

Automatización de mediciones para la obtención de diagramas de Bode con control simultáneo de instrumentos mediante MATLAB

RESUMEN: Este trabajo presenta un sistema automatizado de medición implementado en MATLAB, que permite el control simultáneo de un generador de funciones y un osciloscopio mediante comandos SCPI, utilizando comunicación a través de los protocolos GPIB y USB. El objetivo principal es automatizar la obtención de diagramas de Bode, minimizando el esfuerzo manual y mejorando la repetibilidad. El sistema ejecuta barridos de frecuencia configurables desde 10 Hz hasta 100 MHz, registra automáticamente las mediciones de magnitud y fase de la señal de prueba, genera las gráficas correspondientes en tiempo real. Como caso de validación, se empleó un filtro pasa-bajas RC de primer orden, permitiendo verificar la coherencia de los resultados con el modelo teórico. Esta herramienta reduce el error humano, optimiza el tiempo de adquisición de datos y ofrece una alternativa eficaz para la caracterización práctica de circuitos electrónicos en entornos académicos o de laboratorio.

PALABRAS CLAVE: automatización, diagrama de Bode, caracterización de sistemas, GPIB, MATLAB.



Colaboración

Willy Omar Torres Melchor; Ramón Chávez Bracamontes; Julio Isaac Flores Figueroa; Jaime Jalomo Cuevas, Tecnológico Nacional de México / Instituto Tecnológico de Ciudad Guzmán

Fecha de recepción: 25 de agosto de 2025

Fecha de aceptación: 10 de noviembre de 2025

ABSTRACT: This work presents an automated measurement system implemented in MATLAB, which enables the simultaneous control of a function generator and an oscilloscope using SCPI commands and communication via GPIB and USB protocols. The main objective is to automate the acquisition of Bode plots, reducing manual effort and improving repeatability. The system performs configurable frequency sweeps from 10 Hz to 100 MHz, automatically records magnitude and phase measurements of the test signal, and generates real-time plots. As a validation case, a first-order RC low-pass filter was used to confirm the accuracy of the results compared to the theoretical model. This tool minimizes human error, reduces measurement time, and provides an efficient solution for practical characterization of electronic circuits in academic or laboratory environments.

KEYWORDS: automation, Bode diagram, characterization of systems, GPIB, MATLAB.

INTRODUCCIÓN

En los laboratorios de caracterización y pruebas electrónicas es común la necesidad de automatizar mediciones utilizando diversos instrumentos, como generadores de funciones, fuentes de alimentación, multímetros u osciloscopios. La automatización permite realizar análisis con mayor exactitud y agilidad.

En los últimos años, la instrumentación virtual ha cobrado relevancia como una solución eficaz para optimizar procesos de medición [1]. Por ejemplo, en [2] se implementó un sistema de análisis de impedancia basado en software para sensores QCM, demostrando la capacidad de los entornos virtuales para obtener datos precisos en tiempo real.

Aunque la automatización de instrumentos se remonta a décadas atrás [3], actualmente está impulsada por estándares modernos como SCPI (Standard Commands for Programmable Instruments), cuyos beneficios han sido plenamente adoptados en la última década. El uso de estos comandos se ha consolidado como estándar en la automatización de sistemas de medición, facilitando la interoperabilidad entre instrumentos de distintos fabricantes y reduciendo la dependencia de control manual. En [4] se demostró que el uso de SCPI sobre interfaces como GPIB, USB o LAN permite implementar arquitecturas de control robustas y repetibles en entornos industriales complejos, donde la ejecución manual de pruebas es inviable.

En este contexto, la obtención manual de diagramas de Bode representa un proceso lento, laborioso y susceptible a errores, ya que implica ajustar frecuencias en el generador, capturar los valores de entrada y salida desde el osciloscopio para cada punto, anotar los datos y procesarlos posteriormente en una hoja de cálculo o software auxiliar, lo que conlleva una considerable inversión de tiempo y limita la repetibilidad del procedimiento. En respuesta a esta problemática, se desarrolló una herramienta que automatiza el barrido de frecuencias, registra de manera continua las mediciones de magnitud y fase, y genera gráficas en tiempo real, permitiendo un análisis más preciso, repetible, eficiente y considerablemente más ágil de los datos obtenidos.

