

PSD DE SEÑALES ALEATORIAS (GNURADIO)

David Josué Díaz Ortiz, 2204269, Estudiante Ing. Electrónica,
Duban Yesid Cortes Tabares, 2214644, Estudiante Ing. Electrónica

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

Septiembre 20, 2025

https://github.com/David2204269/CommII_LabB1_G4.git

Abstract

This laboratory practice addresses the application of Power Spectral Density (PSD) for the analysis of random signals using GNU Radio. The main objective is to examine the behavior of random signals through their time-domain representation and spectral characteristics. The experiment involves the generation of random binary sequences, noise characterization, and the extraction of fundamental parameters such as bit rate, bandwidth, and sampling frequency. By configuring different signal models and analyzing their corresponding PSD, this practice enhances the understanding of signal processing principles within communication systems. Furthermore, it highlights the relevance of PSD analysis in practical applications, including audio and image processing.

Keywords: Power Spectral Density (PSD), Random signals, GNU Radio, Noise analysis, Bandwidth.

1 Introducción

Este laboratorio tiene como propósito el estudio y la estimación de la Densidad Espectral de Potencia (PSD), una herramienta fundamental en el análisis de señales aleatorias y en el diseño de sistemas de comunicaciones. La PSD describe cómo se distribuye la potencia de una señal a lo largo del espectro de frecuencias, lo que permite identificar su ancho de banda, componentes dominantes y comportamiento espectral.

Para alcanzar este objetivo, se utilizó la plataforma GNU Radio[1], la cual proporciona un entorno basado en diagramas de bloques que facilita la implementación de técnicas de procesamiento digital de señales (DSP). En la práctica se analizaron señales generadas artificialmente, como señales binarias aleatorias bipolares y ruido blanco, así como señales provenientes de fuentes reales, entre ellas archivos de audio e imágenes.

Durante el desarrollo, se configuraron parámetros clave como la frecuencia de muestreo, la tasa de bits (Rb) y los Samples per Symbol (Sps), lo que permitió observar el efecto de estos factores sobre la representación temporal y sobre la PSD. Además, se emplearon bloques de procesamiento como el Throttle, el Interpolation FIR Filter y la Transformada Rápida de Fourier (FFT) [2], junto con herramientas de visualización en tiempo real para comparar el dominio temporal y frecuencial. Los resultados obtenidos evidenciaron cómo las variaciones en los parámetros modifican directamente la PSD, resaltando su importancia en el análisis y diseño de sistemas de comunicación digitales, así como su aplicabilidad en las telecomunicaciones modernas.

2 Metodología

Para el desarrollo del laboratorio, se empleó el flujograma previamente suministrado para esta práctica, el cual se presenta a continuación:

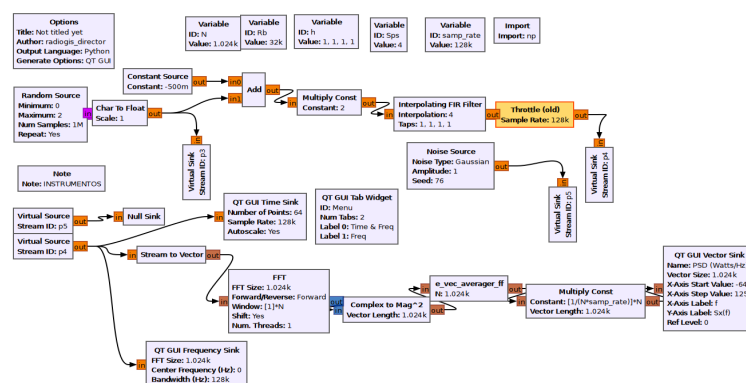


Fig. 1: Flujograma GNU Radio utilizado.

En primer lugar, fue fundamental comprender el dia-

grama de flujo base, ya que a partir de este se resolvieron las preguntas de control del inciso 6 de la guía y se orientó el desarrollo del laboratorio. Este flujograma sirvió como apoyo durante todo el procedimiento y facilitó la interpretación de los resultados.

