

Práctica No.2: PSD DE SEÑALES ALEATORIAS (GNURADIO)

Juan Pablo Gamboa Durán - 2210416
Didier Julián Moreno Ortiz - 2202932
Karla Vanessa Ruíz González - 2202808

https://github.com/JuKevCom/Comu2_C1_judika/tree/Practica_2/Practica_2

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones
Universidad Industrial de Santander

13 de Octubre de 2024

Abstract

This report presents the analysis of the Power Spectral Density (PSD) of random signals using GNURadio as a simulation environment. The study focused on the spectrum representation of real-world signals, such as audio and images, as well as artificial signals. A number of characteristics were examined in order to determine how they affected spectral resolution, including bit rate and samples per symbol (Sps). Furthermore, the behaviour of the PSD for Gaussian white noise and binary signals was noted, emphasising the connection between the frequency domain energy distribution and interpolation.

Palabras clave: Señales aleatorias, muestras por símbolo, interpolación, rata de bits, densidad espectral de potencia

I. INTRODUCCIÓN

El análisis de señales aleatorias permite caracterizar el comportamiento de sistemas que operan bajo condiciones de ruido o señales impredecibles. En este contexto, la Densidad Espectral de Potencia (*Power Spectral Density*, PSD) se erige como una herramienta esencial, proporcionando una representación precisa de cómo se distribuye la energía de una señal en el dominio de la frecuencia [1]. El presente laboratorio se enfoca en el estudio de la PSD utilizando GNURadio, un entorno creado para la simulación y análisis de señales en sistemas de comunicación [2]. Con esto se busca observar cómo diversas señales, tanto generadas artificialmente como provenientes del mundo real, se manifiestan en el dominio de la frecuencia y cómo la variación en los valores de Sps influye en la resolución y calidad de las mismas.

II. METODOLOGÍA

Primero, se verifica el funcionamiento del flujograma presentado en la Figura 1 mediante el análisis de una

señal binaria aleatoria bipolar de forma rectangular. Para esto, se obtienen su representación en el dominio temporal, la densidad espectral de potencia (PSD) y los parámetros principales como la rata de bits, la frecuencia de muestreo y el ancho de banda, para diferentes valores de Sps.

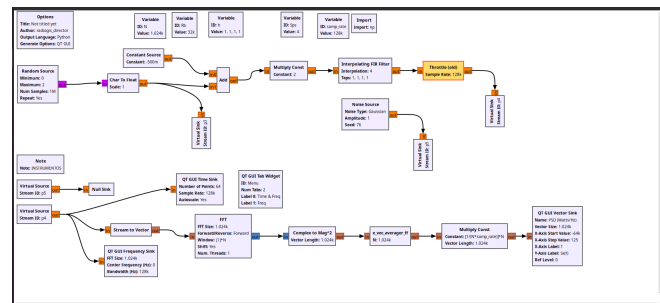


Figura 1. Flujograma base de la práctica.

Luego, se ajustan las Virtual Source del flujograma para permitir el análisis del ruido blanco gaussiano tanto en el dominio temporal como en el espectral, realizando diversas pruebas variando los valores de Sps.

Posteriormente, se estudia cómo varía la señal cuando los bits provienen de fuentes reales, como una cámara fotográfica o un micrófono. Para ello, se reemplaza el bloque Random Source del flujograma por los bloques que permiten leer un archivo de imagen, configurando el File Source para cargar la imagen de una rana (Figura 2) y realizar pruebas de la señal en el dominio de tiempo y el espectro. Posteriormente, el mismo procedimiento se aplica con un archivo de audio, obteniendo y analizando los resultados correspondientes.

Finalmente, se responde una serie de preguntas de control sobre el flujograma, enfocadas en analizar el papel de los diferentes bloques y comprender cómo los cambios en sus parámetros afectan la forma de la señal.

