**COMUNICACIONES II – 27145  
 PSD DE SEÑALES ALEATORIAS (GNURADIO)**

Fabián Yesid Moreno - 2191469

Fabián Leonardo Rincon Lizcano - 2174072

Florez Ramos Jose David - 2174241

**Abstract**

In the field of engineering, Power Spectral Density (PSD) emerges as an essential tool for data analysis and acquisition. This communications lab focuses on the fundamentals of PSD calculation and the different blocks provided by GNURadio. The student is trained to generate patterns that facilitate the calculation of the PSD. The goal of this lab is building functions from code blocks to produce averaging vectors, as well as combining GNURadio blocks for specific random signal applications.

**Resumen**

En el campo de la ingeniería, la Densidad Espectral de Potencia (PSD) emerge como una herramienta esencial para el análisis y adquisición de datos. Este laboratorio de comunicaciones se enfoca en los fundamentos del cálculo de la PSD y los distintos bloques proporcionados por GNURadio. el estudiante se capacita para generar patrones que faciliten el cálculo de la PSD. El objetivo de esta práctica es la construcción de funciones a partir de bloques de código para producir vectores de promedio, así como la combinación de bloques de GNURadio para aplicaciones específicas de señales aleatorias

1. **INTRODUCCIÓN**

la Densidad Espectral de Potencia (PSD) representa una herramienta fundamental para el análisis y la adquisición de datos en diversos campos de aplicación. En las comunicaciones, comprender y calcular la PSD para tener una mejor comprensión de la estructura espectral de las señales y la revelación de patrones ocultos en los datos temporales además de que el estudiante adquieran la capacidad de construir funciones utilizando bloques de código para producir vectores de promedio, y que sean capaces de combinar estos bloques de GNU-Radio de manera efectiva para aplicaciones específicas relacionadas con señales aleatorias

1. **OBJETIVO PRINCIPAL**

El objetivo general de este laboratorio de comunicaciones es capacitar a los estudiantes en la comprensión de los principios del cálculo de la Densidad Espectral de Potencia (PSD) y en el uso efectivo de los bloques proporcionados por GNURadio para este fin. Así poder revelar patrones ocultos en los datos temporales y aplicar estos conocimientos en el diseño y análisis de sistemas de comunicaciones.

**MARCO TEÓRICO**

La Densidad Espectral de Potencia (PSD) es una medida fundamental en el análisis de señales en el dominio de la frecuencia. Representa la distribución de potencia de una señal en el espectro de frecuencia, lo que proporciona información sobre la estructura espectral de la señal.

Para calcular la PSD de una señal, es fundamental comprender la Transformada de Fourier. Esta herramienta matemática permite descomponer una señal en sus componentes de frecuencia, revelando así su contenido espectral. Se estudian los principios básicos de la Transformada de Fourier y su relación con la PSD, así como las técnicas prácticas para su implementación computacional.

este laboratorio se necesita un conocimiento teorico de las señales aleatorias, la Transformada de Fourier y la PSD, así como el uso de GNURadio como herramienta práctica para el análisis y procesamiento de señales en el dominio de la frecuencia para un entendimiento y desarrollo correcto.