Una vez comprendido el funcionamiento del flujograma, se abordó el punto 2 de la guía, orientado al análisis de una señal binaria aleatoria bipolar de forma rectangular. En este apartado se trabajó con el parámetro Samples per Symbol (Sps), el cual define cuántas muestras representan a cada símbolo en la señal digital. Se realizaron pruebas con valores de Sps = 4, 8, 16 y 1, con el objetivo de observar cómo estos afectan la forma de la señal en el tiempo y su correspondiente densidad espectral de potencia. Para que el bloque pudiera adaptarse a los cambios en Sps, fue necesario modificar la variable h , de manera que dependiera de su longitud ($\text{len}(h)$), garantizando así una coherencia entre muestras y símbolos. Este análisis permitió comprender cómo una mayor o menor cantidad de muestras por símbolo impacta la resolución temporal y la distribución espectral de la señal.

En el tercer punto se estudió el comportamiento del ruido blanco tanto en el dominio temporal como en el espectral. Para ello, se configuraron las “Virtual Source”, asignando el nombre p4 al bloque superior (debajo de instrumentos) y p5 al bloque inferior. Posteriormente, se realizaron varias pruebas que permitieron observar cómo se representa el ruido blanco en el tiempo y cómo se distribuye su energía en la PSD.

Siguiendo con el desarrollo, en el cuarto punto se analizó lo que ocurre cuando los bits provienen de una fuente de datos real, en este caso una imagen. Para realizar esta prueba, se revirtió la configuración del punto anterior y se reemplazó el bloque Random Source por File Source y Unpack K Bits, utilizando el archivo rana.jpg. Con esta modificación fue posible observar el comportamiento de los datos de la imagen en el tiempo y en la frecuencia.

Finalmente, en el quinto punto se realizaron experimentos con una señal de audio capturada mediante un micrófono, utilizando el archivo sonido.wav. Los resultados obtenidos en el dominio temporal y en la PSD fueron registrados y documentados para su posterior análisis.

3 Análisis de resultados

Al obtener la forma aleatoria en el tiempo y la PSD, se caracterizaron las señales calculando sus parámetros habituales (Freq de muestreo, rate de bits, etc). En la imagen se ven los 4 casos desde SPS=1 hasta SPS=16 en dicho orden.

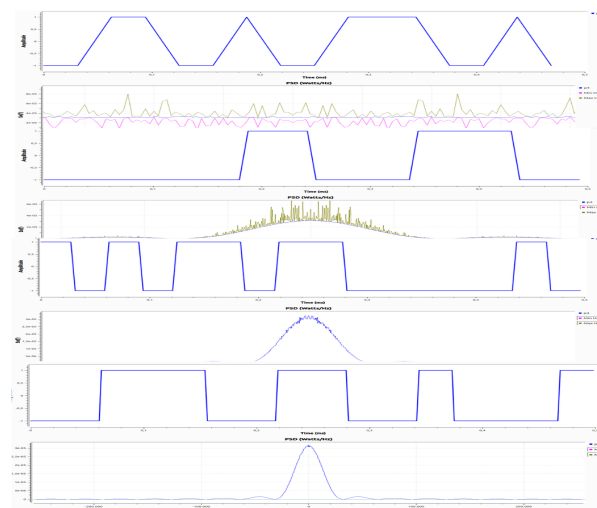


Fig. 2: Señal aleatoria para cada SPS.

Claramente el bit rate es el mismo debido a que lo que varía es la frecuencia de forma proporcional, los resultados se observan en la siguiente tabla:

SPS	Bit rate	Freq. Muestreo	BW	Duración de bit
1	32k	32k	32k	0,03 ms
4	32k	128k	32k	0,07 ms
8	32k	256k	32k	0,035 ms
16	32k	512k	32k	0,06 ms

Tab. 1: Tabla de parámetros

Después se le agregó ruido blanco a la imagen para analizar los efectos sobre las señales debido a este.

