



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

**EXPANSION DE UN SISTEMA DE COMUNICACION
INALAMBRICO DE PORTADORA UNICA CON MODULACION
QPSK Y 8-PSK CON INCLUSION DE FUNCIONES DE CAPA DE
ACCESO, RED Y APLICACION PARA ENVIO Y RECEPCION DE
TEXTO ENCRIPTADO**

**EXAMEN COMPLEXIVO, COMPONENTE PRÁCTICO
INFORME DE PROYECTO**

Previa a la obtención del Título de:

MAGISTER EN TELECOMUNICACIONES

Presentado por:

Carlos E. Hanze Sánchez

GUAYAQUIL – ECUADOR

AÑO 2016

AGRADECIMIENTOS

A mi familia.

DEDICATORIA

A mi madre por su cariño, paciencia y apoyo incondicional en todo momento de mi vida. A mi padre por su ayuda y consejo.

TRIBUNAL DE SUSTENTACIÓN



Francisco Novillo Parales, Ph.D.
EVALUADOR

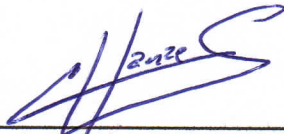


Néstor Arreaga Alvarado, Mg.
EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Informe de Proyecto, me corresponde exclusivamente; y el patrimonio intelectual del mismo, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL".

Art. 12 del Reglamento de Graduación.



Carlos E. Hanze Sánchez
C.I.: 0916836588

RESUMEN

Los sistemas de comunicación hacen uso desde hace décadas de modelos basados en capas, los ejemplos por antonomasia son el modelo TCP/IP y el modelo OSI, los cuales prácticamente están implementados en todo ámbito de comunicaciones eléctricas y ópticas sean a nivel local, metropolitano, de área extendida, o incluso remota en muchos casos. En este documento se expone la expansión de un sistema de comunicación inalámbrica de Portadora única que utiliza modulación QPSK y 8-PSK, a un sistema básico de capas que contempla las siguientes: aplicación, red, acceso y física propiamente dicha, para la transmisión y recepción de texto. El sistema incluirá funcionalidades a nivel de aplicación de encriptación de texto, direccionamiento y detección de errores a nivel de red, y un mecanismo de acceso al medio como CSMA y reconocimiento ACK. Para el diseño e implementación de este sistema, se hará uso del software LabVIEW y para una implementación real se utilizarán equipos basados en Software Defined Radio (SDR) como son los NI Universal Software Radio Peripheral (USRP)

TABLA DE CONTENIDO

AGRADECIMIENTOS	I
DEDICATORIA.....	II
TRIBUNAL DE SUSTENTACIÓN	III
DECLARACIÓN EXPRESA	IV
RESUMEN	V
TABLA DE CONTENIDO	VI
ABREVIATURAS	VIII
LISTA DE FIGURAS	IX
LISTA DE TABLAS	XI
INTRODUCCIÓN	1
CAPÍTULO 1	3
1. CONSIDERACIONES GENERALES DE DISEÑO.....	3
CAPÍTULO 2	4
2. ESPECIFICACIONES TÉCNICAS	4
2.1 Capa de Aplicación	4
2.2 Capa de Red	4
2.3 Capa de Acceso.....	5
2.4 Capa Física	6
CAPÍTULO 3	7
3. IMPLEMENTACIÓN	7
3.1 Capa de Aplicación	7
3.2 Capa de Red	11
3.3 Capa de Acceso.....	14
3.4 Capa Física	17
3.5 Canales Multipaso con Perfiles de Potencia Retraso	25
CAPÍTULO 4	29
4. PRUEBAS DE DESEMPEÑO	29
4.1 Ambiente para pruebas.....	29
4.2 Constelaciones y Diagrama de Ojo	30
4.3 Pruebas BER vs. SNR	37
4.4 Pruebas Tasa de bits (R_b) vs. SNR.....	40

CAPÍTULO 5	44
5. ANÁLISIS DE RESULTADOS	44
5.1 Comparativa de constelaciones y diagramas de ojo	44
5.2 Comparativa escenarios BER vs SNR.....	45
5.3 Comparativa escenarios Tasa bits vs. SNR	46
CONCLUSIONES	47
BIBLIOGRAFÍA	48
ANEXOS	51
ANEXO 1 - Perfiles de potencia retraso.....	51
ANEXO 2 – Detalle bloques subVI transmitter.vi adaptado a sistema.....	53
ANEXO 3 – Detalle bloques subVI receiver.vi adaptado a sistema.....	54
ANEXO 4 – Implementación cifrador RC4 en código LabVIEW	55
ANEXO 5 – Implementación 8-CRC CCITT en código LabVIEW.....	56
ANEXO 6 – Diagrama bloques implementación Transmisor sistema.....	57
ANEXO 7 – Diagrama bloques implementación Receptor sistema	59
ANEXO 8 – Implementación bloque Verifica Dirección Destino de Capa Red	60
ANEXO 9 – Implementación función Escalador Símbolos de Capa Física	61
ANEXO 10 – Implementación función Limitador Símbolos de Capa Física.....	62
ANEXO 11 – Implementación canal multipaso Interiores Residencial T3CA	63
ANEXO 12 – Implementación canal multipaso Exteriores Rural T4CA	64
ANEXO 13 – Implementación canal multipaso Exteriores Rural T4CB	65
ANEXO 14 – Interfaz de usuario Transmisor y Receptor sistema	66

ABREVIATURAS

QPSK	Quadrature Phase Shift Keying
8-PSK	Eight (8) Phase Shift Keying
TCP/IP	Transport Control Protocol / Internet Protocol
OSI	Open Systems Interconnection
SDR	Software Defined Radio
USRP	Universal Software Radio Peripheral
CSMA	Carrier Sense Media Access
ACK	Acknowledge
VI, subVI	Virtual Instrument, sub Virtual Instrument (LabVIEW)
BER	Bit Error Rate
SNR	Signal Noise Ratio
RC4	River Cipher 4
ASCII	American Standard Code for Information Interchange
MAC	Media Access Control
CCITT	Consultatif International Téléphonique et Télégraphique
CRC	Cyclic Redundancy Check
AWGN	Average White Gaussian Noise
KSA	Key Scheduling Algorithm
PRGA	Pseudo-Random Generation Algorithm
1D, 2D	1 Dimensión, 2 Dimensiones
RRC	Root Raised Cosine
ISI	Inter Symbol Interference
R_b	Bit Rate

LISTA DE FIGURAS

Figura 1. Estructura paquete capa red	5
Figura 2. Bloques funcionales capa aplicación Transmisor	8
Figura 3. Pseudo-código algoritmo KSA	8
Figura 4. Pseudo-código algoritmo PRGA	9
Figura 5. Bloques funcionales capa aplicación Receptor	10
Figura 6. Bloques funcionales capa red Transmisor	11
Figura 7. Bloques funcionales capa red Receptor	13
Figura 8. Diagrama función sensar Portadora capa de acceso Transmisor	15
Figura 9. Diagrama función Esperar mensaje ACK capa de acceso Transmisor	16
Figura 10. Diagrama función Enviar mensaje ACK capa de acceso Receptor	17
Figura 11. Diagrama Mapeador QPSK código nativo LabVIEW	19
Figura 12. Diagrama Mapeador 8-PSK código nativo LabVIEW	20
Figura 13. Diagrama Mapeador símbolos a bits QPSK y 8-PSK	24
Figura 14. Implementación Canal T3CB Interiores Residencial	28
Figura 15. Equipos y Conexiones ambiente pruebas	30

Figura 16. Constelación símbolos y diagrama ojo de sistema usando modulación QPSK	31
Figura 19. Constelación símbolos y diagrama ojo sistema usando QPSK y equipos USRP con efectos de canales multipaso	33
Figura 18. Constelación símbolos y diagrama ojo de sistema usando modulación 8-PSK	34
Figura 19. Constelación símbolos y diagrama ojo sistema usando 8-PSK y equipos USRP con efectos de canales multipaso	36
Figura 20. Gráfica BER vs. SNR de sistema usando QPSK	38
Figura 21. Gráfica BER vs. SNR de sistema usando 8-PSK	39
Figura 22. Gráfica Tasa bits vs. SNR de sistema con USRP usando modulación QPSK	41
Figura 23. Gráfica Tasa bits vs. SNR de sistema con USRP usando modulación 8-PSK	43

LISTA DE TABLAS

Tabla 1. Parámetros transmisor	6
Tabla 2. Parámetros receptor	6
Tabla 3. Mapeo bits a símbolo QPSK	18
Tabla 4. Mapeo símbolo a bits 8-PSK	19
Tabla 5. Canal T3CA Interiores residencial	26
Tabla 6. Canal T3CB Interiores residencial	26
Tabla 7. Canal T4CA Exteriores rural	26
Tabla 8. Canal T4CB Exteriores rural	27

INTRODUCCIÓN

Cuando se considera un sistema de comunicación inalámbrico puramente en lo concerniente a transmisor, receptor y medio de transmisión para su análisis y estudio, normalmente se comienza con lo correspondiente a cuestiones inherentes a la parte física tales como modulación, codificación, frecuencias, ancho de banda, etc., así como elementos, métodos, procesos y procedimientos necesarios para que finalmente se logre hacer llegar desde un transmisor a un receptor la información, superando los medios y obstáculos (ruido, dispersión, atenuación, falta de sincronismo, retraso, etc.) de por medio. Cuestiones como el acceso al medio y direccionamiento también son parte del estudio de un sistema de esta naturaleza, así como la parte final que interactúa con un usuario final.

Los sistemas de comunicación hacen uso desde hace décadas de modelos que colocan funciones como acceso al medio, red, transporte, presentación, aplicación en lo que se conoce como capas, los ejemplos por antonomasia son el modelo TCP/IP y el modelo OSI, los cuales prácticamente están implementados en todo ámbito de comunicaciones eléctricas y ópticas sean a nivel local, metropolitano, de área extendida, o incluso remota en muchos casos. Son parte crucial del diseño de la red Internet y son un referente al momento de idear o diseñar un sistema que, sea altamente complejo o relativamente sencillo, su objetivo es el mismo: hacer llegar información de cualquier tipo que la tecnología actual lo permita de un extremo al otro.

En este documento se expone la expansión de un sistema de comunicación inalámbrica de Portadora única que utiliza modulación QPSK y 8-PSK, a un sistema básico de capas que contempla las siguientes: aplicación, red, acceso y física propiamente dicha, para la transmisión y recepción de texto. Para el diseño e implementación de este sistema, se hará uso del software LabVIEW y equipos basados en Software Defined Radio (SDR) como son los NI Universal Software Radio Peripheral USRP.

También se considerará la implementación usando código nativo de LabVIEW a nivel de acceso al medio compartido de un esquema basado en protocolo CSMA (Carrier Sense Multiple Access) sin tomar en cuenta alguna funcionalidad de detección o prevención de colisiones, y reconocimiento de mensaje recibido ACK. A nivel de red se también implementará con código LabVIEW un procedimiento para detección de errores. A nivel de aplicación se construirá en los bloques y estructuras necesarios para la conversión de texto a una secuencia de bits; también se implementará la funcionalidad de encriptación de texto.

Finalmente se incluirán canales inalámbricos multipaso [3] [4] basados en perfiles [2] de potencia retraso (Power Delay Profile) específicos para ambientes interiores residenciales y ambientes exteriores rurales. Se implementarán versiones de los canales usando código nativo de LabVIEW.

Este documento incluirá gráficas de pruebas de desempeño del sistema implementado con versiones tipo simulación (solo software) y utilizando los equipos USRP para las modulaciones consideradas tales como constelaciones de símbolos y diagramas de ojo de la onda compleja. También incluirá comparativas de BER versus SNR y de Tasa de Bits (R_b) versus SNR para los casos del sistema en estado original (sin efectos de canal) como para un canal en específico.

CAPÍTULO 1

1. CONSIDERACIONES GENERALES DE DISEÑO

Se tienen las siguientes consideraciones generales para el diseño del sistema de comunicación:

- Funcionalidad similar a un beeper de texto o pager.
- La lógica del sistema para el receptor es que permanezca el modo de *escuchar* hasta que recibe un mensaje, entonces se detenga.
- La lógica del sistema para el transmisor es que envíe un mensaje, y se detenga.
- El sistema hará uso de un esquema basado en CSMA para acceso al medio compartido sin considerar detección o prevención de colisiones.
- El mensaje de texto se enviará en un solo paquete para no utilizar esquemas de fragmentación y re-ensamblaje de paquetes.
- La cantidad de caracteres enviados tendrá un límite inferior y superior.
- Una sola frecuencia portadora será utilizada para cualquier transmisión o recepción por parte de ambos extremos (esquema *half-duplex*)
- Se realizarán funciones de detección de errores a la información transmitida en el receptor.
- No se realizarán funciones de corrección de errores en el sistema.
- Los canales con perfiles de potencia retraso a ser usados serán ingresados como números complejos con representación rectangular ($a+jb$) Para esto, previamente mediante un subVI por cada canal, se convertirán los perfiles de potencia retraso a esa representación.

CAPÍTULO 2

2. ESPECIFICACIONES TÉCNICAS

2.1 Capa de Aplicación

La longitud del texto a ser enviado es mínimo de 1 carácter y máximo de 96 caracteres.

Se hará uso de encriptación de flujo de datos (stream cipher) mediante el algoritmo River Cipher 4 (RC4) [12]. Un cifrador de flujo de datos es un método que hace uso de una clave simétrica para la encriptación, en este cifrador el texto plano en formato de flujo o secuencia de dígitos (por ejemplo su equivalente decimal ASCII) son combinados con otro flujo de dígitos cifrados pseudo-aleatorio mediante una operación XOR a nivel de bits.

2.2 Capa de Red

Se hará uso de 8 bits de cabecera de paquete (Sync) de la forma 01010101. Las direcciones de red a usarse tendrán formato similar a una dirección MAC, 48 bits agrupados en secuencias de 12 números hexadecimales y visualmente separados por el símbolo ":" cada dos hexadecimales. El campo Payload tendrá una longitud mínima de 64 bits y máximo de 768 bits (96 bytes x 8 bit/byte) Se hará uso de un campo Longitud Payload que tendrá el valor en bits (16 bits) del texto final (Payload) a enviar.

La detección de errores se realizará mediante Cyclic Redundancy Check (CRC) con el polinomio CRC-8 CCITT (x^8+x^2+x+1) en representación normal 0x07. El campo CRC tendrá una longitud de 8 bits.

El paquete resultante de ensamblar los campos de cabecera Sync, direcciones de red origen y destino, longitud Payload, Payload propiamente dicho y CRC se presenta en la figura 1, y tendrá una longitud entre 192 y 896 bits.

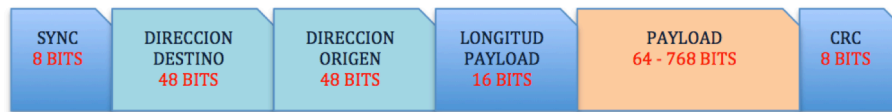


Figura 1. Estructura paquete capa red

2.3 Capa de Acceso

Esta capa realizará funciones básicas de CSMA (Carrier Sense Media Access) de sensar portadora usando método *1_persistent* [5], pero sin considerar detección o prevención de colisiones, y enviar confirmación ACK de mensaje recibido. El método *1_persistent* consiste en que la estación transmisora sensa la portadora o lo que es lo mismo, “escucha” el canal inalámbrico para saber si otra estación esta transmitiendo. Si la portadora está ocupada, sigue sensando o escuchando hasta que la encuentre libre y procede a enviar la información. En caso de colisión, este método espera un tiempo aleatorio y vuelve a sensar la portadora.

Con la funcionalidad de CSMA, el receptor una vez que reciba un mensaje y la capa de red verifique que en efecto es para él, enviará un ACK a su transmisor y detendrá su ejecución. Si el mensaje no es para él, la capa de red no pasa el mensaje a la capa de aplicación y la capa de acceso detiene la ejecución del receptor. Con la funcionalidad de CSMA, el transmisor una vez que haya enviado un mensaje, permanecerá en modo de “escuchar” hasta que se reciba un mensaje ACK por parte del receptor. Si la capa de acceso recibe un mensaje ACK antes de un *timeout* de 10 segundos, se presenta una alerta luminosa al usuario indicando que se recibió el mensaje. Caso contrario, una vez transcurridos los 10 segundos, la capa de acceso detendrá la ejecución del transmisor.

Esta capa no realizará funciones de detección de errores a nivel de tramas de bits.

2.4 Capa Física

Se utilizarán como base de implementación los subVI incluidos dentro del NI Digital Communications Bundle [16], los cuales fueron provistos con restricciones de acceso por contraseña a los diagramas de bloques en su mayoría. Las siguientes configuraciones que constan en la tabla 1 y tabla 2 serán utilizadas en transmisor y receptor:

Modulación digital	QPSK, 8-PSK
Filtro formador pulsos	Raíz Coseno Elevado de longitud 8 y factor rolloff $\alpha=0,5$
Tasa muestreo	10×10^6 muestras por segundo
Tiempo símbolo	100 ns
Ancho banda	40 MHz
Factor sobremuestreo	10
Secuencia entrenamiento insertada	2 secuencias código Barker de longitud 11 (22 símbolos)
Longitud secuencia símbolos Zero (1 + 0i)	8 cabecera y 8 cola
Frecuencia operación RF	934 MHz

Tabla 1. Parámetros transmisor

Demodulación digital	QPSK, 8-PSK
Filtro de acoplamiento	Raíz de Coseno Elevado de longitud 8 y con factor rolloff 0,5
Sincronización tiempo símbolo	Método directo de Máxima Energía
Detección desplazamiento trama	Correlación de ventana deslizante
Detección desplazamiento frecuencia	Algoritmo Moose
Ecuación	método de estimación Lineal Least Square Estimation con criterio de elección Mean Square Error
Ancho banda	40 MHz
Factor sobremuestreo	10
Frecuencia operación RF	934 MHz
Tiempo captura RF	2 milisegundos

Tabla 2. Parámetros receptor

Los siguientes procesos se implementarán en código nativo de LabVIEW: modulación digital (mapeo), demodulación digital (mapeo inverso), sincronización de símbolo, detección y sincronización de desplazamiento de trama y frecuencia, ecualización y remoción de secuencias de símbolos zeros ($1+j0$) y de entrenamiento. Una secuencia de entrenamiento es una secuencia conocida de símbolos complejos que se inserta en una señal a ser transmitida para que puedan realizarse operaciones en el receptor tales como correlación o comparación de símbolos en la señal recibida.

El filtro formador de pulsos y el filtro de acoplamiento hará uso de subVI propios de LabVIEW con esas funcionalidades y adicionalmente se usará el subVI con función de convolución.

Se modificará el subVI top_rx.vi para implementar las funcionalidades de sensar portadora en el transmisor. El envío de mensaje ACK por parte del receptor hará uso del subVI top_tx.vi sin modificaciones.

Adicionalmente, se implementarán versiones propias de subVI para los canales de los perfiles de potencia retraso. Se utilizará solamente el subVI TX_apply_channel.vi del transmisor para aplicar un canal de ruido blanco (AWGN)

CAPÍTULO 3

3. IMPLEMENTACIÓN

3.1 Capa de Aplicación

Transmisor

Tiene las funciones de ingreso de texto a ser enviado, configurar dirección de red del transmisor y dirección de red del receptor al cual va a ser enviado el mensaje; así como el cifrado y descifrado del texto usando encriptación RC4.

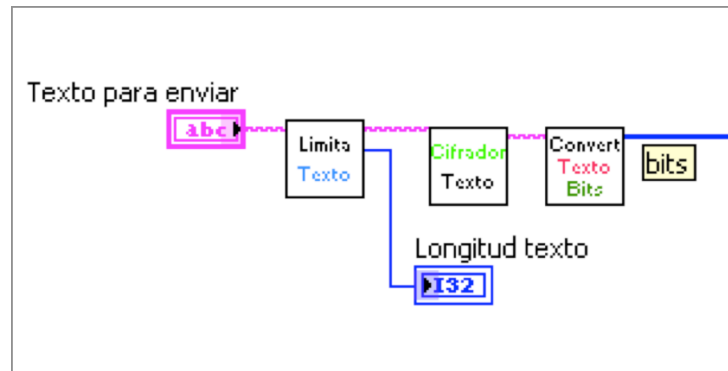


Figura 2. Bloques funcionales capa aplicación Transmisor

Consta de los siguientes bloques funcionales, los cuales se presentan en la figura 2 y se detallan a continuación:

Limita longitud texto, restringe el ingreso de caracteres a una longitud entre 1 y 96 emitiendo cuadros de alerta en caso de no cumplirse con las condiciones. No permite se ejecuten las siguientes partes del sistema hasta que se cumplan las condiciones.

