

Práctica I: Programación en radio definida por software

Samantha Lucía Triana Toloza - 2212249

Jose Alejandro Barrios Pico - 2212278

Repositorio: https://github.com/Josebarr20/CommII_G6_JBST/tree/Practica_1

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

31 de agosto de 2025

Abstract

The purpose of the practice was to develop skills in programming functions within GNU Radio through the implementation of custom blocks and the use of GitHub as a collaborative version control tool. Fundamental aspects of real-time systems and software-defined radio were identified, Python block functions were designed, and their results were evaluated alongside other processing blocks. Additionally, algorithms such as accumulator and differentiator blocks were implemented. Finally, practical applications of the studied concepts were proposed, and the correct integration of contributions into the shared repository was verified.

Index Terms— GNU Radio, SDR, Python blocks, GitHub, Accumulator, Differentiator.

Resumen

La práctica tuvo como propósito desarrollar competencias en la programación de funciones dentro de GNU Radio, a partir de la implementación de bloques personalizados y el uso de GitHub como herramienta de control de versiones colaborativo. Se identificaron aspectos fundamentales de los sistemas en tiempo real y de la radio definida por software, se diseñaron funciones en bloques de Python y se evaluaron sus resultados con otros bloques de procesamiento. Adicionalmente, se implementaron algoritmos como bloques de acumulador y diferenciador. Finalmente, se propusieron aplicaciones prácticas de los conceptos estudiados y se verificó la correcta integración de los aportes en el repositorio compartido.

1. Introducción

El avance de la radio definida por software (SDR) ha permitido que los sistemas de comunicación puedan

ser diseñados y modificados de manera flexible mediante programación. GNU Radio se ha consolidado como una de las principales plataformas de desarrollo, al ofrecer entornos modulares donde es posible combinar bloques ya establecidos con algoritmos personalizados. En este contexto, la práctica se enfocó en la creación y adaptación de bloques en GNU Radio, integrando tanto el uso de interfaces gráficas como la programación de algoritmos personalizados. Además, se gestionó el trabajo colaborativo mediante los comandos de la terminal de Linux asociados a Git, lo que permitió organizar versiones, administrar ramas de desarrollo y consolidar los aportes en un repositorio común de GitHub. Con ello, se fortalecieron competencias en programación, gestión de proyectos y aplicación práctica de la estadística en señales afectadas por ruido.

2. Metodología

La práctica se llevó a cabo siguiendo una serie de etapas organizadas que permitieron tanto la implementación técnica en GNU Radio como la gestión colaborativa en GitHub:

2.1. Configuración del entorno de trabajo

Se creó una rama en GitHub denominada *Practica_1* a partir de la rama principal. En esta se organizaron los directorios requeridos y se configuró el acceso mediante comandos de Git en la terminal de Linux, asegurando la estructura inicial del proyecto.

2.2. Diseño e implementación en GNU Radio

Se construyeron bloques de prueba en la interfaz gráfica y se programó un bloque en Python. Para la validación se incluyeron componentes como *vector source* y *QT GUI sink*, verificando así el correcto flujo de señales.

2.3. Trabajo en ramas individuales

A partir de *Practica_1*, se crearon ramas adicionales donde se desarrollaron algoritmos específicos. En cada una se implementaron los bloques de acumulador, diferenciador y un bloque estadístico.

2.4. Control de versiones y sincronización de cambios

Se emplearon los comandos de Git en la terminal de Linux (`git status`, `git add`, `git commit`, `git push`) para registrar las modificaciones, mantener un control de versiones eficiente y consolidar los aportes de cada participante en el repositorio compartido.

2.5. Aplicación práctica con ruido

Se propuso un escenario de prueba en el que se introdujo ruido a las señales procesadas. Posteriormente, se aplicaron los bloques estadísticos desarrollados para analizar el comportamiento del sistema y evaluar posibles soluciones al problema de interferencia.

2.6. Documentación y consolidación del informe

Los resultados obtenidos fueron documentados en un informe estructurado, el cual se gestionó inicialmente en la rama principal y posteriormente se integró mediante *merge* con las demás ramas de trabajo.

3. Análisis y tratamiento de datos

El análisis de señales digitales en GNU Radio permite validar experimentalmente conceptos teóricos del procesamiento digital, como la integración y la diferenciación discreta. Estas operaciones, además de ser herramientas matemáticas fundamentales, cumplen un papel esencial en aplicaciones de comunicaciones modernas, ya que permiten caracterizar, transformar y filtrar señales en distintos escenarios. Con el fin de observar sus efectos, se implementaron bloques de acumulador y

diferenciador aplicados a una señal cuadrada generada mediante un *vector source*.

En el caso del **acumulador**, la señal cuadrada de entrada fue integrada de manera progresiva, lo que produjo un crecimiento lineal en las regiones de nivel positivo y una disminución en las regiones de nivel negativo, como se muestra en la Fig. 1. El resultado fue una señal de tipo triangular que refleja la acumulación de energía a lo largo del tiempo. Este comportamiento es coherente con la interpretación matemática de la integral discreta, la cual suaviza la señal original y permite analizar tendencias globales en lugar de variaciones instantáneas. En aplicaciones de comunicaciones, este tipo de procesamiento es fundamental para estimar la energía de la señal, detectar patrones acumulativos y realizar tareas de sincronización.

