UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Diseño e implementación de un transmisor AM, FM y ATSC en Radio Definida por Software para docencia

TESIS

Que para obtener el título de

Ingeniera en Telecomunicaciones

P R E S E N T A

Viridiana Mares Rodríguez

DIRECTOR DE TESIS

Dr. José María Matías Maruri

Ciudad Universitaria, Cd. Mx., 2016

Agradecimientos

Al proyecto PAPIME PE107415 denominado “Banco de trabajo de tecnologías de radiodifusión” por el apoyo económico que me otorgó a través de una beca. El trabajo desarrollado es parte de dicho proyecto.

Al proyecto Conacyt-CDTI número 189235, REFUTV, “Desarrollo de Redes en Frecuencia Única para Televisión Digital ATSC”.

Índice general

Índice de figuras

Índice de cuadros

Lista de Términos

AM: Amplitud Modulada.
FM: Frecuencia Modulada.
ATSC: Advanced Television Systems Committee.
SDR: Software Defined Radio.
USRP: Universal Software Radio Peripheral.
SSI: Small Scale Integration.
LSI: Large Scale Integration.
DSP: Digital Signal Processor.
FPGA: Field Programmable Gate Array.
JTRS: Joint Tactical Radio System.
OTAN: Organización del Tratado del Atlántico Norte.
NTSC: National Television System Committee.
PAL: Phase Alternating Line.
SECAM: Séquentiel Couleur à Mémoire.
BPSK: Binary Phase Shift Keying.
QAM: Quadrature Amplitude Modulation.
ISO: International Organization for Standardization.
PCI: Peripheral Component Interconnect.
VME: Versa Module Eurocard.
PCIe: Peripheral Component Interconnect expres.
MIMO: Multiple-input Multiple-output.
CORBA: Common Object Request Broker Architecture.
IEEE: Institute of Electrical and Electronics Engineers.
UHD: USRP Hardware Driver.

DTF: Discrete Fourier Transform.
FFT: Fast Fourier Transform.
IIR: Infinite Impulse Response.
FIR: Finite Impulse Response.
VHF: Very High Frequency.
RDS: Radio Data System.
IFT: Instituto Federal de Telecomunicaciones.
WBFM: Wide Band FM.
GUI: Graphical User Interface.
JCIC: Joint Committee on InterSociety Coordination.
FCC: Federal Communications Commission.
HDTV: High Definition Television.
ITU-R: International Telecommunication Union for Radio communication.
MPEG: Moving Picture Experts Group.
VSB: Vestigial Side Band.
SDTV: Standard Definition Television.
ASK: Amplitude Shift Keying.
FEC: Forward Error Correction.
PSIP: Program and System Information Protocol.
PRBS: Pseudo-Random Binary Sequence.
RS: Reed-Solomon.
TS: Transport Stream.
IF: Intermediate Frequency.

Capítulo 1

Introducción

La radiodifusión es uno de los medios de comunicación masiva de gran uso en las sociedades. En la carrera de Ingeniería en Telecomunicaciones de la UNAM es una materia optativa dentro del módulo de salida llamado "Señales y Sistemas de Radiocomunicación". Esta materia puede ser cursada en el octavo o noveno y último semestre de la carrera. Actualmente la materia de radiodifusión basa su enseñanza en clases teóricas y en simulaciones de software, es por esto que se desea que cuente con un laboratorio que permita trasladar los conocimientos teóricos a la práctica trabajando con señales reales de radio y televisión.

Los programas PAPIME tienen como objetivo apoyar la innovación y los desarrollos que ayuden al mejoramiento de la enseñanza. Este trabajo se realiza dentro del proyecto PAPIME PE107415 el cual tiene como objetivo crear un banco de trabajo para realizar prácticas de radiodifusión. Para realizar dichas prácticas es necesario desarrollar transmisores y receptores de AM, FM y ATSC, así como un grabador y reproductor de señales de radio y televisión digital. A su vez parte del desarrollo realizado en esta tesis, en concreto el desarrollo del transmisor ATSC, puede ser utilizado en el proyecto Conacyt-CDTI número 189235, REFUTV, "Desarrollo de Redes en Frecuencia Única para Televisión Digital ATSC".

El objetivo de esta tesis es implementar tres transmisores que podrán ser usados en la materia de radiodifusión, involucrando las principales tecnologías que son utilizadas actualmente para dar servicio de radio y televisión, AM, FM y ATSC. Imaginar por un momento el costo que tiene un equipo transmisor de ATSC, que es

el nuevo estándar que adoptó México para la emisión de señales digitales de televisión, y el espacio que se tendría que destinar para este laboratorio, sería absurdo. Además comprar transmisores de AM, FM y ATSC no permitiría al futuro ingeniero experimentar con la tecnología en cuestión, solamente serviría para obtener buenas señales de AM, FM y ATSC. Es decir, es necesario contar con transmisores de costo accesible y que puedan ser reconfigurables para la experimentación, por estas dos razones los transmisores desarrollados en esta tesis son implementados en una plataforma llamada Radio Definida por Software (RDS) o en inglés Software Defined Radio (SDR).

SDR en los últimos años ha sido de gran interés para la investigación y la docencia, ya que en un mismo hardware es posible implementar diferentes sistemas de comunicación. Es decir, para la implementación de los transmisores se utiliza como hardware un equipo llamado USRP (Universal Software Radio Peripheral) que es un transceptor genérico y configurable, y como software se utiliza un programa llamado GNU Radio. El comportamiento de cada transmisor depende básicamente de cómo se programe en el software GNU Radio. De esta manera crear un transmisor AM, uno FM y otro ATSC no supone más costo de hardware que el del USRP, y controlar los transmisores mediante software permite que estos sean dinámicos en su configuración. Por lo que, es posible cambiar de un transmisor a otro solo con cargar un programa diferente y además se añade otra característica importante para el laboratorio, es posible utilizar valores diferentes a los predeterminados comercialmente en la emisión de las señales, lo que da lugar a la experimentación y pruebas.

Una ventaja más es que se pueden implementar con el mismo equipo y el mismo software futuras tecnologías de radiodifusión que surjan, esto es importante porque nos encontramos frente a un mundo donde la tecnología avanza y cambia rápidamente.

1.1. Justificación

Actualmente en la carrera de Ingeniería en Telecomunicaciones de la UNAM existen materias como Radiodifusión, que no cuentan con ningún tipo de material para realizar prácticas. Es importante empezar a crear material para ejemplificar

de forma práctica lo que se enseña en la teoría con el fin de producir un mejor aprendizaje en los estudiantes de ingeniería. Implementar transmisores y receptores reconfigurables sobre tecnologías de radiodifusión permite al estudiante observar las implicaciones que tiene cada parámetro del transmisor en la señal, por ejemplo, en su espectro, oscilograma, o incluso en el comportamiento de la señal en el receptor y a partir de ello comprobar y reforzar los conocimientos teóricos.

1.2. Objetivo de la Tesis

El objetivo general de esta tesis es diseñar e implementar transmisores AM, FM y ATSC de configuración dinámica, con tecnología SDR. Transmisores con interfaz gráfica para que el alumno pueda manipular los principales parámetros del transmisor de manera fácil y rápida. Como objetivo específico se tiene que diseñar cada transmisor de manera que permita al alumno comprender conceptos tales como el índice de modulación, ancho de banda, tasa de transmisión, frecuencia de transmisión, etc.

1.3. Organización de la Tesis

En primera instancia es necesario familiarizarse con el concepto de Radio Definida por Software esto implica entender cómo funciona esta tecnología. Después es imprescindible conocer el equipo y software que permiten la implementación de un sistema SDR. Una vez conseguido esto es necesario conocer las bases teóricas de cada una de las tecnologías que se implementarán: AM, FM y ATSC. Después es posible empezar el diseño de los transmisores y para ello se hace un estudio de los bloques que serán utilizados en los transmisores, cómo funcionan y cómo se configuran. Con todo lo anterior es posible la implementación de los transmisores.

Esta tesis se encuentra organizada en siete capítulos siguiendo el orden de temas explicado en el párrafo anterior. El capítulo que sigue a este primer capítulo (Introducción) es el capítulo dos en el que se describe la plataforma en la que se ha desarrollado el trabajo, es decir se explica cómo funciona la plataforma SDR y se da a conocer el software y hardware utilizado en el desarrollo del proyecto. En el

capítulo tres se describen conceptos básicos del procesamiento digital de señales que son necesarios para el diseño de los transmisores. El capítulo cuatro es la descripción del transmisor AM, se ilustra cómo ha sido diseñado e implementado; en el capítulo cinco se describe el transmisor FM y en el capítulo seis se describe el transmisor ATSC. Por último en el capítulo siete se resumen las conclusiones de la tesis.

Capítulo 2

La Radio Definida por Software

La tecnología SDR (Radio Definida por Software, RDS o en inglés Software Defined Radio, SDR) propone un modelo de sistema programable para la generación y recepción de señales RF, independiente del hardware, se trata de un sistema dinámico en el que el comportamiento del mismo depende del software [1]. Un sistema basado en esta plataforma es capaz de modificar su funcionamiento y características dinámicamente, como ancho de banda, modulación, codificación, etc., esto quiere decir que la forma de onda es modificada solo mediante software.

2.1. Historia

Inicialmente se utilizaban componentes analógicos en la construcción de transmisores o receptores independientemente de si la información era digital o analógica; es hasta el desarrollo de los microprocesadores en los años setenta y de los circuitos integrados de pequeña escala y gran escala en los sesenta, en inglés SSI (Small Scale Integration) y LSI (Large Scale Integration) respectivamente, que se empiezan a incluir circuitos digitales en transmisores y receptores de radio. En un principio estos circuitos integrados fueron usados para la codificación y decodificación de información digital y para el control de la frecuencia [2]. Sin embargo la posibilidad de tener un sistema de radio con procesamiento digital fue tangible hasta el desarrollo del DSP (Digital Signal Processor) y del FPGA (Field Programmable Gate Array), es hasta entonces que los sistemas de radio progresaron significativamente. Debido

a la velocidad de procesamiento y la facilidad con la que se configura un FPGA fue posible tener una plataforma SDR.

Cabe destacar que la Radio Definida por Software surge por una necesidad ya que tener transmisores y receptores reconfigurables es de gran eficiencia y aún más si estos pueden ser actualizados de forma remota. Es decir, cualquier cambio en un sistema de radio, un nuevo protocolo o norma, ajustes en la configuración de la red debido al tráfico o incluso algún fallo no es necesario adquirir un nuevo equipo para soportar los cambios. Además las modificaciones pueden hacerse de manera remota reduciendo idas al sitio y por consecuencia reduciendo aún más los costos. Otra necesidad que resuelve la SDR es debida a las comunicaciones multi-estándar necesarias para la interoperabilidad entre usuarios y que en caso de requerir un nuevo esquema de modulación o codificación se pueda descargar y así reconfigurar el sistema [2].

En la actualidad existen sistemas de radio militares basados en SDR como el programa Joint Tactical Radio System (JTRS ó Sistema de Radio Táctica Conjunta) [3]. El Sistema de Radio Táctica Conjunta es un programa del departamento de defensa de los Estados Unidos cuyo objetivo es desarrollar una familia de radios definidas por software para crear redes que permitan el envío y recepción de voz y vídeo en el campo de batalla de forma segura [4]. Un ejemplo es el sistema Thales AN/PRC-148 JEM que se encuentra en venta en producción en volumen. Este es un sistema totalmente SDR capaz de soportar diferentes formas de onda y modos de funcionamiento, es utilizado por las fuerzas de la OTAN alrededor del mundo y es desarrollado por Thales Communications [5].

Es interesante saber que en los últimos años ha surgido una extensión de la Radio Definida por Software que se llama Radio Cognitiva. Un receptor de radio cognitiva debe ser capaz de determinar a partir de las características temporales y espectrales de la señal el tipo de modulación, el formato de los datos e incluso la velocidad de los mismos. En el caso de un transmisor debe ser capaz de seleccionar la frecuencia de operación, el formato de trasmisión y la modulación más apropiada para cada trasmisión. Como un ejemplo, un receptor de radio cognitiva de televisión sería capaz de recibir la señal y determinar automáticamente si se trata de una señal NTSC, PAL, SECAM o ATSC y también sería capaz de procesarla automáticamente [6]. Como se

observa esta tecnología pretende avanzar a grandes pasos.

2.2. Arquitectura de un receptor o transmisor SDR

La arquitectura más simple de un receptor o transmisor SDR es la que se muestra en la figura 2.1 en la que se pueden observar filtros, convertidores de señal analógica-digital, digital-analógica y un procesador. A continuación se describe la diferencia entre la implementación de un receptor y un transmisor con respecto a esta arquitectura.

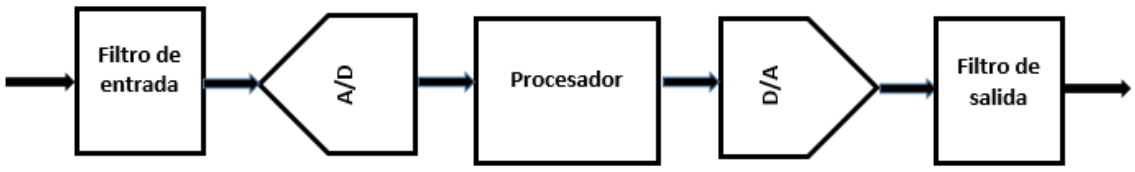


Figura 2.1: Arquitectura básica de un radio definido por software.

En el caso de la implementación de un receptor siguiendo la arquitectura de la figura 2.1 el filtro de entrada debe ser un filtro paso banda con frecuencia central muy cercana a la frecuencia que se desea recibir, el siguiente bloque es un convertidor analógico-digital por lo que la frecuencia de muestreo de este debe ser por lo menos el doble del ancho de banda de la señal recibida, después la señal pasa al bloque procesador donde es obtenida la señal banda base y demodulada. El bloque que sigue es el convertidor digital-analógico cuya frecuencia de muestreo debe ser al menos el doble del ancho de banda de la señal banda base por el teorema de Nyquist. Finalmente el filtro de salida debe ser un paso bajas para recuperar la señal original en banda base.

Si se requiere implementar un transmisor, el filtro de entrada debe ser un paso bajas para limitar a la señal banda base, el convertidor analógico-digital deberá muestrear con una frecuencia que sea por lo menos el doble del ancho de banda de la señal banda base original. Después de que la señal es procesada y modulada, el

convertidor digital-analógico tiene una frecuencia de muestreo de por lo menos el doble del ancho de banda de la señal a transmitir.

En el caso de que la información en banda base sea por naturaleza digital se elimina el A/D con su respectivo filtro en el caso del transmisor y lo mismo se hace para la información en banda base en el receptor.

Cabe destacar que esta arquitectura es independiente del tipo de modulación final, así cualquier esquema de modulación se puede implementar con el mismo hardware ya sea analógico (AM o FM) por ejemplo o digital (p.e. BPSK o QAM).

Es necesario que se utilicen procesadores lo suficientemente rápidos para lograr aplicaciones en tiempo real. De la misma forma también se requieren convertidores A/D y D/A pero las velocidades de conversión requeridas no pueden ser alcanzadas a un costo razonable por lo que han surgido implementaciones en las que es posible trabajar con convertidores más lentos. Un caso de este tipo de implementación alternativa es el modulador y demodulador en cuadratura que se ha convertido en un estándar a la hora de implementar la plataforma SDR. En la figura 2.2 se muestra un esquema de este modulador en cuadratura [2].

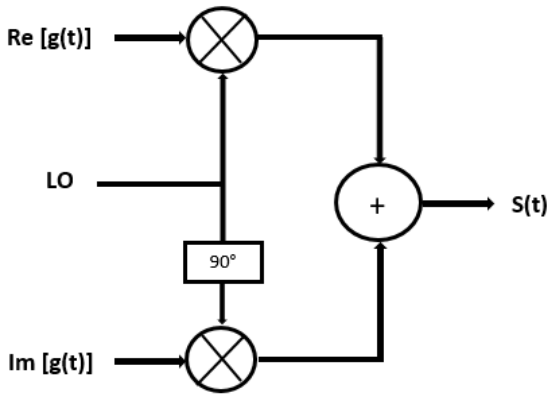


Figura 2.2: Esquema de un modulador en cuadratura.

Con esta implementación los dos convertidores D/A utilizados para generar una señal en fase y otra en cuadratura solo requieren trabajar a la frecuencia de Nyquist de la señal modulada en banda base . Ahora bien si se habla del receptor, un demodulador en cuadratura también es utilizado para reducir la frecuencia de muestreo,

el demodulador mezcla la señal de entrada con la señal del oscilador local una en fase y la otra desplazada 90 grados, y de esta manera pueden ser generadas las componentes I y Q de la señal banda base. Hay que señalar que para demodular AM solo se requiere la componente I, pero para las demás modulaciones se requieren las dos componentes.

Pero también surge como alternativa al demodulador de cuadratura un demodulador como el mostrado en la figura 2.3 que es más simple y no es tan sensible a las variaciones de cada una de las componentes, es el detector Tayloe.

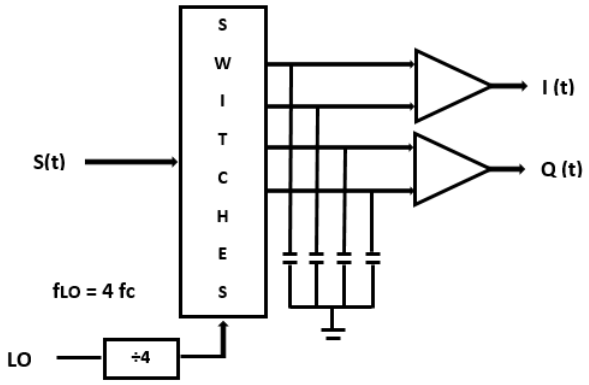


Figura 2.3: Esquema del detector de Tayloe.

Este detector requiere que la frecuencia del oscilador local sea cuatro veces la frecuencia de la señal portadora, la señal del oscilador local es decodificada y conducida a cuatro interruptores RF, las salidas de los interruptores conducen a amplificadores operacionales que están configurados para generan las componentes I y Q. El detector de Tayloe tiene varias ventajas pues no requiere circuitería analógica y por lo tanto el límite de la frecuencia de portadora que se puede recibir se ve limitada solamente por los circuitos digitales. Otra ventaja es que la frecuencia del oscilador local puede ser fácilmente modificada obteniendo así la posibilidad de implementar un receptor sintonizable [2]. Todo esto nos da una idea de cómo se han implementado los convertidores de señal en SDR.

Una vez que se analizó de forma general cómo trabaja una arquitectura SDR es posible comprender una configuración más completa de SDR, como la mostrada en

la figura 2.4 , la cual consta de 5 partes: arquitectura de bus, antenas inteligentes, convertidores A/D y D/A de alta velocidad, procesadores de señal o DSP y paquetes de software [7].

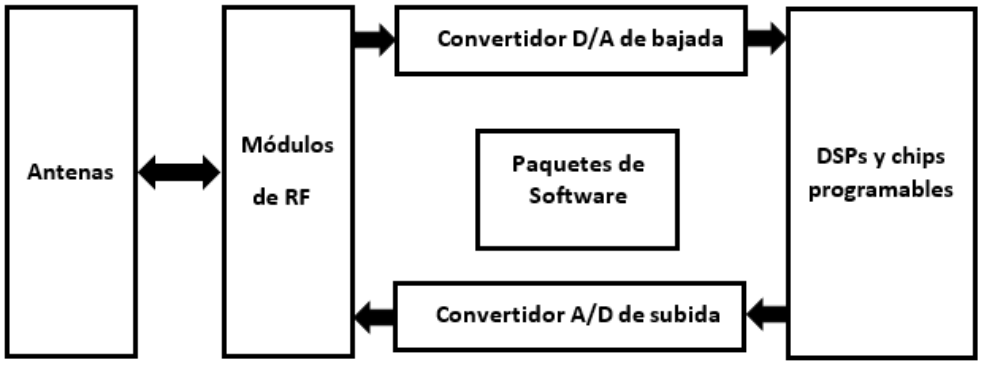


Figura 2.4: Configuración de un SDR.

- Arquitectura de bus

Esta arquitectura, conocida también como arquitectura abierta y en general este tipo de arquitectura es llamada así debido a que es adaptable al usuario. En un hardware SDR es posible acceder directamente a la placa madre (específicamente al FPGA o DSP del hardware SDR) mediante un puerto que nos ofrece. Mediante este puerto es posible conectar dispositivos o tarjetas que el usuario requiera [8]. La tecnología SDR emplea este tipo de arquitectura para satisfacer las necesidades de una red inalámbrica definida por software. Esta arquitectura puede interconectar varios módulos de hardware logrando altas velocidades de transmisión [7]. La arquitectura de bus es definida por la ISO (International Organization for Standardization) y su estructura consiste en tres capas. La capa de interfaz de radio, capa de configuración y la capa de proceso. La capa de interfaz de radio, es como su nombre lo dice la interfaz entre el hardware y el exterior. Esta capa se encarga de coordinar la información de entrada y salida, contiene la información de los algoritmos y las funciones que son necesarias para el sistema. Envía mensajes a la capa de configuración especificando qué algoritmos debe correr el sistema y el orden en que deben ser realizados. La

capa de configuración almacena la información recibida por la capa de interfaz y es la responsable de la construcción de los algoritmos. La capa de configuración por lo tanto envía mensajes a la capa de proceso indicando los comandos que tienen que ser ejecutados para realizar los algoritmos. Finalmente la capa de proceso implementa la función que se encuentra almacenada, es decir realiza los cálculos físicos que le fueron indicados por la capa de configuración [9]. Actualmente los buses más comunes utilizados como sistema de arquitectura abierta son el PCI (Peripheral Component Interconnect) y VME (Versa Module Eurocard). Estos buses también son llamados buses de expansión debido a que, como se mencionó anteriormente, se encargan de conectar la placa principal de un sistema con dispositivos adicionales [10]. El dispositivo utilizado en esta tesis (USRP), el cual se describirá más adelante, cuenta con un puerto físico PCIe (Peripheral Component Interconnect Express) y a este puerto es posible conectar directamente, por ejemplo, una computadora para alcanzar altas velocidades entre el hardware SDR y la PC (latencias de ida y vuelta entre los dos dispositivos de hasta $10\mu s$ [11])

- Antenas inteligentes

Integrar antenas inteligentes y tecnología SDR es de gran interés en la actualidad para incrementar la calidad en un enlace inalámbrico. Un ejemplo de esto es un prototipo que ha sido implementado en la Universidad de Brasilia, el cual puede ser capaz de estimar la dirección con la que llega la señal a partir de un arreglo de cuatro antenas [12] utilizando tecnología SDR. La serie de USRPs tiene la capacidad de implementar sistemas de múltiples antenas (antenas inteligentes) o sistemas MIMO de una manera fácil y de bajo costo. Es posible conseguir sistemas MIMO de NxM [13]. Las antenas inteligentes son controladas mediante el procesador de la señal digital de banda base es decir son controladas por el DSP.

- Convertidores A/D y D/A

Como se pudo intuir anteriormente una parte esencial de un sistema de radio definido por software son los convertidores de señal digital-analógico y viceversa, por lo que es importante detallar su rol en el sistema. En un receptor SDR la señal de RF es convertida en una señal de frecuencia intermedia (IF) a través de mezclador

de frecuencia sintonizable mediante software, después el convertidor A/D digitaliza la señal de frecuencia intermedia. En el caso de un transmisor la señal digital es convertida a su correspondiente señal analógica pero de frecuencia intermedia. Posteriormente mediante un mezclador de igual manera sintonizado por software se consigue la señal a la frecuencia de RF deseada [14]. Todo esto se puede comprender mejor mediante la figura 2.5, donde se observa que en la placa hija se encuentra el primer mezclador para la frecuencia intermedia, después de esto la señal es digitalizada por el convertidor A/D de la placa madre del USRP y finalmente las muestras son enviadas al FPGA quien se encarga de obtener la salida exacta de frecuencia y frecuencia de muestreo [15].

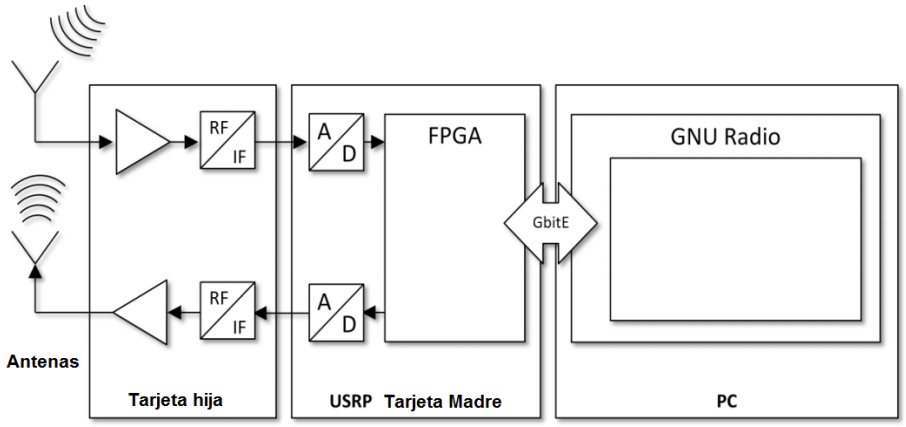


Figura 2.5: Estructura de un SDR basado en un USRP y tarjeta hija.

- Procesador de señales

El procesador es el dispositivo principal en un hardware de SDR, encargado de completar el procesamiento en banda base de las señales, provee la modulación y demodulación de las señales, la sincronización, codificación y decodificación. No solo es posible utilizar DSPs, si no también chips programables como los ASIC (Application Specific Integrated Circuit), los FPGA (Field-Programmable Gate Array), o incluso ambos [7]. En un USRP el FPGA es utilizado como dispositivo principal. Un FPGA explota el procesamiento de datos en paralelo de diferentes formas, por ejemplo al tener una secuencia de bloques programados, el FPGA es capaz de ejecutar múltiples operaciones por bloque además todos los bloques operan de manera simultánea. Su

rendimiento se ve limitado por el número de compuertas que tiene y su frecuencia de reloj. Los FPGA más recientes implementan funciones para realizar tareas que lleva a cabo un DSP pero con mayor eficiencia. Cuando un DSP tiene un diagrama de bloques programados ejecuta los bloques de manera secuencial, en paquetes de un número determinado de muestras, por lo que se ve limitado por su frecuencia de reloj y el número de operaciones que realiza por ciclo [16]. En general utilizar uno u otro depende de la aplicación, el DSP no es lo más óptimo cuando la aplicación tiene una alta demanda de tareas, es por esto que los FPGA son recientemente preferidos por la tecnología SDR.

- Paquetes de software

SDR tiene una arquitectura de procesamiento distribuido en el sentido de que se pueden combinar software y hardware de diferentes vendedores. Utiliza CORBA (Common Object Request Broker Architecture). CORBA es un estándar definido por una organización internacional sin fines de lucro llamada OMG (Object Management Group) fundada desde 1989 [17]. Este estándar es utilizado en sistemas distribuidos y permite la interoperabilidad entre diferentes lenguajes de programación máquinas y productos. La arquitectura CORBA está orientada a objetos por lo que brinda un mecanismo mediante el cual un objeto se puede comunicar con otro independientemente de donde esté situado. Los objetos pueden estar o no en el mismo programa o maquina [18]. CORBA consta de paquetes de control, paquetes de alto nivel y paquetes de interfaz del sistema que le permiten tener portabilidad, es decir software que se puede obtener mediante una descarga [7].

2.3. El software en SDR

Los programas que deben ser cargados a la tarjeta o FPGA pueden ser creados en un lenguaje de diagrama de bloques como Simulink, Labview o GNURadio; lo que facilita mucho la programación de un radio definido por software. Una vez creado el diagrama de bloques, éste es convertido a código C, compilado y cargado al procesador. A continuación se describe el software GNU Radio de libre distribución.

