



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

Josman Esneider Rico Torres - 2201530 Juan Pablo Diaz Lemus - 2200530 Deisy Tatiana Torres Pedraza - 2200502

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

23 de Febrero de 2024

Resumen

En el mundo de las comunicaciones digitales, es indispensable garantizar la fidelidad del mensaje. Para lograrlo, es necesario realizar un correcto tratamiento de la señal y una programación eficiente que permita una sincronización adecuada entre uno o varios mensajes que se deseen transmitir a través del medio correspondiente. Para este propósito, se utilizan técnicas como la programación en tiempo real, empleadas en herramientas como GNURadio. *Palabras clave: GNURadio, Programacion en tiempo real, Sincronización*

1. Introducción

La programación en tiempo real desempeña un papel fundamental en las comunicaciones digitales al permitir la gestión y el control de datos y procesos en tiempo instantáneo. En entornos de comunicaciones digitales, como redes de computadoras, internet y sistemas de telecomunicaciones, la programación en tiempo real garantiza la sincronización precisa de eventos y la entrega oportuna de datos. Esto se logra mediante algoritmos y técnicas que priorizan tareas críticas y minimizan el retardo, en este caso implementando y creando funciones en GNU-radio basadas en python para el tratamiento de señales digitales.

2. Metodología

2.1. Uso de Github como herramienta de trabajo

- 1. En la primer parte se creó una cuenta de GitHub, posteriormente el repositorio "Labcom2"
- 2. Se creó la llave del repositorio y se realizó la clonación.

- 3. Se creó una rama, "Práctica1", en donde se encuentran las ramas de cada participante.
- 4. Se confirmó la subida de documentos y cambios en el repositorio.

2.2. GNURadio

GNURadio ofrece diversas herramientas para el procesamiento de señales digitales. Entre ellas, se encuentra el bloque "Python Block"[1], el cual permite la programación de un algoritmo en Python para utilizarlo en el sistema de bloques requeridos. Para hacer uso de este bloque, es importante tener en cuenta los siguientes puntos:

- 1. El código que define el bloque a configurar puede encontrarse haciendo clic en .ºpen editor"dentro de las opciones del bloque en sí.
- 2. Manejo de entradas y salidas: Se deben identificar y declarar correctamente las entradas y salidas que debe tener el bloque. Para lograrlo, es necesario definirlas en el constructor de la clase en forma de lista, utilizando la definición de variables de numpy, como por ejemplo "np.float32"[2].
- 3. Una vez que se han definido las entradas y salidas, el algoritmo que describe la relación entradasalida, o la función de transferencia del bloque, debe colocarse dentro del método de clase llamado "work". En este método, las entradas y salidas deben definirse como "input items[]z .ºutput items[]", respectivamente, utilizando el índice correspondiente cuando sea necesario.

Con el objetivo de ejemplificar el uso del bloque "Python Block", se implementan tres algoritmos correspondientes a tres funcionalidades distintas:

- 1. **Acumulador**: Este algoritmo suma continuamente las muestras de entrada y proporciona la suma acumulada como salida.
- 2. **Diferenciador**: Este algoritmo calcula la diferencia entre muestras adyacentes y proporciona esta diferencia como salida.
- Análisis estadístico de una señal: Este algoritmo calcula estadísticas básicas de la señal de entrada, como el promedio, la desviación estándar y la varianza, y proporciona estos valores como salidas.

Cada uno de estos algoritmos puede ser implementado dentro del bloque "Python Block"para realizar operaciones específicas de procesamiento de señales digitales en tiempo real.

3. Resultados

Se probó el bloque acumulador con una entrada constante de valor unitario y se obtuvieron los siguientes resultados.

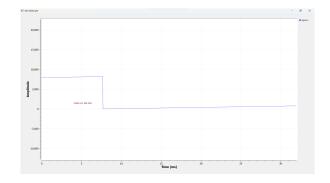


Fig. 1: Resultados bloque acumulador

Se probó el bloque diferenciador con una entrada constante de valor unitario y se obtuvieron los siguientes resultados.

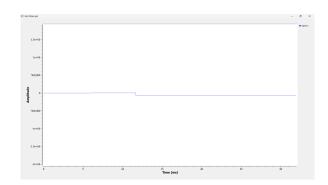


Fig. 2: Resultados bloque diferenciador

Se probó el bloque de análisis estadístico con una entrada correspondiente a ruido blanco, generado por el bloque 'random source' [3] de GNURadio, y se obtuvieron los siguientes resultados.

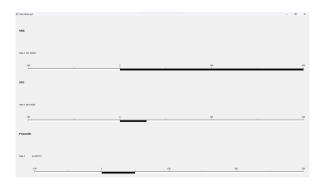


Fig. 3: Resultados bloque de análisis estadístico

Todos los códigos y sistemas utilizados se pueden encontrar en el siguiente enlace:https://github.com/Lilith022/labcom2_2024_1/tree/Practica_1/Practica_1/Lab_1/GNU_Radio

4. Análisis de resultados

4.1. Bloque acumulador

Los resultados del bloque acumulador se vieron acordes a lo esperado. Podemos observar en la figura 1 cómo, cada vez que se resetea la entrada, la señal empieza a crecer de forma lineal por acción del bloque acumulador.

4.2. Bloque diferenciador

Al aplicar una entrada constante unitaria al bloque diferenciador, se observa que cada vez que se resetea la



entrada, la señal adquiere un comportamiento similar al del bloque acumulador, pero en este caso con una pendiente negativa, como se puede apreciar en la figura 2, tal como se esperaba del bloque mencionado.

4.3. Bloque de análisis estadístico

Cuando se aplica una entrada aleatoria, es importante considerar que la varianza y la desviación estándar tienen una relación proporcional con el rango de la fuente aleatoria, que en este caso es de 0 a 100, lo que nos da un span de 100. Al buscar el valor promedio de una señal aleatoria, obtenemos el valor medio de la misma, que estadísticamente es el valor más probable. En este caso, tenemos un valor medio de 50, lo cual concuerda con el rango definido. Además, los valores de varianza y desviación estándar también tienen una magnitud esperada en relación con el span de la fuente aleatoria. Estos resultados se pueden observar en la figura 3.

5. Conclusiones

 GNU Radio es una plataforma versátil para el desarrollo de aplicaciones de radio y procesamiento

- de señales. Proporciona un entorno flexible para diseñar sistemas de comunicación.
- La implementación de algoritmos de Python en GNURadio es una herramienta sumamente útil para el procesamiento de señales digitales, ya que amplía significativamente las opciones disponibles. Esto permite aprovechar las características de Python y bibliotecas como NumPy, Pandas y SciPy, las cuales ofrecen funciones muy útiles en el ámbito de las comunicaciones digitales.
- La probabilidad y la estadística pueden ser herramientas muy útiles para abordar las no idealidades que pueden surgir en las comunicaciones digitales.

Referencias

- [1] H. Ortega and O. Reyes, *Comunicaciones Digitales basadas en radio definida por software*. Editorial UIS, Octubre 2019.
- [2] "Numpy documentation." [Online]. Available: https://numpy.org/
- [3] "Gnu radio open-source software, list of blocks." [Online]. Available: wiki.gnuradio.org