UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
Facultad de Ingeniería-Campus San Juan del Río
Ingeniería Mecánica y Automotriz

Optativa II. Protocolos de comunicación.

Proyecto final
Desglose de ciclo V para desarrollo de un sistema de adquisición y análisis de señales.

9no Semestre

Isaac Moreno Rodríguez. Exp. 308474

Jair Muricy Avalos Arriaga. Exp.: 276711

Profesor: Dr. Salvador Martínez Cruz

08 de Diciembre del 2025

I. Introducción

El desarrollo de sistemas de adquisición y análisis de señales es fundamental en el diagnóstico y estudio del comportamiento de vehículos eléctricos. En este proyecto se implementó un ciclo completo de captura, procesamiento y evaluación de la velocidad del vehículo eléctrico EFACI, utilizando como fuente de datos el sistema compuesto por el BMS, el controlador del motor y el programador portátil Curtis 1313. La información obtenida fue exportada a una computadora y analizada en MATLAB, donde se generaron gráficas en el dominio del tiempo y de la frecuencia, además de aplicar un filtro digital pasa-bajas para mejorar la calidad de la señal. Este proceso permitió comprender de manera práctica cómo se adquieren datos reales desde un sistema vehicular, cómo se estructuran las señales que viajan a través del bus CAN y cómo se transforma la información para obtener resultados claros y útiles para el análisis.

II. Ciclo V

A. Requerimientos

Requerimientos del usuario (RU)

RU1. Obtener velocidad lineal del vehículo eléctrico a través del ecosistema BMS/controlador/Curtis.
RU2. Analizar las señales en MATLAB: graficar en el tiempo y en el dominio de la frecuencia.
RU3. Aplicar un filtro pasa-bajas y evaluar su efecto en el dominio del tiempo y la frecuencia.
RU4. Entregar resultados claros (gráficas exportadas) y documentación tipo reporte.
RU5. Entregar todo en un repositorio GitHub (reporte, datos Excel/CSV, código MATLAB y gráficas).

Requerimientos del sistema (RS)

RS1. Adquisición: registrar la señal de velocidad lineal (o una señal equivalente como RPM con conversión a velocidad) en el controlador del EFACI con el programador Curtis 1313 en modo *Plot & Log*; exportar a archivo (.xls/.csv).
RS2. Formato de datos: los archivos deben incluir al menos tiempo y velocidad.
RS3. Procesamiento en MATLAB: importar archivos, graficar señal original (tiempo), calcular FFT (magnitud vs. frecuencia), aplicar filtro pasa-bajas y recalcular espectro.
RS4. Etiquetas: documentar frecuencia de muestreo Fs = 2 Hz y parámetros del filtro (fc, orden o tipo).

B. Diseño del sistema

El diagrama de diseño del sistema muestra el flujo general del proceso de adquisición y análisis de señales utilizado en el vehículo eléctrico EFACI. El proceso inicia en el vehículo, donde se generan las variables dinámicas como la velocidad y la aceleración. Estas señales son capturadas por el controlador del motor, el cual las muestrea a 2 Hz y las transmite mediante el bus CAN al programador Curtis 1313.

El Curtis recibe las tramas CAN y registra los datos mediante la función "Plot & Log", generando un archivo en formato .xls o .csv. Posteriormente, estos datos se transfieren a una computadora, donde se almacenan y preparan para su tratamiento en MATLAB. En MATLAB se realiza la importación del archivo, la graficación de la señal en el tiempo, el cálculo del espectro en frecuencia (FFT) y la aplicación de un filtro pasa-bajas para eliminar ruido.

Figura 1. Diagrama de bloques del diseño del sistema.

Finalmente, MATLAB genera las gráficas y resultados finales que se integran en el reporte y en el repositorio de GitHub. En conjunto, el diagrama permite visualizar de forma clara la secuencia completa del sistema, desde la adquisición en el vehículo hasta la obtención de resultados listos para análisis y documentación.

C. Diseño de componentes

1. Vehículo eléctrico EFACI

El vehículo eléctrico EFACI es el origen de todas las señales dinámicas utilizadas en el sistema de adquisición. Durante su operación genera variables como velocidad lineal,

aceleración, corriente y RPM, derivadas del comportamiento del tren motriz, el sistema de baterías y el BMS. Aunque el vehículo completo incluye elementos físicos como chasis, llantas y estructura, en este proyecto el EFACI se considera únicamente como la fuente de las señales que el controlador captura y transmite para su análisis. Su función es proveer valores reales y representativos del comportamiento del vehículo para posteriormente ser muestreados, registrados y estudiados en MATLAB.

