

# Práctica 1: Implementación y Análisis de Bloques en GNU Radio para Acumulación y Diferenciación de Señales

1<sup>st</sup> Ricardo Mancipe Florez - 2225073  
*ingenieria electronica*  
*universidad industrial de santander*  
Bucaramanga, santander

2<sup>nd</sup> Bryan Steven Ayala Riveros - 2194676  
*ingenieria electronica*  
*universidad industrial de santander*  
Bucaramanga, santander

3<sup>th</sup> Daniel Gonzales Gamba - 2215727  
*ingenieria electronica*  
*universidad industrial de santander*  
Bucaramanga, santander

**Abstract**—This paper presents the implementation and validation of custom GNU Radio Python blocks for real-time stream processing. An Accumulator and a Differentiator were developed and paired with a lightweight Statistics routine to extract selected time-domain statistical parameters from the processed streams. The results demonstrate the effectiveness of the approach and provide a practical baseline for subsequent laboratory exercises and for basic analysis of software-defined radio systems.

## I. INTRODUCCION

La construcción de un sistema de comunicación exige comprender las características de la señal para aplicar tratamientos adecuados y mejorar la eficiencia. En señales determinísticas se identifican con relativa claridad parámetros como la amplitud, la frecuencia, la fase y el período; sin embargo, cuando el comportamiento es menos predecible conviene apoyarse en parámetros estadísticos de tiempo que sintetizan la variación de la señal en ventanas de observación y facilitan su descripción cuantitativa. En este contexto formativo, el propósito es desarrollar competencias para programar funciones orientadas al análisis temporal. De manera específica, se busca reconocer los aspectos básicos del procesamiento en tiempo y poner en práctica funciones que permitan obtener y evaluar parámetros estadísticos de la señal dentro de un esquema de programación por bloques.

Para ello se emplean tres componentes mínimos: un acumulador, que suma sucesivamente las muestras y facilita observar tendencias o cambios lentos; un diferenciador, que calcula la diferencia entre muestras consecutivas y resalta variaciones rápidas; y un bloque de estadísticos, que estima medidas temporales seleccionadas como media cuadrática, valor RMS,

Potencia promedio y desviación estándar. En conjunto, estos elementos simplifican la observación y el análisis básico en el dominio del tiempo.

## II. METODOLOGIA

Inicialmente se creó la rama Practica\_1 en el repositorio y, dentro de ella, dos carpetas: GNURadio e informes. En GNURadio se guardaron los códigos en Python, los diagramas de bloques y algunas capturas de pantalla de los resultados. En informes se archivó el documento correspondiente a la práctica. Por último, cada integrante del equipo creó su propia rama con el formato, p1\_bryan, p1\_ricardo y p1\_daniel, para trabajo autónomo; al final, los cambios se integraron en Practica\_1.

Antes de usar bloques de procesamiento, se verificó la señal binaria de media cero conectando una fuente vectorial al cual se introdujeron los valores de la forma [1,1,1,-1,-1,-1,1,1,-1,-1,...] donde posteriormente se conecto a un visualizador de tiempo para verificar su comportamiento.

### A. Bloque acumulador

Se implementó un bloque acumulador en Python apoyándonos en la sección 1.2.0.1 del libro guía, su objetivo es sumar muestra a muestra la señal de entrada para evidenciar tendencias y cambios lentos a lo largo del tiempo. Para probarlo, la fuente se conectó al acumulador y su salida a un visualizador de tiempo, comparando la señal original con la acumulada.

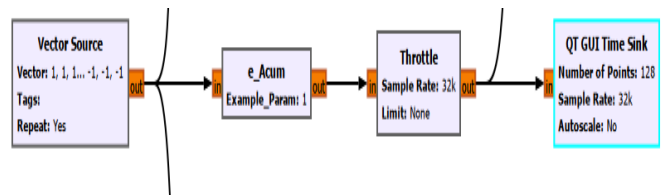


Fig. 2.1 Diagrama de flujo en GNU Radio del bloque acumulador.

### B. Bloque diferenciador

De igual forma se implementó el bloque diferenciador en Python también apoyados en la sección 1.2.0.1 del libro guía, conservando el último valor procesado para dar continuidad entre llamadas. Este bloque calcula la diferencia entre muestras consecutivas, lo que resalta.

Variaciones rápidas o pendientes. La prueba se hizo conectando la fuente al diferenciador y su salida a un visualizador de tiempo para observar la relación entrada-salida.

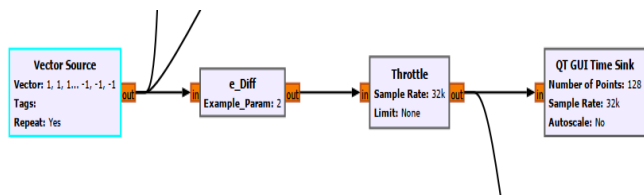


Fig. 2.2 Diagrama de flujo en GNU Radio del bloque diferenciador.

### C. Bloque promedios de tiempos

Se construyó un bloque con una entrada y cinco salidas para obtener parámetros en el dominio del tiempo. Las salidas corresponden a media, media cuadrática, valor RMS, potencia promedio y desviación estándar.

El bloque usa sumas acumuladas y un conteo total de muestras para que los cálculos se mantengan a lo largo del flujo. La conexión fue directa desde la fuente hacia el bloque y de cada salida a su visualizador, de modo que quedaran identificadas las cinco medidas.

De forma general: la media cuadrática promedia los valores al cuadrado; el RMS es la raíz de esa media y representa un nivel eficaz; la potencia promedio se interpreta como el cuadrado del RMS; y la desviación estándar mide la dispersión respecto a la media.

