

PROYECTO FINAL

TRANSMISIÓN INALAMBRICA

Autores

2185106 – Jean Pablo Ruiz Torres

2185571 – Andrés Camilo Fuquen Gil

2185581 – Andrés Camilo Rincón Santana

Grupo de clase:

H1

Subgrupo de clase:

Banco 8.

EL RETO A RESOLVER:

Al finalizar el trabajo, el estudiante estará familiarizado con los conceptos básicos de transmisión inalámbrica y su incidencia sobre el error binario.

El estudiante deberá evaluar el modelo suministrado donde se permita evidenciar el problema del ruido sobre las modulaciones digitales.

Este trabajo está basado en los entregables del proyecto denominado: "[Diseño e implementación del prototipo de un sistema de comunicaciones satelital](#)". desarrollado por los ingenieros Andrés Felipe Pérez Rueda y Carlos Andrés Estupiñán Parra, dirigidos por el doctor Homero Ortega Boada

EL OBJETIVO GENERAL ES:

Desarrollar habilidades en el manejo de GNURadio y resaltar la importancia de la transmisión inalámbrica en el laboratorio de comunicaciones.

ENLACES DE INTERÉS

¿Qué es Gnuradio y que podemos hacer con este programa?

[Enlace a módulos de proyecto de grado](#)

LO QUE NO DEBEN HACER

Plagio: Cualquier intento de plagio de la tesis en mención será causal de nulidad de su proyecto: Las imágenes deben ser de su autoría (traten de acomodar de una manera distinta a la indicada), los párrafos deben ser de su autoría.

Entregar imágenes que no correspondan a las medidas en tiempo y frecuencia de una señal modulada en GMSK (consultar sobre la modulación)

TRABAJO

1. Implementar los esquemas de transmisión y recepción del [Flujogramas GNURADIO](#) donde pueda describir las funciones de cada uno de los módulos que integran el flujograma y su función dentro del sistema, describa el funcionamiento paso a paso de cada uno de los sistemas (15 puntos)
2. Realizar la validación de los sistemas a través de un enlace cableado (cable coaxial del laboratorio) y un enlace inalámbrico (usar frecuencias de operación acordes a los instrumentos que tenemos en el laboratorio), debe establecer las medidas en tiempo (osciloscopio tiempos de bit) y frecuencia (señal modulada en función de la frecuencia de muestreo en el analizador de espectro) para cada uno de los dos casos (10 puntos)
3. Realizar el proceso de sincronización del reloj del sistema, evidenciar el funcionamiento del módulo [Polyphase Clock Sync](#) dentro del sistema de recepción (5 puntos)
4. Evaluar los módulos [Linear Equalizer](#) y [Adaptive Algorithm](#) con respecto al nivel de distorsión de la señal en la recibida (5 puntos)
5. Evaluar el desplazamiento de fase y frecuencia en un enlace inalámbrico usando el módulo [Costas Loop](#) (5 puntos)
6. Implementar un sistema de modulación M aria (mínimo 8 niveles) que permita la transmisión y recepción de un mensaje de texto como lo realizado por los autores del proyecto. (15 puntos)

1. Descripción de las funciones dentro de los flujogramas a implementar:

1.1. Descripción de las funciones de los bloques Transmisión (TX):

Para la parte de la transmisión, existen dos flujogramas uno para cada modulación (QPSK y GMSK).

QPSK: El flujograma para esta modulación es el siguiente:

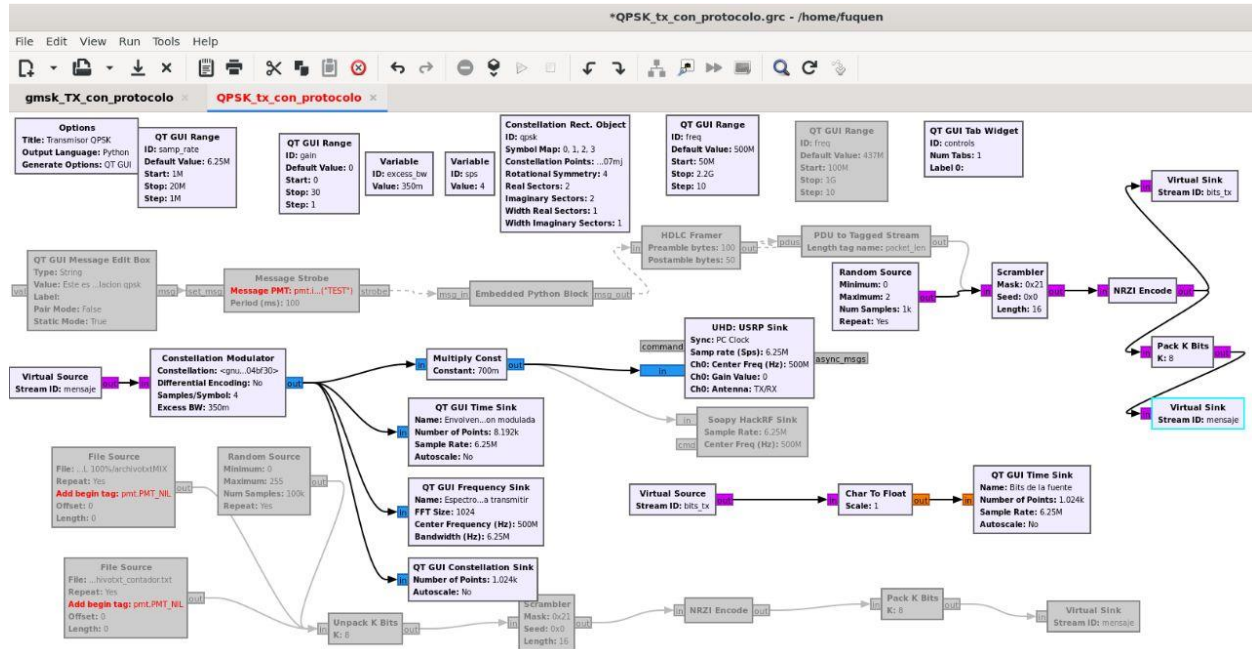


Ilustración 1. Flujograma QPSK TX.

La transmisión de modulación por desplazamiento de fase (QPSK) consiste en hacer variar la fase de la portadora entre un número determinado de valores discretos los cuales serán los símbolos que se verán en el diagrama de constelación, donde la amplitud de los símbolos no se modificará, por lo que se trata de una modulación angular y la señal modulada será digital. A un mayor número de símbolos, se requieren un mayor número de símbolos, como se observa en la siguiente expresión:

$$M = 2^k$$

Donde M es el número de símbolos y k el número de bits. Ya con las anteriores aclaraciones dadas, se procede a observar el funcionamiento del flujograma del transmisor.

Transmisor QPSK: Para esta modulación, se tiene un bloque Random Source, donde su función es generar muestras de números aleatorios, los cuales se van a regir por los parámetros en este caso mínimo 0 máximo 3 y con un numero de muestras de 1k, de ahí ingresan al bloque Scrambler que codifica o empaqueta la información y se envía a NRZI Encode que codifica sin retorno a cero invertido en binario y se usa generalmente para representar voltajes, luego se limitan los bits que queremos con el bloque Pack K. Bits para después comunicar al bloque Virtual Sink con Virtual Source donde ingresa a un bloque Constellation Modulator, donde la entrada es un flujo de bytes y la salida es la señal compleja modulada en banda base ingresa esta señal a Multiply Const que multiplica la información de la señal por una constante en este caso 700m y se trasmite la información mediante el UHD: USRP Sink. Ahora, del bloque Virtual Source también sale un bloque QT GUI Time Sink que simula y se pueden visualizar en la pantalla la señal de la información en el dominio del tiempo y

también el bloque QT GUI Frequency Sink que al igual que el anterior nos muestra las simulaciones, pero en el dominio de la frecuencia, además de los anteriores bloques, hay otro denominado QT GUI Constellation Sink este bloque se utiliza para mostrar gráficamente la constelación con sus símbolos. Se tiene otra señal simulada en el tiempo, pero con el bloque Chart To Fload la señal pasa de caracteres a datos flotantes. Hay bloques que se utilizan como variables de algunos de los anteriores y también bloques de variables con rango especificadas en los bloques QT GUI Range, útiles para el sistema y también tenemos un bloque Constellation Rect. Object en este bloque se pueden definir el número de símbolos que se van a poder apreciar en el diagrama de constelación.

Transmisión GMSK: El flujograma para esta modulación es el siguiente:

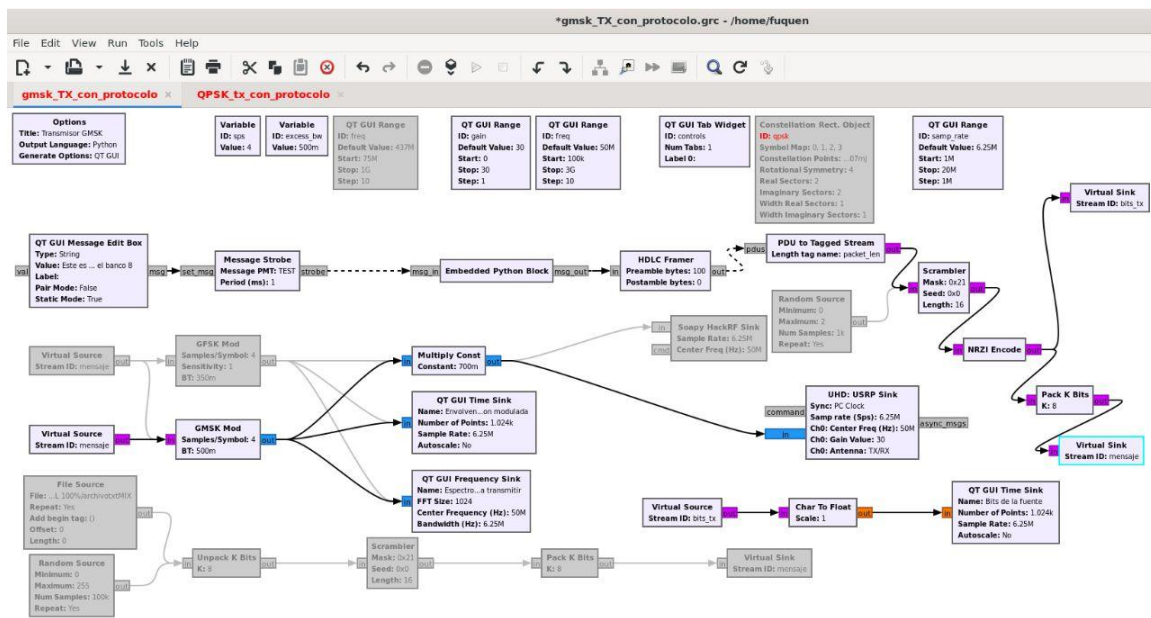


Ilustración 2. Flujograma GMSK TX.

Este flujograma de transmisión TX para GMSK se encarga de transmitir la Modulación por desplazamiento mínimo gaussiano, esta modulación atraviesa un filtro pasa bajas gaussiano antes de ser implementado en el flujograma modulador, lo que suaviza las transiciones de fase de la señal durante la transmisión y así reduce el ancho de banda. La función del bloque QT GUI Message Edit Box en el flujograma es administrar datos como mensajes que se convierten en cadenas, si los datos no son correctos el bloque genera un mensaje de advertencia de registro y no genera los datos, luego ingresa al bloque Message Strobe que envía esos mensajes en un intervalo de tiempo definido en este caso con un periodo de 1 [ms], seguidamente toma la entrada y la multiplica por una constante, el bloque de Python a partir de esa señal con el bloque HDLC Frammer toma binarios y genera bits en decimal y rellena de bits agregados que generan un cuadro completo a la vez, el bloque PDU to Tagged Stream recibe esos bits y convierte la unidad de datos de protocolo en elementos etiquetados, con el Scrambler se codifica la información que se envía a NRZI Encode la codificación sin retorno a cero

invertido está en binario y se usa generalmente para representar voltajes, luego se limitan los bits que queremos con el bloque Pack K. Bits luego comunicaremos al bloque Virtual Sink con Virtual Source, seguidamente se modula GMSK de donde saldrán los bloques Multiply Const que multiplica la información de la señal por una constante en este caso 700m y se transmite la información mediante el UHD: USRP Sink, del bloque Virtual Source también sale un bloque QT GUI Time Sink que simula y se pueden visualizar en la pantalla varias señales en el dominio del tiempo y también el bloque QT GUI Frequency Sink que al igual que el anterior nos muestra las simulaciones pero en el dominio de la frecuencia. Se tiene otra señal simulada en el tiempo, pero con el bloque Chart To Fload la señal pasa de caracteres a datos flotantes. Hay bloques que se utilizan como variables de algunos de los anteriores y también bloques de variables con rango como QT GUI Range útiles para el sistema.

1.2. Descripción de las funciones de los bloques Recepción (RX):

Para la parte de Recepción, existen dos flujogramas uno para cada modulación (QPSK y GMSK).

