

Informe 2: ESTUDIO DE SEÑALES ALEATORIAS Y PSD

Jairo Andres Moreno Lamus - 2212903

Daniel Felipe Orduz Pinzón - 2202333

Ángel Ricardo Castro Jaimes - 2200501

URL Repositorio: https://github.com/DanielO1006/LAB_COMU2_G7^[1]

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

Resumen— This study explores the characterization of random signals and power spectral density (PSD) using GNU Radio, emphasizing how signal energy is distributed across frequency bands. Through experiments, variations in samples per second (Sps) were tested to observe their influence on PSD behavior. Additionally, the research evaluates how adapting sampling parameters affects the reception and processing of diverse signals, such as audio and images, highlighting their implications for signal integrity and system performance. The findings underscore the importance of tailored parameter configurations in optimizing spectral analysis and signal acquisition workflows.

I. INTRODUCCIÓN

En el ámbito de las comunicaciones, la Densidad Espectral de Potencia (PSD) desempeña un papel esencial al cuantificar cómo se distribuye la energía de una señal en el dominio de la frecuencia, permitiendo evaluar aspectos como el ancho de banda, la eficiencia espectral y la presencia de interferencias. En este laboratorio, se emplea GNU Radio para examinar tanto señales aleatorias bipolares como señales derivadas de fuentes reales (imágenes y audio), analizando su comportamiento en los dominios temporal y espectral. El enfoque se centra en cómo variaciones en parámetros como las muestras por segundo (Sps) alteran la forma de la señal y su PSD, lo que contribuye a entender la relación entre configuración de muestreo y características espectrales.

El objetivo principal de esta práctica es profundizar en el estudio de la PSD aplicada a señales aleatorias. Mediante la generación y análisis de distintos tipos de señales en GNU Radio, se busca comprender cómo optimizar parámetros de transmisión y recepción en sistemas digitales. Este conocimiento resulta clave para diseñar sistemas de comunicación eficientes, donde la correcta gestión del espectro y la minimización de distorsiones son fundamentales para garantizar la integridad y calidad de la información transmitida.

II. METODOLOGIA

El laboratorio comienza con la preparación de los recursos necesarios: se descarga el archivo “random binary rect signal.grc”, así como los archivos de imagen y audio, para su implementación

en GNU Radio. Posteriormente, se ejecuta el diagrama de flujo según las instrucciones del manual, verificando su operatividad adecuada. Esto permite analizar las señales binarias aleatorias generadas, integrando los fundamentos teóricos de la Densidad Espectral de Potencia (PSD) descritos en la referencia bibliográfica [2]. El proceso combina la aplicación práctica en la plataforma con la revisión conceptual para evaluar cómo los parámetros configurados influyen en las características espectrales y temporales de las señales.

II-1. Señal binaria aleatoria bipolar

Se analiza una señal binaria bipolar aleatoria, evaluando su densidad espectral de potencia (PSD), ancho de banda (BW), tasa de bits (Rb) y frecuencia de muestreo ($f_s = R_b \times Sps$). Se varía el número de muestras por segundo (Sps) en 4, 8, 16 y 1 para observar cómo afecta la forma de la señal y su distribución espectral, comparando las diferencias entre cada configuración.

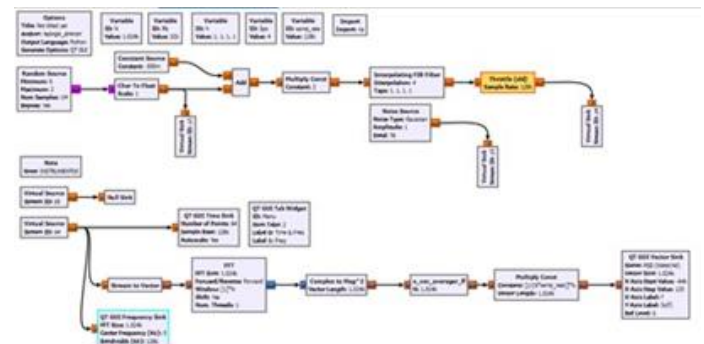


Fig. 1: Diagrama de bloques de señales binarias aleatorias.

