

Simulación de Digitalización y Filtrado de Señales

Edwin Iñiguez Moncada

14/09/2025

1. Implementación

1.1. Fundamentos Teóricos: El Proceso de Conversión ADC

La **digitalización de una señal** es el corazón de la conversión de Analógico a Digital (ADC), un proceso esencial en la ingeniería de sistemas. Este procedimiento transforma una señal continua en tiempo y amplitud en una señal discreta que puede ser procesada por una computadora. La implementación de la práctica simula este proceso a través de dos pasos principales: el **muestreo** y la **cuantización**.

- **Muestreo:** El código simula esta etapa con la función `downsample`. Por ejemplo, en el archivo `adc_y_filtro.py`, una señal con una alta frecuencia de muestreo de 50 Hz (`FS_HIGH`) es **decimada** a una frecuencia de salida más baja de 25 Hz (`FS_OUT`), reduciendo el número de puntos de datos a la mitad.
- **Cuantización:** Este paso, simulado por la función `quantize`, discretiza la amplitud de la señal. En la práctica, se utilizan 8 bits (`BITS=8`), lo que resulta en $2^8 = 256$ niveles de amplitud. El esquema **mid-rise** se eligió sobre **mid-tread** porque **no incluye el cero en el centro de un escalón de cuantización**, sino que lo sitúa entre dos niveles. Esto es particularmente útil para señales con variaciones alrededor de cero, ya que el error de cuantización es simétrico con respecto a este punto.

1.2. Dataset Utilizado

Para la práctica se utilizó el **MotionSense Dataset**, disponible públicamente en GitHub [1]. Este conjunto de datos contiene series de tiempo capturadas con los sensores de acelerómetro y giroscopio de un iPhone 6s durante diferentes actividades físicas (caminar, subir escaleras, correr, etc.). La señal del acelerómetro en el eje X se empleó como variable principal para simular el proceso de digitalización (ADC) y aplicar los filtros de estimación.

Este dataset es adecuado porque refleja un escenario real de adquisición de datos analógicos provenientes de sensores MEMS, con ruido inherente que hace relevante el análisis de técnicas de filtrado.

1.3. Filtros de Estimación: Observador de Luenberger y Filtro de Kalman

1.3.1. Observador de Luenberger

El **Observador de Luenberger**, propuesto por David Luenberger en 1964, es una técnica de estimación de estados. Se utiliza para sistemas dinámicos que no son totalmente observables, es decir, donde no todos los estados internos son medidos directamente por sensores. Su propósito es reconstruir el estado completo del sistema a partir de las mediciones disponibles y las entradas conocidas. El observador funciona como un “gemelo digital” del sistema real.

Su funcionamiento se basa en un modelo matemático del sistema. Si el sistema real es representado por las ecuaciones de estado y salida:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t)$$

El observador de Luenberger utiliza una versión idéntica del modelo, pero con un término de corrección adicional:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t))$$

donde $\hat{x}(t)$ es la estimación del estado, $u(t)$ es la entrada, $y(t)$ es la salida medida y L es la **ganancia del observador**. Este término de corrección es la clave del filtro: compara la salida real ($y(t)$) con la salida estimada ($C\hat{x}(t)$) y ajusta la estimación en la dirección correcta para reducir el error. La ganancia L se puede sintonizar para que el observador converja al estado real tan rápido como se desee.

Aplicación en la práctica: En la práctica proporcionada, el Observador de Luenberger se utiliza para estimar una señal suavizada a partir de los datos ruidosos del sensor. El filtro ajusta la estimación para que siga la tendencia de la señal de entrada, mitigando el ruido.

1.3.2. Filtro de Kalman

El **Filtro de Kalman**, desarrollado por Rudolf E. Kalman en 1960, es un **algoritmo de estimación de estado óptimo** para sistemas lineales dinámicos que operan en presencia de ruido. A diferencia del Observador de Luenberger, el Filtro de Kalman no solo proporciona una estimación de estado, sino que también estima la incertidumbre de esa estimación.

El filtro opera en un ciclo de dos pasos:

1. **Predicción:** Predice el próximo estado del sistema y la incertidumbre asociada (covarianza del error de predicción), basándose en el modelo de estado anterior.
2. **Actualización:** Utiliza una nueva medición ruidosa para corregir la predicción. La **ganancia de Kalman** (K) determina cuánto peso se le da a la nueva medición. Si el ruido de la medición es alto, el filtro confiará más en la predicción; si es bajo, confiará más en la medición.

