

Práctica 1: Implementación de Acumulador, Diferenciador y Bloque de Promedio del Tiempo en GNU Radio

Nestor Santiago Ulloa Reyes - 2215739
Andrés Felipe Quintero Lozano - 2204655

https://github.com/nsantiagou/Comm2_LabA1_G2_Ulloa_Quintero

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones

Universidad Industrial de Santander

Abstract

This report presents the implementation and analysis of three custom blocks in GNU Radio using Python: an accumulator, a differentiator, and a time-average block. The accumulator performed the discrete integration of the input signals, the differentiator highlighted slope variations and abrupt transitions, and the time-average block enabled the real-time calculation of statistical parameters such as mean, variance, power, and RMS value. To validate their performance, deterministic signals such as square and sinusoidal waves were used, and different levels of Gaussian noise were later added. The results show that the implemented blocks allow an effective analysis of both deterministic and probabilistic parameters, providing a versatile tool for improving signal characterization in software-defined radio systems.

1 Introducción

En los sistemas de telecomunicaciones resulta fundamental comprender y analizar los parámetros básicos de las señales, pues de ellos depende el correcto diseño, tratamiento y optimización del proceso de transmisión. Características como la amplitud, frecuencia, fase y período permiten describir el comportamiento determinístico de una señal; sin embargo, en entornos reales, las señales también están afectadas por ruido y comportamientos aleatorios, lo que exige el uso de herramientas estadísticas que faciliten su caracterización.

Una manera de abordar este análisis consiste en implementar bloques personalizados en GNU Radio utilizando Python, lo que ofrece mayor flexibilidad frente a los bloques predeterminados del software. Tal como se expone en la literatura especializada [1], la programación de bloques en GNU Radio constituye una herramienta fundamental para el estudio de parámetros determinísticos y probabilísticos de las señales, permitiendo no solo su caracterización, sino también la optimización de los sistemas de comunicación.

En este contexto, se desarrollaron tres bloques: el acumulador, encargado de realizar la suma discreta de muestras; el diferenciador, que calcula la variación de la señal resaltando pendientes y transiciones; y el bloque de promedio del tiempo, diseñado para estimar en tiempo real parámetros probabilísticos como la media, la media cuadrática, el valor RMS, la potencia promedio y la desviación estándar.

Con estos bloques es posible estudiar simultáneamente transformaciones matemáticas elementales y características estadísticas de las señales, tanto en condiciones ideales como en presencia de ruido gaussiano. De esta forma, la práctica constituye una aproximación integral al análisis de señales determinísticas y aleatorias, además de evidenciar la importancia del uso de programación en GNU Radio para la adaptación y mejora de sistemas de comunicación modernos.

2 Metodología

Se implementaron dos bloques de Python en el GNU-Radio: un acumulador y un diferenciador, en base de lo propuesto en el libro guía [1], de allí usamos los códigos para implementarlos en los respectivos bloques.

• Bloque acumulador

El bloque de Python denominado **Accumulator** fue programado con el objetivo de realizar la integración discreta de la señal de entrada, acumulando el valor de cada muestra de forma progresiva. El comportamiento del bloque se describe matemáticamente mediante la siguiente ecuación:

$$y[n] = y[n-1] + x[n]$$

donde $x[n]$ corresponde a la muestra actual de la señal de entrada y $y[n]$ representa la salida acumulada.

Para la validación del bloque se utilizó una señal determinística de tipo cuadrada, lo que permitió verificar cómo el proceso de acumulación transforma la forma de onda original. En este caso, se observó que la señal acumulada adquiere un comportamiento consistente con la teoría, mostrando esa señal característica de forma triangular.

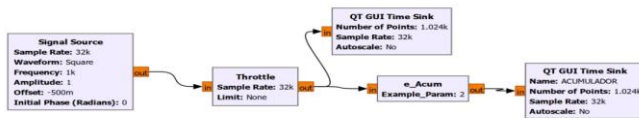


Figura 2.1, diagrama de flujo GNU Radio acumulador

El bloque se integró en un diagrama de flujo en GNU Radio junto con la fuente de señal y un visualizador en tiempo real, lo cual permitió comparar directamente la señal de entrada y la señal procesada por el acumulador, esto se ve más evidenciado en el apartado de resultados.

• Bloque diferenciador

El bloque de Python denominado Differentiator fue implementado con el fin de calcular la variación entre muestras consecutivas de la señal de entrada. De esta manera, el bloque resalta los cambios abruptos de la señal y permite analizar la pendiente de su evolución en el tiempo. El comportamiento matemático del bloque puede expresarse como:

$$y[n] = x[n] + x[n - 1]$$

donde $x[n]$ corresponde a la muestra actual de la señal de entrada y $y[n]$ representa la diferencia obtenida.