II-2. Ruido blanco en tiempo y en PSD

La Figura 2 ilustra la modificación en el bloque virtual source para reemplazar el nodo p5 con p4, monitoreando la señal en tiempo, su PSD y su espectro en frecuencia. Posteriormente, se analiza el efecto de variar las muestras por segundo (Sps) en las propiedades temporales y espectrales de la señal, con el objetivo de relacionar

estos ajustes con cambios en su comportamiento y distribución energética.

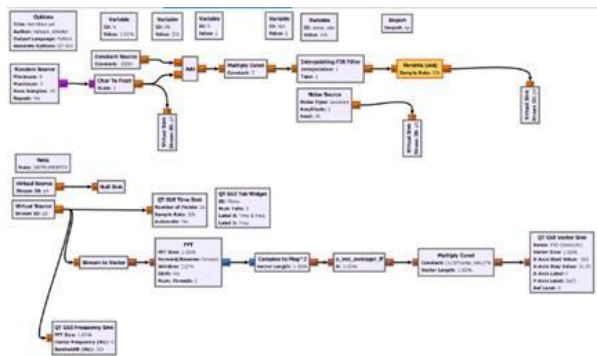


Fig. 2: Diagrama de bloques intercambiando los nodos p5 a p4 en el bloque virtual source.

II-3. Archivo de imagen

La Figura 3 muestra un sistema donde la señal de entrada se adapta con un bloque "file source" y se procesa con "Unpack K Bits" para la imagen "rana.jpg". Se calculan la PSD, el ancho de banda (BW), la tasa de bits (Rb) y el espectro en frecuencia, comparándolos con la señal previa para evaluar el impacto de la fuente y el procesamiento.

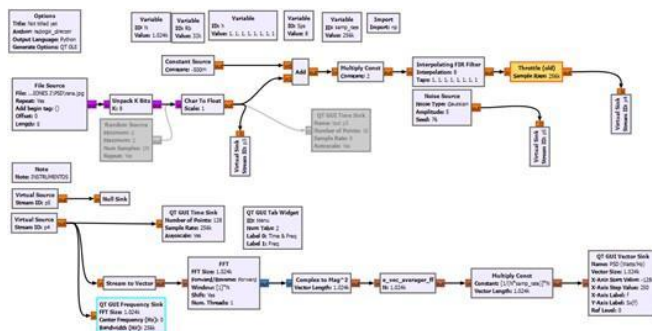


Fig. 3: Diagrama de bloques con imagen como señal de entrada.

II-2. Archivo de audio

La Figura 4 muestra un ajuste en el sistema al reemplazar la señal de entrada por el archivo de audio "sonido.wav", simulando una señal acústica realista. Se miden parámetros como la PSD, el ancho de banda y la frecuencia de muestreo, comparando los resultados con pruebas anteriores que usaron formatos distintos, para analizar cómo varían las características y el comportamiento espectral según la fuente de entrada.

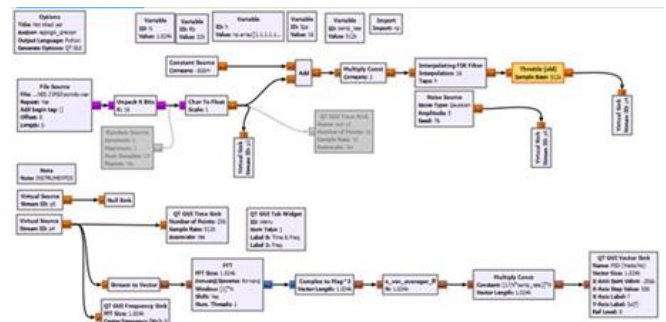


Fig. 4: Diagrama de bloques con audio como señal de entrada.

III. ANALISIS DE RESULTADOS

III-1. Señal binaria aleatoria bipolar

El laboratorio estudió el impacto de Samples Per Symbol (Sps) en una señal binaria bipolar, analizando su efecto en la forma temporal y la densidad espectral de potencia (PSD). Se encontró que valores altos de Sps mejoran la resolución temporal pero aumentan el ancho de banda, mientras que valores bajos causan distorsión y reducen precisión. Esto destaca la importancia de ajustar Sps para equilibrar fidelidad, eficiencia espectral y robustez en sistemas de comunicación digital.