El filtro de Kalman se basa en un modelo probabilístico que considera el ruido del proceso (Q) y el ruido de la medición (R). Su objetivo es minimizar la varianza del error de estimación.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - H\hat{x}_{k|k-1})$$

donde $\hat{x}_{k|k}$ es el estado estimado, y_k es la medición, H es la matriz de medición y K_k es la ganancia de Kalman.

Aplicación en la práctica: El Filtro de Kalman se aplica para estimar la posición verdadera de la señal. Al considerar las propiedades estadísticas del ruido, logra una estimación más precisa que el Observador de Luenberger en este escenario, como se refleja en el menor error de cuantización.

2. Resultados y Análisis

Las gráficas generadas en el notebook demuestran el impacto de la digitalización y el filtrado.

2.1. Señal Filtrada y Decimada

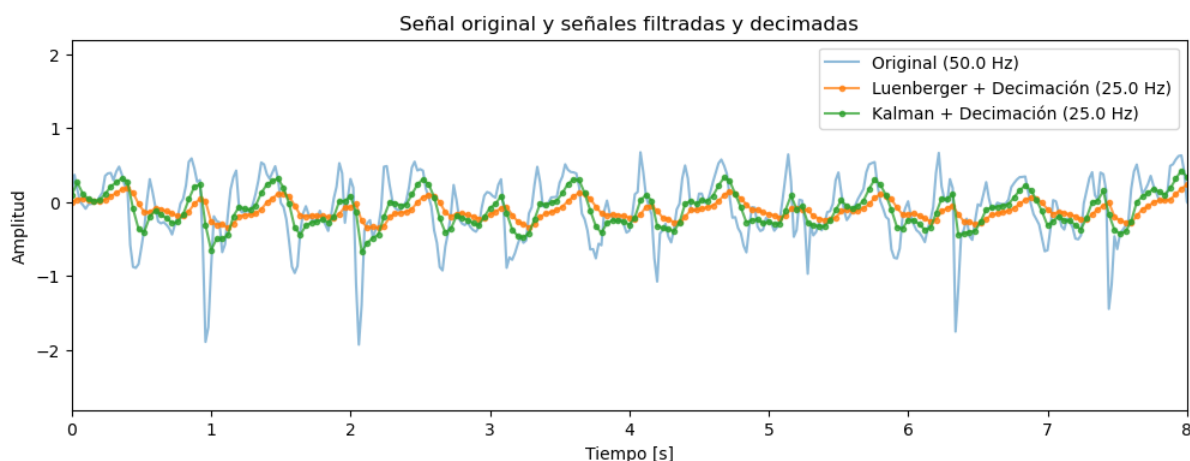


Figura 1: Comparación de la señal original y las señales filtradas y decimadas por ambos métodos.

La figura superior ilustra el resultado del filtrado y la decimación. Se observa el efecto de **suavizado** de ambos filtros, que eliminan las variaciones abruptas de la señal original. El Filtro de Kalman tiende a ofrecer un resultado ligeramente más suave, ya que es especialmente eficaz para mitigar el ruido aleatorio.

2.2. Señal Cuantizada

Al aplicar la cuantización, la señal adquiere una apariencia escalonada, como se aprecia en la siguiente gráfica.

Esta imagen muestra cómo la amplitud de la señal se ha limitado a 256 niveles discretos, resultado de la configuración de 8 bits. El **error de cuantización** es visible en los saltos entre los niveles.

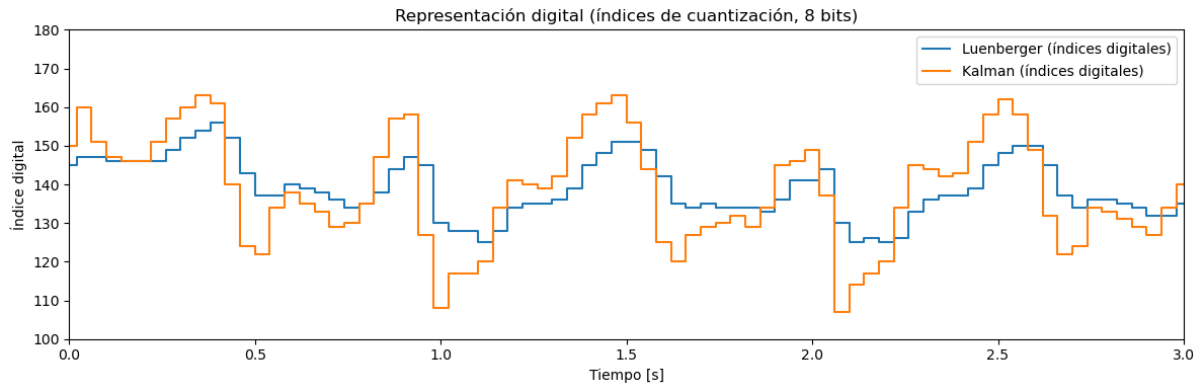


Figura 2: Señales filtradas y posteriormente cuantizadas.

