Comunicaciones Inalámbricas

Laboratorio Final - Curso 2022

Parte I: Sistema Completo en Funcionamiento.

1. Introducción

En este laboratorio probaremos nuestro par receptor/transmisor en la práctica. Es decir, utilizaremos un equipo bladeRF o Adalm-Pluto para transmisión y un dongle RTL-SDR como receptor. Es importante tener claro que estos equipos son relativamente frágiles y caros, por lo que hay que tener ciertos cuidados al manejarlos:

1. NUNCA conectar una entrada directamente a una salida. Semejantes niveles de señal a la entrada del equipo pueden quemarlo. El HackRF sólo tiene un puerto que oficia de entrada y salida, pero existen otros equipos con más de un puerto (USRP, bladeRF, Adalm-Pluto), por lo que esta recomendación será válida cuando trabaje con ellos durante el trabajo final.

2. NUNCA transmitir sin haber previamente conectado una antena en el canal de transmisión. La onda reflejada por semejante "mismatch" de impedancias puede generar daños en el equipo. Como precaución, siempre conviene tener una antena conectada a los puertos de transmisión y recepción, aunque no transmitamos.

3. El equipo no tiene llave de encendido, por lo que no hay que olvidar desenchufarlo cuando lo vayamos a dejar de usar durante un tiempo prolongado. Para el caso del bladeRF conviene desconectar el cable USB por el lado de la PC dado que el conector USB de la placa bladeRF ha demostrado ser frágil.

La idea en este laboratorio es basarnos en lo ya desarrollado en los anteriores, por lo que será mucho menos guiado que estos últimos. En particular, a continuación se especifican dos transmisores posibles, ambos con modulación DQPSK. A partir de ellos, deberá implementar el receptor correspondiente. Como se detalla más adelante, transmitiremos imágenes y música. En el caso de la música, su implementación se considerará exitosa si logra escucharla perfectamente, para el caso de la imagen se computará una métrica para establecer la calidad de la misma en recepción.

2. Transmisor de audio

En la figura 1 se muestra el transmisor de audio. La mayoría de bloques ya fueron comentados en laboratorios anteriores, e incluso se utilizará la misma constelación. Vale recordar que para fijar el filtro SRRC se utilizará la sentencia `firdes.root_raised_cosine(samp_per_sym, samp_per_sym, 1.0, alfa, samp_per_sym*len_sym_srrc)`.

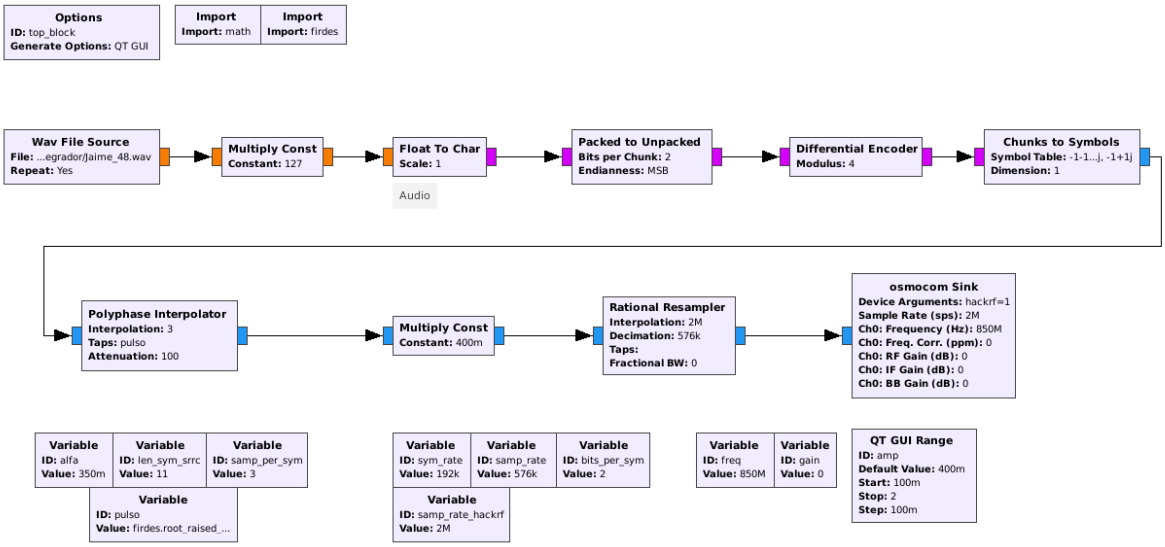

Figura 1: Esquema del transmisor de audio

La parametrización difiere levemente de la utilizada hasta ahora, pero cambiará únicamente la ganancia del filtro. Su importancia quedará clara en lo que sigue.