Sps	Rata de bits (Rb)	fs [kHz]	BW [kHz]
1	32k	32	32
4	32k	128	119.97
8	32k	256	122.82
16	32k	512	189.82

Tab. 1: Datos tomados al variar Sps

En la tabla 1 se observan los diferentes datos obtenidos al variar Sps, se resalta que para la selección del ancho de banda (BW) se consideraron algunos lóbulos de la señal adicionales al lóbulo central que es donde está mayormente concentrada la potencia de la señal en la gráfica de PSD y Frecuencia. En la figura 5 se observa la PSD de una señal binaria aleatoria usando Sps = 8, se puede observar que su ancho de banda es un poco mayor a BW = 122[kHz].

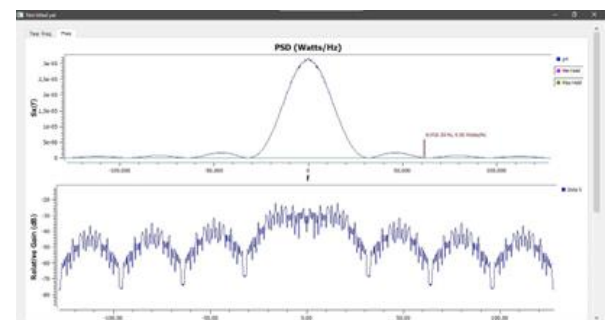


Fig. 5: PSD de Señal binaria aleatoria bipolar junto con su representación en frecuencia y Sps = 8

III-2 Ruido blanco en tiempo y en PSD

Antes del análisis, en la Figura 2, se deben intercambiar los Stream ID en los bloques Virtual Source, cambiando p5 por p4 y viceversa. Esta modificación es crucial para garantizar la correcta visualización de las señales en las Figuras 6, 7 y 8 bajo el parámetro $h = 1$, permitiendo una interpretación adecuada de los resultados.

Durante las pruebas con $Sps = 4, 8$ y 16 , se observó que la PSD se mantiene constante, independientemente de la frecuencia seleccionada. Sin embargo, al incrementar la frecuencia de muestreo (fs), la amplitud de la señal aumenta. Esto indica que, aunque la distribución espectral de energía no se ve afectada, la escala de la señal en el dominio temporal sí cambia. Así, se resalta la importancia de calibrar fs correctamente para mantener la integridad de la amplitud en aplicaciones prácticas.

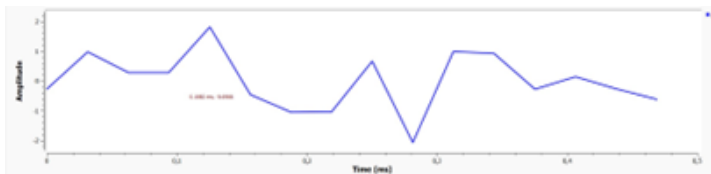


Fig. 6: Ruido blanco en el tiempo con $h = 1$

III-3. Archivo de imagen

Para este análisis, como la imagen se codifica quedará tomando valores binarios con lo cual se puede decir que es una fuente de datos aleatoria, es decir una señal de datos cuadrada, periódica y con armónicos definidos en su PSD, lo que significa que contiene componentes espectrales bien definidos que se concentran en frecuencias específicas, lo que es común para este tipo de señales periódicas. Al aumentar el valor de Sps , se observó que la señal aleatoria se hacía más rectangular, la frecuencia de muestreo se determina de esta forma: $fs = R_b * Sps$. Finalmente, cuando se aumenta el valor de Sps aumenta la amplitud de la PSD.