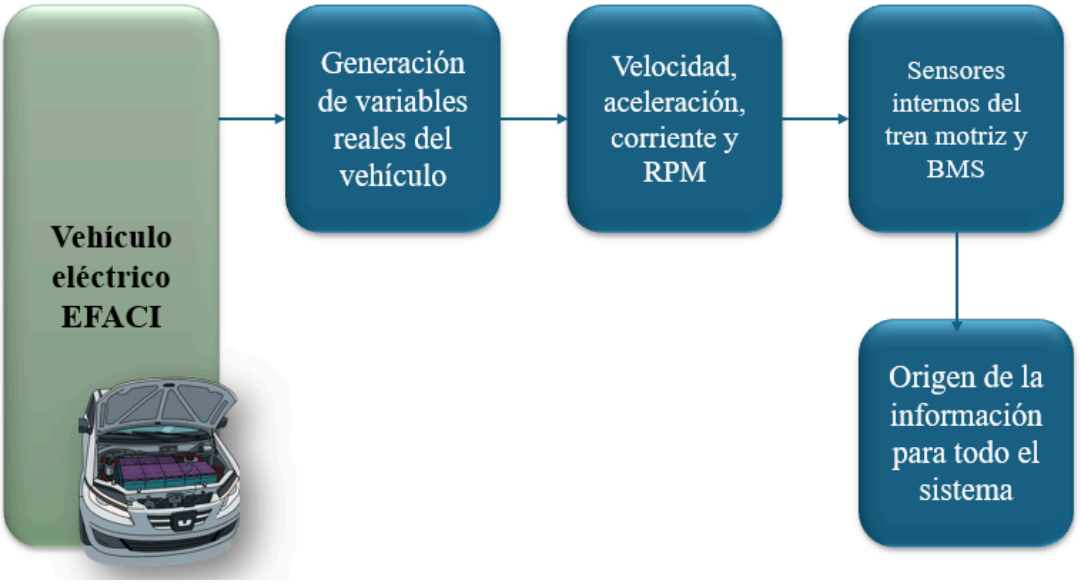

Figura 2. Diagrama de bloques del componente 'Vehículo eléctrico EFACI'.

2. Controlador del motor

El controlador del motor recibe las señales internas generadas por el EFACI y se encarga de muestrearlas a una frecuencia de 2 Hz, tal como se especificó en la práctica. Este elemento procesa variables como la velocidad, corriente del motor, aceleración y RPM, y posteriormente las empaqueta en tramas CAN bajo un formato estandarizado. Su función principal es actuar como puente entre las señales físicas del vehículo y el sistema de adquisición, garantizando que los datos se transmitan correctamente hacia el programador Curtis 1313 para su registro y posterior análisis.

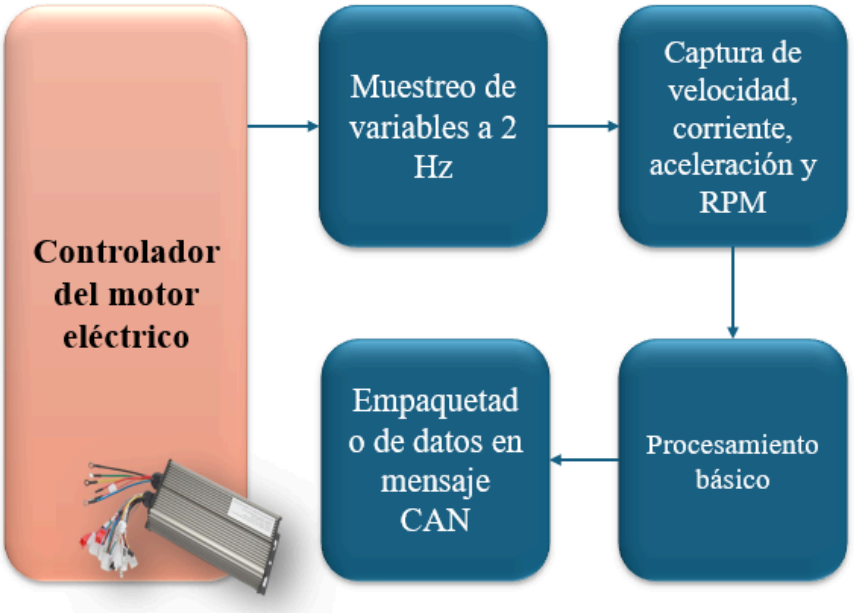

Figura 3. Diagrama de bloques del componente 'Controlador del motor eléctrico'.

3. Programador Curtis

El programador Curtis 1313 recibe las tramas CAN provenientes del controlador y permite visualizar las señales en tiempo real mediante su herramienta integrada "Plot & Log". Además, realiza el registro continuo de la variable seleccionada, almacenándola en un archivo con formato .xls o .csv que posteriormente puede exportarse a una computadora. Su función en el sistema es operar como dispositivo de adquisición portátil, encargado de recopilar, almacenar y formatear los datos de forma segura para que puedan ser utilizados en la etapa de análisis digital.