Cifrador texto, encripta el texto ingresado usando el cifrador de flujo de datos RC4 [12]. Esta encriptación, la cual tiene como entradas el texto a ser cifrado (texto plano) y una clave simétrica de longitud entre 1 y 256 bytes, consta de 3 etapas definidas. La primera etapa hace uso de un algoritmo llamado KSA (Key Scheduling Algorithm) cuyo código se presenta en la figura 3:

```

1 for i from 0 to 255
2 S[i] := i
3 Endfor
4 j := 0
5 for i from 0 to 255
6 j := (j + S[i] + key[i]) mod 256
7 swap values of S[i] and S[j]
8 endfor

```

Figura 3. Pseudo-código algoritmo KSA

Este algoritmo inicialmente crea una secuencia $S[n]$ del 1 al 256 (permutación identidad) mediante un lazo For (línea 1 a 3), luego inicializa el índice $j=0$ (línea

4), y finalmente mediante otro lazo For se van cambiando los valores del índice j y se realiza la permutación de todos los elementos del arreglo $S[n]$ (línea 5 a 8). Al final se obtiene una nueva secuencia que incluye la permutación con todos los 256 posibles valores de $S[n]$. El algoritmo KSA implementado [12] en código nativo LabVIEW contiene una modificación tomada de [13] (se usó **key[i]** en lugar de **key[i mod keylength]**)

La segunda etapa denominada Pseudo-Random Generation Algorithm (PRGA) cuyo código se presenta en la figura 4:

```

1 i := 0
2 j := 0
3 while GeneratingOutput:
4 i := (i + 1) mod 256
5 j := (j + S[i]) mod 256
6 swap values of S[i] and S[j]
7 K := S[(S[i] + S[j]) mod 256]
8 output K
9 endwhile

```

Figura 4. Pseudo-código algoritmo PRGA

Este algoritmo [12], el cual fue implementado en código nativo LabVIEW, toma la nueva secuencia $S[n]$ obtenida al final del algoritmo KSA, procede luego a inicializar los dos índices $i=0$ y $j=0$ a ser usados (líneas 1 y 2), a continuación se inicia un lazo While (líneas 3 a 9) en donde en cada iteración a) se cambian los valores de ambos índices i y j ; b) se realiza la permutación de los valores del arreglo $S[n]$; c) se van creando los valores del arreglo K con variaciones del índice del arreglo $S[n]$ y; d) se obtiene a la salida el valor del arreglo K para la iteración actual. Al final del algoritmo PRGA se obtiene una secuencia pseudo-aleatoria $K[n]$.

La tercera etapa es la aplicación de la encriptación, que consiste en aplicar la operación XOR entre el texto plano a ser encriptado, y la secuencia pseudo-aleatoria generada por RC4. En la implementación de este algoritmo, hay que convertir tanto el texto a encriptar como la secuencia pseudo-aleatoria en secuencias de valores booleanos para poder efectuar la operación XOR.

La clave simétrica usada en este proyecto es de 256 bytes y fue obtenida usando un generador de claves [15], sin embargo podría haberse usado cualquier secuencia de caracteres alfanuméricos entre 1 y 256 caracteres de longitud para este algoritmo. El detalle del algoritmo RC4 implementado se encuentra en el anexo 4.

Texto convierte bits, utiliza una adaptación de la solución encontrada en [8] que convierte cada carácter a su equivalente decimal ASCII, luego convierte estos números a un arreglo de varias dimensiones N de valores booleanos (N es la longitud del texto ingresado), posteriormente se convierte a un arreglo 1D de valores booleanos usando la función *Reshape Array*, y finalmente convierte los valores booleanos en una secuencia (arreglo 1D) de valores 1 y 0.

Receptor

Tiene las funciones de recepción de texto recibido y dirección de red del transmisor que envió el texto, así como configurar la dirección de red del receptor. Consta de los siguientes bloques funcionales, los cuales se presentan en la figura 5 y se detallan a continuación:

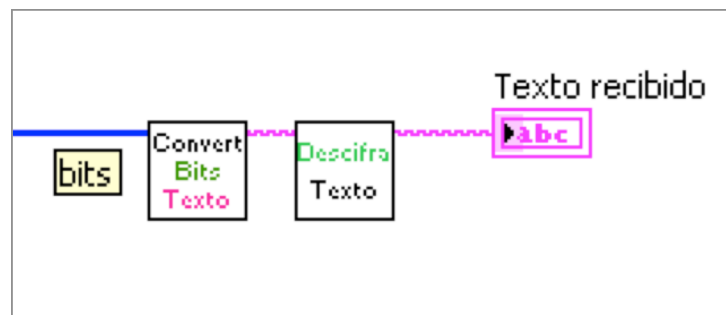


Figura 5. Bloques funcionales capa aplicación Receptor

Bits convierte texto, utiliza una adaptación de la sugerencia encontrada en [8] que convierte los bits a un arreglo 1D de valores booleanos, luego se convierte a un arreglo de varias dimensiones N de valores booleanos ($N = longitud_{bits}/8$) usando la función *Reshape Array*, posteriormente se convierten a números decimales que son valores ASCII, y finalmente todos los decimales ASCII a caracteres de texto.

Descifrador, descifra texto usando cifrado RC4. Se utiliza el mismo bloque del transmisor, solo que en este caso las entradas del bloque RC4 son a) el texto recibido a ser descifrado y; b) la misma clave simétrica usada para encriptación en el transmisor.

3.2 Capa de Red

Transmisor

Realiza la función de pre-ensamblar las tramas de bits del texto junto con las tramas de bits de las direcciones de red (pre-trama), hacer un cálculo de CRC (Cyclic Redundancy Check) a la pre-trama y ensamblar la trama definitiva tal como se detalla en la figura 6:

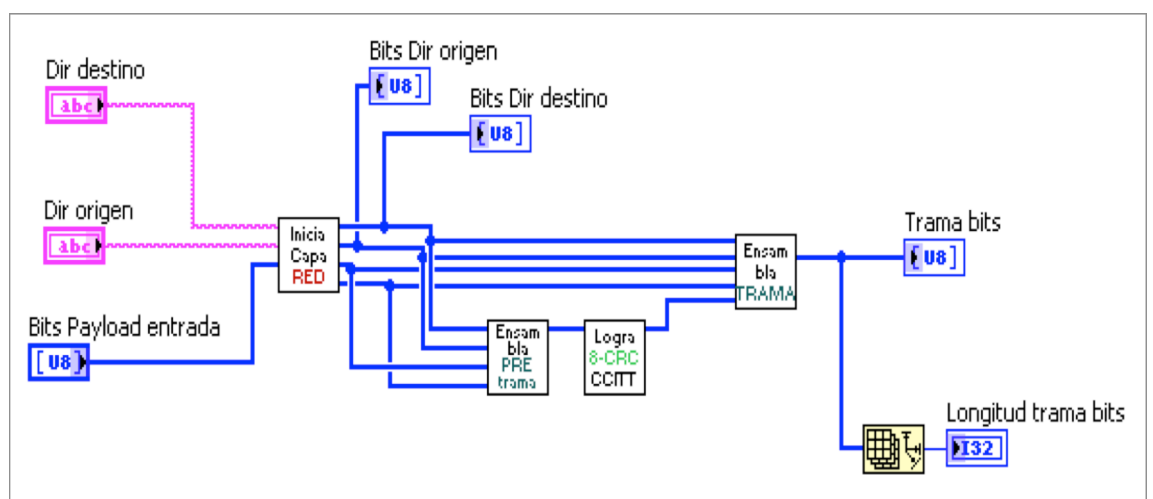


Figura 6. Bloques funcionales capa red Transmisor

Esta capa fue implementada como un subVI denominado **TX capa red.vi** que contiene los siguientes bloques funcionales:

Inicia capa red, convierte las direcciones de red que son texto (secuencia de números hexadecimales) a un arreglo 1D de números decimales usando la función *Hexadecimal String to Number*. Luego convierte este arreglo numérico en un arreglo de varias dimensiones n de valores booleanos ($n=12$ es la

longitud del arreglo numérico y por ende del texto de la dirección de red sin separadores) Posteriormente se convierte a un arreglo 1D de valores booleanos usando la función *Reshape Array*, y finalmente convierte los valores booleanos en una secuencia (arreglo 1D) de valores 1 y 0. Adicionalmente, convierte el valor de longitud de Payload en trama de bits.

Ensambla pre-trama, une las tramas de bits de las direcciones de red, longitud Payload y la trama de Payload propiamente dicha, previo al cálculo de CRC

Calcula CRC bits, aplica un procedimiento [11] implementado en código nativo de LabVIEW similar al que se realiza a nivel de flip-flops en electrónica digital:

- A la secuencia de bits se le añade 8 bits de valor 0 al final (trama aumentada) y es ingresada a una cola FIFO en donde al primer elemento de la cola se realiza la operación XOR con los elementos 6, 7 y 8 del arreglo de la trama aumentada y reemplazados.
- La cola sigue desplazando la secuencia de bits hasta que no quedan elementos de la trama aumentada que entren a la parte posterior de la cola.
- El resultado es una secuencia de 8 bits que servirá para detectar si hay errores en la pre-trama en el receptor (ver Anexo 5 detalle implementación)

Ensambla trama, une las tramas de bits de la secuencia de Sync (01010101), las direcciones de red, longitud Payload, la trama de Payload propiamente dicha y CRC, previo al paso a la capa física.

Receptor

Realiza la función de chequear el CRC del paquete, desensamblar el paquete en tramas correspondientes a la dirección de red destino, dirección de red origen y texto, y verificar si la dirección de red destino es igual a la dirección de red del receptor tal como se detalla en la figura 7:

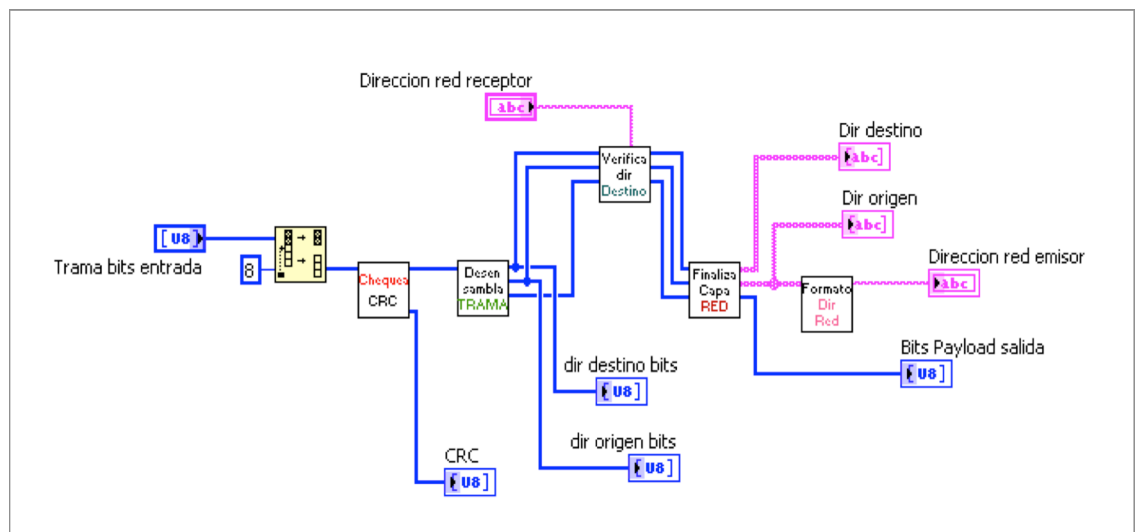


Figura 7. Bloques funcionales capa red Receptor

Esta capa fue implementada como un subVI denominado **RX capa red.vi** que contiene a su vez los siguientes bloques funcionales:

Chequea CRC trama, aplica el mismo procedimiento [11] implementado en código nativo de LabVIEW para calcular CRC en el transmisor. Sin embargo, a la secuencia de bits se le añaden los 8 bits del campo CRC al final (trama aumentada). El resultado de calcular CRC a esta trama aumentada debe ser CERO en caso de estar libre de errores para ser pasada a la capa de aplicación, caso contrario no pasa la trama a esa capa.

Desensambla trama, divide la trama removiendo los bits de Sync (01010101) y separando en tramas independientes para las direcciones de red origen y destino, longitud Payload y trama Payload propiamente dicha.

Verifica dirección destino, compara la trama de dirección de red destino con la dirección de red configurada en el receptor. Si son iguales, pasa la trama de bits de Payload a la capa de aplicación, caso contrario no la pasa a esa capa (ver Anexo 8 para detalle implementación)

Finaliza capa red, convierte cada una de las tramas de bits de las direcciones

de red origen y destino primero en arreglos 1D de valores booleanos, luego convierte a un arreglo de varias dimensiones N de valores booleanos (N longitud arreglo valores booleanos dividido para 4; 4 bits = 1 hexadecimal), posteriormente se convierte a un arreglo de números decimales y finalmente se convierten mediante la función *Number to Hexadecimal String* a un arreglo 1D de caracteres.

Formato dirección red, realiza el proceso de convertir el arreglo 1D de caracteres de la dirección de red al formato de 12 números hexadecimales separados por el símbolo ":" cada dos hexadecimales

3.3 Capa de Acceso

Transmisor

Realiza la función de detectar si la señal portadora se encuentra libre para la transmisión haciendo uso del USRP del transmisor en modo de recepción, utilizando para ello una versión modificada del subVI top_rx llamado **sensar_portadora.vi** dentro de un lazo While, se muestra su implementación en la figura 8:

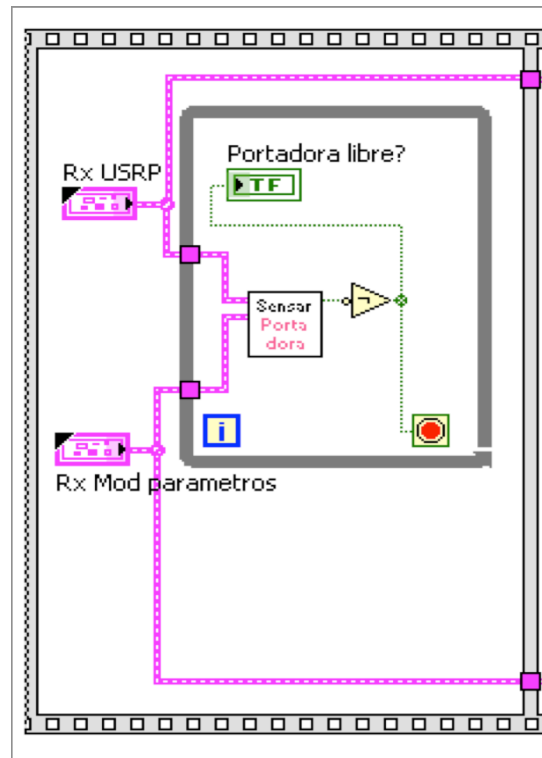


Figura 8. Diagrama función sensor Portadora capa de acceso Transmisor

Este subVI modificado tiene los bloques funcionales:

- RXRF_recv_sensarportadora.vi
- RXRF_trig_capt_sensarportadora.vi

Dentro de éste último se encuentra la funcionalidad de *Threshold Detector* que permite contar el numero de picos sobre un umbral que se producen en una secuencia (arreglo de valores). En este caso la secuencia es el conjunto de valores capturados por el USRP. El subVI *sensar_portadora.vi* genera una salida booleana de "Paquete encontrado". Si no se detecta al menos un pico de señal sobre el umbral establecido ($\text{peak count} > 0$) entonces el valor de "Paquete encontrado" será falso, el lazo While se detendrá y el transmisor procederá a transmitir, caso contrario el subVI *sensar_portadora.vi* seguirá sensando la portadora.

Una vez enviado un mensaje de texto, se activa la función de **esperar recepción ACK**. La recepción de un mensaje ACK pasa por las verificaciones

de CRC y dirección destino que hace la capa de red como consta en la figura 9:

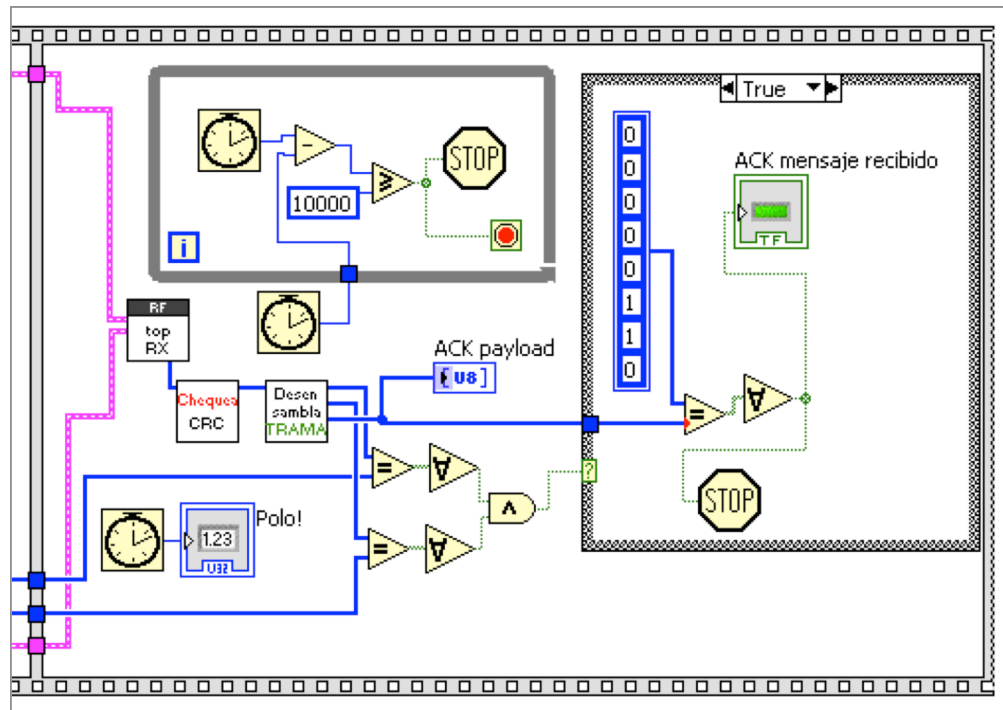


Figura 9. Diagrama función Esperar mensaje ACK capa de acceso Transmisor

El proceso de Timeout hace uso de la función *Tick Count* para tomar los tiempos antes y durante cada ejecución del lazo While, y revisar si la diferencia entre estos dos tiempos es de 10 segundos. En ambos casos, ya sea se reciba un ACK o el Timeout sea igual a 10 segundos, el programa detendrá (función STOP) la ejecución del transmisor.

Receptor

Realiza la función de enviar un **mensaje de confirmación ACK** al transmisor de mensaje recibido, se muestra un detalle en la figura 10:

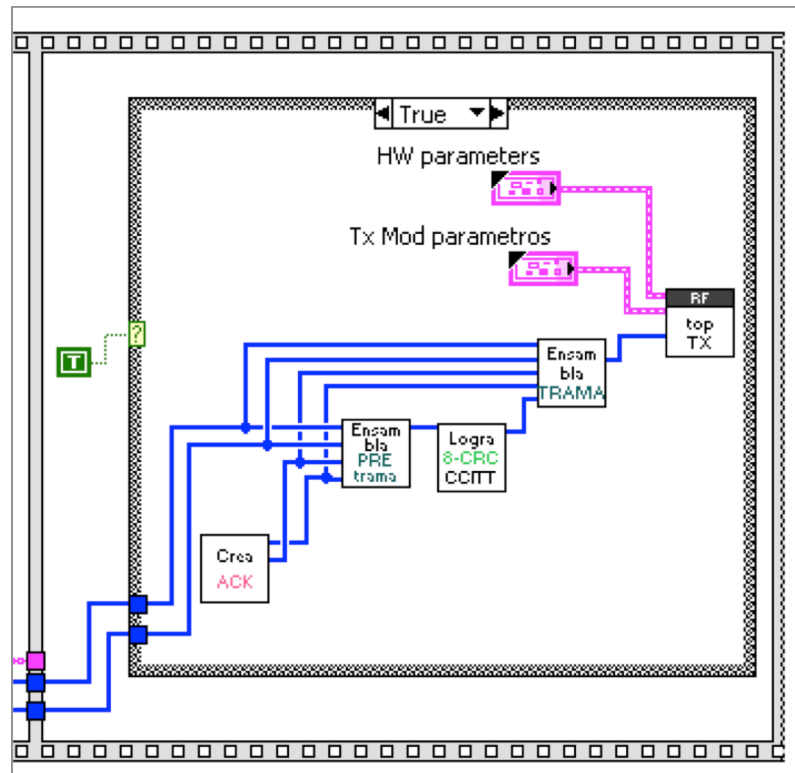


Figura 10. Diagrama función Enviar mensaje ACK capa de acceso Receptor

El envío de un mensaje ACK pasa por el proceso de encapsulación de capa de red para crear un paquete de red, es decir se añaden las direcciones de red origen y destino, longitud del mensaje ACK, el ACK propiamente dicho y el calculo CRC. Una vez construido el paquete es pasado a la capa física (top_tx.vi) para su envío.