2.2.1. Representación binaria

Se esperaba que hubiera una diferencia mínima en cuanto a los resultados después del filtrado con ambas técnicas pero al ser más flexible el filtrado de Kalman, ha permitido ajustar y eliminar de mejor forma el ruido dentro de la señal.

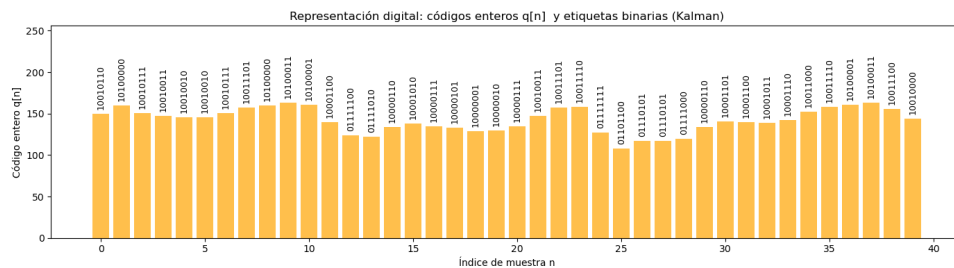


Figura 3: Representación digital: códigos enteros $q[n]$ y etiquetas binarias (Kalman)

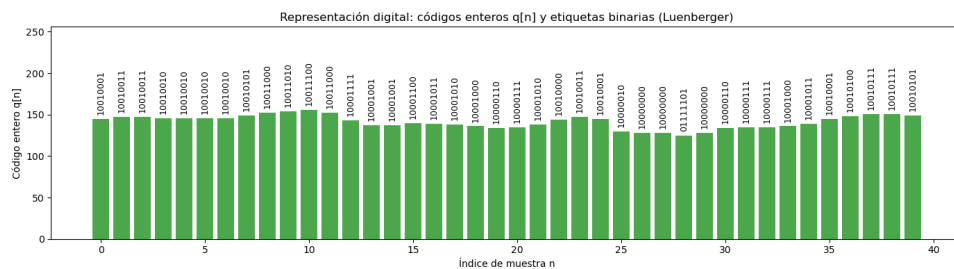


Figura 4: Comparación del Error Cuadrático Medio (MSE) para la cuantización.

El gráfico de barras muestra que el **Filtro de Kalman produce un MSE de cuantización más bajo** que el del Observador de Luenberger. Esto confirma visual y cuantitativamente que, en este escenario con ruido, la estimación del Filtro de Kalman es superior.

Conclusión

Esta práctica ha demostrado cómo la simulación del proceso de conversión ADC, al combinar el **muestreo** y la **cuantización**, transforma una señal continua en una digital.

La inclusión de un paso de filtrado previo es crucial, y el **Filtro de Kalman** se destaca como el método más efectivo, ya que reduce el ruido de manera más eficiente, lo que se traduce en un menor error de cuantización en la señal digital final.

El uso del **MotionSense Dataset** permitió trabajar con datos reales de sensores, lo que da mayor validez al experimento. En particular, la señal del acelerómetro refleja variaciones naturales de movimiento y ruido característico de sensores MEMS. Esto facilitó el análisis del impacto del muestreo y la cuantización en un entorno realista, en lugar de depender únicamente de señales artificiales o simuladas.

Los resultados obtenidos muestran que tanto el **Observador de Luenberger** como el **Filtro de Kalman** cumplen la función de suavizar la señal y mejorar la estimación. Sin embargo, el Kalman sobresale en entornos ruidosos debido a su enfoque probabilístico que equilibra de manera óptima la predicción del modelo y las mediciones observadas. Esto se refleja en un menor **Error Cuadrático Medio (MSE)** y en una señal digital cuantizada con menor distorsión.

3. Citas

Referencias

- [1] Malekzadeh, M. (2018). "MotionSense Dataset." Disponible en: <https://github.com/mmalekzadeh/motion-sense>
- [2] Luenberger, D. G. (1964). "Observing the state of a linear system." *IEEE Transactions on Military Electronics*, 8(2), 74-80.
- [3] Kalman, R. E. (1960). "A new approach to linear filtering and prediction problems." *Journal of Basic Engineering*, 82(1), 35-45.

Anexos

Código Fuente

El código utilizado para esta práctica se encuentra en los siguientes archivos: `adc_y_filtro.py` y `digitalizacionFiltrado.ipynb`.