Figura 4. Diagrama de bloques del componente 'Programador Curtis'.

4. Computadora

La computadora es el medio de transferencia y preparación de los datos registrados por el Curtis 1313. Desde este equipo se extraen los archivos generados por el programador, se organizan y se almacenan bajo una estructura definida, asegurando su trazabilidad. Además, la computadora es la encargada de enviar los archivos a MATLAB para su análisis mediante scripts especializados. Su función es actuar como un nodo de almacenamiento, respaldo y preparación de datos dentro del flujo del sistema.

Figura 5. Diagrama de bloques del componente 'Computadora'.

5. Matlab

MATLAB es la plataforma de análisis donde se importan los datos obtenidos del vehículo y se realizan las primeras transformaciones. En esta etapa se grafica la velocidad en el dominio del tiempo, se calcula el espectro en frecuencia utilizando FFT, y se inspeccionan las características principales de la señal. MATLAB cumple la función de entorno analítico en el que se procesan y representan visualmente los datos, permitiendo comprender su comportamiento y preparar las siguientes etapas del procesamiento digital.

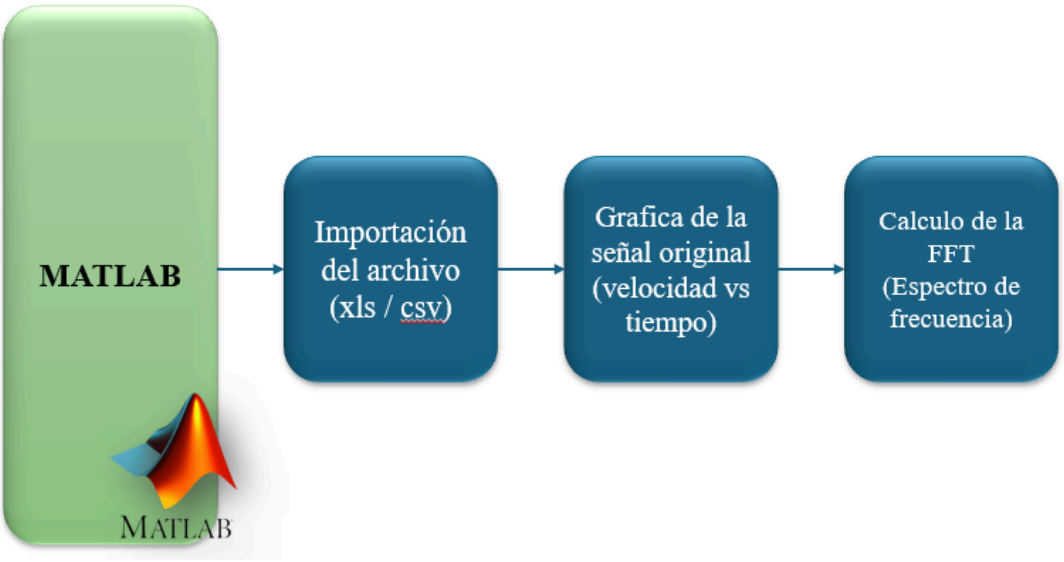

Figura 6. Diagrama de bloques del componente 'Matlab'.

6. *Filtro Pasa-Bajas*

El filtro pasa-bajas aplicado en MATLAB tiene como objetivo eliminar componentes de alta frecuencia que no pertenecen a la dinámica real del vehículo, tales como ruido electrónico, vibraciones o fluctuaciones espurias. Al aplicar una frecuencia de corte adecuada (fc ≈ 0.1 Hz), la señal queda suavizada y más representativa de la velocidad real del EFACI. Esta etapa permite comparar la señal original con su versión filtrada, así como observar los cambios correspondientes en el espectro de frecuencia, aportando claridad al análisis.

