Práctica 2: PSD DE SEÑALES ALEATORIAS (GNURADIO)

JUAN SEBASTIAN AVILA DIAZ - 2200524
JENNIFER TATIANA CHAPARRO LIZARAZO - 2191468
WILLIAM ANDRES PARRA RUEDA - 2184680

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

15 de Marzo de 2024

Resumen

El informe aborda el análisis de la practica de señales en el tiempo y Densidad Espectral de Potencia (DSP), así como el procesamiento de señales y el uso de software de simulación (GNU Radio). Se analizan varios tipos de señales, incluyendo señales aleatorias bipolares, imágenes, señales de audio y señales binarias, evaluando su contenido espectral, ancho de banda y otros parámetros clave.

Palabras clave: Densidad Espectral de Potencia, GNU Radio, señales aleatorias, señales binarias. .

1. Introducción

El análisis de señales desempeña un papel fundamental en los campos relacionados con las comunicaciones. En este informe, se profundiza la comprensión de diversos tipos de señales, abarcando desde señales aleatorias bipolares,unipolares y señales de audio. A través del uso de la herramienta GNU radio, se investiga el comportamiento de estas señales en los dominios del tiempo y la frecuencia. Además, se exploran temas como la la interpolación y la modulación, proporcionando así una visión integral de cómo analizar y manipular señales en diversas aplicaciones.[1]

2. Procedimiento y análisis

Los Software utilizados para llevar a cabo el desarrollo de esta práctica de laboratorio fueron los siguientes:

- Linux (Ubuntu)
- GitHub
- git (terminal)
- GNU Radio

Para iniciar la práctica se creó una nueva rama en el repositorio de GitHub con el nombre de Práctica 2. Se descargó el contenido en el terminal de Linux y en una nueva carpeta se importaron los archivos necesarios para realizar la práctica. Así mismo, se crearon las respectivas ramas de trabajo para cada integrante del grupo.

En el software de GNU Radio se abrió el flujograma $randombinaryrectsignal.grc$ y se comprobó su correcto funcionamiento tal como se muestra en la Figura 1.

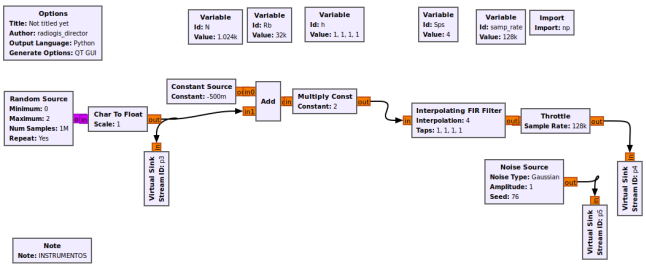

Figura 1. Flujograma en GNU Radio

- Para una señal aleatoria bipolar se obtuvo la forma de la señal en el dominio del tiempo y su densidad espectral de potencia. En la Figura 2 se pueden visualizar estas señales para un SPS = 4. En la primera gráfica se nota el comportamiento bipolar aleatorio de la señal en el tiempo entre -1 y 1, en la segunda gráfica se aprecia la PSD la cuál es muy pequeña teniendo su pico máximo en 300[mW/Hz] y un ancho de banda de 64[kHz].

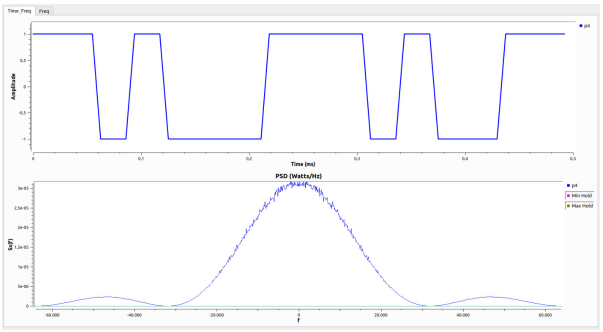

Figura 2. Gráfica en el tiempo y PSD para un Sps=4

Este parámetro Sps es el valor de interpolación para un filtro FIR (Finite Impulse Response). Se realizaron varias pruebas con diferentes valores en Sps. En la Tabla 1 se refleja la variación de los parámetros de la señal con cada prueba. La tasa de bits y el ancho de banda se mantienen igual para todos los valores. El Sps es la longitud de datos en el parámetro h, la frecuencia de muestreo si varía respecto al parámetro Sps

h	sps	rate bit	samp_rate	ancho de banda
1,1,1,1	4	32k	128k	64kHz
1,1,1,1,1,1,1,1	8	32k	256k	64kHz
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1	16	32k	512k	64kHz
1	1	32k	32k	64kHz