3.4 Capa Física

Transmisor

Específicamente dentro el subVI predeterminado [1] transmitter.vi, el cual se muestra en detalle en el anexo 2, se implementaron los bloques con las funcionalidades de:

Mapeador de bits a símbolos para modulación 8-PSK y QPSK implementado en código nativo LabVIEW. En general la expresión para un símbolo M-PSK [14]

esta dada por la ecuación 1:

$$s_i(t) = \sqrt{\frac{2 \cdot E}{T}} \cos(2\pi f_0 t + \phi_i) \quad i = 1, \dots, M \quad (1)$$

Donde E es la energía de símbolo, T es el tiempo de símbolo, f_0 es la frecuencia y ϕ_i es el ángulo de fase que depende del valor de M (número símbolos) El ángulo de fase [14] toma los valores discretos de acuerdo a la ecuación 2:

$$\phi_i = \frac{(2i - N)\pi}{M} \quad i = 1, \dots, M; N \text{ entero } 0 \text{ o } 1 \quad (2)$$

Para el caso de QPSK con $M=4$ y $N=1$, los valores serán $\pi/4$, $3\pi/4$, $5\pi/4$ y $7\pi/4$. En un eje real imaginario, estos ángulos harán que la modulación equivalga a los valores $1+j$, $-1+j$, $-1-j$ y $1-j$. La opción QPSK del sistema convierte una secuencia de 2 bits en un símbolo complejo de acuerdo al mapeo que se encuentra en la tabla 3:

BITS	SIMBOLOS
00	$1 + j$
01	$1 - j$
10	$-1 + j$
11	$-1 - j$

Tabla 3. Mapeo Bits a Símbolos QPSK

En la figura 11 se presenta la implementación del Mapeador de bits a símbolos QPSK para el mapeo específico de los bits 11 al símbolo $-1-j$:

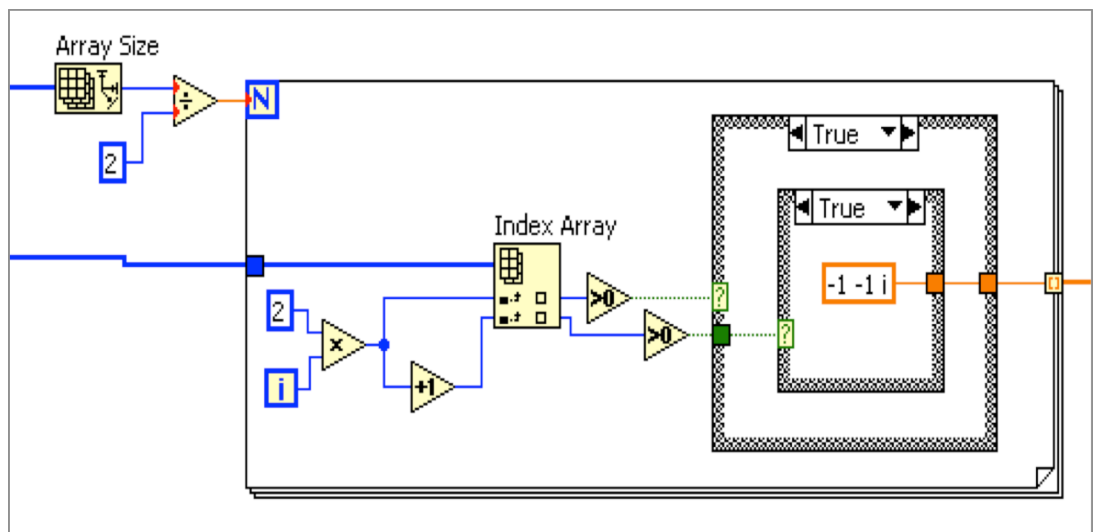


Figura 11. Diagrama Mapeador QPSK mapeo bits 11 a símbolo -1-j

Para 8-PSK con $M=8$ y $N=0$, los valores serán $\pi/4$, $\pi/2$, $3\pi/4$, π , $5\pi/4$, $3\pi/2$, $7\pi/4$ y 2π . En un eje real imaginario, estos ángulos harán que la modulación equivalga a los valores $\sqrt{2}/2+j\sqrt{2}/2$, $0+j$, $\sqrt{2}/2+j\sqrt{2}/2$, $-1+j0$, $-\sqrt{2}/2-j\sqrt{2}/2$, $0-j$, $\sqrt{2}/2-j\sqrt{2}/2$ y $1+j0$. La opción 8-PSK del sistema convierte una secuencia de 3 bits en un símbolo complejo de acuerdo al siguiente mapeo que se encuentra en la tabla 4:

BITS	SIMBOLOS
000	0,7071 + j 0,7071
001	1 + j 0
010	0 - j
011	0,7071 - j 0,7071
100	0 + j
101	-0,7071 + j 0,7071
110	-0,7071 - j 0,7071
111	-1 + j 0

Tabla 4. Mapeo Bits a Símbolos 8-PSK

El valor de 0,7071 es una aproximación a 4 decimales del número irracional $\sqrt{2}/2$. En la figura 12 se presenta la implementación del Mapeador de bits a

símbolos 8-PSK para el mapeo específico de los bits 000 al símbolo $\sqrt{2}/2 + j\sqrt{2}/2$:

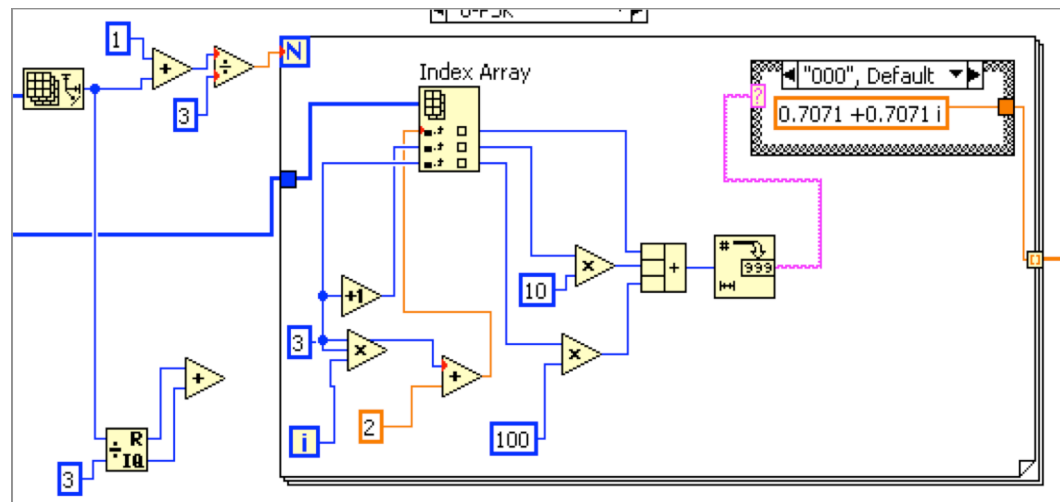


Figura 12. Diagrama Mapeador 8-PSK mapeo bits 000 a símbolo $\sqrt{2}/2 + j\sqrt{2}/2$

Filtro Formador de Pulsos que hace uso de la función *MT Generate Filter Coefficients* el cual genera el filtro $g_{Tx}(t)$ basado en la función *Root Raise Cosine* (RRC) que será convolucionada con los símbolos complejos a la salida del *Mapeador de Bits a Símbolo*, con el fin de darle forma a los pulsos analógicos a ser enviados al USRP para ser transmitidos. La ecuación que describe en tiempo discreto [1] esta convolución es:

$$c[n] = \sum_{m=-\infty}^{\infty} s[m] \cdot g_{Tx}[n - mL] \quad (3)$$

Donde $s[m]$ es el símbolo complejo del *Mapeador de Bits a Símbolos*, L es el factor de upsampling, y $c[n]$ es la salida del filtro formador de pulsos.

El bloque del filtro formador de pulsos también incluye la función de **Interpolación** (upsampling) [1] de los símbolos previo al filtrado.

Transmisión a través del USRP, el subVI **top_tx.vi** es el transmisor propiamente dicho que realiza la comunicación con el USRP. Posee dos opciones de operación en su panel de control: "one-shot" y "simultaneous" [1].

Se configuró como permanente (valor por defecto) la opción de “one-shot” para poder implementar la lógica del transmisor descrita en las consideraciones generales de diseño.

Receptor

Específicamente en el subVI receiver.vi, el cual se muestra en detalle en el anexo 3, se implementaron los bloques con las funcionalidades de:

Filtro de Acoplamiento que hace uso de la función *MT Generate Filter Coefficients* el cual genera el filtro $g_{Rx}(t)$ basado en la función Root Raise Cosine (RRC) que será convolucionada con los valores de la onda compleja recibida, con el fin de maximizar el SNR de la señal recibida. La ecuación 4 describe en tiempo discreto esta convolución:

$$y'[n] = \sum_m g_{Rx}[m] \cdot z[n - m] \quad (4)$$

donde $y'[n]$ es la salida convolucionada del filtro y $z[n]$ es la señal discreta a la salida del USRP.

Sincronización Tiempo de Símbolo usando el algoritmo del método directo [1] de Máxima Energía implementado en código nativo LabVIEW. Este algoritmo está basado en las ecuaciones 5 y 6:

$$J_{aprox}[k] = \frac{1}{P} \sum_{p=0}^{P-1} |r(p \cdot M \cdot T + k)|^2 \quad (5)$$

$$\hat{k} = \mathbf{max}_k J_{aprox}[k] \quad (6)$$

Donde J_{aprox} es la función que aproxima la función $J[k] = E|r(nMT + k)|^2$ que define la máxima energía en tiempo discreto, $r[n]$ es la salida del filtro de acoplamiento, M es el factor sobremuestreo, P es el número de valores de $r[n]$, T es el tiempo de símbolo.

Decimación (downsampling) [1]

Detección y corrección de desplazamiento de trama y frecuencia usando los algoritmos [1] de los métodos de correlación de ventana deslizante (ecuación 9) y algoritmo de Moose (ecuación 10):

$$R[n] = \left| \sum_{k=0}^{N_t-1} t^*[k] \cdot y[n+k] \right|^2 \quad (7)$$

$$\hat{f}_e = \frac{\text{fase} \left\{ \sum_{l=L}^{N_t-1} y[l+N_t] \cdot y^*[l] \right\}}{2\pi T N_t} \quad (8)$$

Donde $t[k]$ es la secuencia de entrenamiento insertada en la señal recibida, $y[n]$ es la salida del filtro de acoplamiento con decimación, N_t es la longitud de la secuencia entrenamiento y T es el tiempo de símbolo.

Cálculo de Ecuador usando algoritmo [1] del método Lineal Least Square Estimation con enfoque directo (calculo del ecuador a partir de la secuencia de entrenamiento) implementados usando la ecuación 9:

$$\mathbf{f}_{n_d} = (\mathbf{Y}^* \times \mathbf{Y})^{-1} \times \mathbf{Y}^* \times \mathbf{t} \quad (9)$$

Donde \mathbf{Y} es una matriz tipo Toeplitz (matriz con diagonales iguales $\forall a_{ij} \in \mathbf{Y}, a_{ij} = a_{(i+1)(j+1)}$) cuyas diagonales son valores de la señal compleja recibida por el receptor que han pasado por los procesos de filtrado de acoplamiento, decimación y corrección de trama y frecuencia si aplicara, y \mathbf{t} es una matriz de una sola columna cuyos valores son los de la secuencia de entrenamiento insertada en la señal recibida en el transmisor.

Aplicación de Ecuador basado en el criterio [1] del mínimo error cuadrático medio $\min J_f[n_d]$ implementado en código nativo LabVIEW usando las ecuaciones 10 y 11:

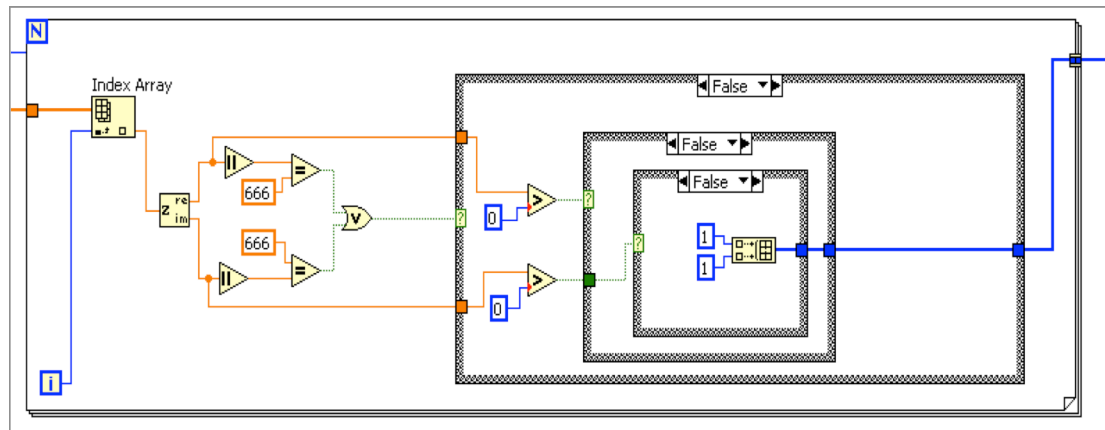
$$J_f[n_d] = \left\| \mathbf{t} - \mathbf{Y}_{n_d} \times \mathbf{f}_{n_d} \right\|^2 \quad (10)$$

$$\check{s}[n - n_d] = \sum_{l=0}^{L_f} f_{n_d}[n] \cdot y[n - l] \quad (11)$$

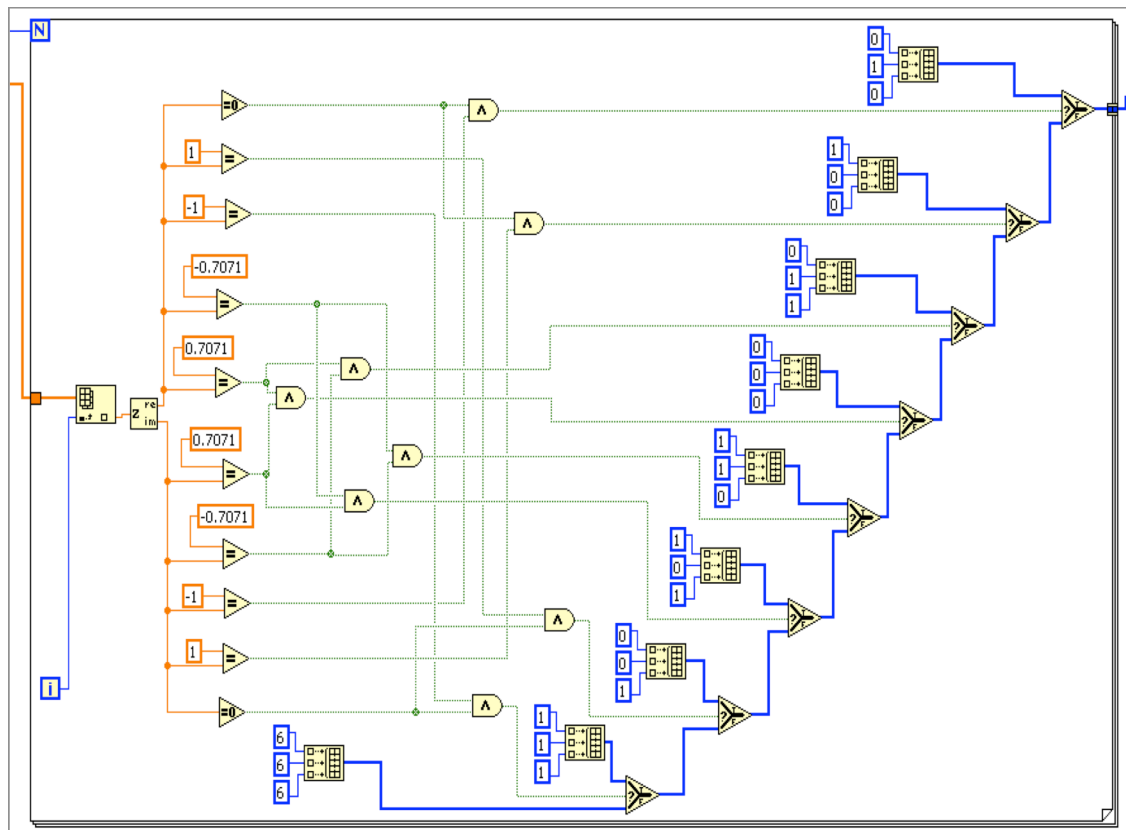
Donde n_d es el retraso de ecualizador, \mathbb{Y}_{n_d} es la matriz indicada antes pero con retraso n_d aplicado a la señal compleja recibida, \mathbb{t} ya fue definido anteriormente, \mathbb{f}_{n_d} y f_{n_d} son la matriz y la secuencia del ecualizador calculado respectivamente, $y[n]$ la señal recibida y $s[n]$ la salida del ecualizador.

Removedor de secuencia de entrenamiento, secuencia de zeros de cola y **limitador** de longitud del arreglo de símbolos recibido, con el fin de evitar que aparezcan bits innecesarios al momento de ser aplicadas las funcionalidad de detección de errores en la capa de red y se presenten falsos positivos. Este último bloque también incluye la función de **escalador de símbolos**, [6] la cual escala los valores de las partes real e imaginaria del símbolo recibido a valores aptos a ser manejados por la siguiente etapa de función de mapeo de símbolo a bit. En los anexos 9 y 10 se muestra la implementación de las funciones de este bloque.

Mapeador de símbolos a bits para la conversión inversa de las modulaciones QPSK y 8-PSK. Es el proceso inverso del Mapeador indicado en el transmisor. En la figura 13(a) se presenta la implementación del Mapeador de bits a símbolos QPSK para el mapeo específico del símbolo $-1-j$ a los bits 11. En la figura 13(b) se presenta la implementación del Mapeador de bits a símbolos 8-PSK para todos los mapeos posibles de manera inversa de símbolos a bits como aparece en la tabla 4:



(a)



(b)

Figura 13. Diagrama Mapeador símbolos a bits en código nativo LABVIEW. (a) QPSK. (b) 8-PSK

Recepción en el USRP, el subVI `top_rx.vi` es el receptor propiamente dicho que realiza la comunicación con el USRP. La lógica original del subVI de

recepción (top_rx.vi) de acuerdo a [1] es que top_rx.vi tiene dos modos de operación: escuchar y recibir. Sin embargo, en top_rx.vi se encuentra una implementación de *timeout* dentro del subVI llamado RXRF_trigger_capture.vi con un valor de 100ms dentro del lazo que tiene el detector de picos de señal. Esto último impide que top_rx.vi permanezca en modo de escucha mas allá de 100 ms. Se quitó este *timeout* para poder implementar la lógica del receptor descrita en las consideraciones generales de diseño. También se modificó dentro de este subVI en el bloque RXRF_init.vi las configuraciones *Reference Frequency Source* y *Timebase Clock Source* que afectan al USRP, de la opción "MIMO" a "Internal". Así cada USRP usa su propia fuente de sincronización, con el fin de evitar un error [9] que se presentaba aleatoriamente al ejecutarse el programa del sistema en LabVIEW. El error textualmente dice "*the local oscillator did not lock within the allotted time*" y está relacionado con el uso del cable MIMO de los USRP para compartir fuentes de referencia entre 2 equipos [10]. Una observación importante es que al tener referencias de sincronismo diferentes, es necesario que el receptor tenga activa la opción de detectar y corregir el desplazamiento de frecuencia *permanentemente*.

3.5 Canales Multipaso con Perfiles de Potencia Retraso

Un canal multipaso [3] es un modelo de tiempo discreto para la representación de canales de tiempo continuo multi-trayectoria que son variantes en el tiempo (canales que producen versiones atenuadas y retrasadas de la señal transmitida a la entrada del receptor), en donde la discretización del tiempo continuo se realiza utilizando el teorema de muestreo [3]. Un perfil de potencia retraso [2] de un canal multipaso es una representación grafica de la potencia relativa (en decibeles) de la señal recibida como una función de los retrasos (en segundos), estos últimos en referencia a una fuente fija de tiempo de retraso. Para el actual sistema, se implementaron los siguientes perfiles de potencia retraso proporcionados para realización de pruebas de desempeño:

a) El canal al cual denominaremos ISI T3CA es un canal multipaso cuyas

características corresponden a un ambiente de interiores (Indoor) tipo residencial con 2 multipaso, su perfil se detalla en la tabla 5:

Retraso (s)	Potencia (dB)
0	0
100×10^{-9}	-13,8

Tabla 5. Canal T3CA Interiores residencial

- b) El canal al cual denominaremos ISI T3CB es un canal multipaso cuyas características corresponden un ambiente de interiores (Indoor) tipo residencial con 4 multipaso, su perfil se detalla en la tabla 6:

Retraso (s)	Potencia (dB)
0	0
100×10^{-9}	-6
200×10^{-9}	-11,9
300×10^{-9}	-17,9

Tabla 6. Canal T3CB Interiores residencial

- c) El canal al cual denominaremos ISI T4CA es un canal multipaso cuyas características corresponden un ambiente de exteriores (Outdoor) tipo rural con 4 multipaso, su perfil se detalla en la tabla 7:

Retraso (s)	Potencia (dB)
0	0
200×10^{-9}	-2
400×10^{-9}	-10
600×10^{-9}	-20

Tabla 7. Canal T4CA Exteriores rural

- d) El canal al cual denominaremos ISI T4CB es un canal multipaso cuyas características corresponden un ambiente de exteriores (Outdoor) tipo rural con 6 multipaso, su perfil se detalla en la tabla 8:

Retraso (s)	Potencia (dB)
0	0
100×10^{-9}	-4
200×10^{-9}	-8
300×10^{-9}	-12
400×10^{-9}	-16
500×10^{-9}	-20

Tabla 8. Canal T4CB Exteriores rural

Como ejemplo, se muestra la implementación del perfil de potencia retraso de la tabla 6 (Interiores residencial) en código nativo de LabVIEW en la figura 14 en la siguiente página. En los anexos 11 al 13 se muestran las implementaciones de las tablas restantes.