2.3.1. GNU Radio

GNU Radio es el software utilizado para desarrollar este proyecto, es un software libre de código abierto y consta de un conjunto de herramientas que proporcionan bloques de procesamiento de señales para desarrollar radios de software. GNU Radio comúnmente se instala en Linux, que es un sistema operativo también de código abierto y de libre distribución pero también es posible instalarlo en el sistema operativo Windows. Los desarrolladores de este software están haciendo esfuerzos para que el uso de su software sobre Windows sea cada vez sea más sencillo. En la actualidad la ejecución y compilación de GNU Radio en Windows puede no ser simple ya que GNU Radio se desarrolló basado en el sistema operativo Linux [19].

GNU Radio tiene una licencia GPL (GNU General Public License), esta licencia garantiza a los usuarios finales la libertad de usar, estudiar, compartir (copiar) y modificar el software [20]. Como ya se ha mencionado anteriormente la programación en una plataforma RDS es relativamente simple debido a que se programa en un ambiente de diagrama de bloques. GNU Radio consta de una variedad de bloques agrupados de acuerdo a su función, a continuación se mostrarán los más relevantes:

- Bloques de control de nivel

En este grupo se encuentran los bloques de control de ganancia y potencia principalmente.

- Bloques generadores de señal

Son los bloques generadores de formas de onda (seno, coseno, etc.), generadores de ruido, generador de señal aleatoria entre otros.

- Bloques para crear variables

Con estos bloques se declaran las variables y parámetros que serán utilizados en el programa.

- Bloques de Audio

Los bloques llamados “fuente” o de entrada de datos permiten tomar información de un archivo con formato WAV o bien toman la información desde el micrófono de la

computadora. Los bloques llamados "sumidero" o bloques de salida de datos pueden guardan la información en archivos con formato WAV o enviar la información al micrófono.

- Bloques Operadores Booleanos

Implementa operadores lógicos (AND, OR, NOT, XOR) que permiten realizar búsquedas complejas.

- Bloques Operadores de Archivos

Estos permiten tomar archivos como entrada al programa o guardan información en un archivo. Es posible utilizar cualquier extensión o formato e incluso solo el nombre del archivo.

- Operadores Matemáticos

Estos permiten implementar operaciones matemáticas como suma, multiplicación, división, sustracción, logaritmos, etc.

- Operadores de Cadenas de bits

Aquí se encuentran los bloques que permiten intercalar datos o repiten datos (cadenas de bits).

- Bloques convertidores

Se trata de bloques que convierten un tipo de dato a otro, se manejan datos de tipo flotante, entero, entero corto, complejo, etc.

- Bloques de filtros

Son bloques que implementan filtros paso bajas, filtros pasa banda, paso altas, también se encuentra "remuestreadores" de señal que permiten ajustar la velocidad de muestreo de la señal entre dos bloques.

Los bloques también se encuentran agrupados de acuerdo a la tecnología de radiodifusión por lo que existen bloques especiales para las diferentes tecnologías que existen (AM, FM, televisión digital, etc.) como moduladores, demoduladores, decodificadores, etc.

En la figura 2.6 se muestra el entorno de programación de GNU Radio, es posible observar en el lado derecho de la figura los bloques descritos antes, esta ventana nos permite agregar los bloques. En el lado superior de la figura se observa una barra de herramientas, en general esta barra permite cargar programas y ejecutarlos, eliminar y copiar bloques. En la parte inferior se observa la consola de GNU Radio que sirve para monitorear la ejecución de un programa. El centro es el espacio en el que se agregan y se configuran los bloques, es el espacio de trabajo. Al abrir el programa por defecto se crea un bloque llamado $top-block$ el cual es el objeto físico que contendrá toda la programación a bloques que se desarrolle.

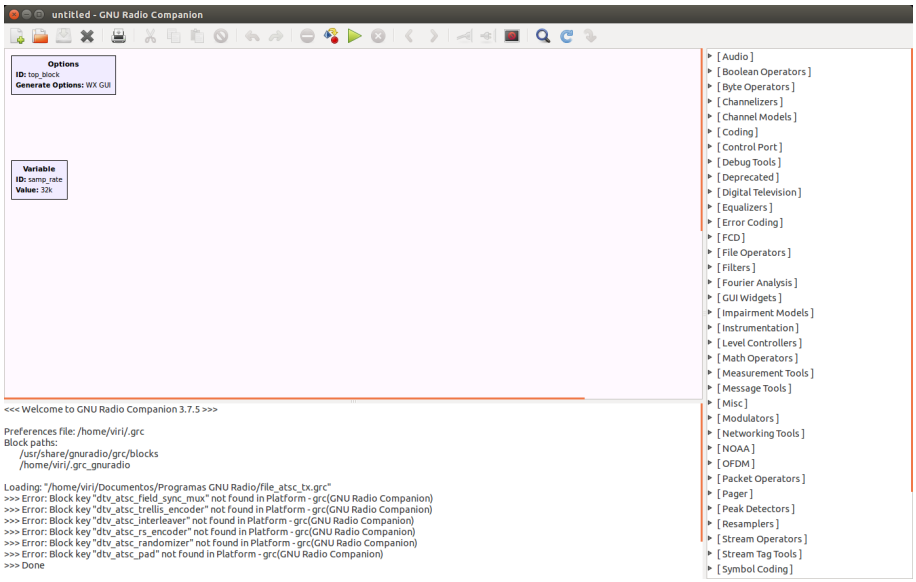


Figura 2.6: Interfaz del software GNU Radio.

GNU Radio trabaja con un concepto llamado ítem [20]. En GNU Radio no todos los bloques procesan muestras, algunos bloques pueden trabajar con bits, bytes, símbolos, paquetes, etc. Al elemento con el que funciona cada bloque se le llama ítem y en cada bloque se específica el tipo de ítem. A continuación se encuentra la nomenclatura utilizada para cada tipo de ítem:

c: complejo (flotante de precisión simple)

z: complejo (flotante de doble precisión)

f: flotante de precisión simple

d: flotante de doble precisión

i: entero (32 bits)
s: entero corto (16 bits)
b: bits o bytes
v: vector de ítems
Cabe resaltar que el programa solo se interesa por el número de bytes por ítem. Cada bloque decide si los ítems se procesan empaquetados o individualmente, el usuario solo tiene que preocuparse porque que el tipo de ítem de un bloque a otro, sea el mismo.

Python

Python es un lenguaje de programación libre y gratuito. Es de alto nivel y orientado a objetos. Es interpretado, es decir se compila línea por línea. La compilación es la traducción del código escrito por el programador a instrucciones ejecutables. Este lenguaje es multiplataforma ya que funciona en casi cualquier sistema operativo [21].

Python es utilizado por GNU Radio como un lenguaje de script [20], esto quiere decir que es utilizado para ejecutar los programas y funciones del sistema. El código del programa se encuentra en C++ y Python es utilizado para ejecutar simultáneamente los programas. En general Python no se usa para el procesamiento de señales en tiempo de ejecución. Cuando se quiere crear un nuevo bloque en GNU Radio se puede programar en Python y son los llamados "Python blocks", sin embargo también es posible programar los bloques mediante C++. Elegir entre un lenguaje de programación u otro depende del objetivo del bloque a crear [20].

2.4. El hardware en SDR

Una limitación en hardware que se presenta en la placa madre de algunos SDRs, es que ésta no cuenta más que con un par de convertidores, uno A/D y el otro D/A, lo que hace imposible la implementación de un receptor y transmisor simultáneamente. Lo anterior puede ser resuelto utilizando una placa hija que cuente con otro par de convertidores, esta solución ha sido la más utilizada por los desarrolladores de equipos SDR. Existen diferentes equipos para implementar sistemas SDR los más conocidos son USRP, HackRF, RTL-SDR y Nuand Blade RF. El USRP implementa

como solución al problema del número de convertidores descrito anteriormente una placa hija, cabe destacar que existen diferentes tipos de placas hija pero aquí solo se describirá la utilizada en esta tesis. A continuación se describirán las características importantes de todo el equipo o hardware utilizado en el desarrollo de los transmisores de esta tesis: un USRP X300, una placa hija WBX y un CPU portátil o Laptop.

2.4.1. USRP

El USRP (Universal Software Radio Peripheral) es un equipo que permite implementar radios definidos por software, ha sido utilizado en numerosos proyectos de todo tipo de SDR [22]. El USRP es fabricado por la compañía americana Ettus Research de NI (National Instruments). En este proyecto se utiliza la serie de USRP X300, la apariencia de este equipo es mostrada en la figura 2.7.

Figura 2.7: Vista superior del USRP X300.

Algunas características generales del USRP X300 son:

- Utiliza una FPGA para un mejor rendimiento, el Xilinx Kintex-7 FPGA y más específicamente la serie XC7K325T. Ésta cuenta con un reloj de 200 MHz y 200 MS/s [1] de ancho de banda para cada canal de 16 bits.
  Para realizar el diseño y la implementación de sistemas electrónicos en el FPGA

[1] 10^6 muestras por segundo

de un USRP se utiliza un lenguaje llamado Verilog. Éste lenguaje fue estandarizado por la IEEE. Verilog es un lenguaje de descripción de hardware o en inglés HDL (Hardware Description Language). Un lenguaje de descripción de hardware sirve para precisar el comportamiento de un circuito electrónico haciendo posible su simulación y análisis. Este lenguaje es compilado mediante Quartus II un compilador de libre uso, por lo que es posible cargar y compilar otros códigos Verilog al FPGA de un USRP [23].

- Cuenta con múltiples interfaces de alta velocidad: Dual 10 Gigabit Ethernet (200 MS/s Full Duplex), PCIe (200 MS/s Full Duplex), ExpressCard (50 MS/s Full Duplex), Dual 1 Gigabit Ethernet (25 MS/s Full Duplex).

- Arquitectura UHD (USRP Hardware Driver) que permite la compatibilidad con diferentes softwares de RDS.

- Arquitectura flexible en cuanto al reloj del USRP.

El USRP logra la sintonización en dos etapas:

1. RF Front-end: en esta etapa la frecuencia de RF es trasladada a una frecuencia intermedia o frecuencia IF (intermediate frequency).

2. DSP: traslada de la frecuencia IF a la frecuencia banda base.
   Típicamente el usuario es quien fija una frecuencia central. La etapa de RF front- end se sintoniza a una frecuencia lo más cercana posible a tal frecuencia central. Después el DSP elimina el error que existe entre la frecuencia sintonizada por el RF front-end y consigue la frecuencia definida por el usuario.

Es posible que el usuario pueda controlar estas dos etapas de sintonización [24]. El USRP utiliza dos mezcladores de señal también llamados procesadores Font-end los cuales realizan la conversión analógica a digital y digital a analógica de las señales [25].

Respecto a la frecuencia de muestreo es importante entender ciertas reglas:
Los parámetros de interpolación y decimación utilizados deben ser estrictamente enteros. Para conseguir esto se debe cumplir que la relación entre la frecuencia del reloj maestro del USRP y la frecuencia deseada dé como resultado un entero. En el

USRP X300 es posible elegir de entre tres frecuencias de reloj maestro con el objetivo de conseguir otras frecuencias de muestreo, el reloj maestro de este hardware puede soportar los valores de 200 MHz, 184.32 MHz y 120 MHz [24].

En la figura 2.8 se observa la estructura interna del USRP X300, figura tomada de la hoja de especificaciones proporcionada por la compañía NI [26]. En el apéndice A se anexan todas las especificaciones de la serie X (X300 y X310).

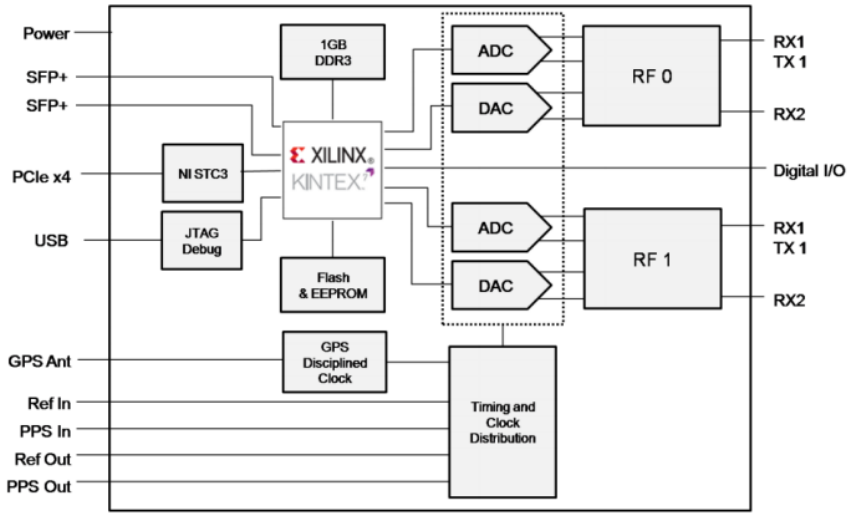


Figura 2.8: Estructura interna del USRP X300.

2.4.2. La placa hija WBX

La tarjeta hija utilizada en la implementación de los transmisores es la WBX la cual trabaja en el rango de los 50 MHz a 2.2 GHz, con una ancho de banda de 40 MHz tanto en transmisión como en recepción.

Las principales características de la tarjeta WBX son:

- Proporciona una potencia de salida de hasta 100 mW y un factor de ruido de 5dB.

- Opera en full dúplex a diferentes frecuencias.

- Los osciladores locales de transmisión y recepción son independientes pero pueden ser sincronizados para trabajar con sistemas MIMO (Multiple-input Multiple-output).
- Cuenta con dos entradas/salidas, una que puede ser utilizada tanto para transmitir como recibir (TX/ RX) y otra que solo permite recepción (RX).

2.4.3. El CPU

Lógicamente se necesita de una maquina o CPU para ejecutar un sistema operativo y por lo tanto utilizar el software GNU Radio. La computadora utilizada en este proyecto es marca DELL modelo Vostro, sus características son las siguientes:

- Procesador de gama alta i7-3632: 4 núcleos, 8 subprocesos, memoria cache de 6MB, frecuencia básica del procesador 2.2 GHz y frecuencia turbo máxima de 3.2 GHz.
- Memoria de 8.042 GB. Disco duro de 750 GB.
- Sistema operativo: Ubuntu 14.04.2 LTS. Versión de GNU Radio utilizada: 3.7.9.1 y versión del controlador UHD (USRP Hardware Driver) 3.7.10.

En la figura 2.9 se observa el diagrama de conexiones utilizado en este proyecto. El USRP y la computadora se comunican a través de una interfaz de 1 Gb Ethernet (25 MS/s Full Duplex) [26]. En la laptop es ejecutado y compilando el software GNU Radio encargado de obtener las muestras que se envían al USRP a partir del diagrama de bloques programado.

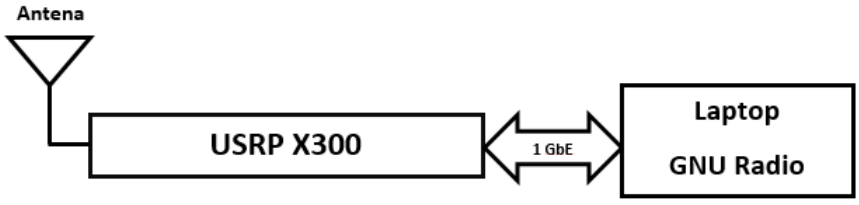


Figura 2.9: Diagrama de conexiones para la implementación de los transmisores en SDR.

Capítulo 3

Conceptos básicos del procesamiento digital de señales

La mayoría de las señales directamente encontradas en la ciencia y la ingeniería son señales continuas, es decir, son señales analógicas. Los procesos de conversión de señales analógica a digital o digital a analógica permiten que los equipos digitales puedan trabajar con estas señales continuas. Este es el caso de un transmisor definido por software, ya que la señal de información puede ser la voz de naturaleza continua, por lo que se requiere un convertidor de señal analógica a digital y una vez procesada y modulada la información se requiere radiar, y la única forma es usando un convertidor digital a analógico.

Una señal digital ha pasado por un proceso de muestreo y cuantización. Estos procesos limitan la información que puede contener la señal digital, es por esto que es importante entender cuánta información es necesaria y cuánta es posible perder en la conversión [27]. A continuación se explicarán conceptos clave del procesamiento digital de señales que son imprescindibles para entender el funcionamiento de GNU Radio y para el diseño de los transmisores.

3.1. El teorema de muestreo

El teorema de muestro fue postulado por Harry Nyquist en 1928 y demostrado matemáticamente por Claude Shannon en 1949 es por esto que también se le puede

encontrar como el Teorema de Nyquist o Teorema de muestreo de Shannon [28]. El teorema de muestreo indica que una señal puede ser correctamente muestreada, si no contiene componentes de frecuencia por encima de la mitad de la frecuencia de muestreo [27], esto es:

$$f_{max} \leq f_s/2, \tag{3.1}$$

Donde f_{max} es la frecuencia máxima de la señal continua y f_s es la frecuencia de muestreo. Dicho de otra manera la frecuencia de muestreo debe ser a lo menos el doble de la frecuencia máxima que contiene la señal que se desea muestrear.

$$f_s \geq 2 * f_{max}. \tag{3.2}$$

Un muestreo se ha realizado de manera correcta si a partir de las muestras tomadas es posible reconstruir la señal original. Cuando no se cumple el teorema de muestreo las muestras tomadas no representan a la señal original analógica si no que éstas muestras pueden estar representando a otra señal analógica. Este fenómeno es conocido como "aliasing" debido a que la señal toma otras frecuencias, podría decirse otras identidades (alias) durante la conversión [27]. Una forma de entender este fenómeno es en el dominio de la frecuencia.

En el dominio del tiempo el muestreo de una señal se logra multiplicando ésta por un tren de impulsos de amplitud unitaria como el mostrado en la figura 3.1 A). El espectro de un tren de impulsos es periódico y las espigas ocurren en la frecuencia de muestreo y sus múltiplos (f_s, $2f_s$, $3f_s$, $4f_s$, etc.), éste es ejemplificado en la figura 3.1 B).

La multiplicación de dos señales en el dominio del tiempo equivale a una convolución en el dominio de la frecuencia, por lo que el espectro original se duplica en cada una de las frecuencias múltiplos de la frecuencia de muestreo. Esto se muestra en la figura 3.1 C) donde se observa el espectro de una señal que ha sido muestreada cumpliendo con el teorema de Nyquist, a partir de este espectro es posible recuperar la señal original. La figura 3.2 A) corresponde con el espectro de la señal analógica antes del muestreo. Cuando no se ha realizado un muestro que cumpla con el teorema de Nyquist ocurre lo mostrado en la figura 3.2 B), los espectros duplicados durante el muestreo se traslapan, lo que hace imposible recuperar la señal original.

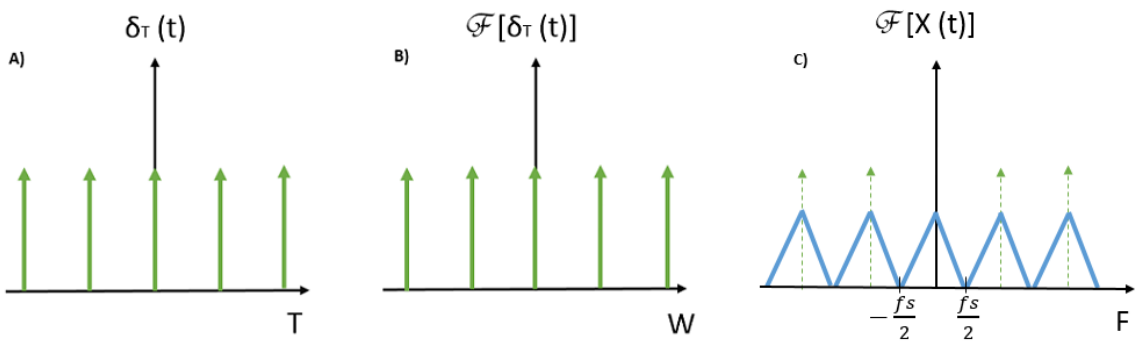


Figura 3.1: A) Tren de impulsos en el dominio del tiempo. B) Espectro de un tren de impulsos. C) Espectro de una señal con un muestreo correcto.

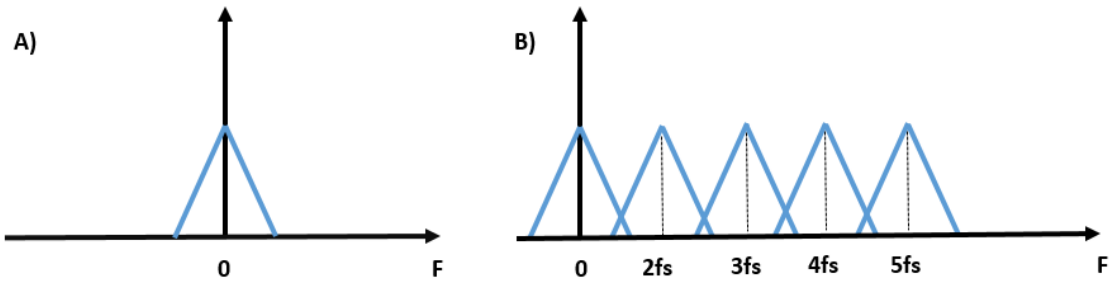


Figura 3.2: A) Espectro de la señal antes del muestreo. B) Espectro de la señal después de un muestreo incorrecto.

Es necesario resaltar que para cumplir con este teorema debe existir un filtro paso bajas antes del convertidor analógico-digital, con lo que se evita que entren al convertidor frecuencias mayores a la mitad de la tasa de muestreo. Algunos filtros utilizados en este tipo de aplicación son el filtro Chebychev, el filtro Bessel y el filtro Butterworth [27].

3.2. Modificación de la frecuencia de muestreo

En muchas ocasiones es necesario cambiar la frecuencia de muestreo varias veces en el mismo sistema, estos sistemas son conocidos como sistemas de tasa múltiple. Cuando una señal analógica es muestreada a una tasa mucho mayor que el doble de su frecuencia máxima es posible que el filtro que se utilice antes del convertidor

sea mucho más simple, por ejemplo un filtro paso bajo RC [27]. Los datos digitales obtenidos tendrán la información de la señal original pero también información innecesaria. Debido a esto se requiere ahora eliminar estas muestras, es posible hacer esto simplemente descartando las muestras innecesarias, proceso llamado diezmado o también llamado decimación. Cuando el proceso para modificar la frecuencia de muestreo, en vez de descartar muestras, añade muestras con valor de 0, es llamado interpolación.

Estas técnicas son utilizadas porque, como se pueden inferir de lo anterior, permiten sustituir componentes analógicos mediante software y en el caso de algunas aplicaciones mejoran el rendimiento de los sistemas [27]. En el software GNU Radio existen bloques entre los que es imposible trabajar con la misma frecuencia de muestreo por lo que la interpolación y la decimación como técnicas de conversión de tasa de muestreo son necesarias.

3.2.1. Definición de interpolación y decimación

La interpolación es el incremento de la frecuencia de muestreo por un factor entero llamado L. Este procedimiento se debe llevar acabo sin modificar el contenido espectral de la señal original (sin modificar las muestras reales de la señal), para lograr esto se añaden L-1 muestras con valor de cero entre las muestras sucesivas de la señal. Con lo anterior el espectro de la señal se mantiene pero aumenta la frecuencia de muestreo. En el dominio de la frecuencia este procedimiento produce una repetición del espectro de la señal en la banda que es de interés [29]. Después se requiere eliminar las repeticiones indeseadas con un filtro digital.

La decimación es la reducción de la frecuencia de muestreo por un factor entero M. Para realizar esto lo más sencillo es quedarse con una de M muestras. Por lo general este proceso causa “aliasing”, para evitar esto se utiliza un filtro paso bajas previo al proceso de decimación.

Filtro interpolador y diezmador

Antes de llevar a cabo un decremento de la frecuencia de muestreo se debe limitar el ancho de banda de la señal con respecto al factor de diezmado, para evitar los efectos indeseables comentados anteriormente (aliasing), este filtro previo es un filtro

diezmador. Por el contrario el filtro interpolador se coloca después de que se haya llevado a cabo el proceso de interpolación. Éste permite recuperar la señal original sustituyendo las muestras añadidas con valor de cero por valores más adecuados que se obtienen mediante la interpolación entre las muestras reales.

Al conjunto formado por el filtro diezmador y el dispositivo que disminuye la frecuencia de muestreo se le denomina diezmador y de igual manera un interpolador se compone de un filtro y el dispositivo que incrementa la frecuencia de muestreo [29].

3.2.2. Interconexión de sistemas de procesado de tasa múltiple

Un sistema de procesado que permita la conversión de tasa de muestreo está formado por la interconexión de interpoladores, diezmadores, y por los filtros digitales que evitan el "alisaing" en el diezmado y eliminan los espectros repetidos causados por la interpolación. La forma en que estos se encuentren interconectados influye directamente en la carga computacional del algoritmo [29]. En general la conexión en cascada de un interpolador (L) y un diezmador (M) no es conmutativa, a menos que M y L sean números enteros primos. Si se observa la figura 3.3 entonces la salida y1 es igual a la salida y2 si M y L son primos.

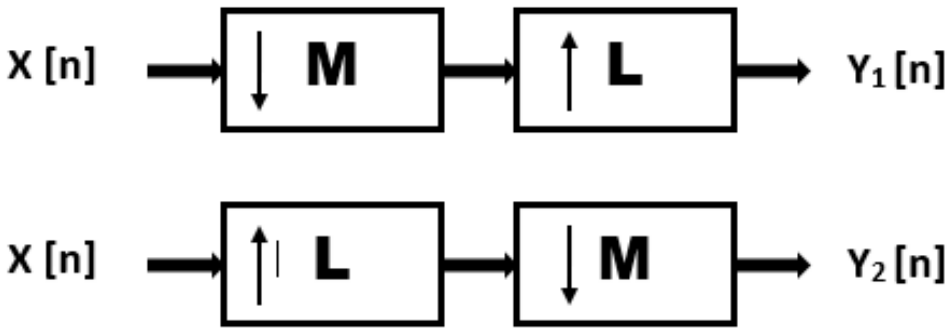


Figura 3.3: Interpolador y diezmador en cascada.

3.3. La transformada rápida de Fourier

Para comprender qué es la transformada rápida de Fourier es necesario hablar en primer lugar de la transformada discreta de Fourier o DTF (Discrete Fourier Transform). La DTF es un caso particular de la transformada de Fourier. La transformada de Fourier es utilizada para tener un análisis espectral de la señal o en otras palabras para conocer características en el dominio de la frecuencia. La transformada discreta se utiliza en el caso de que la señal sea discreta con un número finito de muestras. Sin embargo, implementar directamente la transformada discreta de Fourier a través de un algoritmo ocasiona una eficiencia computacional muy baja [30].

En 1965 se desarrolló un algoritmo conocido como FFT por la necesidad de reducir el tiempo de computación de la transformada discreta [31], el crédito lo tienen los americanos J.W. Cooley y J.W. Tukey. La transformada rápida de Fourier o FFT (Fast Fourier Transform) produce los mismos resultados que la DTF pero su eficiencia es mucho mayor ya que reduce el tiempo de cálculo significativamente, hasta cientos de veces [27]. Un cálculo de FFT toma $N(log2N)$ operaciones mientras que un cálculo de DTF toma N^2 operaciones, donde N es el número de muestras utilizadas para el cálculo. Cuando el número de muestras es reducido la diferencia no es muy relevante pero cuando el número de muestras es muy alto utilizar la DTF se vuelve absurdo. Por ejemplo si N= 4096, el número de operaciones con DTF es de aproximadamente 17 millones mientras que con la FFT se efectúan aproximadamente 50 000 operaciones.

Por las características mencionadas en este subcapítulo el software GNU Radio implementa en los bloques que tienen como objetivo mostrar el comportamiento de una señal en el dominio de la frecuencia el algoritmo FFT en vez del algoritmo DTF.