Tabla 1. Parámetros de la señal según h

- El ruido blanco tipo gausiano es un tipo de ruido aleatorio que se caracteriza por tener una PSD constante en todo el espectro de frecuencia y una distribución gausiana en el dominio del tiempo. Para comprobar este tipo de ruido en la señal, se intercambiaron las posiciones de las "$Virtualsource$" para que se muestre solamente la señal del ruido. La figura 3 muestra la forma del ruido blanco para una amplitud de 100, se puede notas que la señal aleatoria oscila entre -100 y 100, la PSD como se mencionó anteriormente es constante y para este caso se encuentra en 0.08[W/Hz]. Para cuando se le aplicó una amplitud de 1, la PSD disminuyó a 8[$\mu W/Hz$]. Así que la amplitud es directamente proporcional a la PSD.

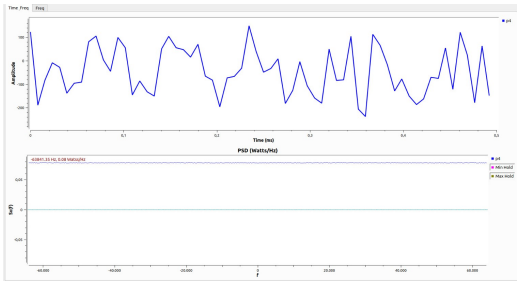

Figura 3. Ruido blanco con amplitud de 100

Preguntas de control

- El bloque "$ConstantSource$" es una fuente constante, la cual en el diagrama esta generando una fuente constante de -500m que posteriormente ingresa al bloque "Add", el cual es un sumador que esta sumando la señal del bloque "$ConstantSource$" con la señal procesada, la salida del bloque "Add" ingresa al bloque "$MultiplyConstant$", el cual tiene como función multiplicar la señal de entrada por una constante ingresada por el usuario, la cual para el diagrama es de 2.

1. El bloque Interpolating FIR Filter es un filtro de interpolación, este toma la señal de entrada con cierta tasa de muestreo, y le realiza un proceso de interpolación para aumentar la tasa de muestreo, agregando muestras adicionales a la señal ingresada y la suaviza mediante el filtro FIR.
2. El parámetro Interpolation es la tasa de interpolación, el cual tiene un valor de SPS en el bloque Interpolating FIR Filter porque SPS es la longitud de los coeficientes de interpolación, se usa para tener una mayor precisión de interpolación, si este valor se varia sin relacionarlos con los taps se genera mas muestras de la señal, pero la señal mantiene su forma original.
3. Para analizar la señal en P3, se intercambia con P4 de la misma manera que con el ruido blanco, de modo que la Virtual source = P3 entre directamente al bloque QT GUI Frecuency y a los demás bloques de análisis.
4. El ancho de banda de la señal en P4 es directamente proporcional a Rb y Sps, y se puede calcular a partir de la ecuación (1), en donde samp rate es la frecuencia de muestreo y este valor es el mismo del ancho de banda ya que

se trata de una señal en bandabase. Rb es la tasa de bits y Sps el parámetro de interpolación del filtro.

$$Bandwith = samp_{rate}(p4) = Rb * Sps \quad (1)$$

5. Si se tiene la frecuencia de muestreo en la señal de salida P4 después de filtrar la señal de entrada, se puede hallar la frecuencia de muestreo a partir de la ecuación (1), con la que a partir del parámetro de tasa de bits (Rb) y conociendo el parámetro que define la interpolación (Sps), se puede hallar la frecuencia de muestreo de la señal filtrada. Por la tanto, esta frecuencia de la señal sin filtrar será dividiendo el efecto del filtro de interpolación. Esto se puede ver en la ecuación (2).

