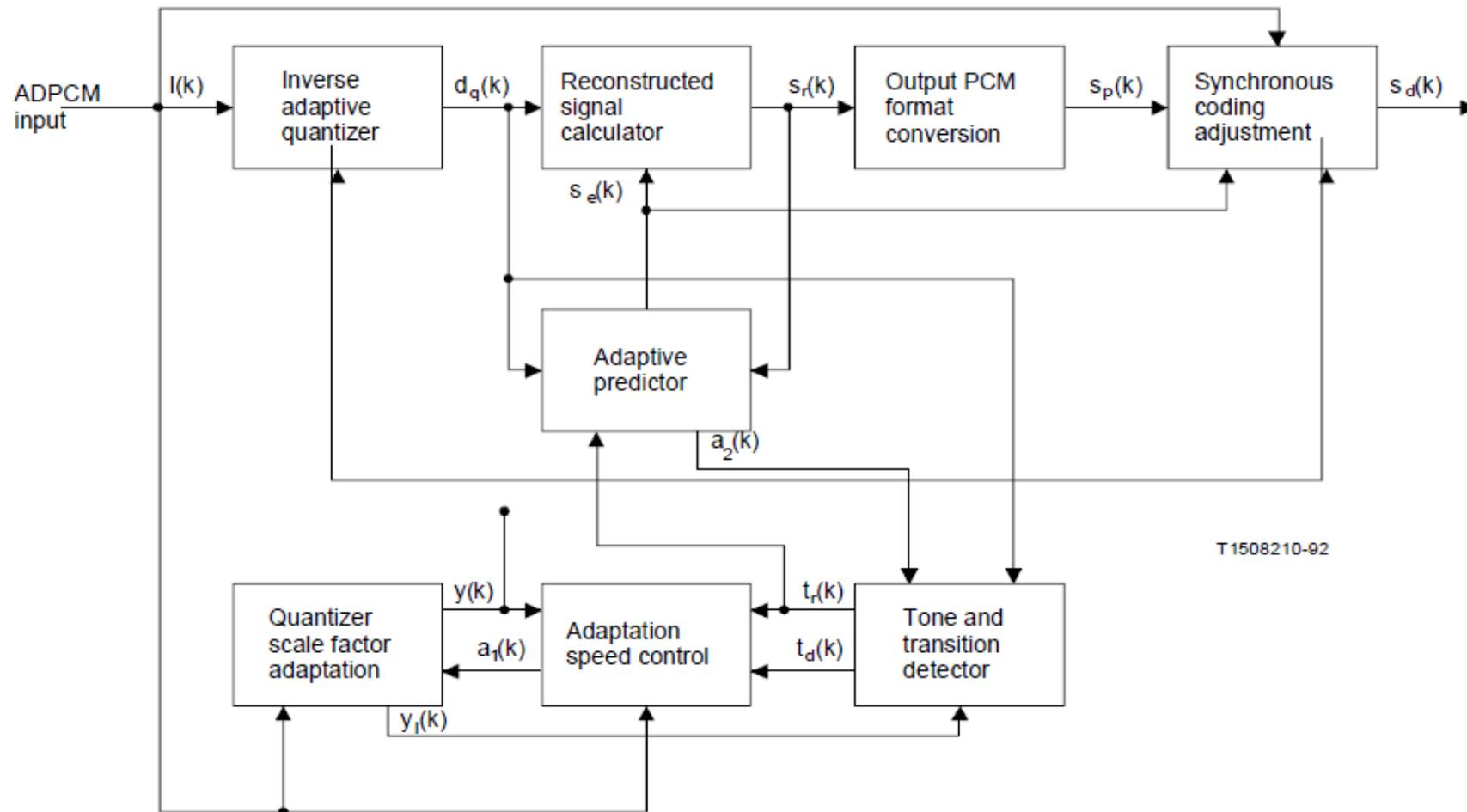


Figura 3. 4. Esquema de bloques del Decodificador G.726.

A continuación se presenta la descripción de todos y cada uno de los bloques diseñados, con el objetivo de clarificar el funcionamiento, tanto del codificador como del decodificador.

- **Conversión del formato PCM de entrada**

Este bloque, Fig. 3.5, se encarga de transformar la señal codificada en *ley A* a un formato de PCM uniforme, es decir, que reconstruye parcialmente la señal que fue codificada por medio del codificador G.711, con la intención de que las siguientes etapas puedan predecir los valores futuros de la señal.

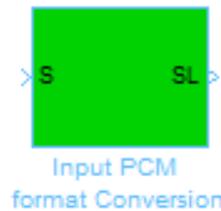


Figura 3.5. Bloque de conversión a PCM.

- **Cálculo de la señal de diferencia**

El presente bloque, Fig. 3.6, se encarga de calcular la diferencia entre la señal PCM uniforme $s_l(k)$, proveniente del bloque anterior, y la señal estimada $s_e(k)$ que emana del predictor adaptativo. En resumen, éste bloque hace las veces de diferenciador.



Figura 3.6. Bloque del cálculo de la diferencia de las señales de entrada y estimada.

▪ Cuantificador Adaptativo

Como ya se mencionó anteriormente un cuantificador, Fig. 3.7, es un elemento que se encarga de asignar ciertos valores discretos, dependiendo del rango de la señal de entrada. Así, éste bloque es un cuantizador no uniforme de 31, 15, 7 o 4 niveles que son usados para cuantificar la señal $d(k)$. Los citados niveles, funcionan a 40, 32, 24 y 16 Kbps.

Es importante recalcar que para el presente proyecto, únicamente se tomó en cuenta el cuantizador no uniforme de 15 niveles, es decir de 32 Kbps.

Para lograr el objetivo de realizar una cuantización no uniforme, éste bloque convierte la señal $d(k)$ a una base logarítmica de 2. Éste valor está limitado por el factor de escala $y(k)$.

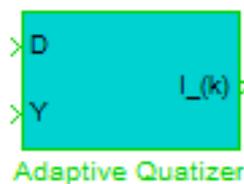


Figura 3. 7. Bloque Cuantizador Adaptativo.

▪ Cuantificador Adaptativo Inverso

Al inverso del bloque anterior, éste bloque, Fig. 3.8, se encarga de reproducir una versión cuantificada de la señal $d_q(k)$ en función del factor de escala $y(k)$. Esta señal pasa al dominio logarítmico.

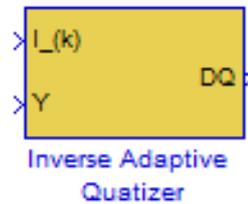


Figura 3. 8. Bloque Cuantizador Adaptativo Inverso.

▪ Adaptación del factor de escala del Cuantificador

Para determinar el factor de escala $y(k)$, se procede conforme el principio de adaptación bimodal, que manifiesta que existen dos velocidades de adaptación. Una rápida, para señales con fluctuaciones prominentes, como las señales de voz, y otra lenta, para señales con mínimas fluctuaciones, como las señales de datos usando un tono. Una combinación entre ambas velocidades da lugar a una adaptación más eficiente.

Esto se explica de la siguiente manera. Las señales de audio y voz, poseen cambios pronunciados y por lo tanto se puede separar a dicha señal en tramos, unos en los cuales las oscilaciones sean grandes, para lo cual se necesita una adaptación rápida y otros en los cuales éstas son imperceptibles, en donde la adaptación lenta es suficiente. Por lo tanto, una mezcla entre ambos factores puede ser la solución ideal.

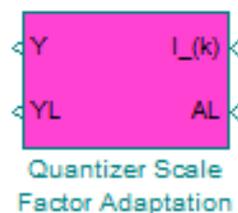


Figura 3. 9. Bloque del Factor de Adaptación de Cuantización de Escala.

▪ Control de la velocidad de Adaptación

El parámetro de control $a_l(k)$, varía en el rango entre 0 y 1. Se acerca a cero cuando las señales son tonos usados para señales de datos y tiende a uno cuando las señales son vocales. Esto quiere decir que, éste factor le imprime mayor o menor velocidad de adaptación al sistema, por lo tanto su importancia dentro del mismo es fundamental, Fig. 3.10.

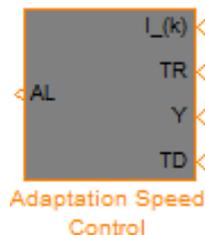


Figura 3. 10. Control de Velocidad de Adaptación.

▪ Calculador de la señal Reconstruida y Predictor Adaptativo

La tarea principal, desarrollada por este bloque, es la de emitir una señal estimada $s_e(k)$, a partir de una señal cuantificada $d_q(k)$. El predictor adaptativo consta de dos secciones que responden eficazmente a la diversidad de las señales de entrada. La primera sección es un banco de ceros de sexto orden y la segunda, un banco de polos de segundo orden. Con esta estructura, éste bloque pretende poseer todas las posibilidades, para que puedan responder a las exigencias planteadas por los distintos tipos de señales.

Estos bloques fueron fusionados, Fig. 3.11, ya que la recomendación así lo requería. Sin embargo, es fácil distinguir que éste bloque cumple dos propósitos, el primero, como ya se mencionó, es predecir una señal, y el segundo es reconstruirla, es decir que no basta con predecir un dato, sino que además debe reconstruirse en función de sus anteriores estados.

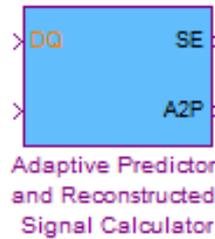


Figura 3.11. Bloque de Predictor Adaptativo y Calculador de la Señal Reconstruida.

▪ **Detector de Tono y Transición**

Éste bloque, Fig. 3.12, es útil en señales que usan un tono para transmitir la información, no necesariamente voz. Es de gran ayuda, ya que al detectar la estabilidad en el tono de la señal, ayuda a que el bloque de control de velocidad de adaptación, disminuya la velocidad y en consecuencia el sistema alcance el equilibrio. Ocurre lo contrario con las señales de voz.

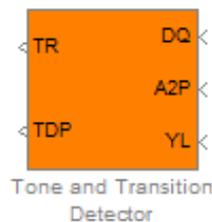


Figura 3.12. Bloque de Detección de Tono y Transición.

En base a lo planteado, en las recomendaciones previamente mencionadas, el diseño implementado es el mostrado a continuación, el mismo que utilizando las herramientas de *Simulink* que constan en sus librerías como embebiendo código *.m* en un bloque *embedded*, abarca todas las funciones y cálculos que constan en el algoritmo de funcionamiento del código (Anexo B).

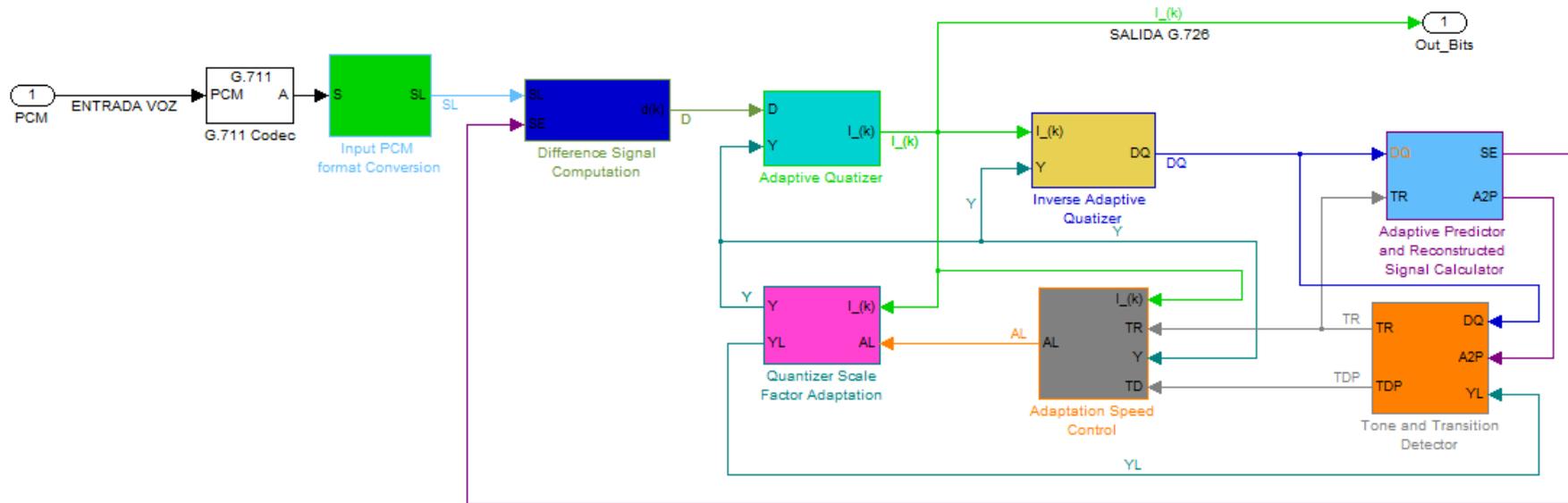


Figura 3. 13. Esquema del Codificador G.726 implementado.

Es importante y fundamental mencionar que debido a la cantidad de bloques utilizados para la implementación del códec, no es posible indicar el interior de cada uno de los bloques, ya que de manera más compacta se lo indica en el Anexo B del presente proyecto.

▪ Implementación del Decodificador

Por el lado del decodificador, tal como se puede observar en la Fig. 3.15, la única diferencia es que además de todos los bloques ya mencionados, se suma uno más, el de Ajuste de Codificación Síncrona. Por dicha razón, en éste punto únicamente se hará mención de dicho bloque, ya que los demás fueron ya descritos.

▪ Ajuste de Codificación Síncrona

Este bloque, Fig. 3.14, tiene por objetivo, eliminar la distorsión acumulativa producida por las codificaciones síncronas en cascada, es decir pasar de PCM a ADPCM y nuevamente a PCM. En consecuencia, elimina la probabilidad de no escuchar idealmente la señal, a la salida del decodificador.

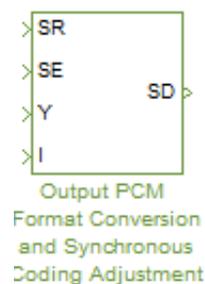


Figura 3. 14. Bloque de Ajuste de Codificación Síncrona.

