



25.20 Análisis de Señales y Sistemas Digitales

# Trabajo Práctico N° 1

## Muestreo

|                        |       |                                |
|------------------------|-------|--------------------------------|
| BELSITO, RAMIRO NAHUEL | 62641 | <i>rabelsito@itba.edu.ar</i>   |
| SAMMARTINO, IGNACIO    | 63053 | <i>isammartino@itba.edu.ar</i> |
| GASTALDI, ROCCO        | 61659 | <i>rgastaldi@itba.edu.ar</i>   |
| TERRA, IGNACIO         | XXXXX | <i>iterr@itba.edu.ar</i>       |
| MINNITI, MATIAS        | 64030 | <i>mminniti@itba.edu.ar</i>    |

Profesores: Daniel Andres Jacoby y Martin Paura Bersan  
Fecha de entrega: de Septiembre de 2025

# Índice

|   |           |
|---|-----------|
| <b>1. Introducción</b>                                  | <b>1</b>  |
| <b>2. Marco Teórico</b>                                 | <b>1</b>  |
| 2.1. Muestreo de Señales . . . . .                      | 1         |
| 2.2. Tipos de Muestreo . . . . .                        | 1         |
| 2.2.1. Muestreo Ideal . . . . .                         | 1         |
| 2.2.2. Muestreo Natural . . . . .                       | 2         |
| 2.2.3. Muestreo Instantáneo (Zero-Order Hold) . . . . . | 3         |
| 2.3. Filtros en el Proceso de Muestreo . . . . .        | 4         |
| 2.3.1. Filtro AntiAliasing . . . . .                    | 4         |
| 2.3.2. Filtro de Reconstrucción . . . . .               | 4         |
| 2.4. Llave Analógica . . . . .                          | 4         |
| <b>3. Implementación</b>                                | <b>4</b>  |
| 3.0.1. Fotos de la Placa en Uso . . . . .               | 5         |
| 3.1. Filtros Pasabajos . . . . .                        | 5         |
| 3.2. Oscilador . . . . .                                | 6         |
| 3.3. Obtención de Señal Muestreada . . . . .            | 7         |
| <b>4. Simulaciones</b>                                  | <b>8</b>  |
| <b>5. Simulaciones en Python</b>                        | <b>8</b>  |
| <b>6. Conclusión</b>                                    | <b>9</b>  |
| <b>Bibliografia</b>                                     | <b>10</b> |

## 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 desarrollado 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

### 2.1 Muestreo de Señales

El objetivo del muestreo de las señales es conseguir obtener una reconstrucción fiel a la señal continua original partiendo de muestras discretas de la misma. Se debe tener en cuenta que la frecuencia de muestreo ( $f_s$ , inversa al período que separa las muestras discretas mencionadas previamente) debe ser, como mínimo, el doble de la frecuencia máxima presente en la señal original ( $f_{max}$ ). Esto se conoce como el Teorema de Nyquist-Shannon y se expresa matemáticamente como:

$$f_s \geq 2f_{\text{máx}}. \quad (1)$$

Tras este proceso, el espectro de la señal sampleada se repite cada  $\pm(N \cdot f_s)$ . Es por eso que, luego del muestreo, la señal original no se recupera sino hasta después de aplicar filtros de alta frecuencia que eliminen dichas replicas. De no cumplirse con el criterio de Nyquist, se produce un fenómeno conocido como *aliasing*, que genera distorsiones en la señal reconstruida debido a que las replicas se ubican dentro del espectro de frecuencia de la señal original.

### 2.2 Tipos de Muestreo

Existen diferentes modelos para describir el proceso de muestreo. Los más relevantes para este trabajo son: el muestreo ideal, el muestreo natural y el muestreo instantáneo (siendo estos últimos los proporcionados por el Sample & Hold).