3.4. Filtros digitales FIR. Ventanas

Los filtros digitales se utilizan con el propósito de separar señales que se han combinado o con el fin de restaurar una señal que se ha deformado por algún motivo. En una comparación entre los filtros analógicos y digitales, los dos son utilizados de la misma forma pero los filtros digitales pueden conseguir resultados superiores a los analógicos [27].

La forma más sencilla de implementar un filtro digital es mediante la convolución de la señal de entrada y la respuesta al impulso del filtro. Cabe recordar que la respuesta al impulso de un sistema permite determinar la información completa del sistema, midiendo la respuesta de éste al excitarlo con una función de impulso unitario [32]. Otra forma de implementar filtros digitales es mediante recursividad [27]. En un filtro recursivo la salida actual depende tanto de las entradas pasadas y presentes como de las salidas pasadas. La respuesta al impulso de un filtro recursivo es infinita por lo que también son llamados filtros IIR (Infinite Impulse Response). Por el contrario un filtro no recursivo tiene una respuesta al impulso finita y es llamado también filtro FIR (Finite Impulse Response). En los filtros no recursivos la salida actual depende únicamente de entradas pasadas y presentes.

Los filtros FIR son de especial interés debido a que son inherentemente estables y no conllevan distorsión de fase [28]. Este tipo de filtros son los que han sido implementados en el software GNU Radio. Uno de los métodos para el diseño de filtros FIR es el de ventanas. Método que también es implementado en algunos bloques de GNU Radio por lo que es importante su estudio para el desarrollo de esta tesis. Este método se basa en el truncamiento de la respuesta al impulso infinita de un filtro ideal. De forma general los pasos de este método son los siguientes:

- Se obtiene la respuesta al impulso del filtro ideal que se desea diseñar (filtro paso bajas, filtro paso banda, etc.).
- Se "Enventana" o trunca dicha respuesta al impulso.
- Se desplaza dicha respuesta un número adecuado de muestras [33] (cuanto sea necesario).

En la figura 3.4 se observa la respuesta al impulso de un filtro ideal en el dominio del tiempo mientras que en la figura 3.5 se observa la misma función pero ya truncada y desplazada a la derecha por lo que la señal va de la muestra 0 a la muestra M. La modificación hecha a la respuesta al impulso de un filtro ideal permite que la señal pueda ser procesada por la computadora, ya que de esta forma se tiene un número finito de muestras y en el caso de un filtro paso banda los índices de las muestras son todos positivos lo cual facilita el cómputo [27].

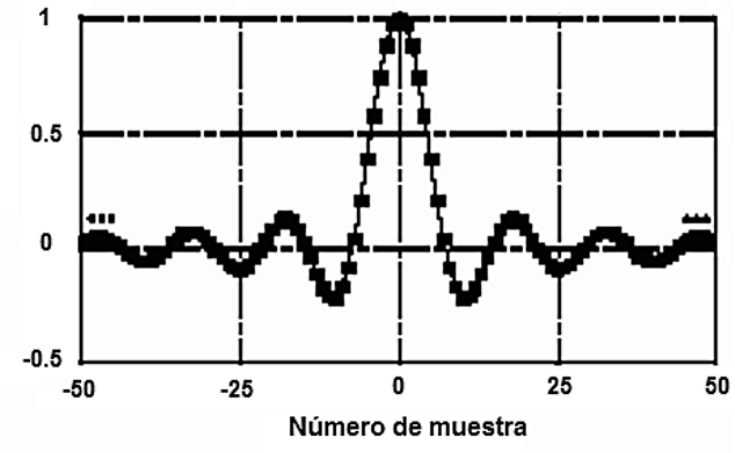


Figura 3.4: Repuesta al impulso de un filtro ideal.

Debido al truncamiento abrupto que se tiene en la respuesta al impulso (figura 3.5), se presentan en el dominio de la frecuencia oscilaciones excesivas en la banda de paso (Efecto de Gibbs). Para reducir este efecto y mejorar la repuesta en frecuencia de estos filtros existen otras ventanas que en general reducen los truncamientos abruptos. Las principales características que son interés de la respuesta a la frecuencia de estas ventanas son: ancho de lóbulo principal, ancho de banda de transición[1] y pico de lóbulo secundario, ya que describen el desempeño de una ventana. Cabe resaltar que para todas estas ventanas hay una relación entre la banda de transición y la banda del lóbulo principal pues cuan mayor sea el lóbulo principal mayor será la banda de transición del filtro [34]. A continuación se describen las ventanas que existen en GNU Radio y que además son las más empleadas.

Ventana Rectangular

Cuando tenemos M puntos de una señal, estamos multiplicando implícitamente la respuesta del filtro FIR por una ventana rectangular. Una ventana rectangular tiene un valor de amplitud igual a uno para las muestras, esto es de la muestra 0 a la muestra M, mientras que su amplitud es cero para todas las muestras restantes. [35]. Debido a esto su forma en el dominio temporal es un rectángulo, como su nombre lo indica. Esta ventana es la más simple y la menos usada [36] porque presenta un

[1]Es la banda que se encuentra entre la banda de paso y banda de rechazo de un filtro.

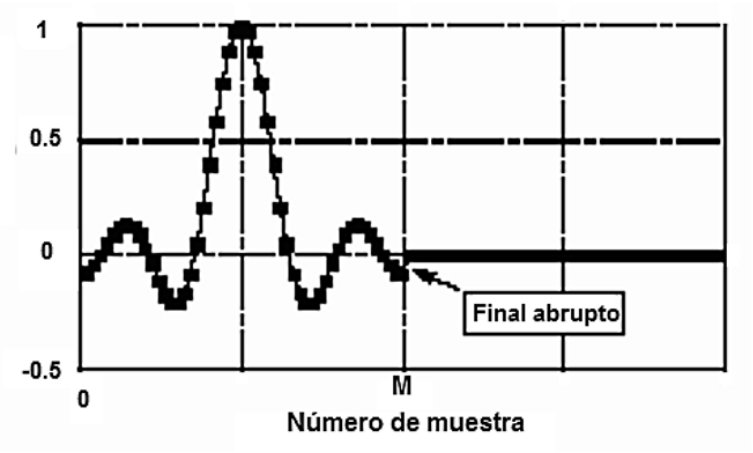


Figura 3.5: Respuesta al impulso truncada y desplazada.

truncamiento abrupto que causa oscilaciones en la banda de paso. En la figura 3.6 se observa el comportamiento de esta ventana en la banda de rechazo. El lóbulo secundario tiene una atenuación de tan solo -13 dB y la atenuación es muy lenta en los demás lóbulos laterales [34].

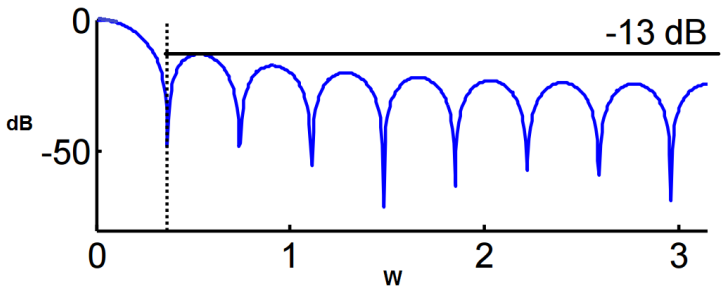


Figura 3.6: Respuesta a la frecuencia de ventana rectangular.

Ventana triangular o ventana de Bartlett

Como su nombre lo indica esta ventana tiene forma de triángulo en el dominio temporal. Ésta presenta en el dominio de la frecuencia un lóbulo de -25 dB, el ancho del lóbulo principal es $8\pi/M$ y el ancho de la banda de transición está dado por $6.1\pi/M$ donde M es el número de muestras.

Ventana Hanning

De esta ventana en adelante (las que se explican posteriormente) ya no se presentan oscilaciones excesivas en la banda de paso debido a que éstas resuelven el problema del truncamiento abrupto. Las ventanas de Hanning y Haming se encuentran dentro de la categoría de las ventanas llamadas coseno alzado, en comparación con la ventana triangular son más suaves en los extremos. En el dominio de la frecuencia se caracteriza por tener un rápido decremento en la amplitud de los lóbulos laterales [36], lo cual es muy deseable porque nos acercamos un poco más al filtro ideal. El lóbulo secundario tiene una amplitud de -31 dB que es mucho mejor que el de la ventana rectangular.

Ventana Hamming

En el dominio de la frecuencia esta ventana se caracteriza por tener un mejor lóbulo secundario, respecto a la ventana Hanning, con una amplitud de -41 dB [34] aunque el decaimiento de los demás lóbulos laterales es muy lento. Las ventanas Hanning, Hamming y Bartlett tienen el mismo ancho del lóbulo principal y por lo tanto anchos de banda de transición muy similares.

Ventana Blackman- Harris

La ventana Blackman-Harris presenta una buena atenuación de los lóbulos laterales, tiene una amplitud relativa de -53 dB en el lóbulo secundario. El ancho del lóbulo principal es $12\pi/M$ y su banda de transición es de $11\pi/M$ [34]. Su forma es también parecida a una campana en el dominio temporal pero la amplitud de esta cae más rápido desde la cima de la campana.

Ventana Kaiser

Esta ventana presenta un buen compromiso entre el ancho de banda del lóbulo principal, la amplitud del lóbulo secundario y el decaimiento de los lóbulos laterales. Es implementada mediante funciones de Bessel y se caracteriza por que es definida por un parámetro llamado β, el cual evita que el ancho de banda del lóbulo principal dependa directamente del número de muestras M como en todas las ventanas

anteriores. En la figura 3.7 [36] se observa un ejemplo de ventana Kaiser cuando β vale 9.5. En general su forma en el dominio temporal es también el de una campana pero su forma específica depende del parámetro β.

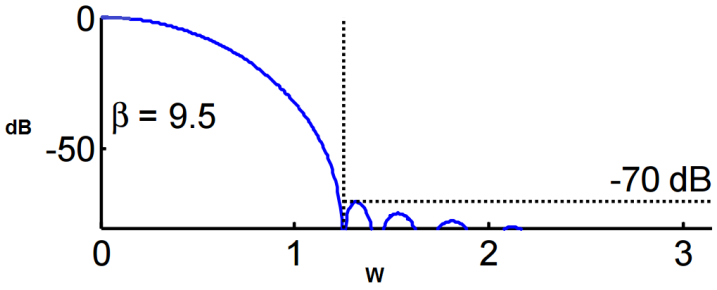


Figura 3.7: Respuesta a la frecuencia de ventana Kaiser.

Las ventanas descritas anteriormente son las ventanas que existen y pueden ser usadas en bloques de GNU Radio que tienen como propósito mostrar el espectro de la señal. Seleccionar qué ventana se utiliza para multiplicar a nuestra señal es muy importante ya que de esto depende el análisis correcto del espectro de la señal. Escoger cual es la mejor ventana depende de la aplicación. Aunque en general algunas ventanas representan desempeños superiores como es el caso de la ventana Blackman-Harris y la ventana Kaiser.

3.5. Señales de tipo complejo

La mayoría de los radios definidos por software, incluyendo el USRP X300 utilizado en esta tesis, procesan las señales divididas en dos componentes, una llamada "I" por su significado en inglés In-Phase y la otra llamada "Q" por Quadrature. En la figura 3.8 se presenta un diagrama que muestra el procesado de dichas componentes. Una señal puede ser representada de la forma $I(t) + jQ(t)$ por lo que también es llamada señal de forma compleja. En el procesamiento digital de señales esta representación es ampliamente usada ya que permite procesar señales cerca de la frecuencia cero [37]. Además con las componentes apropiadas I y Q es posible generar cualquier forma de modulación y de manera análoga con las componentes IQ correctas es posible demodular cualquier señal [38].

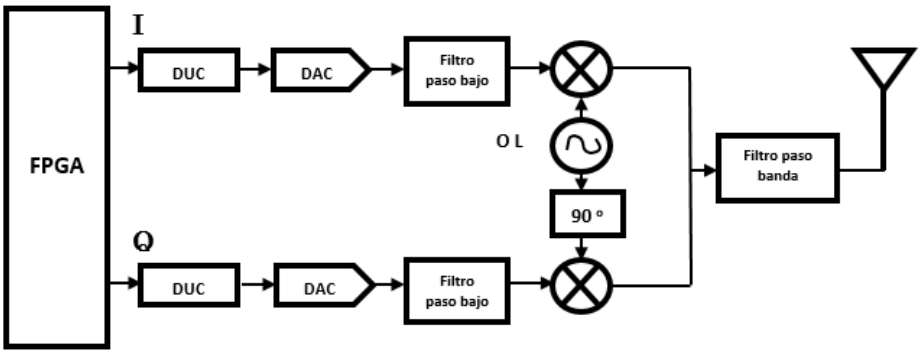


Figura 3.8: Procesado de una señal IQ en un SDR.

El teorema de muestreo indica que la frecuencia de muestreo de una señal analógica de banda base con un ancho de banda de -W a W debe ser al menos dos veces su ancho de banda (2W). Cuando la señal es compleja es posible representar ésta con una tasa de muestreo igual a W y por supuesto cada muestra es un número complejo [39]. Las señales reales son simétricas en magnitud en el dominio de la frecuencia respecto a la frecuencia 0. Una señal compleja no tiene que ser simétrica respecto a 0 en el dominio de la frecuencia, por lo que un muestreo a una velocidad de W no causa traslapes de los espectros repetidos.

3.6. Resumen

En general en este capítulo se describieron los conceptos básicos para llevar a cabo el diseño de los transmisores. El teorema de muestreo tanto para señales reales como complejas es básico para trabajar con el software GNU Radio. Otros conceptos como la interpolación y decimación no son en general básicos pero si necesarios en el caso de los transmisores desarrollados. Entender bajo que principios funcionan los bloques en GNU Radio es fundamental para hacer un diseño correcto por lo que conceptos como la transformada de Fourier y el método de las ventanas para filtros FIR fueron abordados.

Capítulo 4

El transmisor AM

4.1. Datos históricos

El primer tipo de modulación en desarrollarse fue la modulación en amplitud convencional, es decir la modulación en amplitud de doble banda lateral con portadora [40] y es también el tipo de modulación más usado de AM. En 1912 el profesor Edwin Howard Armstrong de nacionalidad estadounidense logra desarrollar el circuito generador heterodino, este desarrollo sirvió de base para que en 1918 produjera la modulación y demodulación de la voz, logrando una transmisión en amplitud modulada y creando el circuito superheterodino [41].

Actualmente la modulación AM es usada en radiodifusión (radio AM), en frecuencias de onda media (540 kHz - 1700 kHz), onda corta (3 MHz - 30 MHz) y solo en Europa frecuencias de onda larga (153kHz-279kHz) [42], también es utilizada en frecuencias VHF (30 a 300MHz) para comunicaciones entre aviones y torres de control de los aeropuertos. Una de las ventajas de la modulación AM es que es simple y por lo tanto barata.

4.2. Bases teóricas de la modulación en amplitud

Una modulación es AM (Amplitude Modulation) cuando la amplitud de una señal de alta frecuencia varía en proporción con una señal que por naturaleza es de baja frecuencia. La señal de baja frecuencia es la información que se desea trans-

mitir también llamada señal moduladora, usualmente se encuentra en el orden de los kHz y la señal de frecuencia alta se le llama portadora usualmente en el orden de los MHz [42]. La señal de información puede contener una sola frecuencia o con mayor probabilidad un rango de frecuencias, por ejemplo la voz tiene frecuencias aproximadamente desde 300 Hz a 3000Hz [31].

En una modulación se quiere generar una señal de radio $s(t)$ modificando una portadora $c(t) = A_c sen(2\pi f_c t)$ para representar el mensaje o señal moduladora $x(t) = A_m sen(2\pi f_m t)$, donde la señal moduladora está simplificada a un tono. Una representación general de una onda de radio es la siguiente: $s(t) = a(t)cos[2\pi f_c t + \phi(t)]$, donde $a(t)$ es la amplitud y $\phi(t)$ la fase. Se pueden usar los parámetros $a(t)$, $\phi(t)$ o incluso una combinación de ambos para representar el mensaje. En la modulación en amplitud se utiliza solo $a(t)$ para representar el mensaje y $\phi(t)$ permanece constante.

La amplitud de una señal de AM varía proporcionalmente con la señal moduladora, la amplitud máxima en volts de la señal de AM es $A_c + A_m$ por lo que la señal de AM ($S_{AM}(t)$) puede ser descrita matemáticamente de la siguiente manera:

$$S_{AM}(t) = [A_c + A_m sen(2\pi f_m t)]sen(2\pi f_c t) \tag{4.1}$$

Si se tiene que no hay señal moduladora es decir $x(t) = 0$ entonces la señal de AM es igual a la portadora.

$$S_{AM}(t) = A_c[sen(2\pi f_c t)] \tag{4.2}$$

Un parámetro importante es el índice de modulación, el cual describe la cantidad de cambio en la amplitud en una onda de AM [31]. Matemáticamente el índice de modulación se expresa en la ecuación 4.3. Mientras que el porcentaje de modulación es simplemente el índice de modulación expresado en forma de porcentaje.

$$m = \frac{A_m}{A_c} \tag{4.3}$$

La onda de AM es una onda compleja formada por un voltaje de corriente directa, la frecuencia de la portadora y la suma y diferencia de las frecuencias es decir $(f_c - f_m)$ y $(f_m + fc)$. En la figura 4.1 se detalla una señal de AM, de doble banda lateral con portadora, en el dominio de la frecuencia , donde $f_{m(max)}$ es la frecuencia máxima de la señal moduladora.

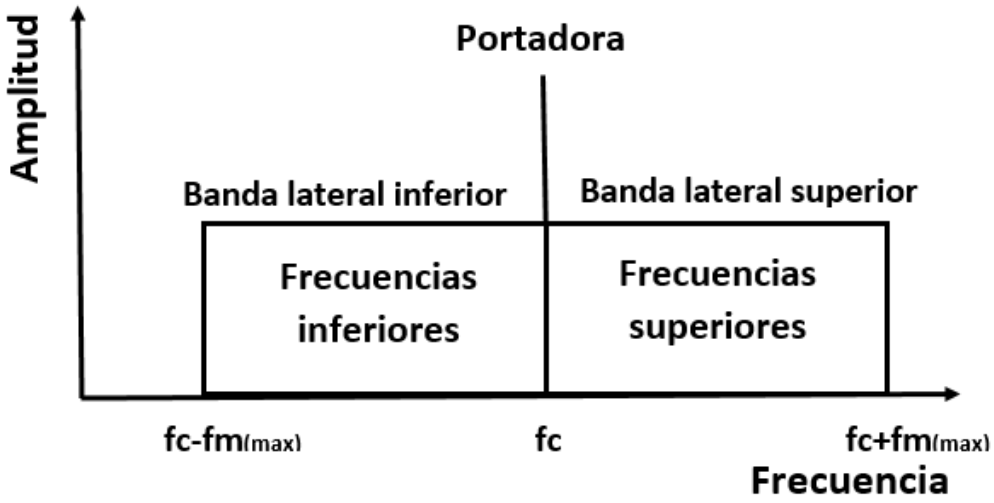


Figura 4.1: Espectro de una señal de AM

Para entender el comportamiento de una señal de AM en el dominio de la frecuencia es suficiente con desarrollar las multiplicaciones de la expresión matemática que representa la señal de AM como se efectúa a continuación. Como primer paso se sustituye el índice de modulación de la expresión 4.3 en la ecuación 4.1 con el fin de tener todas las amplitudes en función de la amplitud de la portadora.

$$S_{AM}(t) = [A_c + mA_c sen(2\pi f_m t)]sen(2\pi f_c t)] \tag{4.4}$$

Efectuando las multiplicaciones se obtiene:

$$S_{AM}(t) = A_c sen(2\pi f_c t) + mA_c sen(2\pi f_m t)sen(2\pi f_c t)$$

Después, la multiplicación de dos señales seno cumple con la siguiente identidad:

$$sen(x)sen(y) = 1/2[cos(x-y) - cos(x+y)]$$

Finalmente, una señal modulada en amplitud queda representada en la ecuación 4.5:

$$S_{AM}(t) = A_c sen(2\pi f_c t) - \frac{mA_c}{2} cos(2\pi(f_c + f_m)t) + \frac{mA_c}{2} cos(2\pi(f_c - f_m)t) \tag{4.5}$$

Observar que si se factoriza A_c de la ecuación 4.4 se obtiene la la expresión 4.6:

$$S_{AM}(t) = A_c[1 + msen(2\pi f_m t)]sen(2\pi f_c t) \tag{4.6}$$

donde $[1 + msen(2\pi f_m t)]$ es un voltaje constante más la señal moduladora, mientras que $A_c sen(2\pi f_c t)$ es el voltaje de la portadora no modulada. Expresar la señal de AM de esta forma, hace evidente que la señal moduladora contiene una componente constante con valor unitario y una componente senodidal con la frecuencia de la señal moduladora [31]. Si se calcula nuevamente la ecuación 4.5 desde la expresión 4.6 se aprecia que la componente constante es la responsable de que aparezca la componente de la señal portadora en la onda modulada.

La ecuación 4.5 es importante porque nos permite determinar el comportamiento de la señal de amplitud modulada en el dominio de la frecuencia así como su distribución del voltaje. El voltaje de la frecuencia lateral inferior y superior es $\frac{mA_c}{2}$ mientras el voltaje de la frecuencia portadora es simplemente A_c. Si se requiere la distribución de la potencia de la onda AM solo es necesario recordar que la potencia disipada en todo circuito eléctrico es igual al cuadrado del voltaje RMS (Root Mean Square) dividido entre la resistencia. Por lo tanto, la potencia de la portadora (P_C) y la potencia de las bandas laterales (P_{BL}) superior e inferior es [31]:

$$P_c = \frac{A_c^2}{2R} \tag{4.7}$$

$$P_{BL} = \frac{(mA_c/2)^2}{2R} \tag{4.8}$$

Respecto al ancho de banda de una señal AM se puede apreciar de la figura 4.1 que es igual a dos veces la frecuencia máxima de la señal moduladora (ecuación 4.9). El ancho de banda de un canal en radiodifusión AM es de tan solo 10 KHz por lo que el ancho de banda de cada banda lateral es de tan solo 5 kHz.

$$B = 2 * f_{m(max)} \tag{4.9}$$

Una representación fasorial de AM permite entender fácilmente cómo es que la señal modulada cambia de amplitud. Cabe recordar que un fasor es otra manera de representar una sinusoidal, el fasor es un número complejo que representa la amplitud y fase de una onda sinusoidal [43].

Con una moduladora de frecuencia única, la onda modulada en amplitud o señal AM se obtiene de la suma vectorial de la portadora y de las frecuencias de lado

superior e inferior. Las dos frecuencias laterales se combinan resultando un vector y después este vector resultante se suma con la portadora. Esta suma vectorial se puede observar en la figura 4.2 [31]. Si se mantiene estacionario el fasor de la frecuencia portadora, el fasor de la frecuencia lateral superior (FLS) gira con respecto a la portadora en sentido contrario a las manecillas del reloj y el fasor de la frecuencia lateral inferior (FLI) gira en dirección a las manecillas de reloj.

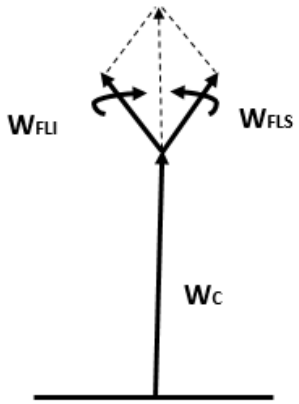


Figura 4.2: Suma de los fasores portadora y frecuencias superior e inferior

Las frecuencias laterales se suman cuando están en fase como es el caso de la figura 4.3 o se restan si están desfasadas. Para una onda AM la amplitud máxima positiva se presenta cuando todos los fasores están en su valor máximo positivo al mismo tiempo, que es también el caso de la figura 4.3. La amplitud mínima positiva se presenta cuando la portadora tiene su valor máximo de amplitud y los fasores laterales su amplitud máxima negativa en el mismo tiempo. Un caso intermedio es cuando el fasor portadora presenta su valor positivo máximo y los fasores de las frecuencias laterales (FLI y FLS) tienen el mismo valor de amplitud pero se encuentran desfasados 180 grados, el vector resultante es el de la portadora como el caso de la figura 4.4.

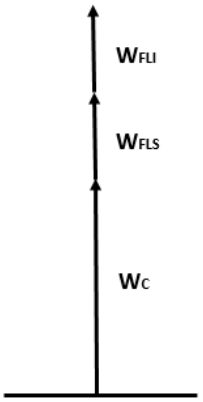


Figura 4.3: Amplitud máxima de la señal modulada

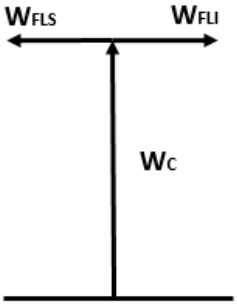


Figura 4.4: Amplitud media positiva de la señal modulada

4.3. Diseño

A continuación se muestra en la figura 4.5 un diagrama general del transmisor de amplitud modulada (de doble banda lateral con portadora) desarrollado en GNU Radio. Para cada etapa o bloque se puede observar la ecuación correspondiente y en algunos casos el oscilograma correspondiente. Se utiliza como señal moduladora una onda senoidal con el fin de explicar el transmisor con base en la teoría descrita en el subcapítulo anterior. El primer bloque de este diagrama es el que permite obtener o generar la señal moduladora, en el diagrama se observa una señal moduladora de amplitud unitaria. En el siguiente bloque se multiplica la señal moduladora por una constante llamada m' con lo que la amplitud de la moduladora dependerá del valor de m'. Cabe aclarar que esta variable no es el índice de modulación, pues no concuerda con la definición de índice de modulación escrita anteriormente. Después a la señal moduladora se le suma una constante llamada k, ésta constante representa un offset para la señal. La onda senoidal que corresponde a este bloque se observa también en la figura 4.5 cuando la constante (m') por la que fue multiplicada en el bloque anterior vale uno y la constante k también vale 1. Como última etapa la señal moduladora es multiplicada por la señal portadora con lo que a la salida de este bloque se tiene la señal modulada en amplitud u onda AM.

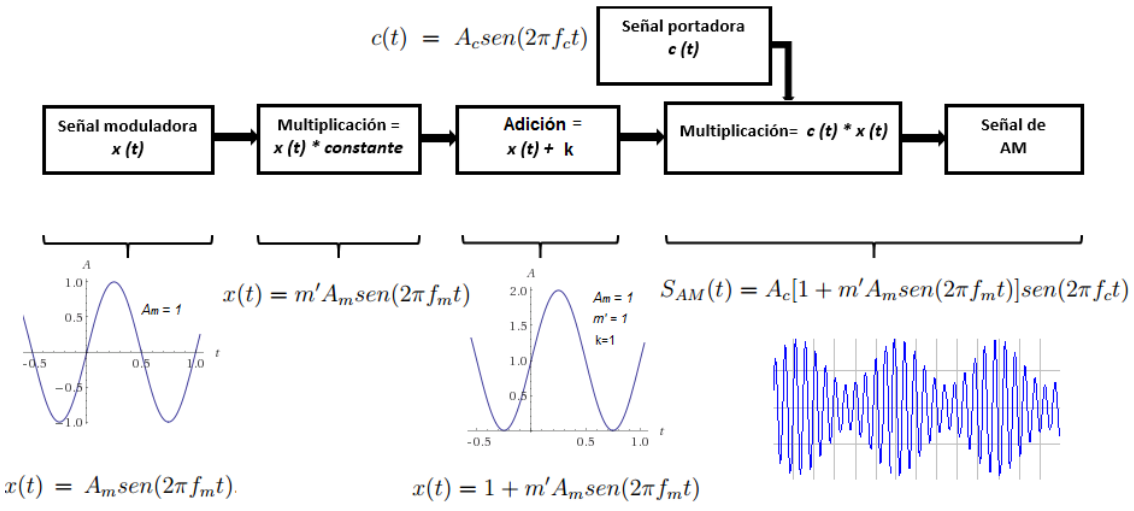


Figura 4.5: Diagrama general del transmisor AM programado en GNU Radio.