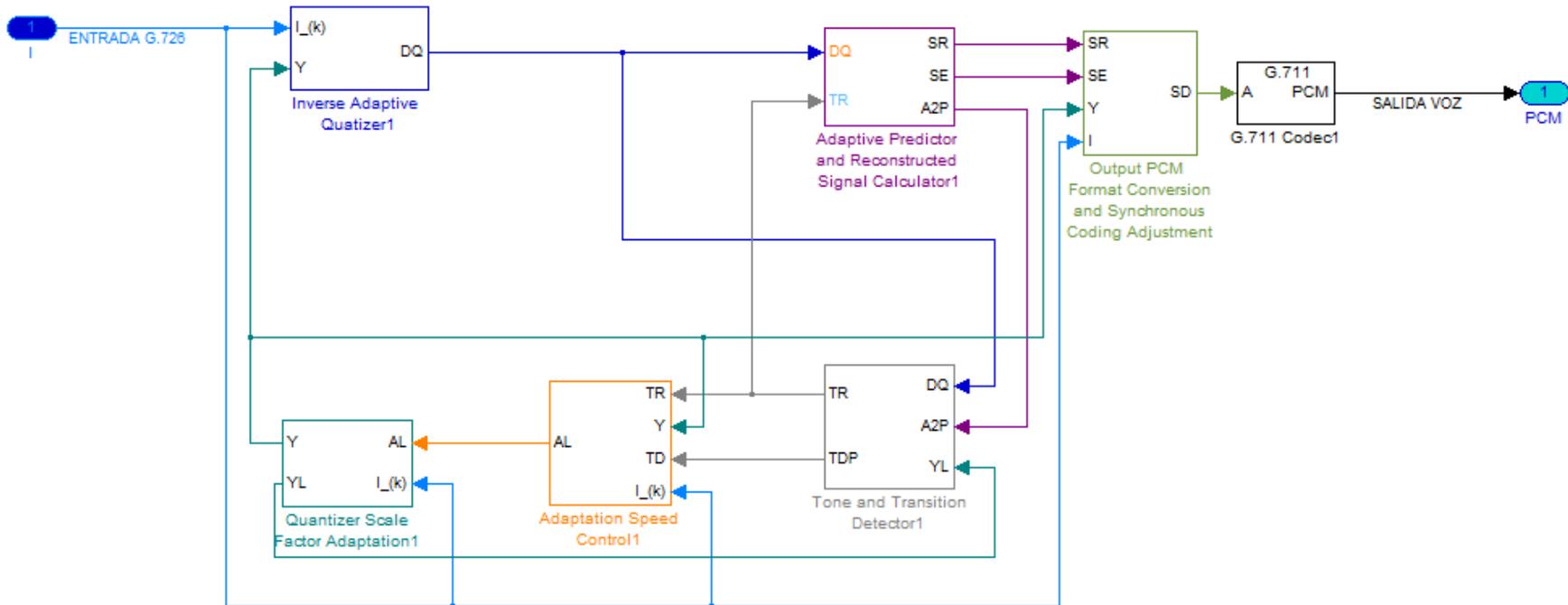


Figura 3. 15. Esquema del Decodificador G.726 implementado.

3.3. Codificación de Canal

3.3.1. Código Convolutivo

El SDR utiliza una codificación de canal tipo convolutivo, debido a que estos códigos de bloques tienen una gran capacidad de corrección de errores, ya que están diseñados para trabajar con canales que presentan mucha interferencia y también gran presencia de ruido no deseado [17].

El proceso de codificación y decodificación será el mismo que se detalló en el CAPÍTULO II de la presente investigación.

La señal obtenida del codificador de fuente será de tipo entero sin signo representada hasta con 8 bits, la cual no es compatible con los tipos de datos que trabajan los códigos convolutivos, por lo que es necesario cambiar los datos al tipo binario, es decir o “cero” o “uno”, de igual forma en los parámetros de configuración de los datos del codificador convolutivo se deberá considerar una *Trellis* con una tasa de transferencia de $\frac{1}{2}$, lo que implica una entrada dos salidas o dos bits de paridad [18].

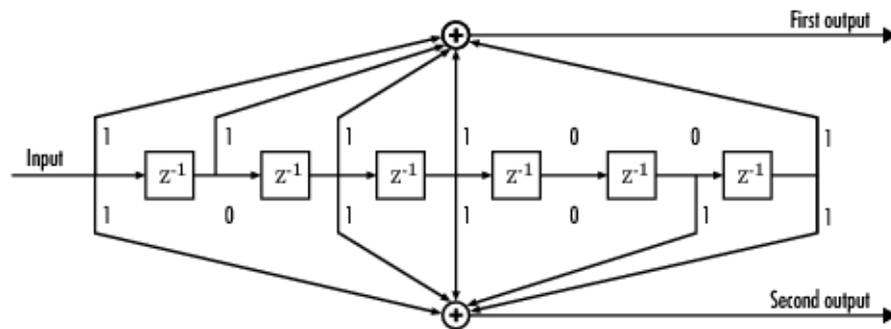


Figura 3. 16. Diagrama de registros del código Convolutivo.

De igual forma el número de registros que utilizará el codificador será de 3, lo que implica que la representación octal del proceso de codificación será [7 5]. Finalmente, la representación de la *Trellis* será por medio del comando poly2trellis de *Matlab*, el mismo que devuelve los coeficientes del código [19].

Conjuntamente con esto, se va a usar en modo de operación continua, esto debido a la exigencia que presentan, especialmente las señales de voz,

obteniendo finalmente el codificador de canal contiguo al codificador de fuente, como indica la Fig. 3.17, y el diagrama del codificador con los respectivos registros como lo indica la Fig. 3.16.

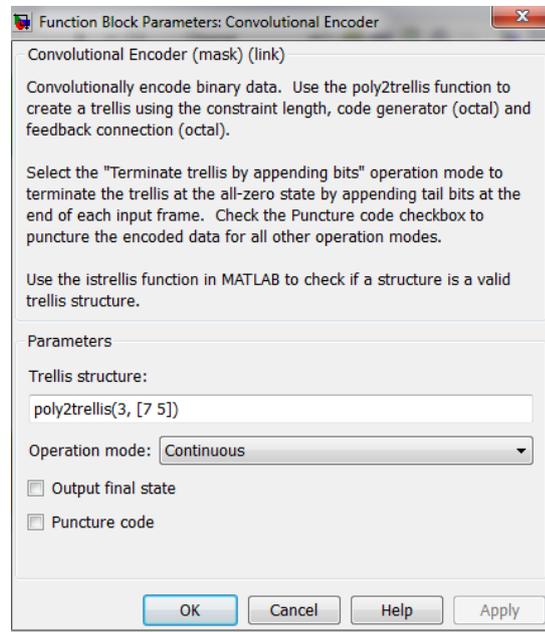


Figura 3. 17. Configuración de parámetros del codificador Convolutcional.

3.3.2.Código de Hamming

Para expresar el código se utiliza la forma n, k , la misma que se construye conociendo la tasa de bits de datos (r) a partir de la cual se procede a calcular los valores fundamentales [19].

$$n = 2^r - 1 \quad \text{Ecuación 3. 1}$$

Por ejemplo si $r = 2$ aplicando la Ec.3.1 se obtiene,

$$n = 2^2 - 1 = 3$$

$k = n - r = 2 - 1 = 1$, de ésta forma se expresa Hamming (3, 1), siendo $r=m$ y $m \geq 3$.

Uno de los códigos más usados es el Hamming (7, 4), en el cual se agregan 3 bits de paridad por cada cuatro bits que conformen el mensaje, los

mismos que intercalan en la palabra a transmitir en las posiciones correspondientes a las potencias de dos, es decir posición 1, 2, 4, 8, etc.

Sin embargo las limitaciones se observan en la cantidad de bits que pueden ser corregidos, debido a que éste código solo puede corregir 1 bit, sin embargo que puede detectar hasta dos bits erróneos estos no son corregidos en igual número; lo que implica que si el error está en más de un bit de forma consecutiva, la decodificación será errónea, consecuentemente la tasa de error será mayor [17].

Una característica importante de éstos códigos es que si se mantiene la potencia promedio de la señal codificada y no codificada, la energía del bit disminuye debido a un “1” cada vez tendrá menor duración; otro aspecto importante es que la tasa de símbolo aumenta en múltiplos de n/k .

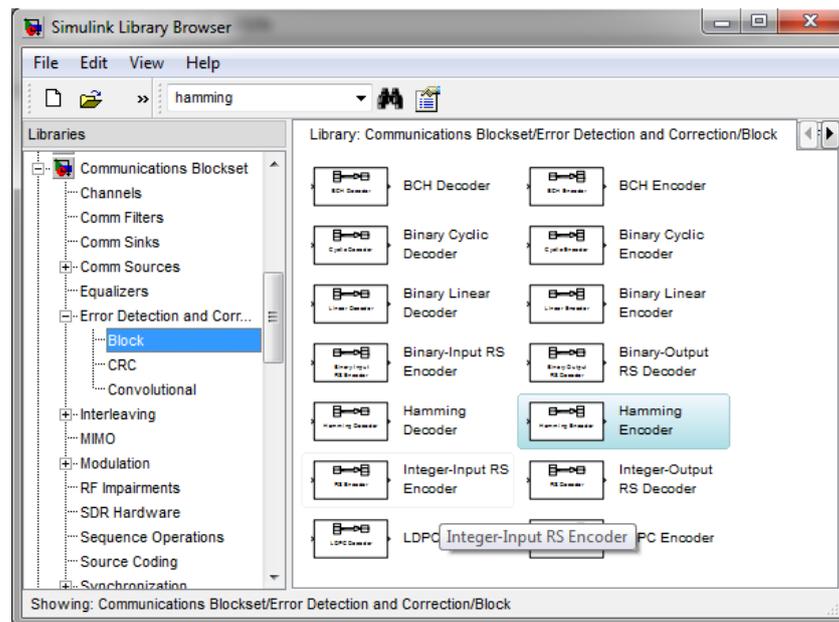


Figura 3. 18. Menú de Simulink.

En función de esto, *Simulink* ofrece un bloque exclusivo, Fig. 3.18, tanto para el codificador como para el decodificador, para el código de *Hamming*. Para su configuración, tan solo hace falta colocar, tanto el valor de n como el de k , en codificador y decodificador, como indica la Fig. 3.19.

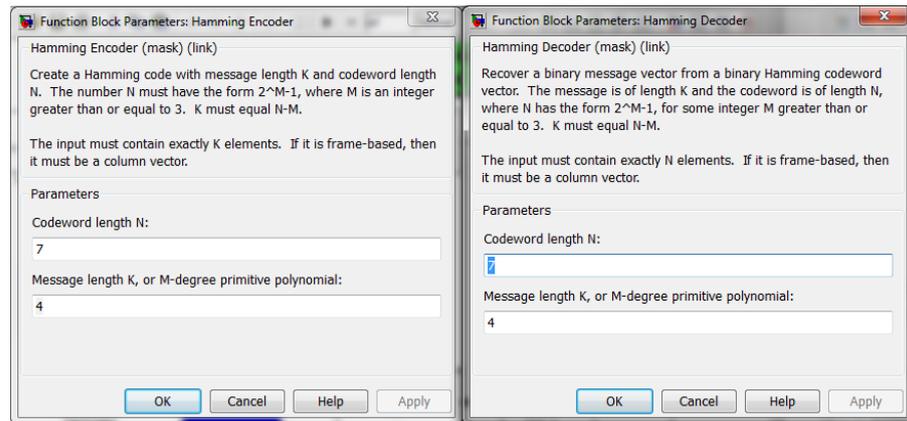


Figura 3. 19. Configuración del bloque codificador y decodificador de Hamming.

3.3.3. Decodificador Viterbi

La decodificación de canal convolucional se la realizó con un decodificador que utiliza el algoritmo de *Viterbi*, el cual se fundamenta en un análisis de la *Trellisa* [19] proveniente del codificador, la cual es analizada en base a un procedimiento especificado en el CAPÍTULO II [18].

Los parámetros de configuración del decodificador *Viterbi*, vienen dados por las características de la *Trellisa*, la cual debe ser la misma utilizada en codificador. Además también se considerará el tipo de dato a utilizar y sobre todo es fundamental el *tracebackdepth*, que es la cantidad de ceros que el proceso ubicará al inicio de la decodificación como retardos, para que se pueda sincronizar el proceso. Este valor es igual al número de registros más los bits de paridad usados para la codificación, si este valor no es el adecuado el proceso de decodificación será totalmente erróneo [17]. Los parámetros de configuración de dicho bloque se muestran en la Fig. 3.20.

Es importante destacar que se puede optar por utilizar un código *punctured* que ayuda a mejorar el proceso, y otro aspecto fundamental es seleccionar el modo de operación del decodificador que puede ser continuo, truncado o terminado, pero que para este diseño se consideró modo continuo ya que este trata a los datos de forma independiente, además este modo debe ser el mismo que se utilizó en el codificador convolucional.

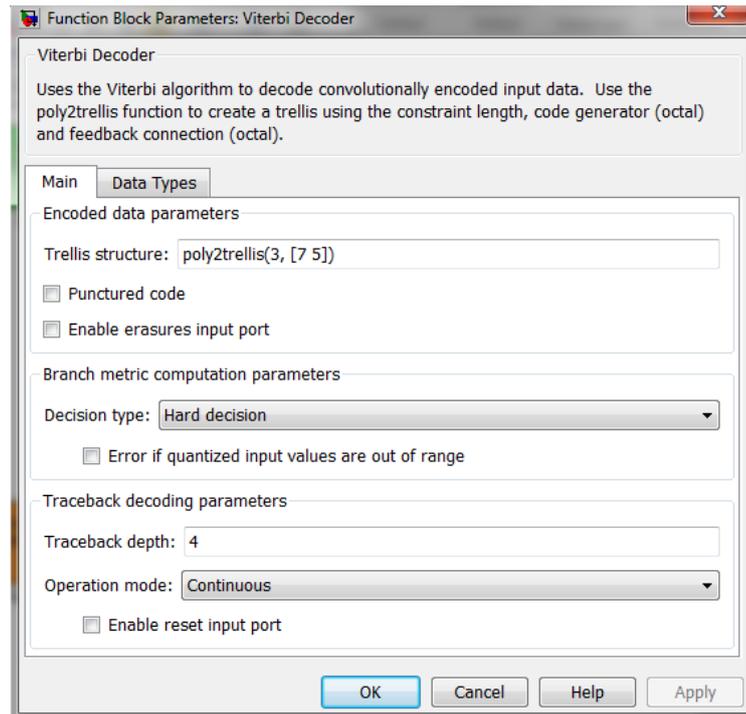


Figura 3. 20. Configuración del decodificador de Viterbi.