#### 2.2.1 Muestreo Ideal

A continuación se muestra el desarrollo en el dominio de la frecuencia: El muestreo ideal es un modelo teórico que supone la multiplicación de la señal  $x(t)$  por un tren de impulsos de Dirac:

$$x^*(t) = x(t) \cdot \delta_T(t) \quad (2)$$

donde  $T_s$  es el período de muestreo y  $\omega_s = 2\pi/T_s$  la frecuencia de muestreo.

$$X^*(\omega) = \frac{1}{2\pi} X(\omega) \otimes \mathcal{F}\{\delta_T(t)\} \quad (3)$$

$$= \frac{1}{T} X(\omega) \otimes \left\{ \omega_s \sum_{m=-\infty}^{\infty} \delta(\omega - m\omega_s) \right\} \quad (4)$$

$$= \frac{1}{T} X(\omega) \otimes \left\{ \sum_{m=-\infty}^{\infty} \delta(\omega - m\omega_s) \right\} \quad (5)$$

$$X^*(\omega) = \frac{1}{T} \sum_{m=-\infty}^{\infty} X(\omega - m\omega_s) \quad (6)$$

Donde  $\otimes$  denota convolución.

En el dominio de la frecuencia, este proceso genera réplicas del espectro original  $X(\omega)$  desplazadas en múltiplos de  $\omega_s$ : Si se cumple el criterio de Nyquist, dichas réplicas no se solapan, y la señal puede recuperarse exactamente mediante un filtro pasa bajos ideal.

Este modelo no es realizable en la práctica, ya que no existen impulsos de duración infinitesimal ni amplitud infinita.

### 2.2.2 Muestreo Natural

El muestreo natural consiste en permitir el paso de la señal durante un intervalo finito de tiempo  $\tau$ , repitiendo este proceso cada  $T_s$ . Matemáticamente, se describe como: En el muestreo natural, a diferencia del muestreo ideal, la señal de entrada  $x(t)$  no se multiplica por un tren de impulsos de Dirac, sino por un tren de pulsos rectangulares de ancho  $\tau$ . De esta manera, la señal muestreada se obtiene como:

$$x_n(t) = x(t) \cdot p_T(t), \quad (7)$$

donde  $p_T(t)$  es un tren de pulsos rectangulares de período  $T_s$  y ancho  $\tau$ .

En el dominio de la frecuencia, la multiplicación en el tiempo se transforma en una convolución:

$$X_n(\omega) = \frac{1}{2\pi} X(\omega) \otimes P_T(\omega), \quad (8)$$

donde  $P_T(\omega)$  es el espectro del tren de pulsos periódicos.

El espectro del tren de pulsos se puede expresar como una serie de impulsos modulados por una función sinc:

$$P_T(\omega) = 2\pi \frac{\tau}{T_s} \sum_{n=-\infty}^{\infty} \text{sinc}\left(\frac{n\omega_s \tau}{2}\right) \delta(\omega - n\omega_s), \quad (9)$$

donde  $\omega_s = \frac{2\pi}{T_s}$  es la frecuencia de muestreo en radianes.

Sustituyendo en la convolución:

$$X_n(\omega) = \frac{1}{2\pi} X(\omega) \otimes \left[ \frac{2\pi}{T_s} \sum_{n=-\infty}^{\infty} \text{sinc}\left(\frac{n\omega_s\tau}{2}\right) \delta(\omega - n\omega_s) \right] \quad (10)$$

$$= \frac{\tau}{T_s} \sum_{n=-\infty}^{\infty} \text{sinc}\left(\frac{n\omega_s\tau}{2}\right) X(\omega - n\omega_s). \quad (11)$$

Es decir, el muestreo natural también genera réplicas del espectro  $X(\omega)$  en múltiplos de  $\omega_s$ , pero ahora cada réplica está atenuada por un factor de tipo *sinc* que depende tanto de  $m$  como del ancho  $\tau$  del pulso. Pero para un  $n$  dado, la sinc permanece constante y no depende de omega

### 2.2.3 Muestreo Instantáneo (Zero-Order Hold)

El bloque *Zero-Order Hold* (ZOH) es el encargado de retener el valor de cada muestra durante un período de muestreo  $T_s$ . A diferencia del muestreo natural o ideal, aquí cada muestra se transforma en un pulso constante de duración  $T_s$ .