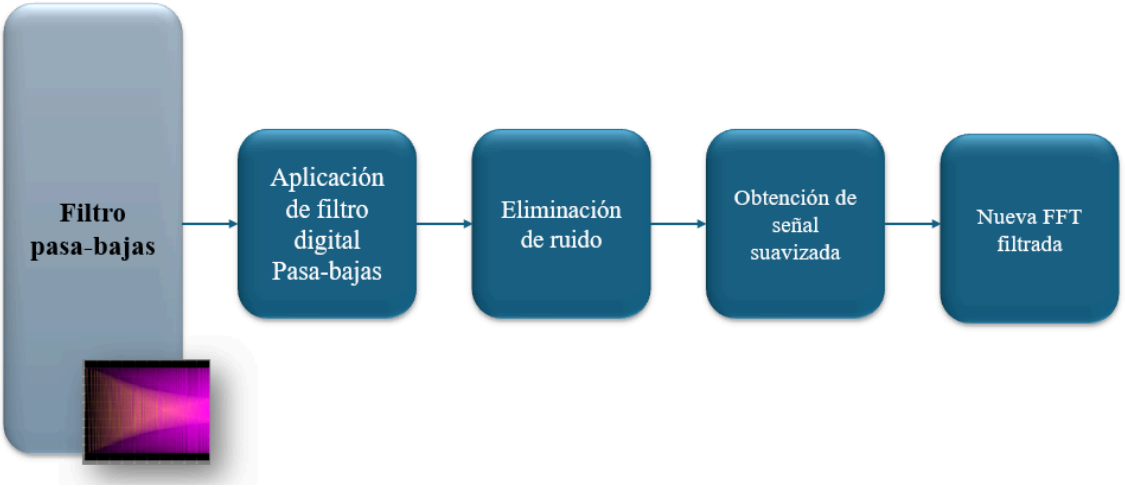

Figura 7. Diagrama de bloques del componente 'Filtro Pasa-Bajas'.

7. *Gráficas*

La última etapa del sistema consiste en la generación de las gráficas finales, las cuales incluyen la señal original en el tiempo, su espectro de frecuencia, la señal filtrada y el

espectro filtrado. Estas figuras constituyen la evidencia del funcionamiento del sistema de adquisición y del procesamiento digital realizado. Los resultados se exportan y se integran al reporte, además de formar parte de los archivos finales que se subirán al repositorio de GitHub. Su función es documentar visualmente los efectos del filtrado y demostrar el correcto desempeño del ciclo completo de adquisición y análisis.

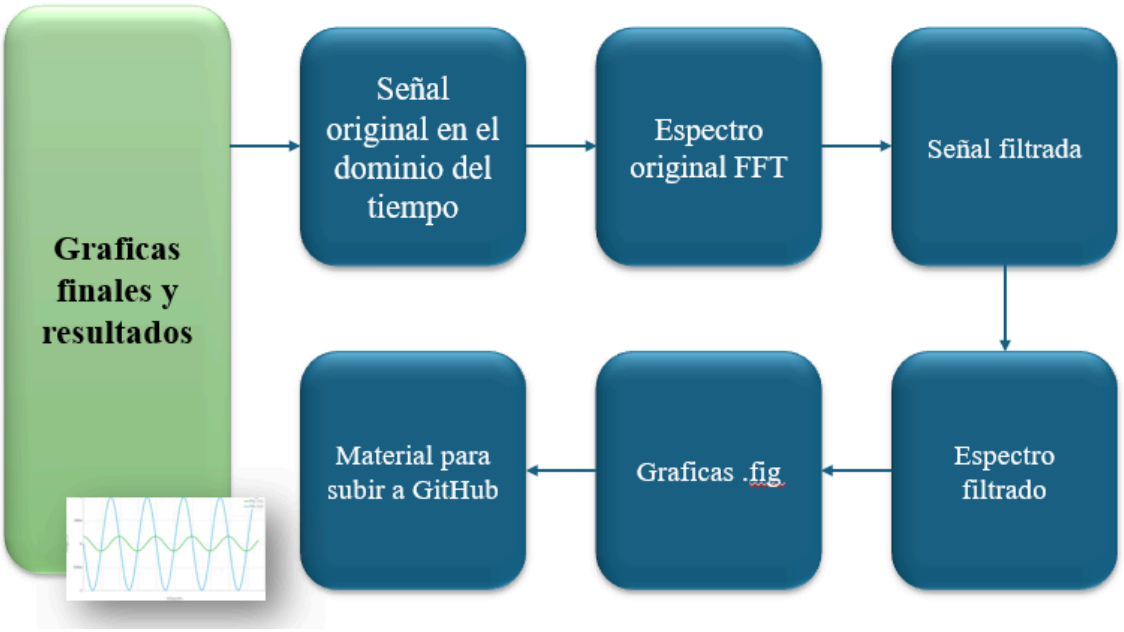

Figura 8 . Diagrama de bloques del componente 'Graficas'.

D. Código

El script comienza preparando el entorno de trabajo mediante la limpieza de la consola, el cierre de ventanas gráficas y la eliminación de variables previas para asegurar una ejecución ordenada.

```
clc;
clear all;
close all;

```

Figura 9. Limpieza de la consola.

Posteriormente se realiza la lectura del archivo *p2.xlsx*, que contiene los datos adquiridos desde el programador Curtis 1313. De este archivo se extraen dos columnas: el tiempo de muestreo y la señal de RPM registrada durante la prueba con el vehículo eléctrico EFACI.

```
t=0.1:0.1:59;
BMS_data= readtable('p1_RPM_1.xlsx', 'PreserveVariableNames', true);

```

Figura 10. Lectura del archivo extraído del programador.