Los únicos bloques nuevos son el `Multiply Const` y el `Osmocom Sink`. La señal que entra a cualquier equipo debe tener magnitud máxima de 1 para que esté en el rango del conversor DA, lo que justifica el primer bloque. Por tanto, el primer paso antes de transmitir será verificar que se cumpla esto. Verifique que si no utiliza este bloque el espectro de la señal transmitida no será el esperado.

El bloque `Osmocom Sink` es un bloque genérico que representa casi todos los equipos de SDR en modo transmisión, para el caso de Adalm-Pluto se utilizará el bloque `PlutoSDR Sink`. Estos bloques tienen varios parámetros, aunque los de mayor interés para nosotros serán tres:

- `Sample Rate`. Ésta es la tasa a la que el equipo le pasa muestras al conversor DA. Cuidado porque no todos los equipos soportan todas las tasas de muestreo. Por ejemplo, en el caso del HackRF se recomienda trabajar a 2 MSps o superior, y por eso se usó el `Rational Resampler` a la entrada. Es conveniente siempre buscar en la web para el equipo que tengamos cuáles tasas son soportadas (muchos equipos incluso advierten en el terminal si la tasa configurada no es soportada).
- `Ch0: Center Freq` (o `LO Frequency` en PlutoSDR Sink). La segunda es la frecuencia de portadora. Trabajaremos en tres posibles bandas: 550MHz, 850MHz y 1150MHz.
- En el bloque `Osmocom Sink`, `Ch0: RF Gain (dB), Ch0: IF Gain (dB), Ch0: BB gain (dB)`. Estas son tres posibles ganancias que se pueden configurar, dependiendo del equipo que tengamos. En el caso del HackRF, IF Gain puede

variar entre 0 y 47 dB, RF Gain puede ser 0 o 14 dB y BB gain no tiene efecto en modo transmisión (aunque en recepción puede variarse entre 0 y 62 dB). Se recomienda inicializar todos estos valores en 0 para luego ir ajustando las ganancias de a poco. Pruebe utilizar distintos valores para el IF Gain y cómo afecta el espectro. En el caso de bladeRF la ganancia puede tomar valor 0 o mayor que 20. En el caso de Adalm-Pluto, con el `PlutoSDR Sink` la ganancia se puede ajustar con el parámetro `Attenuation (dB)`, que puede tomar valores entre 0 y 89.75 con pasos de 0,25 dB. Tomar en cuenta entonces que para aumentar la ganancia es preciso disminuir este parámetro. Nuevamente, y en general para todos los SDR, es necesario buscar en la web cómo es el mapeo y qué valores son válidos para el equipo que tengamos.

Para el receptor debe considerar que a priori uno no sabe qué lugar ocupa cada bit recibido en el byte que fue transmitido, por lo que deberá implementar un mecanismo de *delay* variable hasta percibir que la señal ha sido ajustada.

3. Transmisor de imágenes

El flowgraph de transmisión (ver Figura 3) para este caso es muy similar anterior, con la diferencia que en lugar de leer muestras de un archivo wav, leemos bytes de una imagen monocromática que se encuentra codificada con 8 bits sin signo, lo que permite representar intensidades de gris entre 0 y 255.

