

PROGRAMACIÓN EN RADIO DEFINIDA POR SOFTWARE (GNURADIO)

Juan David Vesga Gomez, 2211643, Estudiante Ing. Electrónica,
Elki Yesid Lozada Cabrera, 2204219, Estudiante Ing. Electrónica

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

Agosto 31, 2025

https://github.com/Juanvesga23/Comm11_A1_G1.git

Abstract

This report presents the development of the first laboratory practice of the Communications II course, focused on software-defined radio programming using GNU Radio. Three blocks were implemented in Python: accumulator, differentiator, and statistician, in order to analyze signals affected by noise. The methodology included managing repositories in GitHub, creating work environments in GNU Radio, and validating the blocks using test signals. The results demonstrated the correct functioning of the blocks, highlighting the ability of the accumulator to integrate the signal, the differentiator to highlight abrupt variations, and the statistical block to quantitatively characterize the effects of noise. In conclusion, the practice allowed for the strengthening of skills in digital signal processing and consolidated the link between theory and practical application in telecommunications.

1 Introducción

La radio definida por software (SDR) ha revolucionado las telecomunicaciones al permitir la implementación de sistemas flexibles y adaptables mediante software. En este contexto, GNU Radio se presenta como una de las plataformas más utilizadas, ya que ofrece un entorno modular con bloques predefinidos y la posibilidad de programar funcionalidades personalizadas en Python.

Su aplicación en el ámbito académico facilita la comprensión práctica de conceptos teóricos en comunicaciones, al permitir la implementación de algoritmos como acumulación, diferenciación y análisis estadístico de señales. Estos procesos no solo permiten caracterizar y comprender el comportamiento de las señales digitales, sino también explorar soluciones frente a fenómenos como el ruido.

El presente informe se centra en el uso de GNU Radio dentro del curso de Comunicaciones II, con el fin de fortalecer competencias en programación y procesamiento digital de señales, promoviendo el diseño y análisis de sistemas que integran teoría y práctica en el campo de las telecomunicaciones.

2 Metodología

Para el desarrollo del laboratorio se siguieron los pasos establecidos en la guía, aplicando tanto los conceptos de radio definida por software como las prácticas de control de versiones en GitHub. El procedimiento se estructuró de la siguiente manera:

- *Gestión del repositorio en GitHub:*

En la primera etapa, se creó una rama denominada Practica 1 a partir de la rama principal. Dentro de esta se generó un directorio con el mismo nombre, que incluyó dos subdirectorios: GNURadio e Informe. Posteriormente, cada integrante del grupo creó su propia rama derivada de Practica 1, lo que permitió trabajar de manera independiente y mantener un control adecuado de versiones.

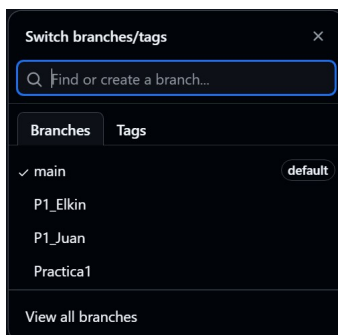


Fig. 1: Ramas creadas en el repositorio.

- **Implementación en GNU Radio:**

En la segunda etapa, se abrió GNU Radio en el entorno Linux y se configuraron los proyectos en el directorio correspondiente. Seguidamente, se desarrolló un bloque en Python siguiendo el tutorial oficial de GNU Radio [1] y las recomendaciones del libro guía [2]. Durante esta fase se implementaron tres algoritmos básicos: Acumulador (integrador), Diferenciador (derivador) y un Bloque de Estadísticas, encargado de calcular parámetros estadísticos fundamentales de la señal.

- **Diseño de aplicaciones:**

Finalmente, se llevaron a cabo pruebas de validación y aplicación con señales de entrada. Para ello, se utilizó una señal de tipo Vector Source conectada a un QT GUI Sink, lo que permitió la visualización de los resultados en tiempo real. Posteriormente, se introdujo una señal senoidal afectada por ruido gaussiano, la cual fue procesada mediante los bloques Acumulador (Figura 2) y Diferenciador (Figura 3). En las salidas se observaron transformaciones notables, que fueron evaluadas aplicando las funciones estadísticas implementadas (Figura 4), con el fin de analizar los efectos del ruido y su mitigación.

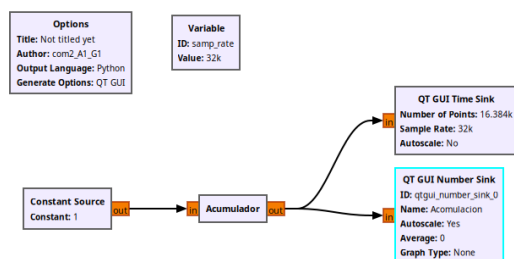


Fig. 2: Bloque acumulador

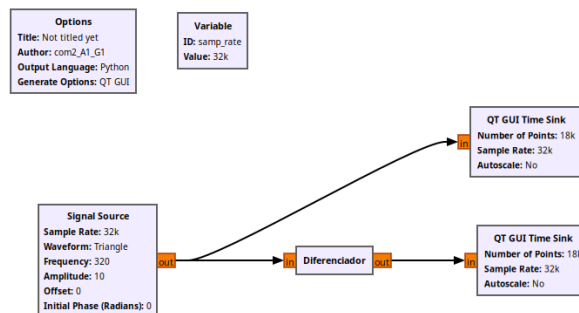


Fig. 3: Bloque diferenciador

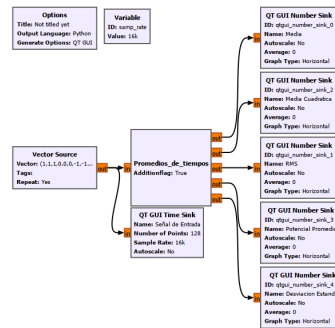


Fig. 4: Bloque promedios de tiempos.