$$samp_{rate(p3)} = \frac{samp_{rate(p4)}}{Sps} \quad (2)$$

- c) A pesar de que ambas señales son binarias, la información proveniente de cada una de ellas es distinta, y la PSD no depende de la forma de onda en la que se muestra la información, si no de las características en frecuencia de su fuente. Para la señal de audio, las variaciones en su frecuencia son muy grandes, ya que esta posee armónicos distintos provenientes de la voz, por lo tanto, su densidad espectral de potencia, se ve como picos grandes en cada frecuencia. La imagen por el contrario, tiene variaciones de frecuencia muy bajas ya que los colores en la imagen son casi constantes estando toda la distribución de información de los píxeles en un solo rango de frecuencias, por eso su PSD, se ve como una distribución gausiana siendo más grande alrededor de una sola frecuencia. En la figura 4 se aprecia este análisis, la primera gráfica se corresponde a la PSD de la imagen, y la segunda gráfica a la PSD del audio.

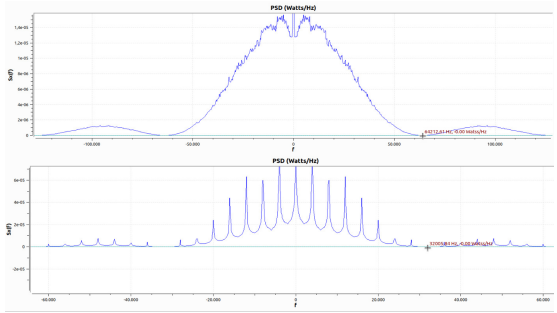

Figura 4. Comparación PSD del audio y de la imagen de la rana

- d) el bloque "$Throttle$" cumple una función importante ya que este bloque regule a la velocidad a la que lo datos fluyen en el sistema, para que la tasa promedio no exceda la tasa especifica.

- e) La PSD de la señal bipolar no existe en la frecuencia de 0[Hz], esto se debe a que la señal bipolar va de -1 a 1 y la potencia de esta se distribuye en ambos lados de cero, por lo tanto, se cancela mutuamente en el componente de frecuencia cero. Por otro lado, para una señal unipolar, que solo oscila por encima de cero, toda la potencia de la señal se concentra en un solo lado de cero, lo que permite que la PSD tenga un valor en 0 Hz. En la figura 5 la primera gráfica corresponde a la PSD de la señal unipolar, mientras que la segunda representa el comportamiento de la PSD de la señal bipolar.

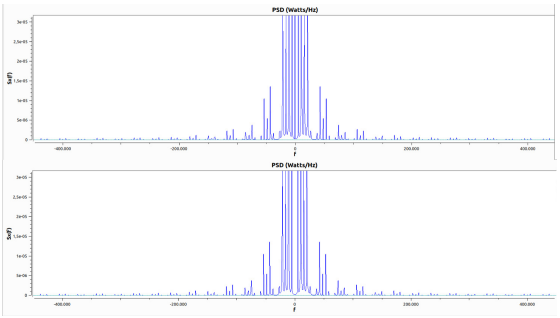

Figura 5. Comparación PSD de una señal unipolar y una bipolar

- f) En la práctica se puede observa que el ruido tiene un ancho de banda infinito al momento de graficar este, ya que se presenta como una constante en todo el rango de frecuencias, esto se debe a que el ruido blanco esta presente en todo el rango de frecuencias.

- g) Teóricamente el ancho de banda de una señal binaria aleatoria es infinito, pero experimentalmente en GNU Radio no se muestra este comportamiento debido a que la señal se enfrenta a limitaciones y restricciones en el software y hardware del sistema. Para observar esto se colocó una interpolación de 65 en el filtro FIR, como se muestra en la figura 5, el ancho de banda con el método de los -20dB es muy cercano a 0.