La respuesta impulsiva del ZOH es un pulso rectangular de ancho  $T_s$ :

$$h_{ZOH}(t) = \begin{cases} 1, & 0 \leq t < T_s \\ 0, & \text{otro caso} \end{cases} \quad (12)$$

Su transformada de Fourier es:

$$H_{ZOH}(\omega) = T_s \text{sinc}\left(\frac{\omega T_s}{2}\right) \quad (13)$$

Si la entrada al bloque es la señal muestreada ideal  $x^*(t)$ , la salida se obtiene por convolución:

$$y(t) = x^*(t) \otimes h_{ZOH}(t), \quad (14)$$

lo cual en frecuencia se traduce en:

$$Y(\omega) = X^*(\omega) \cdot H_{ZOH}(\omega). \quad (15)$$

Recordando que el espectro del muestreo ideal es:

$$X^*(\omega) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(\omega - n\omega_s), \quad (16)$$

la salida del ZOH resulta:

$$Y(\omega) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(\omega - n\omega_s) \cdot T_s \text{sinc}\left(\frac{\omega T_s}{2}\right). \quad (17)$$

Ahora la atenuación de la sinc no se mantiene constante para cada réplica, sino que depende de  $\omega$  para la misma replica.

## 2.3 Filtros en el Proceso de Muestreo

Con el objetivo de eliminar las réplicas de alta frecuencia producidas por el muestreo, es necesario implementar filtros pasabajos.

### 2.3.1 Filtro AntiAliasing

El filtro anti-aliasing se coloca antes del zero-order hold y su función es limitar el espectro de la señal de entrada para que no haya componentes de frecuencias mayores que la mitad de la frecuencia de muestreo (cumpliendo así con el criterio de Nyquist 1). Si bien el filtro asegura una reconstrucción más fiel a la señal original, límita las señales de entrada posibles al sistema.

### 2.3.2 Filtro de Reconstrucción

El filtro de reconstrucción se coloca al final del proceso de muestreo y su función es eliminar las réplicas de alta frecuencia generadas por el muestreo. La frecuencia de corte de este filtro pasabajos debe ser mayor a la frecuencia máxima de la señal original, pero menor a la frecuencia de Nyquist.

## 2.4 Llave Analógica

El dispositivo encargado de realizar el muestreo de la señal es el Sample & Hold, que permite realizar muestreos tanto naturales como instantáneos. Para seleccionar el modo de operación, se utiliza una llave analógica (sincronizada con el duty y la frecuencia del Sample & Hold) que permite el paso de la señal muestreada en el momento correspondiente al modo elegido.

## 3. Implementación

Esquemático del circuito a implementar:

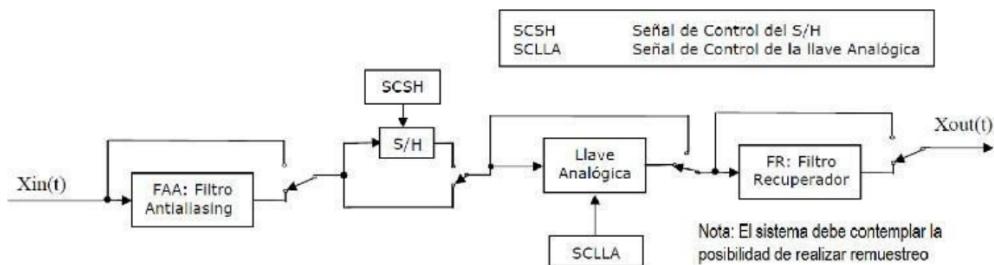


Figura 3.1: Circuito de Muestreo y Retención

Esta implementación del circuito de muestreo y recuperación de la señal analógica nos permite analizar los efectos de las distintas etapas del proceso de muestreo y su importancia en el sampleo de la señal.