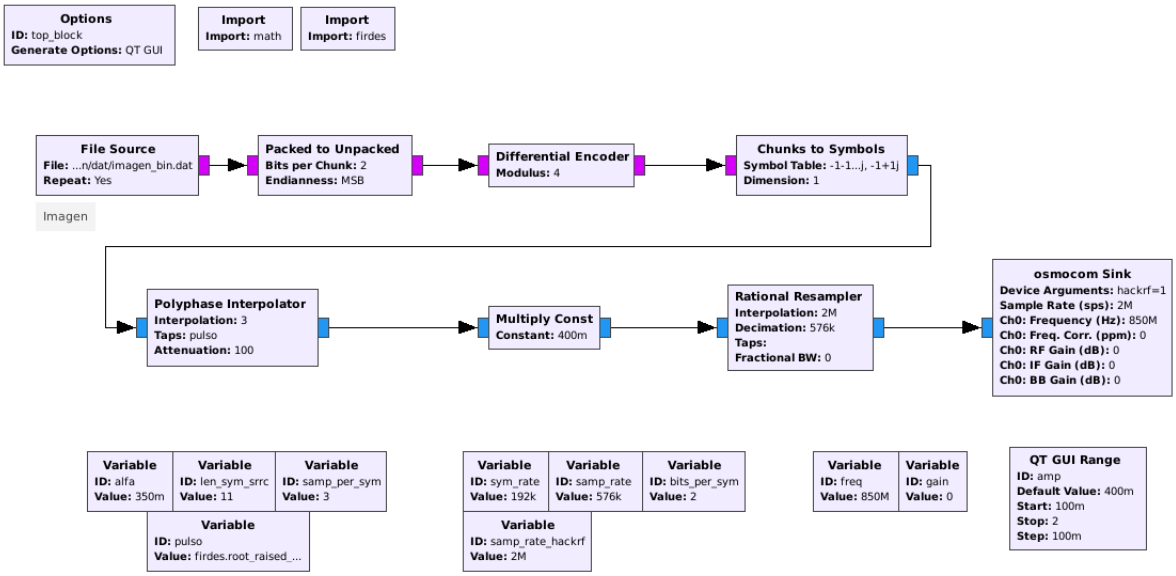

Figura 2: Esquema del transmisor de imágenes

El binario de la imagen se crea a partir del script de python `image2bit` disponible en https://gitlab.fing.edu.uy/comina/lab-integrador. Para crear un binario se

debe ejecutar el siguiente comando en terminal: `python image2bit.py`[1]. Además de convertir una imagen a una secuencia de bytes, este código agrega una secuencia de sincronización al comienzo de la imagen [0,128,255,0,128,255,...,0,128,255] de largo 30 bytes (o píxeles).

El receptor deberá registrar el binario en un archivo .dat utizando un bloque `File Sink`. La decodificación de la imagen se realiza mediante el script `bit2image.py`, verifique que la ruta al binario sea correcta. Para alinear y calcular el BER y el PSNR de la imagen debe colocar la variable `DECODE_IMAGE` en `True`.

Dadas dos imágenes con sus matrices I y K, ambas de tamaño $M \times N$, se define el Mean Square Error (MSE) y el Peak Signal-to-noise Ration measure (PSNR) como:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} ||I(i,j) - K(i,j)||^2$$

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right)$$

Figura 3: Imagen a transmitir

4. Tareas

El objetivo será entonces implementar un receptor para cada transmisor. Recuerde que a diferencia de lo realizado hasta ahora, deberá agregar al menos tres bloques importantes. En orden, éstos serán:

[1]Puede cambiar la imagen `image.png` que viene por defecto por otra siempre y cuando esté en modo escala de gris y tenga las mismas dimensiones.

- `Osmocom Source`, que se encargará de bajar la señal a banda base, filtrar y muestrear. Para recibir señales contará con un RTL-SDR, que responde únicamente al valor de RF Gain (entre 0 y 50 dB). En este caso, la tasa de muestreo puede estar entre 255-300 kSps o 900-3200 kSps, aunque más allá de aproximadamente 2400 hsps puede empezar a perder muestras.
- Un Automatic Gain Control (`AGC`) para mantener la amplitud media de la señal compleja en un valor determinado. GNU Radio ofrece varias posibilidades para este bloque.
- Si tiene un problema de corrimiento de frecuencia puede utilizar un bloque que se encargue de una primera alineación en frecuencia (variaciones grandes que el `Costas Loop` no pueda seguir). Le recomendamos que utilice un `FLL Band-Edge`, que sirve para señales conformadas con un pulso SRRC. El único parámetro ambiguo es el denominado `Prototype Filter Size`, donde debe incluirse el largo del filtro SRRC utilizado en transmisión. Por más detalles, consultar la sección 3.5.5 del libro de Mengali *et al.*

5. Comentarios finales

Por último, un par de comentarios de orden práctico:

- En algunas instalaciones de audio puede tener un comportamiento un tanto "errático". Una solución que muchas veces funciona es poner `plughw:0,0` en el parámetro `Device Name` del `Audio Sink`.
- Un comando rápido y sencillo para analizar el espectro con cualquier equipo SDR es `osmocom_fft` (no aplica para Adalm-Pluto).

6. Informe

El informe debe contener:

1. El espectro de la señal transmitida (visto en recepción) según el re-escalado del bloque `Multiply Const`. Justifique.
2. El diseño de los dos receptores, justificando la elección de los bloques, posición en el diagrama y parametrización.
3. Análisis de lo que se obtiene con las distintas ganancias y frecuencias de transmisión. Incluya pros y contras de cada una, y su elección final (que naturalmente tendrá que estar fundamentada).
4. ¿Qué alcance tiene su transmisor? Aleje el receptor hasta que considere que la calidad del audio o de la imagen es demasiado pobre. Estime el SNR, y en el caso de la imagen registre además el PSNR y el BER. ¿A qué probabilidad de error corresponde?¿Cambia el alcance con la frecuencia?

5. ¿Se le ocurre alguna modificación al transmisor/receptor para aumentar el alcance (sin modificar ganancias o antenas)? Comente ventajas y desventajas del mismo.

6. Ahora modifique el sistema para transmitir y recibir MPSK. ¿Cuál es el máximo M que el sistema soporta?

Para la segunda parte del laboratorio cada grupo podrá elegir entre i) implementar una transmisión OFDM y ii) modificar el sistema de transmisión incorporando alguna técnica de corrección de errores.