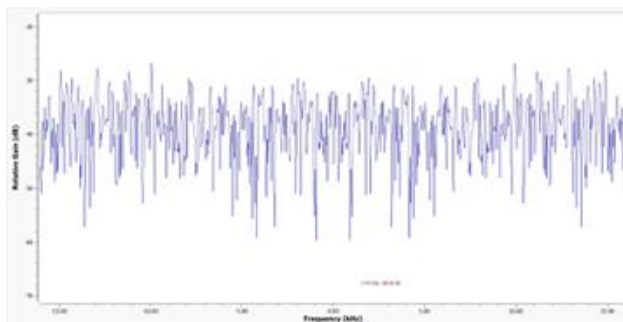


Fig. 7: Ruido blanco en frecuencia con $h = 1$

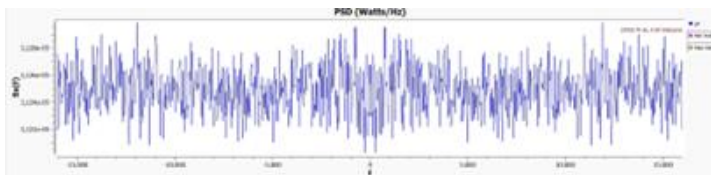


Fig. 8: Ruido blanco en PSD con $h = 1$

En el bloque Unpack K Bits, se probaron $K = 4, 8$ y 16 , eligiendo $K = 8$ para el procesamiento de imágenes, ya que valores mayores causaban solapamiento y distorsión de la señal. Esta configuración garantizó una decodificación precisa sin pérdida de integridad.

Los resultados, registrados en la Tabla 2, compararon la forma de la señal, su PSD y la tasa de bits. Se confirmó que $K = 8$ ofrece un equilibrio óptimo entre resolución de bits y prevención de artefactos en señales basadas en imágenes.

k	PSD [Watts/Hz]	tb [ms]	BW [kHz]
4	6,8E-5	0.0354	64.00
8	7,6E-5	0.0360	62.02
16	1,1E-3	0.0353	64.00

Tab. 2: Datos tomados al usar en el bloque *File Source* el archivo de la imagen

En la figura 9 se observa la señal codificada en el tiempo y también se observa la PSD de la imagen y en la figura 10 se observa la señal en el dominio de la frecuencia, usando $K = 8$ y $Sps = 8$.

III-4. Archivo de audio

Para este análisis, igualmente el archivo de audio se codifica pero esta vez usando $K = 16$ por medio del bloque *Unpack K Bits* [4], ya que haciendo pruebas variando el valor de K , se optó por dejar el valor de $K = 16$ ya que la señal se representa de una mejor manera, también se realizó el análisis con el valor de $Sps = 8$.

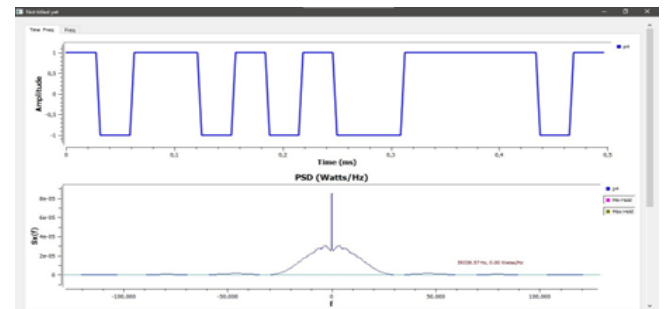


Fig. 9: Señal aleatoria bipolar en el tiempo junto con su PSD respecto al archivo de la imagen

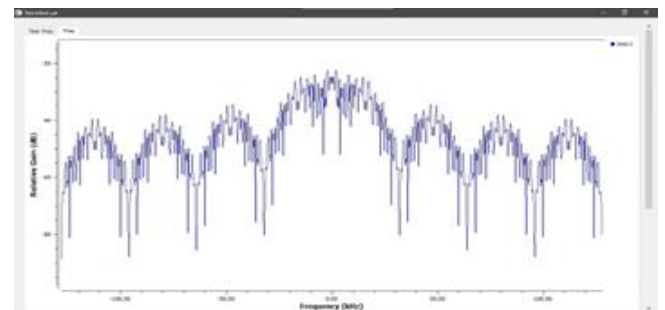


Fig. 10: Señal aleatoria bipolar en frecuencia respecto al archivo de la imagen.

k	PSD [Watts/Hz]	tb [ms]	BW [kHz]
4	6,8E-5	0,0354	64,00
8	7,6E-5	0,0360	62,02
16	1,1E-3	0,0353	64,00