Al final de este transmisor la señal modulada corresponde con la siguiente ecuación:

$$S_{AM}(t) = A_c[k + m' A_m sen(2\pi f_m t)]sen(2\pi f_c t) \tag{4.10}$$

La ecuación 4.10 de AM no corresponde totalmente con la ecuación 4.6 de AM dada anteriormente ($A_c[1+msen(2\pi f_m t)]sen(2\pi f_c t)$). Para que concuerde con dicha ecuación k debe ser igual a 1, aunque puede tomar otros valores, se adopta 1 por simplicidad y con el objetivo de que el índice de modulación pueda ser definido simplemente como se muestra en la siguiente ecuación:

$$m = m' A_m \tag{4.11}$$

La ecuación 4.11 es la expresión matemática del índice de modulación para el transmisor. Por otro lado el bloque que añade la constante con valor de uno a la señal moduladora es también responsable de que en el espectro de la señal de AM aparezca la componente de la frecuencia portadora no modulada.

Como ya se mencionó en el capítulo dos programar en GNU Radio es programar mediante bloques y la conexión entre ellos. A continuación se hará una descripción de los bloques del transmisor de AM (figura 4.5) tal y como existen en GNU Radio.

Bloque 1: Signal Source

Es el primer bloque del diagrama 4.5. Este bloque genera la señal en banda base es decir la información que se desea transmitir o señal moduladora. Existen diferentes propiedades del bloque que pueden ser configuradas:

- Output Type

Este parámetro específica el tipo de dato que se tendrá a la salida del bloque. Existen cuatro tipos de datos: complejo, flotante, entero (de 32 bits) y entero corto (16 bits) [44]. En el transmisor se optó por un dato de tipo flotante.

- Frequency

Es el valor de la frecuencia de salida de datos. Es decir es la frecuencia de la señal moduladora. Esta señal es en realidad la variable f_m que toma valores de 15 kHz a

5 MHz. Se incluyen señales de frecuencia mayor a 20 kHz solamente con el objetivo de observar un mayor número de cambios de amplitud en una misma ventana de tiempo y en el caso del espectro para observar una amplia separación entre portadora y bandas laterales.

- Sample Rate

Con este parámetro se específica la frecuencia de muestreo a la salida del bloque. Del capítulo anterior es importante recordar que se debe cumplir con el teorema de Nyquist. En este caso la frecuencia de muestreo es tres veces la frecuencia máxima de f_m $(3 * 5MHz)$ solamente con el objetivo de tener una tasa de muestreo un poco mayor al teorema de Nyquist. Además que la tasa de muestreo sea un número entero simplifica su modificación posterior.

- Waveform

Aquí se configura la forma de onda y se tienen seis posibles formas de onda: constante, seno, coseno, cuadrada, triángulo y diente de sierra. Cualquiera puede ser elegida.

- Amplitude

Especifica la amplitud pico máxima que tendrá la señal seno o coseno. En caso de que se opte por señales tipo triángulo, cuadrada o diente de sierra, el valor indicado es la amplitud pico a pico. Si la señal es una constante se utiliza el parámetro offset [44]. En el transmisor la amplitud de la señal moduladora puede tomar cualquier valor y es controlado con una variable llamada A_m.

- Offset

Especifica el offset que se le agregará a la señal original, este puede ser de tipo complejo [44]. En otras palabras representa el valor promedio que la señal tendrá, en este caso se fija a cero.

En el cuadro 4.1 se encuentra el resumen de la configuración de éste bloque.

El bloque que acaba de presentarse tiene propiedades mediante las cuales es posible ahorrase los siguientes dos bloques (bloques donde se multiplica la señal moduladora por una constante y donde se le suma una unidad a la señal moduladora). La razón por la que no se incluyen los tres bloques en uno solo es debida a que el bloque Signal Source no siempre será el generador de nuestra señal moduladora y como se verá más adelante no todos los bloques cuentan con éstas propiedades.

Signal Source	
Parámetro	Valor
Output Type	Float
Frequency	f_m
Sample Rate	$15 * 10^6$
Waveform	Sine
Amplitude	Am
Offset	0

Cuadro 4.1: Configuración del bloque "Signal Source" en el tx AM.

Bloque 2: Multiply Const

Es el segundo bloque del diagrama 4.5. La señal moduladora es multiplicada por una constante, la cual puede ser modificada durante la ejecución del transmisor. Matemáticamente este bloque implementa la función: $salida = entrada * constante$. En este caso la entrada es la señal moduladora.

- IO Type

Con este parámetro se indica el tipo de dato de entrada y salida al bloque por lo tanto tiene que ser el mismo tipo de dato que envía el bloque anterior (flotante).

- Const

Especifica el valor por el cual se multiplicará la entrada o en este caso señal moduladora. Para el transmisor esta constante en realidad es una variable llamada m que toma valores de 0 a 2.5. El índice de modulación para el transmisor esta definido como $m = m' * A_m$ pero debido a que el valor por defecto de A_m es 1, m es igual a m'. Para cualquier otro valor de A_m el índice de modulación debe ser calculado.

- Vec Length

El nombre o ID del bloque indica que los ítems que procesa son vectores (nomenclatura encontrada en el capítulo 2) por lo que es necesario indicar la longitud del vector. Para la mayoría de las aplicaciones se utiliza el valor predeterminando que es uno.

Multiply Const	
Parámetro	Valor
Output Type	Float
Constant	m
Vec Length	1

Cuadro 4.2: Configuración del bloque "Multiply Const" en el tx AM.

Bloque 3: Add Const

Este bloque implementa la función: $salida = entrada + constante$. Es utilizado para añadir a la moduladora el voltaje constante con valor 1.

- IO Type

Este parámetro especifica lo mismo que en el bloque anterior por lo que también es de tipo flotante.

- Const

Es el valor de la constante utilizada y en este caso no varía, tiene un valor igual a uno durante toda la ejecución del programa. A la salida de este bloque la señal moduladora es de la forma $[1 + m'Amsen(2\pi fmt)]$.

Add Const	
Parámetro	Valor
Output Type	Float
Constant	1
Vec Length	1

Cuadro 4.3: Configuración del bloque "Add Const" en el tx AM.

Bloque 4: Signal Source

Este bloque es el generador de la señal portadora $c(t) = A_c sen(2\pi f_c t)$. Es el mismo bloque utilizado para generar la señal moduladora por lo que sus parámetros tienen la misma descripción. La señal portadora puede variar en el rango de los 55

MHz a los 65 MHz (variable f_c) y es muestreada a un poco más de dos veces la frecuencia máxima que puede tomar (150 MHz). Lo anterior con el propósito de abarcar mayor ancho de banda y obtener la frecuencia lateral superior de AM que en el mayor de los casos estaría situada a 70 MHz (65 MHz + 5 MHz) por lo que se requiere muestrear a mínimo 140 MHz. La frecuencia de muestreo se controla con una variable llamada *samp_rate*. Respecto a la amplitud de la portadora (Ac), se asignó un valor de uno por simplicidad, valor que permanece constante durante todo el programa. En el cuadro 4.1 se muestra el resumen de la configuración del bloque:

Signal Source	
Parámetro	Valor
Output Type	Float
Frequency	f_c
Sample Rate	*samp_rate*
Waveform	Sine
Amplitude	1
Offset	0

Cuadro 4.4: Configuración del bloque "Signal Source" para f_c en el tx AM.

Bloque 5: Multiply

De forma general este bloque realiza la multiplicación de una o más entradas. Para el caso de este transmisor solo se tienen dos entradas y la función implementada es la siguiente: $salida = moduladora * portadora$. En esencia en este bloque se realiza la modulación en amplitud.

- IO Type

De igual forma que los bloques anteriores es de tipo flotante.

- Num inputs

Este parámetro es de tipo entero e indica el número de entradas al bloque, es decir el número de entradas que se multiplicaran. Es evidente que en este caso se tienen dos entradas.

Este bloque también trabaja con vectores por lo que es necesario especificar la longitud, en el cuadro 4.5 se muestra el resumen de la configuración.

Multiply	
Parámetro	Valor
IO Type	2
Num inputs	2
Vec Length	1

Cuadro 4.5: Configuración del bloque "Multiply" en el tx AM.

Bloque 6: UHD: USRP Sink

Este bloque recibe la señal modulada en amplitud y la envía al USRP para terminar el procesamiento de la señal y finalmente radiar una señal analógica modulada en amplitud. A continuación se describen los parámetros del bloque utilizados en el transmisor.

- In Type

En este bloque los datos flotantes son convertidos a tipo complejo de longitud de 16 bits. Recordar del capítulo dos que el USRP procesa señales de tipo complejo, por lo que a la salida del bloque la señal tiene que ser compleja, representada por dos componentes.

- Sample rate

Es el número de muestras por segundo de entrada al bloque. Para cumplir con el teorema de Nyquist se debe muestrear a 2 veces el ancho de banda de la señal por lo que este parámetro se configuró a 25 MHz pues el ancho de banda más grande que se puede tener es de 10 MHz.

- Center Frequency

Ésta es la frecuencia central del canal RF.

- Antenna

Para dispositivos con una sola antena este espacio puede dejarse en blanco. La tarjeta hija utilizada en esta tesis cuenta con dos antenas por lo que es necesario especificar la antena que se utiliza, en este caso es la TX/RX.

- Bandwidth

El ancho de banda se asigna de manera automática si se deja un valor de 0. Solo ciertos dispositivos permiten tener anchos de banda de filtros reconfigurables.

En el cuadro 4.6 se muestra el resumen de la configuración del bloque UHD. Se añade un parámetro más el cual corresponde con la ganancia de la antena en dB.

UHD: USRP Sink	
Parámetro	Valor
In Type	Complex int 16
Sample rate	25 MHz
Center Frequency	fc
Gain	25
Antenna	TX/RX
Bandwidth	0

Cuadro 4.6: Configuración del bloque “UHD: USRP Sink” en el tx AM.

4.4. Implementación

El diagrama real del transmisor programado a bloques en GNU Radio es el de la figura 4.6.

En resumen el transmisor consta de siete bloques para el procesado de la señal, dos bloques para mostrar las características de la señal modulada (espectro y oscilograma), y cinco bloques que permiten tener el control de las variables involucradas en el programa. Los bloques más importantes del programa son los bloques que involucran el procesado de la señal para obtener una modulación en amplitud. Todos estos bloques ya han sido descritos anteriormente excepto uno, el bloque llamado “Rational Resampler”. Este bloque se utiliza como interpolador con el fin de tener el mismo número de muestras a la entrada del bloque multiplicador, recordar que las

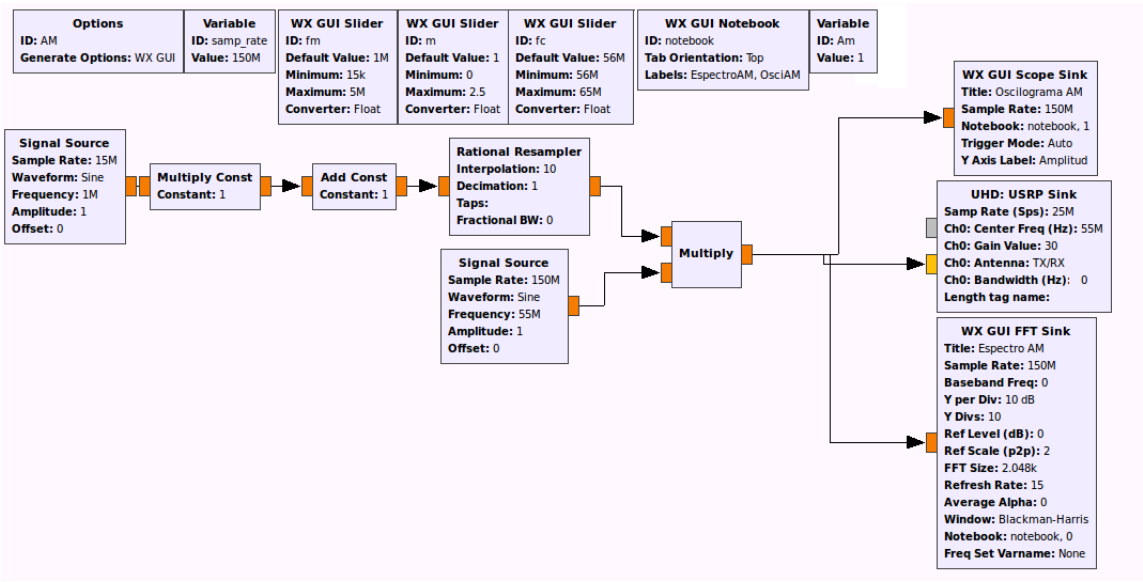


Figura 4.6: Diagrama del transmisor AM programado en GNU Radio.

señales multiplicadas son discretas por lo que se requiere el mismo número de puntos o muestras para la multiplicación. El factor de interpolación es de 10, lo que quiere decir que a la salida de éste se tendrán 15 $MS/s * 10$ [1] dando un total de 150 MS/s que es la frecuencia de muestreo de la señal portadora. La señal moduladora no tiene desde un principio una tasa de muestreo de 150 MS/s con el fin de reducir la carga computacional.

Las variables utilizadas en el programa se pueden observar en la parte superior: *sample_rate*, f_m, m, A_m y f_c. Todas explicadas en el diseño del transmisor. También aparecen dos bloques para obtener el espectro y oscilograma de la señal. Para observar la señal en el dominio de la frecuencia se utiliza un bloque llamado *WX GUI FFT Sink*. Como el nombre del bloque lo indica, este bloque implementa la transformada rápida de Fourier. La señal en el dominio del tiempo es obtenida mediante el bloque *WX GUI Scope Sink* que implementa un osciloscopio. Para estos dos bloques el parámetro *Sample Rate* tiene que ser el mismo que para los bloques anteriores con el fin de que la señal sea representada correctamente. La interfaz gráfica utilizada en este transmisor es la *WX GUI*, permite visualizar las ventanas donde se muestra el oscilograma y espectro de la señal modulada, un pa-

[1] S/s son muestras por segundo

nel de control y las variables que pueden ser modificadas por el usuario. El bloque *WX GUI Notebook* simplemente muestra las ventanas mediante pestañas que siguen un orden indicado. La interfaz descrita anteriormente a través de la cual los alumnos interactuarán con el programa es la mostrada en la imagen 4.7.

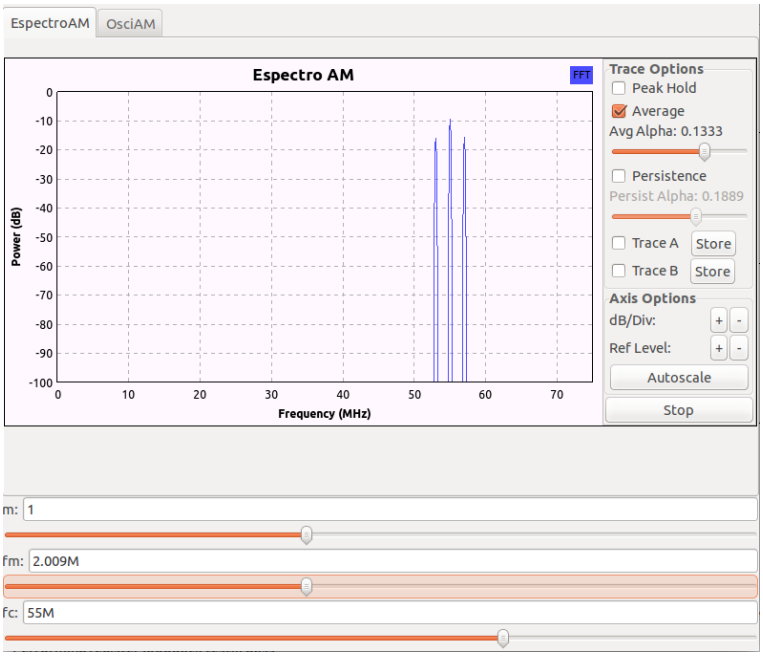


Figura 4.7: Interfaz del programa transmisor AM con el usuario.

El ancho de banda de una señal modulada en amplitud depende de la máxima frecuencia de la información que se transmite, reflejado en la figura 4.8 lado izquierdo, donde se tiene una transmisión a 56 MHz, señal moduladora de 3 MHz e índice de modulación 1. De lado derecho se tiene la misma transmisión pero con un índice de modulación 2.5. En la figura 4.9 se tiene la misma señal pero en el dominio temporal. El oscilograma de lado izquierdo con índice de modulación igual a uno, mientras que el lado derecho es el oscilograma de la señal con índice de modulación 2.5. En el dominio temporal es claro que la sobre modulación causa deformaciones en la señal de AM.

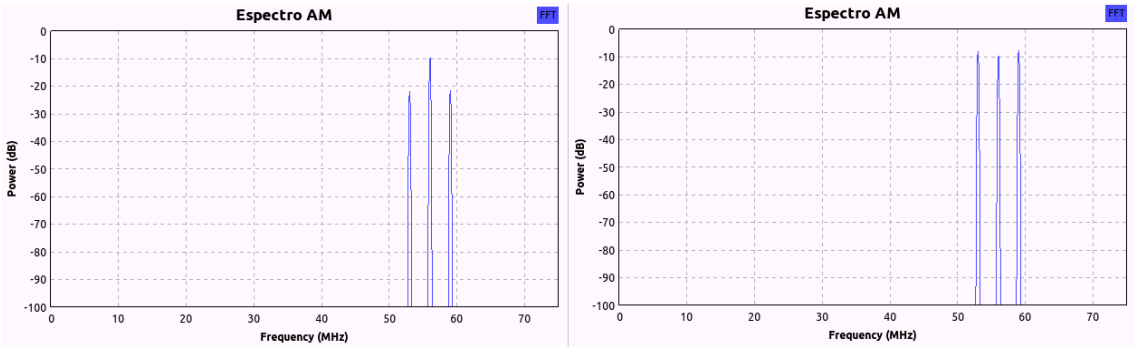


Figura 4.8: Espectros de una señal de AM sin sobre modulación (derecha) y con sobre modulación (izquierda).

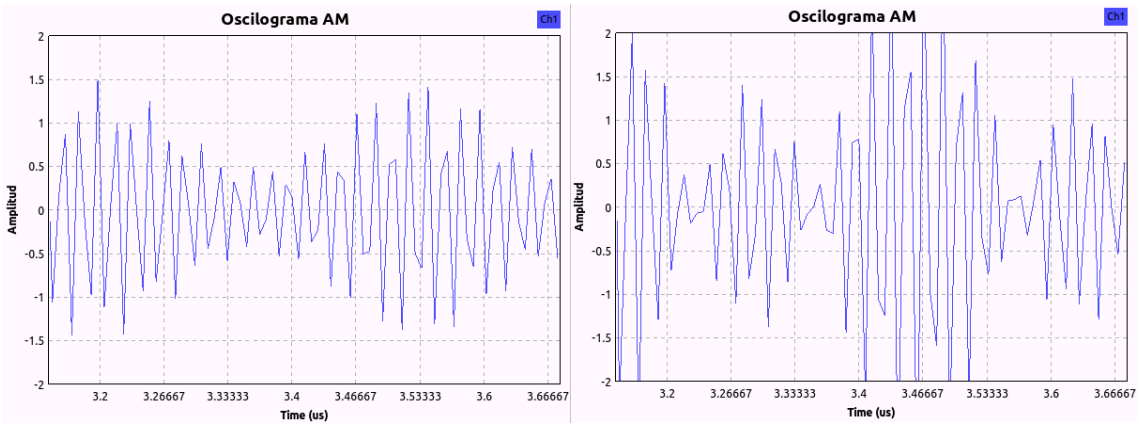


Figura 4.9: Oscilogramas de una señal de AM sin sobre modulación (izquierda) y con sobre modulación (derecha).

Capítulo 5

El Transmisor FM

5.1. Datos históricos

La modulación angular[1] apareció por primera vez en 1931 como una alternativa de la modulación AM pues se sugirió que una modulación angular es más robusta ante el ruido. En 1933 Edwin H. Armstrong creó y produjo el sistema de modulación en frecuencia FM, este sistema demostró tener una mejor recepción frente a tormentas violentas y una mejor fidelidad de sonido que el sistema AM. Sin embargo no fue sino hasta 1940 que surgió la primera estación de FM debido a que la industria de la radio no pudo asumir un cambio de sistema, finalmente en 1941 bajo la dirección de Armstrong se iniciaron transmisiones regulares en FM [41].

A finales de la década de 1960 el sistema FM se estableció claramente como un sistema superior, la mayoría de los aparatos de radio se vendieron en FM, la mayoría de repetidoras de microonda fueron establecidas en FM. La modulación en FM fue utilizada en TV analógica para la modulación del audio y en comunicaciones espaciales. Actualmente la modulación angular sigue siendo ampliamente utilizada en emisiones de radio comercial, televisión y transmisión de sonido, radioteléfonos, y sistemas de comunicaciones por microondas y satélites [31].

[1] La modulación angular tiene dos variantes: modulación de frecuencia (FM) y modulación de fase (PM). Posteriormente se explicará la diferencia entre ambas modulaciones.

5.2. Bases teóricas de la modulación en frecuencia

La modulación angular tiene fuertes ventajas sobre la modulación por amplitud:

- Reducción del ruido: en receptores de FM es posible eliminar perturbaciones o desvanecimientos que sufre la señal a través del medio de transmisión. En AM el ruido provoca variaciones en amplitud por lo que la fidelidad de la señal se ve afectada en el receptor.

- Uso de potencia eficiente: con una modulación AM la amplitud de la envolvente de la portadora varía directamente con la amplitud de la moduladora mientras que en una modulación FM la amplitud de la envolvente de la portadora es constante. En consecuencia un transmisor FM puede operar siempre a la potencia pico [45].

También se presentan desventajas pues se requiere un ancho de banda mayor en una modulación FM.

La característica que define una modulación angular es que la fase de la portadora varía dependiendo de la señal moduladora manteniendo una amplitud constante. Es decir para representar el mensaje se utiliza el parámetro $\phi(t)$ y $a(t)$ permanece constante. La onda de radio que se transmite es $s(t) = a(t)sen[2\pi f_c t + \phi(t)]$. Para representar el mensaje es necesario que $\phi(t)$ sea una función de la señal moduladora [31] lo cual puede expresarse como $\phi(t) = F[x(t)]$ donde x(t) es la señal moduladora. En caso de que sea senoidal tendría la siguiente forma: $x(t) = A_m sen[2\pi f_m t]$. Sin embargo al variar la fase de una onda también varía la frecuencia y si varía la frecuencia inevitablemente varía la fase. Existen dos tipos de modulación angular: la FM y la PM, cuando la propiedad que se hace variar directamente es la frecuencia se habla de FM y cuando se varía la fase directamente se habla de PM. El transmisor desarrollado en ésta tesis utiliza una modulación angular de tipo FM ya que la frecuencia de la portadora varía en proporción directa a la amplitud de la señal moduladora.

De forma general una modulación angular se puede escribir como en la siguiente ecuación:

$$S_{FM}(t) = A_c cos[2\pi f_c t + msen(2\pi f_m t)] \tag{5.1}$$

Donde m es el índice de modulación. Cuando una portadora está modulada en frecuencia el índice de modulación es directamente proporcional a la amplitud de la señal moduladora e inversamente proporcional a su frecuencia [31]. Por lo que el índice de modulación se define como en la ecuación 5.2:

$$m = \frac{kA_m}{f_m} \tag{5.2}$$

Donde k es la sensibilidad a la desviación de un modulador FM en $[Hz/V]$, por lo que el índice de modulación es adimensional. En otras palabras la sensibilidad a la desviación es la función de trasferencia del modulador.

Otro concepto importante en una modulación FM es la desviación máxima de frecuencia la cual se define como el desplazamiento relativo de la frecuencia de la señal portadora con respecto a su valor cuando no hay modulación. La desviación máxima de frecuencia expresada en [Hz] es el producto de la sensibilidad por el voltaje máximo de la señal moduladora, ecuación 5.3:

$$\Delta f = kA_m \tag{5.3}$$

Por lo que el índice de modulación también puede ser visto como la relación de la desviación máxima entre la frecuencia de la señal moduladora sustituyendo la ecuación 5.3 en 5.2. Respecto al porcentaje de índice de modulación, su concepto es muy diferente del de una modulación en amplitud. Para una señal FM el índice de modulación porcentual es la relación de la desviación en frecuencia producida realmente entre la desviación máxima de frecuencia permitida:

$$\%m = \frac{\Delta f_{real}}{\Delta f_{max}} \tag{5.4}$$

El ancho de banda de una señal de frecuencia modulada no solo depende de la frecuencia de la señal moduladora sino también del índice de modulación [31]. En un espectro de FM se pueden producir varios conjuntos de bandas laterales por lo que su ancho de banda es mucho mayor que el ancho de banda necesario en una modulación en amplitud. De la ecuación 5.1 no deriva fácilmente cuales son las componentes espectrales de la onda de FM. Para lograr ver las componentes presentes

existen identidades de funciones de Bessel como la ecuación 5.5 que es posible aplicar directamente a la ecuación 5.1. Resultando la ecuación 5.6 [31].

$$(cos\alpha + mcos\beta) = \sum_{n=-\infty}^{\infty} J_n(m)cos(\alpha + n\beta + \frac{n\pi}{2}) \tag{5.5}$$

$$S_{FM}(t) = A_c\{J_0(m)cos[wct] + J_1(m)cos[(w_c + w_m)t + \frac{\pi}{2}] - J_1(m)[(w_c + w_m)t - \frac{\pi}{2}] + \dots \tag{5.6}$$

donde m es el índice de modulación, J_0 es la componente de la portadora y J_1 es el primer conjunto de bandas laterales. Se observa que se producen conjuntos de componentes laterales infinitos, pero es posible solo considerar las frecuencias laterales significativas (amplitud mayor al 1 %).

Debido a lo anterior existen aproximaciones para el ancho de banda de una señal FM dependiendo de su índice de modulación [31]:

→ Índice de modulación bajo (menor a 1): $B = 2 * f_m$.

→ Índice alto (mayor a 10): $B = 2 * \Delta f$.

→ Regla de Carson (regla general): $B = 2(\Delta f + f_m)$.

→ Tablas de Bessel (ancho real) : $B = 2(n + f_m)$ donde n es la cantidad de bandas laterales significativas, valor que puede ser encontrado en las tablas de Bessel.

El ancho de banda de FM comercial en México abarca de 88 MHz a 108 MHz. Las frecuencias centrales asignadas empiezan desde 88.1 MHz con incrementos de 200 kHz hasta 107.9 MHz. Se permite una desviación máxima de frecuencia de 75 KHz. El ancho de banda asignado para cada canal de FM es de 200 KHz sin embargo el ancho de banda necesario para una estación es de 240 kHz pues las componentes dentro de este ancho de banda se consideran esenciales, una estación de FM no puede exceder de 240 kHz de anchura. Para evitar la interferencia las estaciones de radiodifusión de FM que operen en una misma localidad deben mantener una separación entre las frecuencias portadoras de 800 kHz como mínimo [46]. El ancho de banda de la señal moduladora depende de si se realiza una transmisión mono, estéreo o si incluyen datos RDS (Radio Data System) [47]. Cabe aclarar que el transmisor implementado

para este trabajo es un transmisor monofónico, es decir se utiliza un solo canal para la señal moduladora.

5.3. Diseño

El diagrama general del transmisor FM es el presentado en la figura 5.1. El funcionamiento general del transmisor se explica en el siguiente párrafo.

La señal moduladora en el caso del transmisor de FM puede ser un archivo de audio o un tono. El filtro paso bajas limita la señal de audio a una frecuencia de 15 KHz para concordar con el rango de frecuencias establecido por la IFT [46]. Al demodular una señal de FM se tiene que el voltaje de ruido aumenta conforme aumenta la frecuencia de la señal demodulada. Ésta distribución no uniforme de ruido es inherente a la modulación FM [31]. El amplificador de preénfasis simplemente contrarresta este fenómeno amplificando más la frecuencias altas de la señal moduladora. El siguiente bloque es el encargado de realizar la modulación en frecuencia por lo que a la salida de éste se tiene una señal de frecuencia modulada.