### 3.0.1 Fotos de la Placa en Uso

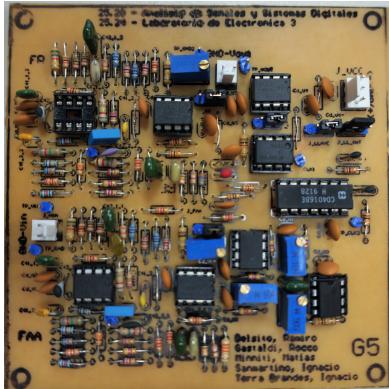


Foto 1

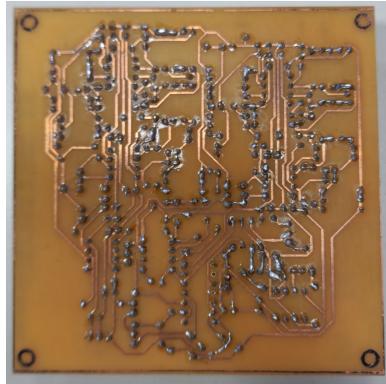


Foto 2

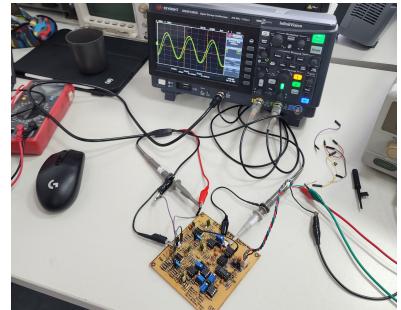


Foto 3

Figura 3.2: Fotos del equipo en el laboratorio

### 3.1 Filtros Pasabajos

La señal a muestrear que limita el diseño de los filtros corresponde a una onda cuadrada de amplitud  $V_{\max}$ , ciclo de trabajo  $D = 0,75$  y período

$$T = \frac{2}{f_i}, \quad f_i = 15 \text{ kHz},$$

de modo que la frecuencia fundamental resulta

$$f_0 = \frac{1}{T} = \frac{f_i}{2} = 7,5 \text{ kHz}.$$

La serie de Fourier de una onda cuadrada con duty  $D$  se obtiene a partir de los coeficientes  
Consideremos una señal cuadrada periódica de amplitud  $A$ , periodo  $T$  y duty cycle  $D$ :

$$x(t) = \begin{cases} A, & 0 \leq t < DT \\ 0, & DT \leq t < T \end{cases}$$

y se repite cada  $T$  segundos. Los coeficientes de la serie de Fourier están dados por:

$$c_n = \frac{1}{T} \int_0^T x(t) e^{-jn\omega_0 t} dt, \quad \omega_0 = \frac{2\pi}{T}$$

Para nuestra señal cuadrada:

$$c_n = \frac{1}{T} \int_0^{DT} A e^{-jn\omega_0 t} dt = \frac{A}{T} \int_0^{DT} e^{-jn\omega_0 t} dt$$

Integrando:

$$c_n = \frac{A}{T} \left[ \frac{e^{-jn\omega_0 t}}{-jn\omega_0} \right]_0^{DT} = \frac{A}{jn\omega_0 T} (1 - e^{-jn\omega_0 DT})$$

Como  $\omega_0 T = 2\pi$ :

$$c_n = \frac{A}{j2\pi n} (1 - e^{-j2\pi n D}), \quad n \neq 0$$

Para  $n = 0$  (componente DC):

$$c_0 = \frac{1}{T} \int_0^T x(t) dt = \frac{A \cdot DT}{T} = AD$$

Usando la identidad  $1 - e^{-j\theta} = 2j \sin(\theta/2)e^{-j\theta/2}$ , la magnitud de cada armónico es:

$$|c_n| = \frac{A}{n\pi} |\sin(n\pi D)|$$

- Para  $D = 0,75$ :

$$|c_n| = \frac{A}{n\pi} |\sin(0,75 n\pi)|$$

Ya que para una buena reconstrucción de la misma es necesario que los filtros capturen la mayor cantidad de harmónicos posibles, se decidió colocar la frecuencia de corte en  $f_c = 80 \text{ kHz}$ , lo cual permite capturar hasta 10 harmónicos (no nulos) de la señal de entrada.

Para el diseño de los filtros se utilizó la aproximación de Cauer (elíptica), con 3 celdas Sedra colocadas en cascada.

### 3.2 Oscilador

En la Fig. 3.3 se muestra el oscilador implementado. El circuito utiliza un temporizador NE555 configurado como generador de impulsos, cuya salida se integra mediante una red RC ( $R_1$ , potenciómetro y  $C_1$ ) para obtener una señal triangular. La frecuencia de oscilación puede ajustarse modificando la resistencia efectiva en el integrador, lo cual se logra a través de un preset en serie con  $R_1$ . De esta forma, es posible alcanzar frecuencias de oscilación de hasta aproximadamente  $200 \text{ kHz}$ .

La señal triangular se aplica luego a dos comparadores implementados con amplificadores operacionales TL082. Estos comparadores se configuran como disparadores de Schmitt, de manera que la histéresis de conmutación puede ajustarse con un preset independiente. Este control permite desplazar los umbrales de conmutación de la señal triangular, modificando así el ciclo de trabajo (*duty cycle*) de la onda cuadrada resultante. En la práctica, el rango de variación obtenido es amplio, abarcando aproximadamente entre 5 % y 95 %.