Tab. 3: Datos tomados al usar en el bloque *File Source* el archivo de *sonido*

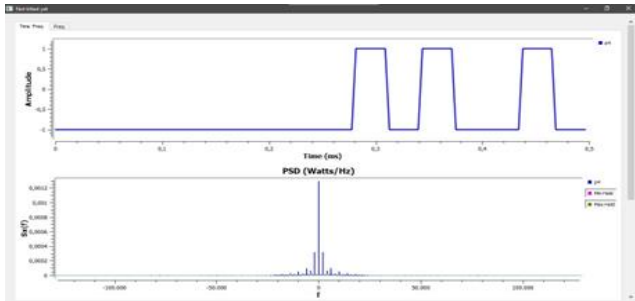


Fig. 11: Señal aleatoria bipolar en el tiempo junto con su PSD respecto al archivo del *audio*

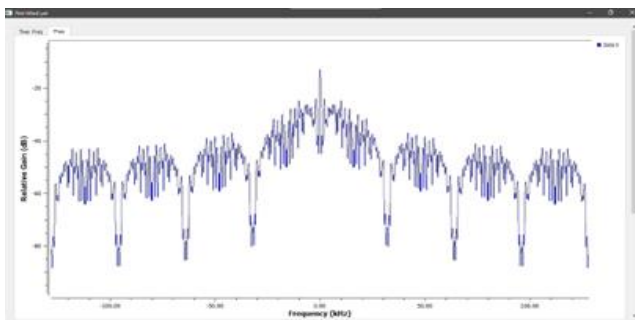


Fig. 12: Señal aleatoria bipolar en frecuencia respecto al archivo del *audio*.

La Tabla 3 muestra el análisis de una señal de audio procesada con el bloque *File Source*. La Figura 11 presenta la señal en el tiempo y su PSD, mientras que la Figura 12 ilustra su espectro en frecuencia, usando $K = 16$ y $Sps = 8$.

El estudio evidenció cómo estos parámetros influyen en la representación temporal y espectral de señales reales no periódicas, como el audio, afectando el ancho de banda y sus características espectrales.

IV. PREGUNTAS Y RESPUESTAS

A. Los bloques "Add" y "Constant Source" añaden un nivel DC a la señal, convirtiéndola en bipolar con valores positivos y negativos. Luego, el bloque "Multiply Const" ajusta la amplitud para mantener los niveles originales (por ejemplo, de $\pm A/2$ a $\pm A$), evitando atenuaciones tras el desplazamiento DC.

B. El bloque "Interpolation FIR Filter" realiza la interpolación de la señal de entrada mediante un filtro FIR, donde el parámetro

de interpolación coincide con el número de muestras por símbolo (Sps), el cual, afecta la visualización de la PSD, y se recomienda utilizar un "QT GUI Time Sink" para analizar las señales en diferentes puntos del flujo.

C. Las señales de audio y de imagen tienen diferencias en su PSD debido a la distribución de frecuencias. En el audio, la energía se concentra en bajas frecuencias (ej., 300 Hz - 3.4 kHz en la voz), mientras que en las imágenes se distribuye más uniformemente, con mayor presencia de altas frecuencias por los bordes y cambios abruptos. Esto explica sus diferencias espectrales, incluso cuando ambas señales son binarias.

D. Debido a la forma que se encuentra implementada GNURADIO el sistema siempre trata de procesar de la manera más rápida los datos, el bloque throttle se encarga de imponer una tasa de muestreo.

E. Aparece una componente en DC, ya que la señal deja de estar centrada en cero, lo que incrementa la energía en bajas frecuencias. Segundo, la distribución espectral cambia, concentrando más potencia en frecuencias bajas en comparación con una señal bipolar, lo que puede afectar la transmisión.

F. En GNU Radio, las señales digitales están muestreadas a una frecuencia finita. Según el Teorema de Nyquist, cualquier señal muestreada solo puede contener componentes de frecuencia hasta cierto punto, Por lo tanto, el ruido blanco generado en GNU Radio estará limitado a cierto rango de frecuencias dado por el límite de Nyquist.

G. En principio sucede algo similar que con lo explicado por el ruido blanco, la frecuencia de muestreo nos limita el rango de frecuencias en donde podemos encontrar la PSD.

H. $N = Sps + 1$

$$I. \frac{Rb * Sps}{2}$$

J. Se calcula dividiendo la frecuencia de muestreo por el número de puntos de la FFT, $\frac{f_s}{N}$

k. Aumentar el parámetro K en el bloque "Unpack K Bits" a 16 incrementa la frecuencia de muestreo de salida en ese factor, lo que genera un flujo de datos más denso y, posiblemente, un incremento en los requerimientos de procesamiento.