Principio del formulario

1. **METODOLOGÍA EXPERIMENTAL**

* Creación de la rama de desarrollo: Se creó una rama para el desarrollo de la práctica desde el repositorio en el navegador siguiendo el tutorial proporcionado. La rama se nombró “Practica\_2”.
* Descarga de cambios: En la estación de cómputo, se descargaron los cambios de las ramas incluidas utilizando el comando git pull en la terminal.
* Creación de carpetas: En la misma terminal, se creó una carpeta llamada “Practica\_2” y dos subcarpetas con los nombres “GNUradio” e “informe”.
* Uso de GNURadio y creación de informe: Se abrió la aplicación GNURadio y se guardaron los cambios del archivo creado en el directorio establecido para la práctica (Practica\_1/GNUradio). Además, en la carpeta Practica\_1/informe, se creó un archivo en Word para iniciar el informe.
* Implementación de bloques de Python: Se siguió el tutorial proporcionado para familiarizarse con el bloque de Python.
* Subida de cambios al repositorio: Se subieron los cambios al repositorio en la nube utilizando los comandos git add ., git commit -m "", y git push.
* Compartir el repositorio: El usuario que creó el repositorio compartió con sus compañeros de grupo para la edición de los contenidos.
* Creación de nuevas ramas: Se crearon nuevas ramas a partir de “Practica 1” para que todos los integrantes del grupo tuvieran acceso a la estructura. Cada integrante creó su propia rama con un nombre único.
* Descarga de cambios en las ramas: En la terminal de cada integrante, se ejecutó el comando git pull.
* Ingreso a la rama personal: Cada integrante ingresó a la rama perteneciente a su nombre utilizando el comando git checkout P1\_Nombre#.
* Implementación del algoritmo en el bloque de Python: Se utilizó el libro guía en la sección 1.2.0.1 e implementó en el bloque de Python el algoritmo sugerido.
* Implementación de bloques de acumulador y diferenciador: Se implementaron los bloques de acumulador y de diferenciador.
* Implementación de un bloque para mostrar estadísticas: Teniendo en cuenta lo aprendido, se implementó un bloque para mostrar la estadística vista en la clase teórica anterior y se sugirió una aplicación.
* Sugerencia de una aplicación y recreación de resultados: Se sugirió una aplicación de los conceptos vistos en clase y se recreó una vista de los resultados.
* Verificación de cambios locales: Se utilizó el comando git status para verificar que hubo cambios de manera local en la PC.
* Aceptación de cambios: Si se encontraron cambios en la ventana de comandos, se mostraron en color rojo los archivos que cambiaron (se agregaron o quitaron). Si se estuvo de acuerdo con esos cambios, se utilizó el comando git add ..
* Confirmación de cambios: Si no se encontraron errores, se ejecutó el comando git commit -m "mensaje". El “mensaje” fue un comentario que hizo referencia a los archivos que se subieron (o eliminaron) o por qué se hicieron los cambios.
* Subida de cambios a la nube: Se ejecutó el comando git push y se verificó que los cambios hayan sido subidos a la nube (plataforma GitHub en el navegador).

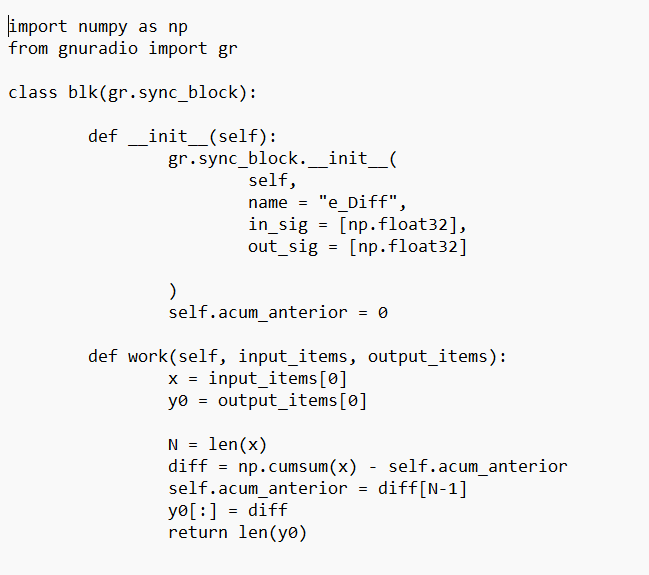
1. **ANÁLISIS DE RESULTADOS**

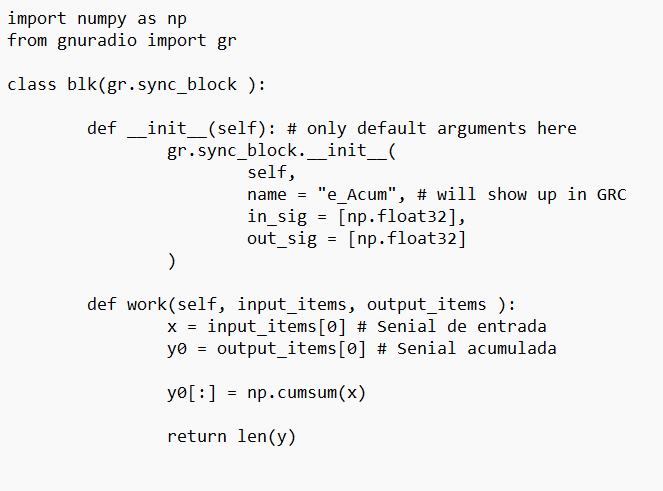
Durante la práctica, se implementaron varios bloques en Python, incluyendo un acumulador y un diferenciador. Estos bloques fueron probados y se observó su comportamiento. Además, se implementó un bloque para mostrar estadísticas basadas en los conceptos vistos en clase.

Los resultados obtenidos a partir de estos bloques fueron analizados y se encontró que funcionaban como se esperaba. El bloque de acumulador acumulaba los valores de entrada y el bloque de diferenciador calculaba la diferencia entre los valores de entrada consecutivos.

El bloque de "promedio de tiempos" está diseñado para suavizar una señal a lo largo del tiempo. Consiste en un bloque acumulador que acumula la señal de entrada y un diferenciador que divide la suma acumulada por el número de muestras acumuladas. En esencia, realiza un promedio ponderado de la señal a lo largo de un intervalo de tiempo.

Además, se sugirió una aplicación para los conceptos vistos en clase y se recreó una vista de los resultados. Esta aplicación demostró la utilidad de los conceptos aprendidos en clase y cómo pueden ser aplicados en situaciones reales.

  
Figura 1. código del bloque diferenciador

  
Figura 2. Código del bloque Acumulador

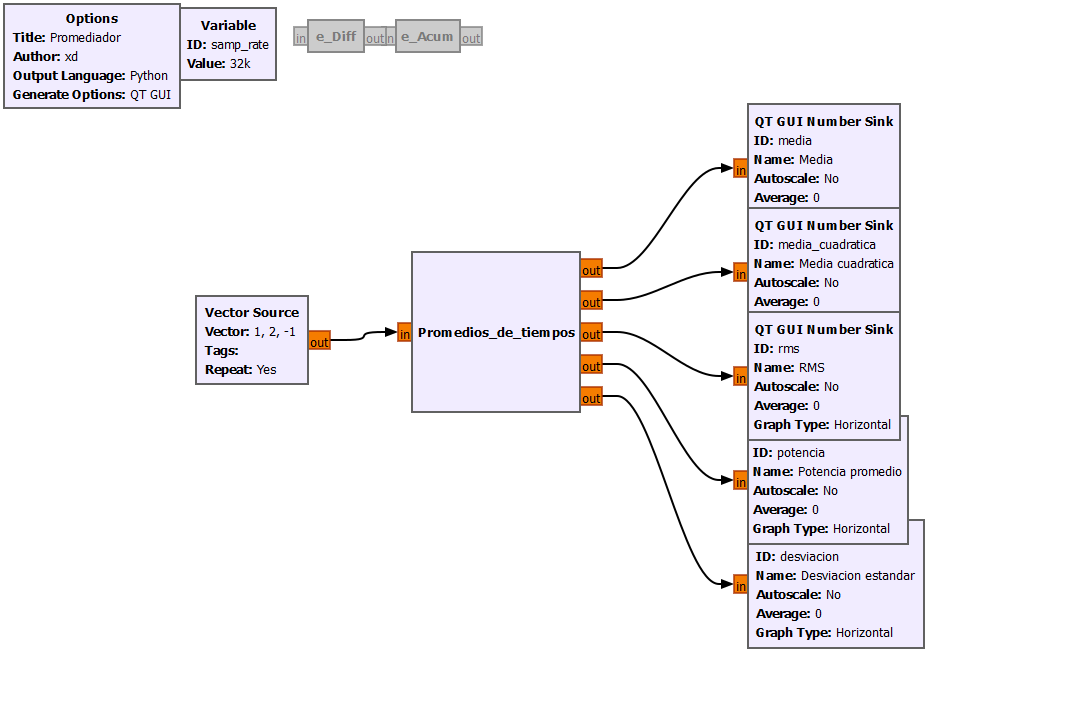
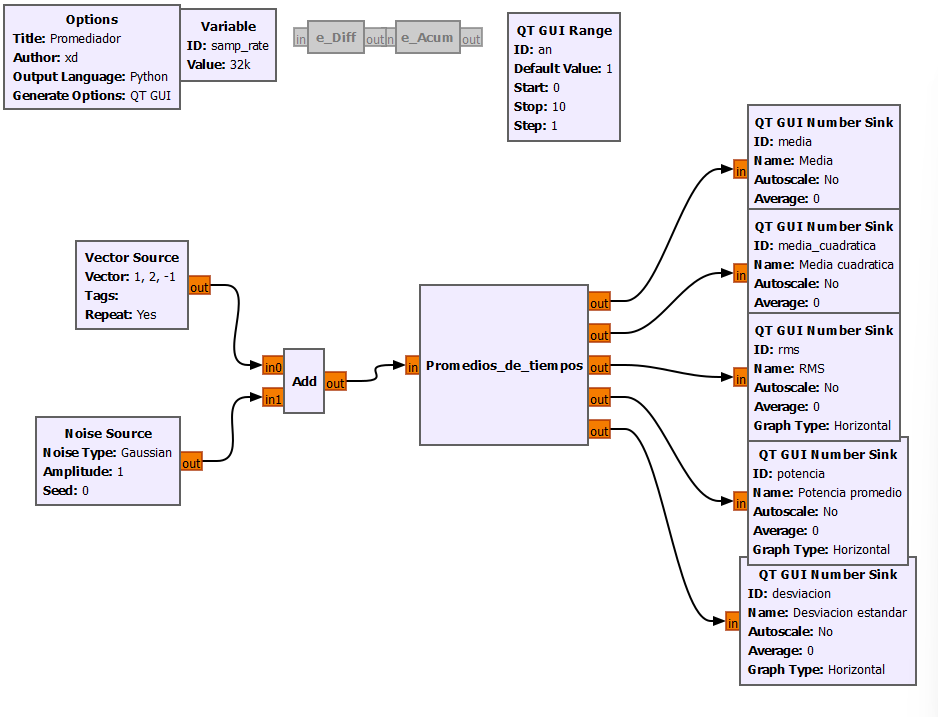
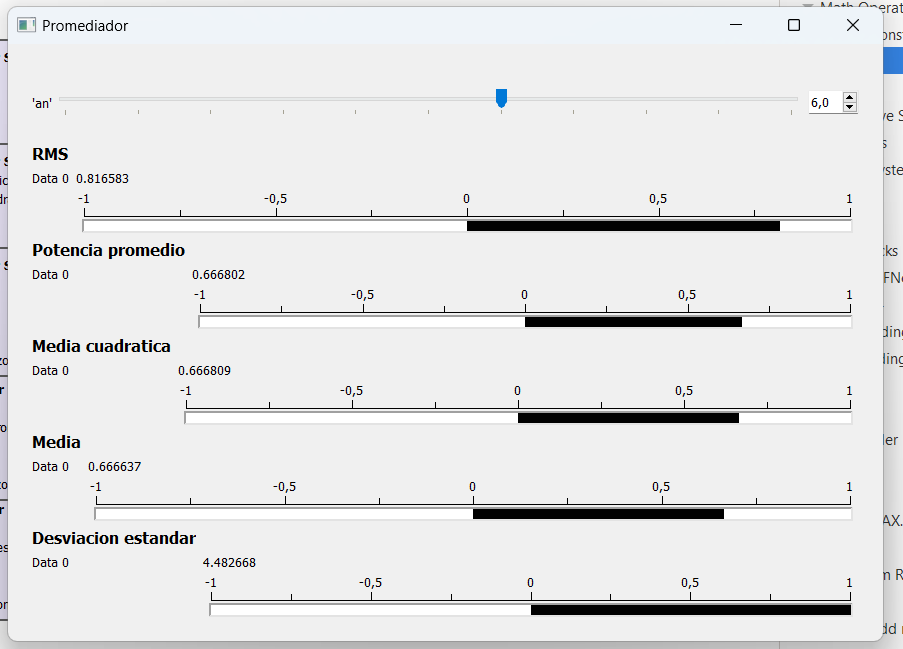