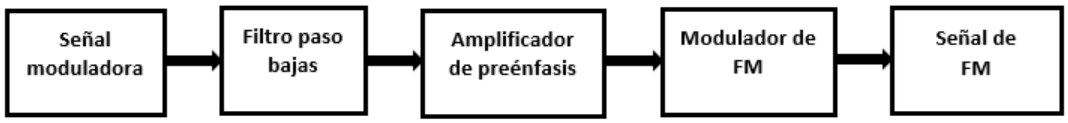


Figura 5.1: Diagrama del transmisor FM programado en GNU Radio.

En la figura 5.2 se muestra el mismo diagrama del transmisor FM pero con los nombres reales de los bloques en el software GNU Radio. A continuación se describen cada uno de los bloques del diagrama 5.2.

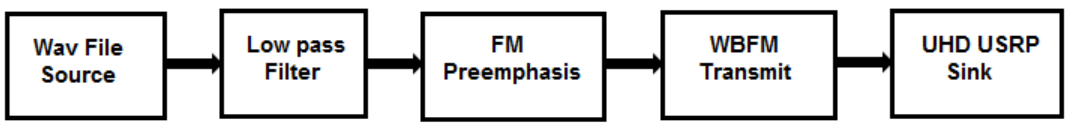


Figura 5.2: Nombres reales de los bloques que conforman el transmisor de FM programado en GNU Radio.

Bloque 1: Wav File Source

Éste bloque es utilizado para obtener la señal de información o moduladora. En este transmisor la fuente es un archivo de audio en formato WAV muestreado a 44.1 kHz. Sin embargo también es posible seleccionar el bloque File Source (cuyos parámetros se describieron en el capítulo anterior) para que la moduladora sea un tono. Los parámetros del bloque Wav File source son los siguientes:

- File

Especifica el nombre del archivo y la ruta del mismo.

- Repeat

Indica si el archivo se repite hasta el fin de la ejecución del programa.

- N Channels

Es el número de canales a leer del archivo WAV. Se utiliza un canal si es una transmisión monofónica o dos canales si es estereofónica [44]. El valor en este caso es 1 pues se trata de un transmisor monofónico.

En el cuadro 5.1 se presenta el resumen de la configuración del bloque.

Wav File Source	
Parámetro	Valor
File	archivo.wav
Repeat	yes
N Channels	1

Cuadro 5.1: Configuración del bloque “Wav File Source” en el tx FM.

Bloque 2: Low Pass Filter

Este bloque es utilizado para limitar el ancho de banda de la señal de audio. En FM la frecuencia máxima de la señal moduladora debe ser de 15 KHz [46]. El bloque es un filtro FIR por método de ventanas.

- Decimation

Este parámetro es utilizado para modificar la frecuencia de muestreo de la señal, en este caso tiene un valor de 1 por lo que en realidad la frecuencia de muestreo no cambia.

- Gain

Es la ganancia del filtro, en este caso tiene un valor de 1 por lo que la señal tiene la misma amplitud a la entrada y salida del filtro.

- Sample rate

Es el número de ítems de entrada y salida al filtro, es decir la frecuencia de muestreo por lo tanto tiene un valor de 44.1 kHz.

- Cuttoff Freq

Es la frecuencia de corte para el filtro paso bajas. Tiene un valor de 15 KHz pues es el ancho de banda máximo de la señal de audio establecido para FM.

- Transition Width

Es la banda de transición del filtro, como se vio en el capítulo dos existe una relación entre la banda de transición y la banda de paso. En una ventana Blackman-Harris la banda de transición es mucho más ancha que en otras ventanas. Éste parámetro se fijó a lo menor posible o permitido por el bloque para esta ventana, el valor conseguido fue de 1 kHz.

- Window

Blackman Harris. En el capítulo dos se exponen las características de la ventana Blackman Harris la cual presenta una buena atenuación de lóbulos laterales, motivo por el que fue elegida.

- Beta

Solo para la ventana Kaiser [44], explicado en capítulo dos. En este transmisor no se usa.

El resumen del bloque se encuentra en el cuadro 5.2.

Low Pass Filter	
Parámetro	Valor
Decimation	1
Gain	1
Sample rate	$44.1 * 10^3$
Cuttoff Freq	$15 * 10^3$
Transition Width	$1 * 10^3$
Window	Blackman-H

Cuadro 5.2: Configuración del bloque "Low Pass Filter" en el tx FM.

Bloque 3: FM Preemphasis

Es el amplificador de preénfasis. Con esto se mantiene una relación a ruido constante después de la modulación.

- Sample Rate

Debe ser el mismo valor del anterior bloque: 44.1 kHz.

- Tau

Las redes de preémfasis comerciales utilizan una constante de tiempo de $75\mu s$ que es el tiempo de carga de un circuito RL o RC. Una red de preénfasis involucra un circuito RL [31]. Éste valor también es el predeterminado.

El resumen del la configuración de este bloque es el presentado en el cuadro 5.3.

FM Preemphasis	
Parámetro	Valor
Sample rate	$44.1 * 10^3$
Tau	$75 * 10^-6$

Cuadro 5.3: Configuración del bloque "FM Preemphasis" en el tx FM.

Bloque 4: WBFM Transmit

Cuando el valor de la desviación máxima de frecuencia es grande, la onda FM es llamada FM de banda ancha, y el índice de modulación también puede tomar valores grandes. Cuando el valor de la desviación máxima de frecuencia es muy

pequeño se le llama FM de banda angosta y el índice de modulación es también muy pequeño [48]. Este bloque genera una FM de banda ancha o WBFM (Wide Band FM). Las propiedades del bloque son las siguientes:

- Audio Rate

Es la frecuencia de muestreo de entrada al bloque: 44.1 kHz.

- Sample Rate

Es la frecuencia de muestreo de la señal modulada como el ancho de banda de la señal de FM es de 240 kHz, se debe muestrear a un mínimo de 480 kHz. El Sample Rate tiene que ser múltiplo del parámetro anterior (Audio Rate), no es posible configurar otros valores. En éste transmisor el valor del parámetro es 396.9 kHz, valor seleccionado debido a que el hardware no puede generar exactamente las 441 000 muestras por segundo. El cociente entre el parámetro Audio Rate y Sample Rate es para nuestro caso 9.

- Tau

Constante de tiempo de carga de un circuito RC o RL. Valor fijado a cero pues en el bloque de preénfasis ya fue considerado.

- Max desviation

Es la desviación máxima de frecuencia de la señal modulada. Mediante la desviación máxima es modificado el índice de modulación de la onda FM. De la teoría de FM explicada anteriormente la desviación máxima de frecuencia se puede escribir como $\Delta f = m * f_m$ (ecuaciones 5.2 y 5.3) por lo que este es el valor programado. El valor m o índice de modulación es una variable y f_m es de 15 kHz, la máxima frecuencia de la señal moduladora. Si la fuente es un tono entonces el valor de f_m es la frecuencia del tono.

En el programa el índice de modulación es controlado con una variable llamada *index_mod* cuyos valores van de 0 a 5. La desviación máxima de frecuencia es controlada con la variable *max_desv* definida como $index\_mod * 15kHz$.

El resumen de la configuración del bloque lo presenta el cuadro 5.4.

WBFM Transmit	
Parámetro	Valor
Audio Rate	$44.1 * 10^3$
Sample Rate	$396.9 * 10^3$
Tau	0
Max desviation	max_desv

Cuadro 5.4: Configuración del bloque "WBFM Transmit" en el tx FM.

Bloque 5: UHD USRP Sink

Este bloque recibe la señal modulada en frecuencia y la envía al USRP para continuar con el procesamiento de la señal y finalmente enviarla a la antena. A continuación se detallan las propiedades del bloque que fueron utilizadas:

- Sample rate

Se envían $396.9 * 10^3$ muestras por segundo.

- Center Frequency

Es la frecuencia del canal de RF en el que se desea transmitir. Se transmite en el 87.5 MHz por defecto, aunque en realidad es posible configurar cualquier otro valor a través de la variable $tunf$.

- Gain

Es el valor de la ganancia que se le dará a la señal en dB. El valor es seleccionado por el usuario a través de la variable *gain*. Valor por defecto de 10 dB.

- Antenna

Es la antena utilizada: TX/RX

- Bandwidth

Este valor se dejó en cero para que el bloque utilice un ancho de banda automático. La configuración del bloque se encuentra en el cuadro 5.5.

UHD USRP Sink	
Parámetro	Valor
Sample rate	$396.9 * 10^3$
Center Frequency	$tunf$
Gain	$gain$
Antenna	TX/RX
Bandwidth	0

Cuadro 5.5: Configuración del bloque “UHD USRP Sink” en el tx FM.

5.4. Implementación

El diagrama real del transmisor FM desarrollado en GNU Radio es el de la figura 5.3.

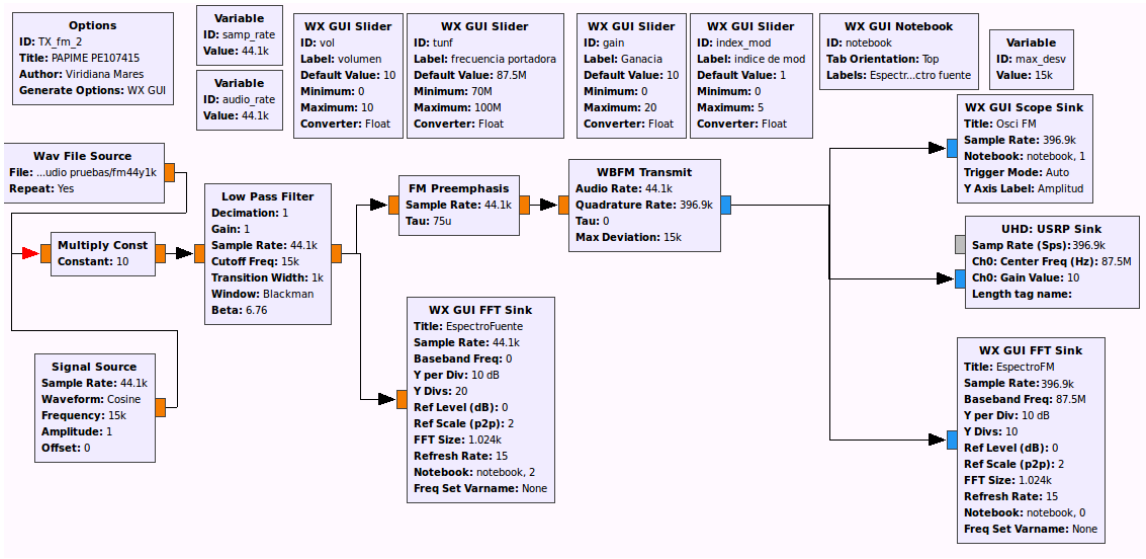


Figura 5.3: Diagrama del transmisor de FM en GNU Radio

El transmisor FM consta de 7 bloques que involucran el procesado de la señal para obtener una modulación en frecuencia. En la figura 5.3 se tiene un bloque más para el transmisor de FM. Después del bloque que lee los datos de un archivo WAV se tiene un bloque llamado “Multiply Const", como su nombre lo indica la señal de entrada es multiplicada por una constante y mediante la constante es controlada la amplitud de la señal moduladora. Por lo que de alguna manera el bloque permite

controlar el volumen de la señal de FM en el receptor, esto solamente por motivos de experimentación. La constante por la cual se multiplica la moduladora es en realidad una variable durante la ejecución del programa llamada *vol* de volumen. La configuración de este bloque se presenta en el cuadro 5.6. Los bloques siguientes son los bloques de preénfasis y el modulador, detallados anteriormente.

Multiply Const	
Parámetro	Valor
IO Type	Float
Constant	vol
Vec Length	1

Cuadro 5.6: Configuración del bloque "Multiply Const" en el tx FM.

El bloque que sigue al modulador es el UHD USRP Sink también descrito a detalle en el diseño del transmisor. Adicionalmente se observan 7 bloques en la figura 5.3 para controlar las variables del programa: *samp_rate*, *audio_rate*, vol, tunf, gain, *index_mod* y *max_desv*, explicadas durante el diseño.

Los bloques *WX GUI FFT Sink* muestran el espectro de la señal moduladora y señal modulada en frecuencia, su configuración se detalla en los cuadros 5.7 y 5.8 respectivamente . El bloque *WX GUI Scope Sink* muestra la señal FM en el dominio temporal, configuración presentada en el cuadro 5.9

WX GUI FFT Sink.	
Parámetro	Valor
Title	Espectro Fuente
Sample rate	$396.9 * 10^3$
Baseband Freq	0
Window	Blackman

Cuadro 5.7: Configuración del bloque "*WX GUI FFT Sink*" para señal moduladora en el tx FM.

En la figura 5.4 se presenta la interfaz del programa con los alumnos. En la figura se aprecian las variables que pueden ser modificadas. Tres pestañas mediante las cuales se puede observar el espectro de la señal FM, su oscilograma y el espectro de la señal moduladora. También se tiene un panel de control con el que es posible

WX GUI FFT Sink	
Parámetro	Valor
Title	EspectroFM
Sample rate	$396.9 * 10^3$
Baseband Freq	$tunf$
Window	Blackman

Cuadro 5.8: Configuración del bloque "*WX GUI FFT Sink*" para señal modulada en el tx FM.

WX GUI Scope Sink	
Parámetro	Valor
Title	OsciFM
Sample rate	$396.9 * 10^3$

Cuadro 5.9: Configuración del bloque "*WX GUI Scope Sink*" para señal modulada en el tx FM

paralizar la señal o aplicar efectos como el promedio o la persistencia con la que se muestra la señal.

Mediante la figura 5.5 se pueden comparar dos señal de frecuencia modulada con diferentes índices de modulación. La señal de la figura 5.5 A) tiene un índice de modulación de 0.5 mientras que la figura 5.5 B) tiene un índice de 3 por lo que su ancho de banda es mucho mayor. En ambos casos la moduladora es una señal de audio.

De igual manera es posible observar el espectro de una señal de FM cuando la moduladora es un tono. En la figura 5.6 A) se aprecia una señal de FM con índice de modulación 0.5. En la figura 5.6 B) se tiene un índice de modulación con valor igual a 1. El tono es de 15 kHz.

La apariencia de una señal modulada en frecuencia en el dominio temporal se puede observar en la figura 5.7, el índice de modulación tiene un valor de 1 y la moduladora es una señal de audio cuyo espectro se encuentra en la figura 5.8.

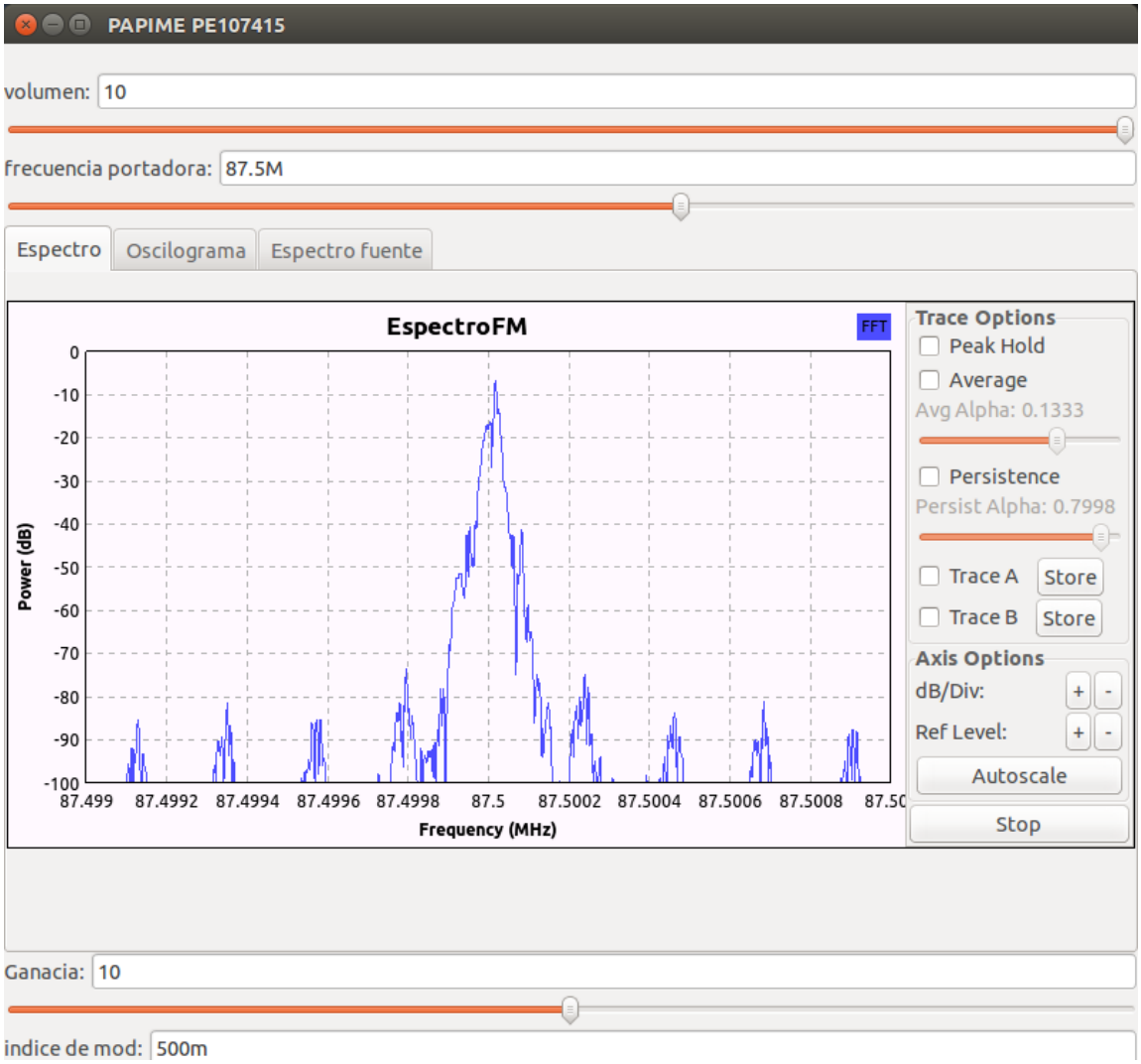


Figura 5.4: Interfaz del transmisor FM con el alumno

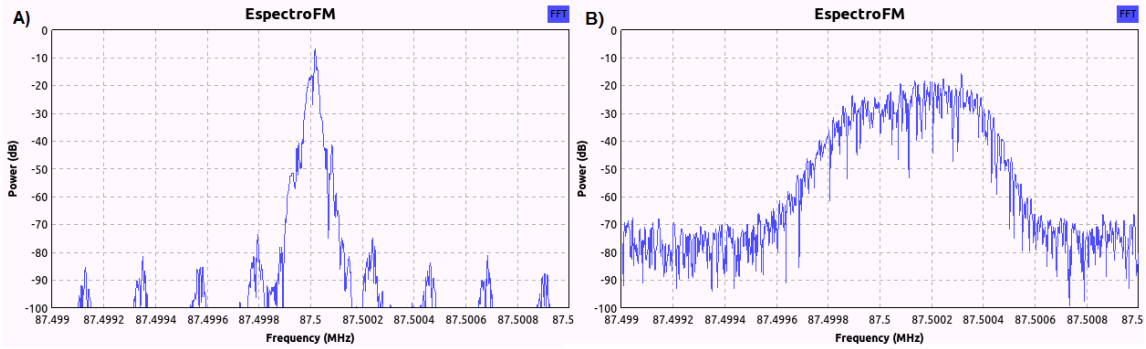


Figura 5.5: A) Señal de FM con índice de modulación de 0.5. B) Señal de FM con índice de modulación de 3.

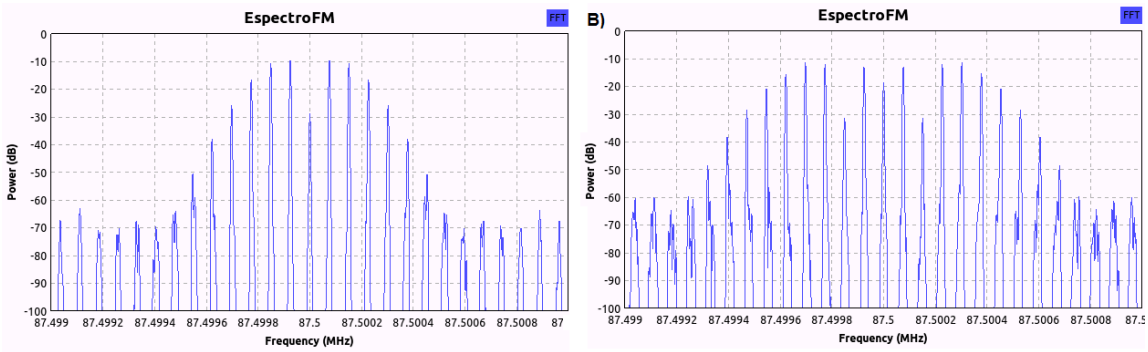


Figura 5.6: Señal de FM cuando la moduladora es un tono. A) Índice de modulación de 0.5. B) Índice de modulación igual a 1.

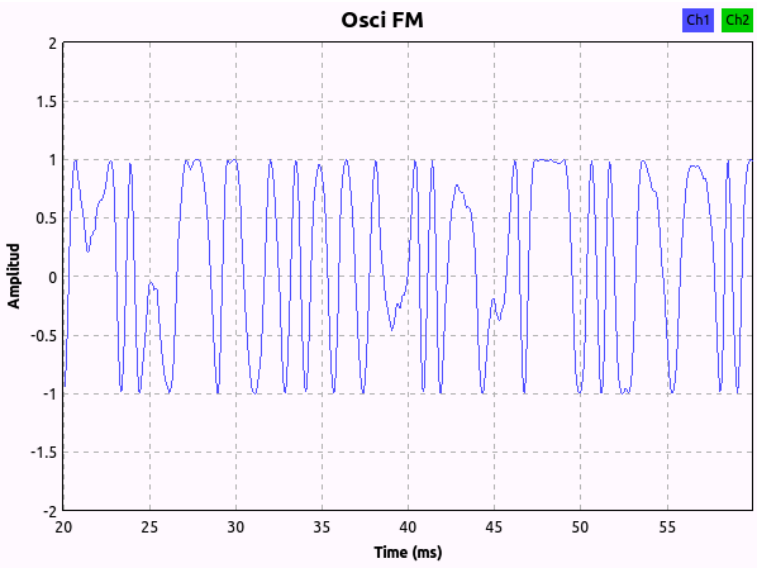


Figura 5.7: Señal de FM en el dominio temporal.

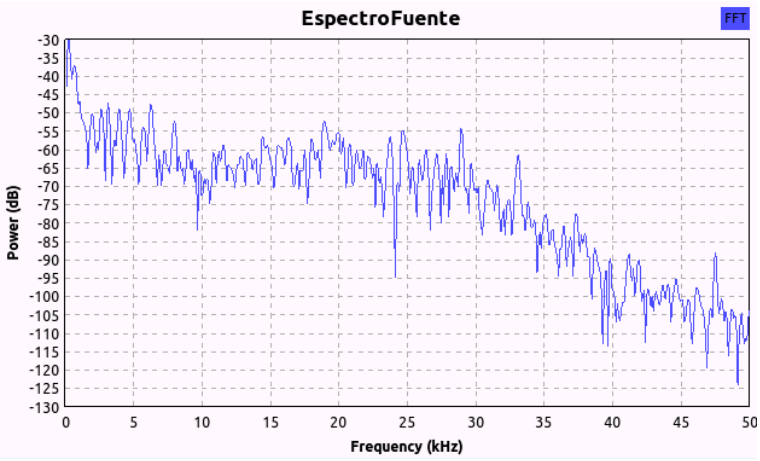


Figura 5.8: Espectro de una señal de audio.

Capítulo 6

El transmisor ATSC

6.1. Datos históricos

ATSC (Advaced Television Systems Committee) es una organización sin fines de lucro que se dedica al desarrollo de normas para televisión digital. Se formó desde 1982 en Estados Unidos por los miembros de la organización de la JCIC (Joint Committee on InterSociety Coordination). Nació con el objetivo de explorar la necesidad de desarrollar un sistema avanzado de televisión y en su caso coordinar y documentar el desarrollo del mismo [49].

En respuesta a una petición presentada por 58 organizaciones de radiodifusión de televisión ante la FCC (Federal Communications Commission) se estableció un comité consultivo (ACATS) que constó de 25 líderes de la industria de televisión. El comité consultivo estableció grupos para estudiar diferentes problemas relativos a servicios, parámetros técnicos, y mecanismos de prueba necesarios para obtener un estándar mejorado de televisión. También se establecieron procesos y mecanismos que permitieran el análisis, prueba y evaluación de los sistemas propuestos [49].

Al inicio, 23 sistemas diferentes fueron propuestos al comité consultivo, desde sistemas analógicos o híbridos (analógico y digital) hasta sistemas de alta definición (HDTV) que utilizaban dos canales de 6 MHz por programa. En 1990 la FCC rechazó todas la propuestas y dejó en claro que se debería tener un estándar completo de televisión de alta definición, transmitir en un solo canal de 6 MHz y sobre las frecuencias existentes de televisión analógica. Para 1991 ya solo se tenían seis diferentes

sistemas de los cuales cuatro eran sistemas de alta definición totalmente digitales. De 1991 a 1992 los seis sistemas fueron probados por tres laboratorios que trabajaron de manera independiente. En 1993 se determinó que un sistema totalmente digital era factible mostrando mejores resultados. Los cuatro sistemas digitales fueron recomendados al comité consultivo con mejoras a cada uno de ellos. El comité consultivo recomendó establecer un único sistema de televisión digital que incorporara los mejores componentes de cada uno de los cuatro sistemas propuestos.

En respuesta a esto, los desarrolladores de los cuatro sistemas digitales formaron una alianza llamada "Digital HDTV Grand Alliance". Los miembros de la Gran Alianza fueron AT&T, General Instrument, North American Philips, MIT, Thomson Consumer Electronics, David Sarnoff Research Center, y Zenith Electronics Corporation [49]. En 1994 fue construido por la Gran Alianza el prototipo final del sistema que cumplía las especificaciones establecidas por el comité consultivo de la FCC. Fue construido de forma modular en diferentes lugares. En 1995 empezaron las pruebas al sistema propuesto por la Gran Alianza y ATSC realizó la documentación del nuevo estándar dividiendo el trabajo en siete grupos de especialistas. El 24 de diciembre de 1996, la FCC adoptó los principales elementos de la norma de televisión digital de ATSC, estableciendo su uso como obligatorio para emitir señales de televisión digital en los Estados Unidos [49].

El 2 de Julio de 2004 México adoptó el estándar A/53 de ATSC para la transmisión digital terrestre de radiodifusión de televisión. Se dio a conocer a través de un acuerdo publicado en el Diario Oficial de la Federación, el cual también establece la política para la transición a la televisión digital terrestre en México [50]. Este acuerdo establecía que el apagón analógico se daría en el 2020 sin embargo seis años después durante el mandato de Felipe Calderón se decidió adelantar la fecha de la transición digital para el 2015. El 31 de marzo de 2015 se venció el plazo fijado por el gobierno de México por lo que el primero de Enero del año en curso cesaron las transmisiones analógicas en todo el territorio nacional. El estándar de televisión analógica que fue utilizado en México por cerca de 70 años fue el National Television Standard Committee (NTSC) de Estados Unidos, el cual permitía la transmisión de señales de televisión analógica en blanco y negro, así como de color [51].

