Informe 2: ESTUDIO DE SEÑALES ALEATORIAS Y PSD

Humberto José Contreras Afanador - 2210404
Diego Andrés García Díaz - 2195533
Darwin Camilo Sánchez Gallo - 2202335

URL Repositorio: `https://github.com/dagdUIS/CommunicationsII_2024_2_G1` **[1]**

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

12 de octubre de 2024

Resumen

This report focuses on the analysis and development of random signals and power spectral density (PSD) using GNU Radio software. The PSD analysis allows understanding how the signal energy is distributed as a function of frequency. For this purpose, several tests are performed by modifying the samples per second (Sps) and observing the changes that occur in the PSD. The reception of different signal sources, such as images and audio, involves adjustments to the sampling parameters which are analyzed to assess the impact on the overall process.

Keywords: *PSD, energy, frequency, Sps, sampling*

1. Introducción

En el campo de las comunicaciones, la **Densidad Espectral de Potencia (PSD)**, es una herramienta fundamental para evaluar la distribución de energía de una señal en función de la frecuencia. El análisis de la PSD facilita el estudio del ancho de banda, eficiencia espectral e interferencias. En esta práctica, se usa GNU Radio para analizar señales aleatorias bipolares y señales provenientes de fuentes reales, como imágenes y archivos de audio; el análisis de estas señales permite observar el comportamiento de estas en los dominios de tiempo y frecuencia, enfocándose en cómo la PSD varía con la modificación de parámetros clave como las muestras por segundo (Sps) y el impacto de estas modificaciones en la forma de la señal y en su PSD.

El principal objetivo de este laboratorio es poder aplicar y profundizar el estudio de la PSD en señales aleatorias. Al generar y analizar diferentes tipos de señales en GNU Radio, se espera poder comprender cómo optimizar la transmisión y recepción de señales digitales en sistemas de comunicaciones, ya que es un aspecto crítico en el diseño de sistemas eficientes.

2. Metodología

El desarrollo del laboratorio inicia con la descarga del archivo **"randombinaryrectsignal.grc"** junto con los archivos de imagen y audio, los cuales se utilizaron en GNU Radio. A partir de esto, se ejecuta el diagrama de flujo siguiendo las indicaciones del documento guía para posteriormente verificar el correcto funcionamiento y realizar el respectivo análisis de las señales binarias aleatorias, también se tienen en cuenta los conceptos de la PSD que se encuentran en el libro guía [2].

2.1. Señal binaria aleatoria bipolar

Para una señal binaria aleatoria bipolar, se determinan diversos parámetros, entre los cuales se incluyen la densidad espectral de potencia (PSD), el ancho de banda (BW), la tasa de bits y la frecuencia de muestreo $(fs = R_b * Sps)$. Estos parámetros se analizan, variando la cantidad de muestras por segundo (Sps) a valores de 4, 8, 16 y 1, observando las variaciones resultantes entre cada señal. [3]

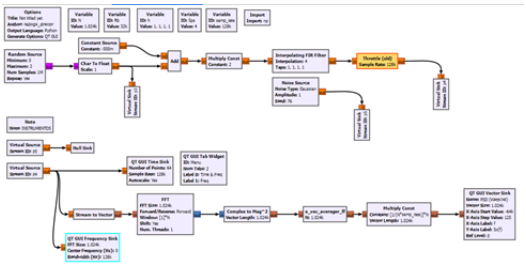

Fig. 1: Diagrama de bloques de señales binarias aleatorias.

2.2. Ruido blanco en tiempo y en PSD

La figura 2 presenta la configuración modificada en el bloque virtual source, donde se intercambia el nodo $p5$ por el $p4$. Se visualizan la señal de salida en el tiempo, la PSD y el espectro de la señal. Luego, se realizan pruebas variando la cantidad de muestras (Sps) para determinar si se produce algún cambio.