QPSK: El flujograma para esta modulación es el siguiente:

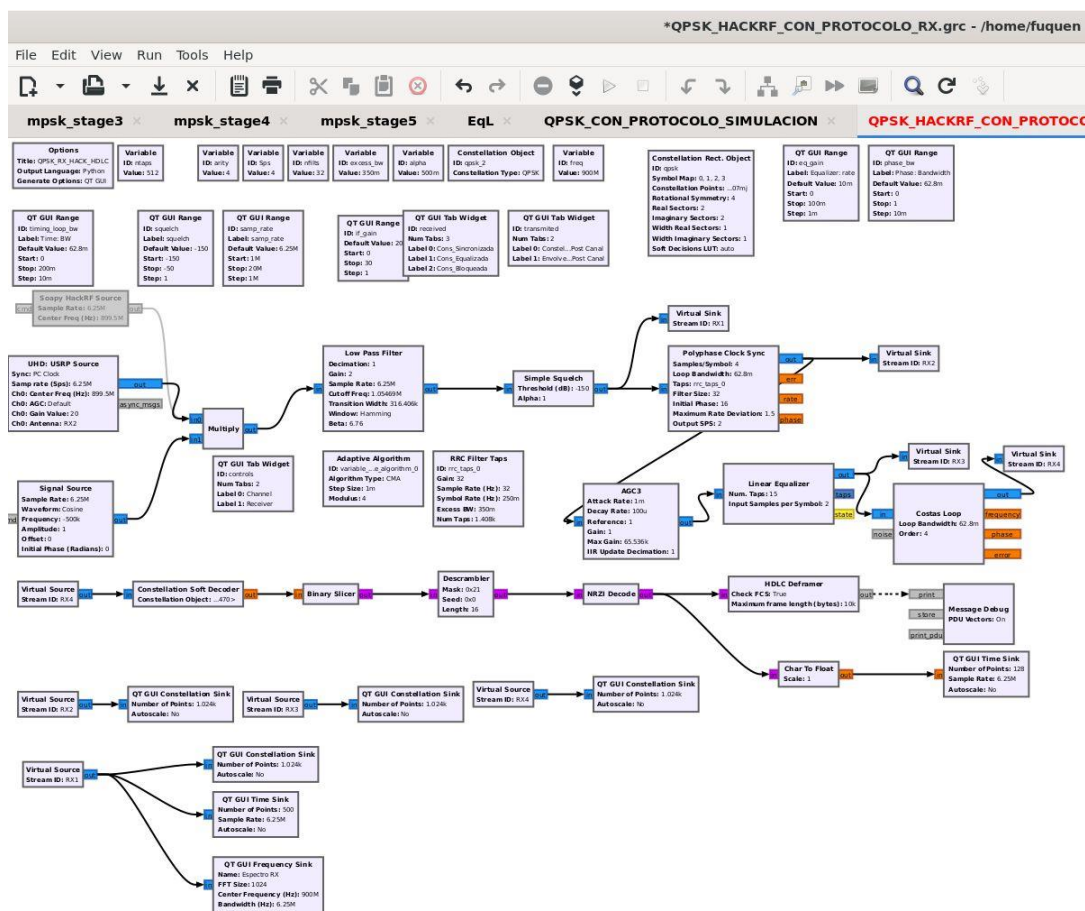


Ilustración 3. Flujograma QPSK RX.

La función del bloque UHD: USRP Sink es recibir la información y se multiplica con una señal generada con el bloque Signal Source y luego esta señal sale para el bloque Low Pass Filter donde pasa por un filtro pasa bajas que envía la señal a Simple Squelch de silenciamiento en la potencia de señal promedio de ahí se comunica una señal RX1 con virtual Sink y además la señal sale para otro bloque Polyphase Clock Sync este bloque realiza la sincronización de tiempo para las señales y se ahí se comunica otra señal RX2 y también con un bloque AGC3 que es un bloque de control automático de ganancia de alto rendimiento, esta señal ingresa a Linear Equalizer que realiza la ecualización lineal en un flujo de muestras complejas, la salida de ese bloque va para un Virtual Sink y esta señal es RX3 y también para un bloque Costas Loop recupera la portadora de bucle de Costas excelente para sincronizar con QPSK de ese bloque sale la señal para otro Virtual Sink y esta señal sería RX4, esta señal ingresa ahora a un bloque Virtual Source que se comunica con la señal RX4 que ingresa a un bloque Constellation Soft Decoder que descodifica los puntos de una constelación de un espacio complejo a bits suaves ya así pasa a Binary Slicer que corta un valor flotante que produce una salida de 1 bit para pasar a un bloque Descrambler este bloque descifra un flujo de entrada y produce el mismo formato en su salida y la envía al bloque NRZI Decode la decodificación sin retorno a cero invertido está en binario y se usa generalmente para representar voltajes, la señal luego ingresa a un bloque HDLC Deframer toma bits desempaquetados y genera blobs binarios y genera blobs binarios que van para Message Debug para depuración de mensajes se utiliza para capturar e imprimir o almacenar mensajes a medida que se reciben, además se obtiene la señal simulada en el tiempo, pero con el bloque Chart To Fload la señal pasa de caracteres a datos flotantes; ahora la señal RX2 es llamada por Virtual Source y se observa la constelación mediante el bloque QT GUI Constellation Sink de igual manera para RX3 y RX4 y finalmente obtenemos la simulación en el dominio del tiempo y de la frecuencia para RX1 y la constelación. En este flujograma tenemos también variables y variables con rango y Widgets para que se vea mejor la información y un bloque Constellation Rect. Object en este bloque se pueden definir el número de símbolos que se van a poder apreciar en el diagrama de constelación.

GMSK: El flujograma para esta modulación es el siguiente:

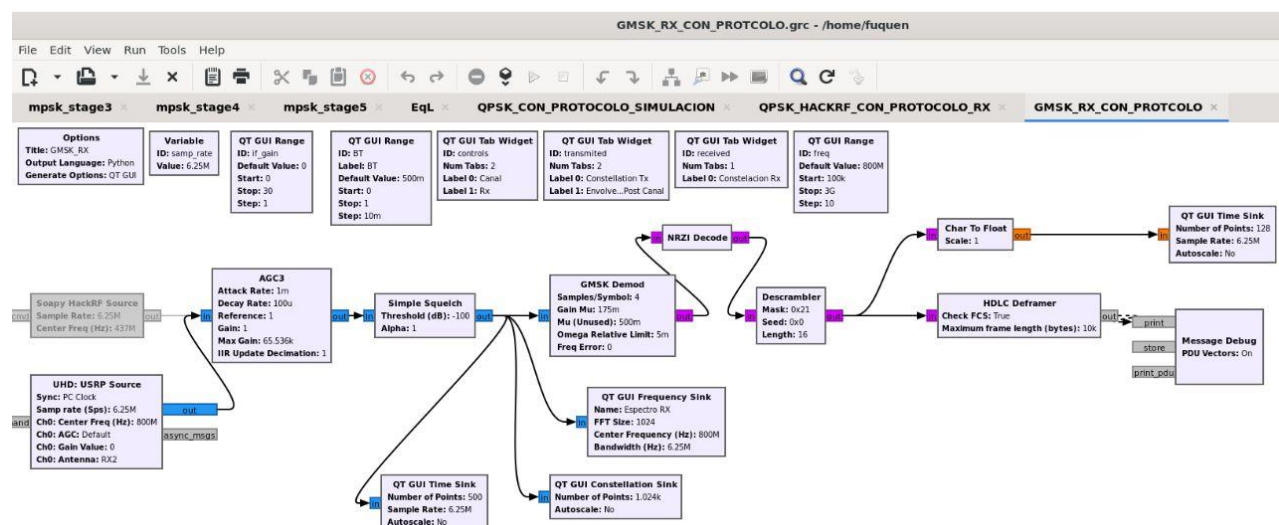


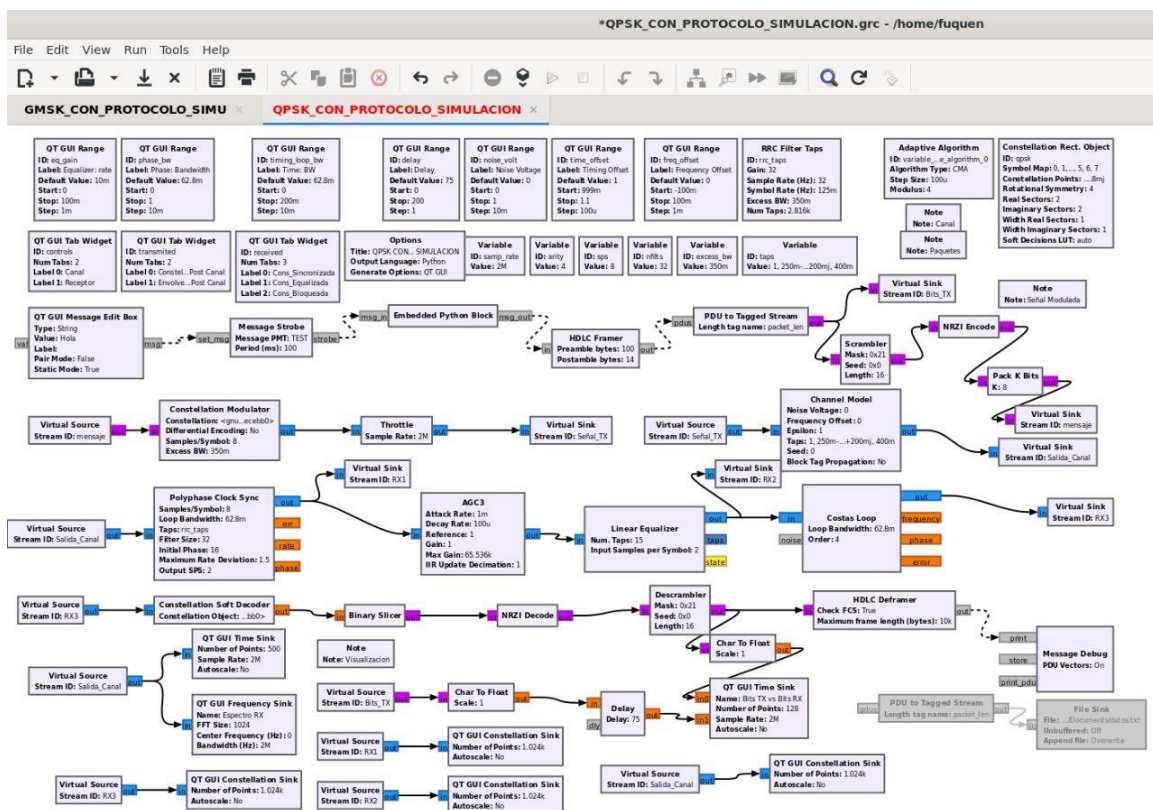
Ilustración 4. Flujograma GMSK RX.

Este flujograma consiste en la recepción GMSK, anteriormente se explicaron los flujogramas de transmisión de GMSK y QPSK ahora estos nos sirven para recibir las señales transmitidas y poder comunicar el sistema de un lugar a otro.

La función del bloque UHD: USRP Sink es transmitir la información al bloque AGC3 que es un bloque de control automático de ganancia de alto rendimiento, luego de esto pasa a un bloque Simple Squelch para silenciamiento en la potencia de la señal promedio y el umbral en dB de ahí se observa la gráfica de la señal en el dominio del tiempo con Time Sink, también en el dominio de la frecuencia con Frequency Sink y la simulación de la constelación, además sale para un bloque GMSK Demod que recibe señal compleja modulada en banda base y entrega a la salida un flujo de bits empaquetados 1 bit por byte, luego ingresa a NZRI Decode la decodificación sin retorno a cero invertido está en binario y se usa generalmente para representar voltajes, ingresa al bloque Descrambler este bloque descifra un flujo de entrada y produce el mismo formato en su salida y se obtiene la señal simulada en el tiempo, pero con el bloque Chart To Fload la señal pasa de caracteres a datos flotantes, además la señal pasa por un bloque HDLC Deframer toma bits desempaquetados y genera blobs binarios que van para Menssage Debug para depuración de mensajes se utiliza para capturar e imprimir o almacenar mensajes a medida que se reciben. También hay bloque de variables sencillas y con rango que se puede variar y unos bloques de Widgets para que se observe mejor la información.

1.3. Descripción de las funciones de los bloques de simulación:

Para la parte de simulación, existen dos flujogramas uno para cada modulación (QPSK y GMSK). **QPSK:** El flujograma para esta simulación es el siguiente:



En el flujograma encontramos las siguientes conexiones, primero encontramos un bloque QT GUI Message Edit Box administra datos como mensajes que convierte en cadenas, si los datos no son correctos el bloque genera un mensaje de advertencia de registro y no genera los datos, luego ingresa al bloque Message Strobe que envía esos mensajes en un intervalo de tiempo definido en este caso con un periodo de 1 [ms], seguidamente toma la entrada y la multiplica por una constante el bloque de Python a partir de esa señal con el bloque HDLC Framer toma binarios y genera bits en decimal y rellena de bits agregados que generan un cuadro completo a la vez, el bloque PDU to Tagged Stream recibe esos bits y convierte la unidad de datos de protocolo en elementos etiquetados, sale para un Virtual Sink que es Bites TX y además sale al Scrambler se codifica la información que se envía a NRZI Encode la codificación sin retorno a cero invertido está en binario y se usa generalmente para representar voltajes, luego se limitan los bits que queremos con el bloque Pack K. Bits luego comunicaremos al bloque Virtual Sink (mensaje), seguidamente se conecta con Virtual Source a donde ingresa a un bloque Constellation Modulator la entrada es un flujo de bytes y la salida es la señal compleja modulada en banda base que sale a Throttle este bloque acelera el flujo de muestras de modo que la tasa promedio no exceda la tasa específica en muestras por segundo y de ahí sale la Señal_TX con el Virtual Sink, luego se conecta esta señal a con el Virtual Source a Channel Model que sirve para simular se usa para ayudar a evaluar y probar varias señales, sale esa señal por un Virtual Sink salida del canal a Polyphase Clock Sync este bloque realiza la sincronización de tiempo para las señales y se ahí se comunica otra señal RX1 y también con un bloque AGC3 que es un bloque de control automático de ganancia de alto rendimiento, esta señal ingresa a Linear Equalizer que realiza la ecualización lineal en un flujo de muestras complejas, la salida de ese bloque va para un Virtual Sink y esta señal es RX2 y también para un bloque Costas Loop recupera la portadora de bucle de Costas excelente para sincronizar con QPSK de ese bloque sale la señal para otro Virtual Sink y esta señal seria RX3 sale par Virtual Source que se comunica con la señal RX3 que ingresa a un bloque Constellation Soft Decoder que descodifica los puntos de una constelación de un espacio complejo a bits suaves ya así pasa a Binary Slicer que corta un valor flotante que produce una salida de 1 bit para pasar a un bloque Descrambler este bloque descifra un flujo de entrada y produce el mismo formato en su salida y la envía al bloque NRZI Decode la decodificación sin retorno a cero invertido está en binario y se usa generalmente para representar voltajes, la señal luego ingresa a un bloque HDLC Deframer toma bits desempaquetados y genera blobs binarios y genera blobs binarios que van para Menssage Debug para depuración de mensajes se utiliza para capturar e imprimir o almacenar mensajes a medida que se reciben; la señal Bias TX con Virtual Source que ingresa a Char To Float se encarga de convertir la información para que pase de ser un flujo de caracteres a datos flotantes y

GMSK: El flujograma para esta simulación es el siguiente:



