

# Comunicaciones Inalámbricas

## Laboratorio Final - 2024

### Parte I - Sistema Completo en Funcionamiento

## 1. Introducción

En este laboratorio probaremos nuestro par receptor/transmisor en la práctica. Es decir, utilizaremos dos equipos SDR, cada uno conectado a un PC, oficiando uno de receptor y otro de transmisor. 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.
2. NUNCA transmitir sin haber previamente conectado una antena a la salida. 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 RF1 y RF2, 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.

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 música. Su implementación se considerará exitosa si logra escucharla perfectamente.

## 2. Transmisión sin entramado

En la figura 1 se muestra el primer transmisor. 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)`. 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 `PlutoSDR Sink`. La señal que entra al SDR debe tener magnitud adecuada para que esté en el rango del convertor DA, lo que justifica el primer bloque. Por tanto, el primer paso antes de transmitir

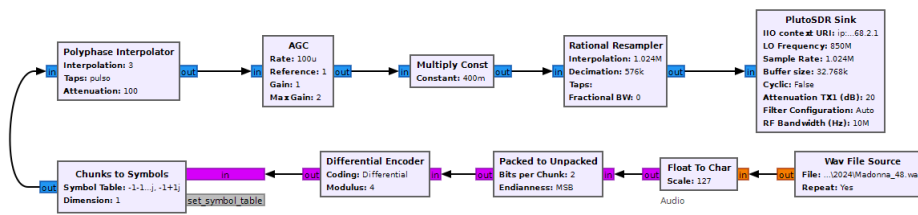


Figura 1: Esquema del transmisor sin entramado

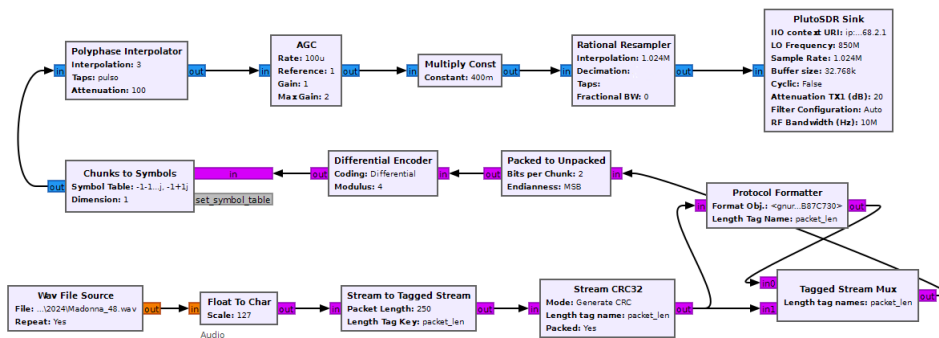


Figura 2: Esquema del transmisor con entramado

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 PlutoSDR Sink contiene varios parámetros.

- **Samp Rate.** Ésta es la tasa a la que el SDR le pasa muestras al convertor DA. Puede ser necesario **Rational Resampler** si la fuente genera muestras a otra tasa.
- **LO Frequency.** La segunda es la frecuencia de portadora. Trabajaremos en tres posibles bandas: 550MHz, 850MHz y 1150MHz.
- **Attenuation TX1.** Determina la potencia de transmisión, atenuando sobre la potencia máxima.

### 3. Transmisión con entramado

El segundo transmisor es esencialmente igual al anterior, al menos a nivel de modulación (ver figura 2). La diferencia reside en los bloques **Stream To Tagged Stream**, **Stream CRC32**, **Tagged Stream Mux** y **Protocol Formatter**. Estos bloques generan tramas de largo fijo con un encabezado que se transmite al receptor y le permite a este la decodificación y entrega de los datos de información correctamente. Ver la referencia GNURadio Packet Communications.

Considerar que para posibilitar el entramado es necesario transmitir información adicional (overhead) en el aire, además de los datos útiles (payload). Tome en cuenta esto a la hora de asignar el valor de las tasas de símbolos y muestreos en los bloques necesarios. Experimente con diferentes valores de largo de trama y evalúe beneficios y costos.

## 4. Transmisor de imágenes

El flowgraph de transmisión para este caso es muy similar anterior, con la diferencia que en lugar de leer muestras de un archivo wav, leemos desde un archivo 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.

El archivo 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.py1`. 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 debería registrar el payload recibido en un archivo .dat utilizando 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:

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

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

## 5. Otras pruebas

Por último, haremos dos experimentaciones empíricas:

- ¿Qué alcance tiene su transmisor? Aleje el receptor hasta que considere que la calidad del audio es demasiado pobre. Estime el SNR. ¿A qué probabilidad de error corresponde?
- Ahora modifique el sistema para transmitir y recibir MPSK. ¿Cuál es el máximo  $M$  que el sistema soporta incluso en condiciones ideales?

## 6. Comentarios finales

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:

- **PlutoSDR Source**, que se encargará de aplicar un filtro pasa-banda, bajar la señal a banda-base y muestrear.
- Un *Automatic Gain Control* (AGC) para mantener la amplitud media de la señal compleja en un valor fijo determinado. GNURadio ofrece varias posibilidades para este bloque.
- Un bloque que se encargue del alineación en frecuencia. Le recomendamos que utilice el **FLL Band-Edge**, que sirve para señales conformadas por 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 3.5.3 del libro de Mengali *et al.*

Por último, un par de comentarios de orden práctico:

- En algunas instalaciones el 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**.
- Es posible que **Symbol Sync** funcione mejor que **Polyphase Clock Sync**. Debe explicar su funcionalidad completamente si los usa.

## 7. Informe

El informe debe contener:

1. El espectro de la señal transmitida con y sin el re-escalado. Justifique.
2. El diseño de los dos receptores, justificando la elección de los bloques, posición en el diagrama y parametrización.
3. Un análisis de pros y contras de los dos modos de transmisión, tomando en cuenta las particularidades de la implementación de GNURadio.
4. 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).
5. ¿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?
6. ¿Se le ocurre alguna modificación al transmisor/receptor para aumentar el alcance (sin modificar ganancias o antenas)? Comente ventajas y desventajas del mismo.
7. Resultados sobre el máximo tamaño de la constelación y su respectivo análisis.