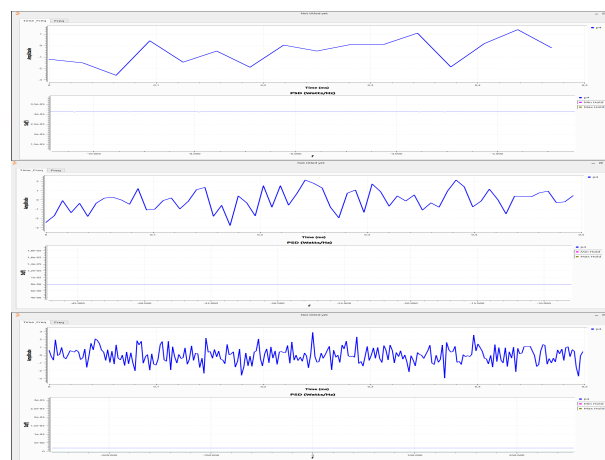


Fig. 3: Grafico en tiempo y frecuencia variando Sps en ruido blanco.

Para analizar el ruido blanco, se configuraron las fuentes virtuales con el bloque superior en **p4** y el inferior en **p5**. Con esta disposición se realizaron pruebas tanto con ruido blanco puro como con señales moduladas a diferentes valores de SPS.

En la figura correspondiente al ruido blanco puro se observa, en el dominio temporal, una señal aleatoria sin estructura definida, que fluctúa alrededor de cero. Esto concuerda con su definición teórica: media nula y varianza finita.

En la representación espectral (PSD), la densidad de potencia se mantiene prácticamente plana a lo largo de todo el espectro. Aunque aparecen pequeñas variaciones por la naturaleza aleatoria de la señal, el comportamiento global confirma que la energía del ruido blanco se distribuye de manera uniforme.

Posteriormente se añadieron modulaciones con distintos SPS. En el tiempo, un mayor número de muestras por símbolo genera una señal más densa y continua, aunque siempre aleatoria. En el espectro, la PSD sigue siendo plana, pero al aumentar SPS crece la frecuencia de muestreo y se amplía el rango de frecuencias visibles. Pese a ello, la potencia por unidad de banda se conserva, reafirmando el carácter de banda ancha del ruido.

Finalmente, se analizaron señales reales como una imagen y un audio para comprobar cómo varía su representación en tiempo y en frecuencia frente al ruido blanco.

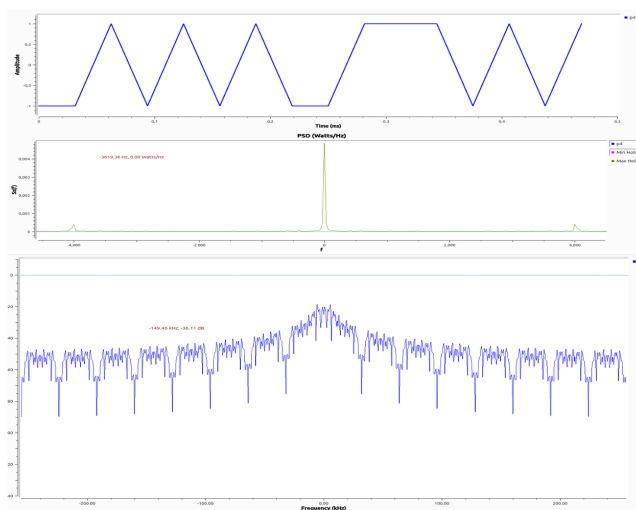


Fig. 4: Respuesta a una imagen en tiempo, frecuencia y ganancia.

Al reemplazar la fuente aleatoria por un archivo real (rana.jpg), se observa en el dominio del tiempo (SPS=1)

una señal digital con transiciones más regulares respecto al ruido, lo que refleja la estructura propia de los datos de la imagen. En la PSD, a diferencia del ruido blanco plano, aparecen picos definidos en torno a la frecuencia central, indicando que la señal no ocupa de manera uniforme todo el espectro, sino que concentra su energía en ciertas bandas. Esto confirma que el contenido de la fuente (imagen) aporta una organización no aleatoria a los bits transmitidos.

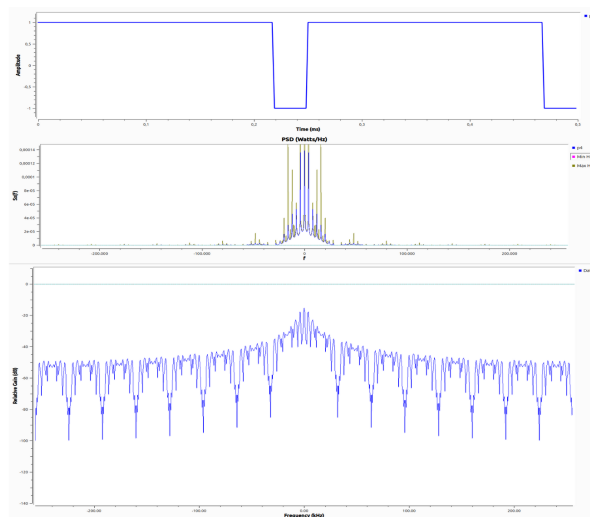


Fig. 5: Respuesta a un audio con SPS=16.