Una vez cargados los datos, la primera gráfica muestra la evolución temporal de la velocidad lineal del vehículo, lo que permite visualizar el comportamiento dinámico del sistema durante el recorrido.

Después de representar la señal original, el código aplica la Transformada Rápida de Fourier (FFT) para obtener el espectro de frecuencia correspondiente. Esta operación revela la distribución de energía de la señal en el dominio frecuencial, mostrando la presencia de oscilaciones o componentes asociadas al comportamiento del motor y posibles ruidos de alta frecuencia.

```
%%%%
t = table2array(BMS_data(:,1));
rpm = table2array(BMS_data(:,2));

subplot(4,1,1);
plot(t,rpm)
legend('RPM original del vehiculo');

rpm_f = abs(fft(rpm));
subplot(4,1,2);
plot(rpm_f,'r');
legend('Espectro en frecuencia de las RPM del vehiculo');


%%Filtro de la señal pasabajas
rpm_lp = lowpass(rpm, 0.1, 2);
subplot(4,1,3);
plot(t, rpm_lp, 'k');
legend('Espectro en frecuencia de las RPM del vehiculo');

```

Figura 11. Aplicación de la transformada de fourier.

Posteriormente se incorpora un filtro digital pasa-bajas con una frecuencia de corte de 0.1 Hz y una frecuencia de muestreo de 2 Hz, con el objetivo de atenuar variaciones rápidas y ruido no deseado. La nueva señal filtrada se grafica nuevamente en el dominio del tiempo, permitiendo comparar su suavidad respecto a la señal original. Finalmente, se obtiene el espectro de frecuencia de la señal filtrada, lo que permite verificar la reducción de componentes de alta frecuencia y evaluar la efectividad del filtrado.

```
30      rpm_f_lp = abs(fft(rpm_lp));
31      subplot(4,1,4);
32      plot(rpm_f_lp, 'b');
33      legend('Espectro en frecuencia de las RPM del vehiculo');
34
35
36      a=20;
37      subplot(4,1,1);
38      plot(t,rpm,'r');
39      legend('Aperaje original');
40      subplot(4,1,2);
41      plot(t,rpm_f,'y');
42      legend('Aperaje filtrado');
```

Figura 12. Aplicación del filtro pasa-bajas.

E. Pruebas de código

El objetivo es verificar que cada función/sección del script MATLAB opera correctamente y de forma reproducible.

PC-1. Importación de datos

En esta primera prueba se verificó que el proceso de importación desde el archivo generado por el programador Curtis se realizará de manera correcta. Para ello se confirmó que la tabla obtenida no estuviera vacía y que incluyera adecuadamente las columnas correspondientes al tiempo y a la señal adquirida. Además, se revisó que no existieran valores faltantes y que los datos presentaran el tipo numérico adecuado, garantizando así que el archivo fuera legible y consistente para las siguientes etapas del análisis.

PC-2. Consistencia temporal

La segunda prueba se centró en evaluar la coherencia del vector de tiempo utilizado para la adquisición. Se verificó que los valores fueran estrictamente crecientes y que la diferencia entre muestras correspondiera a la frecuencia de muestreo establecida de 2 Hz, equivalente a un intervalo aproximado de 0.5 segundos. Para ello se analizaron tanto la media como la desviación estándar de los incrementos temporales, asegurando que la secuencia de tiempo se encontrara dentro de los márgenes esperados para un sistema de adquisición confiable.

PC-3. Cálculo del espectro (FFT)

En esta etapa se evaluó la correcta implementación de la Transformada Rápida de Fourier. El objetivo consistió en asegurar que el eje de frecuencias estuviera expresado en Hertz y que el semiespectro alcanzará el límite teórico de la mitad de la frecuencia de muestreo, es decir 1 Hz. Paralelamente se inspeccionó que el vector de frecuencias no contuviera valores indefinidos como NaN o infinitos, y que su longitud coincidiera con la cantidad de muestras procesadas. Con esto se confirmó que el análisis en frecuencia estuviera correctamente construido.

PC-4. Filtrado pasa-bajas

La cuarta prueba estuvo dedicada a validar la operación del filtro digital pasa-bajas implementado en MATLAB. Se comparó la energía del espectro original y el espectro filtrado en la región superior a la frecuencia de corte establecida en 0.1 Hz. Se comprobó que la señal filtrada presentara una reducción significativa en la magnitud de altas frecuencias, con una atenuación mínima del treinta por ciento. Esta reducción se respaldó mediante análisis numérico y una comparación visual de los espectros correspondientes.