3.4. Modulación

3.4.1. En Banda Base: QPSK

El SDR diseñado, estará implementado con una modulación QPSK [20], es decir que modifica la fase de la señal de forma cuaternaria, lo que implica una mayor eficiencia, una disminución del uso del ancho de banda, pero a su vez mayor complejidad en la codificación, en consecuencia ésta modificación angular permite que la fase de la portadora varíe en valores discretos (cuatro para este caso). Este bloque es posible encontrarlo dentro de *Simulink*, y su constelación puede ser configurada según las necesidades, tal como indica la Fig. 3.21.

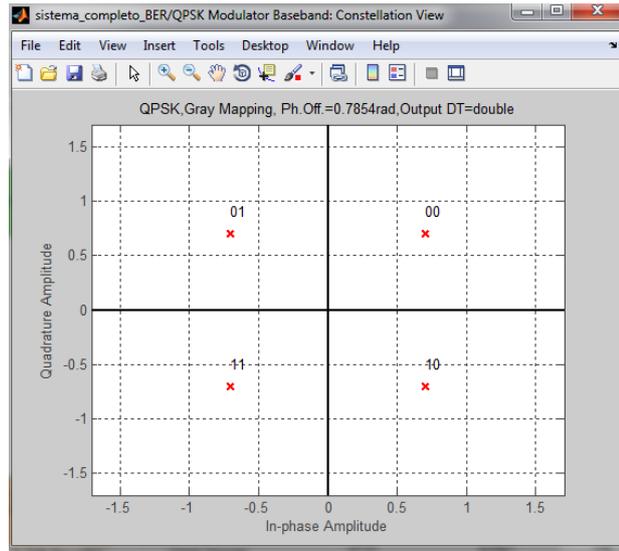


Figura 3. 21. Constelación del bloque de modulación QPSK de Simulink.

Para estas condiciones de modulación de la señal los cálculos de la probabilidad de error se rigen a la Ec. 3.2 como se indica a continuación [15]:

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad \text{Ecuación 3. 2}$$

En tanto que la tasa de transmisión se analiza con la Ec. 3.3:

$$P_s = 1 - (1 - P_b)^2 = 2Q\left(\sqrt{\frac{E_s}{N_0}}\right) - Q^2\left(\sqrt{\frac{E_s}{N_0}}\right) \quad \text{Ecuación 3. 3}$$

Y su diagrama de constelación será de orden 2 debido a que trabaja con 4 símbolos, además utilizará código Gray y sus intervalos de variación serán 90° o $\pi/2$, pero es importante considerar que si los datos no cambia de una fase a la otra la portadora tampoco lo hace, mientras que si un bit cambia la portadora varía 90° y si los dos bits cambian la portadora varía 180° .

3.4.2. En Frecuencia: FM

Esta implementación se la muestra en la Fig. 3.22, de la cual se debe considerar que el retardo es igual a uno, ya que al modularse en función de seno y coseno la sumada del cuadrado de cada uno de estos ($i^2(n) + q^2(n)$)

[9] será igual a uno y la señal al multiplicarse por uno no sufre ninguna modificación.

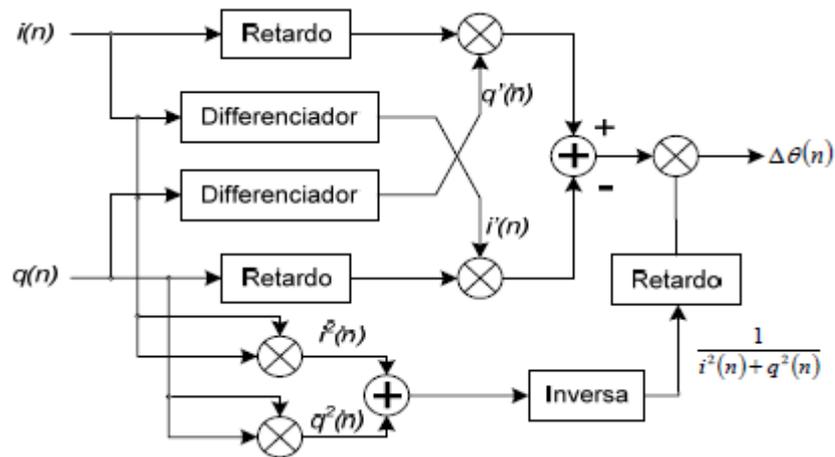


Figura 3. 22. Método de demodulación FM de Richard G. Lyon.

Por lo tanto, y tomando en cuenta lo que se acaba de citar, se procedió a implementar éste tipo de modulación y demodulación, tal como indican las Fig. 3.23 y Fig. 3.24.

Para el diseño del modulador se hizo uso de un bloque integrador. Con esto lo que se consigue es realizar una sumatoria de todas las muestras, esto teniendo en cuenta que la señal está discretizada, y en consecuencia lo que se busca es obtener el módulo y la fase de la señal, tal como se citó en el CAPÍTULO II.

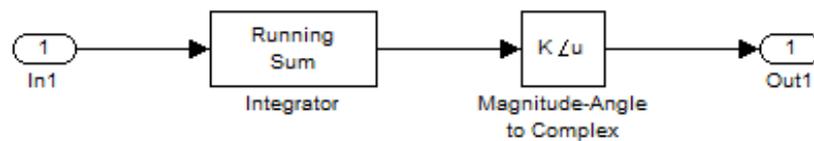


Figura 3. 23. Implementación modulador FM.

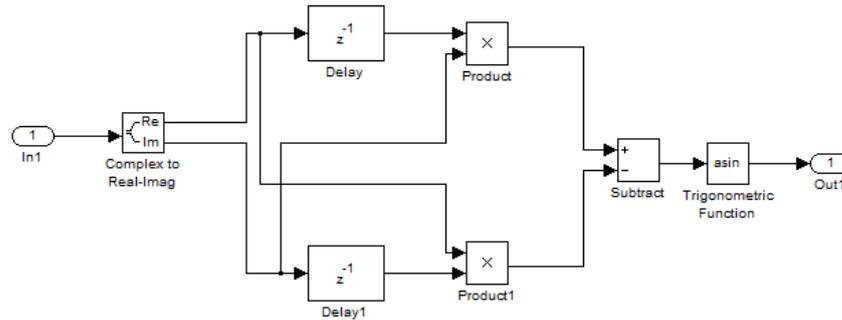


Figura 3. 24. Implementación demodulador FM indirecto.

Para el caso del diseño del demodulador, se siguió el método de Richard G. Lyon [9], como se muestra en la Fig. 3.22. El uso de la función de *arco seno*, se justifica de manera que ha sido utilizada para limitar la salida del demodulador en el rango entre ± 1 .

3.5. Manejo y Configuración de los bloques de *Simulink USRP*

3.5.1. Configuración módulo *sink*

Luego de la etapa de modulación, sea en banda base o FM, se debe conectar el sistema al *USRP*, el mismo que de la librería instalada en *Simulink*[®] se debe seleccionar la tarjeta *USRP Sink*, ya que está trabaja como un transmisor, es decir es el dispositivo de *front-end*, en tanto que para la recepción se utilizará el *USRP Source*, es decir, que al *USRP* va a ser utilizado como fuente.

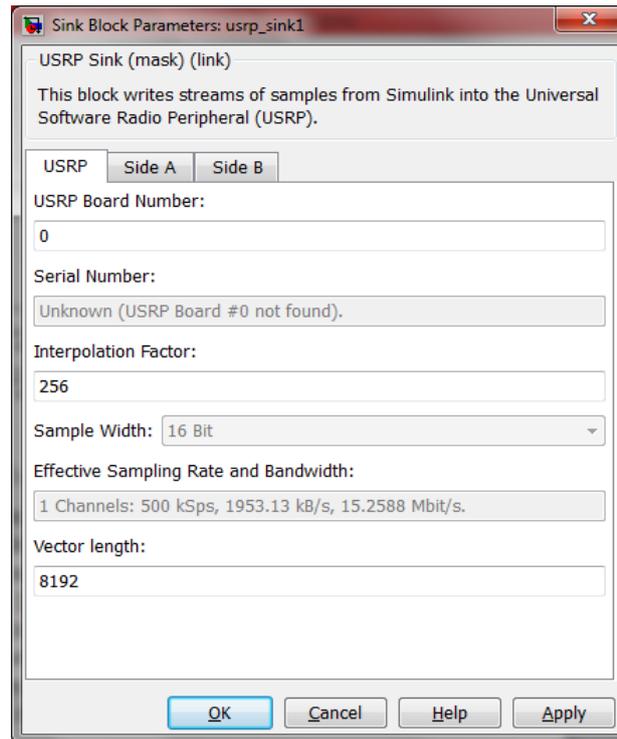


Figura 3. 25. Interfaz de configuración del *USRP sink* en *Simulink*.

La configuración fundamental que requiere el *USRP* son: factor de interpolación y la longitud del vector, es decir, la cantidad de datos o muestras por segundo a ser procesadas y la longitud de la palabra codificada, respectivamente, como se observa en la Fig. 3.25.

El módulo *sink* posee tres pestañas de configuración. Entonces se puede seleccionar el *SIDE A* o *SIDE B*, en los cuales se deben configurar la frecuencia de trabajo, la ganancia y la forma de expresión de los datos, esto se detalla de mejor forma en la Fig. 3.26.

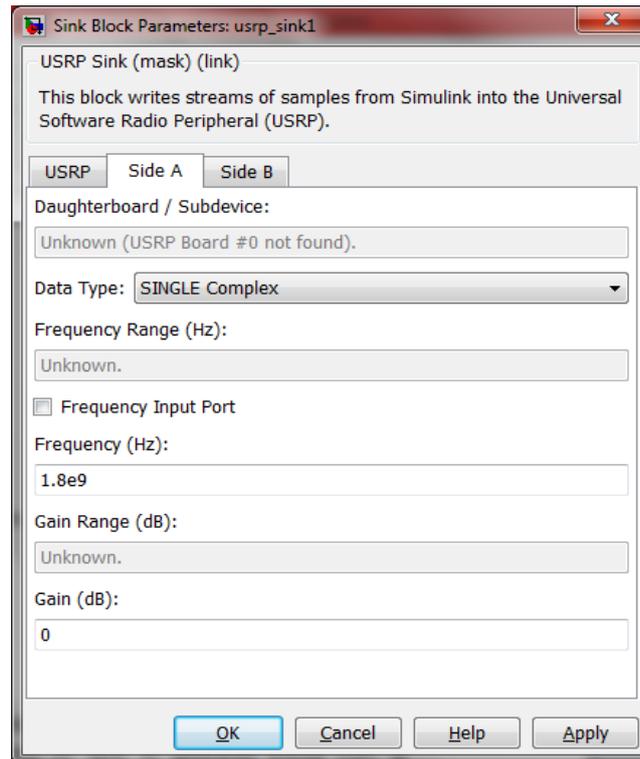


Figura 3. 26. Configuración de frecuencia de trabajo del módulo *USRP sink* de *Simulink*.

3.5.2. Configuración módulo *source*

Para la configuración de éste módulo, al igual que el anterior, se debe tomar en cuenta lo que se mencionó en el CAPÍTULO II, en lo referente a la capacidad de los conversores *ADC* y *DAC*. Los primeros, tienen una capacidad teórica de procesamiento máxima de 64 millones de muestras por segundo, es decir, ésta es la capacidad del receptor. Dicho de otra manera, tiene un ancho de banda de 64MHz. Sin embargo, en el caso del *DAC*, la situación se duplica, es decir, tiene una capacidad de procesamiento de 128 millones de muestras por segundo, lo que equivale a decir que se pueden transmitir señales de hasta 128MHz de ancho de banda.

Para calcular el ancho de banda a ser usado por las tarjetas, se debe seguir la Ec. 3.4:

$$BW_{TX} = \frac{128e^6}{\text{Factor de Interpolación}} \quad \text{Ecuación 3. 4}$$

Lo anterior para el transmisor, mientras que para el receptor, se usa la Ec. 3.5:

$$BW_{RX} = \frac{64e^6}{\text{Factor de Decimación}} \quad \text{Ecuación 3. 5}$$

En consecuencia, siempre deberá cumplirse la Ec. 3.6 que indica:

$$BW_{TX} = BW_{RX} \quad \text{Ecuación 3. 6}$$

Con la finalidad de mantener la igualdad entre el transmisor y el receptor.

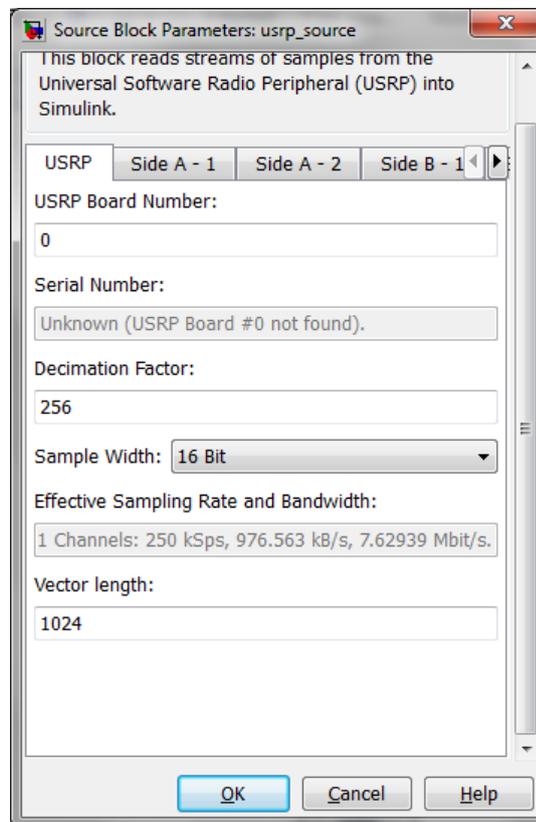


Figura 3. 27. Interfaz de configuración del *USRP source* en *Simulink*.

Al igual que en el inciso anterior, la configuración deberá ejecutarse, tal cual indican las Fig. 3.27 y Fig. 3.28.

