25.20 Análisis de Señales y Sistemas Digitales

Trabajo Práctico N° 1

Muestreo

BELSITO, RAMIRO NAHUEL	*62641*	`rabelsito@itba.edu.ar`
SAMMARTINO, IGNACIO	*63053*	`isammartino@itba.edu.ar`
GASTALDI, ROCCO	*61659*	`rgastaldi@itba.edu.ar`
TERRA, IGNACIO	*XXXXX*	`X@itba.edu.ar`
MINITI, MATIAS	*XXXXX*	`X@itba.edu.ar`

Profesores: Daniel Andres Jacoby y Martin Paura Bersan

Fecha de entrega: de Septiembre de 2025

Índice

1. Introducción

En esta experiencia de laboratorio se desarrolló un sistema capaz de realizar muestreo (instantáneo y natural) de señales analógicas. Para ello, se utilizaron varios bloques que permitirán mayor control y entendimiento sobre el procesamiento de la señal y sus efectos. De esta forma se logró observar los distintos comportamientos de la señal en cada etapa del proceso de muestreo. Además, se implementó el trabajo tanto en una PCB como en simulación en LTSpice y en un entorno propio desarrllado en Python, lo que permitió comparar los efectos de los distintos componentes y las diferencias entre los resultados ideales, los modelados y los reales.

2. Marco Teórico

3. Circuito de Muestreo

4. Filtros Pasa Bajos

5. Simulaciones

Se implemento una simulación en Python para observar los efectos del muestreo instantáneo y natural sobre señales analógicas. El código desarrollado permite observar que sucede en cada etapa del proceso de muestreo, ademas de tener la capacidad de poder saltear etapas y observar los efectos de cada una, tanto en el dominio del tiempo como en el de la frecuencia.

El código se implemento en Python utilizando las librerías numpy, scipy, matplotlib, y PyQt5. La interfaz gráfica permite decidir con que señal de entrada se quiere trabajar, la frecuencia de muestreo, el tipo de muestreo (instantáneo o natural), y la frecuencia de corte de los filtro anti-alias y de reconstrucción. También permite variar el duty cycle del sample and hold. Ambos filtros (anti-alias y de reconstrucción) se implenetaron usando la funcion de aproximación de Cauer (elliptic) de orden 6, al igual que en la placa del circuito impreso. Además, indica con puntos naranjas donde se muestreó la señal de entrada A continuación se muestran algunas capturas de pantalla de la interfaz gráfica:

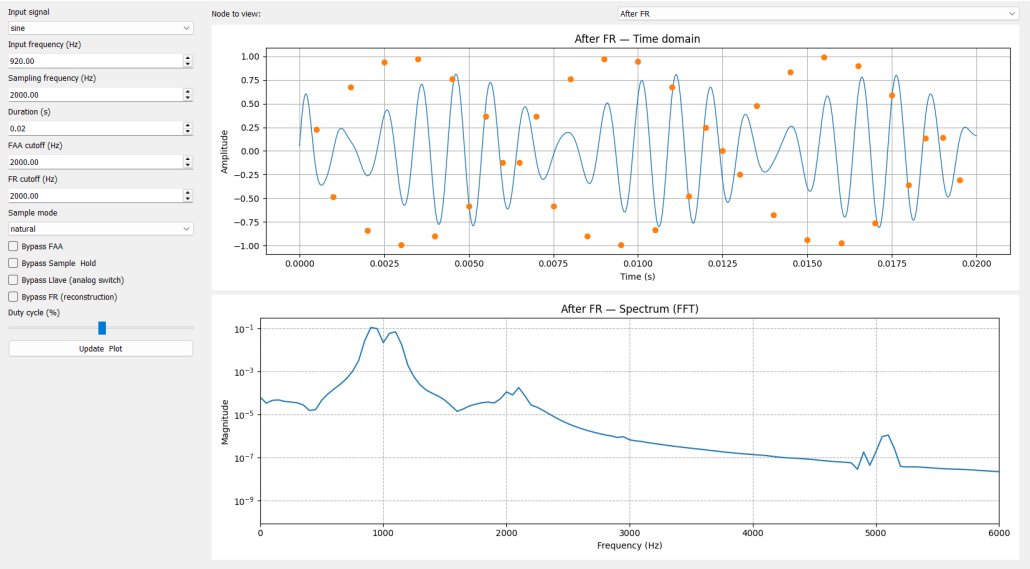

Figura 5.1: Interfaz gráfica - Ejemplo de señal en la GUI

Donde se puede apreciar el efecto de "beating" ya que el la señal de entrada es cercana a $f_s/2$, el filtro recuperador no es capaz de eliminar el aliasing y se observa una señal modulada en amplitud.

6. Simulaciones en Python

7. Conclusión

8. Anexo

Repositorio del código:

https://github.com/ramibelsito/ASSD—Simulador-de-Muestreo