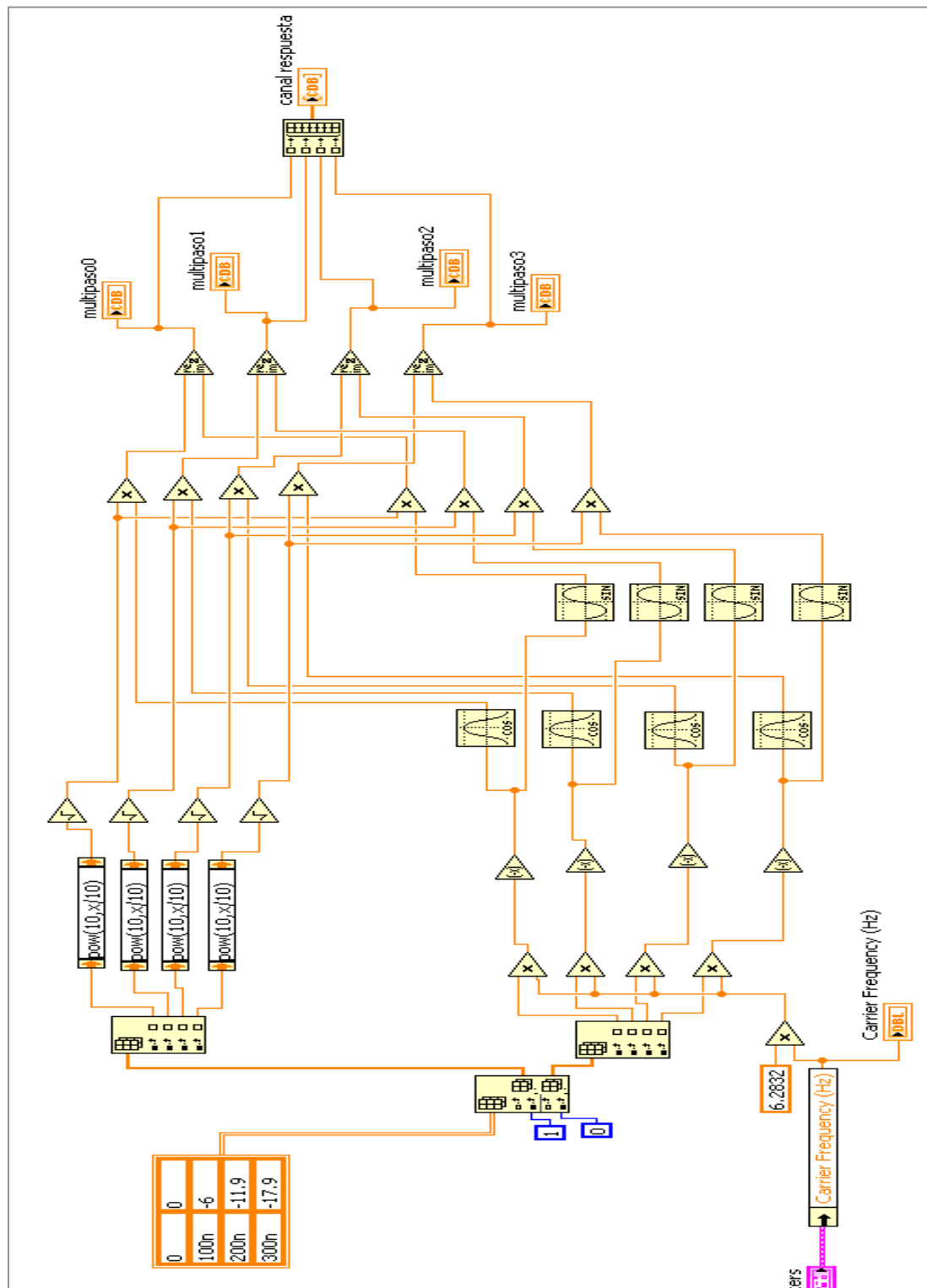


Figura 14. Implementación LabVIEW canal T3CB Interiores Residencial

Estos canales fueron incluidos en la parte del transmisor del sistema, específicamente en el bloque *Tx Apply Channel.vi* que se encuentra dentro de *transmitter.vi*. A manera de explicación de cómo opera el subVI del canal, se puede mencionar que la tabla del canal es un arreglo 2D que luego es separada en 2 arreglos 1D (potencias y retrasos) Para la aplicación del canal al sistema, se requiere que cada uno de los multipaso de las tablas sean convertidos a números complejos con representación rectangular ($a + j b$) Primero, se convierten las potencias en dB a unidades de watts usando la ecuación 12:

$$P = 10^{\frac{\text{Potencia [dB]}}{10}} \quad (12)$$

y luego a cada potencia se le aplica la raíz cuadrada \sqrt{P} . Los retrasos deben ser convertidos a unidades angulares en radianes mediante la ecuación 13:

$$\varphi = 2 \cdot \pi \cdot f \cdot \tau_k \quad (13)$$

Donde φ es el ángulo en radianes resultante, f es la frecuencia de la portadora del sistema y τ_k es el retraso del multipaso k de la tabla del canal. Una vez obtenidos los valores de potencia en watts y los retrasos convertidos en ángulos radianes, se tiene una forma polar del multipaso complejo $P \angle \varphi$. Par convertir a forma rectangular, aplicamos la expresión $P \cdot \cos \varphi + j P \cdot \sin \varphi$. Una vez obtenidos los valores complejos de cada multipaso, estos son ingresados en la parte del *Channel Response* del transmisor para aplicarlos a la señal a transmitirse.

CAPÍTULO 4

4. PRUEBAS DE DESEMPEÑO

4.1 Ambiente para pruebas

Se utilizó para las pruebas 2 equipos NI USRP modelo 2920, cada uno con una

antena omnidireccional; ambos equipos se interconectaron mediante un cable MIMO para ser gestionados mediante un solo computador, adicionalmente este cable permite que los USRP compartan fuentes de referencia [10] y además, que los subVI de transmisor y receptor usen una estructura de cola (queue) para compartir la secuencia de entrenamiento y la secuencia de bits transmitida al receptor para efectos de cálculo BER. El computador usado tiene instalado el software LabVIEW 2014. La conexión de red entre USRP y computador es Gigabit Ethernet. La figura 15 muestra el detalle del ambiente de pruebas. En capa de aplicación se hizo uso de envío de texto de longitud 96 caracteres con encriptación RC4 aplicada; la capa de red no influye en estas pruebas; y en capa de acceso no se hizo uso de las funciones de sensar portadora y envío/recepción ACK.

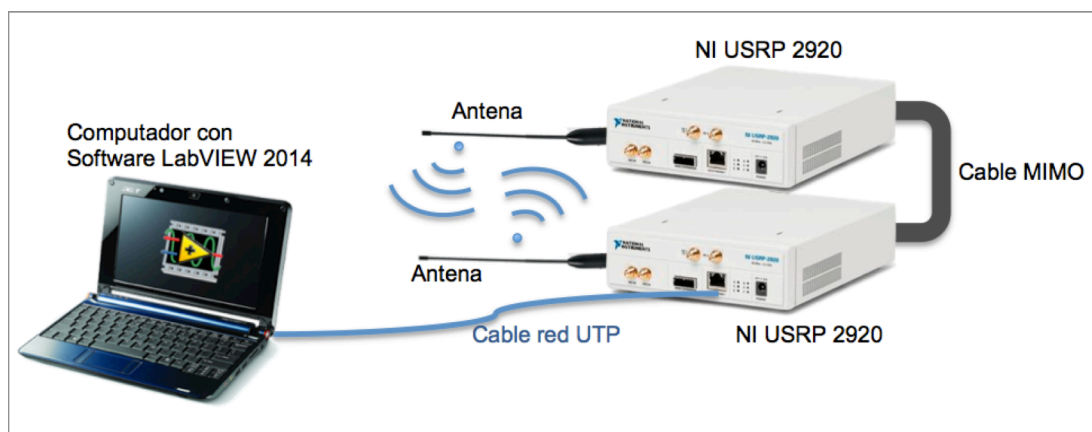


Figura 15. Equipos y conexiones ambiente pruebas

4.2 Constelaciones y Diagrama de Ojo

Se presentan las gráficas de constelaciones de símbolos y diagramas de ojo del sistema para visualizar el desempeño de ambas modulaciones QPSK y 8-PSK en los 4 escenarios de los canales de perfiles de potencia retraso:

Sistema con Modulación QPSK

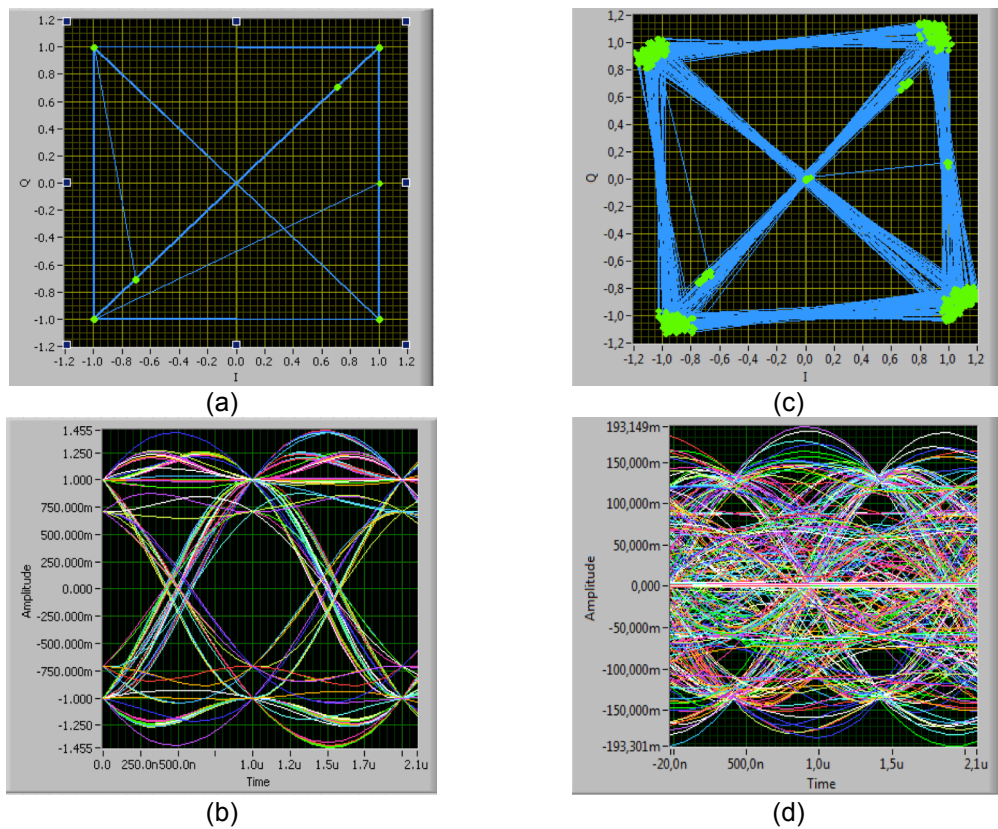


Figura 16. Constelación símbolos y diagrama ojo de sistema usando modulación QPSK. (a) simulación SIN equipos USRP; (b) simulación CON equipos USRP.

Las figuras 16(a) y 16(b) presentan la constelación de símbolos y diagrama de ojo de la señal recibida por el sistema en el receptor usando modulación QPSK sin hacer uso de los equipos USRP, es decir en condiciones ideales. Los puntos verdes en las 4 esquinas del cuadrado dentro de la figura 16(a) representan los símbolos del texto transmitido, los restantes puntos verdes pertenecen a los símbolos de la secuencia de entrenamiento y secuencia de símbolos $(1+j0)$ zeros. En la figura 16(b) se observan varias muestras (curvas) de la señal recibida, en donde se pueda apreciar la apertura del ojo en condiciones ideales. Las figuras 16(c) y 16(d) presentan en cambio presentan la constelación de símbolos y diagrama de ojo de la señal recibida por el sistema en el receptor haciendo uso de los equipos USRP. Puede observarse la dispersión de los

símbolos de la figura 16(c) debido al ruido presente en el enlace inalámbrico, así como una ligera rotación de la constelación, debido principalmente a que los 2 equipos usan sus propias referencias de frecuencia. Se está haciendo uso de corrección de desplazamiento de frecuencia tal como se indicó previamente en la parte de implementación de capa física del receptor. El diagrama de ojo permite apreciar los efectos del ruido y de la Interferencia Intersimbólica del canal inalámbrico real al no haber una apertura clara.

Sistema con Modulación QPSK y efectos de canales multipaso

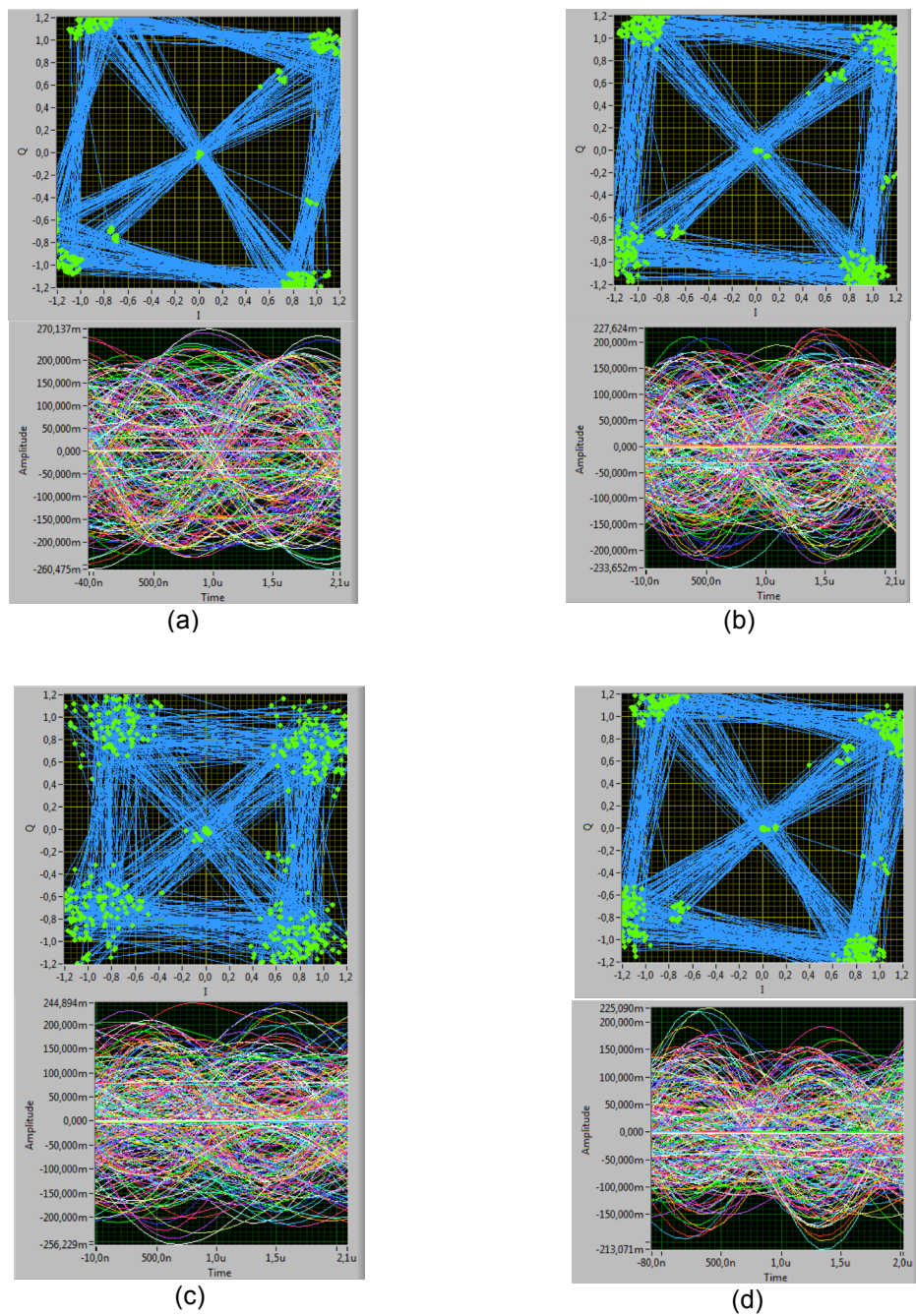


Figura 17. Constelación símbolos y diagrama ojo de sistema usando QPSK y equipos USRP con efectos de canales multipaso. (a) Canal T3CA; (b) Canal T3CB; (c) Canal T4CA; Canal T4CB.

En cada una de las figuras 17(a) a17(d) se muestran combinados la constelación y diagrama de ojo de la señal recibida por el sistema en el receptor con QPSK, pero con los efectos de los 4 canales multipaso cuyos perfiles de potencia retraso se mostraron en las tablas 5 a 8 en la parte de implementación. A cada prueba de un canal le fue aplicada una ecualización de longitud igual al numero de multipaso del canal, a excepción del canal T4CA (figura 17c), con el cual fue necesario usar un ecualizador de longitud 7. También se colocaron valores idénticos para cada canal en la opción de *Channel Estimate Length* para poder visualizar el multipaso en los perfiles de potencia retraso obtenidos y que constan en el anexo 1.

Sistema con Modulación 8-PSK

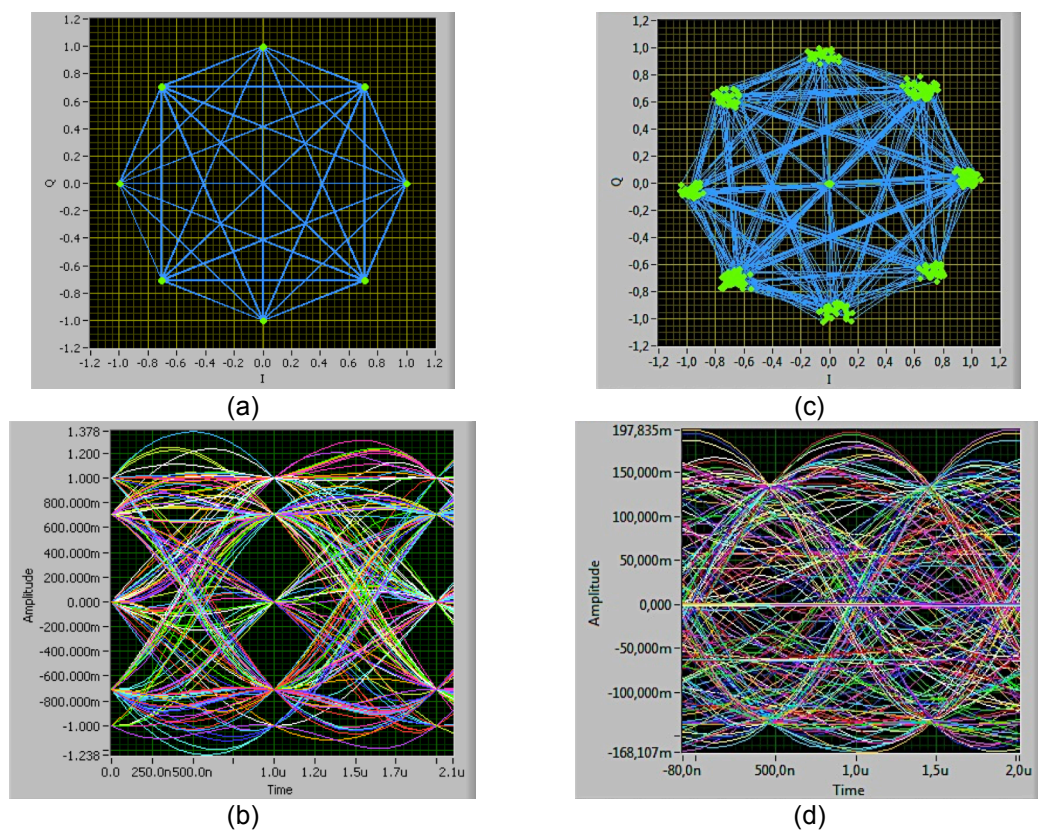


Figura 18. Constelación símbolos y diagrama ojo de sistema usando modulación 8-PSK. (a) simulación SIN equipos USRP; (b) simulación CON equipos USRP.

Las figuras 18(a) y 18(b) presentan la constelación de símbolos y diagrama de ojo de la señal recibida por el sistema en el receptor usando modulación 8-PSK sin hacer uso de los equipos USRP, es decir en condiciones ideales. Los puntos verdes en los 8 vértices del octágono dentro de la figura 18(a) representan los símbolos del texto transmitido; los puntos pertenecientes a la secuencia de entrenamiento y secuencia de símbolos zeros ($1+j0$) coinciden con los de la constelación. En la figura 18(b) se observan varias muestras (curvas) de la señal recibida, en donde se pueda apreciar la apertura del ojo en condiciones ideales. Las figuras 18(c) y 18(d) presentan en cambio presentan la constelación de símbolos y diagrama de ojo de la señal recibida por el sistema en el receptor haciendo uso de los equipos USRP. Puede observarse la dispersión de los símbolos en la figura 18(c) debido al ruido presente en el enlace inalámbrico, una rotación debido principalmente a que los 2 equipos usan sus propias referencias de frecuencia, la cual es menos evidente que en QPSK, pero existe. Se está haciendo uso de corrección de desplazamiento de frecuencia tal como se indicó previamente en la parte de implementación de capa física del receptor. El diagrama de ojo permite apreciar los efectos del ruido y de la Interferencia Intersimbólica del canal inalámbrico real al no haber una apertura clara.