Para validar su funcionamiento se empleó una señal determinística de tipo también cuadrada. De acuerdo con la teoría, al diferenciar una señal cuadrada se obtiene un tren de impulsos. Los resultados obtenidos en GNU Radio permitieron verificar este comportamiento, observando claramente cómo el bloque resalta las variaciones rápidas

de la señal de entrada.

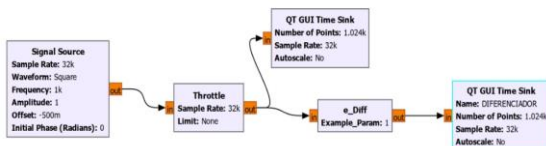


Figura 2.2, diagrama de flujo GNU Radio diferenciador

El bloque se integró en un diagrama de flujo dentro de GNU Radio junto con la fuente de señal

correspondiente y un bloque de visualización en tiempo real. De esta manera, fue posible comparar en paralelo la señal original y su versión diferenciada, contrastando teoría y práctica.

El análisis realizado confirma que el bloque diferenciador es una herramienta fundamental para resaltar cambios en la señal y constituye una base práctica para el estudio de transformaciones matemáticas aplicadas a sistemas de comunicación, esto se ve más evidenciado en el apartado de resultados.

• Bloque promedio del tiempo

El bloque de Python denominado Time Average fue implementado para calcular parámetros estadísticos de la señal de entrada en tiempo real. A diferencia de los bloques acumulador y diferenciador, que realizan transformaciones matemáticas directas sobre la forma de onda, este bloque está orientado al análisis probabilístico de la señal, permitiendo obtener medidas de tendencia central y dispersión.

Entre los parámetros calculados por el bloque se encuentran:

- **Media (μ):** valor promedio de las muestras en un intervalo de tiempo.
- **Potencia promedio (P_p):** estimación de la energía de la señal en función del tiempo.
- **Valor cuadrático medio (RMS):** medida de la amplitud eficaz de la señal.
- **Varianza (σ^2):** indicador de la dispersión respecto a la media.
- **Desviación estándar (σ):** raíz cuadrada de la varianza, útil para cuantificar la magnitud de las fluctuaciones.

Para validar el funcionamiento del bloque se utilizó una señal de onda cuadrada, a la que se añadió ruido gaussiano con diferentes niveles de potencia. De esta forma, fue posible analizar cómo los parámetros estadísticos varían dependiendo del nivel de perturbación presente en el sistema.

(Cuadrada), su versión acumulada y diferenciada.

En la Fig. 3.2 se observa el comportamiento esperado de ambos bloques: la señal de entrada (arriba) es una onda cuadrada; el **Diferenciador** (centro) produce pulsos bipolares en cada transición —coincidiendo con la derivada temporal— sin componente continua apreciable, mientras que el **Acumulador** (abajo) integra las muestras y genera una onda triangular cuya pendiente cambia según el nivel de la cuadrada, efecto potenciado por el offset aplicado. Los *number sinks* respaldan estas observaciones: la salida del acumulador presenta una media diferente de cero (≈ -1.07) y una desviación estándar notable (≈ 1.25), mientras que la media y el RMS del diferenciador se mantienen próximos a cero. Estos resultados validan la implementación y resaltan la importancia de controlar la componente DC u offset en aplicaciones con acumuladores para evitar deriva o desbordamientos en sistemas digitales.

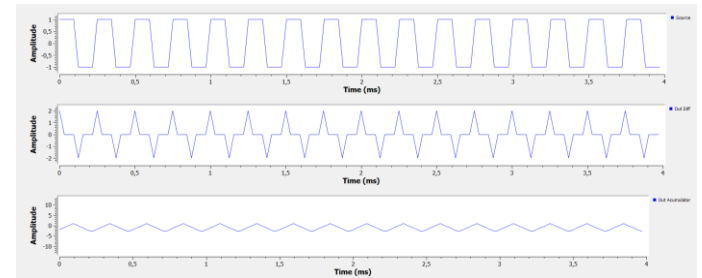


Fig. 3.2, Resultado del bloque promedio del tiempo.

4 Conclusiones

El uso de funciones propias de Python, complementadas con las herramientas sugeridas por el libro guía, permitió validar y comparar distintos métodos de reducción de ruido. Durante el desarrollo del código se realizaron pequeñas correcciones como arreglar la indentación, unificar las comillas y reordenar algunas líneas necesarias para garantizar el correcto funcionamiento de los bloques en GNU Radio.