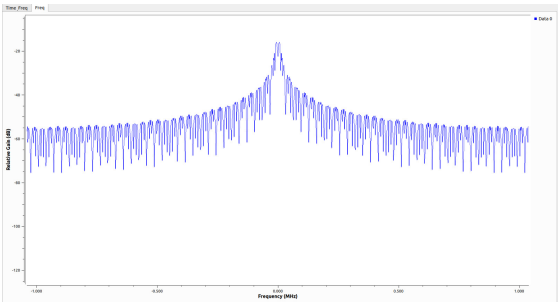

Figura 5. Ancho de banda finito para una señal binaria aleatoria en GNU Radio

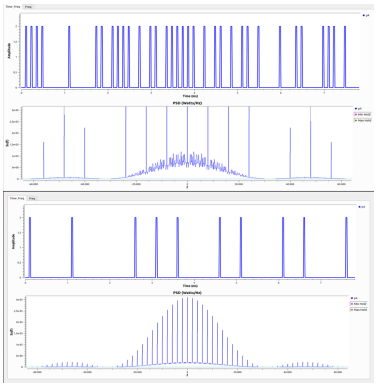

Figura 6. Comparación de Unpack K Bits en 4 y en 16

- h) La PSD de una señal binaria aleatoria, tiene la forma de una sinc(x), la variación del número de lóbulos depende únicamente del filtro de interpolación según lo que se experimentó en la práctica. La frecuencia de muestreo debe cumplir el criterio de nyquist para mostrar correctamente la densidad espectral de potencia. El número de lóbulos se define por el número de la interpolación, es decir, la longitud del vector h. Por lo tanto, N=Sps, contando el lóbulo en 0 por dos lóbulos.

- i) El rango de frecuencias que ocupa todo el espectro está definido como el ancho de banda mostrado en el simulador de GNU Radio, este parámetro está definido por la frecuencia de muestreo, por lo tanto, conociendo Rb y Sps, $Rango = f_s = R_b * Sps$.

- j) Para calcular la resolución espectral se realiza mediante la siguiente fórmula: $\frac{fm}{N}$, donde fm es la frecuencia de muestreo y N es el numero de muestras.

- k) Cuando se coloca el parámetro K=4 la información se empaqueta con 4 bits por paquete. En las dos primeras gráficas de la figura 6 se aprecia la transmisión de bits y la PSD de este comportamiento. Debido a que se empaqueta menos información, la señal resultante tiene menos variabilidad en los símbolos transmitidos. Esto coincide en la PSD con menos componentes de frecuencia y una distribución de potencia más concentrada en frecuencias específicas. En las dos siguientes gráficas de la figura 6 se muestra el comportamiento del empaquetamiento de bits con K=16 en donde cada paquete tiene 16 bits, por lo tanto aumenta la variabilidad de los símbolos transmitidos y su PSD tiene más componentes de frecuencia y una distribución de potencia más amplia en el dominio de la frecuencia.

- l) El empaquetamiento de bits se relaciona directamente con la tasa de bits (Rb), teniendo el ancho de banda y el número de lóbulos (Sps), junto con las ecuaciones (1) y (2), se puede determinar que la frecuencia de muestreo a la salida del bloque Unpack K Bits es la misma tasa de transmisión de bits (Rb).

- m) La frecuencia de muestreo a la salida es la frecuencia de muestreo de la entrada multiplicada por el número de bits por paquete (K).

- n) La frecuencia de muestreo a la salida del bloque Char to Float es la misma que la del punto anterior, es decir $fs * K$, ya que este bloque solamente convierte los datos de tipo caracter a tipo flotante y la frecuencia de muestreo a su entrada es la misma que en su salida.

- o) Para que la PSD de una señal aleatoria fuera muy parecida a la que se obtiene del ruido blanco, se hicieron varias pruebas de las cuales se concluyó que cuando la variable h tiene una longitud de 1 se llega a lo deseado.

- p) Para obtener la señal diente de sierra se cambió el arreglo que estaba en la variable h, por una lista creada a partir de la función np.arange, además se modificó el valor del bloque constant source por 0 y la entrada se dejó con el bloque random.

- q) Para que la señal tomara la forma de codificación de línea Unipolar RZ, se aumentó el tamaño del vector h para que tomara una forma completamente cuadrada y al bloque constant source se le asignó un valor de 0 para que la señal se mostrara como se esperaba. En la Tabla 2 se denota el comportamiento en el dominio del tiempo.