Sistema con Modulación 8-PSK y efectos de canales multipaso

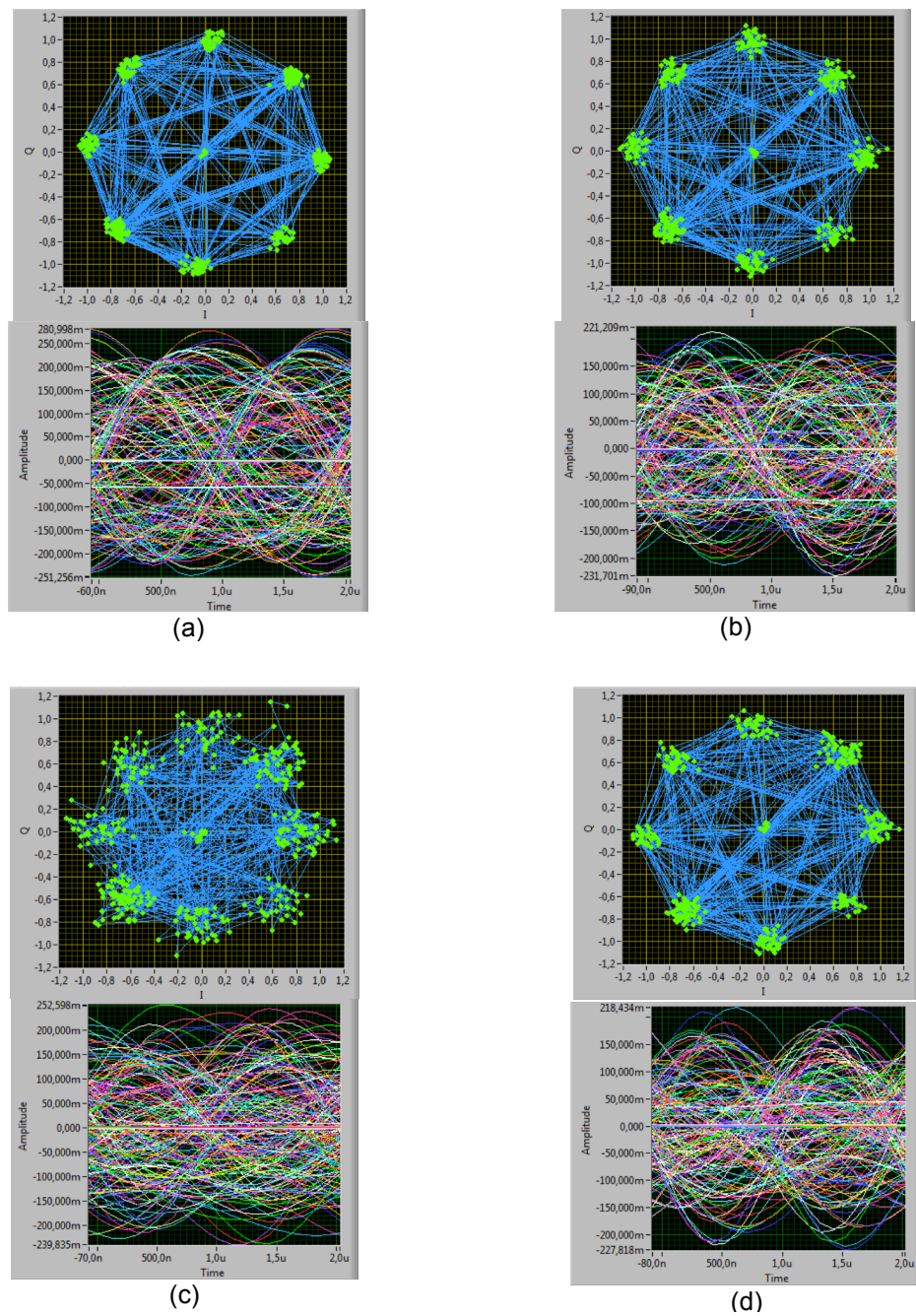


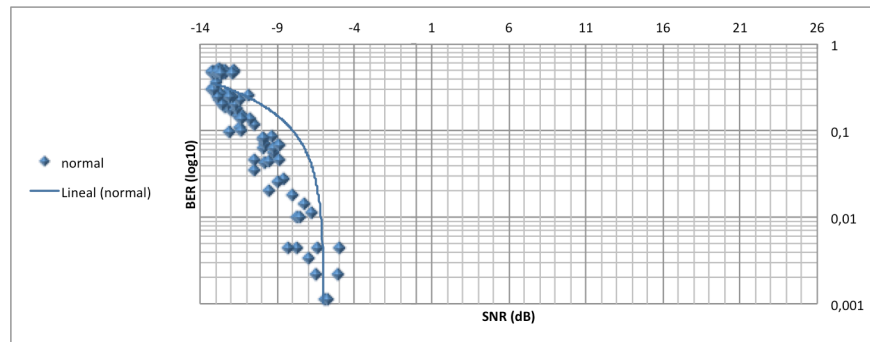
Figura 19. Constelación símbolos y diagrama ojo de sistema usando 8-PSK y equipos USRP con efectos de canales multipaso. (a) Canal T3CA; (b) Canal T3CB; (c) Canal T4CA; Canal T4CB.

En cada una de las figuras 19(a) a 19(d) se muestran combinados la constelación y diagrama de ojo de la señal recibida por el sistema en el receptor con 8-PSK, pero con los efectos de los 4 canales multipaso cuyos perfiles de potencia retraso se mostraron en las tablas 5 a 8 en la parte de implementación. Cada prueba siguió un proceso similar a las de QPSK, es decir a canal le fue aplicada una ecualización de longitud igual al número de multipaso del canal, a excepción del canal T4CA (figura 19c) que necesitó un ecualizador longitud 7. También se colocaron valores idénticos para cada canal en la opción de *Channel Estimate Length* para poder visualizar el multipaso en los perfiles de potencia retraso obtenidos y que constan en el anexo 1.

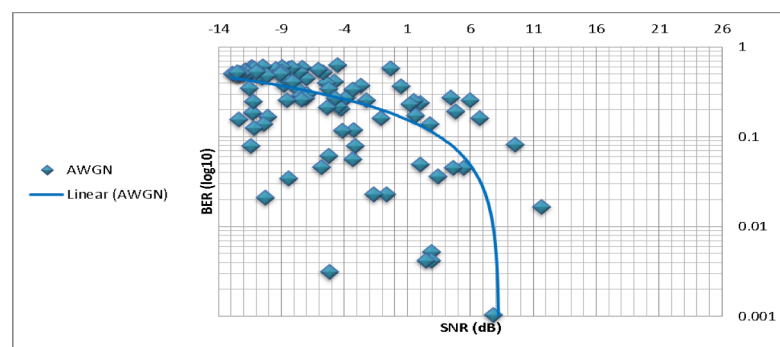
4.3 Pruebas BER vs. SNR

Se presentan las gráficas de BER vs. SNR del sistema para visualizar el desempeño a nivel de errores de bits de ambas modulaciones 8-PSK y QPSK en los 4 escenarios de los canales de perfiles de potencia retraso. En cada gráfica constan adicionalmente a los valores obtenidos en pruebas, las gráficas de tendencia de los valores usando un método lineal en MS Excel.

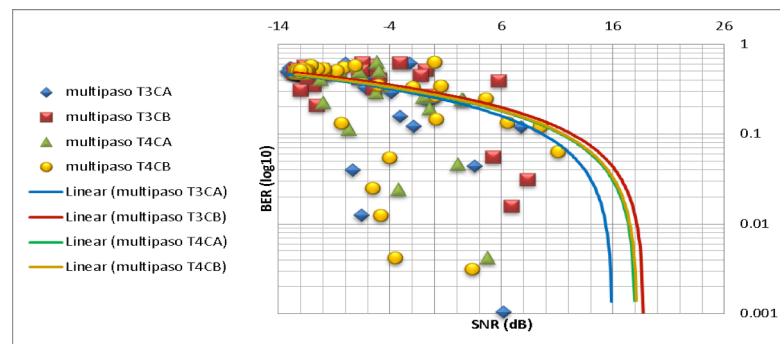
Sistema con Modulación QPSK y efectos de canales multipaso



(a)



(b)



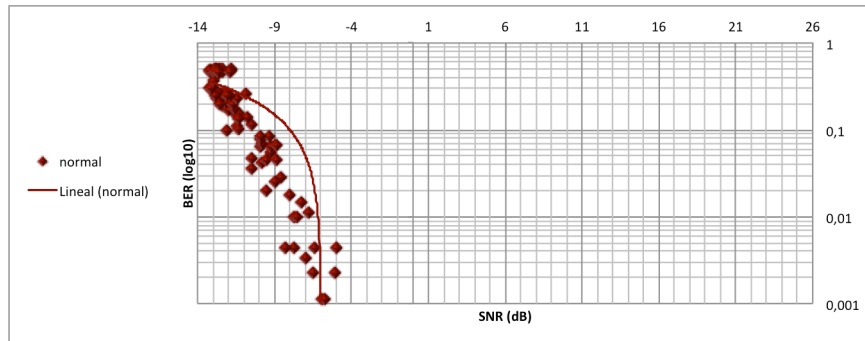
(c)

Figura 20. Gráfica BER vs. SNR de sistema usando QPSK. (a) Simulado; (b) con USRP y canal AWGN; (c) con USRP y Canales multipaso de tablas perfiles potencia y AWGN.

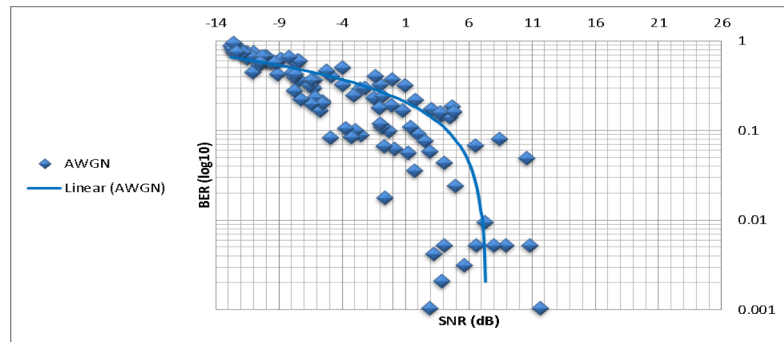
Para la creación de la gráfica de la figura 20(a), 20(b) y 20(c), se ingresaron en forma de iteraciones (usando un lazo For) valores de ruido en el canal AWGN en el rango de -60 a 19 dB en pasos de 1 dB (80 iteraciones), que se tradujeron aproximadamente a valores de SNR entre 25 y -12 dB. La cantidad de valores obtenidos en cada prueba es menor o igual a 80 debido a que se eliminaron los

valores en los cuales BER=0.

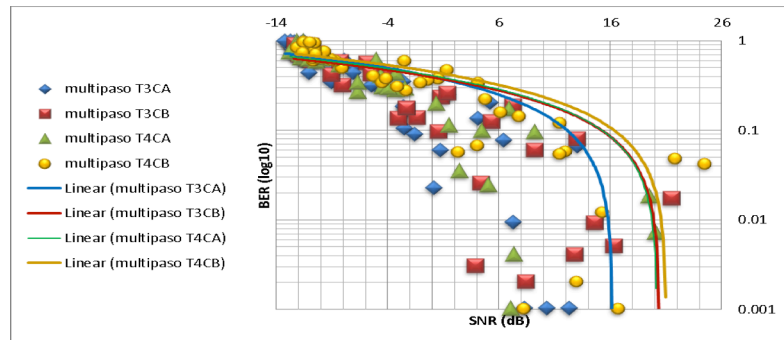
Sistema con Modulación 8-PSK y efectos de canales multipaso



(a)



(b)



(c)

Figura 21. Gráfica BER vs. SNR de sistema usando 8-PSK. (a) Simulado; (b) con USRP y canal AWGN; (c) con USRP y Canales multipaso de tablas perfiles potencia y AWGN.

Para la creación de la gráfica de la figura 21(a), 21(b) y 21(c), se ingresaron al igual que QPSK en forma de iteraciones (usando un lazo For) valores de ruido en el canal AWGN en el rango de -60 a 19 dB en pasos de 1 dB (80

iteraciones), que se tradujeron aproximadamente a valores de SNR entre 25 y -12 dB. La cantidad de valores obtenidos en cada prueba es menor o igual a 80 debido a que se eliminaron los valores en los cuales BER=0.

4.4 Pruebas Tasa de bits (R_b) vs. SNR

Se presentan las gráficas de BER vs. SNR del sistema para visualizar el desempeño a nivel de tasa de bits de ambas modulaciones 8-PSK y QPSK en los 4 escenarios de los canales de perfiles de potencia retraso usando los equipos USRP:

Sistema con Modulación QPSK y efectos de canales multipaso

Para la creación de la gráfica con QPSK, se utilizaron dos lazos For anidados. En el lazo externo se ejecutaron 5 iteraciones en pasos de tasa de símbolo 1M, 5M, 10M, 12,5M y 20M, equivalentes a las tasas de bits 2, 10, 20, 25 y 40 Mbps

En el lazo interno se ejecutaron 21 iteraciones con valores de ruido en el canal AWGN en el rango de -20 a 0 dB en pasos de 1 dB, que se tradujeron aproximadamente a valores de SNR entre 18 y -9 dB. En total el sistema realizó 105 iteraciones por cada canal, dando un total de iteraciones en pruebas de R_b vs. SNR de:

$$105 \frac{\text{iteraciones}}{\text{canal}} \times 5 \text{ canales} = 525 \text{ iteraciones}$$

Se tomó como criterio para establecer que el sistema transmite exitosamente para un valor específico combinado de tasa bits R_b y de SNR y así crear la gráfica, que tanto la secuencia de bits del paquete de red de transmisor y receptor sean iguales, es decir valor booleano Verdadero. Los resultados se presentan en la figura 22(a) hasta 22(e)

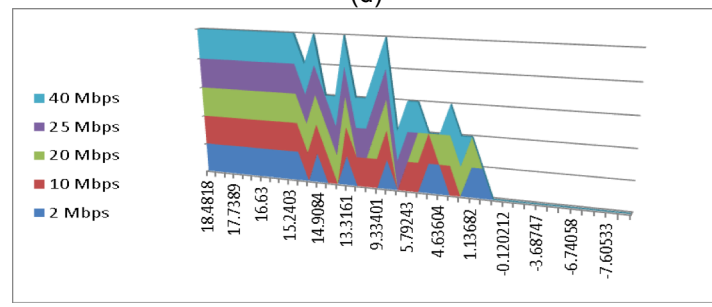
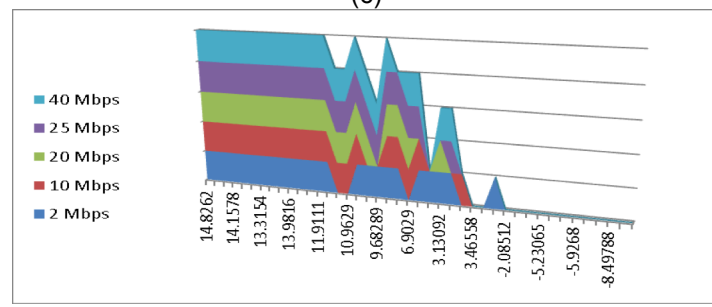
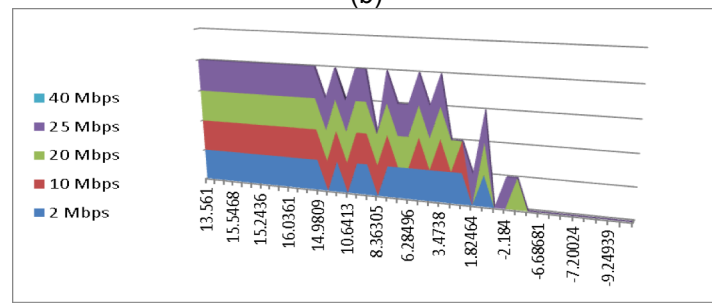
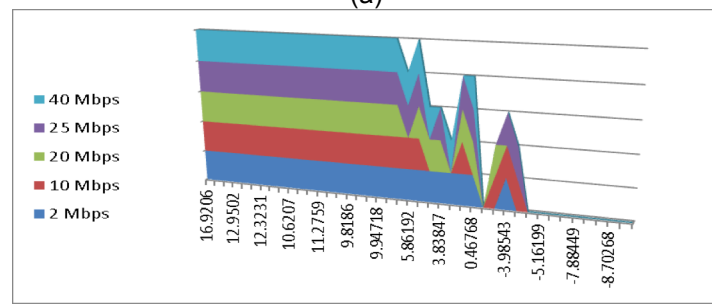
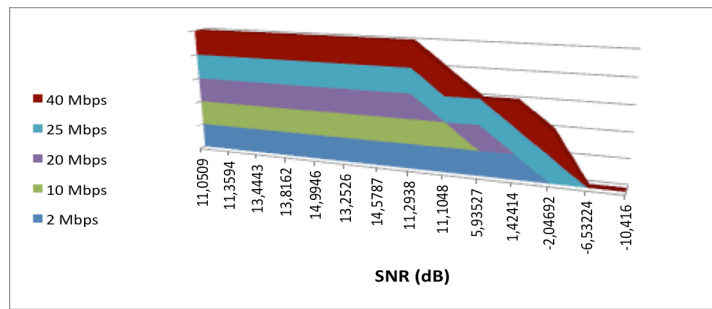


Figura 22. Gráfica R_b vs. SNR de sistema con USRP usando QPSK. (a) Canal AWGN; (b) Canal multipaso T3CA; (c) Canal multipaso T3CB; (d) Canal multipaso T4CA; (e) Canal multipaso T4CB.

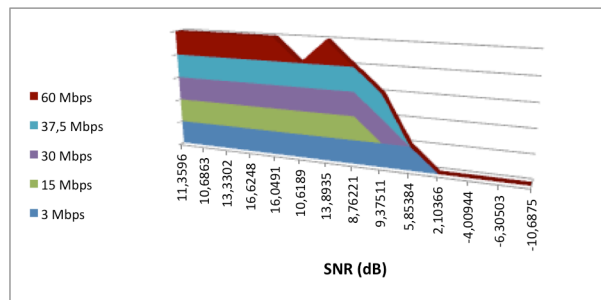
Sistema con Modulación 8-PSK y efectos de canales multipaso

Para la creación de la gráfica con 8-PSK, también se utilizaron dos lazos For anidados. En el lazo externo se ejecutaron 5 iteraciones en pasos de tasa de símbolo 1M, 5M, 10M, 12,5M y 20M, equivalentes a las tasas de bits 3, 15, 30, 37.5 y 60 Mbps

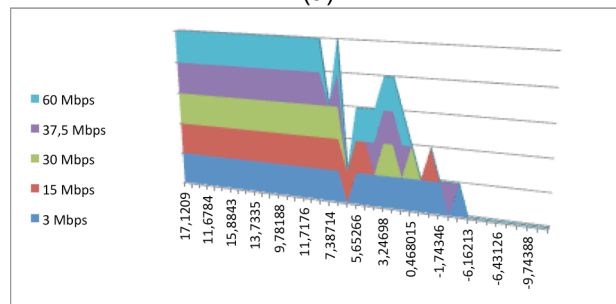
En el lazo interno se ejecutaron 21 iteraciones con valores de ruido en el canal AWGN en el rango de -20 a 0 dB en pasos de 1 dB, que se tradujeron aproximadamente a valores de SNR entre 18 y -9 dB. En total el sistema realizó 105 iteraciones por cada canal, dando un total de iteraciones en pruebas de R_b vs. SNR de:

$$105 \frac{\text{iteraciones}}{\text{canal}} \times 5 \text{ canales} = 525 \text{ iteraciones}$$

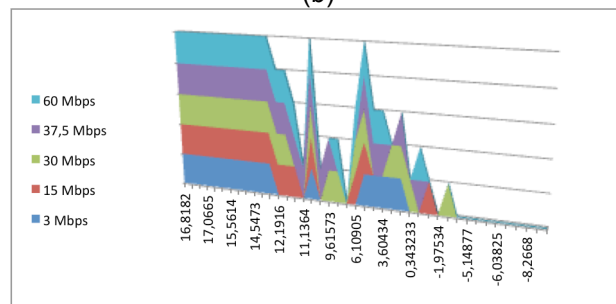
Se tomó como criterio para establecer que el sistema transmite exitosamente para un valor específico combinado de tasa bits R_b y de SNR y así crear la gráfica, que tanto la secuencia de bits del paquete de red de transmisor y receptor sean iguales, es decir valor booleano Verdadero. Los resultados se presentan en la figura 23(a) hasta 23(e)



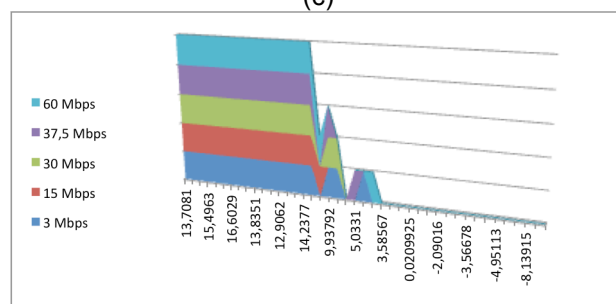
(a)



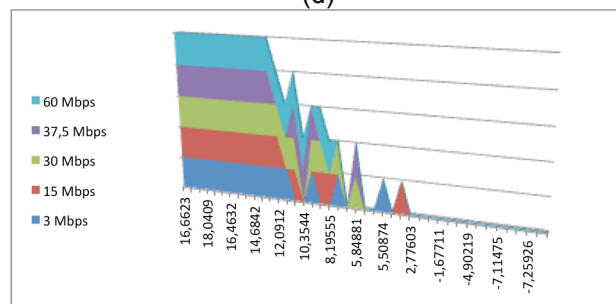
(b)



(c)



(d)



(e)

Figura 23. Gráfica Tasa bits vs. SNR de sistema con USRP usando modulación 8-PSK. (a) Canal AWGN; (b) Canal multipaso T3CA; (c) Canal multipaso T3CB; (d) Canal multipaso T4CA; (e) Canal multipaso T4CB.

CAPÍTULO 5

5. ANÁLISIS DE RESULTADOS

5.1 Comparativa de constelaciones y diagramas de ojo

El sistema con ambas modulaciones QPSK y 8-PSK presenta comportamientos similares en cuanto a la dispersión de los símbolos al ser expuestos a los efectos del canal AWGN y los 4 canales multipaso. Un ejemplo de ello es que ambos presentan una alta dispersión en el canal de la tabla 7 (exteriores tipo rural T4CA) en las figuras 17(b) y 19(b), mientras que en los 3 canales restantes presentan una dispersión creciente e medida se incrementan los multipaso del canal.

Todas las constelaciones de símbolo en los escenarios de pruebas realizados con los USRP presentan una leve rotación debido a que el transmisor y receptor están usando cada uno su propia fuente de referencia de sincronismo interna tal como se explicó en la parte de implementación de capa física. Esta rotación sería mayor y no permitiría la recepción coherente de los símbolos si es que no estuviera activada permanentemente la función de detección y corrección de desplazamiento de frecuencia.