El cambio de la fuente de bits desde una secuencia aleatoria a fuentes reales (imagen y audio) modifica tanto la forma de onda en el tiempo como la distribución de energía en el espectro.

Imagen: señal más estructurada, con espectro concentrado.

Audio: señal más continua, con espectro más amplio y rico en componentes.

4 Preguntas de Control

- ¿Qué papel juega la siguiente combinación de bloques?



Fig. 6: Combinación de bloques de GnuRadio.

La combinación de bloques Char to Float, Add y Multiply Const convierte la señal binaria (0 y 1) en

una señal bipolar (-1 y +1). Este proceso es indispensable para modulaciones digitales que requieren señales balanceadas.

- b. ¿Qué papel juega el bloque “Interpolation FIR Filter”, ¿cómo funciona?

El bloque Interpolation FIR Filter aumenta la tasa de muestreo para asegurar el número de muestras correcto por símbolo. Simultáneamente, un filtro FIR aplica un filtrado para dar forma a los pulsos, controlando así el ancho de banda y la forma de la señal.

1. ¿Por qué el parámetro “Interpolation” en el bloque vale “SPS” y qué pasa si se coloca otro valor?

El parámetro de interpolación debe ser igual a Sps para mantener la relación correcta entre símbolos y muestras. Si los valores no coinciden, la señal se distorsionará en el dominio temporal y su Densidad Espectral de Potencia (PSD) se verá afectada..

2. Si tuviese que analizar la señal en p3, ¿qué cambios realizaría en la instrumentación (esquema de GNU Radio)?

Para visualizar la señal en p3, simplemente cambia la conexión del bloque Virtual Source de p4 a p3. Esto permitirá que la señal de p3 pase por el mismo proceso de análisis que se usa para p4 y p5.

3. ¿Qué fórmula permite conocer el ancho de banda de la señal en p4 si se conoce Rb y Sps?

El ancho de banda de la señal en p4 se puede estimar utilizando la fórmula: Ancho de banda $\approx Rb/SPS$, donde Rb es la tasa de bits y SPS el número de muestras por símbolo.

4. ¿Qué fórmula permite conocer la frecuencia de muestreo en p3, si se conoce la frecuencia de muestreo en p4 y Sps?

La frecuencia de muestreo en p3 se calcula dividiendo la frecuencia de muestreo en p4 entre SPS.

- c. ¿Por qué razón la PSD de las señales binarias que provienen de una señal de audio es diferente a la que proviene de una foto siendo ellas igualmente señales binarias bipolares de forma rectangular?

La diferencia radica en que el audio tiene variaciones dinámicas y temporales, mientras que los datos binarios de una imagen generan patrones estáticos o repetitivos, lo que resulta en una distribución espectral diferente.

- d. ¿Qué papel juega el bloque “Throttle”?

El bloque Throttle controla la velocidad de procesamiento de muestras en GNU Radio, evitando que la CPU se sobrecargue. Además, permite que la simulación se ejecute a una velocidad estable y lo más cercana posible a un sistema físico real.

- e. ¿Qué pasaría con la PSD si no se hace la conversión a señal bipolar, sino que la señal binaria en p4 solo tiene valores de 0 ó 1 en lugar de -1 ó 1?

La PSD mostraría una componente de DC más pronunciada y se alterarían las simetrías espectrales.

- f. Se supone que el ruido blanco tiene un ancho de banda infinito, ¿coincide esto con lo observado en GNU Radio?, ¿por qué?

En GNU Radio no se observa un espectro infinito sino limitado. Esto ocurre por las restricciones de muestreo y el ancho de banda de los bloques de procesamiento.