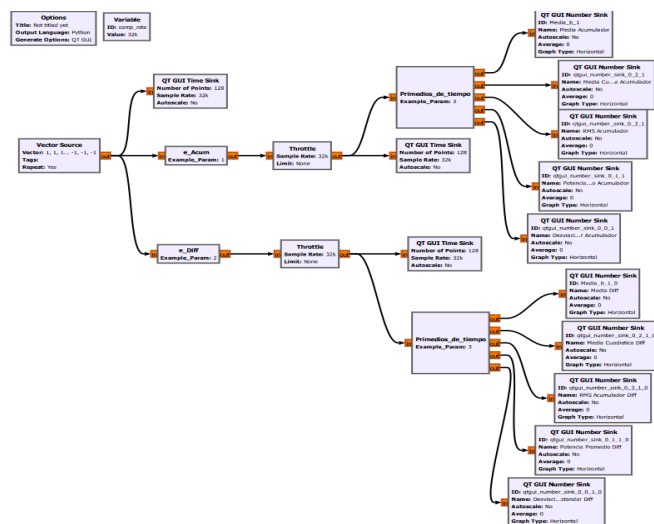


Fig. 2.3 Diagrama de flujo en GNU Radio Del bloque promedios del tiempo.

### III. ANALISIS DE RESULTADOS

En la Fig. 2.1 se observa la señal de entrada junto con la salida del acumulador. Al acumular las muestras, la salida actúa como una integración discreta: en cada tramo donde la entrada permanece constante, la salida evoluciona de forma lineal; cuando cambia el nivel, la pendiente cambia de signo. El resultado global adopta una forma triangular, que es lo esperable al integrar por tramos una señal cuadrada bipolar. Cuando la secuencia es simétrica, la forma triangular tiende a regresar al entorno del nivel inicial al completar el patrón.

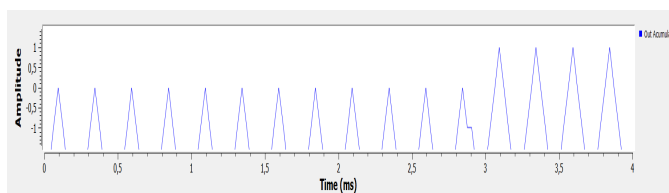


Fig. 3.1 Resultado bloque acumulador.

La Fig. 2.2 muestra la señal de entrada y la salida del diferenciador. En los intervalos en que la entrada se mantiene constante, la salida permanece cercana a cero; en los instantes de cambio de nivel aparecen picos de signo opuesto que marcan las transiciones. Este comportamiento es el esperado al diferenciar una señal cuadrada, ya que la salida responde a los saltos de la entrada y su amplitud es proporcional al cambio entre niveles.

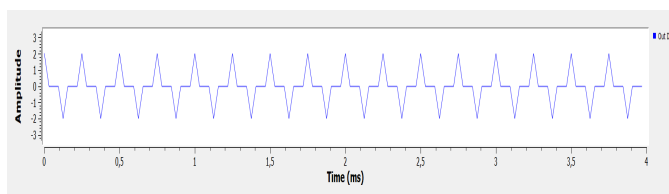


Fig. 3.2 Resultado bloque diferenciador.

En la Fig. 2.3 se ilustran las cinco salidas del bloque de estadísticos

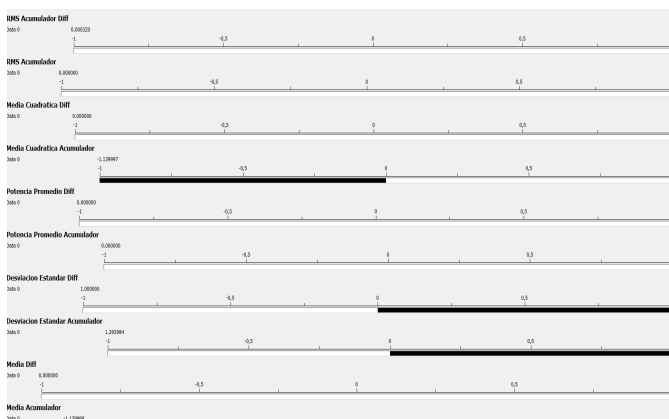


Fig. 3.3 Resultados estadísticos de la señal.

#### IV. CONCLUSIONES

Al revisar el libro guía y llevarlo a la práctica, entendimos mejor cómo programar bloques en Python para el análisis en tiempo. Nos apoyamos en las secciones 1.1–2.7 para las definiciones y en la 1.2.0.1 como referencia de cómo estructurar el bloque. Con pequeños ajustes, los bloques corrieron de forma continua y pudimos ver claramente qué hace cada etapa sobre la señal. Esto cumple el objetivo de desarrollar competencias para interactuar con la programación de funciones y evaluar sus salidas.

Los bloques acumulador y diferenciador nos facilitan el análisis porque ofrecen dos miradas complementarias de la señal. El acumulador muestra su tendencia y nivel a lo largo del tiempo, útil para interpretar la media cuadrática, el RMS y la potencia. El diferenciador resalta las transiciones y la variabilidad, lo cual se refleja en la desviación estándar.

Cuidar la media cero fue clave. Con el vector inicial, los bloques no se comportaban bien y podían saturarse; tras ajustar la secuencia y comprobar la media, las salidas y las medidas se entendieron correctamente. En adelante, primero revisar la fuente y luego procesar.

#### V. REFERENCIAS

[1] Ortega Boada, H.; Reyes Torres, O. M. Comunicaciones digitales basadas en radio definida por software. Volumen II. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones, Universidad Industrial de Santander (UIS). Secciones usadas: 1.1–2.7 y referencia operativa 1.2.0.1.