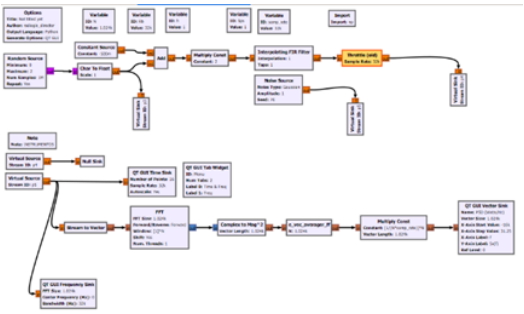

Fig. 2: Diagrama de bloques intercambiando los nodos p5 a p4 en el bloque virtual source.

2.3. Archivo de imágen

Mediante la figura 3, se presenta el diagrama de flujo en el que la señal de entrada es modificada a través de un bloque tipo "$file\ source$". Además, se incluye un bloque "$Unpack\ K\ Bits$", que permite recibir la señal de entrada correspondiente a la imagen "$rana.jpg$". Esta señal es procesada por el sistema para determinar la PSD, el BW, el Rb y el espectro de la señal. Finalmente, se realiza un análisis de la variación de estos parámetros en comparación con la señal previamente estudiada.

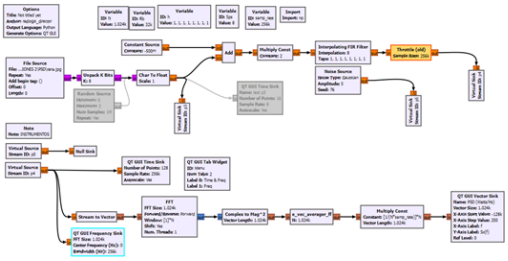

Fig. 3: Diagrama de bloques con imagen como señal de entrada.

2.4. Archivo de audio

La figura 4 muestra una modificación en el diagrama de flujo, específicamente en el bloque "$file\ source$", donde se reemplaza la señal de entrada por un archivo de audio "$sonido.wav$", que representa una señal proveniente de un micrófono en el mundo real. Al pasar por el sistema, se obtendrán la PSD (Densidad Espectral de Potencia), el BW (ancho de banda), la fs (frecuencia de muestreo) y otros parámetros, con el fin de comparar los resultados con las pruebas previas que utilizaron diferentes formatos en las señales de entrada.

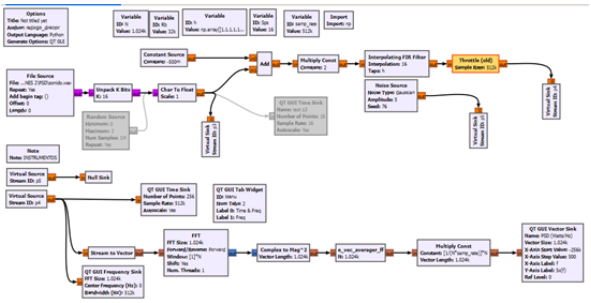

Fig. 4: Diagrama de bloques con audio como señal de entrada.

3. Análisis de resultados

3.1. Señal binaria aleatoria bipolar

Se ejecutó el diagrama de flujo, se llevó a cabo una detallada evaluación del flujograma propuesto para la práctica, centrándose en el análisis de una señal binaria aleatoria bipolar de forma rectangular. A través de diversas pruebas y experimentos realizados con diferentes valores de Samples Per Symbol (Sps), se examinaron tanto la forma temporal de la señal como su densidad Espectral de Potencia (PSD). Los resultados obtenidos proporcionaron una comprensión profunda del comportamiento del sistema bajo diferentes condiciones, destacando la influencia de las Sps en la representación y la distribución espectral de la señal.