La función del bloque QT GUI Message Edit Box en el flujograma es administrar datos como mensajes que convierte en cadenas, si los datos no son correctos el bloque genera un mensaje de advertencia de registro y no genera los datos, luego ingresa al bloque Message Strobe que envía esos mensajes en un intervalo de tiempo definido en este caso con un periodo de 1 [ms], seguidamente toma la entrada y la multiplica por una constante el bloque de Python a partir de esa señal con el bloque HDLC Framer toma binarios y genera bits en decimal y rellena de bits agregados que generan un cuadro completo a la vez, el bloque PDU to Tagged Stream recibe esos bits y convierte la unidad de datos de protocolo en elementos etiquetados, sale para un Virtual Sink que es Bites TX y además sale al Scrambler se

codifica la información que se envía a NRZI Encode la codificación sin retorno a cero invertido está en binario y se usa generalmente para representar voltajes, luego se limitan los bits que queremos con el bloque Pack K. Bits luego comunicaremos al bloque Virtual Sink (mensaje) con Virtual Source (mensaje) seguidamente se modula GMSK de donde saldrán los bloques Multiply Const que multiplica la información de la señal por una constante en este caso 1, esta señal ingresa al bloque Throttle este bloque acelera el flujo de muestras de modo que la tasa promedio no exceda la tasa específica en muestras por segundo y de ahí sale la señal TX con el Virtual Sink, luego se llama mediante el bloque Virtual Source a la señal TX e ingresa a Channel Model que sirve para simular se usa para ayudar a evaluar y probar varias señales, sale esa señal por un Virtual Sink salida del canal esta señal se conecta con el bloque Virtual Source a AGC3 que es un control automático de ganancia que pasa a GMSK Demod este bloque recibe señal compleja modulada en banda base y entrega a la salida un flujo de bits empaquetados 1 bit por byte, luego ingresa a NRZI Decode la decodificación sin retorno a cero invertido está en binario y se usa generalmente para representar voltajes, esta señal pasa a Descrambler este bloque descifra un flujo de entrada y produce el mismo formato en su salida he ira a HDLC Deframer toma bits desempaquetados y genera blobs binarios que van para Menssage Debug para depuración de mensajes se utiliza para capturar e imprimir o almacenar mensajes a medida que se reciben; después se conecta a la señal Bias TX con Virtual Source que ingresa a Char To Float se encarga de convertir la información para que pase de ser un flujo de caracteres a datos flotantes y pasa a un Delay este bloque retrasa la entrada por un numero de muestras y visualizamos esa señal en el dominio del tiempo y también la que señal que viene de Descrambler pasando por un Char To Float, finalmente encontramos los bloque de simulación en el dominio del tiempo y la frecuencia para la señal Salida_Canal y la constelación de las señales Salida_Canal y Señal_TX. También encontramos bloque de variables y variables con rango y bloques de notas para dar informaciones.

1.4. Descripción de los Bloque utilizados en los flujogramas:

- **Options:** Este bloque cumple con la función de asignarle la información al flujograma, con la cual se pueden identificar el título del flujograma o el autor de este, el ID y el lenguaje con el que va a operar.
- **QT GUI Range:** Por medio de este bloque se puede establecer una variable y un rango de operación para esta, donde se pueden asignar los valores inicial y final de dicho rango, y el paso con el que puede variar la variable establecida. A la hora de simular un flujograma, en la pantalla de simulación aparecerá un slide con el que se pueden asignar distintos valores, dentro del rango establecido, a la variable. Para el caso de este flujograma las variables

asignadas a este bloque son la frecuencia de muestreo (`samp_rate`), una ganancia (`gain`) y una frecuencia (`freq`).

- **Variable:** Este bloque es muy similar al QT GUI Range, con la diferencia que la variable que sea asignada a este bloque tendrá un valor fijo. Las variables asignadas para la modulación QPSK fueron muestras/símbolo (`Sps`) y `excess_bw`. Es muy común que a este bloque este asignada la variable de la frecuencia de muestreo.
- **Embedded Python block:** Este bloque permite crear un nuevo bloque (personalizado) en Python sin necesidad de crear e instalar un módulo, cuando se inserta este bloque el código precargado simplemente toma el flujo de entrada y lo multiplica por una constante. Los parámetros para este bloque son simplemente los del bloque de Python incrustado.
- **HDLC Framer:** Este bloque toma blobs binarios de PMT y genera cuadros HDLC como bits desempaquetados y relleno de bits agregados se genera un cuadro completo a la vez; si no hay suficiente espacio en el búfer de salida para que quepa un marco, se coloca en una cola. Como resultado, es posible que los diagramas de flujo que solo se ejecutan para un número finito de muestras no reciban todos los marcos en la cola, debido a la granularidad del planificador.
- **Virtual Sink:** Este bloque tiene la función de comunicarse con el bloque Virtual Source, se le asigna una variable la cual será la misma que se le asigne al bloque Virtual Source. Se puede ver como la asignación de un nombre para un cable, donde un extremo pertenece al Virtual Sink o inicio y el otro extremo al Virtual Source (final).
- **Scrambler:** Este bloque tiene la función de codificar un flujo de información. El parámetro Mask es un polinomio para LFSR; Seed es el registro de desplazamiento inicial; Length es la longitud del registro de desplazamiento.
- **Pack K Bits:** Este bloque se encarga de empaquetar un número determinado de bits, donde el parámetro k, determina el número de elementos (Bytes) que se van a combinar para dar como resultado un solo valor.
- **Virtual Source:** Este bloque recibe la información del bloque Virtual Sink, continuando así con el flujograma por medio de una variable en común entre los dos bloques.
- **GMSK Mod:** Bloque que se encarga de realizar una modulación GMSK, donde sus parámetros principales son las muestras/símbolo el ancho de banda.
- **Throttle:** Este bloque acelera el flujo de muestras de modo que la tasa promedio no exceda la tasa específica en muestras por segundo, los bloques aceleradores solo deben usarse si no hay bloques limitadores de velocidad generalmente es hardware; no está destinado ni es

eficaz para controlar con precisión la tasa de muestras, se conecta normalmente directo a la salida de bloques que no son por hardware.

- **Channel Model:** Bloque que sirve para simular un modelo de canal básico que se puede usar para ayudar a evaluar, diseñar y probar varias señales, formas de onda y algoritmos, además permite establecer el voltaje de una fuente de ruido AWGN, un desplazamiento de frecuencia, un desplazamiento de tiempo de muestra. Los parámetros son Noise voltage es el nivel de ruido AWGN como voltaje que se hallara para una señal de ruido deseada; Frequency offset es el desplazamiento en frecuencia normalizado; Epsilon es el desplazamiento en tiempo de muestras para las distintas velocidades; Taps para emular un perfil de retardo multitrayecto está establecido en 1+1j; Seed genera números aleatorios para la fuente de ruido; Block tag propagation si esta activado las etiquetas no podrán propagarse a través de este bloque.
- **AGC3:** Este bloque es una forma de control automático de ganancia de alto rendimiento, este utiliza un cálculo lineal para una adquisición muy rápida. Los parámetros son Attack rate para la tasa de actualización del bucle cuando está en modo de ataque; Decay rate se utiliza para la tasa de actualización del bucle cuando está en modo de caída; Reference es un valor que se toma de referencia para ajustar la potencia de la señal; Gain es el valor inicial de ganancia; Max gain es el valor máximo de ganancia; IIR update decimation es para ver cuando ha avanzado el número de muestras antes de hacer el nuevo cálculo para actualizar la ganancia IIR.
- **GMSK Demod:** Este bloque recibe señal compleja modulada en banda base y entrega a la salida un flujo de bits empaquetados 1 bit por byte, para la modulación por desplazamiento mínimo gaussiano. Los parámetros son Samples/symbol son las muestras por baudios; Gain Mu que controla la tasa de ajuste mu; Mu es retraso fraccional; Omega Relative Limit este establece la variación máxima en omega; Freq Error es para el error de tasa de bits como una fracción.
- **HDLC Deframer:** Es un bloque que toma bits desempaquetados y genera blobs binarios de PDU (unidades de datos de protocolo). Los parámetros son longitud mínima que es para el tamaño mínimo de trama en bytes y longitud máxima que es el tamaño máximo de trama en bytes.
- **Message:** Este bloque 'envía mensajes en un intervalo definido, toma un mensaje PMT y lo envía cada milisegundo. Útil para probar/depurar el sistema de mensajes'. Los parámetros son Message PMT es el mensaje para enviar como PMT; Period (ms) es el intervalo de tiempo con el cual se envían los datos a la salida repetidamente.

- **Char To Float:** Este bloque se encarga de convertir la información para que pase de ser un flujo de caracteres a datos flotantes.
- **Delay:** Este bloque retrasa la entrada por un numero de muestras, los retrasos positivos insertan cero elementos al comienzo de la transmisión y los retrasos negativos descartan elementos de la transmisión. El parámetro Delay es el número de muestras/artículos a retrasar.
- **File Sink:** El bloque se utiliza para escribir una secuencia en un archivo binario, puede leer entornos de programación que lean archivos binarios. Los parámetros son File es la ruta del archivo para abrir y escribir la salida; Unbuffered especifica si la salida se almacena en la memoria intermedia; Append file ofrece una opción para agregar al archivo o sobrescribir el archivo.
- **QT GUI Constellation sink:** Este bloque se utiliza para para mostrar gráficamente la constelación de IQ de varias señales, el puerto de mensajes se denomina "in" y admite transmisión de datos o mensajes complejos. Los parámetros Number of points es el número de puntos de la señal; Autoescale para que los ejes se vean correctos en la gráfica.
- **Protocol Formatter (Async):** Este bloque utiliza un objeto de formato de encabezado para agregar un encabezado a una PDU (unidad de datos de protocolo). El parámetro Format obj se usará al leer el encabezado.
- **Constellation Rect. Object:** En este bloque se pueden definir el número de símbolos que se van a poder apreciar en el diagrama de constelación, definiendo la ubicación que van a tener estos símbolos en el diagrama.
- **QT GUI Tab Widget:** Este bloque crea un widget que es una pequeña aplicación diseñada para facilitar el acceso a las funciones con pestañas para organizar otros widgets.
- **QT GUI Message Edit:** Este bloque consiste en administrar datos a través de una interfaz de paso de mensajes, primero se verifica el tipo correcto de los mensajes (entero, flotante, cadena o complejo) y luego se convierten en cadenas para mostrar en los campos de entrada si los datos no son del tipo correcto y la conversión falla, el bloque genera un mensaje de advertencia de registro, pero no genera los datos. Los parámetros para este bloque son Value es el valor predeterminado del mensaje, se ingresa como una cadena independientemente del tipo; Label es una etiqueta para saber cuál es el campo de entrada en la pantalla; Pair mode es verdadero si un par clave: valor; Static mode verdadero si el campo de entrada clave es un cuadro de texto estático. El puerto de salida 'msg' produce mensajes basados en el texto de los campos de entrada y el tipo de datos establecido por el argumento de tipo.

- **Embedded Python Block:** Este bloque permite personalizar un nuevo bloque por medio de código en Python. Para que el nuevo bloque se ejecute de la manera deseada, este debe ser llamado con el parámetro ID del bloque Embedded Python Block.
- **Constellation Modulator:** El bloque sirve para modulación genérica diferencial filtrada por RRC, la entrada es un flujo de bytes y la salida es la señal compleja modulada en banda base. Los parámetros son constellation determina el tipo de modulación; Differential encoding se usa para codificación diferencial; Samples/Symbol son las muestras por baudio; Excess BW es el exceso de ancho de banda del filtro de coseno elevado (RRC).
- **Multiply Const:** La función de este bloque es multiplicar por una constante determinada la información de una señal que sea su entrada. Esta constante puede ser escalar o vectorial.
- **RRC Filter Taps:** La función de este bloque es el filtrado de raíz de coseno elevado, se utiliza para filtrados adaptados de pulsos en los sistemas de comunicaciones digitales. Los parámetros de este bloque son la ganancia del filtro (Gain); la frecuencia de muestreo en muestras por segundo (Sample Rate); velocidad por símbolo (Symbol rate) que tiene que ver con la tasa de muestreo, sus unidades pueden ser [muestras/segundo] o [muestras/símbolo]; exceso de ancho de banda (Excess BW) puede ser llamado alfa; número de toques (Num taps) tiene un valor por defecto que es $11 \cdot \text{samp_rate}$.
- **Adaptive Algorithm:** Es un bloque que crea un algoritmo adaptivo que se utiliza en el ecualizador lineal, para el cálculo de la señal de error y como se actualizan las derivaciones. Los parámetros son Algorithm type se pueden utilizar los algoritmos LMS, NLMS, CMA; Step size este especifica la rapidez para la cual converge el algoritmo, si es alto el ecualizador se vuelve inestable; Modulus donde se especifica el número de puntos de constelación.
- **Note:** Se usa para agregar una nota o comentario a un diagrama de flujo. El parámetro nota (Note) es el comentario que se quiere hacer a algún bloque.
- **Polyphase Clock Sync:** Este bloque sincroniza el tiempo por medio de bancos de filtros polifásicos. Los parámetros son Samples/symbol se encarga de saber el número de muestras por símbolo porque por defecto devuelve un único punto que representa el símbolo y esto es para el bloque de sincronización del reloj; Loop BW se utiliza para establecer la ganancia del bucle de control interno y se debe colocar un valor pequeño; Taps llamados grifos de filtro son un parámetro muy importante, un beneficio de este algoritmo es que puede colocar el filtro coincidente aquí como los toques, por lo que obtiene tanto el filtro coincidente como la corrección de tiempo de muestra de una sola vez; Filter size se puede configurar la cantidad de filtros y pueden ser 32 con estos 32 se obtiene una resolución muy buena en la fase para

producir un ISI (interferencia entre símbolos) muy pequeño; Initial phase es un parámetro configurable y se refiere a la ruta de filtro que el algoritmo observa inicialmente; Maximum rate deviation esto es lo lejos que permitimos que d_rate oscile, positivo o negativo, desde 0, restringir la tasa puede ayudar a evitar que el algoritmo se aleje demasiado para bloquearse en momentos en que no hay señal; SPS de salida (Output SPS) es el número de muestras de salida por símbolo, se agregó para trabajar mejor con los ecualizadores, que modelan mejor el canal si tienen 2 muestras/sim.