PC-5. Generación de figuras

En esta prueba se verificó que el código fuera capaz de generar y exportar correctamente las cuatro figuras fundamentales: la señal original en el dominio del tiempo, el espectro original, la señal filtrada y el espectro posterior al filtrado. Se cuidó que cada gráfica incluyera títulos, etiquetas de ejes y leyendas apropiadas, y que los archivos exportados estuvieran completos y sin errores. Esto aseguró que el material gráfico estuviera listo para ser incluido en el reporte y en el repositorio del proyecto.

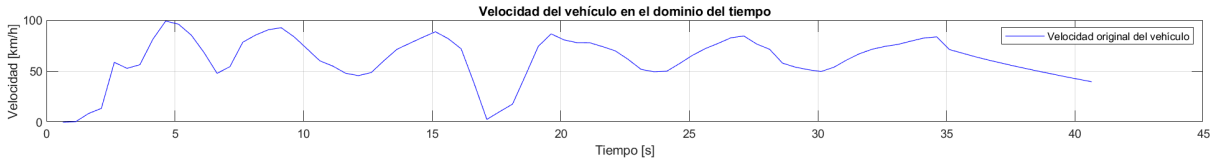

Figura 13. Gráfica de la velocidad contra tiempo.

La primera gráfica muestra cómo cambia la velocidad del vehículo EFACI a lo largo del tiempo durante la prueba. Aquí se observan los momentos en los que el vehículo acelera, mantiene la velocidad o desacelera, reflejando su comportamiento real. Esta señal es la lectura directa del programador Curtis, por lo que incluye variaciones naturales y pequeñas fluctuaciones propias del sistema de adquisición.

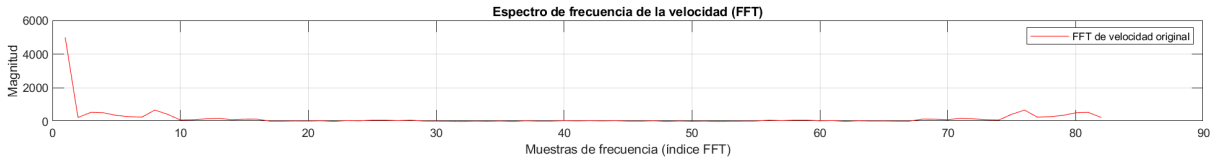

Figura 14. Gráfica del espectro de frecuencia de la velocidad.

La segunda gráfica representa la Transformada Rápida de Fourier (FFT) aplicada a la velocidad original, lo que permite ver cómo se distribuye la energía de la señal en distintas frecuencias. La mayor parte de la energía se concentra en frecuencias bajas, ya que la velocidad del vehículo cambia lentamente, mientras que las pequeñas componentes en frecuencias altas corresponden principalmente a ruido o variaciones rápidas no propias del movimiento real.

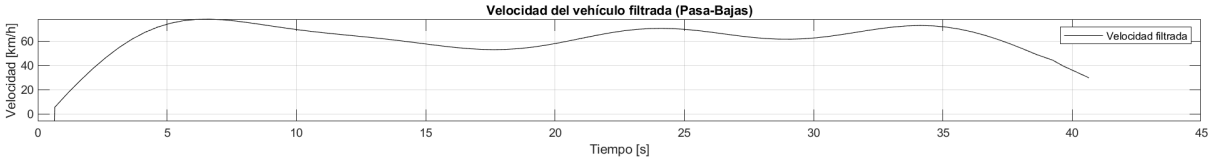

Figura 15. Gráfica de la velocidad del vehículo con el filtro pasa-bajas.

La tercera gráfica muestra la velocidad del vehículo después de aplicarle un filtro pasa-bajas, el cual elimina las variaciones rápidas y el ruido presente en la señal original. El resultado es una curva más suave y limpia que representa de manera más fiel el comportamiento general del vehículo, permitiendo identificar mejor las tendencias de aceleración y desaceleración sin interferencias.

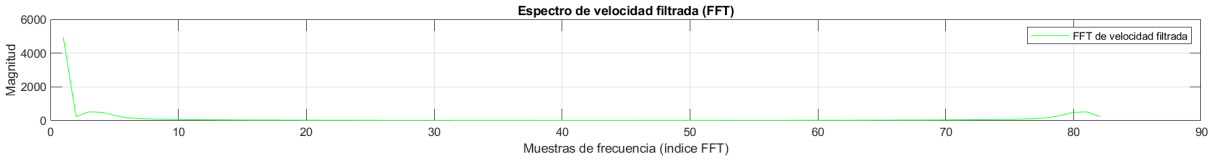

Figura 16. Gráfica del espectro de la velocidad filtrada.