En resumen, el diseño cumple con los requerimientos planteados: generar una señal cuadrada con frecuencia ajustable y con *duty cycle* variable de forma independiente, empleando un oscilador basado en el NE555, un integrador RC y comparadores con histéresis.

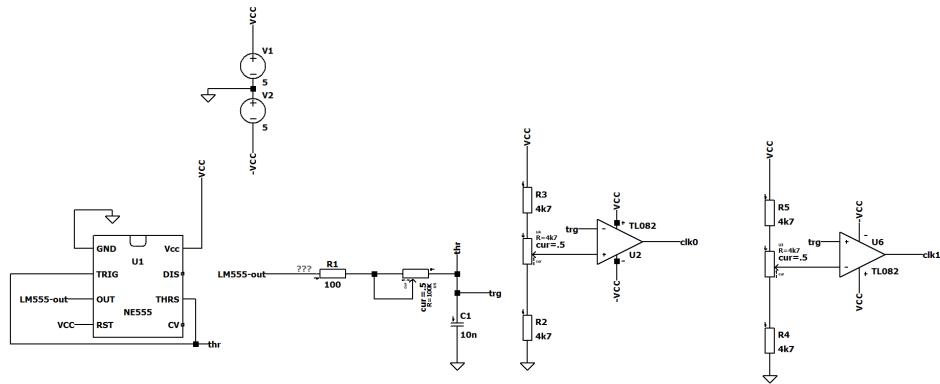


Figura 3.3: Esquemático del Oscilador

&lt;“‘&lt;HEAD =====

»””»&gt;50835e1da62df98259a326d951c03eed0cc756d0

### 3.3 Obtención de Señal Muestreada

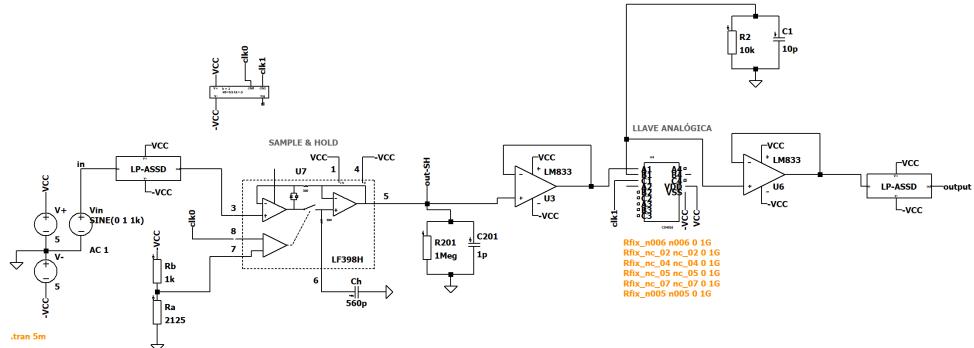


Figura 3.4: Simulación del Circuito de Muestreo y Retención

El circuito implementado emplea un integrado LF398 configurado como *sample and hold*, en conjunto con una llave analógica (CD4016), con el objetivo de muestrear y posteriormente retener la señal de entrada durante un intervalo de tiempo definido. De esta manera, la señal analógica puede ser tomada en instantes discretos, procesada y posteriormente recuperada a través del filtrado.

La elección del capacitor de retención ( $C_H$ ) en el *sample and hold* resulta crítica, dado que determina el compromiso entre tiempo de adquisición y estabilidad de la tensión retenida. En este caso se seleccionó un valor de 560 pF, que permite un tiempo de adquisición reducido gracias a la constante de tiempo  $R_{on} \cdot C_H$  asociada al transistor de muestreo interno, garantizando que el condensador alcance rápidamente el nivel de la señal. Al mismo tiempo, este valor asegura una tasa de caída (*droop rate*) suficientemente baja durante el intervalo de retención, minimizando los errores introducidos por corrientes de fuga. Para corrientes típicas del orden de nanamps, la pendiente de descarga resulta del orden de decenas de milivoltos por milisegundo, lo cual es adecuado para las frecuencias de operación consideradas.