En los bloques acumulador y diferenciador se observó un comportamiento esperado cuando la señal de entrada tiene media distinta de cero: si los valores están mayoritariamente en el eje positivo, las salidas tienden a crecer indefinidamente. En una implementación digital real este crecimiento provoca desbordamiento (overflow) y el sistema reinicia o envía valores erráticos, por lo que es imprescindible aplicar estrategias prácticas (por ejemplo, ajustar un offset) para controlar la evolución de la señal y evitar efectos no deseados.

La implementación del bloque acumulador en Python dentro de GNU Radio suma de forma consecutiva las muestras de la señal; ajustar el offset resulta fundamental para evitar la cancelación en señales cuadradas simétricas y para que la

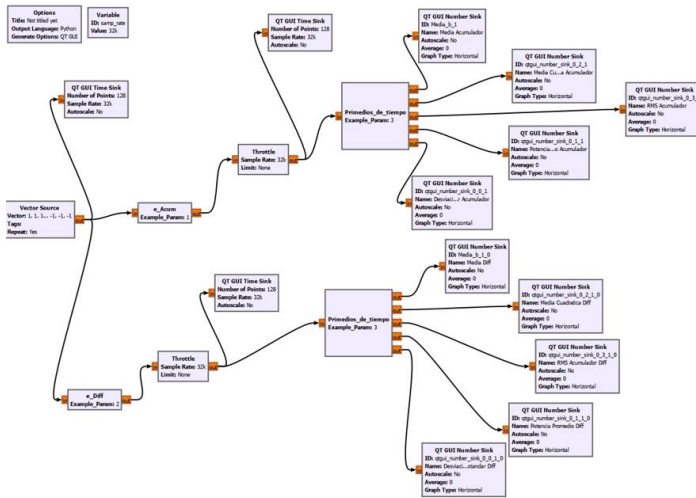


Figura 2.3, Diagrama de flujo en GNU Radio que muestra las estadísticas de una señal y reducción de ruido Gaussiano.

El bloque se integró en un diagrama de flujo en GNU Radio, conectándose en paralelo a la fuente de señal y a un bloque de visualización en tiempo real. Esto permitió tanto observar la señal afectada por ruido como calcular en simultáneo los parámetros estadísticos.

Los resultados obtenidos evidencian que el bloque de promedio del tiempo constituye una herramienta eficaz para la caracterización probabilística de señales, complementando el análisis determinístico realizado con los bloques acumulador y diferenciador, esto se ve más evidenciado en el apartado de resultados.

3 Resultados

En la Fig. 3.1 se muestra la señal cuadrada de entrada (parte inferior) con un offset de 0.5, la salida del bloque Acumulador (gráfica central) y la salida del bloque Diferenciador (gráfica superior). Se aprecia que la onda acumulada adquiere una forma triangular, ya que cada nivel constante de la señal cuadrada se integra de manera lineal mientras dure ese nivel. Por otro lado, en el bloque diferenciador se observa cómo la señal cuadrada genera pulsos en cada transición, representando correctamente la derivada de la entrada.

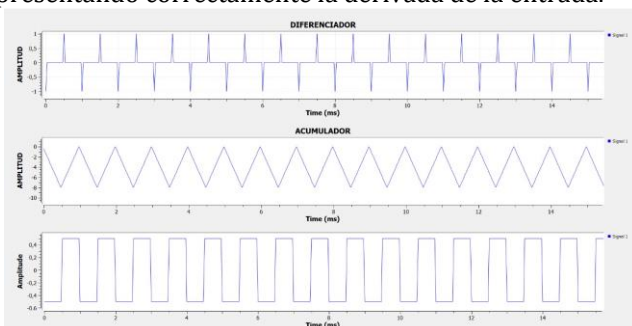


Fig. 3.1, Resultado de la señal de entrada



forma acumulada sea representativa. Asimismo, se comprobó que la salida de los bloques en GNU Radio es del tipo *stream*, lo cual exige un tratamiento distinto al de los datos en forma de *vector*. Por esta razón reestructuramos el código del libro guía para manejar las señales como vectores cuando era conveniente, facilitando su procesamiento y permitiendo que los bloques produzcan el comportamiento esperado en tiempo real.

5 Referencias

- [1] H. O. Boada and O. M. R. Torres, *Comunicaciones Plus: Volumen II*, Bucaramanga, Colombia: Editorial UIS, 2022