- **Linear Equalizer:** Este bloque realiza la ecualización lineal en un flujo de muestras complejas, lo que hace es igualar la señal de entrada por medio de un filtro FIR, el ecualizador diezma la tasa de símbolos con respecto a las muestras por parámetro de símbolo. Los parámetros son Num taps es el número de toques para el filtro FIR; Input samples per-symbol se establece el número de muestras que queremos obtener.
- **Costas Loop:** Es un módulo de recuperación de portadora de bucle de Costas, muy bueno para señales BPSK, QPSK y 8PSK, este bucle se bloquea en la frecuencia central de una señal y la convierte en banda base. Los parámetros son Loop bandwidth el ancho de banda del bucle interno de segundo orden; Order es el orden de los bucles puede ser 2, 4 u 8, se utiliza 2 para BPSK donde la parte real de la señal de salida es la señal BPSK de banda base y la parte imaginaria es la señal de error; se utiliza 4 para QPSK donde se emiten tanto I como Q (real e imaginario) y 8 para 8PSK.
- **Constellation Soft Decoder:** Este bloque decodifica los puntos de una constelación de un espacio complejo a bits suaves según el mapa y la LUT (tabla de búsqueda) de decisión suave del objeto. El parámetro Constellation object es un objeto de constelación.
- **Binary Slicer:** Este bloque se encarga de "cortar un valor flotante producido por la salida de 1 bit, a entrada positiva produce un 1 binario y la entrada negativa produce un cero binario".
- **NRZI Decode:** La decodificación sin retorno a cero invertido es un código binario en el que los unos están representados por una condición significativa normalmente es un voltaje.
- **Descrambler:** Este bloque descifra un flujo de entrada usando LFSR (registro de desplazamiento con retroalimentación lineal) este bloque funciona únicamente en un flujo "binario desempquetado", y produce el mismo formato en su salida. Los parámetros son la Mask es la máscara de polinomio; Seed es el contenido del registro de desplazamiento inicial; Length es para la longitud del registro de desplazamiento.
- **Message Debug:** Este bloque de depuración de mensajes se utiliza para capturar e imprimir o almacenar mensajes a medida que se reciben, se puede conectar cualquier bloque que

produzca mensajes. El parámetro es PDU vectors determina si el vector uniforme se muestra o no y decide si se activa o se desactiva.

- **UHD: USRP Sink:** Este bloque se utilizado para transmitir muestras a un dispositivo USRP actúa como transmisor.
- **NRZI Encode:** La codificación sin retorno a cero invertido es un código binario en el que los unos están representados por una condición significativa normalmente es un voltaje.
- **QT GUI Time Sink:** La función de este bloque se puede apreciar a la hora de correr un flujograma, ya que en la pantalla de simulación aparece un campo en la pantalla donde se pueden visualizar varias señales en el dominio del tiempo. Los parámetros para destacar para este bloque son el número de puntos que se van a graficar y la frecuencia de muestreo.
- **QT GUI Frequency Sink:** Este bloque, de igual forma que el anterior se ve un campo en la pantalla de simulación en el dominio de la frecuencia, en el que se puede apreciar el espectro en frecuencia de una señal. Los parámetros para destacar son la frecuencia central, la frecuencia de muestreo y el tamaño de la FFT.
- **Random Source:** Este bloque opera como un generador de muestras de números aleatorios, los cuales se van a regir por los parámetros que se le asignen a este bloque. Dichos parámetros son los márgenes mínimo y máximo hasta donde van a variar estos números aleatorios, y también se define la cantidad de muestras que se van a tomar.

2. Validación de los Sistemas:

Luego de que se haya identificado el funcionamiento de los bloques que conforman los flujogramas tanto de transmisión como de recepción de las respectivas modulaciones (QPSK y GMSK), se procedió a realizar la implementación de estos y observar que tipos de señales y de información se obtenía en el TX y en RX. Este proceso se realizó inicialmente por medio de un enlace cableado, para el cual se hizo necesario hacer equipo entre dos grupos de trabajo, en este caso entre los bancos 7 y 8, donde uno haría el papel de TX y el otro de RX.

GMSK: Transmisión: (Banco 7 fue receptor) La primera modulación que se implemento fue la **GMSK**, donde el flujograma para la transmisión fue el siguiente:

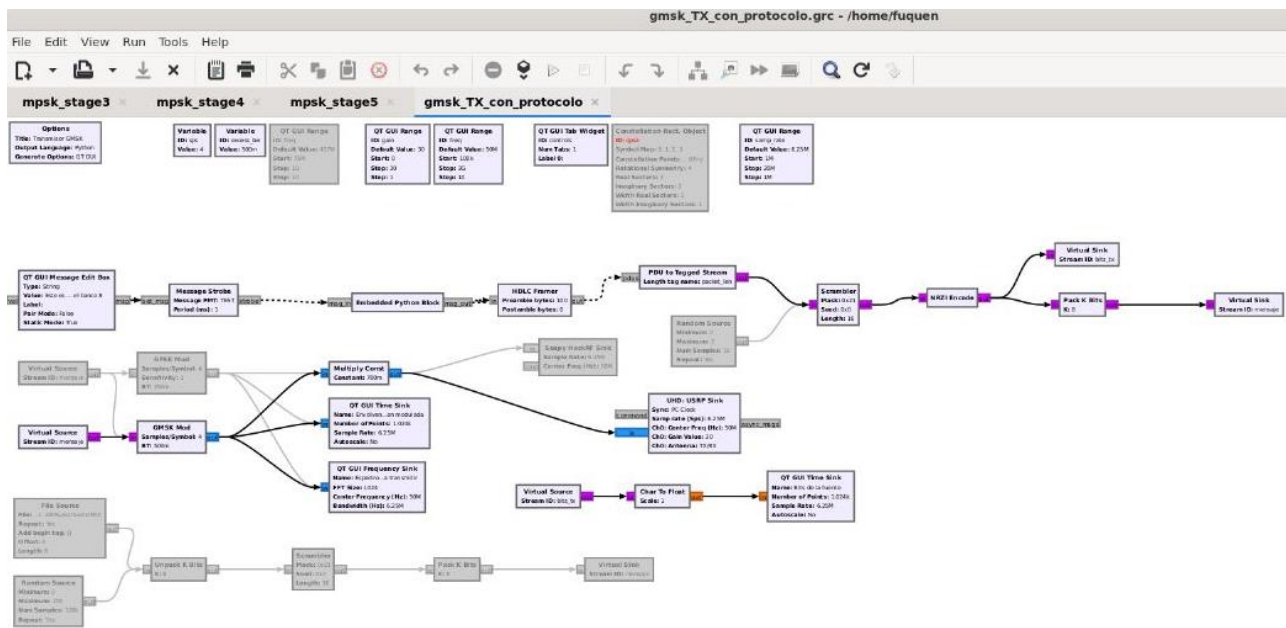


Ilustración 7. Flujograma GMSK TX.

Dicho flujograma fue modificado en determinados valores de interés como lo es la frecuencia central en la que se podrá ver el espectro en frecuencia, la cual fue de 50MHz, este valor fue escogido con el fin de que en el osciloscopio se pudiera obtener el tiempo del bit. Para el enlace cableado, se hizo uso de dos radios USRP, donde uno de los extremos de un cable coaxial se acoplo en el terminal de transmisión de un radio y el otro extremo al terminal de recepción del segundo radio. Es importante que tanto los flujogramas de TX y RX tengan el mismo valor de frecuencia central, teniendo en cuenta el rango de frecuencias en las cuales operan los radios.

A la hora de que se ejecute el flujograma en la parte de transmisión para la modulación GMSK, se puede visualizar la información a transmitir, la envolvente compleja de la información y el espectro en frecuencia, donde se observa que el centro se encuentra en los 50 MHz establecidos anteriormente. También, en la parte superior se puede ver los parámetros que se pueden variar y así observar cómo se alteran los diagramas al darse las modificaciones y el mensaje a transmitir. La pantalla con las gráficas mencionadas es la siguiente:

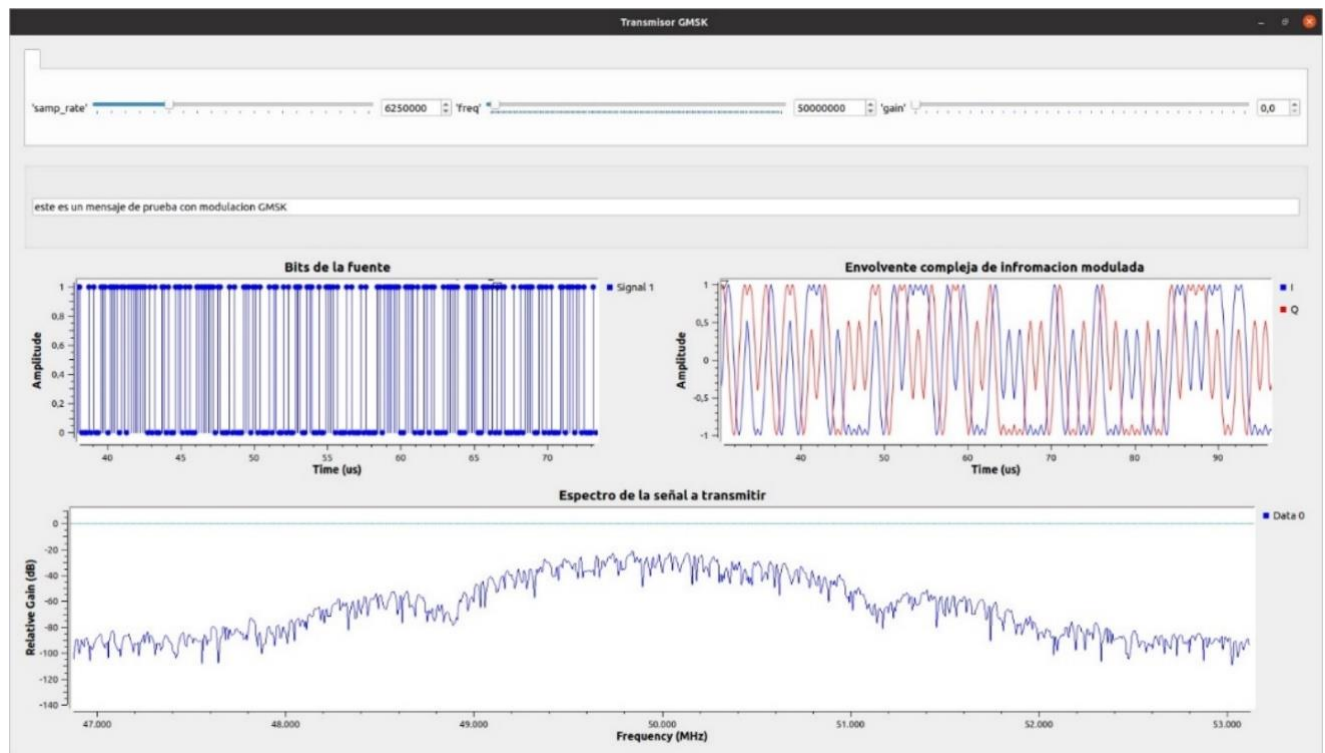


Ilustración 8. Simulación GMSK TX.

Por la parte de recepción también se ejecuta el flujograma, dándole la orden al radio correspondiente para que opere con la frecuencia central de 50 MHz, para que coincida con la frecuencia especificada en el transmisor. En la simulación del receptor se puede observar un diagrama de constelación y un espectro en frecuencia en el cual se observa una campana de Gauss a causa del filtro gaussiano. Para que exista una comunicación eficiente entre el TX y RX, tanto el diagrama de constelación como el espectro en frecuencia deben mostrarse de una manera estable, como la que se aprecia en la siguiente imagen:

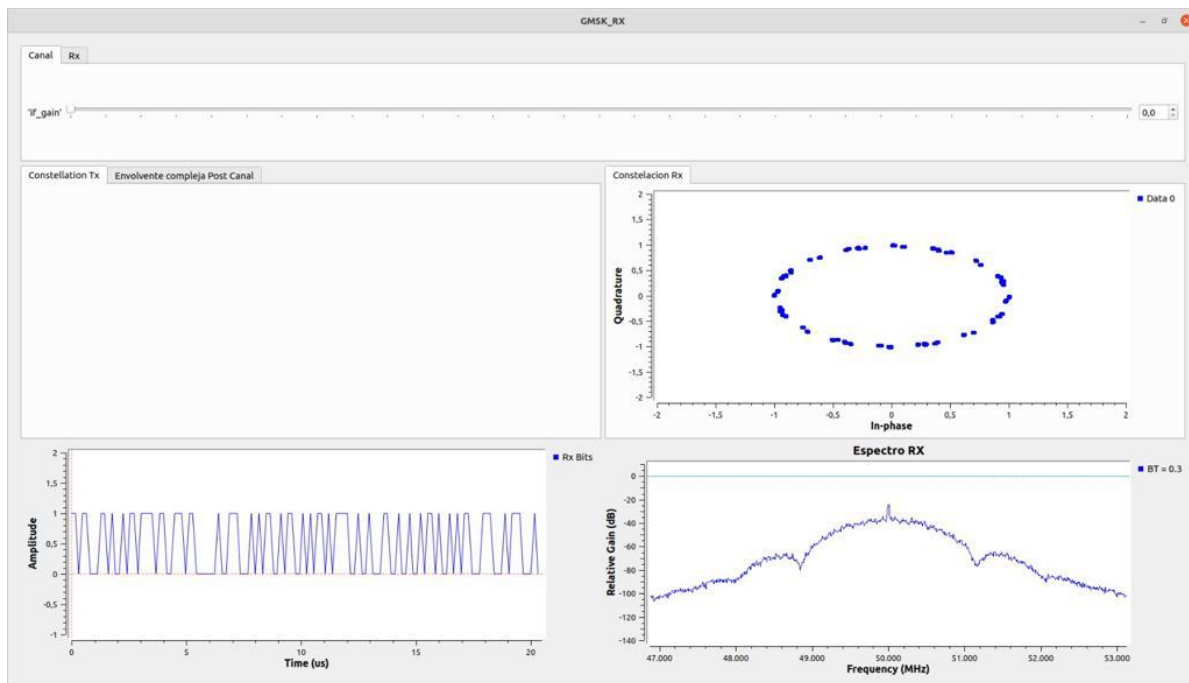


Ilustración 9. Simulación GMSK RX.