6.2. Características de transmisión de una señal ATSC

El diagrama más básico de un transmisor de televisión digital terrestre, definido por la ITU-R es el mostrado en la figura 6.1 [52]. El diagrama consta de tres subsistemas: codificación y compresión de fuente, multiplexación y transporte, y transmisión de la señal de RF. El primer subsistema implica la codificación y el uso de métodos para la reducción de los bits que representan la fuente, el siguiente subsistema que es la multiplexación implica dividir la señal digital de audio y vídeo en paquetes para unir todos los flujos de entrada en uno solo. En televisión digital se utiliza el MPEG-2 transport stream para empaquetar los datos y multiplexarlos, este formato fue desarrollado para aplicaciones en las que el ancho de banda es reducido por lo que es necesario un transporte de información eficiente. El último subsistema consiste básicamente en la codificación de canal y la modulación, cabe destacar que para el estándar ATSC existen dos modos en la modulación, modo de transmisión terrestre (8-VSB) que es el más usado y el modo de altas tasas de transmisión (16-VSB) [53].

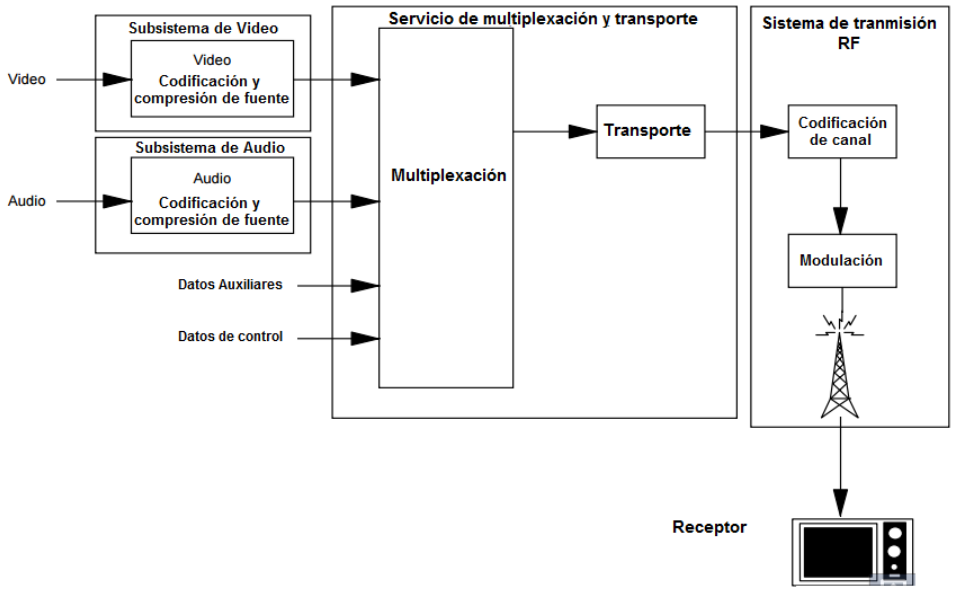


Figura 6.1: Modelo de transmisor digital terrestre definido por la ITU-R .

En ATSC como en la mayoría de los estándares de televisión digital se utiliza un

formato MPEG-2 para la señal en banda base. La señal de vídeo está codificada en MPEG-2 mientras que la señal de audio esta codificada en Dolby AC-3. En ATSC las señales de vídeo pueden ser de dos tipos: SDTV (Televisión de Definición Estándar) o señales de HDTV (Televisión de Alta Definición). El modo de transmisión terrestre o modulación 8-VSB puede tener una tasa de hasta 19.39 Mbps en el ancho de banda de 6MHz. El modo 8-VSB se explica de forma general en el siguiente párrafo.

En general los ocho niveles de la modulación se consiguen a partir de un codificador de Trellis de ocho niveles. Para llegar a la modulación 8-VSB primero se genera una modulación por desplazamiento de amplitud 8ASK, dicha señal modula a una portadora en amplitud por lo que se genera un espectro de doble banda lateral [54]. El modo de modulación VSB es decir banda lateral vestigial, consiste en tomar de un espectro de doble banda lateral, una banda completa y el vestigio de la otra banda, esto se consigue mediante un filtro de respuesta suave como el representado en la figura 6.2, el cual permite el paso de una banda lateral y de las primeras componentes en frecuencia de la otra banda, la banda vestigial se utiliza con el objetivo de incrementar la relación señal a ruido en bajas frecuencias del mensaje.

Figura 6.2: Doble banda lateral y banda lateral vestigial respectivamente.

La modulación descrita anteriormente es un método de portadora única basado en modulación IQ [54] que usa solo el eje I, por lo que su constelación son 8 puntos todos sobre el eje I como se muestra en la figura 6.3.

El diagrama general de un transmisor de ATSC es el mostrado en la figura 6.4. Consta de varias etapas y como se puede observar los datos son procesados en primer lugar por un aleatorizador, posteriormente son procesados por un codificador de Reed-Solomon donde se añaden 20 bytes de paridad a cada paquete del MPEG-2. El bit de sincronización de cada paquete de datos no es tratado de la misma forma que los datos, pues mediante este bit se localizan los paquetes de datos. Los datos son

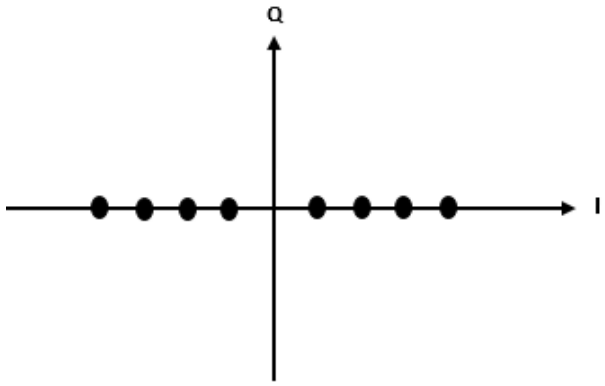


Figura 6.3: Constelación de una señal 8ASK

entonces intercalados por el siguiente bloque y luego procesados por el codificador de Trellis. El aleatorizador de datos, el codificador de Reed Solomon, el intercalador y el codificador de Trellis son usados para tener una corrección tipo FEC (Forward Error Correction) [54] por lo que también al conjunto de estos bloques se les llama etapa FEC. En el multiplexor son añadidos datos de sincronización de segmento y de campo de datos. Finalmente se inserta una portadora piloto y los datos entran al modulador VSB para su transmisión a la frecuencia de RF conseguida por el convertidor D/A. En los siguientes subcapítulos se describirá con más detalle cada bloque del transmisor.

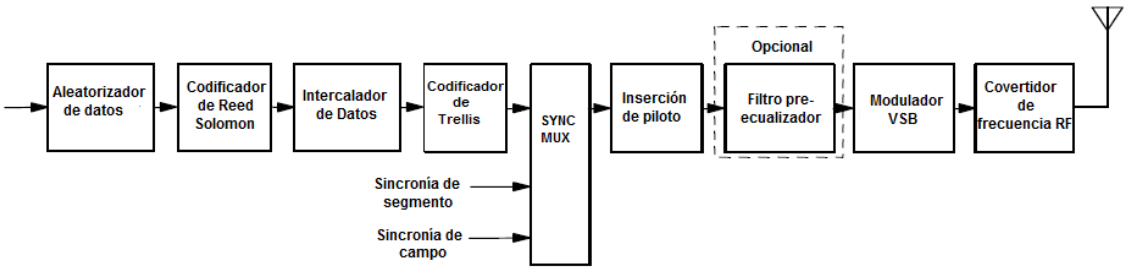


Figura 6.4: Diagrama general de un transmisor ATSC

6.2.1. Aleatorizador de datos

Lo primero que se realiza antes de hacer cualquier procesado a la señal es la sincronización de la misma por lo tanto esto es realizado en la interfaz de banda base. El inicio y fin de cada paquete del MPEG-2 debe ser identificado y esto se logra mediante un byte de sincronía que siempre es el primer byte de cada paquete de 188 bytes [55].

Una vez que los datos han sido sincronizados se ”surten” en el aleatorizador con la excepción de las sincronizaciones de segmento y de campo. El flujo de transporte MPEG-2 incluye tablas PSIP y flujos elementales de vídeo MPEG-2 y audio codificado en Dolby AC-3. La tasa de datos de entrada es de 19.3926585 Mb/s [54].

- PSIP es un Protocolo de Información de Programa y Sistema, es utilizado para transportar los metadatos de cada uno de los canales y para publicar información de los programas de televisión, lo que lleva a la generación de tablas que contienen toda esta información.
- MPEG-2 es un grupo de estándares para la codificación de audio y vídeo digital, el flujo de transporte MPEG (MPEG transport stream) contiene no solo datos de vídeo y audio si no también datos del protocolo PSIP, este tipo de archivos tiene extensión ts.
- Dolby AC-3 es una tecnología de compresión de audio que básicamente elimina todos los sonidos originales que el ser humano no es capaz de percibir.

El aleatorizador de datos fragmenta las secuencias largas de unos y ceros que causarían problemas de sincronización en el receptor. Éste bloque ejecuta una operación lógica XOR entre los datos entrantes y una secuencia binaria pseudo-aleatoria (PRBS) generada con un registro de corrimiento de 16 bits y nueve bits de retroalimentación. Ocho de las salidas del registro de corrimiento son seleccionadas para formar el byte pseudoaleatorio, cada bit del byte pseudoaleatorio se utiliza con un bit de entrada para la operación XOR [53]. La palabra pseudo-aleatoria es definida en un momento durante el intervalo de sincronización. Este proceso da lugar a lo que se conoce como espectro disperso o uniformemente distribuido, cabe destacar nuevamente que la información de sincronización no pasa por este proceso.

6.2.2. Codificador de Reed-Solomon

El siguiente bloque es un codificador de Reed-Solomon. Éste tiene un esquema de codificación de bloque que puede corregir ráfagas de errores hasta cierto límite determinado por la cantidad de redundancia. El código de RS es un código no binario pues la corrección se realiza a nivel de símbolo y no de bit. Es sistemático debido a que las palabras a codificar no son alteradas y solamente se añaden los símbolos de paridad. Es cíclico pues si se desplaza circularmente una palabra válida, se produce otra palabra válida. Otra característica es su linealidad pues si se suman dos palabras del código se produce una palabra válida para el código [56].

El codificador procesa un bloque de símbolos sin codificar a los que agrega redundancia y por lo tanto genera un bloque de mayor longitud. El codificador de RS en el transmisor de ATSC agrega 20 bytes a los 188 bytes existentes dando un total de 208 bytes por paquete [54]. Un Reed-Solomon se especifica como RS(n,k), donde n es el número total de símbolos después del codificador y k es el número de símbolos del mensaje original, por lo que el codificador de RS para un transmisor de ATSC se especifica como RS(208, 188). Con los 20 bytes agregados es posible tolerar hasta 10 bytes erróneos por paquete que pueden ser corregidos en el receptor. Si el paquete contiene más de 10 bytes erróneos la corrección de error de Reed-Solomon falla y el paquete es descartado en el receptor por el decodificador MPEG. Para marcar un paquete del Transport Stream como erróneo, el bit indicador de error en el encabezado del flujo de transporte es puesto en 1. En la sección 6.2.9 se describe con detalle el Transport Stream y su encabezado. Respecto al byte de sincronía éste es verificado y remplazado [55].

6.2.3. Entrelazado de Datos

Después del codificador de Reed-Solomon se encuentra un bloque encargado de dispersar los datos en el tiempo con el objetivo de evitar errores ráfaga prolongados [55]. El intercalado es a 4 ms de profundidad o visto de otra forma tiene una longitud de 52 segmentos. Solo los datos (incluyendo los bytes de paridad del RS) deben ser intercalados [53]. Los paquetes generados son también de 208 bytes utilizando un byte de sincronía. Es decir este bloque toma 207 bytes de cada paquete y los procesa.

Alterar los datos antes y después del codificador de Reed Solomon mediante el aleatorizador y la etapa de entrelazado respectivamente hace que la codificación sea sumamente robusta. Para tener una idea aproximada de esto, supóngase que la tasa de errores de un caudal binario sin protección contra errores es de $10^{-}3$, cuando se utiliza este tipo de codificación la tasa de errores puede reducirse a $10^{-}6$. En la practica se consiguen tasas de error del orden de $10^{-}9$ o menores [56]. Aun así el empleo de una sola etapa de codificación no es suficiente debido al medio de propagación.

6.2.4. Codificador de Trellis

Con el objetivo de conseguir una máxima protección contra errores los sistemas de televisión utilizan dos niveles de codificación (codificación concatenada), el primer nivel es llamado código externo y el segundo es llamado código interno. El código interno es el que actúa directamente sobre el modulador y es un código convolucional mientras que el código externo es un código de bloque [56]. En el transmisor de ATSC el código externo o código de bloque es el codificador de Reed Solomon descrito anteriormente. El codificador de Trellis es el código interno dentro del transmisor de ATSC.

Los datos entran a un segundo bloque de corrección de error. Un codificador convolucional de Trellis lo que significa que la codificación se da por secuencias continuas y no por bloques. Al codificador entran los paquetes de 207 bytes que fueron procesados por el bloque anterior.

El proceso es el siguiente: el flujo de bits se divide en dos, un bit entra a un pre-codificador con una relación de código de 1, mientras que el otro bit es el que entra al codificador de Trellis con una relación de código de 1/2 es decir por cada bit que entra salen dos [54]. Otra manera de ver este proceso es que cada byte es dividido en 4 símbolos de dos bits y luego por cada símbolo de dos bits se obtienen 3 bits. Éste bloque tiene una relación de código global de 2/3 [53]. Enseguida los tres flujos de bits generados, dos por el codificador de Trellis y uno por el pre-codificador entran a un mapeador. Cada símbolo de 3 bits es mapeado a uno de los 8 valores reales que forman los ocho niveles de la modulación (-7, -5, -3, -1, 0, +1, +3, +5, +7) [53].

A la salida de este bloque se tienen 828 símbolos por paquete donde cada símbolo es formado por 3 bits. Es decir 4 símbolos por los 207 bytes procesados (207 bytes * 4 símbolos/byte = 828 símbolos).

6.2.5. Multiplexor y sincronización de datos

Éste se encarga de producir a intervalos definidos patrones de sincronización especiales que se transmiten como la información de sincronización para el receptor [54]. Estos patrones se utilizan para la sincronización de campo y segmento. Para realizar esto se insertan señales que ayudan como la señal de sincronización de segmento y de campo [55]. Los datos codificados por la etapa FEC, la sincronización de segmento y la sincronización de campo producidas son combinadas en el multiplexor [54].

6.2.6. Inserción de piloto

Una señal piloto que también ayuda en el receptor a la de-modulación de la señal de RF es insertada posteriormente justo antes de la modulación en amplitud. Del bloque anterior se tiene una señal que toma 8 niveles discretos de amplitud -7, -5, -3, -1, +1, +3, +5 y +7 [53], a dicha señal se le agrega una componente continua con amplitud relativa de +1.25, por lo que todos los valores originales de la señal [54] son modificados. Este desplazamiento de DC se realiza con el propósito de crear una pequeña portadora piloto. Ésta piloto es añadida tanto a datos de carga como a datos de sincronización [53].

6.2.7. Modulador VSB

La modulación en amplitud se produce en un mezclador que tiene como entradas la señal de 8 niveles y una portadora [54]. Después de éste mezclador se obtiene la portadora piloto de ATSC.

El modulador de VSB recibe $10.76 * 10^6$ símbolos por segundo por lo que el ancho de banda mínimo en el caso de una transmisión de doble banda lateral sería de 10.76 Ms/s [54]. El canal disponible solo tiene un ancho de banda de 6 MHz entonces la señal pasa a través de un filtro que en si es el filtro de banda lateral vestigial comentado anteriormente, obsérvese la figura 6.5.

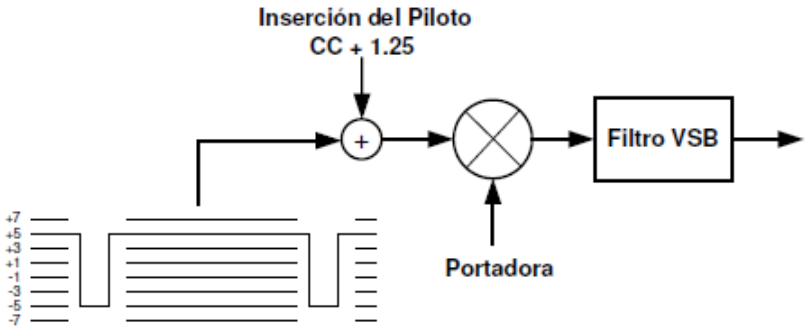


Figura 6.5: Modulación e inserción de piloto.

6.2.8. Descripción del espectro, constelación y desempeño del sistema 8-VSB

La respuesta del filtro de banda lateral vestigial es plana en casi toda la banda excepto en las bandas de transición, es la repuesta de un filtro coseno elevado de fase lineal [57], respuesta observada en la figura 6.6. La piloto de la señal ATSC está situada 309.441 kHz por encima del borde inferior del canal de televisión digital [53]. La figura 6.6 es en realidad el espectro ideal de una señal ATSC.

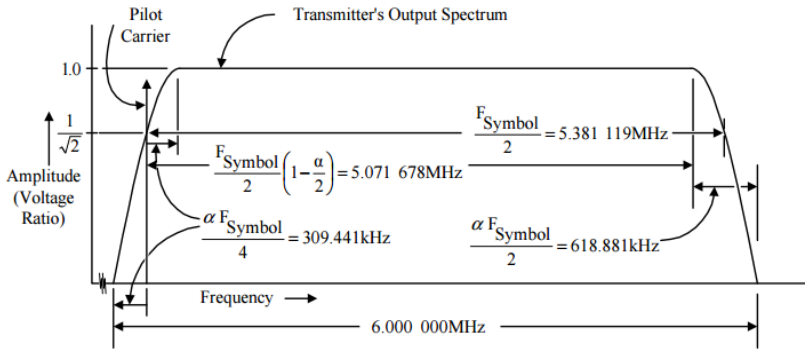


Figura 6.6: Espectro ideal ATSC.

En la figura 6.7 se tiene la constelación de una señal 8-VSB, cabe destacar que esta constelación es producida después del filtrado de banda lateral, debido a este filtrado el diagrama de constelación aparece con líneas verticales. En realidad el filtrado de la señal no es hecho por un filtro analógico convencional. La señal que llega al filtro es dividida en dos, una es aplicada directamente al mezclador I y la

otra pasa primero por un transformador de Hilbert [1] y enseguida entra al mezclador Q. El transformador Hilbert junto con el modulador IQ causa la supresión parcial de la banda lateral inferior que se obtiene debido a la configuración de las amplitudes y fases de la señal modulada en amplitud. [54].

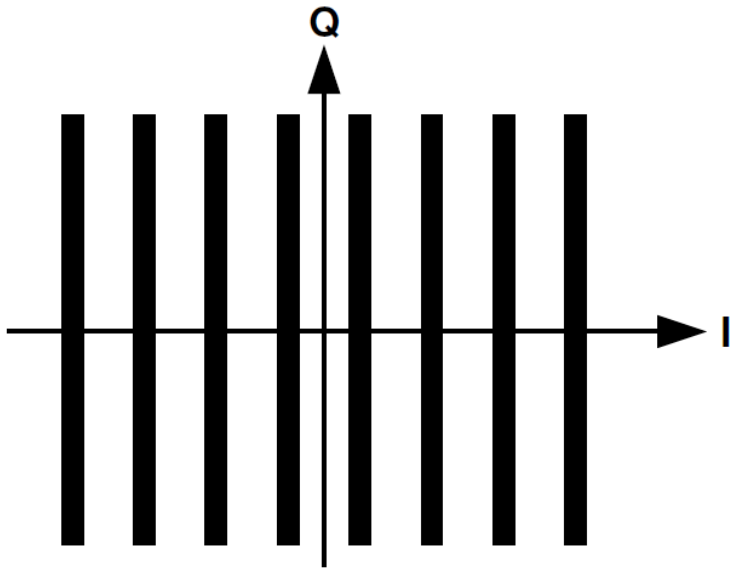


Figura 6.7: Constelación de una señal 8-VSB.

En la figura 6.8 se presenta una curva del BER (Bit Error Rate) contra la energía por símbolo entre la densidad espectral de potencia de ruido (E_s/N_0). El sistema de codificación fue conectado al sistema de decodificación mediante un generador de ruido aditivo Gaussiano. Esta curva fue obtenida a través de una simulación con el objetivo de analizar el desempeño de la codificación de canal utilizada en el sistema 8-VSB. Mediante esta simulación se concluyó que la codificación de canal del sistema ATSC 8-VSB es robusto frente a ruido aditivo Gaussiano y frente ruido impulsivo [58].

[1] Un transformador de Hilbert es un desfasador de 90 grados para todas las frecuencias de una banda [54].

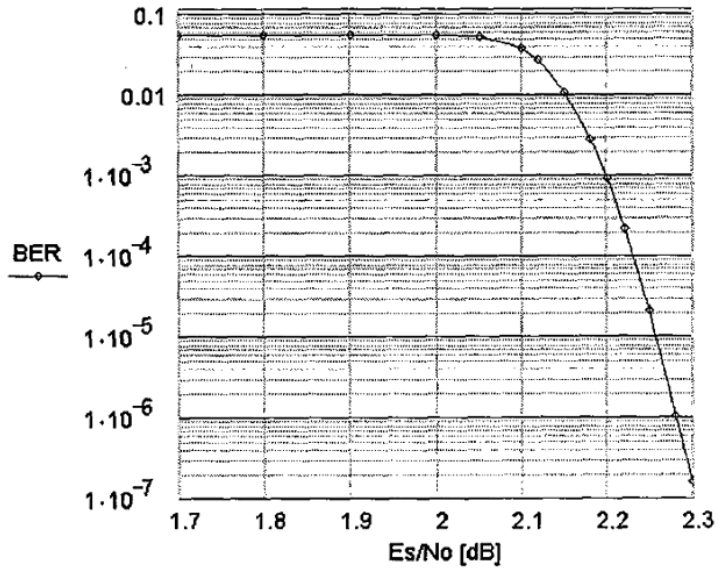


Figura 6.8: BER para una codificación de canal 8-VSB.

6.2.9. Descripción de la trama de datos

Los datos de ATSC se encuentran organizados en dos campos, cada campo consta de 313 segmentos como se muestra en la figura 6.9. El primer segmento de cada campo es una señal de sincronización, esta señal contiene varias secuencias pseudo-aleatorias que son utilizadas por el ecualizador del receptor [54]. Cada segmento de los 312 restantes contiene un paquete de 188 bytes más una cabecera llamada RS-FEC, cada paquete de 188 bytes dedica un byte para sincronía.

Un segmento de datos consiste en un total de 832 símbolos, los 4 primeros están dedicados a la sincronización, cabe destacar que estos bits de sincronización también son los bits de sincronización de los paquetes de 188 bytes del MPEG-2. Esto nos lleva a que los 828 símbolos restantes de los 832 totales equivalen a paquetes de 187 bytes de información más la cabecera RS-FEC. Para obtener los 8 niveles se tienen 3 bits por símbolo, es decir se tienen 828 símbolos por 3 bits cada uno, resultando un total de 2484 bits por segmento. Esto puede comprobarse efectuando los siguientes cálculos [53]:

- Total de bytes por segmento: 187 $bytes + 20$ $bytes\ de\ cabecera = 207$ bytes

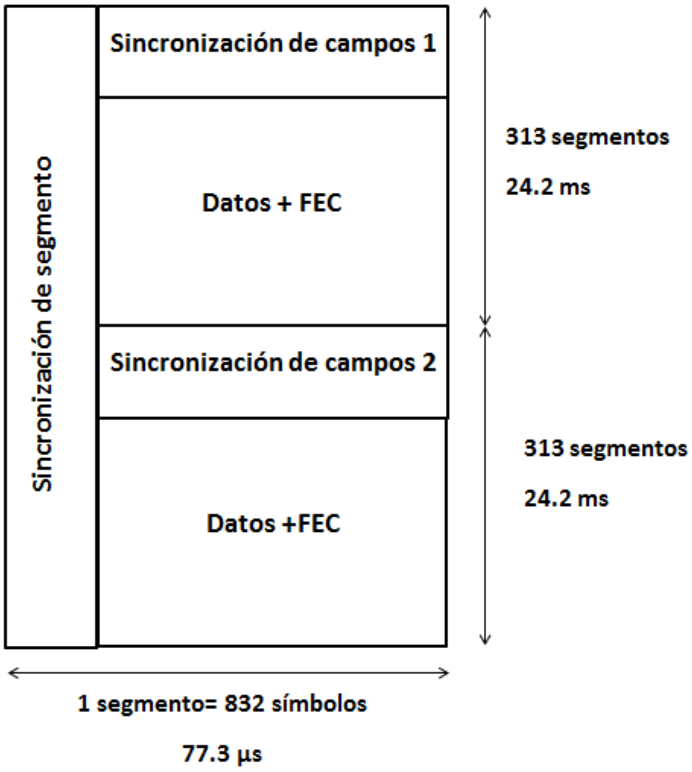


Figura 6.9: Cuadro de datos 8VSB.

- Total de bits por segmento: 207 $bytes \times 8\ bits = 1656$ bits
- Total de bits por segmento después del codificador de Trellis.
  El codificador Trellis tiene una relación de código de valor 3/2 por lo que se tiene: $\frac{3}{2} \times 1656\ bits = 2484$ bits

Este último resultado coincide con lo calculado anteriormente y también se concluye que este es el número de bits necesarios para enviar un paquete con protección pues ya se le han insertado los 20 bytes de cabecera.
Con todo lo anterior también es posible hacer los cálculos para la tasa de transmisión [53]:

- Tasa de símbolos dada por la siguiente ecuación: $\frac{4.5}{286} \times 684\ bits = 10.76$ MHz
- Calculo de la tasa de segmentos: $\frac{10.76}{832} = 12.04 * 10^3$ Segmentos/s

En general se envían los bits de sincronización de datos de campo, los bits de sincronización de segmento de datos y los bits de información.

6.3. Diseño

El diagrama general de un transmisor programado en el software GNU Radio es muy similar al de la figura 6.4. De cualquier manera en la figura 6.10 se presenta el diagrama del transmisor de ATSC con los bloques tal y como existen en GNU Radio. Es importante aclarar que otros bloques fueron necesarios para el manejo de datos en el software, los cuales serán presentados en la parte de implementación del transmisor. A continuación se explicarán los bloques del diagrama 6.10 así como el nombre que asigna el software GNU Radio a los paquetes de datos antes y después de cada bloque.

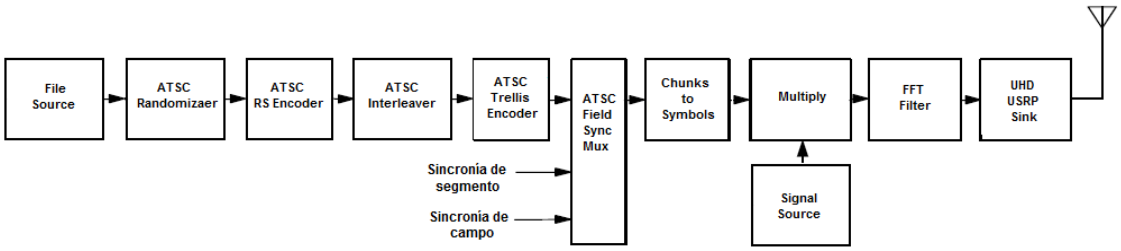


Figura 6.10: Transmisor de ATSC programado en GNU Radio

Bloque 1: File Source

Este bloque lee el valor de los datos en formato binario del archivo especificado [44]. El archivo fuente puede ser obtenido con un bloque de salida o sumidero de GNU Radio. Es mediante este bloque que el transmisor obtiene el archivo con formato MPEG-2 mencionado anteriormente. La descripción de cada una de las propiedades de este bloque es la siguiente:

- File

Especifica el nombre del archivo fuente y la ruta.

- Output Type

Especifica el tipo de dato que se tendrá a la salida. Para este transmisor el tipo de dato a la salida es el byte.

- Repeat

Se indica si el archivo solo debe ser leído una vez o si se debe repetir la lectura del archivo hasta el fin de la ejecución del programa.