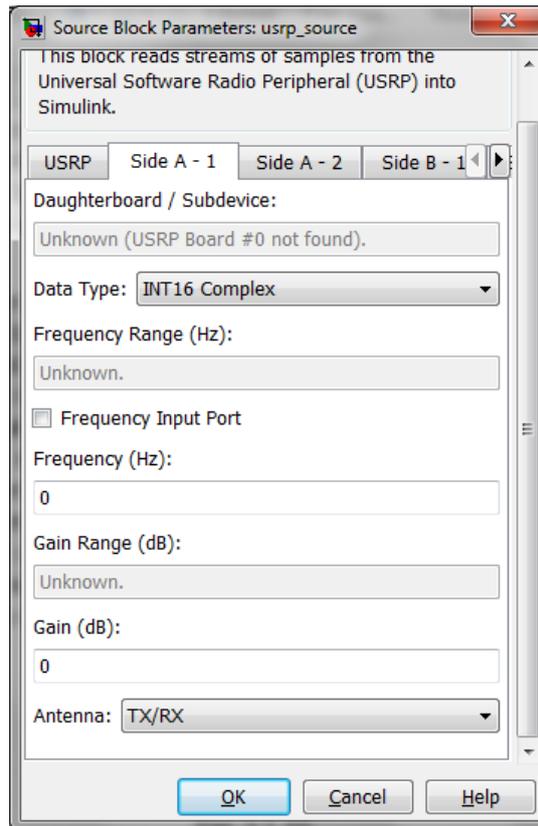


Figura 3. 28. Configuración de frecuencia de trabajo del módulo *USRP source* de *Simulink*.

La manera de configuración de la frecuencia de trabajo viene dada única y exclusivamente por las *daughterboards*, las mismas que por sus características son capaces de trabajar en distintos rangos de frecuencia, como se mencionó en el CAPÍTULO II.

Finalmente para propagar la señal hacia el espacio, se debe aplicar una modulación analógica, la misma que es implementada por medio de *software* y de igual forma utilizando las potencialidades de *Simulink*, en este punto es importante destacar que para que la señal se propaga eficientemente es fundamental tener una potencia adecuada de salida, la cual se la obtiene a través de un amplificador de potencia, que básicamente representa *hardware*, debido a que las frecuencias de trabajo son bajas, la potencia a utilizar permitida es alta hasta el orden de las decenas de *watios*. Sin embargo se ha determinado por la fórmula de propagación en el espacio libre que con una potencia de salida de 1 *watio*, es suficiente para alcanzar una cobertura de

aproximadamente 100 m, además se debe considerar las condiciones en las que puede trabajar el *USRP*, ya que para utilizar un valor de potencia más elevado, también habría que realizar adecuaciones de ventilación y disipación de calor para las tarjetas a fin de evitar mayores daños.

CAPÍTULO IV

SIMULACIONES, PRUEBAS Y RESULTADOS EXPERIMENTALES

4.1. Introducción

En éste capítulo se presentan todos los modelos diseñados, y además se exponen varias características de los mismos, que serán el fundamento esencial del desarrollo de éste proyecto.

A modo de comparación, además, se ponen de manifiesto las gráficas de la tasa de error de bit (*BER*) a fin de proporcionar el sustento técnico necesario para proceder a la implementación del sistema.

En la sección de las pruebas experimentales se asientan varias de las incógnitas planteadas al inicio del proyecto, con el propósito de demostrar la consecución de los objetivos planteados.

4.2. Simulaciones

4.2.1. Modelo I

El primer modelo consta de los siguientes componentes:

- En el Transmisor:
 - Codificador de Fuente: G.711
 - Codificador de Canal: Convolutacional
 - Modulación: QPSK
- En el Receptor:

- Decodificador de Fuente: G.711
- Decodificador de Canal: Viterbi
- Demodulación: QPSK

Por lo tanto, el modelo diseñado es el siguiente:

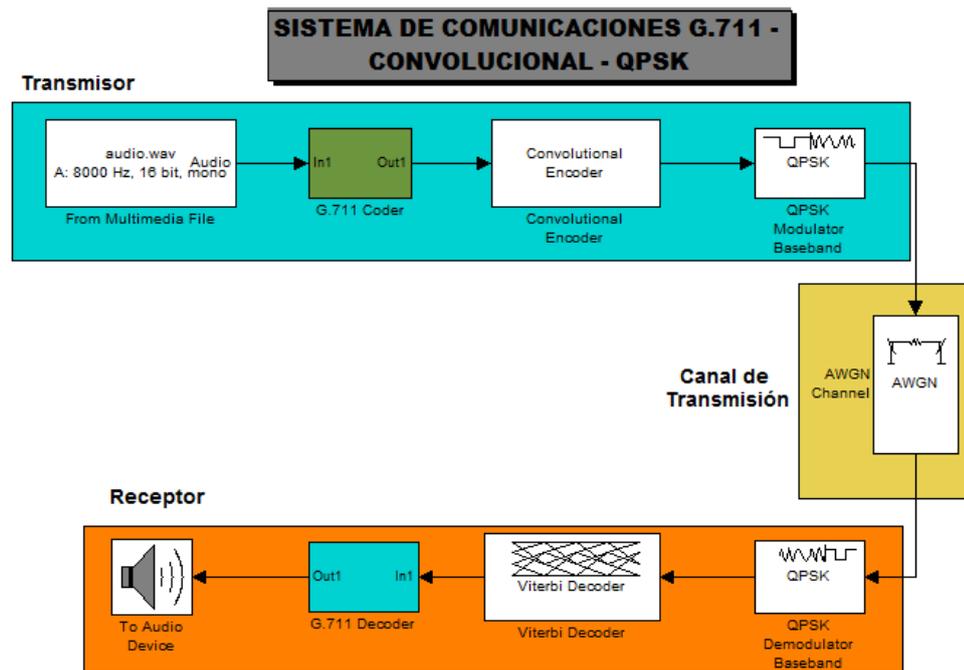


Figura 4. 1. Sistema de Comunicaciones propuesto Modelo I.

4.2.2. Modelo II

El segundo modelo consta de los siguientes componentes:

- En el Transmisor:
 - Codificador de Fuente: G.711
 - Codificador de Canal: Hamming
 - Modulación: QPSK
- En el Receptor:

- Decodificador de Fuente: G.711
- Decodificador de Canal: Hamming
- Demodulación: QPSK

Por lo tanto, el modelo diseñado es el siguiente:

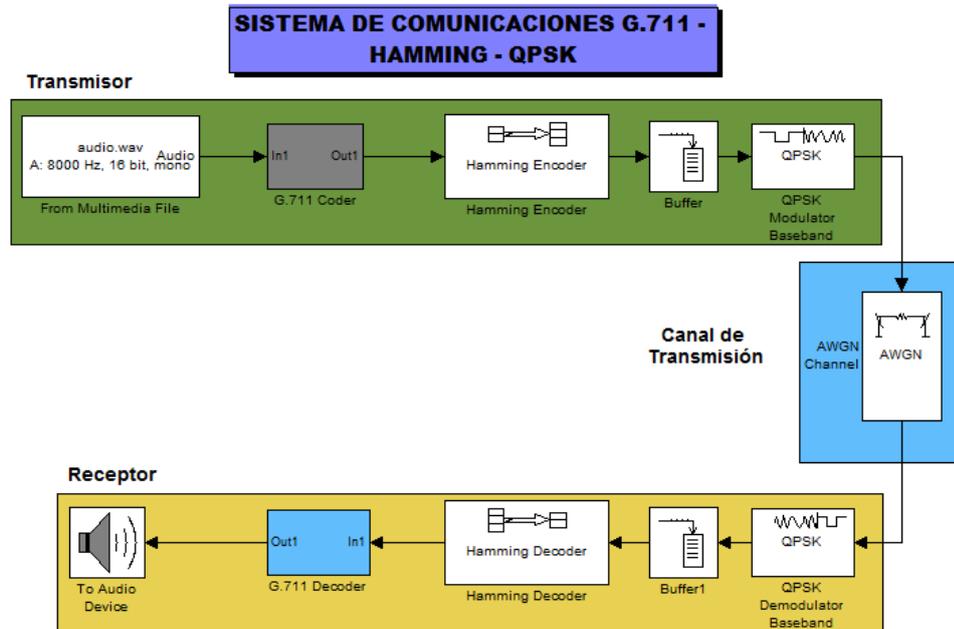


Figura 4. 2. Sistema de Comunicaciones propuesto Modelo II.

4.2.3. Modelo III

El tercer modelo consta de los siguientes componentes:

- En el Transmisor:
 - Codificador de Fuente: G.726
 - Codificador de Canal: Convolutacional
 - Modulación: QPSK
- En el Receptor:

- Decodificador de Fuente: G.726
- Decodificador de Canal: Viterbi
- Demodulación: QPSK

Por lo tanto, el modelo diseñado es el siguiente:

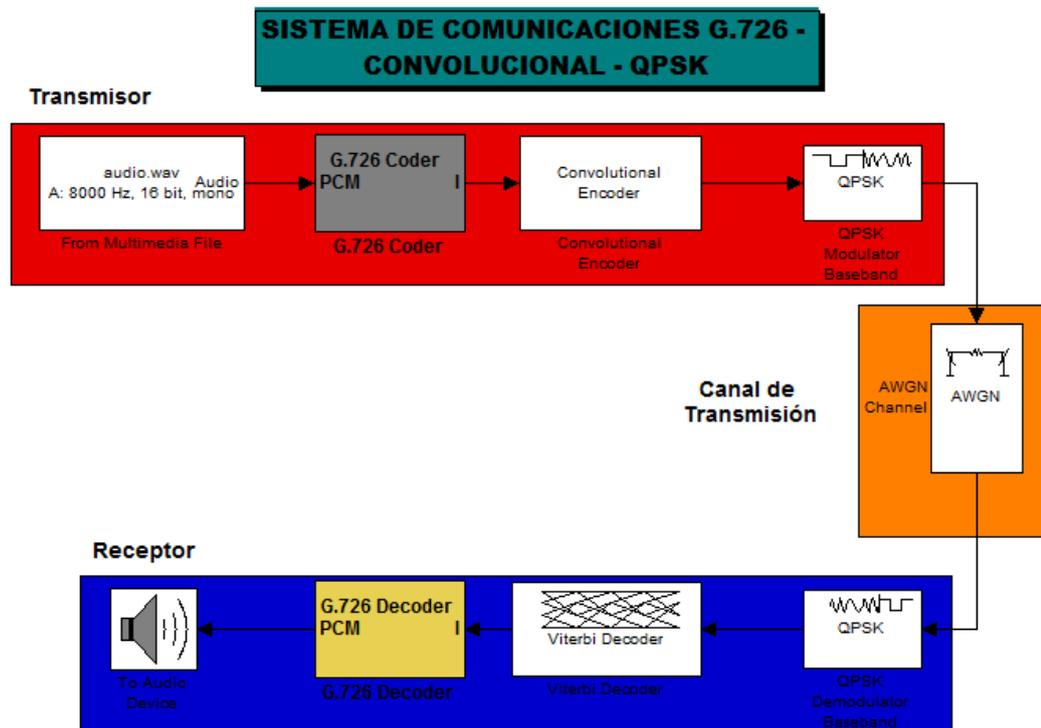


Figura 4. 3. Sistema de Comunicaciones propuesto Modelo III.

4.2.4. Modelo IV

El cuarto modelo consta de los siguientes componentes:

- En el Transmisor:
 - Codificador de Fuente: G.726
 - Codificador de Canal: Hamming
 - Modulación: QPSK

- En el Receptor:
 - Decodificador de Fuente: G.726
 - Decodificador de Canal: Hamming
 - Demodulación: QPSK

Por lo tanto, el modelo diseñado es el siguiente:

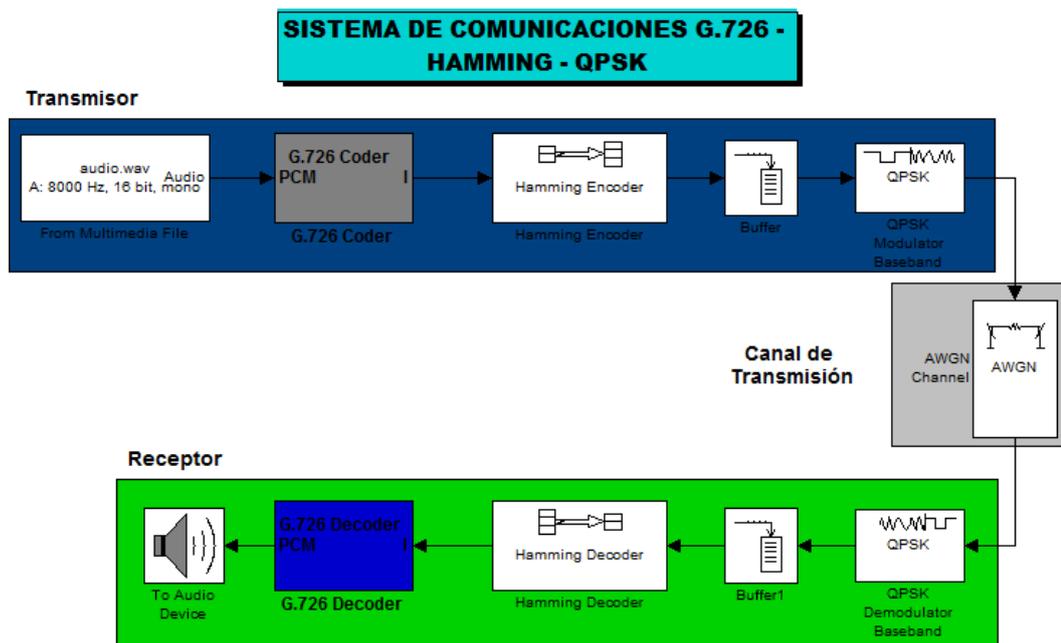


Figura 4. 4. Sistema de Comunicaciones propuesto Modelo IV.