El mensaje recibido se puede ver en la parte inferior del flujograma, pero este llega de manera hexadecimal (recuadro rojo), como se aprecia en la siguiente figura:

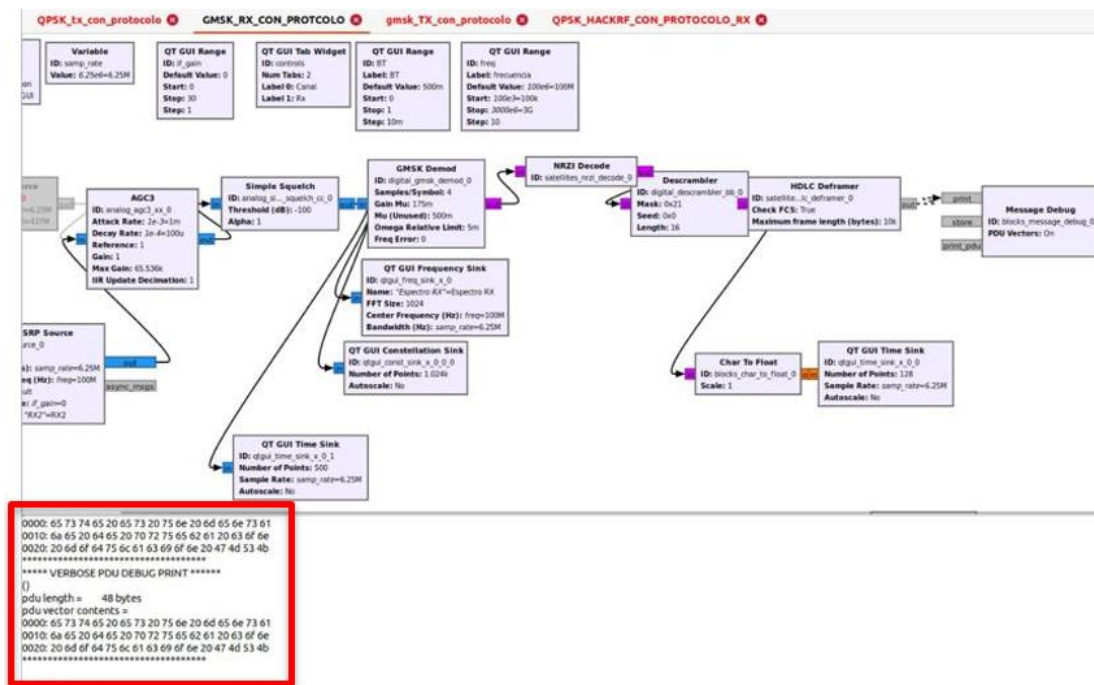


Ilustración 10. Recepción de mensaje – Enlace cableado.

Con el fin de determinar cuál fue el mensaje que llega al receptor, se utiliza una herramienta para pasar datos de hexadecimal a ASCII, dando como resultado el mensaje del recuadro verde:

RapidTables

Home » Conversion » Number conversion » Hex code to ASCII text

Hex to ASCII Text String Converter

Enter hex bytes with any prefix / postfix / delimiter and press the Convert button
(e.g. 45 78 61 6d 70 6c 65 21):

From: Hexadecimal To: Text

Open File

Paste hex numbers or drop file

```
65 73 74 65 20 65 73 20 75 6e 20 6d 65 6e 73 61
6a 65 20 64 65 20 70 72 75 65 62 61 20 63 6f 6e
20 6d 6f 64 75 6c 61 63 69 6f 6e 20 47 4d 53 4b |
```

Character encoding: ASCII

Convert Reset Swap

este es un mensaje de prueba con modulacion GMSK

Ilustración 11. Mensaje recibido – Enlace cableado.

Se puede evidenciar que el mensaje mostrado en la **Ilustración 8**, debe coincidir con el mensaje transmitido, un factor importante es la cantidad de caracteres que el transmisor envía, ya que el ruido hace que la comunicación no se dé una manera eficiente cuando el mensaje es de corta longitud.

Luego de validar la parte de la comunicación por medio de un enlace cableado, se procedió a realizar la validación por un enlace inalámbrico. Para esto se hizo uso de las antenas de los radios, las cuales fueron acopladas en los mismos terminales que el cable coaxial utilizado en el enlace anterior.

Por parte de la simulación del TX, se tiene los diagramas del mensaje, su envolvente compleja y su espectro en frecuencia, el cual presenta un poco más de distorsión que el visto en la **Ilustración 6**, esto se debe a que el medio por el que se envía el mensaje es el aire, y este está expuesto a recibir alteraciones de distintas fuentes. Para este caso se transmitió el mensaje con una frecuencia central de 100 MHz, con el fin de validar el sistema con un valor diferentes. Las gráficas por parte del TX del GMSK son las siguientes:

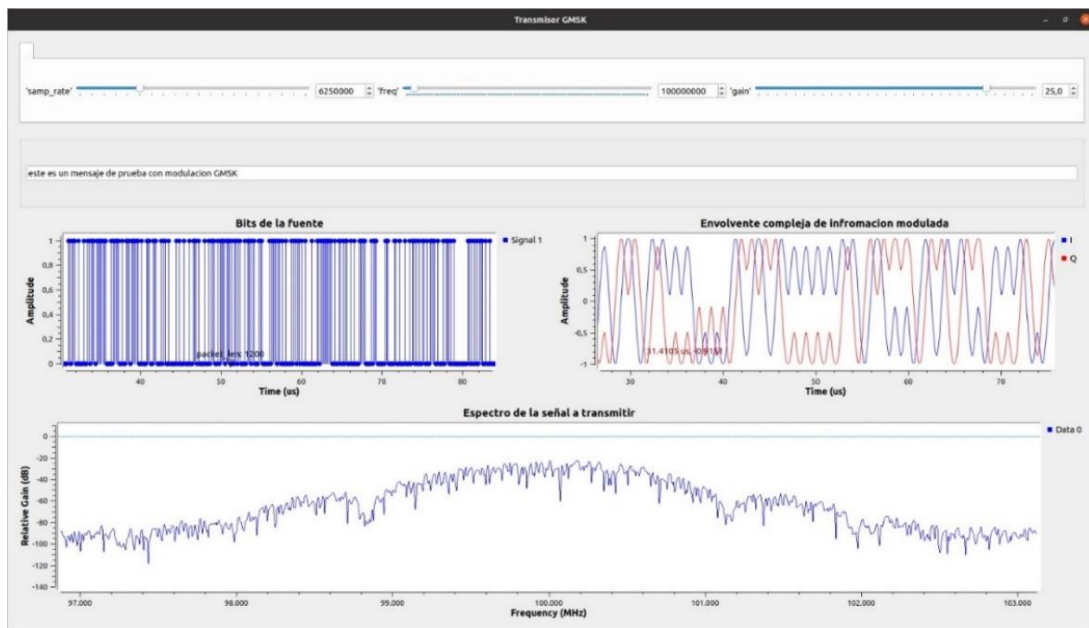


Ilustración 12. Simulación GMSK inalámbrico TX.

El efecto de cambiar de medio de transmisión también se puede observar en el diagrama de constelación y del espectro en frecuencia vistos en la simulación del receptor, donde en la constelación las muestras en los símbolos se encuentran más dispersos que las vistas en la **Ilustración 7**. Por parte del espectro en frecuencia, ya no se puede apreciar la forma de la campana de Gauss de una manera definida.

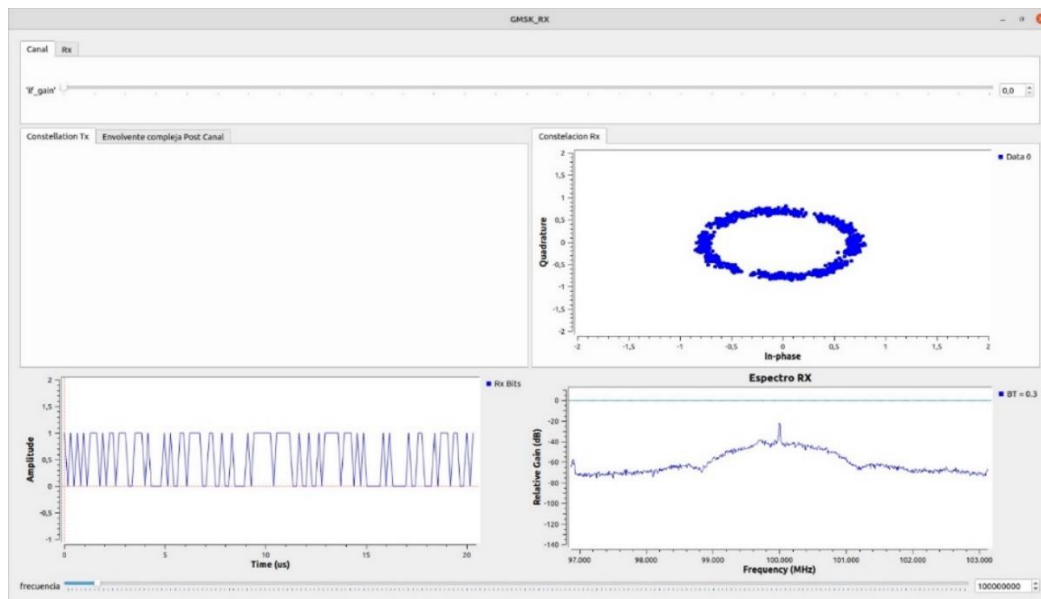


Ilustración 13. Simulación GMSK inalámbrico RX.

Para que el mensaje sea transmitido de la mejor manera se ubicaron los radios con su respectiva antena de tal forma que no haya objetos que obstaculicen el medio de transmisión y genere interferencias entre TX y RX. La disposición que se tuvo en el laboratorio entre los radios, y los computadores que sirvieron como Transmisor (Izquierda) y Receptor (Derecha), fue la siguiente:



Ilustración 14. Comunicación inalámbrica en el laboratorio.

Recepción (Banco 10 fue transmisor): Después de haber transmitido el mensaje, también se validó el flujograma para la recepción de este, donde los compañeros del banco 10 transmitieron un mensaje a 800 MHz. El diagrama de constelación y el espectro en frecuencia se aprecian en la **Ilustración 12**.

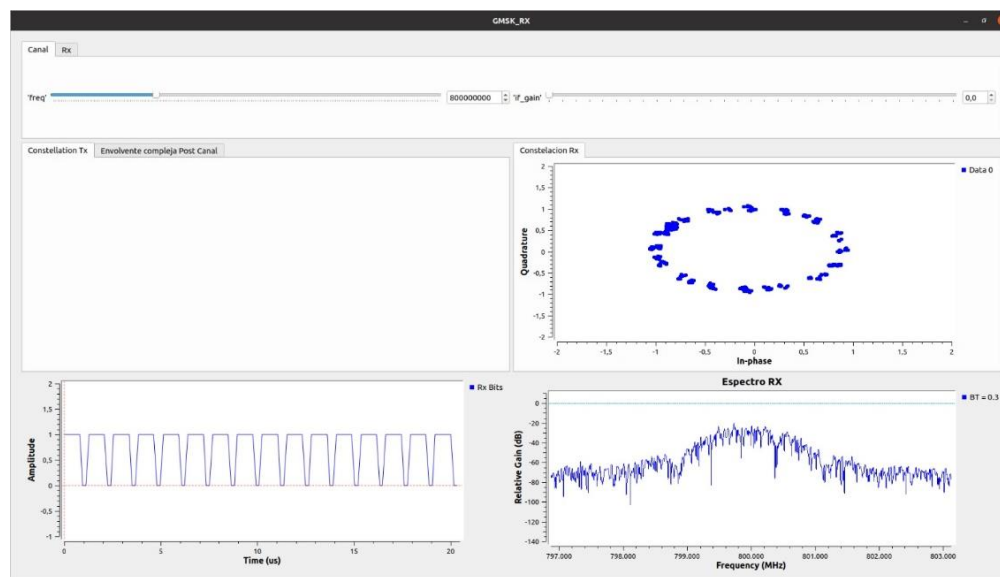


Ilustración 15. Simulación GMSK inalámbrico RX.

Como se aprecia en la anterior figura, tanto el diagrama de constelacion como el espectro en frecuencia no tuvieron un poco de distorsion, por lo que se procedio a visualizar el flujograma visto en la **Ilustracion 13**, el cual corresponde al RX, ya que en su parte inferior (Recuadro azul) se encuentra el mensaje en hexadecimal.

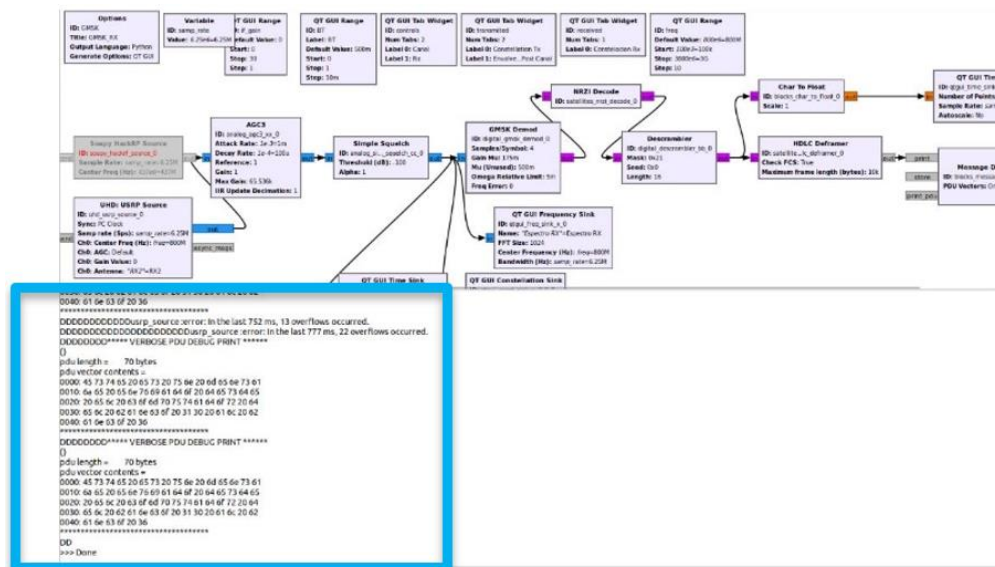


Ilustración 16. Recepción de mensaje – Enlace inalámbrico.

Dicho mensaje es pasado de hexadecimal a ASCII mediante una herramienta de conversión, pudiendo obtener así el mensaje visto en el recuadro amarillo.

RapidTables

Inicio > Conversión > Conversión de números > Código hexadecimal a texto ASCII

Convertidor de cadena de texto hexadecimal a ASCII

Ingrese los bytes hexadecimales con cualquier prefijo/postfijo/delimitador y presione el botón Convertir
(por ejemplo, 45 78 61 6d 70 6c 65 21):

De: hexadecimal A: Texto

Pegar números hexadecimales o soltar archivos

```
45 73 74 65 20 65 73 20 75 6e 20 6d 65 6e 73 61
6a 65 20 65 6e 76 69 61 64 6f 20 64 65 73 64 65
20 65 6c 20 63 6f 6d 70 75 74 61 64 6f 72 20 64
65 6c 20 62 61 6e 63 6f 20 31 30 20 61 6c 20 62
61 6e 63 6f 20 36
```

Codificación de caracteres

ASCII

Este es un mensaje enviado desde el computador del banco 19 al banco 6

Ilustración 17. Mensaje recibido – Enlace inalámbrico.