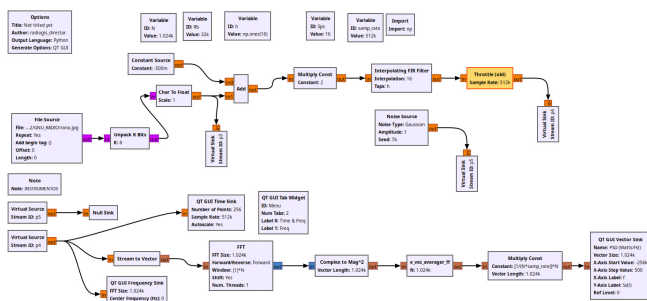


Figura 2. Fugograma con un archivo de imagen como fuente.

III. RESULTADOS

Durante la fase inicial, se buscó comprobar el funcionamiento del flujograma propuesto para la práctica (Figura 1). Para esto, se analizaron diversos parámetros para algunos valores básicos de Sps (Cuadro I).

Cuadro I
Parámetros Principales

Sps	R_b [bit/ms]	f_s [kHz]	B [Hz]
1	32	32	∞
4	32	128	63621,7
8	32	256	63499,9
16	32	512	63615,3

Del cuadro I, se puede deducir que los valores R_b y Sps son independientes entre sí, lo cual tiene sentido, puesto que el *bit rate* es un valor constante que representa la velocidad de transmisión de bits, mientras las Sps determinan la cantidad de muestras a tomar para la representación de cada símbolo. Además, se logra evidenciar la relación 1 en los datos y un comportamiento similar al ruido gaussiano en cuanto al ancho de banda para Sps de 1 (Figura 3).

$$f_s = Sps \cdot R_b \quad (1)$$

Habiendo concluido ya el breve acercamiento al flujograma, se analizó el comportamiento en tiempo y frecuencia del ruido gaussiano, realizando las respectivas modificaciones a la instrumentación. Tras tal cambio, se obtuvieron dos gráficas de tiempo y PSD, una para $Sps = 1$ (Figura 4) y otra para $Sps = 16$ (Figura 5).

De las anteriores gráficas se puede inferir que existe una relación entre la Sps y la potencia de la señal porque, aunque ambas poseen un ancho de banda infinito, su valor de PSD es una constante que disminuye conforme el valor de Sps aumenta. La temática vista en clase confirma la existencia teórica de dicha relación, puesto que la potencia de la PSD es inversamente

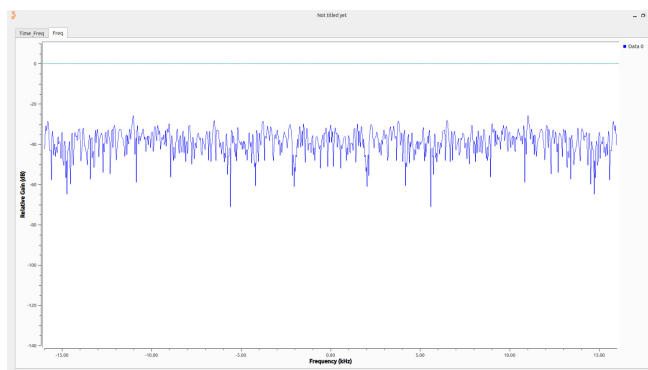


Figura 3. Espectro para $Sps=1$.

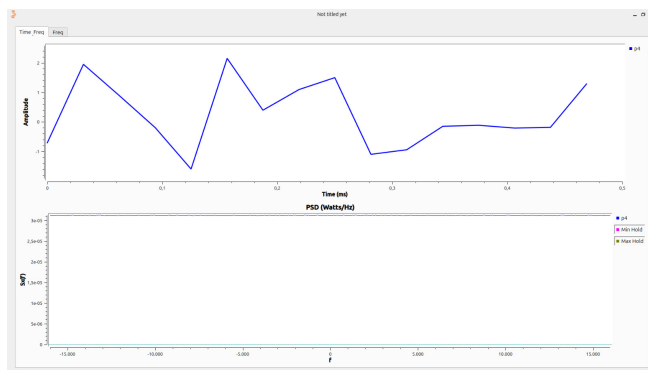


Figura 4. Gráficas para ruido gaussiano con $Sps = 1$.