- g. Se supone que una señal binaria aleatoria de forma rectangular tiene un ancho de banda infinito, ¿coincide esto con lo observado en GNU Radio y por qué?

No. Ya que GNU Radio solo se representa un rango finito debido al muestreo y a las limitaciones del sistema. Esto provoca que los armónicos de alta frecuencia se atenúen y no se visualicen en el espectro.

- h. ¿Qué fórmula podría ayudar a calcular el número de lóbulos de la PSD de señal binaria aleatoria de forma rectangular cuando se conoce la frecuencia de muestreo y Sps?

El número de lóbulos puede estimarse como:

$$N_{\text{lóbulos}} \approx \frac{F_s}{Rb/SPs}$$

considerando que el lóbulo central se cuenta como dos, ya que tiene el doble de ancho que los demás.

- i. ¿Cómo se calcula todo el rango de frecuencias que ocupa el espectro cuando se conoce R_b y S_{ps} ?

El rango espectral se obtiene como:

$$F_s = R_b \times S_{ps}$$

- j. ¿Cómo se calcula la resolución espectral del analizador de espectros, cuando se conoce N y la frecuencia de muestreo?

La resolución espectral está dada por:

$$\Delta f = \frac{F_s}{N}$$

donde N es el número de puntos de la FFT.

- k. ¿Qué pasaría si en el bloque “Unpack K Bits” se configura el parámetro K como 16?

La señal se dividiría en grupos de 16 bits por bloque. Esto reduce la granularidad temporal de los datos y puede modificar su representación, afectando la interpretación de la señal en el análisis.

- l. ¿Cómo calcularía la frecuencia de muestreo a la entrada del bloque “Unpack K Bits” si conoce el número de lóbulos de la PSD y el ancho de banda de la señal?

Se puede estimar aplicando el criterio de Nyquist: la frecuencia de muestreo debe ser al menos el doble del ancho de banda observado y ajustarse al número de lóbulos presentes en la PSD.

- m. ¿Cómo calcularía la frecuencia de muestreo a la salida del bloque “Unpack K Bits” si conoce la frecuencia de muestreo a la entrada?

La frecuencia de muestreo se mantiene constante, ya que este bloque solo reorganiza los datos en paquetes de bits, sin modificar la tasa de muestreo.

- n. ¿Cómo calcularía la frecuencia de muestreo a la salida del bloque “Char to Float” si conoce la frecuencia de muestreo a la entrada?

La frecuencia de muestreo no cambia. El bloque solo transforma el tipo de dato de entero (char) a flotante (float), sin alterar la tasa de muestreo.

- o. ¿Para qué caso de S_{ps} la PSD de una señal binaria aleatoria bipolar es similar a la PSD de ruido blanco?

Cuando el valor de S_{ps} es bajo, la señal tiende a perder correlación entre muestras, resultando en un espectro más uniforme y similar al del ruido blanco.

- p. ¿Qué cambios mínimos haría al flujograma, manipulando principalmente h , si desea que los bits en la señal binaria aleatoria tomen la forma de dientes de sierra?

Se deben ajustar los parámetros del vector h y la interpolación en el FIR Filter, de manera que la forma de pulso se modifique gradualmente en el tiempo, generando la pendiente lineal característica de la señal tipo dientes de sierra.

5 Conclusiones

- La PSD depende del origen de la señal: en imágenes se observan picos definidos, mientras que en audio la energía se reparte en un espectro más amplio.
- El audio requiere mayor ancho de banda para transmitirse con calidad, mientras que las imágenes resultan más eficientes en el uso del espectro.
- Ajustar parámetros como el SPS en GNURadio fue clave para obtener resultados claros y entender la relación entre tiempo y frecuencia.
- El uso de GNURadio facilitó el análisis y mostró el valor de las herramientas digitales en el estudio de comunicaciones.

References

- [1] GNU Radio Project, *GNU Radio: Free & Open-Source Toolkit for Software Radio*, 2025. Disponible en: <https://www.gnuradio.org/>
- [2] Omar Javier Tíjaro Rojas, *Práctica de PSD de Señales Aleatorias en GNU Radio*, 2025. Disponible en: <https://github.com/hortegab/ComdigPractices2021sii/blob/main/Fase%20I/Pract3/randombinayrectsignal.grc>