La simulación del mensaje transmitido a 800MHz, es la que se observa en la **Ilustración 15**, y se aprecia que a diferencia del espectro recibido, este tiene la forma de la campana de Gauss mucho mas definida.

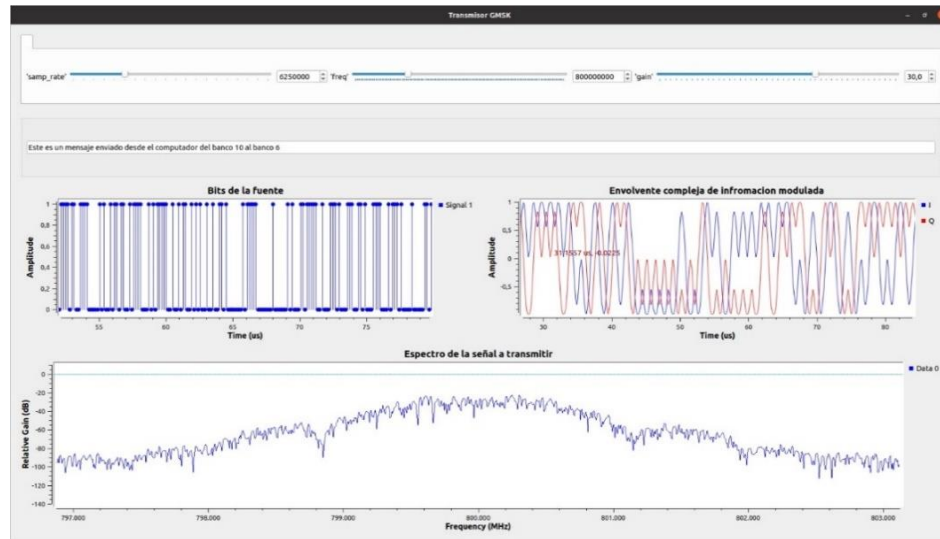


Ilustración 18. Simulación GMSK inalámbrico TX.

Luego de haber validado tanto la parte de transmisión como la de recepción del mensaje, se hizo una verificación por medio de los equipos del laboratorio (osciloscopio y analizador de espectro). Por medio de cableado entre el radio y estos equipos, se pueden tomar los datos de ancho de banda y de tiempo de bit.

Ancho de Banda: Este dato se toma del espectro en frecuencia, el cual se puede ver en el analizador de espectro en la **Ilustración 14**. Sobre este espectro son ubicados dos cursores, uno a cada lado del lóbulo principal:

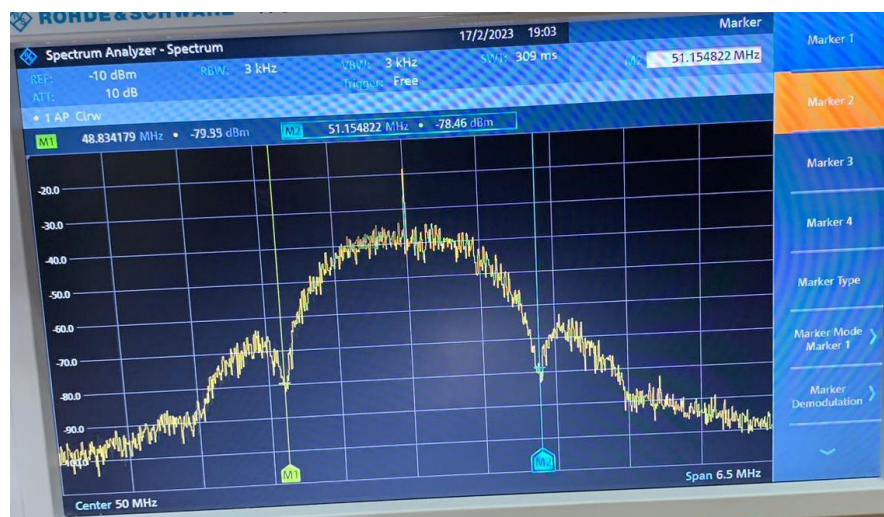


Ilustración 19. Espectro visto en el analizador.

Los dos cursores están ubicados en determinados valores de frecuencias de la siguiente manera:

$$M1 = 48.834179 \text{ MHz}$$

$$M2 = 51.154822 \text{ MHz}$$

Pudiéndose así calcular el ancho de banda, como la diferencia entre las frecuencias dadas por los cursores:

$$BW = 51.154822 - 48.834179 = 2.320643 \text{ MHz}$$

Tiempo de bit: Para la obtención del tiempo de bit, solo basta con obtener el inverso del valor de BW obtenido anteriormente, de la siguiente manera:

$$tb = \frac{1}{BW} = \frac{1}{2.320643 \text{ MHz}} = 431 \text{ nseg}$$

Para verificar este dato, también se puede utilizar el osciloscopio, donde se pueden ver los datos en el tiempo. En esta señal en el tiempo se busca el lóbulo más pequeño, el cual equivale a un bit de información, y haciendo uso de los cursores se toma el tiempo que hay desde donde comienza hasta donde termina este lóbulo, eso se puede apreciar mejor en la siguiente figura:

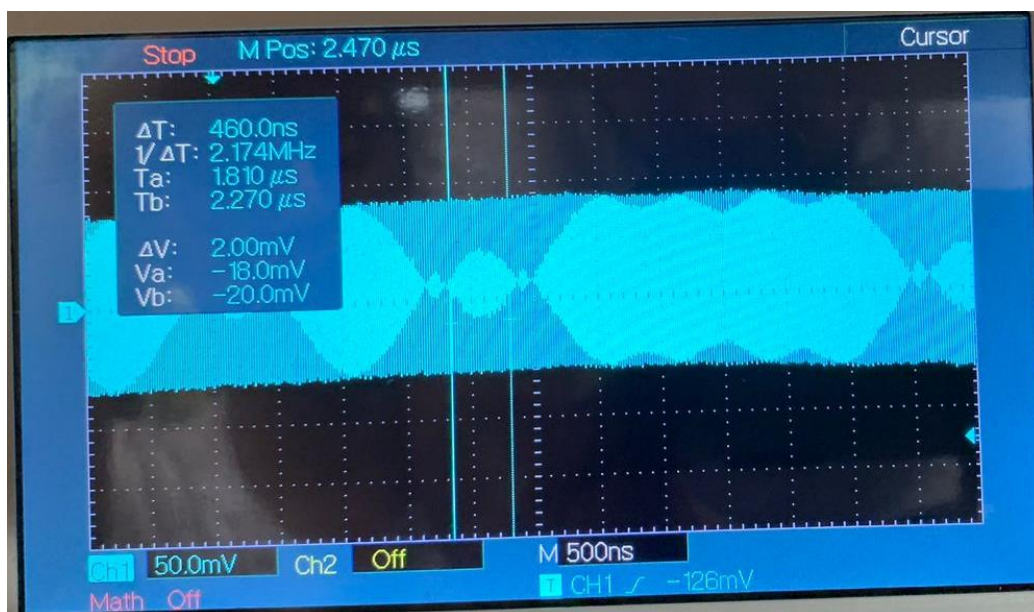


Ilustración 20. Obtención tiempo del bit.

El tiempo de bit obtenido por medio del osciloscopio fue de 460 nseg, el cual es un valor muy cercano a los 431 nseg obtenidos por medio del ancho de banda. En la **Ilustración 18** se pueden observar el analizador de espectro y el osciloscopio en función, con las señales en el tiempo y en frecuencia, y el acople cableado que existe entre ellos.



Ilustración 21. Osciloscopio y analizador en función.

QPSK: Transmission: La otra modulación a implementar fue la **QPSK**, donde en la simulación vista en la **Ilustración 19** se pueden apreciar los diagramas de la señal con la información empaquetada en el tiempo, la envolvente compleja, su espectro en frecuencia y el diagrama de constelación con cuatro símbolos semidefinidos, ya que las muestras que los conforman están dispersas.

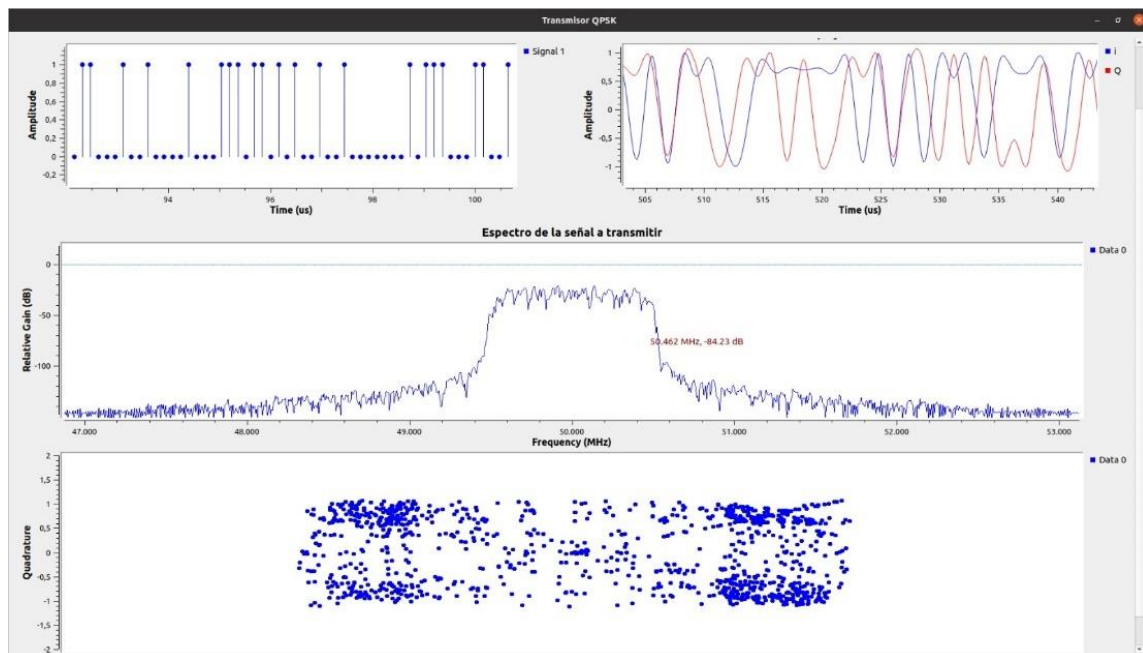


Ilustración 22. Simulación QPSK TX.

La simulación anterior se dio como resultado al ejecutar el flujograma de la **Ilustración 21**, donde se puede definir el número de símbolos y la frecuencia central a la cual serán transmitidos.

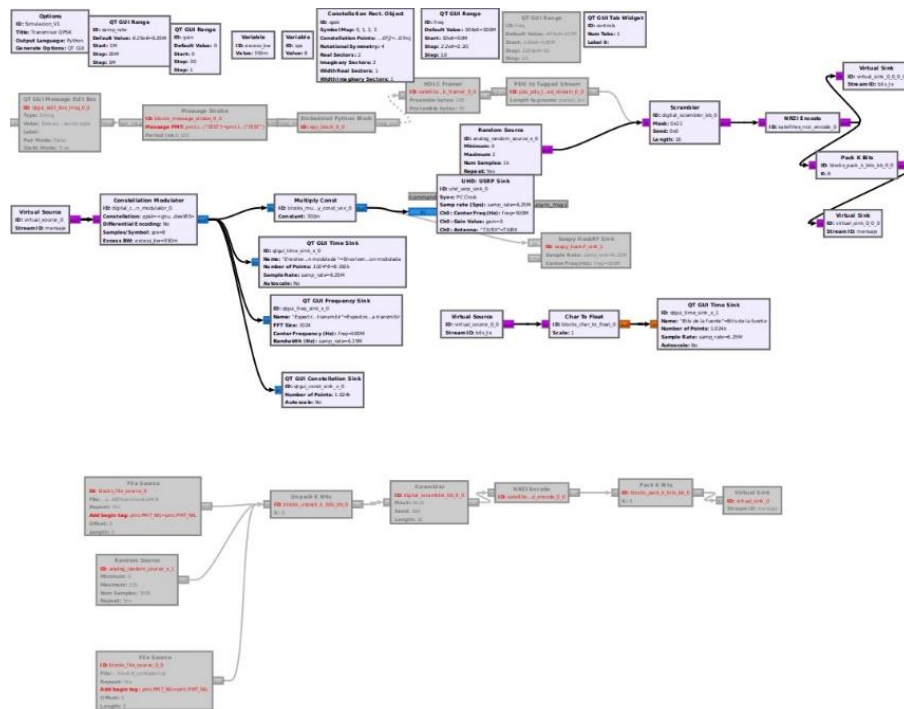


Ilustración 23. Flujograma QPSK TX.

Con el fin de hacer una comparacion con el espectro y la señal en el tiempo obtenidos, se hizo uso de los equipos de laboratorio para asi poder obtener los valores de ancho de banda (BW) y de tiempo del bit (tb). Por medio del analizador de espectro se visualiza el siguiente espectro en frecuencia del mensaje transmitido:

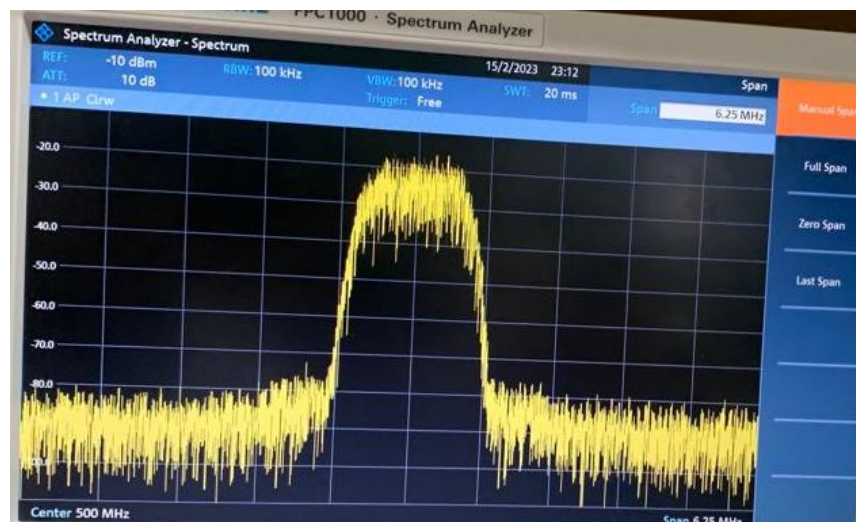


Ilustración 24. Espectro visto en el analizador.