El objetivo de este trabajo es desarrollar e implementar un sistema de medición automatizado mediante MATLAB, que controle simultáneamente un generador de funciones y un osciloscopio, para generar diagramas de Bode sin intervención manual. Como beneficio cuantificable, la herramienta reduce el tiempo total de caracterización en más de un 90 % respecto al procedimiento manual, con mayor precisión y repetibilidad en las pruebas.

A diferencia de trabajos previos centrados exclusivamente en simulaciones o instrumentación virtual, el sistema propuesto integra control simultáneo de múl-

tiples instrumentos físicos, adquisición real de datos y graficación automatizada en tiempo real, lo cual permite replicabilidad inmediata en entornos de laboratorio educativo.

En consecuencia, se plantea la siguiente hipótesis

La implementación de un sistema automatizado de medición, que controle instrumentos mediante MATLAB y protocolos GPIB/USB, permitirá obtener diagramas de Bode de manera más eficiente, precisa y repetible en comparación con el procedimiento manual tradicional.

MATERIAL Y MÉTODOS

Una adecuada selección y configuración de los instrumentos de medición es fundamental para garantizar la fidelidad de los datos. En [5] se demostró que, en sistemas automatizados, una elección apropiada de generadores, osciloscopios y multímetros, junto con una calibración precisa, puede reducir significativamente el error sistemático, mejorando la confiabilidad en mediciones de magnitud y fase. De acuerdo con la documentación oficial de MathWorks [6], la interfaz GPIB permite conectar instrumentos de prueba directamente a MATLAB, facilitando el envío de comandos y la adquisición de datos.

En esta sección se describe detalladamente el procedimiento seguido para la automatización de mediciones utilizando MATLAB, el protocolo GPIB, y los instrumentos empleados. Se explican las conexiones físicas, la configuración del entorno de trabajo y el desarrollo del código encargado de realizar el barrido de frecuencia, la adquisición de datos y finalmente sus gráficas.

Conexión de instrumentos y arquitectura del sistema

Para establecer comunicación entre la computadora y los equipos de medición, se utilizó el siguiente arreglo:

El generador de funciones Tektronix AFG3251

[7] fue uno de los instrumentos principales en este sistema. Este equipo permite generar señales senoidales de frecuencia y amplitud controladas mediante comandos SCPI. El instrumento se ilustra en la Figura 1.



Figura 1. Generador de señales Tektronix AFG3251.

Fuente: Recuperada de <https://www.axiomtest.com/Signal-Generators/ARB,-Data,-Function,-Pattern-and-Pulse/Tektronix/AFG3251/Arbitrary-...-Function-Generator,-240-MHz,-1-Ch/>

Para controlar este generador desde la computadora, se utilizó un adaptador USB- GPIB Keysight 82357B [8], el cual permite la transmisión de comandos SCPI bajo el estándar IEEE 488.2. Este se puede observar en la Figura 2.



Figura 2. Adaptador USB-GPIB Keysight 82357B.

Fuente: recuperada de <https://www.keysight.com/us/en/product/82357B/usb-gpib-interface-high-speed-usb-2-0.html>.

El segundo instrumento utilizado fue el osciloscopio Agilent TECHNOLOGIES DSO7034A [9], el cual se conectó directamente a la computadora mediante un cable USB tipo B. Este instrumento se muestra en la Figura 3.

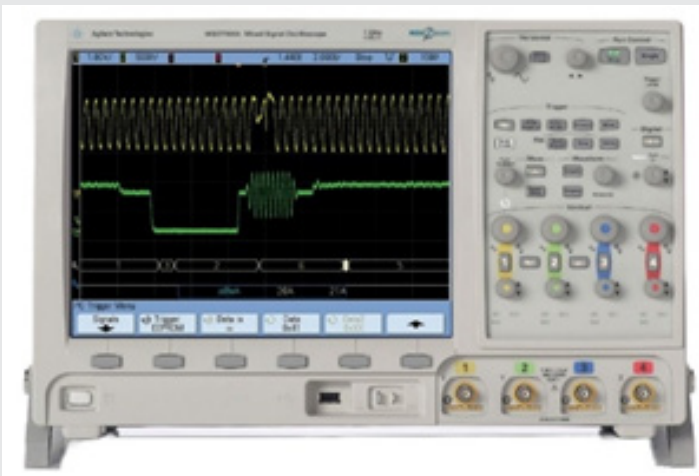


Figura 3. Osciloscopio Agilent TECHNOLOGIES DSO7034A.

