



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 (Sample & Hold)	2
2.3. Llave Analógica	3
2.4. Filtros en el Proceso de Muestreo	3
3. Circuito de Muestreo	3
4. Filtros Pasa Bajos	3
5. Simulaciones	3
6. Simulaciones en Python	4
7. Conclusión	4
8. Anexo	4

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

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:

$$x_{sn}(t) = x(t) \cdot p_T(t), \quad (7)$$

donde $p_T(t)$ es un tren de pulsos rectangulares de ancho τ y período T_s . En frecuencia, el espectro resultante es:

$$X_{sn}(f) = X(f) * P_T(f), \quad (8)$$

con $*$ denotando la convolución y $P_T(f)$ dado por:

$$P_T(f) = \frac{\tau}{T_s} \sum_{n=-\infty}^{\infty} \text{sinc}(f\tau) \delta(f - nf_s). \quad (9)$$

Este tipo de muestreo introduce una envolvente sinc que atenúa las componentes de alta frecuencia, lo cual debe tenerse en cuenta en la etapa de reconstrucción.

2.2.3 Muestreo Instantáneo (Sample & Hold)

El muestreo instantáneo, o de retención, consiste en capturar el valor de la señal en un instante y mantenerlo constante durante un intervalo τ . Se modela como el muestreo ideal seguido de la convolución con un pulso rectangular de duración τ :

$$x_{si}(t) = \left[x(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \right] * \text{rect}\left(\frac{t}{\tau}\right). \quad (10)$$

En frecuencia, se obtiene:

$$X_{si}(f) = f_s \cdot \sum_{n=-\infty}^{\infty} X(f - nf_s) \cdot \tau \cdot \text{sinc}(f\tau). \quad (11)$$

La consecuencia de este tipo de muestreo es que el espectro queda multiplicado por una función sinc, lo que implica una atenuación dependiente de la frecuencia. Por este motivo, además del filtro pasa bajos, suele ser necesario un bloque de ecualización que compense la distorsión introducida por la envolvente.

2.3 Llave Analógica

Las llaves analógicas permiten habilitar o bloquear el paso de una señal en función de una señal de control (generalmente un tren de pulsos). Su correcta implementación es clave para lograr un muestreo preciso, ya que deben presentar baja resistencia en conducción, alta velocidad de conmutación y baja capacitancia parásita.

En combinación con el bloque Sample & Hold, las llaves analógicas permiten implementar tanto el muestreo natural como el instantáneo, dependiendo de la configuración del sistema y del control aplicado.

2.4 Filtros en el Proceso de Muestreo

Para que el muestreo y la posterior reconstrucción sean correctos, es necesario incorporar filtros pasa bajos en dos etapas fundamentales:

- **Filtro Anti-Aliasing (FAA):** Se coloca antes de la etapa de muestreo. Su objetivo es atenuar las componentes de frecuencia superiores a $f_s/2$, evitando aliasing. En teoría se modela como un filtro ideal con transición abrupta, pero en la práctica se implementan aproximaciones (Butterworth, Chebyshev, Cauer, etc.), que ofrecen un compromiso entre complejidad y desempeño.
- **Filtro Recuperador (FR):** Se ubica a la salida del sistema y se encarga de eliminar las réplicas espectrales no deseadas, suavizando la señal muestreada y recuperando lo más fielmente posible la señal original.

En resumen, el correcto diseño del sistema de muestreo requiere la interacción entre la frecuencia de muestreo, la implementación de la llave analógica, el bloque Sample & Hold y el uso adecuado de filtros. La calidad de la reconstrucción dependerá de la coherencia entre estos elementos y del cumplimiento del criterio de Nyquist.

3. Circuito de Muestreo

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:

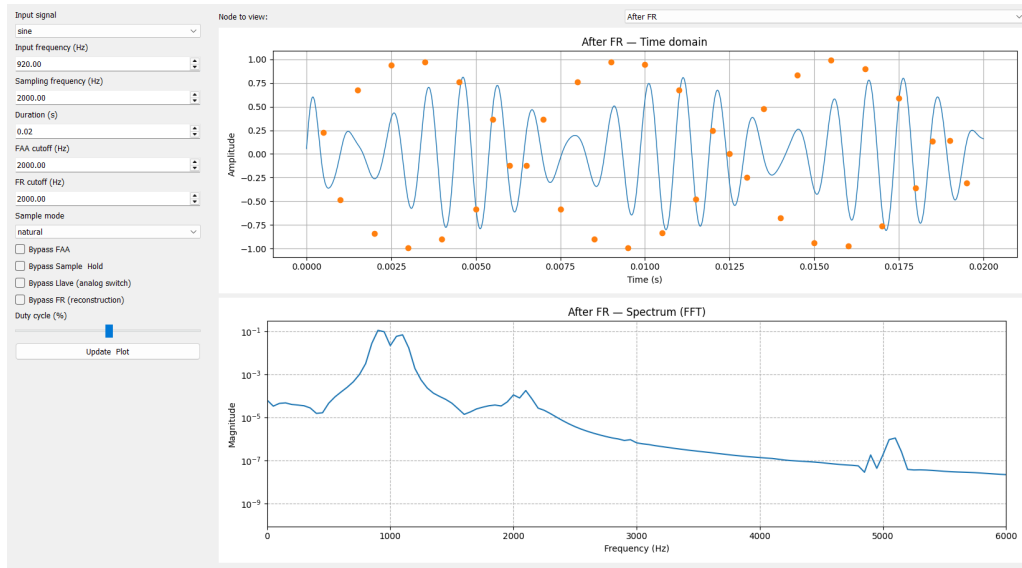


Figura 5.1: Interfaz gráfica - Ejemplo de señal en la GUI

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>