- r) Para la codificación Manchester se manipuló el vector h con la codificación correspondiente para que la señal binaria aleatoria pasara por el filtro y se comportara de la siguiente manera: Cada bit 1 corresponde a una transición de bajo a alto y cada bit 0 corresponde a una transición de alto a bajo. En la Tabla 2 se puede visualizar el resultado.
- s) Para conseguir que la señal resultante cumpliera con lo que se espera de una modulación OOK, se cambió el arreglo que estaba presente en h, por una señal sinusoidal concatenada con una constante de valor 0, para esto se utilizaron las funciones np.concatenate y np.arange.
- t) Como se quería obtener una modulación BPSK en la salida se decidió cambiar el arreglo que estaba presente en la variable h por dos funciones sinusoidales concatenadas, una de amplitud negativa y la otra de amplitud positiva (lo anterior se hizo con la finalidad de realizar el cambio de fase que se hace presente en este tipo de modulación). Además para poder analizar el resultado de una manera eficiente se cambió la entrada por un vector que se repetía de manera infinita tomando valor en 0 y 1.
- u) La señal resultante tomó la forma de una modulación ASK, al cambiar h por una función sinusoidal. Además para que los resultados fueran más fáciles de analizar, se cambió la entrada random por un vector previamente definido.
- v) La señal resultante tomó la forma de los latidos del corazón, cambiando h por una función el tipo sinc(x). Para que esta fuera muy similar a lo que se esperaba, resultó de suma importancia variar los pasos para los cuales se tomaron valores. Se cambió la entrada random por un vector que alternaba de 0 a 1 con la finalidad de obtener una respuesta más amena.
- w) Para obtener los pulsos rizados se utilizó la función de numpy np.concatenate, agregando así una exponencial decreciente con magnitud negativa, mas un arreglo de unos y otra exponencial decreciente pero con amplitud positiva. La entrada random se remplazó con un vector definido y así se obtuvo una gráfica muy similar a la deseada.
- x) Analizando los resultados obtenidos para la señal binaria unipolar y la bipolar, se puede concluir que la diferencia que existe en la PSD, es que en la unipolar la componente espectral que se encuentra en el 0 es nula y en la bipolar si toma valor.

En la Tabla 2 se pueden visualizar las modulaciones trabajadas en este laboratorio y las cuales fueron analizadas en este informe.

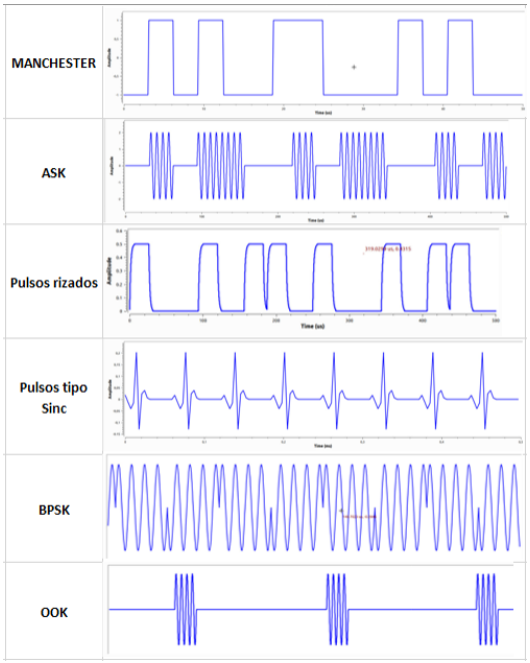

Tabla 2. Gráficas en el dominio del tiempo de las modulaciones vistas en la práctica

3. Conclusiones

- Cuando se realiza el empaquetamiento de una señal con una mayor cantidad de bits se puede notar como la PSD toma muchas más componentes espectrales y es mucho más densa.
- La variación en las componentes espectrales obtenidas en la PSD de una imagen dependen de qué tanto varíen los cambios de intensidad y frecuencias en los pixeles, y al analizar el audio se concluye que simplemente depende de las variaciones en los tonos del mismo.
- Debido a las limitaciones que tiene GNURADIO se observa que no es capaz de cumplir conceptos que teóricamente son válidos, esto se reflejó en el caso de la señal rectangular, en la cual se obtuvo un ancho de banda finito lo que era contradictorio con el concepto teórico.

Referencias

[1] U. I. d. S. Homero Ortega, Oscar Reyes, "Comunicaciones digitales basadas en radio definida por software," p. 19, 2019.