Fuente: recuperada de <https://www.keysight.com/us/en/product/DSO7034A/oscilloscope-350-mhz-4-analog-channels.html>.

Finalmente, el esquema de conexión completa del sistema se muestra en la Figura 4. En él se puede observar cómo se conectan entre sí los instrumentos, el circuito bajo prueba y la computadora.

Configuración del entorno de trabajo

Según [10], para comenzar a trabajar con GPIB en MATLAB, es necesario identificar los dispositivos VISA conectados y establecer la comunicación mediante objetos de interfaz.

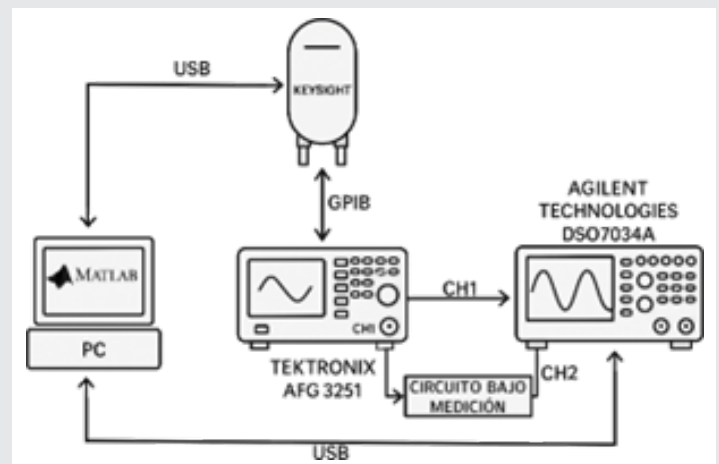


Figura 4. Diagrama general de conexión del sistema de medición.

Fuente: Elaboración propia.

Para lograr la comunicación entre MATLAB y los instrumentos se deben realizar los siguientes pasos.

Instalación de Keysight IO Libraries Suite

Se instaló el software IO Libraries Suite de Keysight, el cual permite reconocer automáticamente los instrumentos conectados vía USB y GPIB. Este software proporciona una interfaz gráfica donde se listan todos los dispositivos disponibles a través del estándar VISA [11], indicando su dirección, protocolo y estado de conexión, similar a las soluciones propuestas por National Instruments en su guía de automatización con NI-VISA [12]. Aunque en este trabajo se utilizó MATLAB, también existen alternativas como PyVISA [13], que permiten la comunicación con instrumentos mediante Python.

Tal como se muestra en la Figura 5, ambos instrumentos fueron detectados correctamente y se encuentran listos para su control desde MATLAB.

Reconocimiento de instrumentos en MATLAB

En caso de que algún dispositivo no sea detectado, se recomienda consultar [14], la instalación de drivers VISA y el estado del adaptador GPIB mediante la herramienta de diagnóstico del IO Libraries Suite [11].

Una vez que los instrumentos fueron detectados en IO Libraries Suite, se procedió a establecer la comunicación desde MATLAB utilizando los siguientes comandos:

Para el osciloscopio Agilent DSO7034A, que se comunica por USB:

```
obj1 = visa('agilent', 'USB0::2391::5941::MY48260514::0::INSTR');
fopen(obj1);
```

Para el generador de funciones Tektronix AFG3251, conectado mediante GPIB:

```
F1 = gpib('AGILENT', 7, 2); % Board index 7,
dirección GPIB 2
fopen(F1);
```