En cuanto a la llave analógica CD4016, se emplea para controlar de forma precisa el instante de muestreo y transferencia de la señal. La selección de este dispositivo se justifica por su simplicidad de control y su capacidad de operar adecuadamente en el rango de tensiones utilizado en el circuito. La resistencia de conducción ( $R_{on}$ ) del CD4016, junto con el valor del capacitor de retención, también influye en el tiempo de establecimiento de la muestra, siendo la elección del capacitor coherente con las limitaciones impuestas por el interruptor.

En conjunto, la configuración implementada permite obtener un muestreo confiable de la señal de entrada, con errores acotados tanto en la fase de adquisición como en la de retención, asegurando un desempeño adecuado para las condiciones de frecuencia y precisión requeridas.

#### 4. Simulaciones

#### 5. Simulaciones en Python

Se implementó 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, además de tener la capacidad de poder saltar etapas y observar los efectos de cada una, tanto en el dominio del tiempo como en el de la frecuencia.

El código se implementó en Python utilizando las librerías numpy, scipy, matplotlib, y PyQt5. La interfaz gráfica permite decidir con qué 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 implementaron usando la función 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:

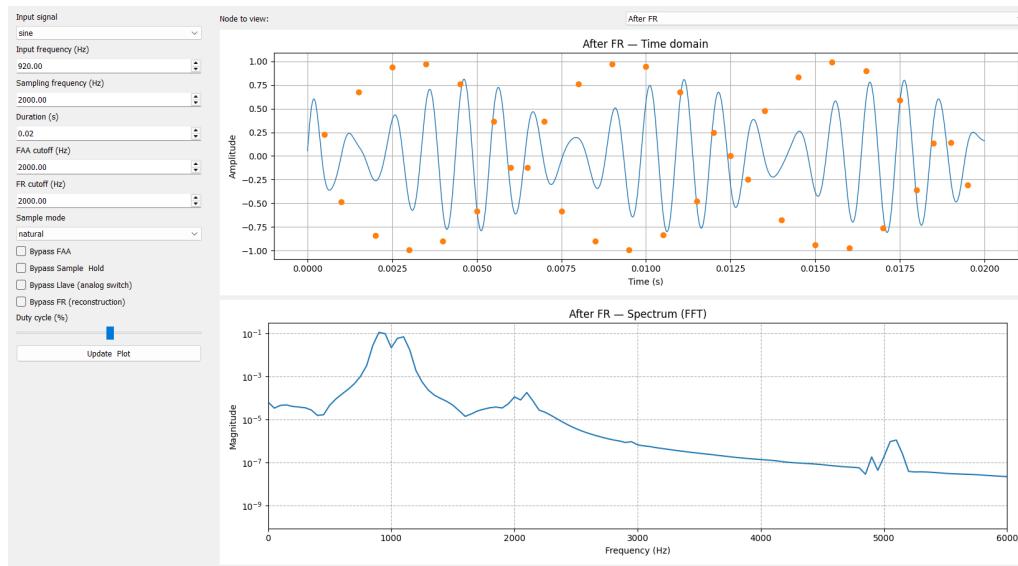


Figura 5.5: Interfaz gráfica - Ejemplo de señal en la GUI

Donde se puede apreciar el efecto de “beating” ya que 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.

Todo el código fuente se encuentra en el repositorio de GitHub: [1](#)

## 6. Conclusión

En esta práctica se logró implementar y analizar un sistema completo de muestreo de señales analógicas, incluyendo las modalidades ideal, natural e instantánea (Zero-Order Hold). A través de la combinación de simulaciones en Python, LTSpice y la implementación en placa, se pudo observar de manera comparativa cómo cada tipo de muestreo afecta la señal y su espectro en frecuencia.

Se verificó experimentalmente el Teorema de Nyquist-Shannon, confirmando que la frecuencia de muestreo debe ser al menos el doble de la frecuencia máxima de la señal para evitar aliasing. Además, se evidenció la importancia de los filtros anti-aliasing y de reconstrucción, ya que su correcta implementación permite preservar la mayor cantidad de armónicos de la señal original, garantizando una reconstrucción fiel.

En conclusión, esta experiencia permitió consolidar conceptos teóricos de muestreo, procesamiento de señales y filtrado, y desarrollar habilidades prácticas para diseñar y analizar sistemas de adquisición de señales analógicas en un entorno tanto simulado como real.

## Bibliografía

[1] Repositorio: <https://github.com/ramibelsito/ASSD---Simulador-de-Muestreo.git>