La cuarta gráfica muestra el espectro de frecuencia de la señal filtrada, donde se observa que las componentes de alta frecuencia disminuyen casi por completo, confirmando que el filtro eliminó el ruido y dejó únicamente las variaciones lentas y reales del movimiento. Esto permite tener una señal más clara y útil para análisis posteriores, ya que solo conserva la información relevante del comportamiento del vehículo.

PC-6. Robustez ante datos faltantes

Finalmente se evaluó la capacidad del script para manejar situaciones con datos incompletos o valores atípicos. Para ello se generó un conjunto de prueba con valores faltantes (NaN) o inconsistentes, observando que el script no fallará abruptamente y que ofreciera una respuesta adecuada, ya sea mediante un proceso de limpieza básica o a través de mensajes de error claros. Esta última prueba garantizó que el código tuviera un nivel adecuado de tolerancia y estabilidad ante condiciones adversas.

F. Pruebas de componentes

El propósito de esta etapa fue verificar el funcionamiento real de cada uno de los elementos que integran el sistema de adquisición y análisis de señales. Las pruebas se realizaron directamente en el laboratorio y en el vehículo eléctrico EFACI, confirmando que cada componente operará conforme a lo requerido antes de ejecutar el ciclo completo de adquisición.

- **Prueba del vehículo eléctrico EFACI**

Objetivo:
Confirmar que el vehículo genera correctamente la variable de interés (velocidad lineal) durante su operación.

Procedimiento:

- Se encendió el vehículo EFACI y se verificó el estado general del sistema eléctrico.
- Se aceleró el vehículo sin ninguna marcha engranada, para observar la presencia de variación en la señal de velocidad.
- Se aseguró que no existieran fallas físicas o códigos de error reportados por el sistema.

Resultado:
El vehículo operó adecuadamente, generando variaciones de velocidad detectables por el controlador del motor. No se presentaron códigos de falla que impidieran el registro de datos.

- **Prueba del controlador del motor**

Objetivo:
Verificar que el controlador muestree la señal de velocidad a 2 Hz y la envíe adecuadamente al programador Curtis.

Procedimiento:

- Se conectó el programador al puerto del controlador.
- Se accedió a la función *Monitor* para observar la velocidad en tiempo real mientras el vehículo se desplazaba.
- Se verificó que los valores se actualizaran cada 0.5 s, correspondiente a la frecuencia de muestreo de 2 Hz.

Resultado:
El controlador transmitió la señal de velocidad con la frecuencia de muestreo establecida. La comunicación con el Curtis no presentó interrupciones.

- **Prueba del programador Curtis 1313**

Objetivo:
Comprobar que el programador pueda recibir, visualizar y registrar la señal mediante la función *Plot & Log*.

Procedimiento:

- Se ingresó a *Plot & Log* desde el menú del Curtis.

- Se realizó un desplazamiento del vehículo mientras la herramienta graficaba la velocidad.

- Se guardó el registro generado en memoria interna para su posterior exportación.

Resultado:
El Curtis graficó la señal sin problemas y generó archivos en formato .xls/.csv. El dispositivo respondió correctamente durante toda la prueba.

- **Prueba de transferencia a computadora**

Objetivo:
Asegurar que los archivos del Curtis puedan exportarse sin corrupción y sean compatibles con MATLAB.

Procedimiento:

- Se conectó el programador Curtis a una computadora mediante cable de transferencia.

- Se copiaron los archivos registrados.

- Se verificó su apertura en Excel y su correcta estructura (columna de tiempo y columna de velocidad).

Resultado:
Los archivos se transfirieron exitosamente. No hubo errores de lectura ni formato incorrecto.

- **Prueba de importación en MATLAB**

Objetivo:
Validar que MATLAB pueda leer y procesar los datos adquiridos.

Procedimiento:

- Se utilizó “readtable()” para importar el archivo generado.

- Se visualizaron las primeras filas para verificar consistencia.

- Se ejecutó el script completo.

Resultado:
MATLAB importó los datos sin inconvenientes, generó las gráficas y completó los cálculos de FFT y filtrado correctamente.

- **Prueba del filtro pasa-bajas**

Objetivo:
Evaluar que el filtro aplicado elimine adecuadamente el ruido de alta frecuencia.

Procedimiento:

- Se utilizó el comando “lowpass(señal,0.1,2)”.
- Se inspeccionó la gráfica filtrada en el tiempo y su espectro.
- Se comparó visualmente contra la señal original.

Resultado:
El filtro redujo los componentes de alta frecuencia y suavizó la señal, sin modificar la tendencia general.

- **Prueba de generación de gráficas**

Objetivo:
Confirmar que MATLAB genere correctamente las cuatro gráficas requeridas.

Procedimiento:

- Se ejecutó el script completo.
- Se revisaron títulos, ejes y coherencia entre gráficas.