3 Análisis de resultados

En primer lugar, se analizó la señal de acumulación, en la cual se evidenció un crecimiento progresivo de la amplitud a medida que se sumaban los valores de entrada. Este comportamiento confirma que el bloque cumple su función de integrar de manera discreta la señal, generando una forma de onda que aumenta de manera lineal y reinicia su valor al alcanzar un límite (Figura 5).

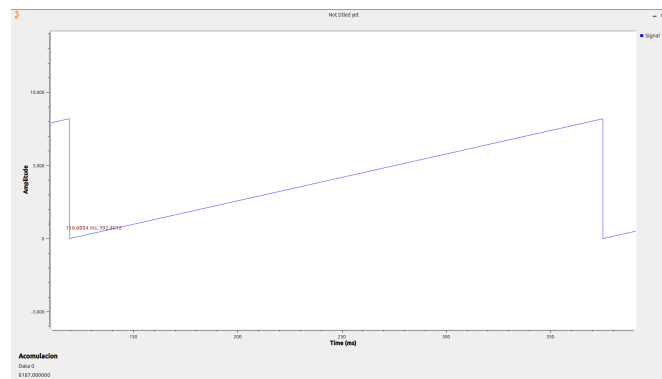


Fig. 5: Grafica acumulador.

Posteriormente, al aplicar el bloque diferenciador, la señal de salida mostró los cambios abruptos de pendiente de la señal original. Se observó que la salida enfatiza las variaciones rápidas, lo que valida su utilidad en la detección de transiciones y en el análisis de la dinámica de la señal. En comparación con la entrada, la salida presentó amplitudes más sensibles a las variaciones instantáneas, confirmando el efecto esperado (Figura 6).

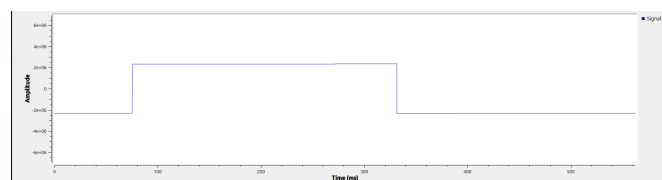


Fig. 6: Grafica diferenciador.

Finalmente, se implementaron bloques de análisis estadístico que permitieron calcular parámetros como la media, RMS, potencia promedio, desviación estándar y media cuadrática. Los valores obtenidos (media = 0.66, desviación estándar = 1.24, RMS = 0.81) reflejan una caracterización cuantitativa de la señal. En particular, la media positiva

indica un leve sesgo en la señal, mientras que la desviación estándar muestra una dispersión moderada respecto a la media. Estos resultados son fundamentales para el tratamiento de señales reales, ya que permiten cuantificar el efecto del ruido y evaluar la estabilidad del sistema (Figura 7).

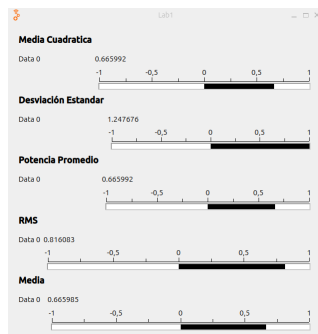


Fig. 7: Valores promedio.

4 Conclusiones

- La implementación del bloque de acumulación permitió comprobar su función como un integrador discreto, generando un crecimiento progresivo en la amplitud de la señal hasta alcanzar un límite, lo cual concuerda con el comportamiento teórico esperado.
- El bloque diferenciador demostró su utilidad en la detección de cambios abruptos y variaciones rápidas de la señal, resaltando las transiciones y confirmando su aplicación en análisis de dinámica de señales en sistemas de comunicaciones.
- El uso del bloque de estadísticas posibilitó la caracterización cuantitativa de la señal mediante parámetros como media, desviación estándar, potencia promedio y RMS. Estos indicadores evidenciaron la presencia de un leve sesgo en la señal y una dispersión moderada, mostrando la importancia de la estadística en la evaluación de calidad y estabilidad de señales.
- Finalmente, se demostró que la combinación de bloques matemáticos y estadísticos constituye una estrategia eficaz para el procesamiento y análisis de señales en presencia de ruido, ofreciendo bases sólidas para el diseño de aplicaciones reales en telecomunicaciones.

References

- [1] "Creating your first block," GNU Radio Wiki, May 4 2024, tutorial on creating a signal processing block with Embedded Python Block. [Online]. Available: https://wiki.gnuradio.org/index.php?title=Creating_Your_First_Block
- [2] H. O. Boada and O. M. R. Torres, *Comunicaciones Digitales basadas en radio definida por software*. Bucaramanga, Colombia: Editorial UIS, 2019, primera edición. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones, Universidad Industrial de Santander.