Por medio de los cursores del analizador y posicionándolos en los extremos del lóbulo principal, se puede determinar el BW como la diferencia de las frecuencias dadas por los marcadores. También, el

BW se puede obtener de la simulación, donde se puede apreciar un valor en pantalla, en la **Ilustración 20**, el cual es:

$$M1 = 50.462 \text{ MHz}$$

La frecuencia central para este caso es de 50 MHz, y el valor anterior se encuentra posicionado en un extremo del lóbulo principal, por lo que la diferencia de estos dos valores daría como resultado la mitad del ancho de banda.

$$\frac{BW}{2} = 50.462 - 50 = 0.462 \text{ MHz}$$

Despejando de la expresión anterior, el ancho de banda para la modulación QPSK sería el siguiente:

$$BW = 0.462 * 2 = 0.924 \text{ MHz}$$

Nuevamente se podría obtener del tiempo de bit, solo con el BW obtenido por medio de la simulación, de la siguiente manera:

$$tb = \frac{1}{BW} = \frac{1}{0.924 \text{ MHz}} = 1.082 \text{ useg}$$

Para verificar que el valor de tb obtenido por medio del BW, se hizo uso del osciloscopio, donde se pueden ver los datos en el tiempo. En esta señal en el tiempo se busca el lóbulo más pequeño, el cual equivale a un bit de información, y haciendo uso de los cursores se toma el tiempo que hay desde donde comienza hasta donde termina este lóbulo, eso se puede apreciar mejor en la siguiente figura:

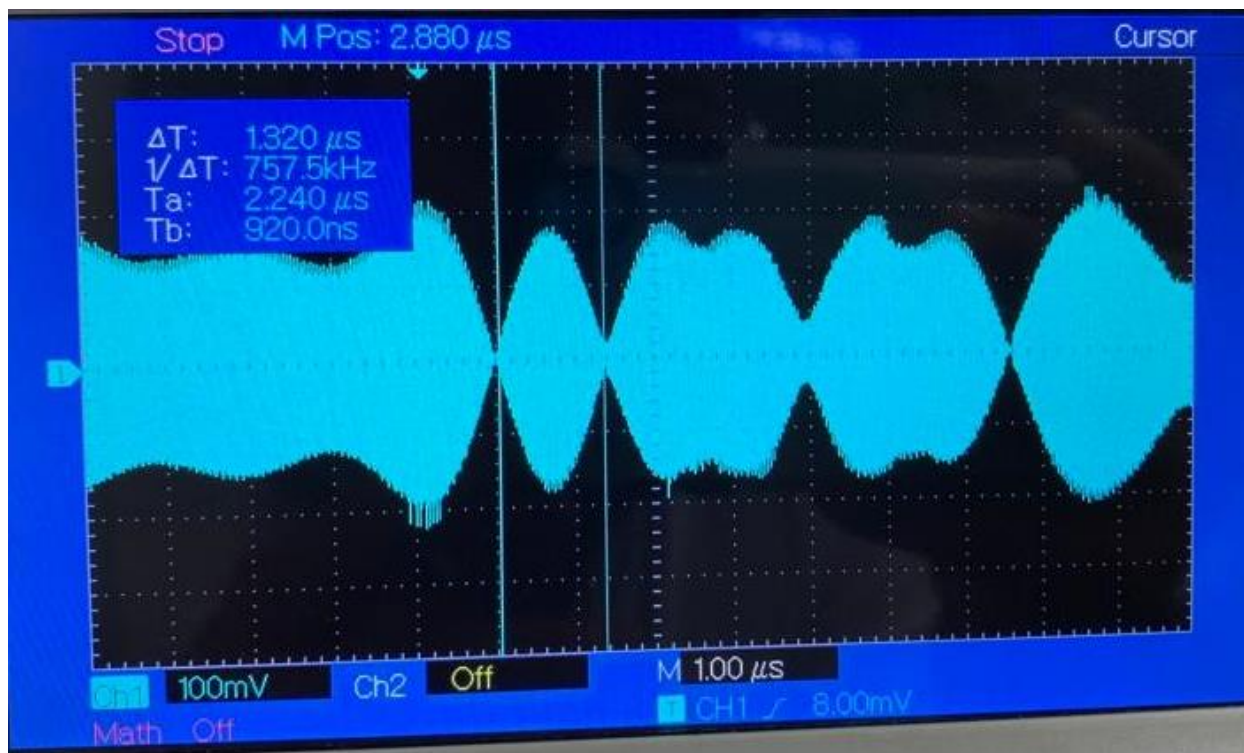
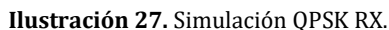


Ilustración 25. Obtención tiempo del bit por medio del osciloscopio.

[illegible]

A la hora de ejecutar el flujograma anterior, se puede ver en la pantalla de la simulación el diagrama de constelación en el cual se obtienen nuevamente los cuatro símbolos transmitidos con un poco de distorsión. Es importante que tanto en el flujograma de TX y RX, estén especificados el mismo número de símbolos y la misma ubicación en el plano, esto con el fin de que haya una comunicación más eficiente. En la simulación también se puede ver la señal del mensaje en el tiempo y su espectro en frecuencia.



3. Polyphase Clock Sync: Sincronización de tiempo, el flujograma es el siguiente:

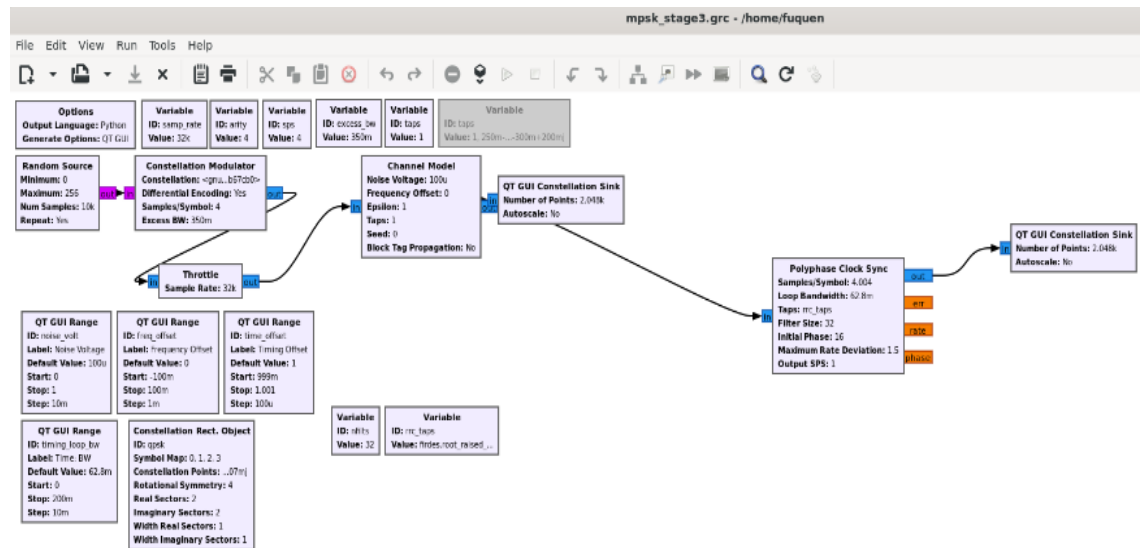


Ilustración 28. Flujograma de Polyphase Clock Sync (PCS).

El bloque de Polyphase Clock Sync (PCS) se utiliza para la sincronización del tiempo de los símbolos en sistemas de comunicación digital. Se utiliza para sincronizar el reloj de muestreo del receptor con el reloj de símbolos del transmisor, lo cual es necesario para que el receptor muestree y demodule correctamente los símbolos recibidos.

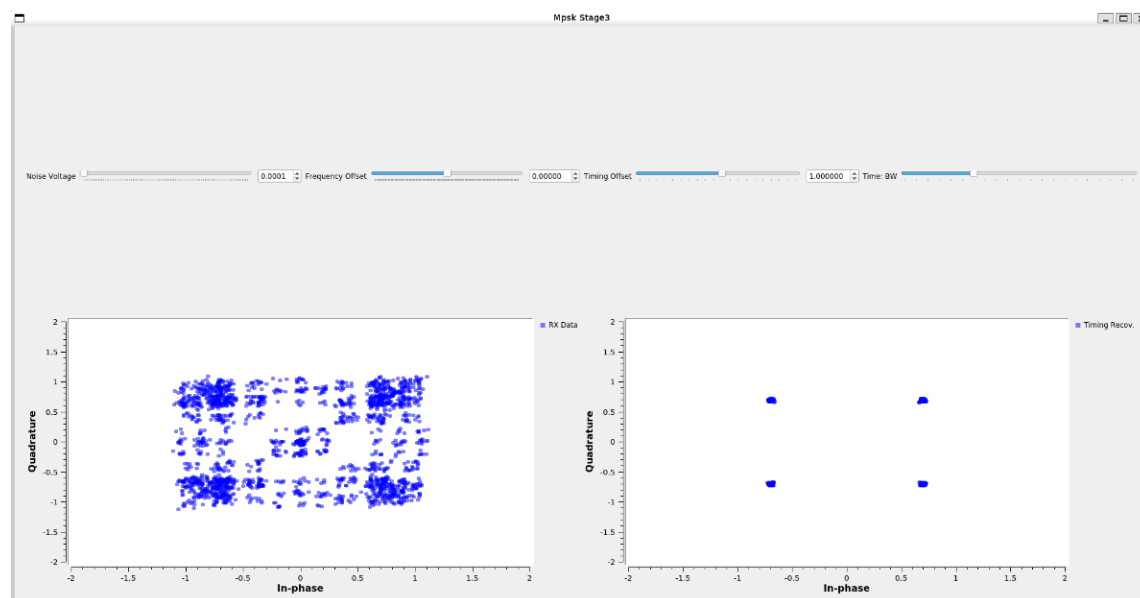


Ilustración 29. Simulación del flujograma de Polyphase Clock Sync (PCS).

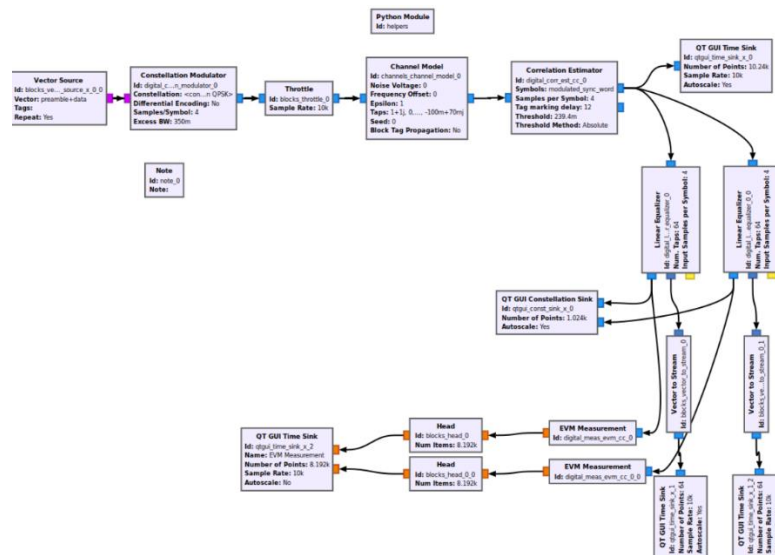


Ilustración 30. Flujograma linear equalizer.

Este diagrama de bloques sirve para corregir la distorsión y la interferencia en los sistemas de comunicación. Se utiliza para mitigar los efectos de las degradaciones del canal, como el desvanecimiento por trayectos múltiples, ajustando la amplitud y la fase de las señales recibidas. El ecualizador lineal funciona con una señal recibida y le aplica una corrección en función de las características del canal.

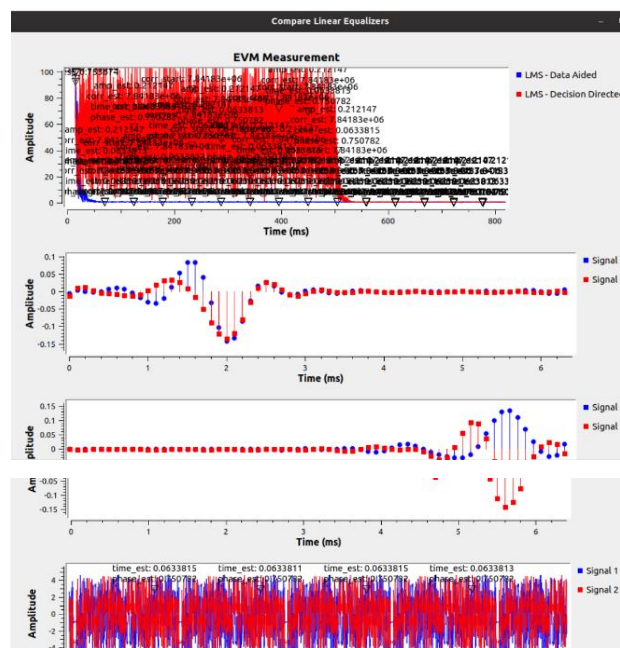


Ilustración 31. Simulación linear equalizer.

4.2. Adaptive Algorithm:

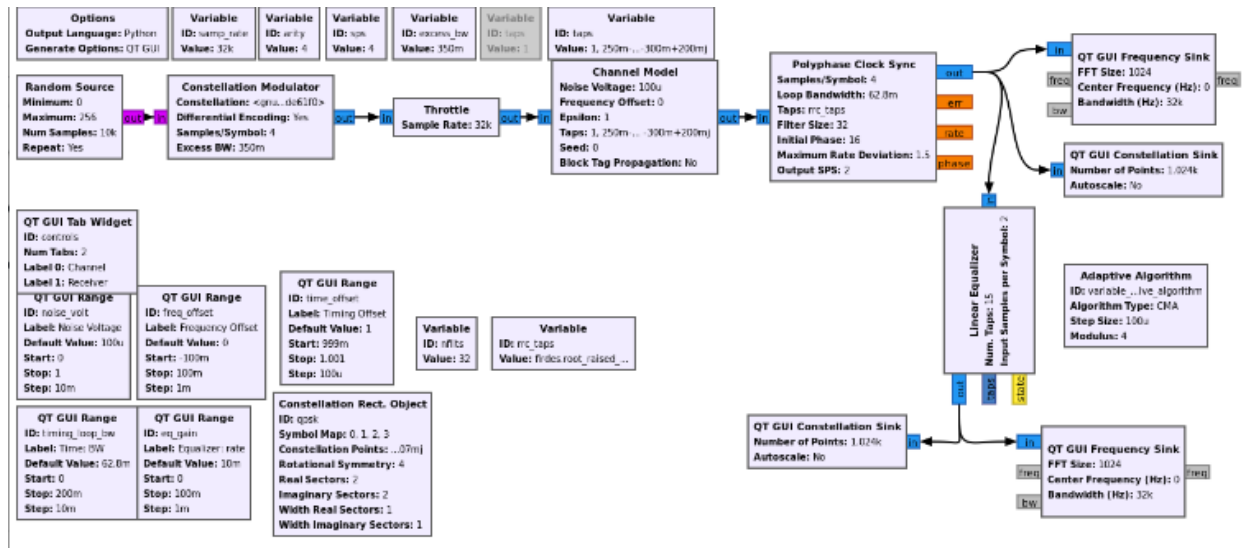


Ilustración 32. Flujograma linear equalizer con Adaptive Algorithm.