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 la

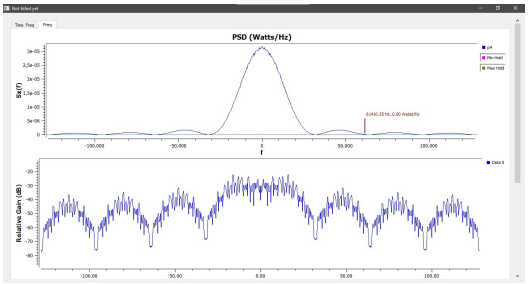

Fig. 5: PSD de Señal binaria aleatoria bipolar junto con su representación en frecuencia y $Sps = 8$

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 $BW = 122[kHz]$.

3.2. Ruido blanco en tiempo y en PSD

Cabe aclarar que antes de realizar el análisis, en el diagrama de flujo de la figura 2, se deben cambiar los nombres correspondientes al ***Stream ID*** que esta contenido en los bloques ***Virtual Source***, que inicialmente estaban como p5 y p4 respectivamente, se deben cambiar a p4 y p5, esto con el fin de poder observar las señales que se observan en las figuras 6, 7 y 8, aclarando que se usa $h = 1$.

Además, para cada una de las anteriores pruebas (haciendo también que $Sps = 4$, $Sps = 8$ y $Sps = 16$) se observó que no importa el valor de frecuencia que se tenga, el valor de la PSD no varía, y a medida que se aumenta el valor de la frecuencia de muestreo también aumenta la amplitud de la señal.

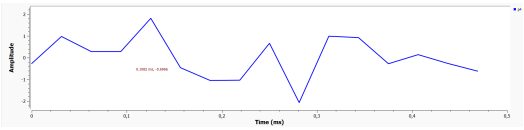

Fig. 6: Ruido blanco en el tiempo con h = 1

3.3. 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

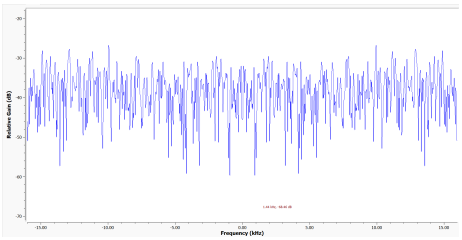

Fig. 7: Ruido blanco en frecuencia con h = 1

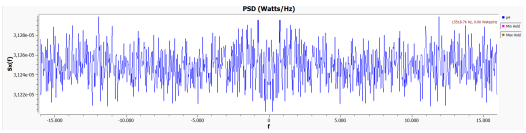

Fig. 8: Ruido blanco en PSD con h = 1

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 hacia más rectángular, 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.

En el bloque *Unpack K Bits* [4], se probó cambiando el valor de K con 4, 8 y 16 y para el caso de la imágen se optó por el valor de $K = 8$, ya que al aumentar el valor de K la señal tiende a convolucionar datos con otros datos (solapamiento). En la tabla 2 se observan los datos tomados al realizar el análisis usando como archivo una **imágen** en el bloque *File Source*.

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 *imágen*

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 $Sps = 8$.

3.4. Archivo de audio

Para este análisis, igualmente el archivo de audio se codificara pero esta vez usando $K = 16$ por medio del bloque *Unpack K Bits* [4], ya que haciendo pruebas va-

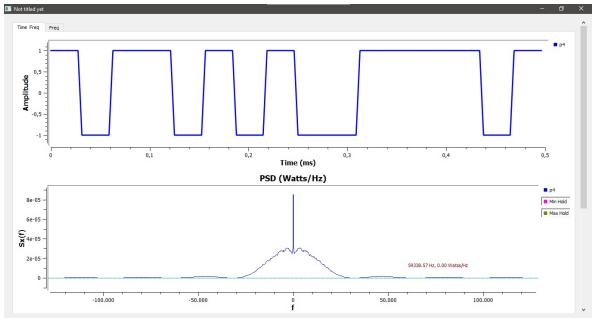

Fig. 9: Señal aleatoria bipolar en el tiempo junto con su PSD respecto al archivo de la *imágen*