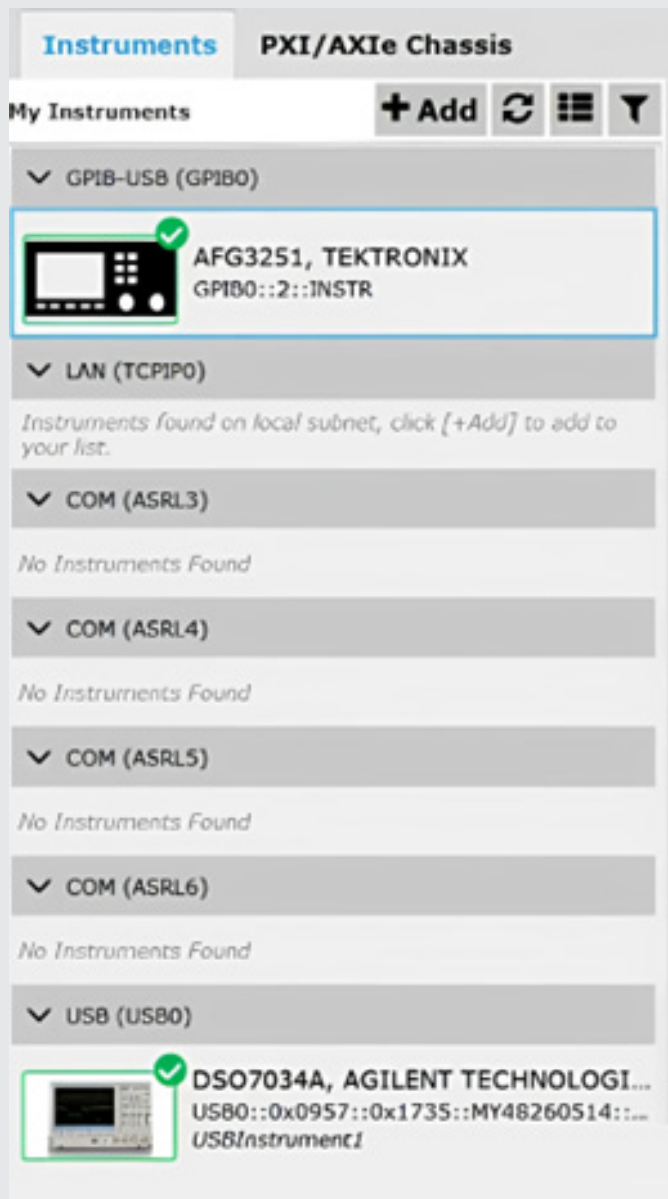



Figura 5. Detección de instrumentos en Keysight IO Libraries Suite.

Fuente: Recuperada de captura de pantalla del software Keysight IO Libraries Suite.

Estas instrucciones permitieron abrir los canales de comunicación desde MATLAB hacia cada instrumento.

Parámetros de prueba y funcionamiento del código

El sistema automatizado de medición está basado en un barrido de frecuencia configurable, durante el cual se miden automáticamente la magnitud y fase entre las señales de entrada y salida del circuito bajo prueba. A continuación, se describen los parámetros definidos al inicio del código y su función dentro del proceso.

Parámetros principales del barrido

En la Tabla 1 se observan los parámetros configurables del sistema, los cuales pueden ajustarse según los requerimientos del análisis del circuito.

Tabla 1. Parámetros configurables del sistema.

| Parámetro | Descripción |
|--------------------|--|
| <u>fi</u> | Frecuencia inicial del barrido (mínimo 10 Hz). |
| <u>ff</u> | Frecuencia final del barrido (máximo 100 MHz). |
| <u>amp</u> | Amplitud de la señal generada (valor pico en volts). |
| <u>factor_paso</u> | Factor que determina el paso dentro de cada década. |

Fuente: Elaboración propia.

donde:

- fi y ff establecen los límites inferior y superior del barrido de frecuencia.
- amp define la amplitud de la señal senoidal generada. Este valor se puede modificar según la sensibilidad del circuito bajo prueba o la capacidad del osciloscopio.
- factor_paso determina cuántos puntos se medirán por cada década logarítmica.

Procedimiento para el barrido de frecuencias

El código realiza un ciclo que recorre el rango de frecuencias de forma logarítmica, evaluando la respuesta del circuito en cada punto. En cada iteración se llevan a cabo los pasos que se muestran a continuación, y que también se detallan en el diagrama mostrado en la Figura 6.