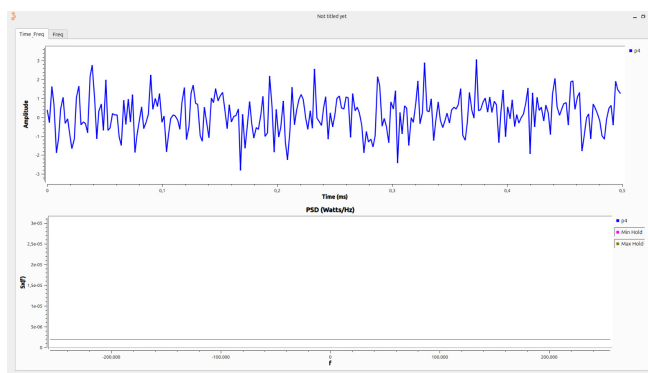


Figura 5. Gráficas para ruido gaussiano con $Sps = 16$.

proporcional a f_s , y f_s es directamente proporcional al valor de Sps .

De forma similar, una señal binaria aleatoria de forma rectangular tiene un ancho de banda infinito debido a sus transiciones abruptas (Figura 6). Sin embargo, en la práctica, el ancho de banda está limitado por la resolución del sistema y las frecuencias de muestreo utilizadas. En GNURadio, la PSD de una señal binaria

aleatoria muestra una extensión amplia, aunque no infinita, debido a las limitaciones inherentes al procesamiento de datos y técnicas de muestreo.

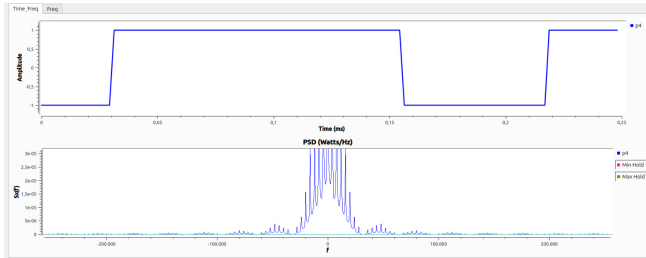


Figura 6. Gráfica para señal cuadrada.

Tras tal análisis, y con las configuraciones adecuadas al flujograma, se desarrolló el mismo proceso pero para señales del mundo real, en este caso, una imagen y un audio (Cada uno de manera individual).

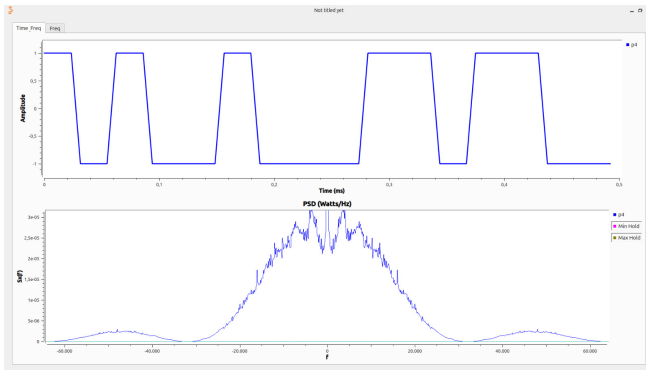


Figura 7. Gráficas para imagen con $Sps = 4$.

En la figura 7 se observa una forma de PSD suave y distribuida, con la mayor parte de la potencia concentrada en frecuencias más bajas. Este tipo de PSD es típica de imágenes naturales, donde los cambios suaves en la intensidad (como en regiones suaves de una imagen) dominan sobre los detalles más finos. De esto se deduce que las imágenes tienden a tener más energía en las bajas frecuencias espaciales (suavidad) y menos en las altas frecuencias (detalles).