El sistema con modulación QPSK, debido a la forma de su constelación, conserva sus puntos lo mas cerca de su posición ideal (figura 16a) y tienden menos a confundirse con otros puntos, lo que si sucede con 8-PSK dado que su disposición de símbolos están muy cerca, por ejemplo con el canal la tabla 7 (exteriores tipo rural) el sistema usando modulación 8-PSK (figura 19c) no permite la llegada de la secuencia de bits al receptor sin al menos un error en la mayor parte de los intentos.

Los diagramas de ojo muestran como los canales multipaso afectan al sistema produciendo que no se observe una apertura de ojo clara e incluso inexistente

con todos los canales junto con el efecto del canal AWGN. Sin embargo, la ecualización lineal en su mayor parte y el escalador de símbolos en la etapa previa a la demodulación sea QPSK o 8-PSK en menor contribución, permiten recuperar lo suficiente los símbolos para que sean recuperados sin errores hasta un cierto nivel de AWGN (-60 a -10 dB) como se observa mas adelante en las gráficas de BER vs. SNR.

El canal de tabla 7 (T4CA) al presentar un mayor efecto al sistema tanto en QPSK como 8-PSK, hizo necesario que se le deba aplicar una ecualización de longitud 7. Se tomó esta decisión que permite tener transmisión exitosa en escasos intentos en QPSK y 8-PSK junto con valores de ruido AWGN bajo los -10 dB, dado que al estar usando un tiempo de símbolo de 100ns, este canal tiene 4 pasos con retrasos de 200ns. En total al considerar el tiempo de símbolo 100ns, se tienen en realidad 7 pasos.

5.2 Comparativa escenarios BER vs SNR

Como referencia, se presentan en las figuras 20(a) y 21(a) la relación BER vs. SNR del sistema en condiciones ideales (solo simulación en software) con ambas modulaciones QPSK y 8-PSK en donde se observa que el rango de SNR no supera los -4 dB, indicando que en un ambiente ideal no se necesitan valores altos de SNR para *comenzar a obtener* una tasa baja de errores en el orden de 10^{-3} .

Después de realizar varios ensayos, se encontraron los valores aceptables de ruido (-60 a 19 dB) para ingresar a las pruebas con el fin de obtener valores de BER=0, o valores repetidos que corresponden cuando el error es alto.

Todas las pruebas del sistema tanto simuladas como con USRP presentaron un comportamiento similar en el extremo izquierdo (correspondiente a un bajo nivel de SNR) comenzando todas desde -14 dB en donde se observan los valores altos de BER entre 1 y 10^{-1} .

Las pruebas del sistema con los USRP tendieron a extender en ambas modulaciones las gráficas hacia la derecha en varias unidades de decibeles de SNR hasta llegar a un tope de 20 a 21 dB en promedio en comparación con las pruebas ideales de las figuras 20(a) y 21(a), lo que indica que en un medio inalámbrico *real* se necesita un mayor SNR para poder obtener una transmisión baja en errores.

El sistema con modulación 8-PSK en comparación con la modulación QPSK necesita ligeramente un mayor SNR para poder transmitir con la misma tasa de errores de bit en los canales con multipaso igual o mayor a 4. Ambas modulaciones tienen un comportamiento similar en el canal multipaso de la tabla 5 que es solo de 2 multipaso.

El sistema con modulación 8-PSK llega a valores mas altos de error de bit aproximadamente entre 0,7 y 0,8 en comparación a QPSK que llega a valores de BER aproximadamente entre 0,5 y 0,6.

5.3 Comparativa escenarios Tasa bits vs. SNR

Después de realizar varios ensayos, se encontró los valores aceptables de ruido (-20 a 0 dB) para ingresar a las pruebas del sistema que permitan que el error de bit sea suficiente para que se presente una desigualdad de al menos un bit entre la información transmitida y recibida (valores booleanos 0) que permita obtener valores booleanos que no sean solamente valores 1 (recepción exitosa sin errores)

Se observa que el sistema usando modulación QPSK (figura 22) tiene un desempeño máximo de transmisión exitosa a una tasa de bits de 40 Mbps hasta un valor de 11 dB de SNR (sin efectos de canal multipaso) En el caso de considerar efectos de canal multipaso, el desempeño máximo de transmisión éxitos hasta 40 Mbps es el siguiente: el canal de la tabla 5 (2 multipaso) es similar (incluso parece mejor) al del sistema sin efectos de canal llegando hasta un SNR=10 dB, en el resto de canales de las tablas 6, 7 y 8 solo pueden

transmitir exitosamente hasta SNR entre 13 y 15 dB. Los canales multipaso de las tabla 5 y 6 presentan que puede obtener *al menos* una transmisión exitosa hasta aproximadamente SNR=0 dB y SNR=3 dB; los canales de las tablas 7 y 8 solo pueden hacerlo hasta SNR=9 dB aproximadamente.

Se observa que el sistema usando modulación 8-PSK (figura 23) tiene un desempeño máximo de transmisión exitosa a una tasa de bits de 60 Mbps hasta un valor antes de 10 dB de SNR (sin efectos de canal multipaso) En el caso de considerar efectos de canal multipaso, el desempeño máximo de transmisión éxitos hasta 60 Mbps es el siguiente: el canal de la tabla 5 es similar al del sistema sin efectos de canal llegando hasta un SNR=11 dB, el resto de canales (tablas 6, 7 y 8) solo pueden transmitir exitosamente hasta SNR entre 14 y 16 dB. Los canales multipaso de las tabla 5 y 6 presentan que puede obtener *al menos* una transmisión exitosa hasta aproximadamente SNR=3 dB y SNR=6 dB; el canal de la tablas 7 pueden hacerlo hasta SNR=14 dB aproximadamente y el ultimo canal (tabla 8) solo puede hacerlo hasta SNR=10 dB aproximadamente.

CONCLUSIONES

1. Se ha podido diseñar e implementar las funciones de aplicación, red y acceso en un sistema inalámbrico de una sola portadora con modulaciones QPSK y 8-PSK para transmisión de texto. El proyecto pudo completarse basado en las consideraciones iniciales de diseño y adaptándolas hasta llegar a las especificaciones y la implementación en sí del sistema en código nativo de LabVIEW en la mayor parte del sistema. Las capas del sistema guardan independencia entre sus funciones, aunque todas dependen de la información que pase a través de ellas (excepto aplicación)
2. Se desarrolló e implementó una lógica para CSMA que permite sensar la

portadora (canal inalámbrico) para determinar si está libre o no para la transmisión de información así como reconocimiento ACK de mensajes recibidos. Todo esto se hizo adaptando el funcionamiento de los subVI de [1] y de los USRP.

3. Se implementó un mecanismo de cifrado basado en RC4 a nivel de aplicación que permite hacer mas seguro la transmisión de texto en el medio inalámbrico.
4. También se ha incluido detección de errores a nivel de red con la implementación de 8-CRC CCITT, el cual ante al menos un error de bit, no permite que pase la información a la capa de aplicación.
5. A nivel de pruebas de desempeño del sistema, a nivel de un menor error de bit (BER) la modulación QPSK presenta una mayor fortaleza frente a 8-PSK como era de esperarse. Sin embargo, las pruebas de tasa de bits R_b indican que tienen casi un desempeño similar en lograr una máxima capacidad de transmisión para niveles de SNR similares.

BIBLIOGRAFÍA

- [1] R. W. Heath Jr., Digital Wireless Communication: Physical Layer Exploration Lab Using the NI USRP, National Technology and Science Press, 2012.
- [2] T. S. Rappaport, Wireless Communications: Principles and Practice, Prentice Hall, 1996, pp. 159-163
- [3] D. Tse and P. Viswanath, Fundamentals of Wireless Communication, Cambridge University Press, 2005, pp. 32-51
- [4] B. Sklar, Digital Communications: Fundamentals and Applications, Prentice Hall, 2001, pp. 944-958

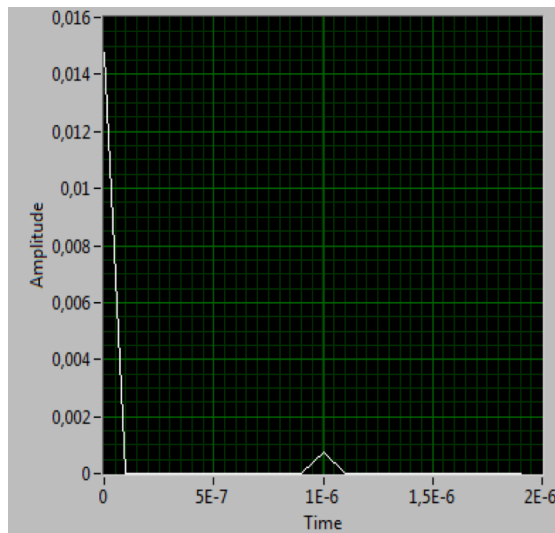
- [5] A. S. Tanenbaum, *Redes de Computadoras*, Pearson Educación de México, 2003, pp. 255-256
- [6] C. L. Clark, *LabVIEW Digital Signal Processing and Digital Communications*, The McGraw-Hill Companies, 2005, pp. 115-137
- [7] "National Instruments Community Native LabVIEW DES Encryption," 11 7 2008. [Online]. Available: <https://decibel.ni.com/content/docs/DOC-2480>. [Accessed 8 2016].
- [8] "National Instruments Forums LabVIEW text to bit stream," 27 8 2015. [Online]. Available: <http://forums.ni.com/t5/LabVIEW/text-to-bit-stream/td-p/1236834>. [Accessed 8 2016].
- [9] "National Instruments USRP Software Radio," 08 04 2014. [Online]. Available: <http://forums.ni.com/t5/USRP-Software-Radio/quot-MIMO-Cable-link-did-not-lock-within-alloted-time-quot-error/td-p/2942528>. [Accessed 8 2016].
- [10] "USRP Hardware Driver and USRP Manual Device Synchronization," [Online]. Available: http://files.ettus.com/manual/page_sync.html. [Accessed 8 2016].
- [11] "Example of generating an 8-bit CRC," [Online]. Available: https://en.wikipedia.org/wiki/Computation_of_cyclic_redundancy_checks#/media/File:CRC8-gen.gif. [Accessed 8 2016].
- [12] "RC4," [Online]. Available: <https://en.wikipedia.org/wiki/RC4>. [Accessed 9 2016].
- [13] "RC4 Example," [Online]. Available: <https://sandilands.info/sgordon/teaching/reports/rc4-example.pdf>. [Accessed 9 2016].
- [14] B. Sklar, *Digital Communications: Fundamentals and Applications*, Prentice Hall, 2001, pp. 173-174
- [15] "RSA Key Generator," [Online]. Available: <http://www.csfieldguide.org.nz/en/interactives/rsa-key-generator/index.html>. [Accessed 9 2016].

- [16] "NI USRP-292x/293x Datasheet," [Online] . Available:
<http://www.ni.com/datasheet/pdf/en/ds-355>. [Accessed 9 2016].

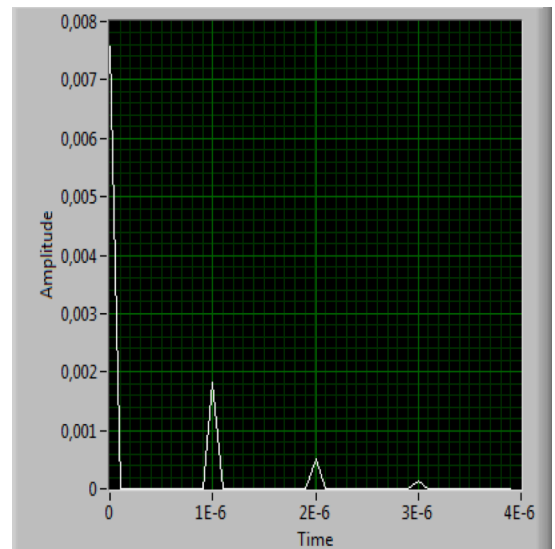
ANEXOS

ANEXO 1 - Perfiles de potencia retraso

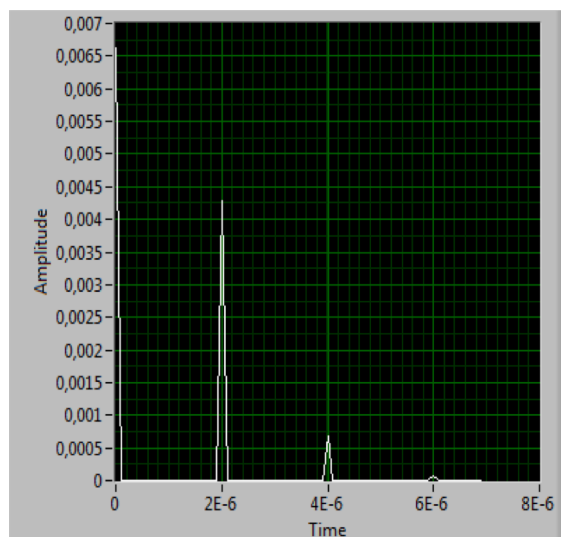
Modulación QPSK



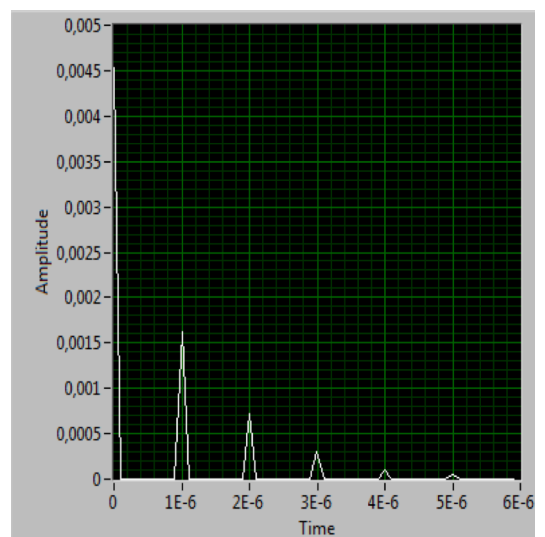
1. Perfil de potencia retraso de canal Interiores Residencial T3CA.



2. Perfil de potencia retraso de canal Interiores Residencial T3CB.

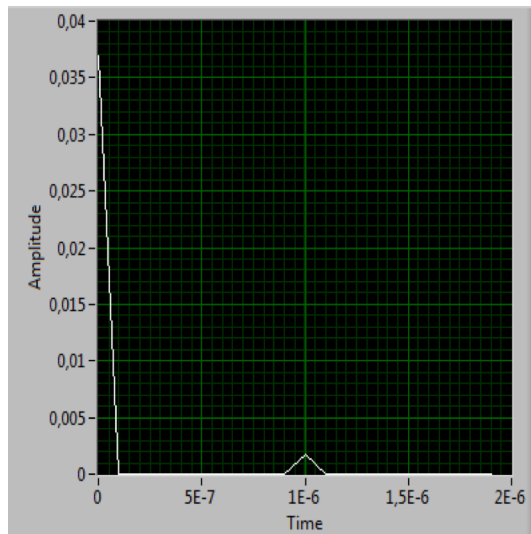


3. Perfil de potencia retraso de canal Exteriores Rural T4CA.

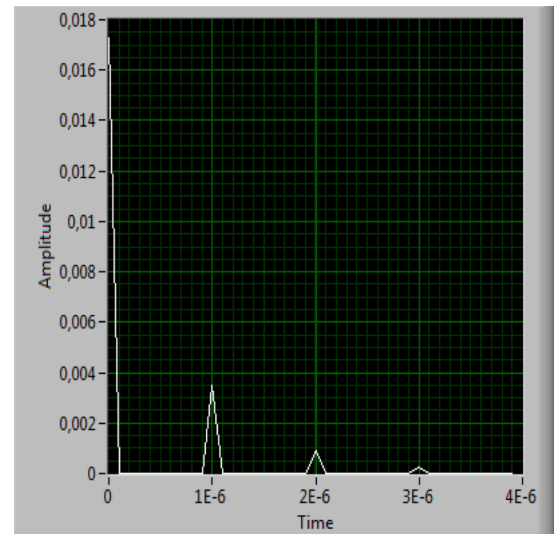


4. Perfil de potencia retraso de canal Exteriores Rural T4CB.

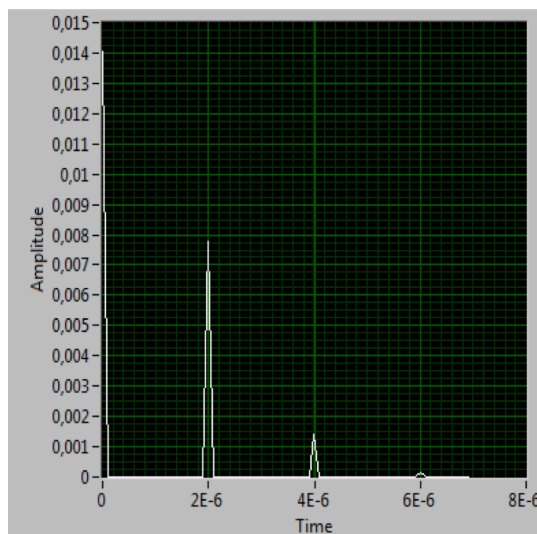
Modulación 8-PSK



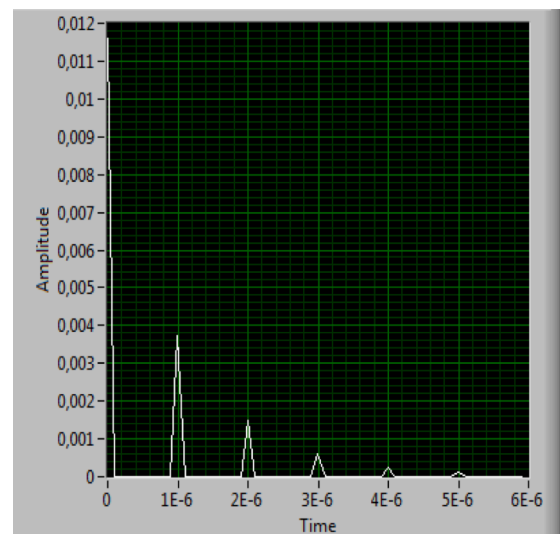
1. Perfil de potencia retraso de canal Interiores Residencial T3CA.