- Vec Length

Este campo debe ser siempre de tipo entero, especifica la longitud del vector de ítems a procesar. En este caso los ítems son bytes.

En la tabla 6.1 se resume la configuración del bloque.

File Source	
Parámetro	Valor
File	archivo.ts
Output Type	byte
Repeat	yes
Vec Length	1

Cuadro 6.1: Configuración del bloque "File Source" en el tx ATSC.

Bloque 2: ATSC Randomizer

Su función es la del bloque aleatorizador descrito anteriormente. El paquete a la entrada del bloque es llamado *atsc_mpeg_packet*. El paquete a la salida del bloque es llamado *atsc_mpeg_packet_no_sync* [57]. Este bloque no tiene ninguna propiedad que pueda ser modificada.

Bloque 3: ATSC RS Encoder

Implementa un codificador de Reed-Solomon para ATSC. El paquete a la entrada del bloque es identificado como *atsc_mpeg_packet_no_sync* y a la salida como *atsc_mpeg_packet_rs_encoded* [57]. De igual manera es un bloque sin propiedades.

Bloque 4: ATSC Interleaver

Éste bloque es el que realiza el entrelazado de los datos de ATSC. El paquete a la entrada se llama *atsc_mpeg_packet_rs_encoded* y a la salida su nombre es el mismo. Es un bloque sin propiedades.

Bloque 5: ATSC Trellis Encoder

Implementa un codificador de Trellis de ATSC. Los paquetes a la entrada del bloque se identifican como *atsc_mpeg_packet_rs_encoded* mientras que a la salida del bloque se identifican como *atsc_data_segment* [57]. Bloque sin propiedades.

Bloque 6: ATSC Field Sync Mux

Es el bloque de multiplexor y sincronía de datos de ATSC. Los paquetes a la entrada se llaman *atsc_data_segment* y los paquetes a la salida se identifican con el mismo nombre [57]. También es un bloque sin propiedades a configurar.

Bloque 7: Chunks to Symbols

De forma general, este bloque mapea cadenas de bits a valores flotantes o complejos. Realiza el mapeo de símbolos a valores complejos o flotantes [59], esto se indica a través de su tabla de mapeo. Con este bloque se inserta la señal piloto, sumando le a cada símbolo generado por el codificador de Trellis un valor de 1.25. Los parámetros o propiedades del bloque son las siguientes:

- Symbol Table

Es la lista o tabla que asocia las entradas con los símbolos de salida. El formato es $out[nD + k] = symbol\_table[in[n]D + k]$ donde $k = 0, 1, ..., D - 1$ [59].

- Dimension

Es la dimensión de la tabla de mapeo, el valor predeterminado es uno. La configuración se detalla en el cuadro 6.2.

Bloque 8: Chunks to Symbols	
Parámetro	Valor
Input Type	Byte
Output Type	Complex
Symbol Table	$[symbol + 1.25 for symbol in [-7, -5, -3, -1, 1, 3, 5, 7]]$
Dimension	1

Cuadro 6.2: Configuración del bloque "Chunks to Symbols" en el tx ATSC.

Bloque 8: Multiply

El bloque tiene la función del mezclador estudiado anteriormente, es un bloque que multiplica toda la señal procesada con una portadora. La configuración del bloque se muestra en el cuadro 6.3.

Multiply	
Parámetro	Valor
IO Type	Complex
Num Inputs	2
Vect Length	1

Cuadro 6.3: Configuración del bloque "Multiply" en el tx ATSC.

Bloque 9: Signal Source

Con este bloque se genera una portadora que debe estar situada a 309.441 kHz de la frecuencia inferior del espectro de ATSC, valor que es llamado $pilot\_freq$ en el programa. La variable $symbol\_rate$ es de $10.76 * 10^6$ valor que corresponde con la tasa de símbolos generada y calculada previamente. Las propiedades de este bloque fueron descritas a detalle en el diseño del transmisor AM. En el cuadro 6.4 se encuentra el resumen de la configuración de éste bloque.

Bloque 10: FFT Filter

Es el filtro VSB del transmisor de ATSC.

Signal Source	
Parámetro	Valor
Output Type	Complex
Frequency	$-3000000 + pilot\_freq$
Sample Rate	$symbol\_rate$
Waveform	Cosine
Amplitude	0.9
Offset	0

Cuadro 6.4: Configuración del bloque "Signal Source" en el tx ATSC.

- Type

Parámetro que indica el tipo de dato a la entrada, salida y el tipo de dato del vector (taps) a utilizar [59].

- Taps

Este parámetro se configura dependiendo del filtro que se desee implementar: filtro paso bajas, paso altas, paso banda, banda de rechazo, Hilbert, coseno elevado, etc. Para un filtro coseno elevado debe configurase de la siguiente manera [60]:

```
gr.firdes.root_raised_cosine (  \
double gain,
double sampling_freq,
double symbol_rate
double alpha
int ntaps )
```

La configuración del bloque se presenta en el cuadro 6.5.

Bloque 11: UHD: USRP Sink

Este bloque recibe la señal de ATSC y la envía al USRP para terminar el procesamiento y finalmente radiarla. A continuación se describen los parámetros del bloque utilizados en el transmisor.

- In Type

FFT Filter	
Par.	Valor
Type	Complex to Complex
Deci.	1
Taps	$firdes.root\_raised\_cosine(0.1, symbol\_rate, symbol\_rate/2, 0.1152, 100)$

Cuadro 6.5: Configuración del bloque "FFT Filter" en el tx ATSC.

Los datos flotantes son convertidos a tipo complejo de longitud de 16 bits. Recordar del capítulo dos que el USRP procesa señales de tipo complejo, por lo que a la salida del bloque la señal tiene que ser compleja.

- Sample rate

Es el número de muestras por segundo de entrada al bloque. Se configuró a 10.76 MHz, valor que corresponde con la tasa de símbolos de la señal. Éste parámetro es llamado $symbol\_rate$ en el programa.

- Center Frequency

Ésta es la frecuencia central del canal RF. Es la variable llamada $center\_freq$.

- Gain

Es la ganancia que se le da a la señal, es la variable llamada gain.

- Antenna

La antena utilizada es la TX/RX.

- Bandwidth

El ancho de banda se asigna de manera automática si se deja una valor de 0.

En el cuadro 6.6 se muestra el resumen de la configuración del bloque UHD.

UHD: USRP Sink	
Parámetro	Valor
In Type	Complex int 16
Sample rate	*symbol_rate*
Center Frequency	fc
Gain	gain
Antenna	TX/RX
Bandwidth	0

Cuadro 6.6: Configuración del bloque "UHD: USRP Sink" en el tx ATSC.

6.4. Implementación

Uno de los Transport Stream (TS) utilizado en el transmisor fue obtenido de internet, mientras que el otro fue obtenido con un grabador de señales de ATSC y un convertidor a Transport Stream desarrollado en GNU Radio. Para verificar que los Transport Stream utilizados eran correctos se recurrió a un analizador de TS. El programa fue descargado de internet [61]. Cada TS fue identificado por el analizador como un TS de ATSC y decodificado correctamente. Algunos de los datos obtenidos adicionalmente son los mostrados en la figura 6.11. En la imagen se observa una tasa de bit de 19.39 Mbps, la cual corresponde con la cifra mencionada anteriormente. Este TS tiene 26 secciones de PAT (Program Association Table), 6 secciones de PMT (Program Map Table), no tiene secciones de CAT (Conditional Access Table). Todas estas secciones son tablas PSI (Program Specific Information) lo que quiere decir que son tablas que contienen información de programas. Las siguientes secciones son la ETT (Extended Text Table), la PSIP (Program and System Information Protocol) y EIT (Event Information Table). Estas tres secciones son tablas con información del sistema y del protocolo que utiliza ATSC para transportar los metadatos [62].

El diagrama general del transmisor ATSC programado en GNU Radio es el de la figura 6.12. En él se observa que el primer bloque del transmisor es el File Source tal y como se describió en el diseño, sin embargo el segundo bloque es un bloque llamado ATSC Pad.

El bloque ATSC Pad convierte paquetes de 188 bytes a 256 bytes, esto se hace para resolver un problema llamado alineamiento de buffer o memoria en la compu-

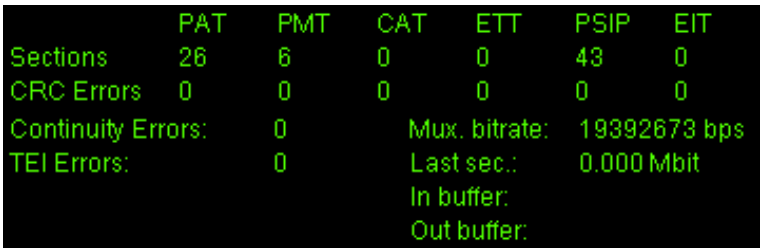


Figura 6.11: Características de un Transport Stream

tadora. Para que el CPU pueda tener acceso a un dato en memoria éste debe estar en una localidad de memoria cuya posición sea múltiplo de su ancho en bytes. Es decir en un sistema de 32 bits (4 bytes) las localidades de memoria deben ser múltiplos de 4. Si los datos no están alineados el acceso a ellos es mucho más lento pues se requieren muchos más ciclos para leer los datos. La computadora utilizada para el desarrollo de los transmisores es de 64 bits (8 bytes) por lo que necesitamos tener datos de ancho que sea múltiplo de 8.

Los paquetes del TS son de 188 bytes por lo que el bloque ATSC Pad añade 68 bits para obtener paquetes de 256 bytes número que además es potencia de 2 ($2^8 bytes$). Para verificar que el bloque funcione correctamente se grabó la salida mediante un bloque File Sink y se analizó el archivo mediante un programa en lenguaje C que muestra el archivo byte por byte en formato hexadecimal, el código se encuentra en el apéndice B. El archivo en formato hexadecimal a la salida del bloque Pad es el de la figura 6.13. En él se observan los paquetes de 188 bytes seguidos de 68 bytes de relleno con valor de 0. Cabe destacar que se observa el byte de sincronía al principio del paquete cuyo valor es de 0x47 [53].

El bloque que sigue es el aleatorizador que recibe 187 bytes para multiplicarlos por la palabra pseudo-aleatoria. En la figura 6.14 se aprecia que el byte de sincronía fue remplazado por 4 bytes. Éstos bytes son banderas que de igual manera contribuyen a la sincronía [63] [64]. Cabe resaltar que el tercer byte lleva la secuencia de los paquetes, en la figura 6.14 se observan los dos primeros paquetes completos y la primera línea del tercer paquete. Ahora se tiene paquetes de 187 bytes más 4 de sincronía, dando un total de 191 bytes de información útil. Para seguir teniendo alineamiento de memoria se tienen ahora 65 bytes de relleno. Los paquetes siguen siendo de 256 bytes.

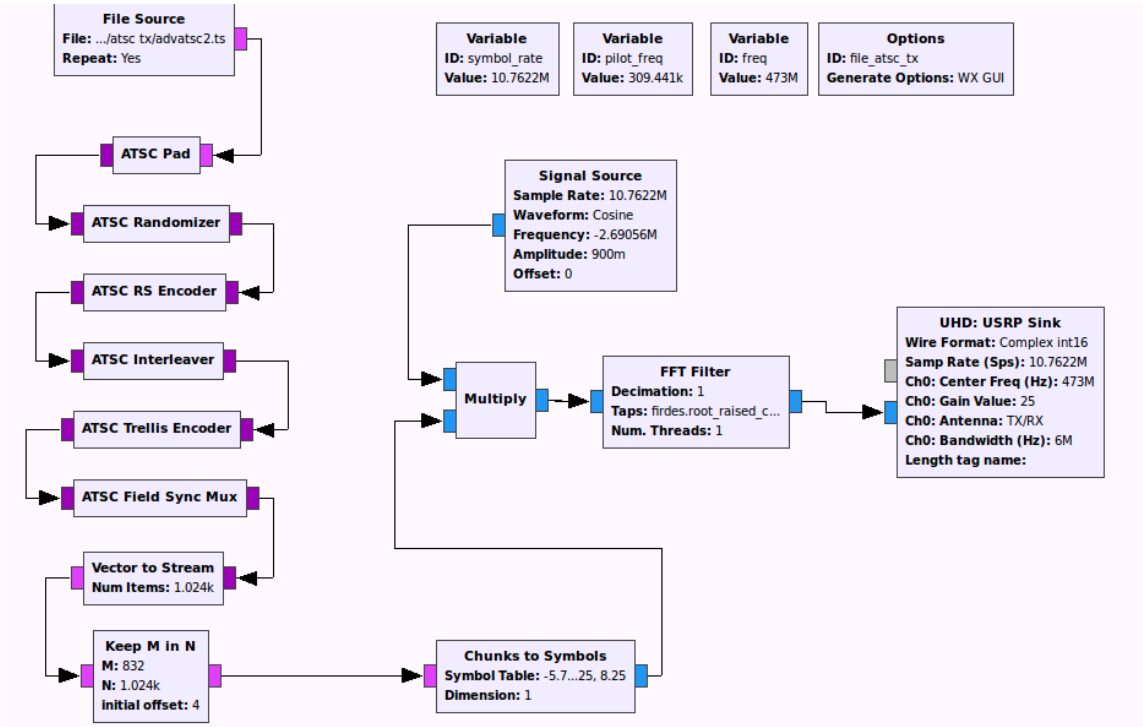


Figura 6.12: Diagrama de un transmisor de ATSC desarrollado en GNU Radio

El cuarto bloque en la implementación del transmisor es el codificador de Reed Solomon y después de este bloque se tienen los datos mostrados en la figura 6.15. Se observa un paquete al cual se le añadieron 20 bytes de paridad. En la figura se indica el inicio y fin del paquete de 187 bytes. Al principio se observan 4 bytes de encabezado. En total se tienen 211 bytes de información útil. Ahora solo se tienen 45 bytes de relleno que son los bytes con valor de 0.

El siguiente bloque intercala los datos cada 52 paquetes. Hay que recordar de la figura 6.9 que cada paquete dura 77.3μs entonces si se multiplica $52 * 77.3\mu$s dan 4.0196 ms que es la profundidad del intercalado visto anteriormente. En la figura 6.16 se observa un paquete que ha sido rellenado en su mayoría con ceros con el objetivo de hacer el entrelazado de datos.

El siguiente bloque es el codificador de Trellis donde por cada dos bits que entran salen tres y cuyo funcionamiento se explicó con detalle anteriormente. El bloque que sigue al codificador de Trellis es el bloque que introduce la sincronización de campo y segmento. En la figura 6.17 se observa un paquete completo a la salida de éste bloque.

	1	2	3	4	5	6	7	8		9	10	11	12	13	14	15	16
1	47	00	31	13	36	fc	57	d3	-	a7	5e	dc	75	d6	05	39	dc
2	63	9d	00	44	66	49	3c	1b	-	1b	46	0d	e7	b7	df	36	0c
3	5a	34	6a	66	da	34	68	d1	-	86	1b	46	8d	1a	30	c3	68
4	d1	a3	46	18	6d	1a	34	68	-	c3	0d	a3	46	8d	18	61	b4
5	68	d1	a3	0c	36	8d	fd	09	-	e2	82	0f	cd	d0	02	2b	01
6	2c	00	f3	fb	7d	00	0f	2c	-	80	99	ff	95	bf	9e	ed	fb
7	3c	50	21	7d	57	60	80	05	-	d4	13	3f	d3	81	7c	fa	00
8	1e	4e	f2	90	b5	b8	c1	0c	-	01	49	e0	03	05	24	2b	62
9	03	83	49	0d	bf	d1	43	46	-	0c	5b	f3	b6	f1	5b	46	8d
10	1a	30	c3	68	d1	a3	46	18	-	6d	1a	34	68	c3	0d	a3	46
11	8d	18	61	b4	68	d1	a3	0c	-	36	8d	1a	34	61	86	d1	a3
12	46	8c	30	da	34	68	d1	86	-	1b	46	8d	1a	00	00	00	00
13	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
14	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
15	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
16	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00

Figura 6.13: Paquete de datos a la salida del bloque “Pad”

El paquete o segmento tiene una longitud de 1024 bytes, número que es múltiplo de ocho y al mismo tiempo potencia de dos. Se tienen 836 bytes de información útil y 188 bytes con valor de 0 (bytes de relleno). En la figura 6.17 es posible observar que la secuencia de bytes añadida por el bloque es la 06 01 01 06, la cual se observa al comienzo de cada segmento y es la sincronización de segmento [64]. Los siguientes bytes de información útil después de esta secuencia son los bytes que se obtuvieron después del codificador de Trellis. Recordar que a la salida del codificador de Trellis se tienen 828 símbolos de 3 bits cada uno. La figura 6.17 muestra los 828 símbolos, en ella se marca el inicio y fin de los 828 símbolos. A manera de comprobación: se tiene un total de 33 filas de información útil del lado izquierdo de la figura 6.17, de lado derecho un total de 20 filas. Por lo tanto se tienen 53 filas es decir 848 bytes (53*16) de información. Después hay que restar 8 bytes de cabecera y 12 bytes de relleno de la última fila con lo que se obtienen los 828 símbolos. Cabe destacar que el bloque Sync también introduce cada 312 segmentos un segmento completo que es la sincronización de campo.

El bloque subsecuente es un bloque llamado Vector to Stream. Para GNU Radio todos los bloques mencionados hasta éste punto procesan vectores de ítems, en este caso procesan vectores de bytes. Éste bloque recibe vectores de bytes o paquetes de bytes y entrega la información byte por byte al siguiente bloque debido a que el

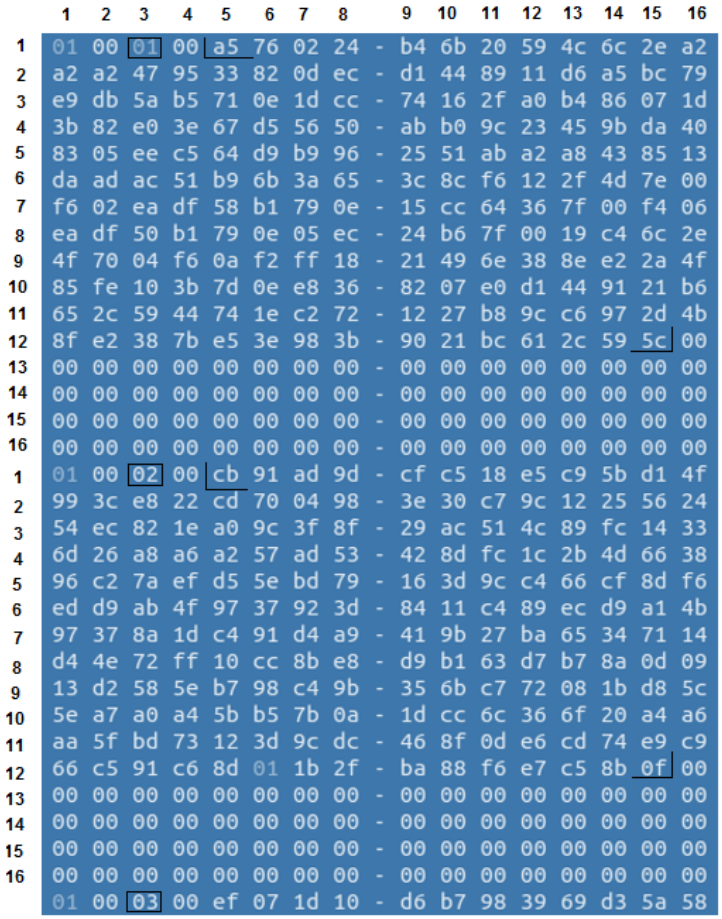
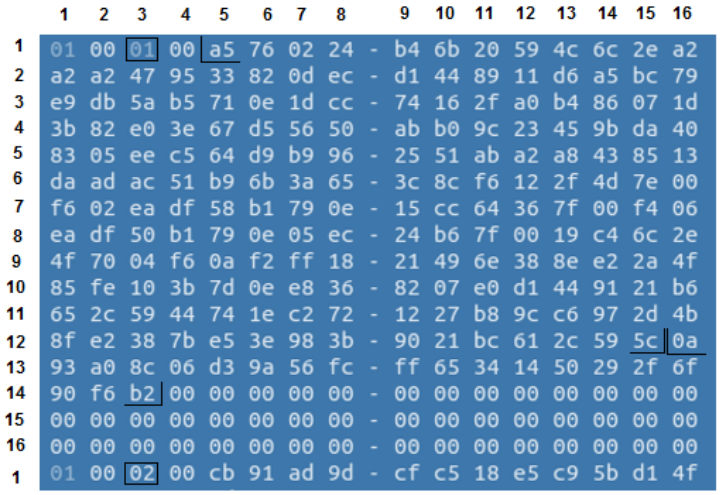


Figura 6.14: Paquete de datos a la salida del bloque ‘Randomizer”

bloque que continuará con el procesamiento de la señal de ATSC no puede procesar los bytes por vectores o paquetes. En éste bloque la información no es modificada y esto se puede constatar comparando la salida del bloque Vector to Stream con la salida del bloque anterior (Field Sync Mux). El comparador de archivos binarios utilizado es el VBinDiff de código abierto y libre distribución, en Linux puede ser descargado mediante el siguiente comando:

```
sudo apt-get install vbindiff
```

El programa muestra los bytes en formato hexadecimal y remarca con color rojo los bytes diferentes. El resultado de la comparación se muestra en la figura 6.18, el



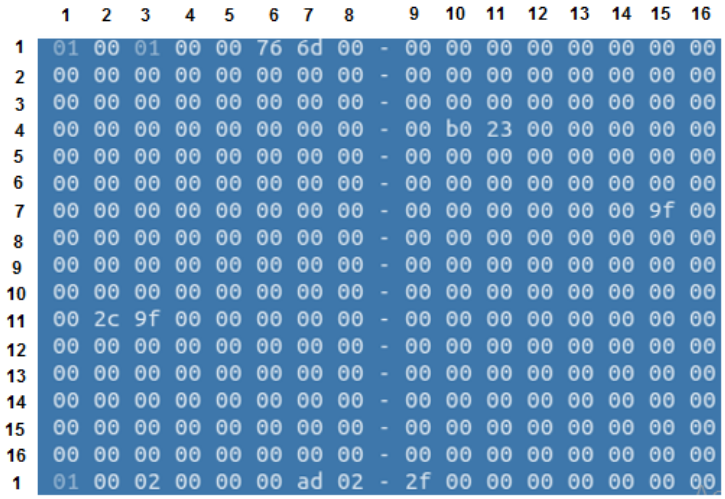
	1	2	3	4	5	6	7	8		9	10	11	12	13	14	15	16
1	01	00	01	00	a5	76	02	24	-	b4	6b	20	59	4c	6c	2e	a2
2	a2	a2	47	95	33	82	0d	ec	-	d1	44	89	11	d6	a5	bc	79
3	e9	db	5a	b5	71	0e	1d	cc	-	74	16	2f	a0	b4	86	07	1d
4	3b	82	e0	3e	67	d5	56	50	-	ab	b0	9c	23	45	9b	da	40
5	83	05	ee	c5	64	d9	b9	96	-	25	51	ab	a2	a8	43	85	13
6	da	ad	ac	51	b9	6b	3a	65	-	3c	8c	f6	12	2f	4d	7e	00
7	f6	02	ea	df	58	b1	79	0e	-	15	cc	64	36	7f	00	f4	06
8	ea	df	50	b1	79	0e	05	ec	-	24	b6	7f	00	19	c4	6c	2e
9	4f	70	04	f6	0a	f2	ff	18	-	21	49	6e	38	8e	e2	2a	4f
10	85	fe	10	3b	7d	0e	e8	36	-	82	07	e0	d1	44	91	21	b6
11	65	2c	59	44	74	1e	c2	72	-	12	27	b8	9c	c6	97	2d	4b
12	8f	e2	38	7b	e5	3e	98	3b	-	90	21	bc	61	2c	59	5c	0a
13	93	a0	8c	06	d3	9a	56	fc	-	ff	65	34	14	50	29	2f	6f
14	90	f6	b2	00	00	00	00	00	-	00	00	00	00	00	00	00	00
15	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
16	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
1	01	00	02	00	cb	91	ad	9d	-	cf	c5	18	e5	c9	5b	d1	4f

Figura 6.15: Paquete de datos a la salida del bloque “RS Encoder”

archivo fue revisado en su totalidad y ningún byte es diferente. En la parte superior se encuentran los datos de salida del bloque Filed Syn Mux mientras que en el lado inferior los datos a la salida del bloque Vector to Stream.

El bloque que sigue en la implementación del transmisor de ATSC es el bloque Keep M in N el cual guarda M elementos de cada N indicados [59]. Este bloque toma 832 elementos de cada 1024 elementos con un offset de 4 por lo que empieza a contar a partir del byte 4. De la figura 6.17 se puede ver que los bytes que guarda son los bytes de sincronización de segmento (secuencia de bytes 06 01 01 06) y los 828 símbolos obtenidos por el codificador de Trellis. Se puede deducir que con este bloque son eliminados los bytes de relleno que servían para alineamiento de memoria, como la información ya no es procesada por paquetes estos ya no son necesarios, también son eliminados los 4 bytes al comienzo de cada paquete. La configuración de este bloque es la mostrada en el cuadro 6.7. Los bloques que siguen son los bloques Chunks to Symbol, Multiply, Signal Source, y el FFT Filter los cuales ya han sido descritos anteriormente.

El último bloque es el bloque UHD USRP Sink el cual envía las muestras al USRP para su procesamiento. Como se observó una característica en la implementación del transmisor que eleva considerablemente la carga computacional es el problema de alineamiento de memoria, sumado a esto se utiliza un bloque más para después



	1	2	3	4	5	6	7	8		9	10	11	12	13	14	15	16
1	01	00	01	00	00	76	6d	00	-	00	00	00	00	00	00	00	00
2	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
3	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
4	00	00	00	00	00	00	00	00	-	00	b0	23	00	00	00	00	00
5	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
6	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
7	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	9f	00
8	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
9	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
11	00	2c	9f	00	00	00	00	00	-	00	00	00	00	00	00	00	00
12	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
13	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
14	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
15	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
16	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
1	01	00	02	00	00	00	ad	02	-	2f	00	00	00	00	00	00	00

Figura 6.16: Paquete de datos a la salida del bloque “Interleaver”

Keep M in N	
Parámetro	Valor
Type	Byte
M	832
N	1024
initial offset	4

Cuadro 6.7: Configuración del bloque “Keep M in N” en el tx ATSC.

quitar los bytes de relleno aumentando el tiempo en el que se procesa la señal. El problema que presenta este transmisor es que la computadora no logra entregar lo suficientemente rápido las muestras al USRP. En la consola de GNU Radio aparece una “U” lo que confirma el problema [15]. Esto también se ve reflejado en el LED del USRP X300 el cual se enciende cuando existe una transmisión, durante la ejecución del transmisor de ATSC este LED no enciende de forma continua.

Aunque el transmisor no trabaja en tiempo real es posible estudiar varios aspectos del estándar ATSC. En la figura 6.19 se muestra el espectro de la señal de ATSC obtenido mediante el bloque $WX\ GUI\ FFT\ Sink$ con un sample rate igual a $10.76 * 10^6$.