En la figura 8 se puede apreciar una PSD con múltiples picos en frecuencias específicas, estos picos se corresponden con los armónicos de la señal, ya que una onda cuadrada contiene múltiples armónicos de la frecuencia fundamental, en este caso, se encuentra centrada alrededor de 0 Hz y se ven muchos picos a lo largo del espectro (Esto es consistente con la naturaleza de la onda cuadrada, que tiene componentes en una serie de armónicos impares que decaen en magnitud a medida que la frecuencia aumenta).[3]

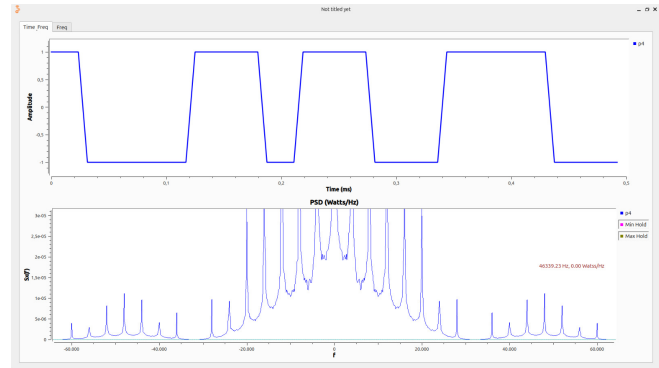


Figura 8. Gráficas para audio con $Sps = 4$.

Para poder analizar los resultados obtenidos, es importante entender el funcionamiento del diagrama de bloques que se está implementando. En la Figura 9, se presenta un diagrama compuesto por tres bloques conectados entre sí. Estos bloques se encuentran al principio del proceso y desempeñan un papel crucial, ya que permiten convertir la señal de entrada en una señal binaria bipolar.

El proceso de conversión se realiza en primer lugar usando el bloque, *Constant Source*, el cual genera una constante de valor -0.5 . Esta señal constante pasa luego por el bloque *Add*, cuya función es combinar la señal de entrada con la constante mencionada, generando un nivel de offset en la señal. Por último, el bloque *Multiply Constant* multiplica el valor de la señal que recibe por un factor de 2, esto ajusta la señal de salida, para dejarla en un rango entre -1 y 1 . De esta manera se logra obtener una señal binaria bipolar de manera rápida y efectiva.



Figura 9. Descripción de la imagen.

Otro bloque fundamental para el desarrollo de esta implementación fue el bloque *Interpolating FIR Filter*, el cual aumenta la tasa de muestreo de una señal, obteniendo más muestras en un intervalo de tiempo. El funcionamiento de este bloque se basa en dos principios: el proceso de interpolación, que permite insertar más muestras entre las muestras originales de la señal, y el proceso de filtrado, utilizando un filtro de respuesta finita al impulso (FIR). Este filtro se encarga de suavizar la señal interpolada, asegurando que las muestras adicionales no introduzcan artefactos o distorsiones, lo que resulta en una señal más fluida.

En el bloque *Interpolating FIR Filter*, se utilizó el

parámetro *Interpolation* tomando como referencia el valor de *Sps*. Esto se debe a que *Sps* define las muestras por símbolo, y al usar este valor se asegura que se generen suficientes muestras por símbolo para una correcta representación de la señal. Cambiar este valor afectaría la cantidad de muestras por símbolo, lo que podría tener efectos positivos, como una mayor resolución temporal, o efectos negativos, como la degradación en la calidad de la señal debido a una representación insuficiente de los símbolos.

El bloque *Throttle*, limita la tasa a la que se procesan los datos para evitar sobrecargar el sistema o una pérdida de datos y para garantizar una emulación precisa de condiciones de transmisión en tiempo real.