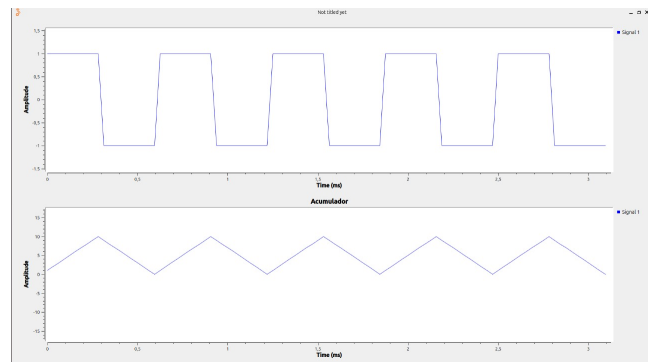


Fig. 1: Señal de entrada y salida del Acumulador procesado en GNURadio.

Por otro lado, el **diferenciador** resaltó los cambios abruptos en la señal cuadrada, como se muestra en la Fig. 2. Mientras que los niveles constantes no presentaron variación significativa, en las transiciones de la onda cuadrada se observaron picos positivos y negativos que corresponden a los saltos entre niveles. El resultado obtenido es consistente con la operación de derivada discreta, que enfatiza los puntos de cambio en el dominio temporal. Esta propiedad es ampliamente utilizada en la detección de bordes en procesamiento digital, en la identificación de transiciones rápidas y en aplicaciones de comunicaciones donde es necesario extraer información de la modulación por cambios de amplitud o fase.

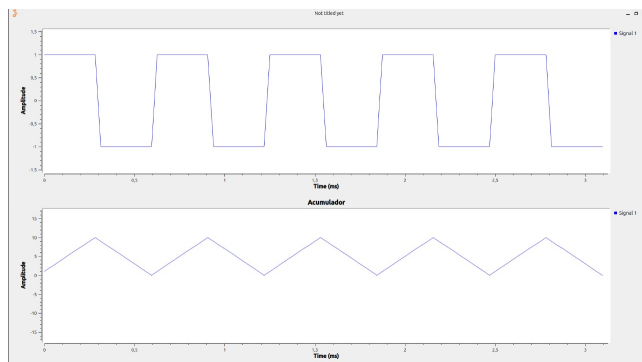


Fig. 2: Señal de entrada y salida del diferenciador procesado en GNURadio.

Adicionalmente, se comparó la señal original con las señales procesadas. El acumulador permitió obtener una visión global del comportamiento acumulativo de la señal, mientras que el diferenciador facilitó la localización precisa de los puntos de cambio. De esta manera, ambos bloques se complementaron, mostrando que la combinación de integración y diferenciación proporcionara herramientas poderosas para la caracterización de señales.

Al considerar escenarios reales de comunicaciones, la presencia de ruido puede alterar significativamente la forma de onda de las señales digitales. En este sentido, los bloques diseñados pueden servir como filtros analíticos: el acumulador ayuda a suavizar fluctuaciones aleatorias, mientras que el diferenciador puede resaltar la presencia de interferencias rápidas. Estos resultados son aplicables en sistemas modernos como receptores SDR [1], detección de tramas en protocolos inalámbricos y análisis estadístico de calidad de señal.

4. Conclusiones

El desarrollo de la práctica permitió comprender de manera aplicada los fundamentos de la radio definida por software (SDR) mediante el uso de GNU Radio como plataforma de diseño y experimentación. La implementación de bloques personalizados en Python, junto con la integración de bloques acumulador, diferenciador y estadístico, demostró la versatilidad de esta herramienta

para el análisis y procesamiento de señales en tiempo real.

El tratamiento de señales afectadas por ruido evidenció la importancia de aplicar técnicas estadísticas en la caracterización y mejora de la calidad de la información transmitida. Estos resultados resaltan la relevancia de la estadística y del procesamiento digital de señales (DSP) en escenarios de comunicaciones modernas, donde la presencia de interferencias y perturbaciones es una condición inevitable. Asimismo, la práctica permitió establecer una conexión entre la teoría de sistemas de comunicación y su implementación en un entorno flexible, resaltando el papel de la SDR en aplicaciones actuales como la radio cognitiva, la simulación de sistemas inalámbricos, la detección de espectro y el desarrollo de prototipos para tecnologías emergentes como 5G y el Internet de las Cosas (IoT) [2].

De manera general, GNU Radio se consolida como una herramienta fundamental en la formación y práctica de la ingeniería de telecomunicaciones, al brindar un entorno que integra programación, análisis de señales y validación experimental de conceptos, todo ello con un alto grado de aplicabilidad en los retos tecnológicos contemporáneos [3].

Referencias

- [1] D. Joosens, N. BniLam, R. Berkvens, and M. Weyn, "Software-defined radio-based internet of things communication systems: An application for the dash7 alliance protocol," *Applied Sciences*, vol. 15, no. 1, 2025. [Online]. Available: <https://www.mdpi.com/2076-3417/15/1/333>
- [2] J. Liang, H. Chen, and S. C. Liew, "Design and implementation of time-sensitive wireless iot networks on software-defined radio," 2020. [Online]. Available: <https://arxiv.org/abs/2006.09970>
- [3] M. Rowe, "Sdr adds flexibility to communications systems - 5g technology world," 7 2021, [Online; accessed 2025-08-25]. [Online]. Available: <https://www.5gtechnologyworld.com/sdr-adds-flexibility-to-communications-systems/>