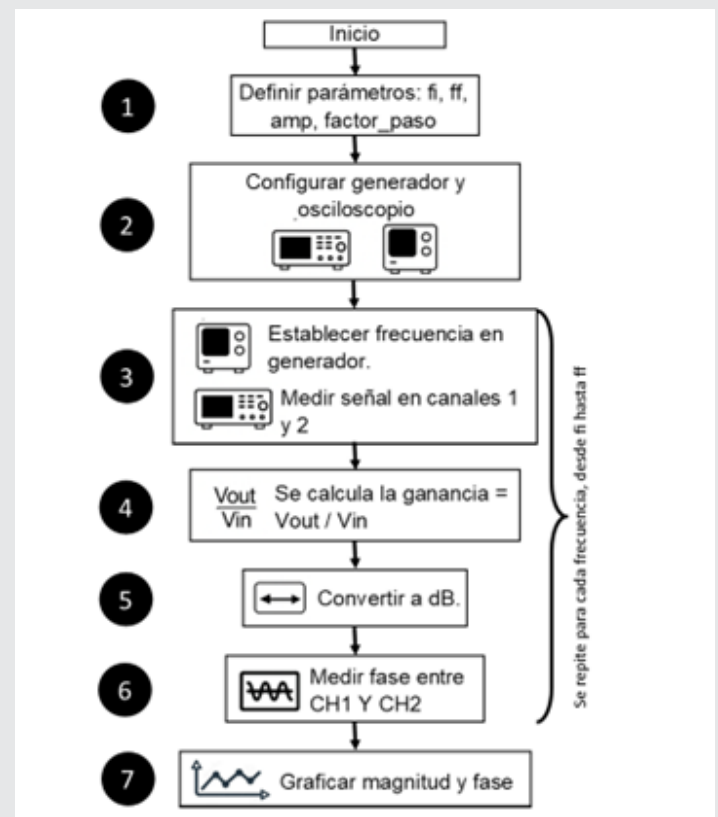


Figura 6. Ejemplo del ciclo de barrido de frecuencias

Fuente: Elaboración propia.

1. Se definen los parámetros de medición deseados para el análisis en el código de MATLAB.
2. Se envían los comandos SCPI al generador y al osciloscopio para establecer los nuevos parámetros.
3. Se mide la señal de salida (Vpp) en el canal 2 del osciloscopio.
4. Se calcula la ganancia como el cociente entre Vpp de salida y Vpp de entrada.
5. Se convierte la ganancia a decibels (dB).
6. Se mide la fase entre las señales del canal 1 (entrada) y canal 2 (salida).
7. Los datos de frecuencia, magnitud y fase se almacenan en matrices para su posterior procesamiento y generación de las gráficas correspondientes.

El sistema fue programado en MATLAB utilizando estructuras de control como bucles for y funciones específicas para la comunicación con instrumentos (fopen, fprintf, query). Para cada frecuencia del barrido, se establece la señal de salida en el generador utilizando comandos SCPI. A continuación, se adquieren las señales de entrada y salida desde el osciloscopio, midiendo sus valores pico a pico (Vpp) por medio de funciones de consulta (query) a los canales CH1 y CH2.

La ganancia se calcula como el cociente entre Vout y Vin (Ecuación 1), y posteriormente se convierte a decibels utilizando la fórmula que se presenta en la Ecuación 2.

$$Magnitud = \left(\frac{V_{out}}{V_{in}} \right) \quad \text{Ec. (1)}$$

$$G_{dB} = 20 \cdot \log_{10} \left(\frac{V_{out}}{V_{in}} \right) \quad \text{Ec. (2)}$$

La expresión presentada en la Ecuación 2 es una conversión general de ganancia a escala logarítmica y es válida para cualquier sistema lineal. Es importante no confundir esta fórmula con la pendiente del diagrama de Bode, que sí depende del orden del sistema: 20 dB/década para primer orden, 40 dB/década para segundo orden, y así sucesivamente.

La fase se determina con base en el retardo temporal entre las señales medidas, tomando en cuenta la configuración de escala temporal y el desfase entre los dos canales.

Visualización de resultados

Una vez completado el barrido de frecuencias, los datos almacenados en matrices son procesados para graficar la respuesta en frecuencia del circuito. Se generan dos diagramas de Bode: uno representa la magnitud como una relación directa entre Vout y Vin, y el otro la expresa en decibels.