4.2.5. Modelo V

El quinto modelo consta de los siguientes componentes:

- En el Transmisor:
 - Codificador de Fuente: G.711
 - Codificador de Canal: Convolutacional
 - Modulación: FM

- En el Receptor:
 - Decodificador de Fuente: G.711
 - Decodificador de Canal: Viterbi
 - Demodulación: FM

Por lo tanto, el modelo diseñado es el siguiente:

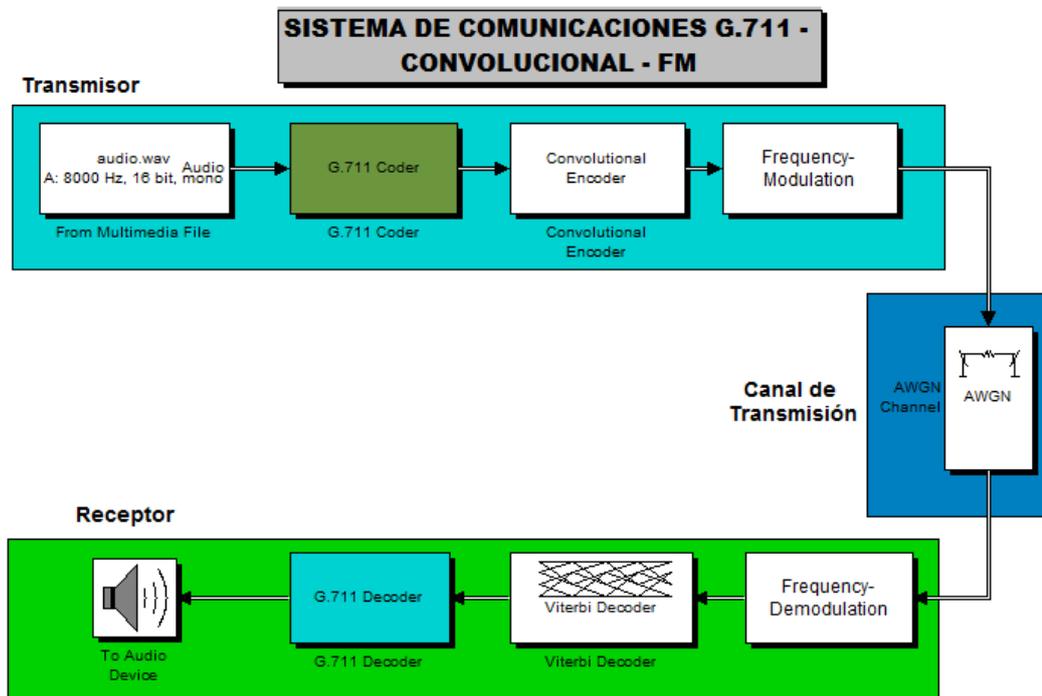


Figura 4. 5. Sistema de Comunicaciones propuesto Modelo V.

4.2.6. Modelo VI

El sexto modelo consta de los siguientes componentes:

- En el Transmisor:
 - Codificador de Fuente: G.711
 - Codificador de Canal: Hamming
 - Modulación: FM

- En el Receptor:
 - Decodificador de Fuente: G.711
 - Decodificador de Canal: Hamming
 - Demodulación: FM

Por lo tanto, el modelo diseñado es el siguiente:

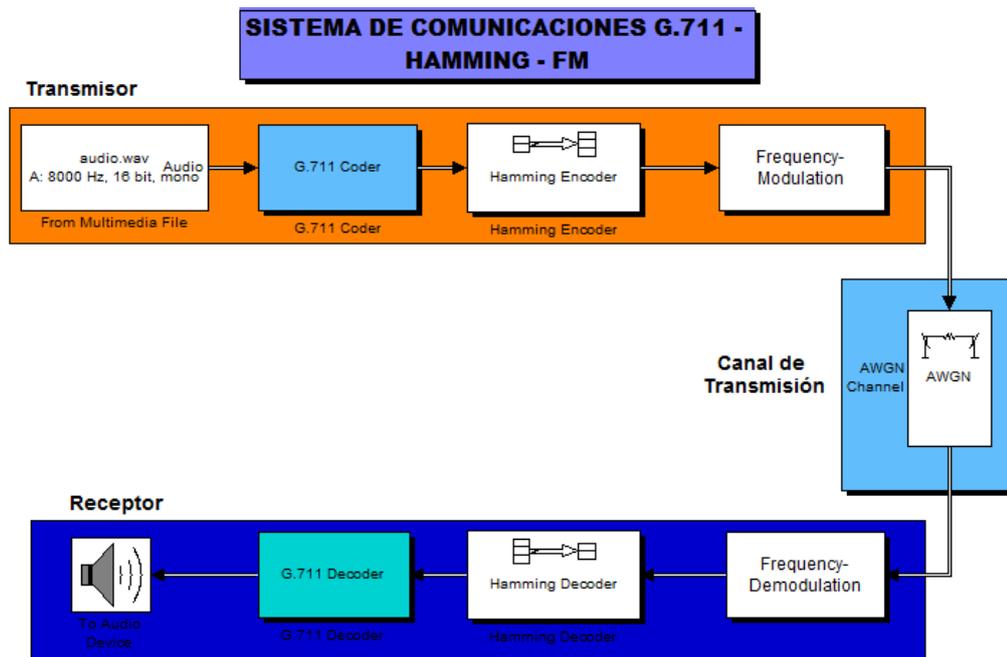


Figura 4. 6. Sistema de Comunicaciones propuesto Modelo VI.

4.2.7. Modelo VII

El séptimo modelo consta de los siguientes componentes:

- En el Transmisor:
 - Codificador de Fuente: G.726
 - Codificador de Canal: Convolutacional
 - Modulación: FM

- En el Receptor:
 - Decodificador de Fuente: G.726
 - Decodificador de Canal: Viterbi
 - Demodulación: FM

Por lo tanto, el modelo diseñado es el siguiente:

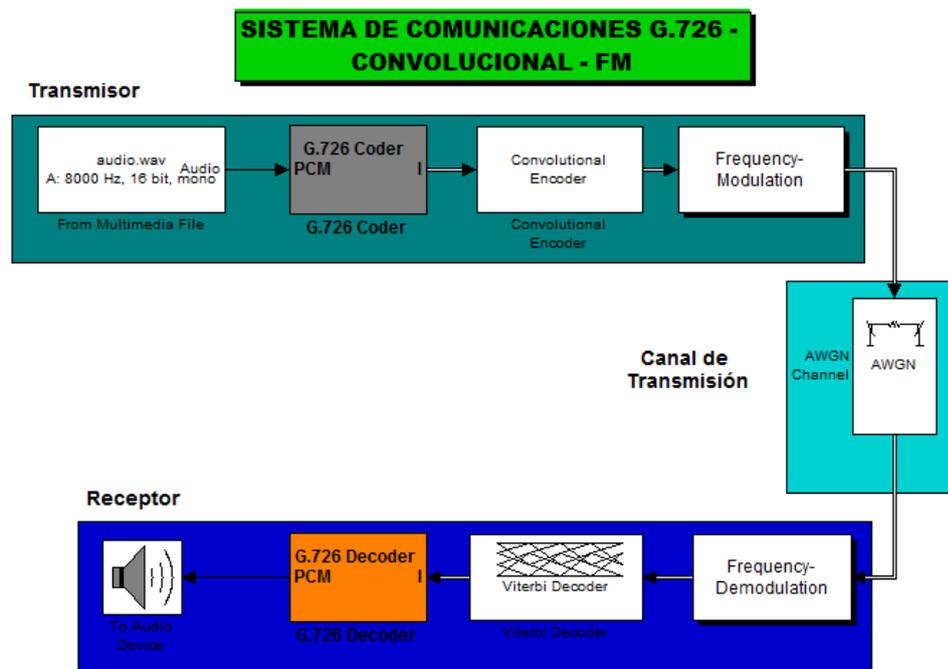


Figura 4. 7. Sistema de Comunicaciones propuesto Modelo VII.

4.2.8. Modelo VIII

El último modelo consta de los siguientes componentes:

- En el Transmisor:
 - Codificador de Fuente: G.726
 - Codificador de Canal: Hamming

- Modulación: FM
- En el Receptor:
 - Decodificador de Fuente: G.726
 - Decodificador de Canal: Hamming
 - Demodulación: FM

Por lo tanto, el modelo diseñado es el siguiente:

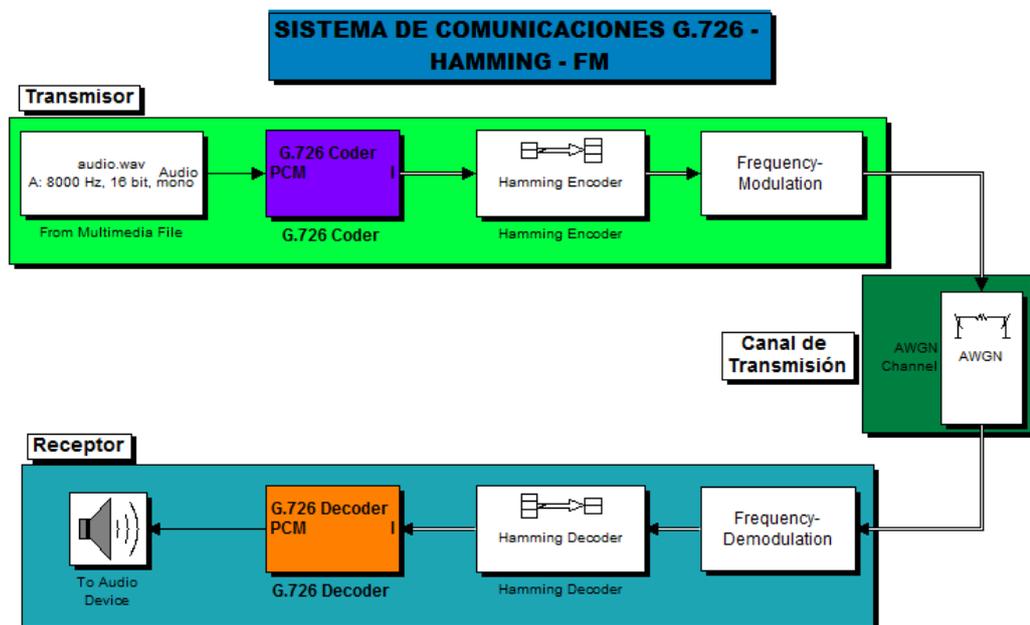


Figura 4. 8. Sistema de Comunicaciones propuesto Modelo VIII.

4.3. Cálculos teóricos

4.3.1. Calculo de la Tasa de Error

La tasa de error del bit, como su nombre lo indica mide las diferencias existentes entre un bit transmitido y su correspondiente bit recibido, para este caso esta relación se la puede medir directamente en una simulación utilizando las herramientas de *Simulink*[®] en dos puntos, el primero es comparando bit a bit en los procesamientos que utilizan estos datos como son

el codificador convolucional y el decodificador *Viterbi* ya que estos trabajan directamente con bits, además también se realizó esta medición comprando las diferencias entre los datos emitidos por la fuente (micrófono) y los datos obtenidos por el parlante a la llegada, pero con la salvedad que éstos datos no son binarios, si no son de tipo entero por lo cual primero se los transforma a binario y luego se realiza la comparación, en el primer caso como muestra la Fig. 4.9 el BER obtenido fue de 0, mediante el uso del bloque específico de *Simulink* para éste cálculo se pudo obtener dichos datos; es importante considerar los retardos que ha sufrido el proceso para configurar el bloque de cálculo del error y así evitar obtener resultados erróneos.

Dicho bloque arroja como respuesta un vector de 3 elementos de los cuales el primero es la tasa de bit erróneos, el segundo el número de bit erróneos a la llegada y el tercero muestra el número total de bits transmitidos, para esta simulación se utilizó un canal de ruido gaussiano blanco con relación energía del bit sobre ruido en el rango de -60 hasta 20.

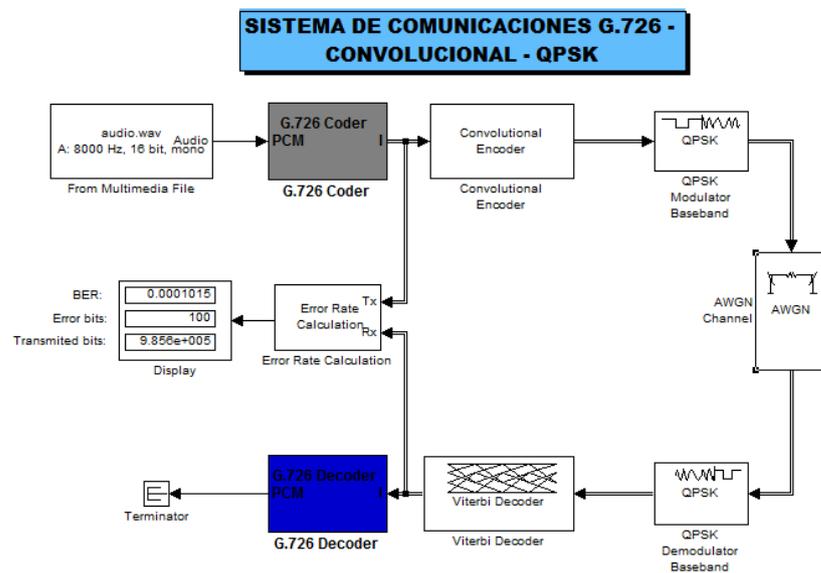


Figura 4. 9. Cálculo de la tasa de bits erróneos.

4.3.2.Cálculo del Ruido de Cuantificación.

El proceso de cuantización genera una diferencia entre la señal original y la señal cuantizada, a esta diferencia se la conoce como ruido de cuantización y su magnitud se la encuentra aplicando la Ec. 4.1:

$$e_q = x_q(n) - x(n) \text{ [99]}, \quad \text{Ecuación 4. 1}$$

Además se debe conocer que el rango de este error será el indicado en la Ec. 4.2:

$$-\frac{\Delta}{2} < e_q < \frac{\Delta}{2}; \quad \text{Ecuación 4. 2}$$

$$\Delta = \frac{R}{L}$$

Dónde R=Rango de cuantización y L= Número de niveles de cuantización. Para este caso el rango de cuantización es de 0 a 65565 y el número de bits utilizados para cuantizar, es decir los niveles es de 16 bits, aplicando la Ec. 4.2 se obtiene el valor de delta de la siguiente manera:

$$\Delta = \frac{65535}{65536} = 0,99998$$

Para la presente investigación dicho cálculo no es posible ya que la digitalización de la voz la hace el algoritmo que controla el micrófono y la señal obtenida ya está digitalizada.

4.3.3.Rendimiento del Error PSK

La expresión general para la probabilidad de error del bit de un sistema PSK de fase-M es la indicada en la Ec. 4.3:

$$P_e = \frac{1}{\log_2 M} \text{erf}(z) \quad \text{Ecuación 4. 3}$$

en donde erf (z) = función de error y se calcula en función de la Ec. 4.4

$$z = \sin \frac{\pi}{M} \sqrt{\log_2 M} \sqrt{\frac{E_b}{N_0}} \quad \text{Ecuación 4. 4}$$

$$\frac{E_b}{N_0} = \frac{C}{N} \times \frac{B}{f_b}$$

E_b/N_0 = relación de densidad de potencia de energía por bit a ruido

C/N = relación de potencia de portadora a ruido

B/f_b = relación del ancho de banda de ruido a la tasa de bits

4.3.4. Calculo de la relación señal Ruido (SNR)

Un sistema de comunicaciones tiene como mayor amenaza al ruido debido a que todos los componentes electrónicos generan ruido térmico, una simple resistencia produce variaciones de su valor en función del calor disipado del orden de microvoltios, además se introduce el ruido de la red eléctrica mal regulada, esta tensión medible y amplificable por las sucesivas etapas del amplificador es el ruido de fondo, por lo que en un sistema ideal la SNR debería ser muy baja, pero en la práctica esto no sucede.