2. Perfil de potencia retraso de canal Interiores Residencial T3CB.

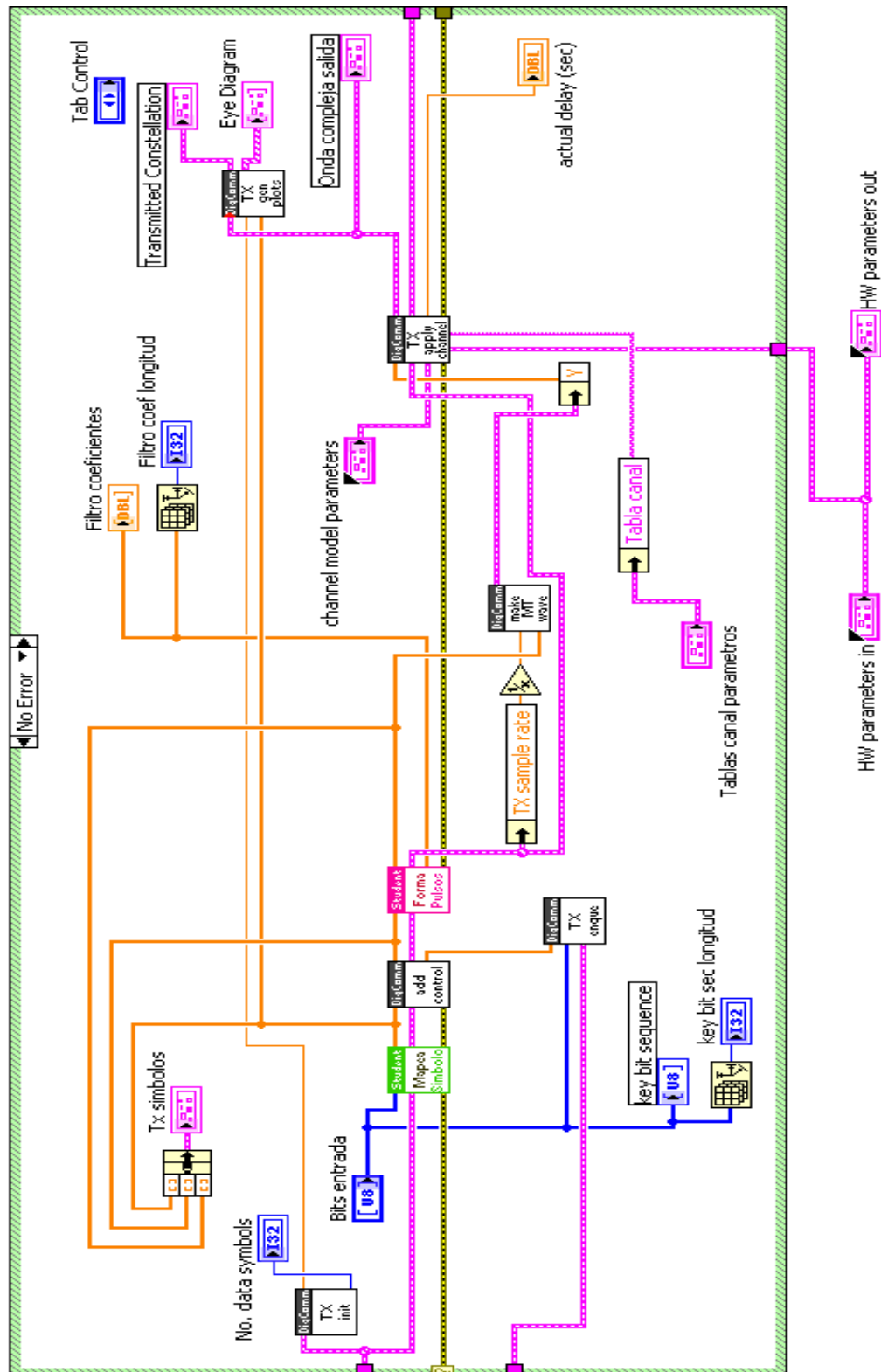


3. Perfil de potencia retraso de canal Exteriores Rural T4CA.



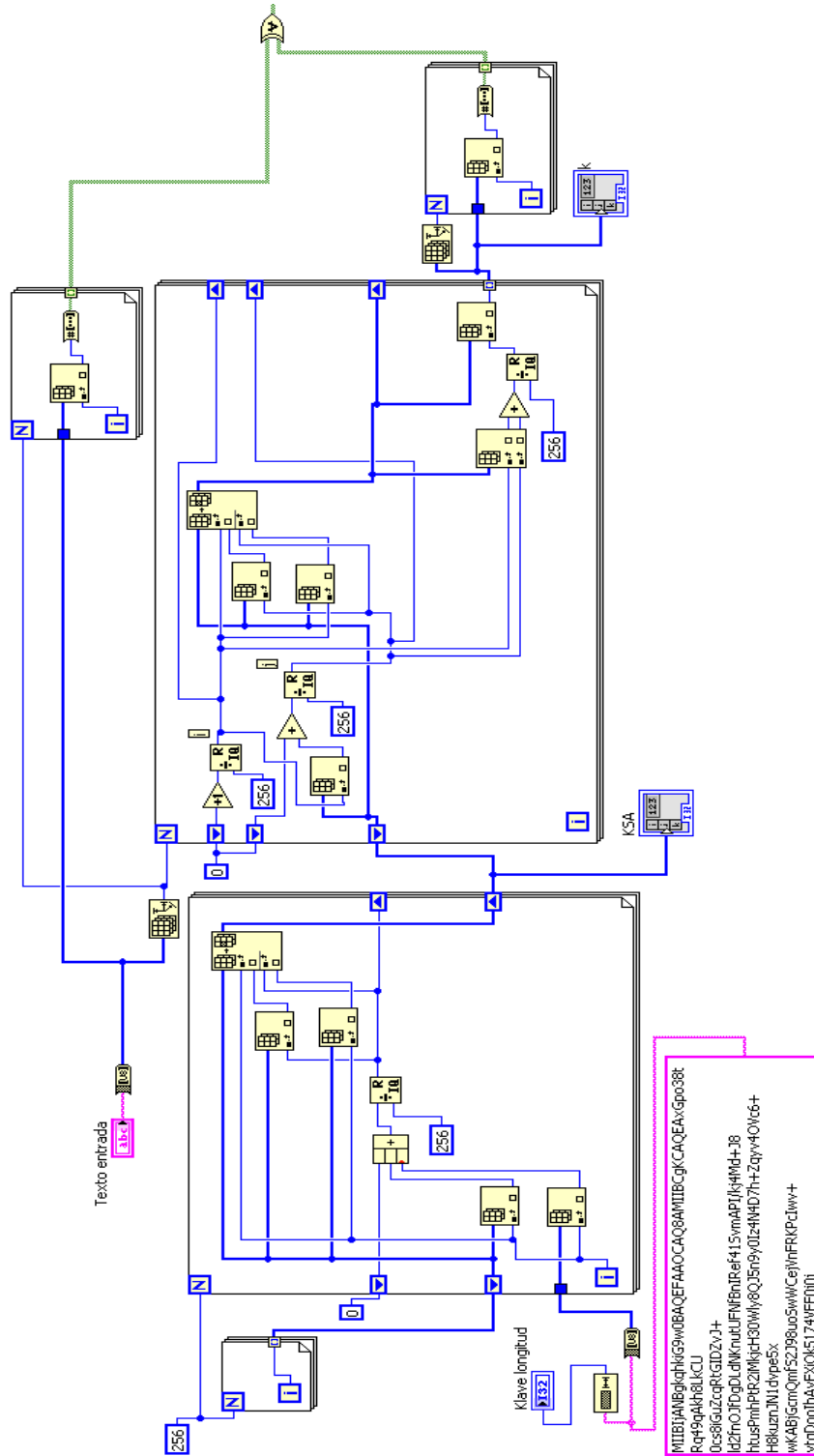
4. Perfil de potencia retraso de canal Exteriores Rural T4CB.