En la figura 6.20 se muestra la constelación de una señal de ATSC obtenida a

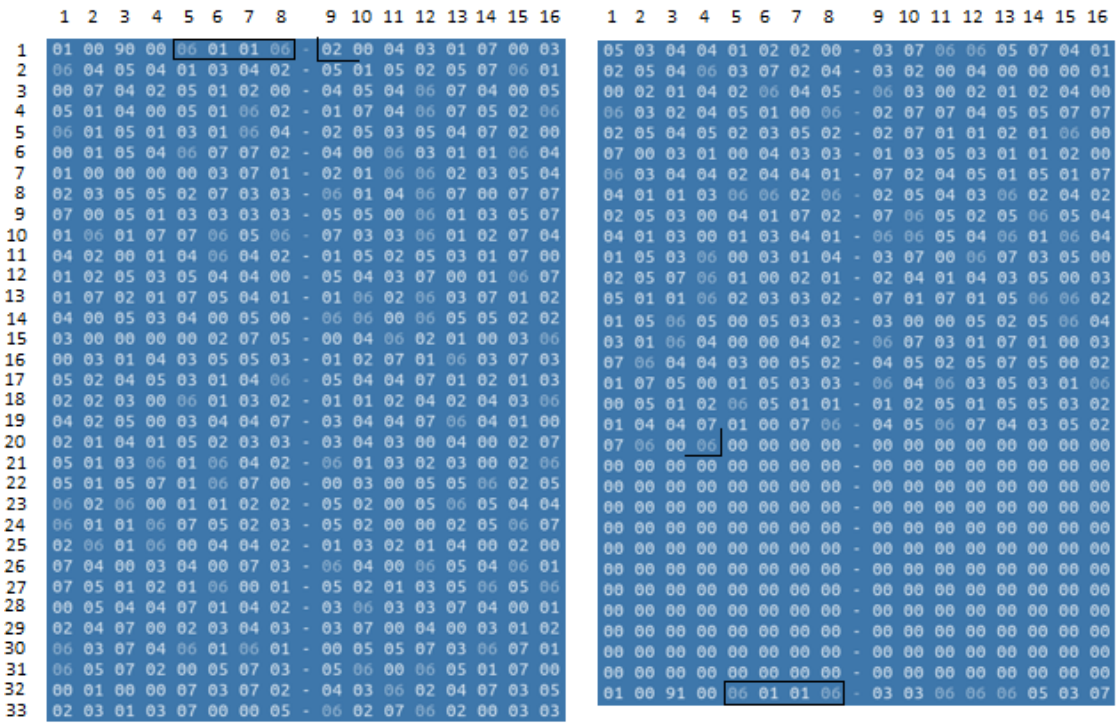


Figura 6.17: Paquete de datos a la salida del bloque “Field Sync Mux”

partir del transmisor. De igual forma en la figura 6.21 se distinguen los ocho niveles de la modulación 8-VSB, esta señal fue tomada a la salida del codificador de Trellis.

```
pruebasyncm
0355 BE17: 05 04 02 01 01 05 00 00  06 05 07 07 05 06 05 01
0355 BE27: 00 03 01 05 01 02 06 06  05 03 06 03 03 02 00 04
0355 BE37: 02 04 02 06 03 05 04 00  07 04 01 05 02 04 07 07
0355 BE47: 00 03 04 03 00 04 00 02  07 00 05 07 03 06 06 01
0355 BE57: 05 04 05 04 01 04 05 00  07 06 07 05 04 02 00 04
0355 BE67: 05 00 07 05 00 07 07 01  05 03 03 02 07 04 02 00
0355 BE77: 07 07 04 05 02 07 03 07  02 04 04 07 05 04 05 03
0355 BE87: 06 06 06 05 06 02 02 03  07 05 02 02 03 04 00 07
0355 BE97: 01 07 03 04 04 00 02 02  07 00 06 00 06 03 07 03
pruebavect
0355 BE17: 05 04 02 01 01 05 00 00  06 05 07 07 05 06 05 01
0355 BE27: 00 03 01 05 01 02 06 06  05 03 06 03 03 02 00 04
0355 BE37: 02 04 02 06 03 05 04 00  07 04 01 05 02 04 07 07
0355 BE47: 00 03 04 03 00 04 00 02  07 00 05 07 03 06 06 01
0355 BE57: 05 04 05 04 01 04 05 00  07 06 07 05 04 02 00 04
0355 BE67: 05 00 07 05 00 07 07 01  05 03 03 02 07 04 02 00
0355 BE77: 07 07 04 05 02 07 03 07  02 04 04 07 05 04 05 03
0355 BE87: 06 06 06 05 06 02 02 03  07 05 02 02 03 04 00 07
0355 BE97: 01 07 03 04 04 00 02 02  07 00 06 00 06 03 07 03
```

Figura 6.18: Comparación de datos entre los bloques “Field Sync Mux” y “Vector to Stream”

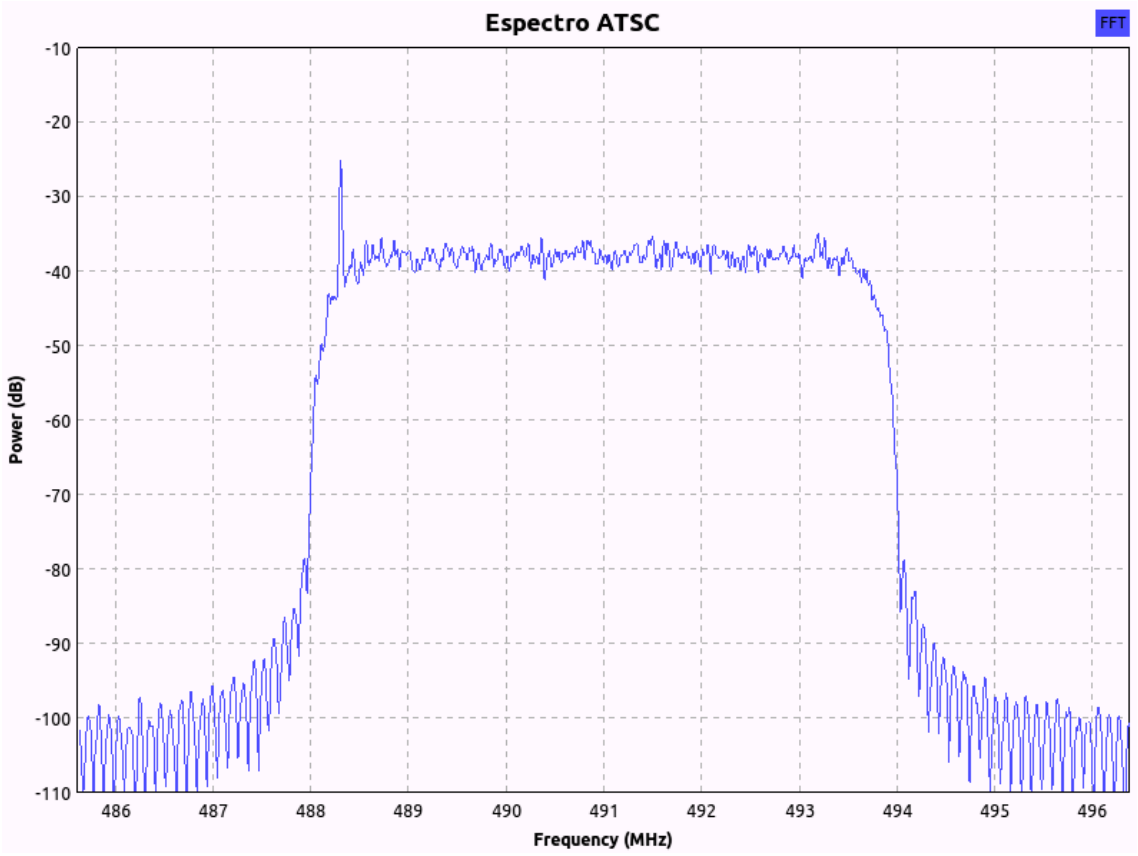


Figura 6.19: Espectro de una señal de ATSC

Figura 6.20: Constelación de una señal de ATSC obtenida a través de GNU Radio.

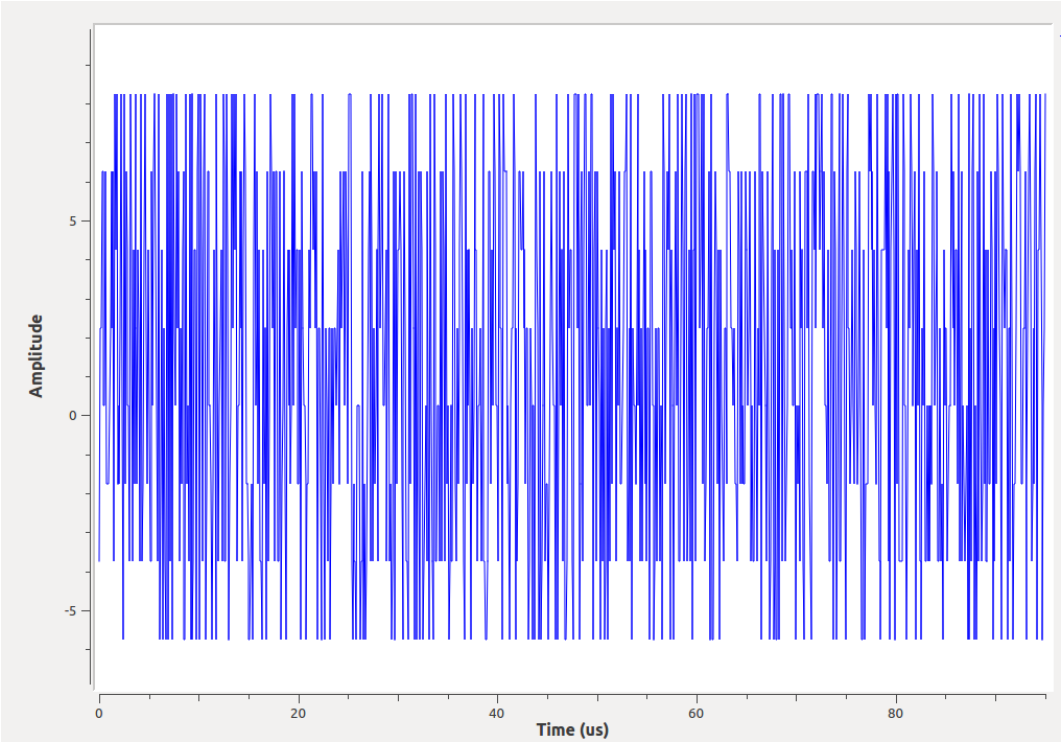


Figura 6.21: Ocho niveles de la modulación 8-VSB.

Capítulo 7

Conclusiones

La presente tesis tuvo como objetivo implementar tres tipos de transmisores para ser usados en prácticas de la materia de radiodifusión de la Facultad de Ingeniería. Para el logro del principal objetivo se realizó una investigación sobre la tecnología SDR. Posteriormente sobre los módulos de GNU Radio para proponer un diseño. Los transmisores desarrollados tienen características que contribuyen a la compresión de los conceptos claves de AM, FM y ATSC. A continuación se describen las características generales de cada transmisor.

Transmisor AM

Mediante este transmisor es posible estudiar el espectro de una señal de AM. La forma general del espectro, la potencia de la componente portadora y de las bandas laterales, el efecto que tiene la modificación del índice de modulación sobre el espectro y la relación entre el ancho de banda de una señal AM y el ancho de banda de la señal moduladora.

También es factible el estudio de la señal en el dominio temporal. Es posible observar los cambios de la señal portadora con relación a la señal moduladora y el efecto que tiene la sobremodulación sobre la señal o cualquier otro caso de índice de modulación.

El alumno podrá realizar transmisiones de ondas AM y recibirlas a través de un receptor de AM también definido por software.

El transmisor FM

Con el transmisor desarrollado es posible estudiar el espectro de una señal de frecuencia modulada. Su forma en general y la dependencia del ancho de banda con el índice de modulación de la señal de FM.

En el dominio temporal es posible observar la apariencia de una señal modulada en frecuencia.

El alumno podrá llevar a acabo emisiones en FM y recibirlas a través de un receptor de FM definido por software o a través de una aplicación de radio FM en un celular.

El transmisor de ATSC

Mediante este transmisor es posible estudiar el espectro de una señal digital de televisión con el estándar ATSC. La forma general del espectro, la posición de la portadora piloto en el espectro y el ancho de banda de una señal ATSC.

En el dominio temporal es posible observar los ocho niveles que toma la señal 8 ASK antes de generar la modulación 8-VSB. Con éste transmisor no es posible realizar emisiones de TV digital en tiempo real.

Trabajos futuros

Como trabajo futuro sería deseable aumentar un control de nivel de ruido a cada transmisor. Este indicaría la amplitud del ruido que se le añade a la señal antes de ser transmitida y con esto estudiar los efectos que el ruido tiene en el espectro de la señal y en el receptor.

En el transmisor de ATSC sería posible simular el fenómeno de multitrayecto y estudiar los efectos que éste tiene en el espectro de la señal y en el receptor.

El desarrollo más importante a futuro es la elaboración de las prácticas de laboratorio que involucren los transmisores, las cuales tiene que ser desarrolladas considerando las características de cada transmisor. Así como también lograr emisiones de TV digital en tiempo real y recibirlas en un televisor digital.

Apéndice A

Data Sheet USRP^{TM} X300 and X310 X Series

Ettus research a National Instruments Company

FEATURES

Two wideband RF daughterboard slots
o Up 120MHz bandwidth per channel
o Selection covers DC to 6 GHz
Large, customizable Kintex-7 FPGA
o USRP X300 - XC7K325T
o USRP X310 – XC7K410T
UHD architecture provides compatibility:
o GNURadio
o C++ API/Python
o Other third-party frameworks and applications
Multiple high-speed interfaces
o Dual SFP(+) ports for 1/10 Gigabit Ethernet
o PCIe x4
Flexible clocking architecture
o Configurable sample clock
o Optional GPS-disciplined OCXO

o Coherent operation with 10 MHz/1 PPS

SAMPLE APPLICATIONS

Advanced Wireless Prototyping (WiFi/Cellular)

Massive MIMO Testbeds

Passive RADAR

Signals Intelligence

USRP X300 and X310 Product Overview

The Ettus Research USRP X300 and X310 are high-performance, scalable software defined radio (SDR) platforms for designing and deploying next generation wireless communications systems. The hardware architecture combines two extended bandwidth daughterboard slots covering DC – 6 GHz with up to 120 MHz of baseband bandwidth, multiple high-speed interface options (PCIe, Dual 1/10 GigE), and a large user-programmable Kintex-7 FPGA in a convenient desktop or rack-mountable half-wide 1U form factor. In addition to providing best-in-class hardware performance, the open source software architecture of the USRP X300 and X310 provides cross-platform UHD driver support making it compatible with a large number of USRP supported development frameworks, reference architectures, and open source projects.

Spec	Typ.	Unit
Power		
DC Input	12	V
Power Consumption (2x SBX-120)	45	W
Conversion Performance and Clocks		
ADC Sample Rate (max)	200	MS/s
ADC Resolution	14	bits
DAC Sample Rate (max)	800	MS/s
DAC Resolution	16	bits
Host Sample Rate (16b) **	200	MS/s
Internal Reference Accuracy	2.5	ppm
Accuracy w/ GPSDO Option (not locked to GPS)	20	ppb

Spec	Typ.	Unit
RF Performance (with SBX-120)		
SSB/LO Suppression	-35/50	dBc
Phase Noise		
3.5 GHz	1.0	deg RMS
6 GHz	1.5	deg RMS
Power Output	>10	dBm
IIP3 (@ typ NF)	0	dBm
Typical Noise Figure	8	dB
Physical		
Dimensions (half-wide, 1U)	27.7 x 21.8 x 3.9	cm
Weight (w/ 2x SBX-120)	1.7	kg

*All specifications are subject to change without notice.
** Host sample rate dependent on selected interface and host-PC performance.

Figura A.1: Especificaciones USRP X300

Apéndice B

Código para visualizar archivos en formato hexadecimal

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#define KGRN  "\x1B[32m"
#define KWHT  "\x1B[37m"
#define KNRM  "\x1B[0m"

int main(int argc, char *argv[])
{
FILE *fp;
int count;
unsigned long int while_counter = 0;
int ret_read;
unsigned char buffer[16];
unsigned char byte;

if(argc != 3){
printf("Este programa muestra un archivo en formato hexadecimal\n");
```

```
printf("El parametro \"numero\" se indica en decimal (0-255)\n");
printf("El byte seleccionado se marca con verde\n\n");
printf("Uso del programa:\n%s archivo numero\n\n", argv[0]);
exit(0);
}

byte = atoi(argv[2]);
fp = fopen(argv[1], "rb");

printf("\n\n\t\t ");
printf("01 02 03 04 05 06 07 08 - 09 10 11 12 13 14 15 16\n\n");

do{
//memset(buffer, '\0', sizeof(buffer));
ret_read = fread(buffer, 1, 16, fp);
printf("%.10x0\t ", while_counter);
for(count = 0; count < ret_read; count++){
if(buffer[count] == byte)
printf(KGRN "%.2x ", buffer[count]);
else
printf(KWHT "%.2x ", buffer[count]);
if(count == 7)
printf("- ");
}
printf(KNRM "\n");
while_counter++;
}while(ret_read == 16);

fclose(fp);
return 0;
}
```

Bibliografía

[1] S. Mamidi, E. Blem, M. J. Schulte, J. Glossner, D. Iancu, A. Iancu, M. Moudgill, and S. Jinturkar, "Instruction set extensions for Software Defined Radio," *Microprocessors and Microsystems, Science Direct*, vol. 33, no. 4, pp. 260–272, 2009.

[2] L. S. Nagurney, "Software defined radio in the electrical and computer engineering curriculum," *Frontiers in Education Conference IEEE.*, pp. 1–6, October 18, 2009.

[3] J. L. ShantonIII, "A Software Defined Radio Transformation," *Military Communications Conference IEEE. MILCOM*, pp. 1–5, October 18, 2009.

[4] "Harris tactical comunications home," Febrero de 2016. [Online]. Available: http://rf.harris.com/capabilities/jtrs_capabilities/

[5] "Thales defense and security," Febrero de 2016. [Online]. Available: http://www.thalescomminc.com/content/anprc148jtrsenhancedmbitrjem.aspx

[6] H. R. M. Ramos, "Arquitecturas de radio cognitiva: una revisión actual," *Tecnura*, vol. 18, no. 39, pp. 181–196, 2014.

[7] B. Li, "Analysis and design of Software Defined Radio," *Internet Computing and Information Services (ICICIS), 2011 International Conference on, IEEE*, pp. 415–418, September 17, 2011.

[8] D. A. Patterson and J. L. Hennessy, *Estructura y diseño de computadores.* Editorial Reverté, 2004.

[9] J. H. Reed, *Software Radio: A Modern Approach to Radio Engineering*. USA, New Jersey: Prentice Hall Professional, 2002.

[10] J. C. G. Cano, *PCPI - Montaje de componentes informáticos*. Editex, 2010.

[11] "Ettus research a national instruments company," Marzo de 2016. [Online]. Available: https://www.ettus.com/product/details/PCIE-KIT

[12] F. A. C. Garcia, R. Schena, F. M. da Silva, M. Nogueira, E. Wolski, A. G. M. Lima, and L. R. A. X. de Menezes, "Estimation of DOA for a smart antenna using a front-end based in FPGA foreseeing a SDR architecture," *Antennas and Propagation. EuCAP. First European Conference on*, pp. 1–4, November 6, 2006.

[13] "Evolving wireless technology, crowded spectrum, and the USRP platform by John Malsbury, June, 2012," Marzo de 2016. [Online]. Available: https://www.ettus.com/

[14] H. Arslan, *Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems*. Florida, U.S.A: Springer Science and Business Media, 2007.

[15] M. Fähnle, *Software-Defined Radio with GNU Radio and USRP/2 Hardware Frontend: Setup and FM/GSM Applications, Thesis*. Ulm, Germany: Hochschule Ulm, University of Applied Sciences, Institute of Communication Technology, 2010.

[16] "Enabling technologies for SDR: Comparing FPGA and DSP performance, Berkeley Design Technology, Inc. U.S.A," Marzo de 2016. [Online]. Available: http://www.bdti.com/MyBDTI/pubs/20061115_sdr06_fpgas.pdf

[17] "The object management group (OMG)," Marzo de 2016. [Online]. Available: http://www.omg.org/gettingstarted/gettingstartedindex.htm

[18] H. Balen, M. Elenko, J. Jones, and G. Palumbo, *Distributed Object Architectures with CORBA*. Cambridge, United Kingdom: Press Syndicate of the University of Cambridge, 2000.

[19] "GNU Radio. Windows Installation," Marzo de 2016. [Online]. Available: https://gnuradio.org/redmine/projects/gnuradio/wiki/WindowsInstall

[20] "GNU Radio. The free and open software radio ecosystem," Marzo de 2016. [Online]. Available: http://gnuradio.org/redmine/projects/gnuradio/wiki

[21] S. Bassi, *Python en 8 clases: Aprendiendo a programar con Python.* Genes Digitales, 2013.

[22] E. A.Thompson, N. Clem, I. Renninger, and T. Loos, "Software-defined GPS receiver on USRP-platform," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1352–1360, 2012.

[23] N. C. Karmakar, *Handbook of Smart Antennas for RFID (Radio-frequency identification) Systems.* New Jersey, U.S.A: John Wiley and Sons, 2011.

[24] "USRP hardware driver and USRP manual," Marzo de 2016. [Online]. Available: http://files.ettus.com/manual/page_usrp_x3x0.html

[25] P. Balister and J. H. Reed, "USRP hardware and software description," *Bradley Dept. of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University*, no. 9, pp. 1–15, 2006.

[26] "USRP X300," Marzo de 2016. [Online]. Available: https://www.ettus.com/product/details/X300-KIT

[27] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing.* San Diego California, U.S.A: California Technical Publishing, 1997.

[28] E. B. Albertí, *Procesado Digital de Señales - I: Fundamentos Para Comunicaciones y Control.* Barcelona, España: Ediciones UPC, 2006.

[29] M. Martínez, L. Gómez, A. J. Serrano, J. Vila, and J. Gómez, *Filtros Digitales.* Universidad de Valencia, Escuela Técnica Superior de Ingeniería, Departamento de Ingeniería Electrónica, Curso 2009-2010.

[30] J. E. G. Barajas, *Series y Transformada de Fourier para Señales Continuas y Discretas en el Tiempo.* Bogotá D.C., Colombia: Omnia Publisher SL, 2015.

[31] W. Tomasi, *Sistemas de Comunicaciones Electrónicas.* Phoenix, Arizona: Pearson Educación, 2003.

[32] G. James and D. Burley, *Matemáticas avanzadas para ingeniería.* Coventry, Reino Unido: Addsison Wesley Longman, 2002.

[33] E. S. Olivas, *Tratamiento digital de señales.* México: Pearson Educación, 2003.

[34] S. Dogra and N. Sharma, "Comparison of different techniques to design of filter," *International Journal of Computer Applications*, vol. 97, no. 1, pp. 25–29, 2014.

[35] J. P. Cáceres, *Análisis Espectral 1: Transformada Corta de Fourier y Ventanas.* Stanford, Estados Unidos: Center for Computer Research in Music and Acoustics, Stanford University, 2007.

[36] *DSP and Digital Filters. Windows Filter Design.* Imperial College London. Department of Electrical and Electronic Enginnering, 2015.

[37] S. Katz and J. Flynn, "I and Q Components in Communications Signals," Marzo de 2016. [Online]. Available: http://www.csun.edu/~skatz/katzpage/sdr_project/sdr/I_andQ.pdf

[38] S. Ireland and P. Harman, "Watch your I and Q signals," *RadCom*, pp. 2–5, Enero, 2007.

[39] M. Renfors, *Sampling and Multirate Techniques for Complex and Bandpass Signals.* Finland: Tampere University of Technology, Department of Communications Engineering, 2010.

[40] M. F. Zanuy, *Sistemas de comunicaciones.* Barcelona, España: Marcombo, 2001.

[41] O. Szymanczyk, *Historia de las telecomunicaciones mundiales.* Buenos Aires, Argentina: Editorial Dunken, 2013.

[42] *Signals and Modulation (Draft ELEC 350).* Victoria, Canadá: University of Victoria. Electrical and Computer Enginnering, 2015.

[43] R. E. Tohmas and A. J. Rosa, *Circuitos y señales.* New York, U.S.A: Editorial Reverté, 2002.

[44] "GNU Radio Companion. Block Documentation, University of Victoria. Electrical and Computer Engineering," Marzo de 2016. [Online]. Available: http://www.ece.uvic.ca/~elec350/grc_doc/ar01s02s01.html

[45] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Señales y sistemas.* Cambridge, Massachusetts: Prentice-Hall, 1998.

[46] IFT, *Disposición Técnica IFT-002- 2014: Especificaciones y requerimientos mínimos para la instalación y operación de las estaciones de radiodifusión sonora en frecuencia modulada.* Diario Oficial de la Federación, 2014.

[47] J. M. Millán, *Configuración de infraestructuras de sistemas de telecomunicaciones.* Madrid, España: Ediciones Paraninfo, 2014.

[48] A. Yadav, *Analog Communication System.* Nueva Delhi, India: University Science Press, 2008.

[49] I. Wu, S. Hirakawa, U. H. Reimers, and J. Whitaker, "Overview of digital television development worldwide," *Proceedings of the IEEE*, vol. 94, no. 1, pp. 8–21, Enero, 2006.

[50] "Diario oficial de la federación. Acuerdo por el que se adopta el estándar tecnológico de televisión digital terrestre," Marzo de 2016. [Online]. Available: http://www.dof.gob.mx/nota_detalle.php?codigo=678631&fecha=02/07/2004

[51] "Apagón analógico, la tv ya no será la misma, forbes méxico," Abril de 2016. [Online]. Available: http://www.forbes.com.mx/apagon-analogico-la-tv-ya-sera-la-misma/

[52] ITU-R, *A guide to digital terrestrial television broadcasting in the VHF/UHF bands.* Radiocommunication Study Groups. Document 11-3/3-E. Task Group 11/3, 1996.

[53] ATSC, *A/53: ATSC Digital Television Standard. Part 2 – RF/Transmission System Characteristics.* Washington, D.C.: ATSC, 2007.

[54] W. Fischer, *Tecnologías para la Radiodifusión Digital de Vídeo y Audio. Una Guía Practica para Ingenieros.* Múnich, Alemania: Springer, 2007.

[55] D. Sparano, *What Exactly is 8-VSB.* N.Y, U.S.A: Master of Science degree from Rensselaer Polytechinc Institute, 1997.

[56] C. P. Vega and J. M. Zamanillo, *Fundamentos de Televisión Analógica y Digital.* Cantabria, España: Servicio de Publicaciones de la Universidad de Cantabria, 2003.

[57] "GNU Radio 3.6.5 Documentation, gnuradio.atsc: Signal Processing Blocks," Marzo de 2016. [Online]. Available: http://gnuradio.org/doc/sphinx-3.6/atsc/blks.html

[58] F. D. Castro, M. D. Castro, and D. S. Arantes, "8-VSB channel coding analysis for DTV broadcast," *: IEEE Consumer Electronics, 2000. ICCE. Digest of Technical Papers. International Conference on*, pp. 144–145, June 15, 2000.

[59] "GNU Radio Manual and C++ API Reference 3.7.9.1," Marzo de 2016. [Online]. Available: https://gnuradio.org/doc/doxygen/index.html

[60] "PHP5 Database Access Demo, FIR Filter," Marzo de 2016. [Online]. Available: http://swigerco.com/gnuradio/fir_filter.html

[61] "MPEG-2 Transport Stream Analysis and Recording. TSReader," Marzo de 2016. [Online]. Available: http://www.coolstf.com/tsreader/

[62] W. Fischer, *Digital Television: A Practical Guide for Engineers.* Berlín, Alemania: Springer, 2013.

[63] "GNU Radio: atsc_types.h," Marzo de 2016. [Online]. Available: http://gnuradio.org/doc/doxygen-3.2/atsc__types_8h-source.html

[64] "Atsc Data Types," Marzo de 2016. [Online]. Available: http://swigerco.com/gnuradio/atsc_data_types.html

DECLARACIÓN

La información presentada en este trabajo se obtuvo de diversas fuentes que se consideran fidedignas y se consignan puntualmente en las referencias. El uso dado a la información es de naturaleza estrictamente de investigación académica y de divulgación, sin fines de lucro o de otra índole. Se ha hecho también el mayor esfuerzo por acreditar debidamente datos, opiniones y contenidos presentados, por lo que cualquier error u omisión en ello, es del todo involuntario.

México, D.F., 30 de Marzo de 2016

Mares Rodríguez Viridiana