La SNR se calcula como la diferencia entre el nivel de la señal cuando el aparato funciona a nivel nominal de trabajo y el nivel de ruido cuando, a ese mismo nivel de trabajo, cuando no se introduce señal.

Según estudios se ha observado que cada bit adicional aumenta la razón señal-ruido en $6dB$. La fórmula exacta da una SNR de aproximadamente $98dB$ para 16 bits y una SNR de $146dB$ para 24 bits, la misma se calcula aplicando la Ec. 4.5.

$$SNR = 20 \log_{10} \frac{x_{rms}}{e_{rms}} \quad \text{Ecuación 4. 5}$$

dónde el valor RMS es la raíz cuadrada de la potencia promedio de la señal.

Denotando la potencia de la señal con P_x y la potencia del error con P_e podemos, aplicando una propiedad básica de los logaritmos, describir como:

$$SNR = 10 \log_{10} \frac{P_x}{P_e}$$

La potencia del ruido de cuantización está dada por la varianza de la variable aleatoria continua asociada, y resulta de aplicar la Ec. 4.6 así:

$$P_e = \int_{-q/2}^{q/2} \frac{1}{q} e^2 de = \frac{q^2}{12} \quad \text{Ecuación 4.6}$$

La medición de potencia a través del analizador de espectros arroja que la potencia de la salida de señal es 8 dBm, pero mediante la utilización de un amplificador de potencia se logró alcanzar los 33 dB, lo que es aproximadamente 2 *watios*, en este caso $P_x=2$, así el cálculo sería el siguiente:

$$SNR = 10 \log_{10} \frac{24}{q^2}$$

Se debe considerar que un computador no representa números usando dígitos decimales, sino dígitos binarios, si con b bits se representan $2b$ números distintos y el rango binario de -1 y 1 , se representan 2^{b-1} niveles de cuantización debajo de cero y 2^{b-1} arriba de cero, de esto se considera que el tamaño del escalón en función de la Ec. 4.7 detallada a continuación:

$$q = \frac{1}{2^{b-1}} \quad \text{Ecuación 4.7}$$

Finalmente la SNR sería:

$$SNR = 10 \log_{10} (24 * 2)^{2b-2}$$

y para un cálculo rápido y aproximado:

$$SNR \approx 6 \cdot b \text{ si son 32 bits es } 6*32 \text{ semejante a } 192 \text{ dB}$$

Esto se pudo comprobar mediante la medición realizada utilizando el bloque de medición SNR que dispone *Simulink*, en el cual simulando que el proceso funcione con una canal de ruido gaussiano blanco con una relación E_b/N_0 en el rango de 1 a 10 dB y se obtuvo un resultado de 183.1 dB, en promedio.

Al realizar el cálculo con 16 bits teóricamente se debe obtener una respuesta en el orden de los 90-100 dB y en la práctica utilizando el bloque *Simulink* se obtuvo respuestas en el orden de los 40 dB.

4.4. Comparación del BER de los modelos con modelación digital

4.4.1. Sistemas con Códigos Convolucionales y modulación QPSK

Seguidamente, y mediante el uso de la herramienta *BERtool* de *Matlab*, se pondrán a prueba los sistemas diseñados, con la finalidad de ofrecer un aporte significativo al comparar el cálculo de la cantidad de *bits* erróneos, en función del aumento de la relación E_b/N_0 . Éste procedimiento se lo realiza tomando las siguientes consideraciones:

- Codificador de Canal: Convolutacional.
- Modulador: QPSK
- Canal: AWGN
- Demodulador: QPSK
- Decodificador: Viterbi

Tomando en cuenta lo citado previamente, las gráficas obtenidas son las siguientes:

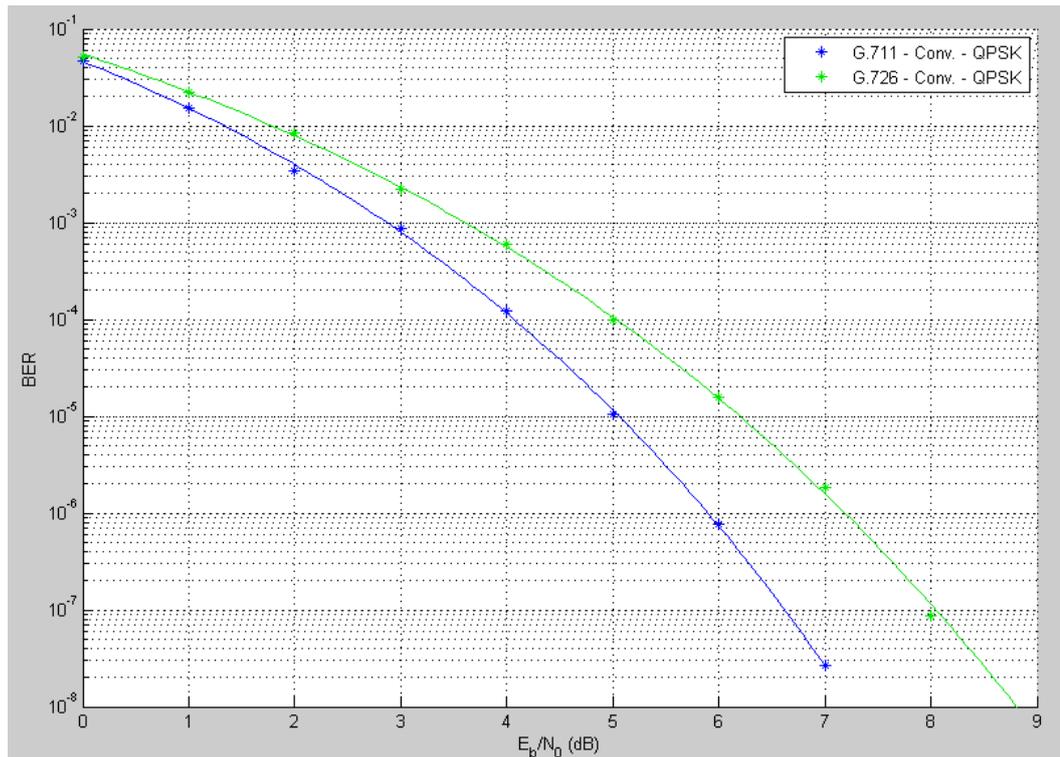


Figura 4. 10. Gráfica comparativa del BER de los modelos I y III, con código Convolutivo.

Como se puede observar en la Fig. 4.10, los sistemas diseñados, son similares. Esto quiere decir, que bajo las características de los modelos 1 y 3, son modelos que bajo ciertas características, mantienen un rendimiento aceptable. Como es posible verificar, con una señal con relación $\frac{E_b}{N_0} \geq 6dB$, es muy seguro que la información tendrá un *bit* erróneo por cada millón de *bits* transmitidos.

La razón por la cual esto es cierto, es debido a que tanto el códec *G.711* como el códec *G.726*, tienen características de adaptabilidad de la señal recibida, en consecuencia, y como ya se citó, la probabilidad de errores disminuye.

Por otro lado, es también importante percatarse de que el modelo que utiliza el códec *G.711*, tiene la capacidad de cometer menos errores, sobre las mismas características, que el códec *G.726*. Para aclarar éste punto, es necesario recordar que el códec implementado dentro del presente proyecto de tesis, utiliza la mitad de bits que los que utiliza el códec *G.711*. Dicho de

otra manera, la probabilidad de que un dato, conformado por los 4 *bits* que envía el códec G.726, aumentará, ya que un bit erróneo en 8 *bits* no es tan significativo como si lo es sobre 4 *bits*. Sin embargo la diferencia no es tan pronunciada.

4.4.2.Sistemas con Códigos de Hamming y modulación QPSK

Tal y como se indicó en el inciso anterior, y tomando en consideración el presente caso, se ha procedido a simular los diseños con las siguientes características:

- Codificador de Canal: Hamming
- Modulador: QPSK
- Canal: AWGN
- Demodulador: QPSK
- Decodificador: Hamming

Como es posible interpretar, las características son muy similares, con la gran diferencia de que en el presente caso, el código de Hamming realiza la detección de hasta dos errores y con la posibilidad de corregir uno de ellos, por lo tanto, la probabilidad de error aumenta.

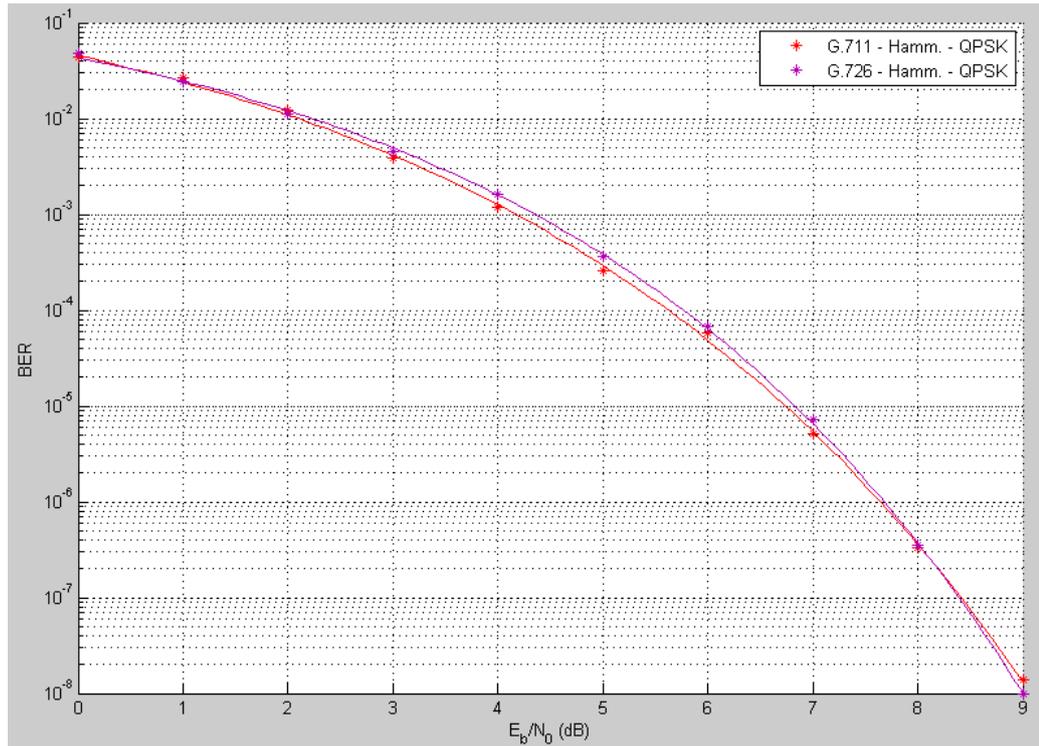


Figura 4. 11. Gráfica comparativa del BER de los modelos II y IV, con código de Hamming.

Bajo las mismas características, y tal como indica la Fig. 4.11, ambos sistemas son muy similares, es decir, que si en un determinado sistema de comunicaciones se decidiese usar el código de Hamming como codificador de canal, no importará el codificador de fuente que se elija. Sin embargo, algo que queda claro es que sería obvio escoger el códec G.726, ya que representa la misma calidad de voz pero a la mitad de bits transmitidos por muestra.

4.4.3. Comparación entre los cuatro modelos que usan modulación QPSK

Para dar un contraste claro en esta sección, se ha procedido a simular los cuatro sistemas a fin de encontrar el sistema con menor cantidad de errores sobre cualquier canal de comunicación, con lo cual se obtuvo los siguientes resultados:

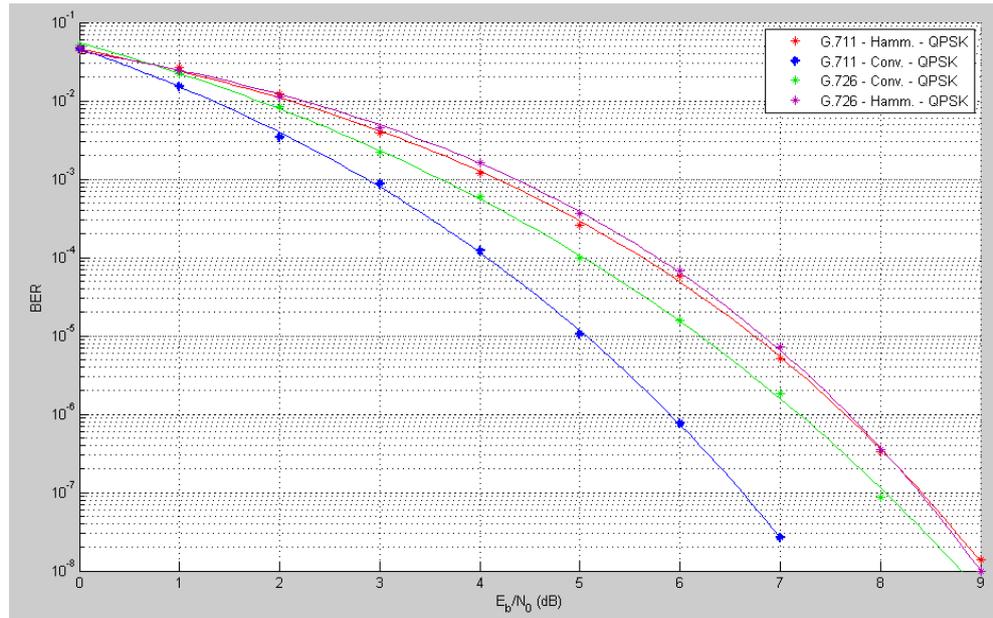


Figura 4. 12. Gráfica comparativa del BER de los sistemas diseñados.

En la Fig. 4.12, se puede constatar que el mejor sistema para implementar sería el modelo I, debido a que es más inmune al ruido, además la tangibilidad de la voz se conserva de buena manera durante el procesamiento de la señal, presenta además una tasa de error mucho menor a los demás modelos; lo que permite considerarlo como uno de los modelos más óptimos a utilizarse.

4.5. Implementación de los modelos diseñados

Como ya se pudo observar, gracias al uso de las simulaciones provistas en *Matlab*, en la sección previa, los sistemas que tendrán mejor desempeño, debido a la necesidad de una potencia menor, y debido a que la potencia de salida de las tarjetas es limitada, son los modelos I y III. Por ello dichos modelos fueron implementados, al igual que los demás, y consecuentemente fueron los que mejor desempeño práctico presentaron en la recepción de la señal.

