



25.20 Análisis de Señales y Sistemas Digitales

Trabajo Práctico N° 1

Muestreo

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

Profesores: Daniel Andres Jacoby y Martin Paura Bersan

Fecha de entrega: de Septiembre de 2025

Índice

| | |
|---|----------|
| 1. Introducción | 1 |
| 2. Marco Teórico | 1 |
| 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 |
| 3. Implementación | 4 |
| 4. Filtros Pasa Bajos | 4 |
| 5. Simulaciones | 4 |
| 6. Simulaciones en Python | 5 |
| 7. Conclusión | 5 |
| 8. Anexo | 5 |

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 frecuencias máxima presente en la señal original (f_{max}). Esto se conoce como el Teorema de Nyquist-Shannon y se expresa matemáticamente como:

$$f_s \geq 2f_{m\acute{a}x}. \quad (1)$$

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 ω_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 τ , 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 τ . 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 τ .

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 ω_s , pero ahora cada réplica está atenuada por un factor de tipo *sinc* que depende tanto de m como del ancho τ del pulso. Pero para un n dado, la *sinc* permanece constante y no depende de ω

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 ω para la misma réplica.

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, limita 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:

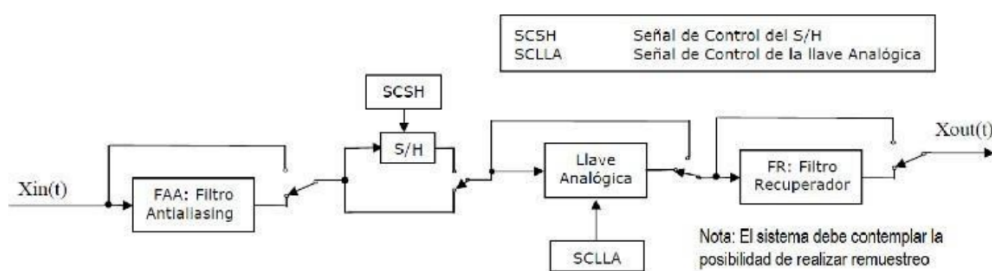


Figura 3.1: Circuito de Muestreo y Retención

4. Filtros Pasa Bajos

5. Simulaciones

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 filtros 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 (elíptico) 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:

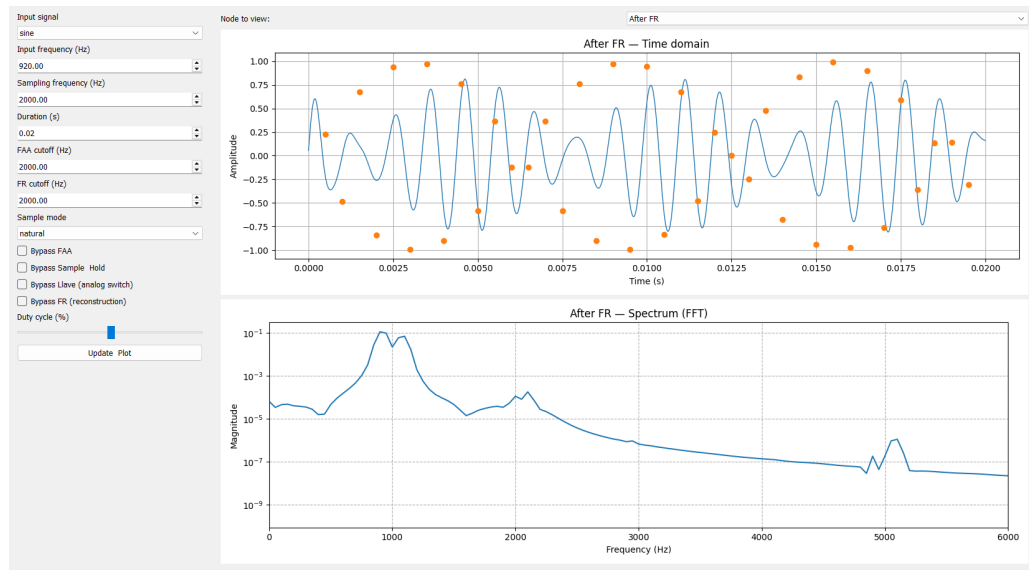


Figura 5.2: Interfaz gráfica - Ejemplo de señal en la GUI

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.

6. Simulaciones en Python

7. Conclusión

8. Anexo

Repositorio del código:

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