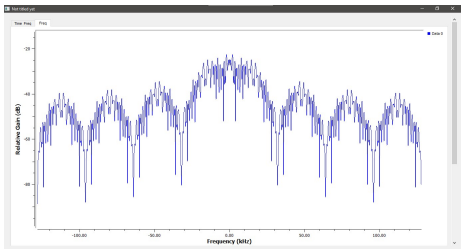

Fig. 10: Señal aleatoria bipolar en frecuencia respecto al archivo de la *imágen*

riando el valor de K, se optó por dejar el valor de $K = 16$ ya que la señal se representaba de una mejor manera, también se realizó el análisis con el valor de $Sps = 8$.

k	PSD [Watts/Hz]	tb [ms]	BW [kHz]
4	0,10E-3	0,0363	64,00
8	0,25E-3	0,0360	62,02
16	1,25E-3	0,0354	64,00

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

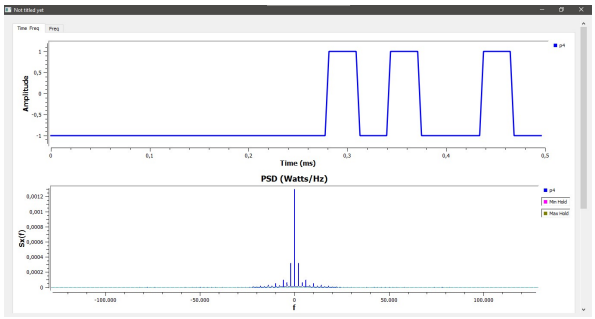

Fig. 11: Señal aleatoria bipolar en el tiempo junto con su PSD respecto al archivo del *audio*

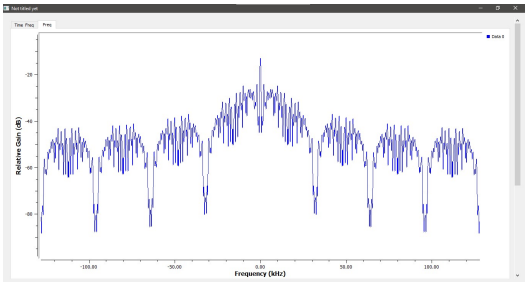

Fig. 12: Señal aleatoria bipolar en frecuencia respecto al archivo del *audio*

En la tabla 3 se observan los datos tomados al realizar el análisis usando como archivo un **audio** en el bloque *File Source*. En la figura 11 se observa la señal codificada en el tiempo y también se observa la PSD del audio y en la figura 12 se observa la señal en el dominio de la frecuencia, usando $K = 16$ y $Sps = 8$.

4. Preguntas

El bloque "Add" suma señales de entrada, incluida una señal que ha sido convertida a formato flotante junto a una fuente constante, multiplicándose luego por 2 para modificar la amplitud de la señal. Por su parte, 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.

El bloque *"Throttle"* limita la tasa de procesamiento para prevenir la saturación del sistema, mientras que al transformar una señal binaria de valores 0 o 1 a formato bipolar (de 1 a 1) se elimina el componente de DC, cambiando la concentración de energía en la PSD, que con una señal de valores 0 y 1 concentra la energía en la frecuencia cero. La PSD de ruido blanco teóricamente tiene un ancho de banda infinito, pero en GNU Radio está limitada por la frecuencia de muestreo y la resolución del software; algo similar ocurre con las señales binarias rectangulares, cuyo ancho de banda infinito en teoría se limita en GNU Radio a causa de la tasa de muestreo y la resolución.