ANEXO 2 – Detalle bloques subVI transmitter.vi adaptado a sistema

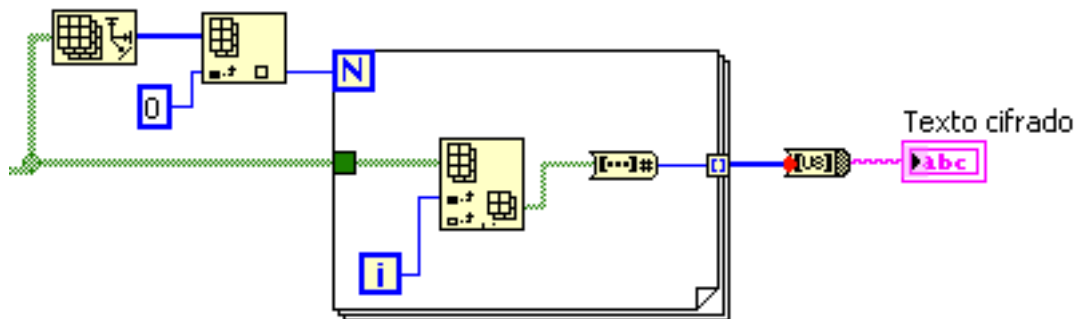


ANEXO 4 – Implementación cifrador RC4 en código LabVIEW

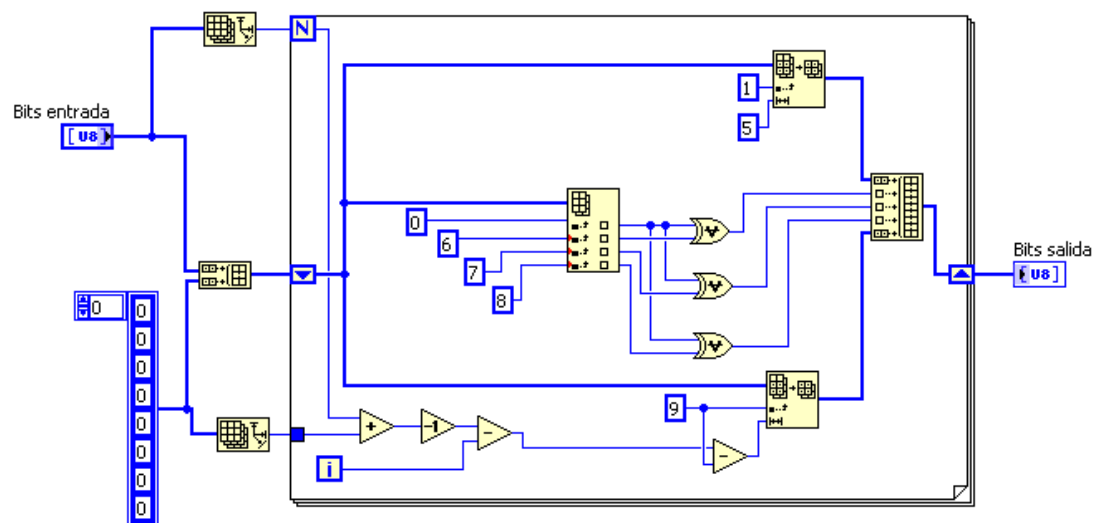
Parte 1



Parte 2

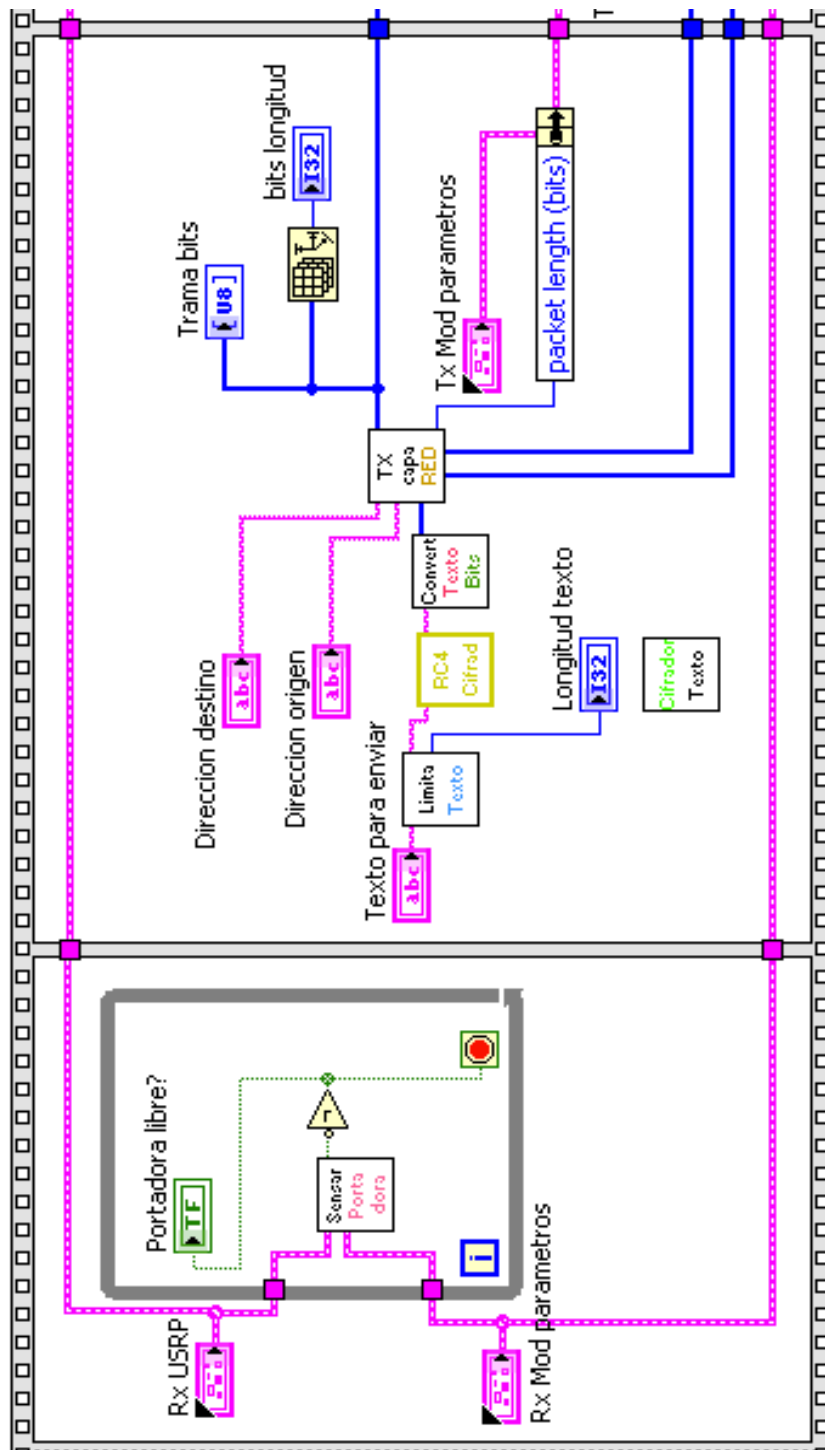


ANEXO 5 – Implementación 8-CRC CCITT en código LabVIEW

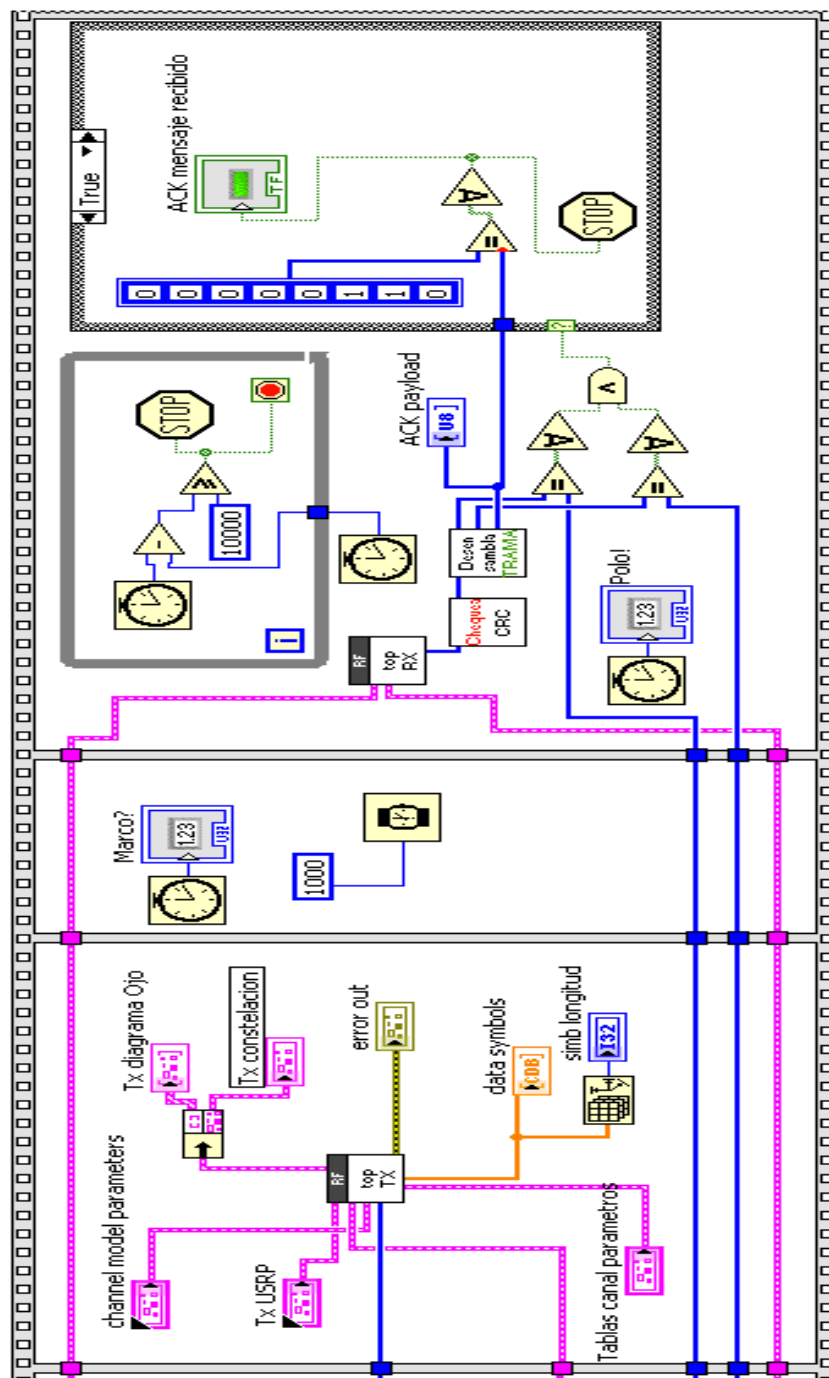


ANEXO 6 – Diagrama bloques implementación Transmisor sistema

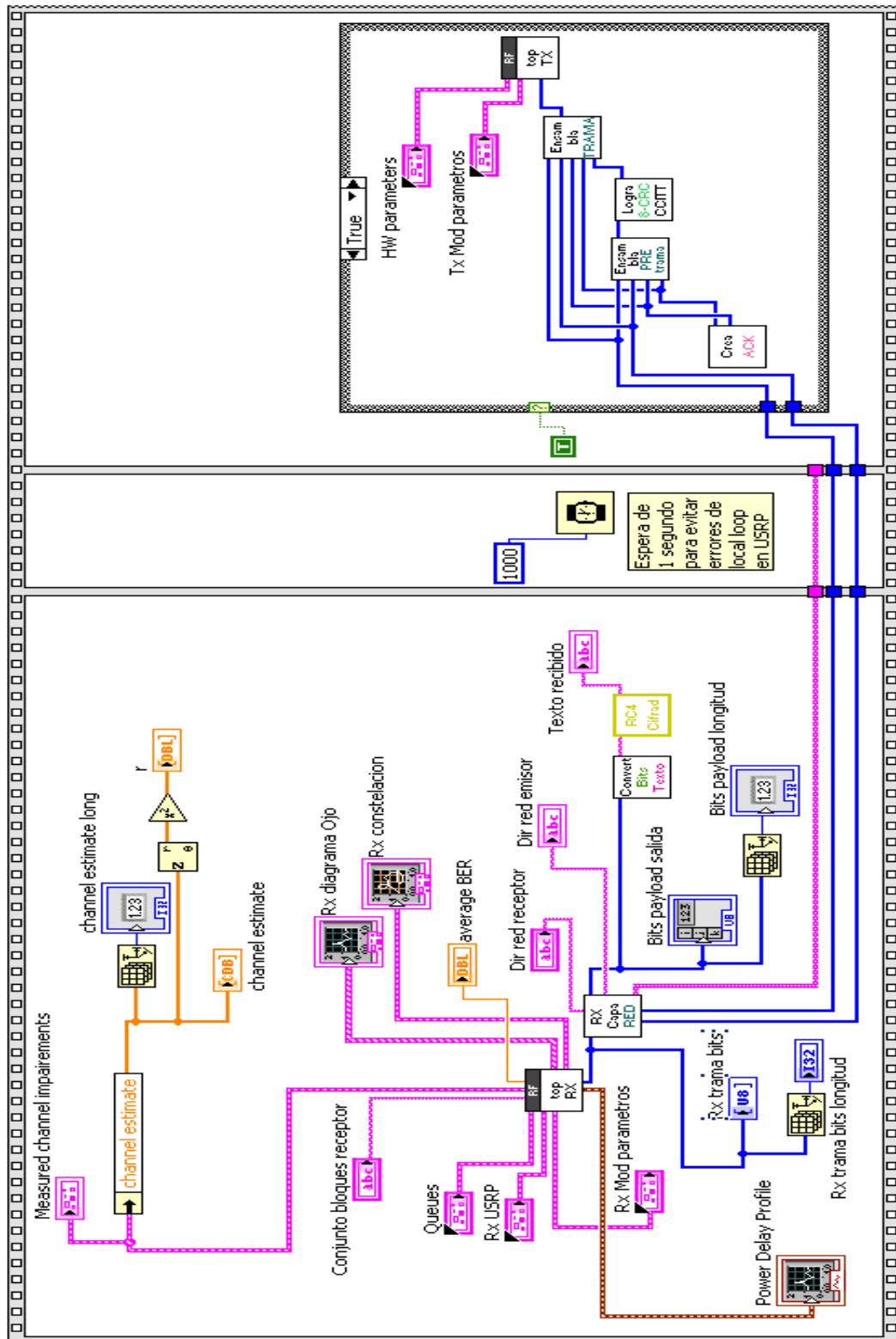
Parte 1



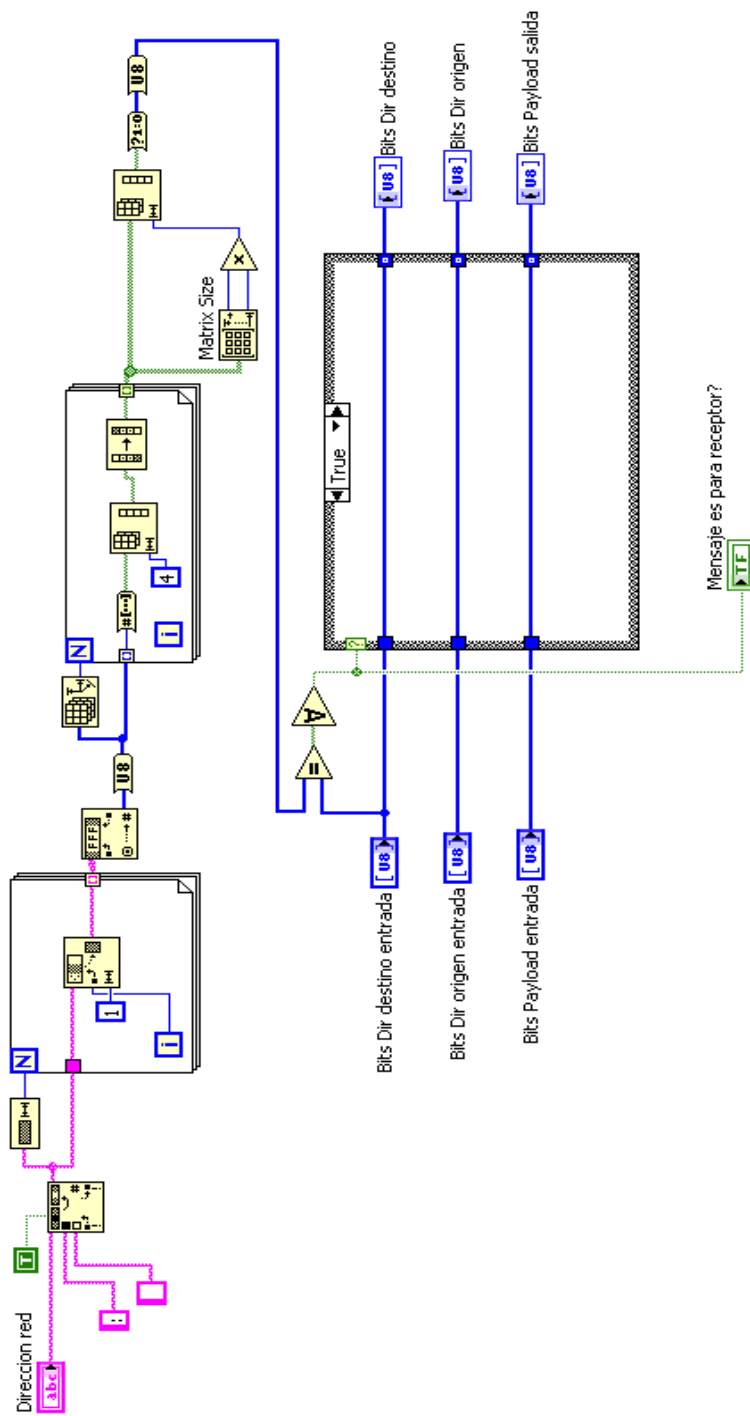
Parte 2



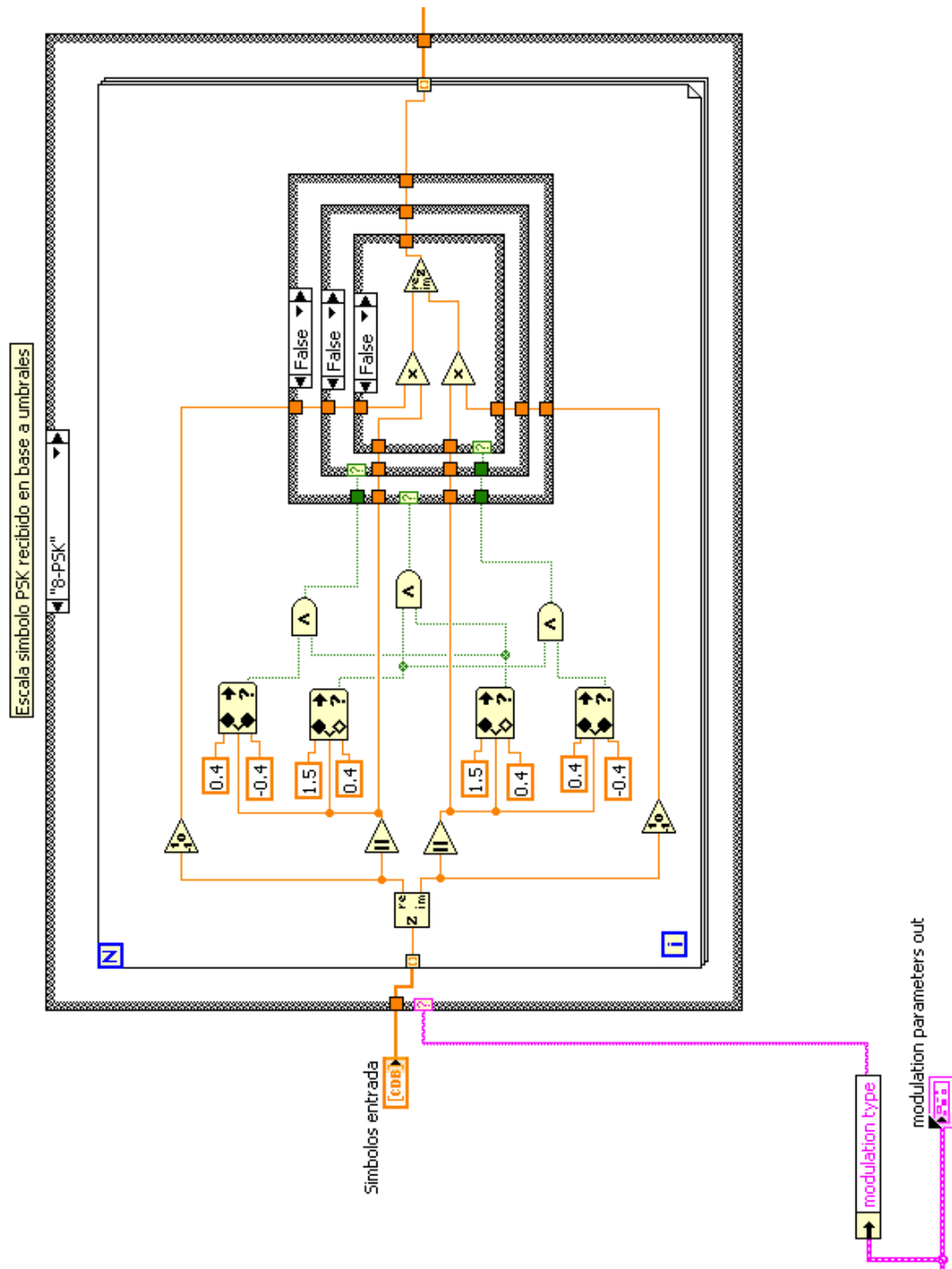
ANEXO 7 – Diagrama bloques implementación Receptor sistema



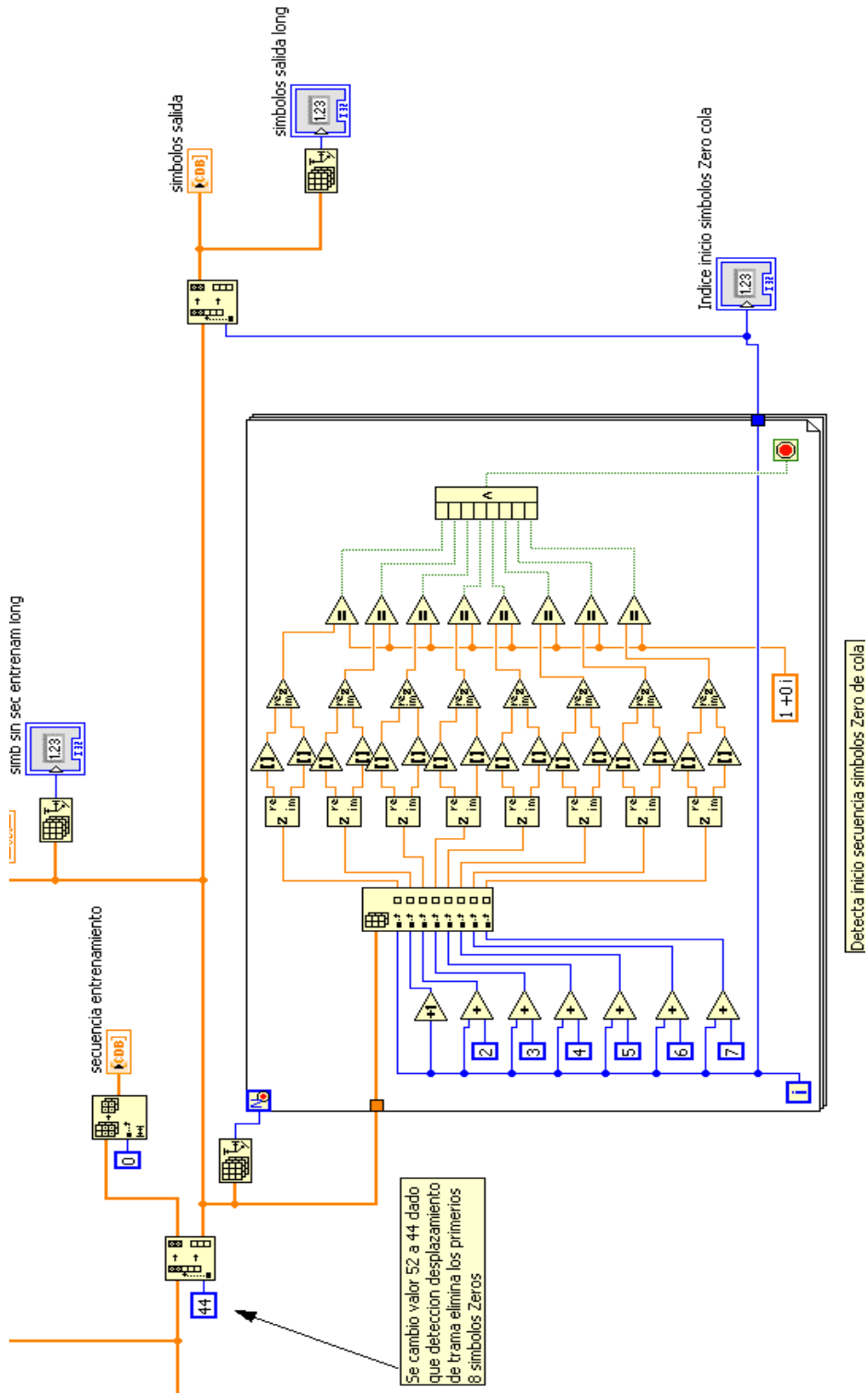
ANEXO 8 – Implementación bloque Verifica Dirección Destino de Capa Red



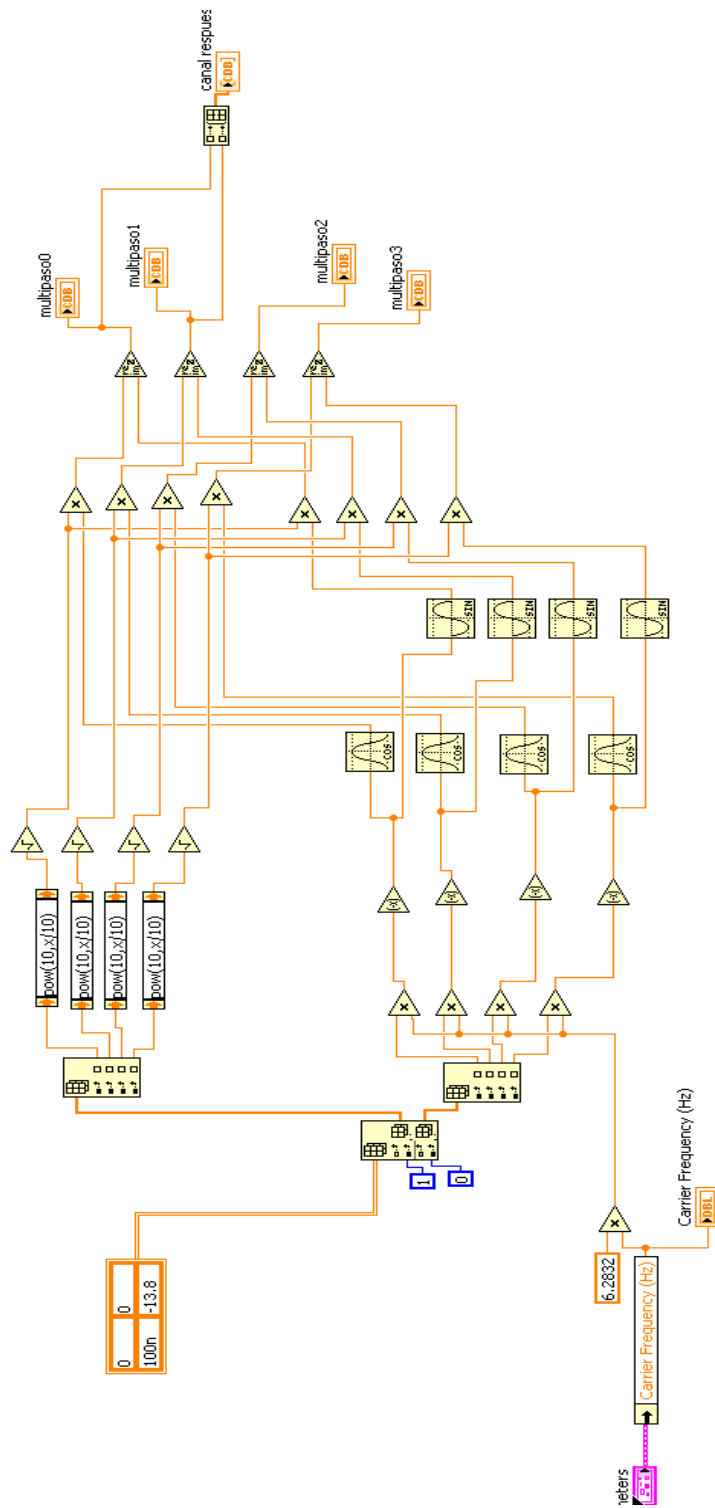
ANEXO 9 – Implementación función Escalador Símbolos de Capa Física



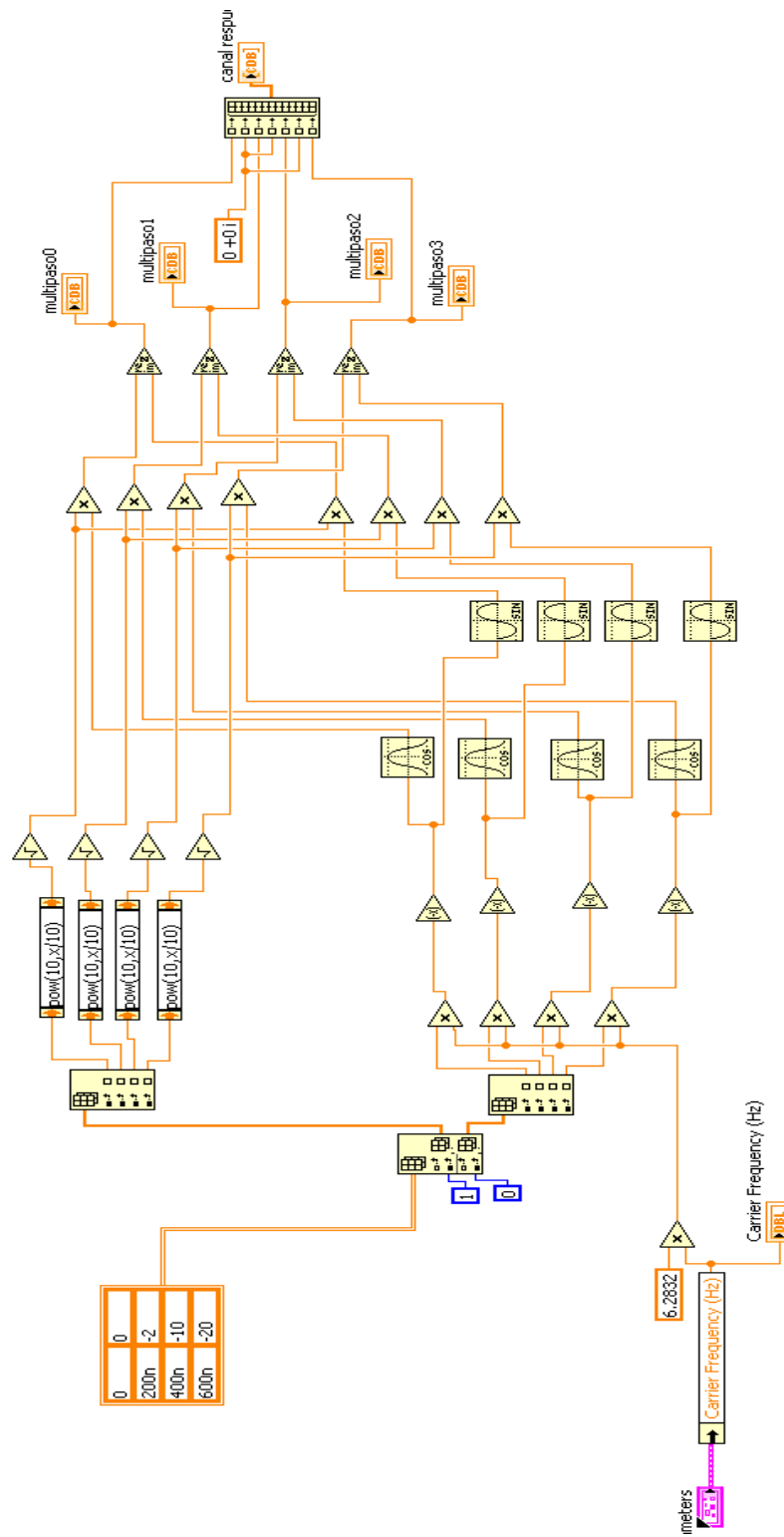
ANEXO 10 – Implementación función Limitador Símbolos de Capa Física



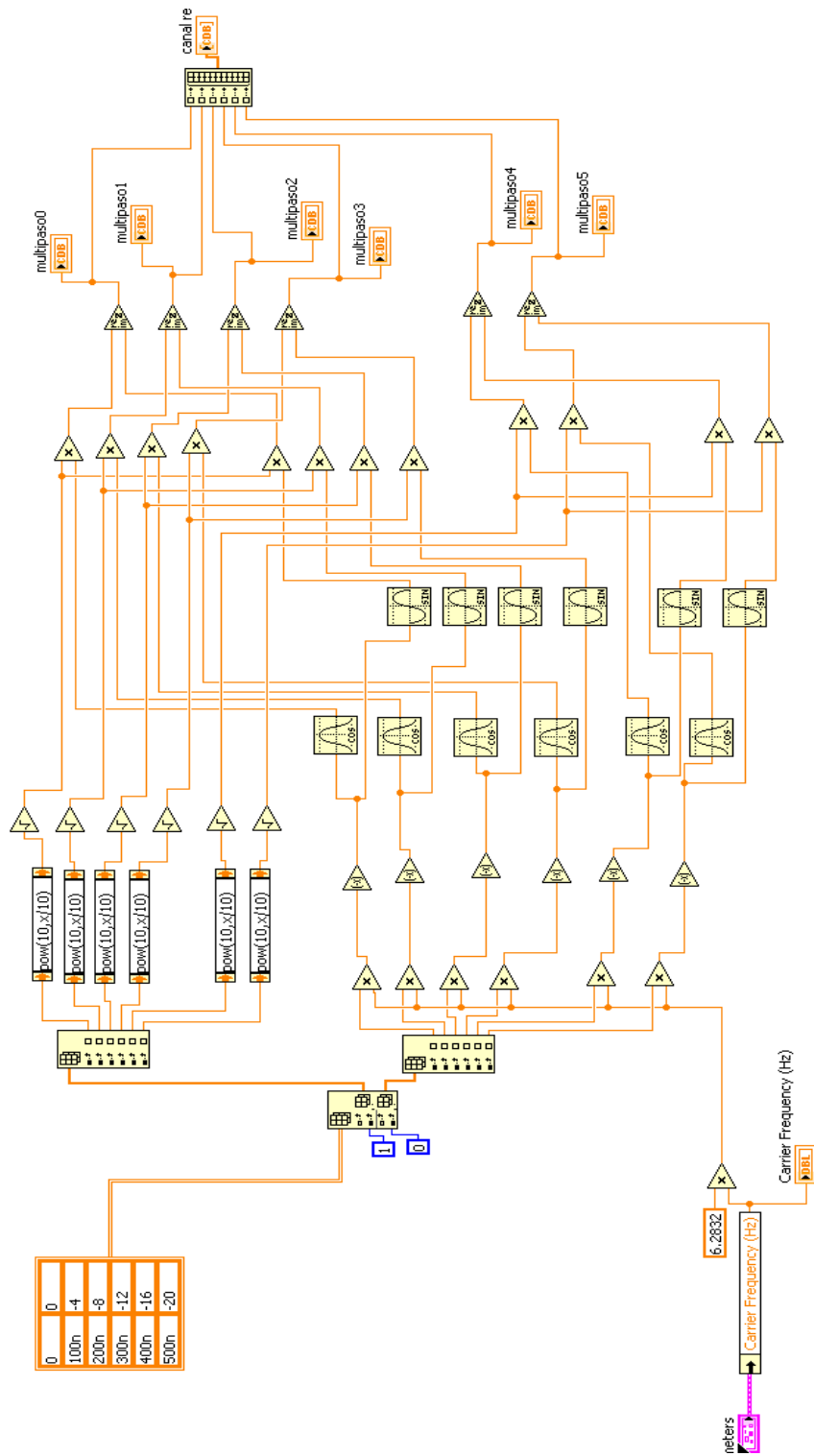
ANEXO 11 – Implementación canal multipaso Interiores Residencial T3CA



ANEXO 12 – Implementación canal multipaso Exteriores Rural T4CA



ANEXO 13 – Implementación canal multipaso Exteriores Rural T4CB



ANEXO 14 – Interfaz de usuario Transmisor y Receptor sistema

The image displays two screenshots of a software interface for a transmitter and receiver system. The top screenshot shows the transmitter configuration, and the bottom screenshot shows the receiver configuration.

Top Screenshot (Transmitter Configuration):

- Project Name:** Proyecto CHS Tx LSMA.vi Front Panel
- Text to send:** No esta muerto lo que yace eternamente y aun con los raros evos la muerte puede morir
- Length of text:** 0
- Carrier frequency:** Portadora libre? (checked)
- Destination:** Charlie Brown [01:23:45:67:89:ab]
- Origin:** Fe-dc:ab:98:76:54
- ACK message received:** ACK mensaje recibido
- Channel Parameters:**
 - Tabla canal: Tabla 3 indoor_res_canal A
 - Modelo canal selectivo frecuencia: Multipaso
- Constellation Diagram (Tx):** Shows a single point at (1.2, 0.0) on a grid with x-axis from -1.2 to 1.2 and y-axis from -0.8 to 0.5.

Bottom Screenshot (Receiver Configuration):

- Project Name:** Proyecto CHS Rx ACK.vi Front Panel
- Received Text:** Texto recibido
- Sender Address:** Dir red receptor: Charlie Brown (01:23:45:67:89:ab)
- Control Information:** Training Sequence (derived): 707m +707m i; Zero Pad Length: 8
- Channel Estimation/Equalizer Parameters:**
 - Equalization Method: Direct
 - channel estimate length: 1
 - equalizer length: 1
 - equalizer delay: -1
- Synchronization:** Synch: Timin; fixed; Symbol: Max; Frame: Slidin; Corr: (set delay to -1 for equalizer to choose optimal delay)
- Constellation Diagram (Rx):** Shows a single point at (1.2, 0.0) on a grid with x-axis from -1.2 to 1.2 and y-axis from -100.0000m to 525.682m.