4.5.1. Materiales

Los materiales que fueron utilizados para el desarrollo del presente proyecto fueron:

- 1 Portátil *Dell* modelo *Inspiron N4010*
 - Procesador: Core i3 M380 @ 2.53GHz
 - Sistema Operativo: Windows 7 (64 bits)
 - Disco duro: 640GB
 - Memoria RAM: 6GB

- 1 Portátil *Sony* VAIO PCG 61315L
 - Procesador: Core i3 M370 @ 2.4GHz
 - Sistema Operativo: Windows 7 (64 bits)
 - Disco duro: 320GB
 - Memoria RAM: 4GB

- 2 Tarjetas USRP v1.0

- 1 Tarjeta daughterboard LFTX (30MHz)

- 1 Tarjeta daughterboard LFRX (30MHz)

- 1 Tarjeta daughterboard BasicTX (1 - 250MHz)

- 1 Tarjeta daughterboard BasicRX (1 - 250MHz)

- 1 Analizador de Espectros Agilent N1996A-506

4.5.2. Diseños Implementados con el uso de los módulos USRP de Simulink

Para proceder a implementar los diseños, y debido a las características y necesidades de trabajo en tiempo real, fue necesario optimizar en especial el

códec *G.726*, y que esté en la capacidad de trabajar en tiempo real. Para lograr dicho objetivo, *Simulink* posee una herramienta denominada *Real Time Workshop*, que permite optimizar cualquier sistema de bloques llevando el sistema a código C (ver Anexo C).



Figura 4. 13. Montaje del sistema de comunicaciones.

El sistema de comunicaciones implementado, tal como se indica en la Fig. 4.13, consta de dos computadores portátiles, dos tarjetas *USRP* con sus respectivas *daughterboards*.

Por otro lado, el sistema implementado, se divide en un transmisor y un receptor, tal y como lo ilustran las Fig. 4.14 y Fig. 4.15, respectivamente.

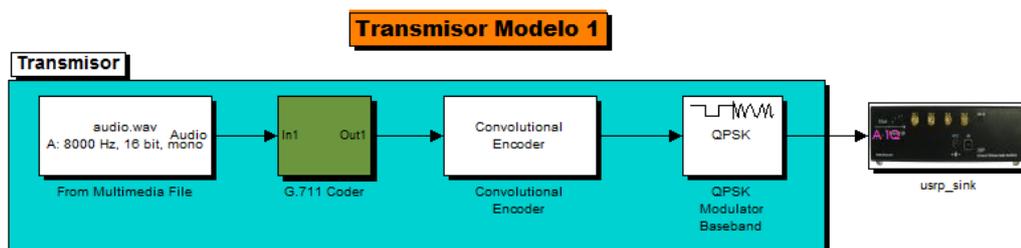


Figura 4. 14. Implementación del transmisor del Modelo I.

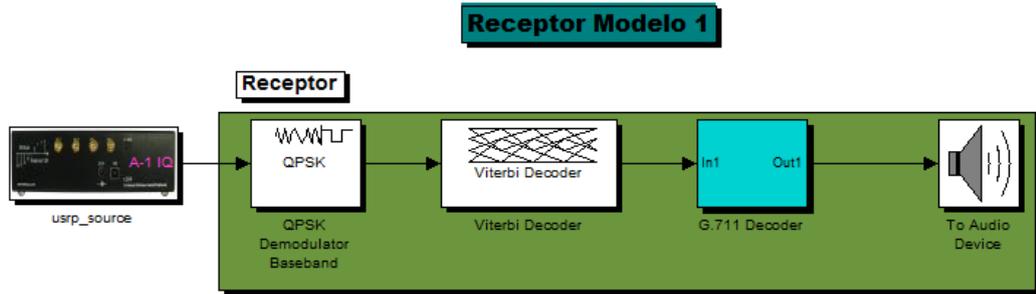


Figura 4. 15. Implementación del receptor del Modelo I.

4.6. Pruebas Experimentales

4.6.1. Resultados de las pruebas Experimentales en el Analizador de Espectros Agilent N1996A-506

En ésta sección se presentan las mediciones realizadas a los modelos más representativos, que fueron implementados y en los cuales fue posible la recepción de la señal previamente transmitida.

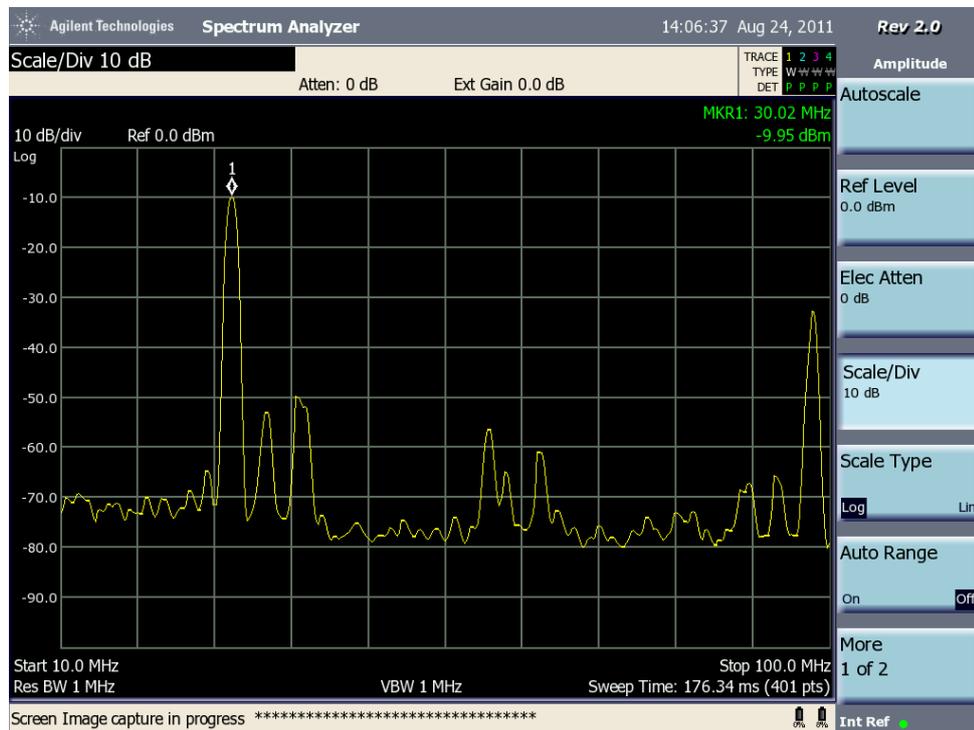


Figura 4. 16. Potencia de transmisión con Modelo I.

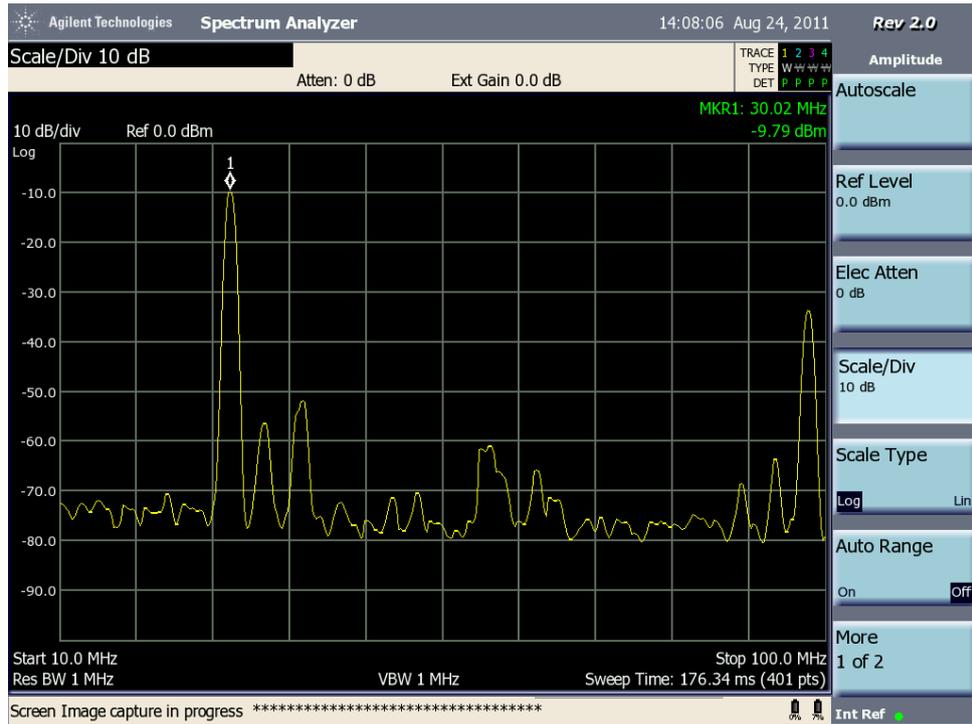


Figura 4. 17. Potencia de transmisión con Modelo III.

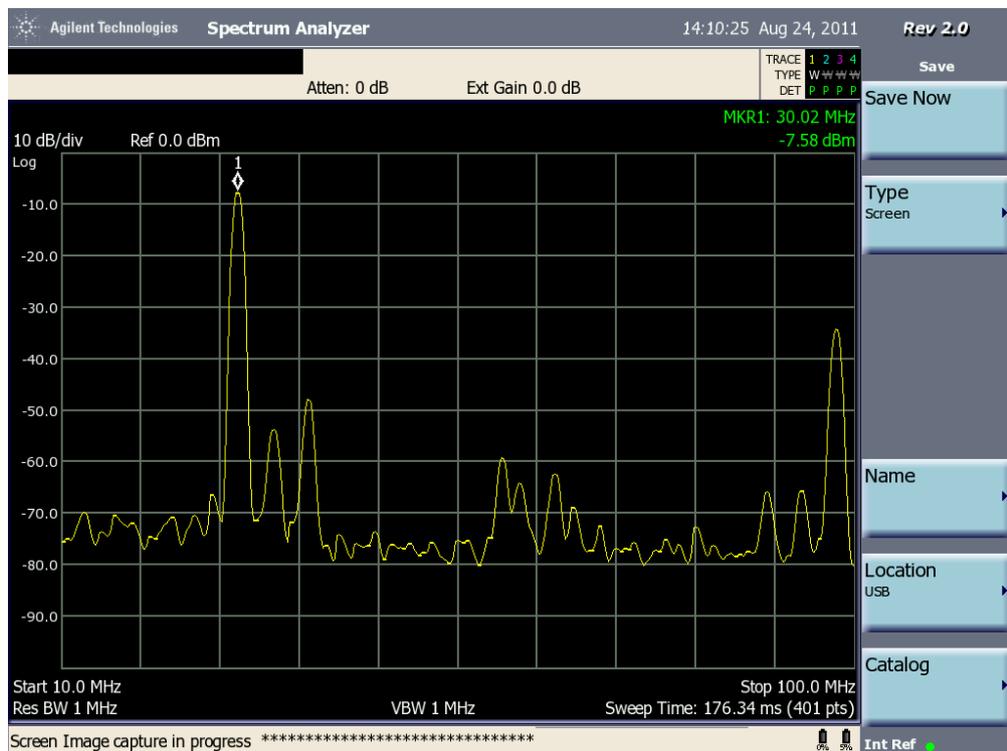


Figura 4. 18. Potencia de transmisión con Modelo V.

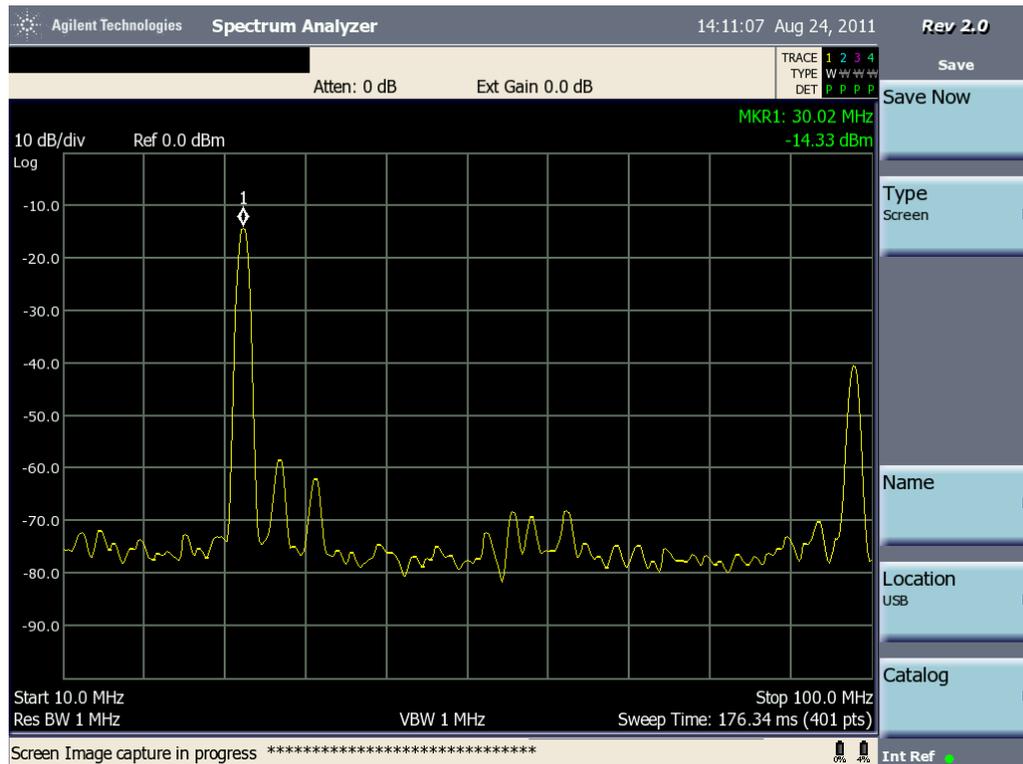


Figura 4. 19. Potencia de transmisión con Modelo VII.

Como es posible observar en la Fig. 4.19, se obtuvo el peor resultado en potencia de transmisión. Esto se debe esencialmente a que el modelo VII utiliza modulación en frecuencia por lo tanto la potencia de transmisión máxima es la indicada. En contraste con esto, las Fig. 4.16, Fig. 4.17 y Fig. 4.18, son muy similares, sin embargo la calidad en la recepción de la señal difiere en cuanto se usa el codificador convolucional, ya que demuestra ser muy superior que cuando se utiliza el codificador de *Hamming*, por lo tanto y en consecuencia, en la sección siguiente se demuestra lo que se acaba de citar.