El diagrama de Bode es una herramienta esencial para analizar la respuesta en frecuencia de sistemas lineales,

ya que permite visualizar de forma clara el comportamiento de magnitud y fase respecto a la frecuencia. Según [15], este tipo de representación facilita la evaluación de la estabilidad y el diseño de sistemas de control retroalimentados.

Las gráficas fueron construidas en MATLAB utilizando funciones como semilogx para el eje de frecuencia logarítmica, y subplot para organizar ambas curvas en una sola figura. Además, se añadieron elementos de formato como etiquetas de ejes (xlabel, ylabel), títulos y cuadrículas (grid on) para facilitar el análisis visual de los resultados.

Este paso final permite observar de forma inmediata el comportamiento del circuito bajo prueba a lo largo del rango de frecuencias definido por el usuario.

RESULTADOS

Para validar el funcionamiento del sistema automatizado, se utilizó un circuito pasa-bajas de primer orden, compuesto por una resistencia en serie de 1 kΩ y un capacitor de 1 μF a tierra. Este arreglo, mostrado a continuación en la Figura 7, fue empleado como circuito bajo prueba durante el barrido de frecuencias.

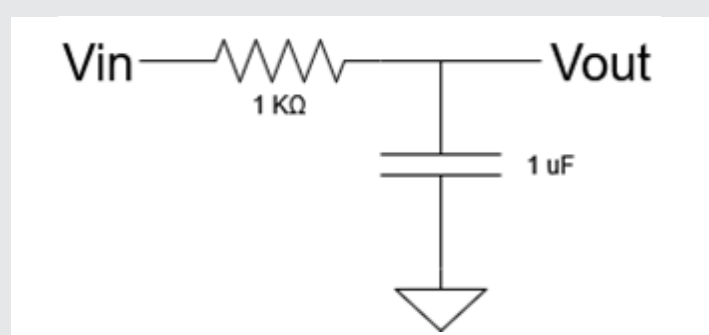


Figura 7. Arreglo del circuito pasa-bajas.

Fuente: Elaboración propia.

Los parámetros establecidos para llevar a cabo el análisis en frecuencia se especifican en la Tabla 2.

Tabla 2. Valores establecidos para el análisis en frecuencia.

| Parámetro | Descripción |
|-------------|-----------------------------|
| fi | 10 Hz. |
| ff | 1 kHz. |
| amp | 1 V. |
| factor_paso | 0.5 (Dos puntos por década) |

Fuente: Elaboración propia.

En las Figuras 8 y 9 se muestran los diagramas de Bode obtenidos a partir del barrido de frecuencia. En la gráfica de la Figura 8, la ganancia se mantiene prácticamente constante en el rango de frecuencias bajas y comienza a disminuir de manera progresiva a partir de aproximadamente 50 Hz, alcanzando su frecuencia

de corte a los 159 Hz, y en la Figura 9, se observa un descenso progresivo en la fase conforme aumenta la frecuencia, alcanzando un desplazamiento cercano a $\pm 90^\circ$ en frecuencias altas, lo cual coincide con el comportamiento característico de un filtro pasa-bajas de primer orden.

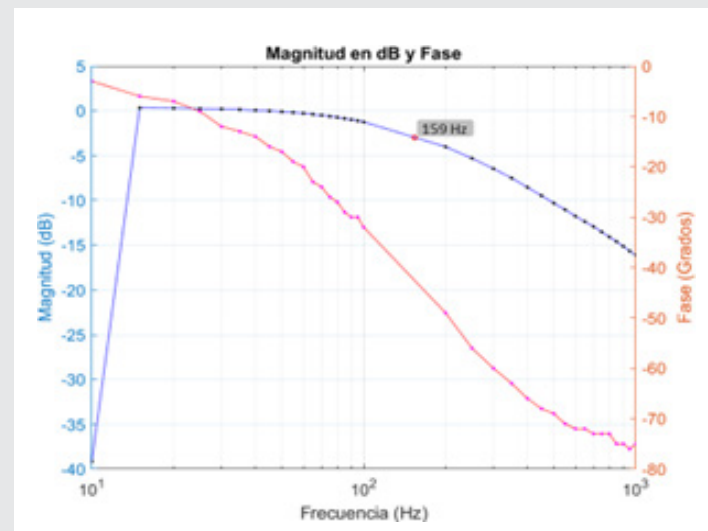


Figura 8. Diagrama de Bode: magnitud en decibels
Fuente: Captura de pantalla en MATLAB.