Si la señal binaria no se convierte a una señal bipolar (manteniendo valores de 0 y 1 en lugar de -1 y 1), la energía en la PSD de la señal estaría más concentrada en las frecuencias bajas, lo que disminuiría la eficiencia espectral. Además, el contenido espectral sería menos amplio, ya que la alternancia de niveles en una señal bipolar distribuye la energía de manera más equitativa.

El número de lóbulos de la PSD de una señal binaria aleatoria rectangular puede calcularse utilizando la siguiente fórmula:

$$N = 1 + 2 \cdot Sps \quad (2)$$

Donde N es el número total de lóbulos y Sps es el número de muestras por símbolo. Cada símbolo contribuye con un lóbulo central y dos lóbulos laterales, y el lóbulo central cuenta como dos debido a su mayor ancho.

En relación al bloque *Unpack K Bits*, si el valor de K se configura como 16, esto afectaría la cantidad de bits desempaquetados en cada iteración, lo que impactaría directamente en la resolución temporal de la señal. Un mayor valor de K resulta en una menor cantidad de símbolos procesados por unidad de tiempo, mientras que valores menores de K permitirían mayor detalle temporal. En la PSD, un mayor valor de k se vería como una mayor cantidad de picos de frecuencia (Figuras 10 y 11), mostrando de esta manera, una mejor distribución de la potencia.

Para calcular la frecuencia de muestreo a la entrada del bloque *Unpack K Bits* en función del número de lóbulos de la PSD y el ancho de banda de la señal, se puede utilizar la siguiente relación:

$$B \times \frac{\text{número de lóbulos}}{2} = \frac{f_s}{2}$$

Donde B es el ancho de banda de la señal y f_s es la frecuencia de muestreo.

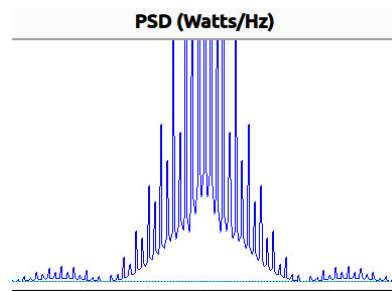


Figura 10. Gráficas para $k=8$.

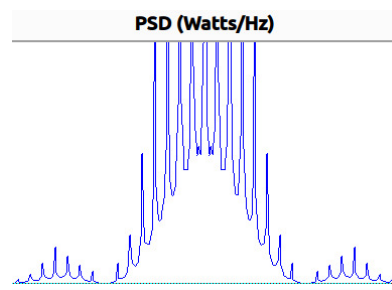


Figura 11. Gráficas para $k=16$.

En cuanto a la salida del bloque *Char to Float*, la frecuencia de muestreo a la salida permanece igual a la frecuencia de muestreo a la entrada, ya que este bloque solo realiza una conversión de formato sin modificar la tasa de muestreo.

También se determinó que la PSD de una señal binaria aleatoria bipolar es más similar a la PSD del ruido blanco cuando el valor de Sps es igual a 4 (Figura 12).

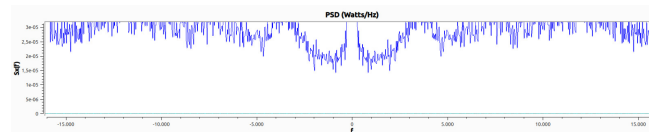


Figura 12. PSD para $Sps=4$.

Finalmente, se realizaron reconfiguraciones tanto al vector h como a *multiply constant* dentro del flujo grama, con la finalidad de generar distintas formas de onda en el tiempo (Cuadro II).

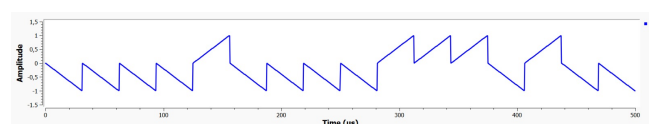


Figura 13. Gráfica en tiempo para Dientes de Sierra.