L. Si se desea calcular la frecuencia de muestreo a la entrada del bloque "Unpack K Bits" mediante el número de lóbulos y el ancho de banda, se puede utilizar la fórmula: $f_s = N * BW$

M. La frecuencia de muestreo en la salida del bloque K bits sería:

$$F_{\text{out}} = K \cdot F_{\text{sin}}$$

N. El bloque char to float no realiza ningún cambio representativo para la frecuencia de muestreo.

O. Para $S_{\text{ps}}=1$, la PSD de una señal binaria aleatoria bipolar es similar a la PSD del ruido blanco, ya que los valores de la señal son casi independientes y la energía se distribuye uniformemente en el espectro.

P. Para la señal Dientes de Sierra se debe ajustar h para crear una rampa con una caída abrupta.

Q. En Unipolar RZ, la señal debe volver a cero a la mitad de cada bit.

R. Para generar una señal Manchester NRZ, es necesario generar una transición de fase en el centro de cada bit.

S. La señal OOK se logra apagando la señal para un bit "0" manteniéndola activa para un "1" y podríamos realizar esto eliminando el nivel DC que agregamos.

T. En *BPSK*, se ajusta la fase de la señal, cambiando en 180° entre los bits.

U. Para *ASK*, se varía la amplitud de acuerdo con el valor del bit.

V. En la señal de latidos del corazón (*ECG*), se crean pulsos de amplitud controlada

W. Para hacer pulsos rizados, se crea una señal con unos y ceros y se aplica un filtro que suaviza los cambios. Esto hace que los bordes de los pulsos tengan curvas en lugar de ser rectos. Cuanto más fuerte sea el filtro, mayor será el rizado.

X. Señal Unipolar: Tiene un nivel DC y, por lo tanto, tiene una mayor concentración de energía en frecuencias bajas (esto se refleja en una mayor densidad de potencia en bajas frecuencias).

Señal Bipolar: Tiene tanto componentes positivos como negativos, lo que tiende a distribuir la energía de manera más uniforme en el espectro de frecuencias.

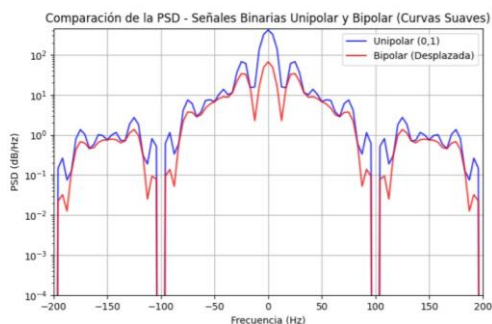


Fig.13 comparación entre señal bipolar y unipolar

V. Conclusiones

1. El análisis espectral muestra que las señales de audio tienen una PSD concentrada en frecuencias bajas y medias, reflejando su naturaleza continua. En contraste, las señales de imagen con $K=8K=8$ presentan un espectro más disperso debido a sus rápidas variaciones. Esto resalta la importancia de un procesamiento adecuado según las características de cada tipo de señal.

2. El aumento de S_{ps} mejora la resolución temporal y la calidad de la señal, suavizando transiciones y reduciendo el contenido de alta frecuencia. Sin embargo, también incrementa el ancho de banda requerido, lo que puede afectar la eficiencia del canal de transmisión.

3. Las señales polares concentran más potencia en bajas frecuencias por su componente DC, mientras que las bipolares tienen una distribución más uniforme. Esta diferencia influye en la eficiencia de transmisión y en la respuesta de los sistemas de acoplo capacitivo.

REFERENCIAS

- [1] Homero Ortega Boda, Oscar Mauricio Reyes Torres. *Comunicaciones Digitales basadas en radio definida por software*. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones.
- [2] GNU Radio. *Python Module - GNU Radio*. Disponible en: <https://wiki.gnuradio.org/index.php?title=Python+Module>
- [3] B. de Multison, "Frecuencia de muestreo: qué es y cómo calcularla." [Online]. Available: <https://multisononline.com/blog/de-analogico-a-digital-frecuencia-de-muestreo-y-tasa-de-bits-n2>