4.7. Comparación de Resultados

En seguida se presenta una tabla comparativa de los sistemas implementados y las mediciones realizadas.

La valoración de la tangibilidad de la voz, está hecha en base a la nitidez de la voz considerando una escala: Buena, Mala y Ruido. Esto según si: se reconoce

la señal sin problemas, se reconoce la señal con dificultades y únicamente se escucha ruido.

Tabla 4. 1. Tabla comparativa de los modelos propuestos.

	Relación Señal Ruido [dB]	Tangibilidad de la voz.	Potencia de salida [dBm]	Tasa de transmisión [KSps]
Modelo I	55	Buena	- 8	64
Modelo II	60	Mala	- 10	56
Modelo III	55	Buena	- 10	56
Modelo IV	48	Mala	- 9	28
Modelo V	115	Buena	- 7	16
Modelo VI	110	Ruido	- 10	14
Modelo VII	120	Buena	- 10	8
Modelo VIII	115	Ruido	- 10	7

En éste punto puede afirmarse que el modelo I es el de mejor desempeño, por los argumentos detallados en la sección 4.4.3 y en vínculo con las mediciones de potencia realizadas con el analizador de espectros Agilent N1996A-506, por lo tanto, tomando en consideración todos estas mediciones se concluye que los sistemas que presentan mejor desempeño son el modelo I y el modelo V.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Los codificadores de forma de onda, como los *PCM* o *G.7XX*, realizan el proceso de codificación, sin tomar en cuenta la naturaleza de la señal de entrada, ya que únicamente se considera la forma de la misma, por lo que son muy eficientes al trabajar con tasas de bits altas como 32Kbps, en tanto que resultan ineficientes al trabajar con tasas de bits muy bajas, para dicho caso se utilizan los Vocoders (*LPC: Linear Predictive Code*) o Híbridos (*RELP: Residual Exited Linear Prediction*).
- Una ventaja de los sistemas digitales sobre los análogos es su capacidad de detectar y corregir errores, aunque esta última sea limitada. El tratamiento digital es conveniente para optimizar los sistemas de comunicación. Sobre esto, es posible afirmar que, para digitalizar por completo un sistema de comunicaciones, se debe tomar en consideración la sección de RF, ya que al intentar adaptar una sección analógica como es la mencionada, se pueden inducir errores que irán en detrimento de la señal a transmitir, por lo cual, un acoplamiento ideal debe realizarse por medio de una interfaz que permita el ensamble entre ambas secciones.
- Es importante mencionar que el códec *G.726*, implementado dentro del presente proyecto, permitió verificar que el proceso de codificación de la voz y posterior reconstrucción, es de mejor calidad, en comparación con el proceso realizado por el códec *G.711*. En función de esto, fue posible transmitir la señal al espacio y obtener, en la recepción, una calidad de la

voz aceptable. Por lo tanto se concluye que debido al ahorro que presenta el códec *G.726*, es más eficiente que el códec *G.711*.

- Un codificador convolucional, puede ser decodificado con un decodificador de probabilidad *a posteriori* o con *Viterbi*. Para este último, se debe considerar que el *traceback depth* o la cantidad de ceros que se debe insertar para compensar el desplazamiento provocado en la codificación, se debe analizar en función de la *trellis* utilizada y de la tasa de *bits*, caso contrario la información arriba en forma desordenada y la decodificación producirá demasiados errores. En conclusión, y de manera general, el valor del *traceback depth*, debe ser el mismo que el número de *bits* por muestra usados en la codificación, ya que de esta manera el decodificador desechará la primera muestra para sincronizarse con las siguientes.
- El objetivo de la codificación de canal es detectar y corregir errores, para de esta forma reducir la probabilidad de error necesaria para garantizar que la información que llegue al destinatario sea lo más semejante a la enviada. En función de lo citado, se pudo comprobar de forma práctica que una señal, aún con ruido inducido por el canal, o debido al procesamiento realizado por las tarjetas *USRP*, en éste caso, se puede transmitir y el decodificador corrige varios de estos errores, en tanto que una señal que no ha recibido una codificación de canal y durante el proceso, adopta errores, en la recepción no se podrá mejorar o corregir estos errores y el resultado será una señal deteriorada o una señal completamente ruidosa.
- La tarjeta *daughterboard BasicTX*, a pesar de que en su especificación esta detallado que la banda de trabajo va de 1 a 250MHz, en la práctica presenta problemas para operar en toda la banda y se comprobó que el rango real de funcionamiento es de 1 a 44MHz, y es el caso en el que la tarjeta propaga adecuadamente las señales al espacio.
- Conforme a lo experimentado en la presente investigación, se afirma que en un sistema de comunicaciones digital con modulación en banda base, optimiza el uso del ancho de banda. Para aclarar esto, se toma en

consideración el caso de la modulación QPSK, en donde el uso del ancho de banda necesario para la transmisión fue de la mitad de lo esperado. Esto se debe a que hace uso de 2 *bits* para cada una de los símbolos de la constelación, como fue expuesto en el CAPÍTULO II. Por lo tanto, y de manera general, se concluye que en una modulación *M*-PSK, el número de veces a fraccionar el ancho de banda necesario para transmitir una señal digital, estará determinado directamente por el número de *bits* utilizados para dicha modulación.

- El *USRP v.1* no tiene la capacidad de garantizar una buena comunicación utilizando cualquier modulación en banda base, ya que en ésta se requiere que la fase llegue de forma exacta al receptor, para que éste pueda decodificar la información. Como se puede interpretar, las tarjetas no tienen el aforo suficiente para mantener la fase de la señal, ya que debido a la transformación de digital a analógico y viceversa, hace que la ésta varíe de forma abrupta en el proceso aludido que genera el *FPGA* en conjunto con los conversores del *USRP*. Por lo tanto se concluye que las tarjetas no funcionan bien en procesos con modulación digital.
- Debido a los inconvenientes suscitados en las tarjetas *USRP v1.0*, como ya se citó, y a pesar de que de forma teórica se demostró que, de los modelos propuestos, el modelo I presenta un mejor desempeño y mayor robustez al necesitar poca potencia para obtener resultados aceptables, el modelo V fue, en la sección práctica, el de mejor desempeño, ya que al transmitir señales de audio, de manera general, permitió una recepción tangible y aceptable de la misma. Con ello se confirma claramente que la transición entre la sección digital y analógica es fundamental, ya que las tarjetas *USRP v1.0*, realizan un tratamiento aceptable de las señales analógicas, mas con las señales digitales, presentan inconvenientes.
- Con la finalidad de que el *USRP* no emita mensajes de *Ou* (*overrun*) y *uU* (*underrun*), se debe evitar que la información que pasa a través del *USB* supere el límite de 32 MBps. Esto se optimiza mediante el manejo de la interpolación y decimación en el transmisor y receptor respectivamente, ya

que al incrementar estos valores se reduce el embotellamiento de la información a través del puerto *USB* y se elimina la probabilidad del surgimiento de *overruns* y *underruns*, que indican desincronización entre las tarjetas y el computador.

- El trato analógico de las señales, como por ejemplo una modulación de FM, ayudó en gran medida para que la recepción, mediante el uso de las tarjetas *USRP v1.0*, de la señal sea constante. Sin embargo, y a pesar de todos los esfuerzos aunados en el desarrollo del proyecto, el desempeño de las mismas no es tan robusto. En función de lo mencionado, se puede afirmar que el desarrollo de cualquier proyecto, usando las tarjetas *USRP v1.0*, debe enfocarse en el trato analógico de las señales, ya que de esta manera se puede asegurar que la recepción no tendrá tantos inconvenientes a la hora de reproducir la información que ha sido transmitida.
- Los modelos diseñados en la presente investigación, en función de las pruebas y mediciones realizadas, constituyen un aporte significativo en el ámbito de los sistemas de comunicación digital, mediante el uso de los radios definidos por *software*. En éste proyecto, pudo comprobarse que el salto de las comunicaciones hacia la nueva era digital, cada vez es más necesario; esto se fundamenta principalmente en la consecución del proyecto, y conjuntamente con éste, la versatilidad que presentan los sistemas de radio mencionados. Además y sobre todo, la confiabilidad que presentan a la hora de encriptar las señales a fin de dotarlas de seguridad, lo cual es conveniente en aplicaciones militares, especialmente.
- En cuanto a lo que atañe a la compresión de la voz, se debe tomar en consideración que, la codificación inicial, es decir, el paso entre la sección analógica de la voz, a una sección digital es realizada por el módulo del micrófono de *Simulink*, el mismo que utiliza *16bits* por muestra para dicha digitalización. A esto se debe añadir que, debido a la implementación del códec *ITU-G.726* de 15 niveles, se logró comprimir cada una de dichas

muestras hasta *4bits*, lo que supone una optimización del 75%, en lo referente al uso del ancho de banda.

5.2. Recomendaciones

- Para la instalación del driver *USB* interfaz *USRP- PC*, se debe considerar que si el equipo es de 64 bits, la implementación difiere de un equipo de 32 bits, cuyo driver es universal; en tanto que en *64bits*, el driver debe ser diseñado como se indicó en la presente investigación en el CAPÍTULO II.
- En la fase de transmisión-recepción debe considerarse como un requisito indispensable la potencia de la señal, porque a pesar de que esta información no detalla ningún manual ni librería del *USRP*, es fundamental radiar la potencia necesaria para garantizar la llegada de la información al receptor, debido a que el *USRP* está limitado en potencia. Esto implica que, para trabajar con una frecuencia ya sea HF, VHF y UHF se requiere adaptar un amplificador de potencia para mejorar la cobertura de la señal.
- En la configuración de *USRP v1.0*, se debe considerar que los factores de interpolación en transmisión y decimación en recepción, no deberán ser iguales, por el contrario el factor de decimación es la mitad del factor de interpolación, para que la tasa de muestreo se la misma y la tangibilidad o las características de la señal no se distorsionen.
- Se recomienda el uso de una etapa de filtrado de la señal, antes de la recepción de la misma, ya que como se puede constatar en las Fig. 4.16 a 4.19, los armónicos generan ruido sobre la información que se desea decodificar.
- Se recomienda que para trabajar con un alto volumen de datos en un diseño SDR de alto desempeño, se trabaje con el *USRP v.2.0* o con el modelo *N210*, cuya capacidad garantizará el adecuado procesamiento de los datos, así como también la sincronización requerida para una transmisión-recepción exitosa.

Bibliografía

- [1] Sitio web de Wireless Innovation Forum. [En línea] [Citado el: 14 de Marzo de 2011.] <http://www.wirelessinnovation.org>.
- [2] Tuttlebee, Walter. *Software Defined Radio: Origins, Drivers and International Perspectives*. West Sussex : John Wiley & Sons, 2002. 0470844647.
- [3] GNU radio. [En línea] GNR radio, 12 de junio de 2008. [Citado el: 10 de enero de 2011.] gnuradio.org/redmine/attachments/download/129.
- [4] Hamza, Firas Abbas. The USRP under 1.5X Magnifying Lens! GNU Radio. [En línea] 12 de Junio de 2008. [Citado el: 20 de Febrero de 2011.] http://gnuradio.org/redmine/attachments/129/USRP_Documentation.pdf.
- [5] Ettus. USRP User's and Developer's Guide. [En línea] [Citado el: 21 de enero de 2011.] www.olifantasia.com/gnuradio/USRP/files/USRP_guide.pdf.
- [6] Burns, Paul. *Software Defined Radio for 3G*. Norwood : Artech House, 2003. 1-58053-347-7.
- [7] Bard, John y Kovarik, Vincent J. *Software Defined Radio: The Software Communications Architecture*. Chichester : John Wiley & Sons, 2007. 978-0-470-86518-7.
- [8] Johnson, C. Richard y Sethares, William A. *Telecommunication breakdown concepts of communication transmitted via software defined radio*. Upper Saddle River : Prentice Hall, 2003. 0131430475.
- [9] Lyons, Richard. *Understanding Digital Signal Processing*. Segunda Edición, Prentice Hall, 2001. 0-201-63467-8.
- [10] Software Defined Radio: Challenges and Opportunities. Ulversø, Tore. 2010 : IEEE.
- [11] Tuttlebee, Walter. *Software Defined Radio: Enabling Technologies*. West Sussex : John Wiley & Sons, 2002. 0-470-84600-3.

- [12] Tomasi, Wayne. *Sistemas de Comunicaciones Electrónicas*. México : Prentice Hall, 2003.
- [13] Gunn, James E. Sitio web de Wireless Innovation Forum. [En línea] 11 de Marzo de 2011. [Citado el: 11 de Marzo de 2011.] http://data.memberclicks.com/site/sdf/tut-Market_Studies_Overview.pdf.
- [14] Total Telecom. SDR: Softly Does It. [En línea] Telecom, 12 de Marzo de 2011. [Citado el: 12 de Marzo de 2011.] http://data.memberclicks.com/site/sdf/tut-SDR_article.pdf.
- [15] Haykin, Simon. *Sistemas de Comunicacion*. México : LIMUSA S.A, 2001. Primera Edición.
- [16] Axelson, Jan. *USB Complete: The Developer's Guide*. Madison : Lakeview Research LLC, 2009. 978-1-931448-08-6.
- [17] Codificación de canal y Modulación. Vega, Constantino Perez. 2001.
- [18] Codificación de canal. [En línea] Universidad de Valladolid. [Citado el: 30 de marzo de 2011.] <http://www.isa.cie.uva.es/proyectos/codec/marco4.html>.
- [19] Proakis, John. *Digital Communications*. Cuarta. New York : McGraw-Hill, 2008. 0072321113.
- [20] Blake, Roy. *Sistemas Electrónicos de Comunicaciones*. Ontario : Thomson, 2004. 0-7668-2684-8.
- [21] Carlson, Bruce, Crilly, Paul y Rutledge, Janet. *Sistemas de Comunicación*. Cuarta. s.l. : McGraw Hill, 2007. 0-07-011127-8.
- [22] Stremmer, Ferrel. *Sistemas de Comunicación*. s.l. : Alfaomega, 1989. 958-50-0009-3.
- [23] Dorf, Richard y Svoboda, James. *Circuitos Eléctricos*. Sexta. México D.F. : Alfaomega, 2006. 970-15-1098-4.

[24] Hak5. Video: Pulse Code Modulation. Hak5. [En línea] Hak5, 2010. [Citado el: 30 de Abril de 2011.] http://hak5.org/category/episodes/season_8/page/2.