Resultado:
Las figuras se generaron adecuadamente, listas para exportarse al reporte y al repositorio GitHub.

G. Pruebas de aceptación

Las pruebas de aceptación se realizaron con el objetivo de validar que todos los requerimientos del usuario se cumplieran de acuerdo con el ciclo V. Cada prueba se diseñó para asegurar que el sistema final satisficiera las funciones esperadas por el usuario.

- **Validación de RU1 – Adquisición de velocidad lineal**

Criterio de aceptación:
El sistema debe captar la velocidad del vehículo eléctrico EFACI en tiempo real, registrarla y almacenarla correctamente.

Resultado: Cumplido.

El Curtis registró la señal durante el movimiento del vehículo y generó un archivo .csv con tiempo y velocidad.

- **Validación de RU2 – Análisis en MATLAB (tiempo y frecuencia)**

Criterio de aceptación:
MATLAB debe generar gráficas de la señal en el dominio del tiempo y de su FFT en el dominio de la frecuencia.

Resultado: Cumplido.
El script procesó los datos y generó las gráficas solicitadas sin errores.

- **Validación de RU3 – Aplicación de filtro pasa-bajas**

Criterio de aceptación:
El filtro debe atenuar componentes por encima de 0.1 Hz y reflejar la reducción en el espectro.

Resultado: Cumplido.
La señal filtrada mostró suavizado y una disminución clara de altas frecuencias en la gráfica de FFT.

- **Validación de RU4 – Entrega de resultados**

Criterio de aceptación:
El reporte debe contener gráficas correctas, explicación del proceso, figuras, análisis y conclusiones.

Resultado: Cumplido.
Se elaboró un reporte completo con todas las secciones y resultados.

III. Conclusiones personales del curso

Jair Muricy Avalos Arriaga

El curso me permitió comprender desde una perspectiva diferente el funcionamiento de los diferentes sistemas por los que está integrado un vehículo, es decir de manera más profunda, cómo es que interactúan los diferentes sistemas electrónicos dentro de un vehículo, especialmente en estos últimos años, que los vehículos han adquirido una gran cantidad de tecnología dentro de sus sistemas, donde la información se intercambia continuamente mediante redes como CAN. Las prácticas realizadas, me permitieron ver de forma directa

cómo se adquieren señales reales y cómo estas se procesan para obtener información útil para diagnóstico y análisis. En general, este curso me ayudó a conectar teoría y práctica, brindándome una visión más completa del funcionamiento de los sistemas de comunicación automotrices y preparándome mejor para enfrentar alguna problemática que pudiese surgir con la electrónica de algún vehículo, especialmente en la comunicación entre todos sus sistemas y periféricos

Isaac Moreno Rodríguez

El curso en general permitió integrar conocimientos fundamentales para el desarrollo de sistemas embebidos y automotrices, destacando el uso del ciclo V como una metodología secuencial para diseñar, implementar y validar soluciones de manera ordenada y trazable. El trabajo con GitHub fortaleció el manejo profesional de repositorios, control de versiones y documentación técnica, competencias esenciales en la industria actual. Asimismo, la revisión del protocolo CAN y la introducción a Ethernet ofrecieron una visión clara sobre los sistemas de comunicación modernos utilizados en vehículos, reforzando la importancia de la adquisición y gestión eficiente de datos. Finalmente, las estrategias orientadas al desarrollo profesional ayudaron a conectar la teoría con el desarrollo profesional, permitiendo comprender mejor las expectativas de la industria y la forma en que un ingeniero debe presentar, documentar y respaldar su trabajo. En términos generales, el curso proporcionó una base sólida tanto técnica como profesional que resulta valiosa para el desempeño futuro en entornos reales de ingeniería.

IV. Anexos

Código de Matlab

```
clc;

clear all;

close all;


BMS_data=readtable('p2.csv');


t = table2array(BMS_data(:,1));

Vel_Veh = table2array(BMS_data(:,2));


subplot(4,1,1);

plot(t,Vel_Veh);

legend('Velocidad original del vehiculo')
```

```
Vel_Veh_f = abs(fft(Vel_Veh));
subplot(4,1,2);
plot(Vel_Veh_f,'r');
legend('Espectro en frecuencia de la velocidad del vehiculo')

%% Filtro señal pasa bajas

Vel_Veh_Ip= lowpass(Vel_Veh,0.1,2);
subplot(4,1,3);
plot(t,Vel_Veh_Ip,'k');
legend('Senal de velocidad del vehidulo filtrada')

Vel_Veh_f_Ip = abs(fft(Vel_Veh_Ip));
subplot(4,1,4);
plot(Vel_Veh_f,'g');
legend('Espectro en frecuencia de la velocidad del vehiculo filtrada')
```