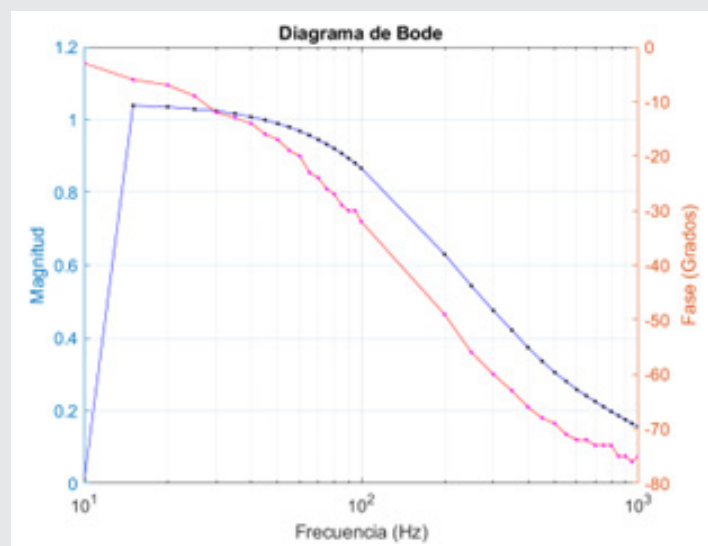


Figura 9. Diagrama de Bode: magnitud como ganancia directa (V_{out}/V_{in}).
Fuente: Captura de pantalla en MATLAB.

La frecuencia de corte está dada por la ecuación (3).

$$f_c = \left(\frac{1}{2\pi RC} \right) \quad \text{Ec. (3)}$$

Sustituyendo los valores utilizados en el experimento ($R = 1k\Omega$, $C = 1\mu F$) se obtiene un valor teórico de 159.15 Hz, el cual se aproxima al resultado experimental observado mediante el sistema automatizado.

Para reforzar la validación de los resultados, se simuló el mismo circuito pasa-bajas en el software LTspice,

utilizando los mismos valores de resistencia y capacitancia. Los resultados obtenidos en la simulación fueron comparados con las mediciones experimentales, mostrando un comportamiento consistente en términos de magnitud y fase.

Las Figuras 10 y 11 muestran la respuesta simulada del circuito en LTSpice, donde se confirma la frecuencia de corte teórica y la pendiente característica de atenuación, en concordancia con las mediciones realizadas por el sistema automatizado.

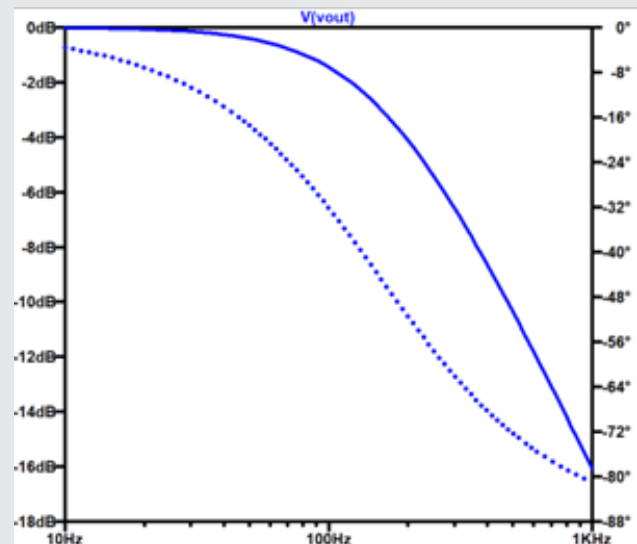


Figura 10. Gráfica de Simulación en LTSpice.
Fuente: Captura de pantalla en LTSpice.

Figura 11. Valores obtenidos en simulación. