

Práctica 1: PROGRAMACIÓN EN RADIO DEFINIDA POR SOFTWARE (GNURADIO)

JHORMAN MALDONADO REY-2192296

JOSÉ MANUEL DELGADO CARDENAS-2195131

JHAN CARLOS CARDENAS PARRA-2184218

https://github.com/D2-2192296-jhorman/CommII_A1_G10

28 de febrero del 2025

Abstract

This report shows the implementation of customised blocks in the GNU Radio tool for signal processing, in order to provide the necessary knowledge to be able to analyse and program software defined radios using GNU Radio, using theoretical concepts and applying them to practice, using SDR fundamentals.

Palabras clave: GNURadio, Estadística, Bloques.

1. Introducción

Se realizó esta práctica basándose en la programación orientada en radio definida por software haciendo uso de la herramienta GNU Radio, siendo una plataforma flexible para diseñar y probar algoritmos mediante el uso de bloques funcionales y personalizados, fortaleciendo así el manejo y habilidades en el uso de esta. El propósito principal fue entender el

comportamiento de las señales a lo largo del tiempo y aplicar estos conceptos a situaciones reales.

2. Metodología

La práctica se inició configurando el repositorio en GitHub a través de la terminal Linux del equipo del laboratorio, creando las ramas necesarias y organizándolas en subcarpetas específicas para esta primera etapa. En cuanto a la implementación del algoritmo en el bloque de Python, se tomó como referencia el libro guía en la sección 1.2.0.1, de donde se extrajeron los códigos pertinentes para construir los bloques y realizar las operaciones de promedio de tiempo (Ortega, 2019). Así mismo, se desarrollaron e implementaron los bloques correspondientes al acumulador y al diferenciador.

```
import numpy as np
from gnuradio import gr

class blk(gr.sync_block):

    def __init__(self):
        gr.sync_block.__init__(
            self,
            name = "e_Diff",
            in_sig = [np.float32],
            out_sig = [np.float32]
        )
        self.acum_anterior = 0

    def work(self, input_items, output_items):
        x = input_items[0]
        y0 = output_items[0]

        N = len(x)
        diff = np.cumsum(x) - self.acum_anterior
        self.acum_anterior = diff[N-1]
        y0[:] = diff
        return len(y0)
```

Figura 1: Código bloque diferenciador.

```
import numpy as np
from gnuradio import gr

class blk(gr.sync_block):

    def __init__(self): # only default arguments here
        gr.sync_block.__init__(
            self,
            name = "e_Acum", # will show up in GRC
            in_sig = [np.float32],
            out_sig = [np.float32]
        )

    def work(self, input_items, output_items):
        x = input_items[0] # Señal de entrada
        y0 = output_items[0] # Señal acumulada

        y0[:] = np.cumsum(x)
        return len(y)
```

Figura 2: Código bloque acumulador.

Con base en los conocimientos adquiridos, se desarrolló un bloque dedicado a visualizar las estadísticas abordadas en la clase teórica anterior, y se propuso su aplicación práctica.

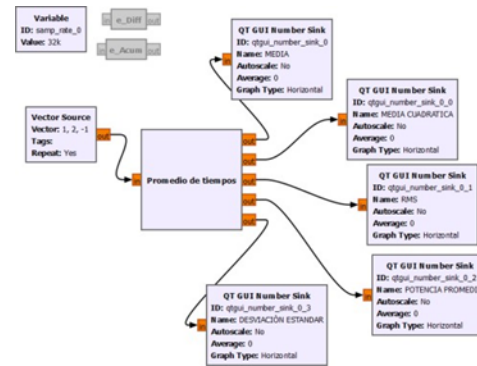


Figura 3: Diagrama del diferenciador.

Este consistió en desarrollar un bloque en GNU Radio para el cálculo de promedios temporales. Dicho bloque recibe una señal compuesta por la suma de un vector con una fuente de ruido, y genera cinco salidas: la media, la media cuadrática, el valor RMS, la potencia promedio y la desviación estándar. Estas métricas se visualizan utilizando el bloque QT GUI Number Sink.

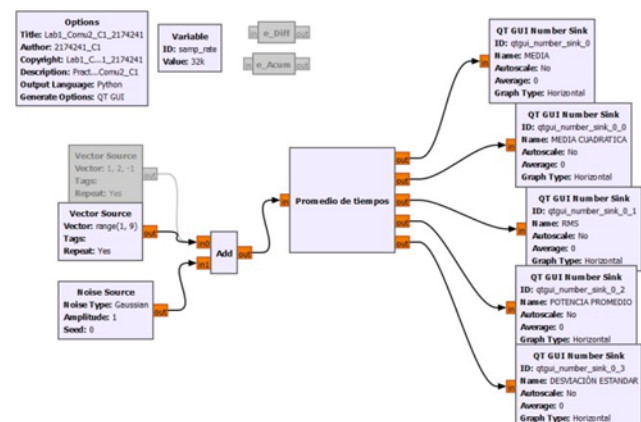


Figura 4 Diagrama de bloques en GNU Radio.

Finalizando de realizar los esquemas, y tomando las evidencias de cada uno, se guardó esta información en el repositorio en GitHub, en una rama dedicada a guardar los avances y resultados obtenidos “D2-2192296-jhorman/main”.

3. Análisis de Resultados

Para ver mejor el resultado de la implementación de los bloques, se utilizó un vector con de entrada [2, 3, -2] y después se probaron distintos rangos de 0 a 9. Los resultados obtenidos se compararon con las fórmulas teóricas y, en ambos casos, las salidas numéricas de los bloques de media, media cuadrática, RMS, potencia promedio y desviación estándar coincidieron con los valores esperados. Esto confirmó el correcto funcionamiento del bloque.

Resultados Obtenidos:

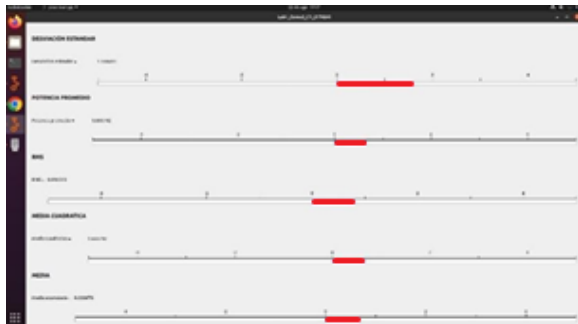


Figura 5 Resultado con vector de tres elementos.

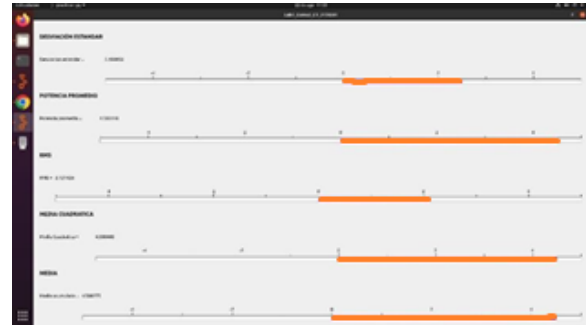


Figura 6 Resultado con números del 1 al 9.

- **Media:** Consistente con el valor esperado para las entradas.
- **Media cuadrática:** Resultado cercano a la teoría.
- **RMS:** Valores precisos de acuerdo con los datos ingresados.
- **Potencia promedio y desviación estándar:** Correctamente calculados.

En la figura 1 y la figura 2 se mostró el código de los bloques diferenciador y acumulador, respectivamente. La figura 3 presentó el diagrama de bloques para calcular el promedio de tiempos. Basándose en este esquema, en la figura 4 se incluyeron los bloques de entrada con el vector y el rango de números. Luego, en las figuras 5 y 6 se mostraron los resultados obtenidos para cada una de las entradas. Estos diagramas de bloques permitieron hacer los cálculos de medidas estadísticas vistas en clase, usando la opción del bloque de Python de GNU Radio.

5.Conclusiones

- Esta práctica permitió desarrollar habilidades esenciales en el uso de Git, GitHub y GNURadio, aplicando conceptos clave del procesamiento de señales y la gestión colaborativa del código.
- La experiencia destacó la importancia del trabajo en equipo, donde cada integrante aportó al desarrollo del proyecto cumpliendo con su respectiva tarea, lo que reforzó la colaboración en un entorno compartido.
- En cuanto a la implementación en GNU Radio, se trabajó con bloques como el acumulador y el diferenciador, lo que permitió analizar cómo las señales se modifican con el tiempo. Además, el uso del bloque de

promedios de tiempo hizo posible calcular y visualizar en tiempo real estadísticas clave como la media, media cuadrática, valor RMS, potencia promedio y desviación estándar. Estos análisis resultan fundamentales en aplicaciones como demodulación, filtrado de ruido y monitoreo de sistemas de comunicación.

- En general, la práctica no sólo reforzó conceptos teóricos, sino que brindó una perspectiva más clara sobre cómo estas herramientas se aplican en entornos reales para optimizar sistemas de comunicación.

Referencias

Ortega, H. (2019). *Comunicaciones digitales basadas en radio definida por software*. Bucaramanga: Editorial UIS.