Cuando se aplica un ecualizador lineal en GNU Radio utilizando un algoritmo adaptativo, puede mejorar el rendimiento de la ecualización sobre un ecualizador lineal fijo o estático. Un ecualizador adaptativo puede ajustar continuamente sus derivaciones de filtro para adaptarse mejor a la respuesta del canal y corregir cualquier distorsión o deterioro en la señal recibida.

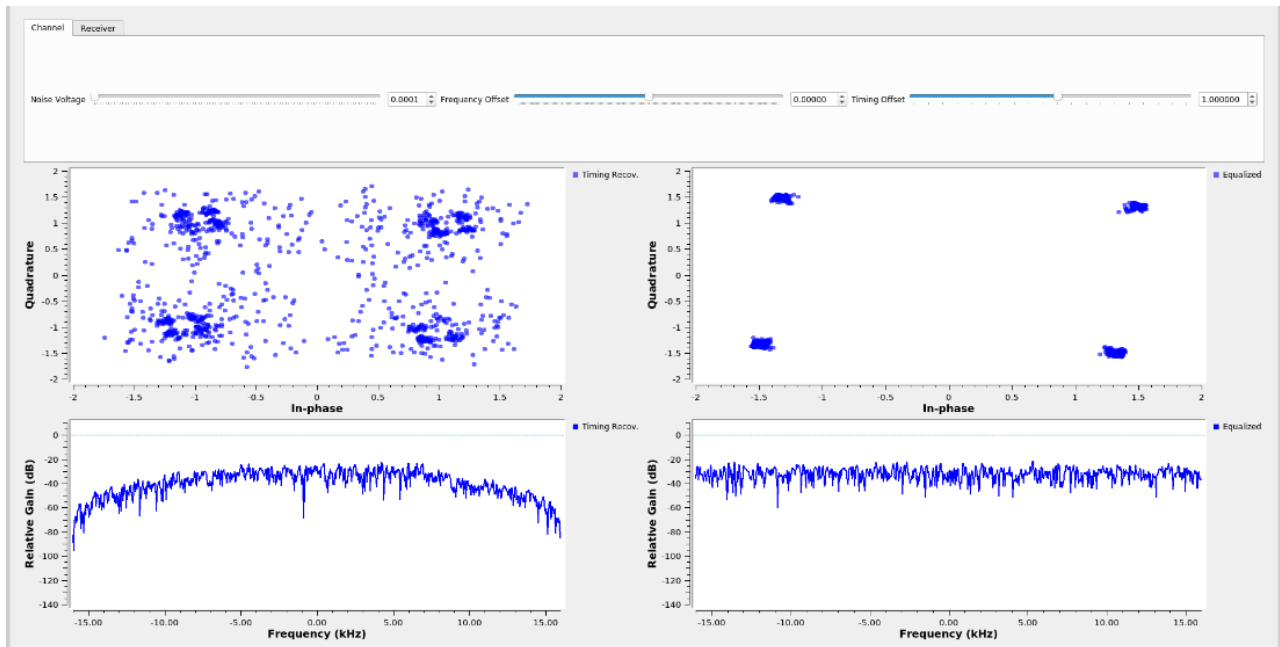


Ilustración 33. Simulación linear equalizer con Adaptive Algorithm.

5. Costas Loop:

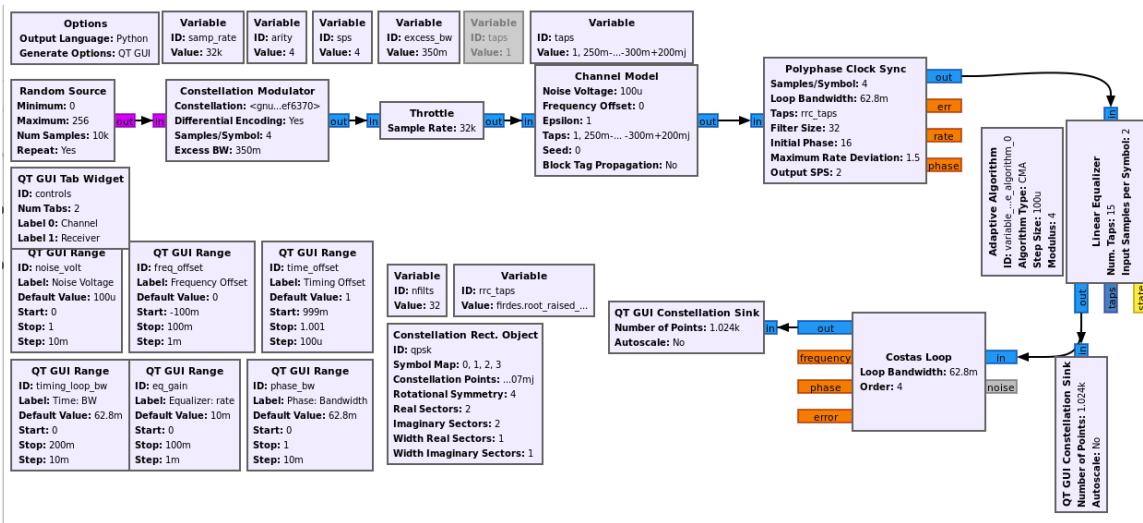


Ilustración 34. Flujograma linear equalizer con Adaptive Algorithm y costas loop.

La combinación de un ecualizador lineal adaptativo y un bucle Costas se puede utilizar para mejorar el rendimiento de un sistema de comunicación digital que utiliza modulación de amplitud en cuadratura (QAM).

Al combinar el ecualizador lineal adaptativo y los bloques de bucle Costas en GNU Radio, la señal QAM recibida se puede corregir tanto en amplitud como en distorsión de fase causada por el canal de comunicación. El ecualizador lineal adaptativo corrige las distorsiones de amplitud y fase debidas a los efectos de desvanecimiento y trayectos múltiples, mientras que el costas loop corrige los errores de fase causados por el desplazamiento de la portadora y los errores de temporización del símbolo.

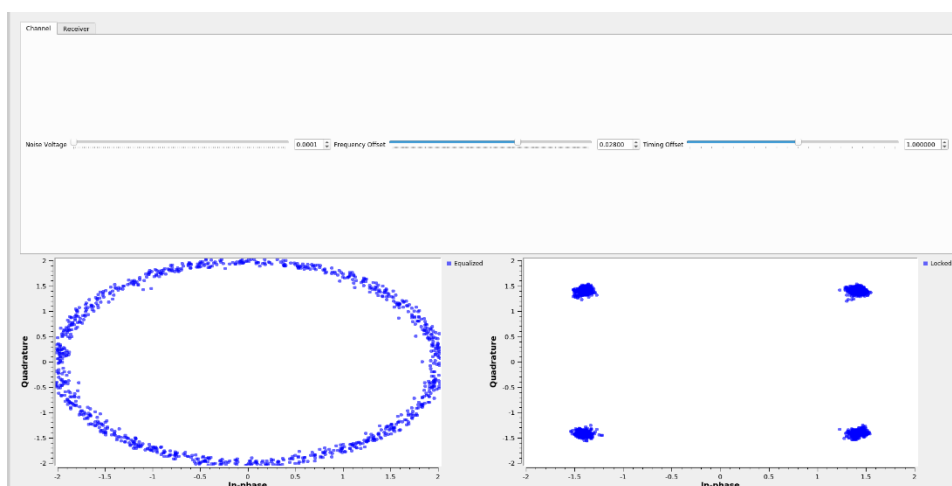


Ilustración 35. Simulación linear equalizer con Adaptive Algorithm y costas loop.

6. M-aria: Para el caso del presente trabajo, se tiene una modulación M-aria la cual es la modulación QPSK, donde se denominaría una M-aria de 4 niveles ya que la modulación trabajada anteriormente, cuenta con 4 símbolos, los cuales pueden ser vistos en el diagrama de constelación que se encuentra en la **Ilustración 20**. Si se desea incrementar los niveles en la modulación M-aria, también de debe incrementar el número de símbolos, siendo estos valores del mismo valor. Este valor se va a obtener como potencias de 2, donde va a depender del número de bits k:

$$M = 2^k$$

Cuando se trabaja con un mayor número de bits, se disminuye la probabilidad de error a la hora de hacer una comunicación digital, es decir, se hace más eficiente el envío de información del transmisor al receptor.

6.1. M-aria de 8 Niveles: Para la obtención de los ocho niveles de esta modulación, es necesario aumentar el número de bits, con el fin de que la potencia de dos sea igual a ocho:

$$M = 2^3 = 8$$

Ya con el número de bits (k = 3) y con el número de símbolos (M = 8), se procede a darle una ubicación en el diagrama de constelación, por medio del bloque **Constellation Rect. Object**, en el cual se agregaron el número de símbolos y la ubicación de estos en el diagrama de constelación. Cabe recalcar que el flujograma base para realizar estas modificaciones fue el de la modulación QPSK. La pantalla de la simulación después de las variaciones es la siguiente:

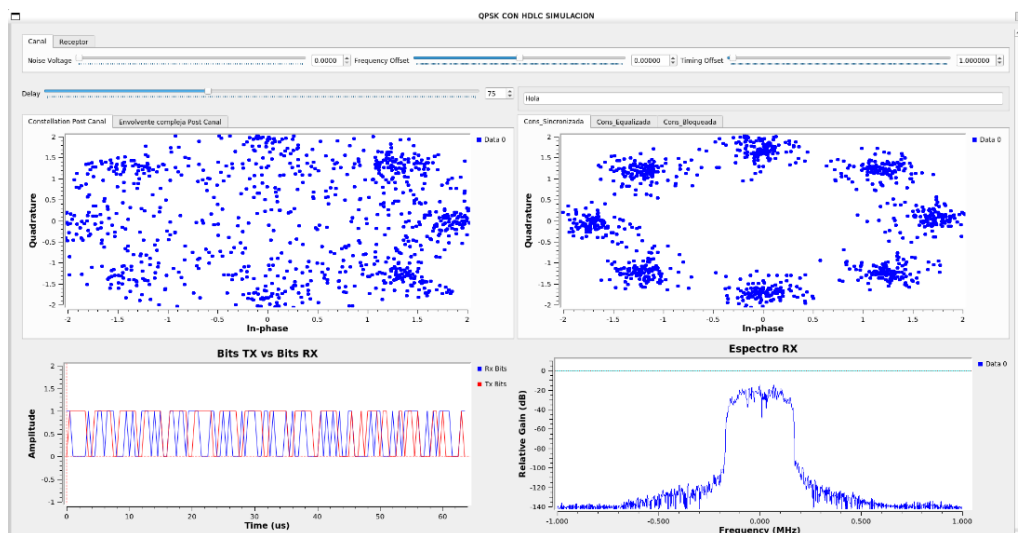


Ilustración 36. Sistema de Modulación M-aria de 8 niveles.

En la anterior figura se pueden apreciar los ocho símbolos en el diagrama de constelación, confirmándose que es una M-aria de 8 niveles.

6.2. M-aria de 16 Niveles: De igual manera, se busca la cantidad de bits que se requieren, siendo $k=4$ y el número de símbolos de 16.

$$M = 2^4 = 16$$

El número de símbolos es de 16, esta cantidad es la que tiene que aparecer en el diagrama de constelación a la hora en que se simule.

Es importante que tanto la parte de TX y de RX cuenten con la misma cantidad de símbolos, por lo que la inserción de los símbolos y de su posición se debe hacer para ambos flujogramas.

7. Conclusiones:

- A la hora de que se vaya a realizar la transmisión de un mensaje, es importante que tanto la parte de TX y RX cuenten con los mismos parámetros como la frecuencia central, el número de símbolos, entre otros, con el fin de que la comunicación entre ambas partes sea lo más eficiente posible.
- Cuando se cambió el enlace entre TX y RX de un enlace cableado a un enlace inalámbrico, se pudo notar que en el inalámbrico existía un poco más de ruido y distorsión, esto a causa de los obstáculos que se le presentaron a la señal a la hora de ser transmitida.
- En general, el bloque de Polyphase Clock Sync es una herramienta importante para la sincronización de tiempo de símbolos en sistemas de comunicación digital, y se usa ampliamente en diversas aplicaciones, como radio definida por software, comunicación inalámbrica y procesamiento de señales digitales.
- El ecualizador lineal es una herramienta importante para mejorar el rendimiento de los sistemas de comunicación digital al reducir los efectos de las deficiencias del canal en la señal transmitida.
- El uso de un ecualizador lineal adaptativo puede mejorar el rendimiento de los sistemas de comunicación digital ajustando continuamente la ecualización para que coincida con las características cambiantes del canal, lo que da como resultado una mejor calidad de la señal y un mejor rendimiento del sistema.
- En pocas palabras el uso de un ecualizador lineal adaptativo con costas loop puede mejorar significativamente el rendimiento de los sistemas de comunicación digital que utilizan QAM al mitigar los efectos de las deficiencias del canal, mejorar la recuperación en cuestión de tiempo de los símbolos y mejorar la calidad de la señal recibida.

- Es importante que el mensaje que vaya a ser transmitido tenga una longitud amplia de caracteres, esto con el fin de que el ruido no lo afecte con mucho impacto. Un mensaje con pocos caracteres es más propenso a que se le sumen interferencias y se dificulte mucho más la comunicación entre el TX y el RX.

REFERENCIAS

GNURadio. (2018). Gnuradio.org. Online Available:

https://wiki.gnuradio.org/index.php/Main_Page