RESUMEN

En esta práctica de laboratorio, se exploró el uso de GNU Radio como herramienta para el procesamiento de señales digitales. Se implementaron diversos bloques para la generación, muestreo, filtrado y visualización de señales, evaluando el impacto de la frecuencia de muestreo en la calidad de la señal procesada.

Además, se confirmó la importancia del teorema de Nyquist, pues una frecuencia de muestreo inadecuada que genera aliasing, afectando la reconstrucción de la señal original. También se comprobó que la interpolación y la decimación son técnicas clave para modificar la tasa de muestreo sin distorsionar la señal, siempre que se utilicen filtros adecuados. Finalmente, se destacó la versatilidad de GNU Radio para diseñar y simular sistemas de comunicación en tiempo real, facilitando el análisis sin necesidad de hardware costoso. Esta práctica permitió afianzar conocimientos sobre muestreo y procesamiento digital de señales en un entorno experimental.

INTRODUCCIÓN

El procesamiento de señales es una disciplina clave en ingeniería, donde la teoría de muestreo juega un papel esencial para convertir señales analógicas en digitales sin pérdida de información. La frecuencia de muestreo es crucial para evitar el aliasing, un fenómeno que distorsiona la señal cuando no se cumple el criterio de Nyquist. Herramientas como GNU Radio facilitan la experimentación en laboratorios de comunicaciones, permitiendo el análisis y diseño de sistemas sin necesidad de hardware especializado.

Además, técnicas como la interpolación y la decimación permiten ajustar la tasa de muestreo según las necesidades del procesamiento, optimizando el almacenamiento y la transmisión de señales. Sin embargo, una frecuencia de muestreo inadecuada puede causar distorsiones y pérdida de información, afectando la calidad del análisis y la interpretación de los datos. Este documento explora estos conceptos fundamentales y sus aplicaciones en el procesamiento digital de señales.

PROCEDIMIENTO

* Se abrió el entorno de GNU Radio y se guardó el proyecto con extensión. grc, diferenciándolo del archivo .py generado automáticamente.
* Se añadieron los bloques Signal Source, Throttle, QT GUI Frequency Sink y QT GUI Time Sink para generar, visualizar y controlar señales.
* Se conectaron los bloques asegurando su correcta comunicación, verificando que los textos cambiaran de rojo a negro.
* Se usaron bloques Variable y QT GUI Range para ajustar frecuencia, amplitud y forma de onda.
* Se asignaron las variables creadas a las propiedades del bloque Signal Source para modificar su comportamiento.
* Se añadió el bloque QT GUI Chooser para cambiar entre distintos tipos de onda de forma interactiva.
* Se ejecutó el diagrama de flujo, observando las señales en los dominios de tiempo y frecuencia al modificar parámetros.
* Durante el experimento, se tomaron capturas de las señales generadas para analizar su comportamiento al interpolar, diezmar y aplicar el teorema de Nyquis