Figura 3. Bloque de Promedios

  
Figura 4. Bloque de ruido y promedio de tiempos

  
Figura 5. Visualización

**Posibles aplicaciones**

*Filtrado de Ruido:*

El bloque de "promedio de tiempos" puede ser utilizado para filtrar el ruido en una señal, ya que promedia varias muestras a lo largo del tiempo, suavizando las variaciones rápidas que pueden deberse al ruido. Se puede aplicar este bloque en situaciones donde la señal está afectada por ruido, como comunicaciones inalámbricas, y observar cómo mejora la calidad de la señal.

*Monitoreo de Sensores:*

En entornos donde los sensores pueden producir medidas ruidosas, el bloque de "promedio de tiempos" puede usarse para obtener lecturas más estables y representativas.

Se puede Implementar este bloque en sistemas de monitoreo ambiental, como sensores de temperatura o humedad, para mediciones más confiables.

*Seguimiento de Movimiento:*

Se puede utilizar el bloque en sistemas de seguimiento de vehículos o personas para mejorar la precisión de las posiciones estimadas.

*Análisis de Señales Biológicas:*

En señales biológicas como electrocardiogramas (ECG) o electroencefalogramas (EEG), el ruido puede afectar la interpretación de eventos importantes. El bloque ayuda a obtener una representación más clara.

**6. CONCLUSIONES**

De la práctica de laboratorio presentada previamente podemos concluir que:

* Aprendizaje de Git y GitHub: La práctica proporcionó una valiosa experiencia en el uso de Git y GitHub. Los comandos de Git como git pull, git add ., git commit -m "", y git push se utilizaron para gestionar el código y los cambios en el repositorio.
* Implementación de bloques en Python: La implementación de bloques en Python, incluyendo un acumulador y un diferenciador, proporcionó una comprensión práctica de estos conceptos.
* Aplicación de conceptos de clase: La sugerencia de una aplicación para los conceptos vistos en clase demostró cómo estos conceptos pueden ser aplicados en situaciones reales.
* Colaboración y trabajo en equipo: La práctica también destacó la importancia de la colaboración y el trabajo en equipo. Cada miembro del equipo trabajó en su propia rama y contribuyó al proyecto general.
* Verificación de cambios locales y subida a la nube: El proceso de verificación de cambios locales y subida de los mismos a la nube (plataforma GitHub en el navegador) reforzó la comprensión de cómo gestionar y compartir el código en un entorno de desarrollo colaborativo.

En resumen, la práctica fue una experiencia de aprendizaje valiosa que proporcionó una comprensión práctica de varios conceptos y herramientas importantes. También destacó la importancia de la colaboración y el trabajo en equipo en el desarrollo de software.

**Referencias**

[1] J. Mitola and G. Q. Maguire, “Cognitive radio: making software radios more personal,” in IEEE Personal Communications, vol. 6, no. 4, pp. 13-18, Aug. 1999.

[2] E. Blossom, “GNU radio: Tools for exploring the radio frequency spectrum,” in Linux Journal, vol. 2004, no. 122, Jun. 2004.

[3] P. A. Laplante and S. J. Ovaska, Real-Time Systems Design and Analysis: Tools for the Practitioner, 4th ed. Hoboken, NJ, USA: Wiley, 2011.

[4] GNU Radio, “Creating Your First Block,” GNU Radio Tutorials, [Online]. Available: https://wiki.gnuradio.org/index.php?title=Creating\_Your\_First\_Block.