Además, la cantidad de lóbulos en la PSD de una señal binaria aleatoria rectangular se calcula como $N = Sps + 1$, y el rango de frecuencias que ocupa el espectro se determina con $\frac{Rb*Sps}{2}$. La resolución

espectral del analizador de espectros se calcula dividiendo la frecuencia de muestreo por el número de puntos de la FFT, $\frac{fs}{N}$. 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. 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 $fs = N * BW$. La salida del bloque *"Char to Float"* mantiene la misma frecuencia de muestreo que la entrada, mientras que una señal binaria aleatoria bipolar presenta una PSD similar al ruido blanco cuando el número de muestras por símbolo es 1 ($Sps = 1$), ya que se alcanza una variabilidad máxima entre muestras consecutivas, lo que da un espectro amplio y uniforme.

Para ajustar el flujograma y lograr diferentes formas de señal, como *Dientes de Sierra, Unipolar RZ, Manchester NRZ, OOK, BPSK, ASK*, o una señal que imite los latidos del corazón (*ECG*), es esencial modificar el parámetro h y aprovechar el *Filtro FIR Interpolador*. Para la señal Dientes de Sierra se debe ajustar h para crear una rampa con una caída abrupta. En *Unipolar RZ*, la señal debe volver a cero a la mitad de cada bit, mientras que para *Manchester NRZ*, es necesario generar una transición de fase en el centro de cada bit. La señal *OOK* se logra apagando la señal para un bit "0z manteniéndola activa para un "1". En *BPSK*, se ajusta la fase de la señal, cambiando en 180° entre los bits. Para *ASK*, se varía la amplitud de acuerdo con el valor del bit. En la señal de latidos del corazón (*ECG*), se crean pulsos de amplitud controlada.

5. Conclusiones

- Usar el bloque *Unpack K Bits* en GNU Radio con diferentes valores de K ($K = 8$ para imágenes y $K = 16$ para audio) en caso del análisis realizado, influye directamente en la calidad y resolución de las señales que se analizan. Para las imágenes, usar $K = 8$ implica que cada píxel será representado con 8 bits, lo que permite un nivel estándar de detalle visual. En el caso de la señal de audio, el uso de $K = 16$ ofrece una mayor resolución, permitiendo una representación más precisa de la señal, lo que resulta esencial para mantener la fidelidad del audio en su transmisión y análisis espectral.
- Las señales de audio tienden a tener una PSD más concentrada en bandas de frecuencias bajas y medias, reflejando la naturaleza continua y de baja frecuencia de la voz o los sonidos capturados. En contraste, las señales de imagen procesadas a través de la codificación $K = 8$ tienden a tener un espectro más disperso, debido a las variaciones rápidas en el contenido de la imagen, lo que indica una mayor complejidad de frecuencia.
- El parámetro Sps tiene un impacto directo en la representación temporal de las señales y en su PSD, a medida que se incrementa el número de muestras por símbolo, se mejora la resolución temporal y la calidad de la señal, lo que permite captar más detalles de las transiciones entre los bits. Esto es especialmente importante para señales binarias, donde una mayor cantidad de Sps suaviza las transiciones abruptas, reduciendo el contenido de alta frecuencia en la señal, sin embargo, este aumento también conlleva un mayor ancho de banda, lo que significa que la señal ocupará más espacio en el espectro de frecuencias, lo que puede generar la necesidad de mayor capacidad en el canal de transmisión.
- Finalmente, el contenido del desarrollo de está practica se encuentra almacenado en el repositorio cuyo link se encuentra al inicio del documento [1], los archivos relacionados al desarrollo de esta práctica se encuentran en la rama **Laboratorio_3**.

Referencias

[1] D. García, D. Sánchez, and H. Contreras, "Repositorio CommunicationsII_2024_2_G1." [Online]. Available: https://github.com/dagdUIS/CommunicationsII_2024_2_G1

[2] H. Ortega, B. Bodá, O. Mauricio, and R. Torres, "Comunicaciones digitales basadas en radio definida por software." [Online]. Available: https://sites.google.com/saber.uis.edu.co/comdig

[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

[4] G. R. Block Docs, "Unpack k bits." [Online]. Available: https://wiki.gnuradio.org/index.php/Unpack_K_Bits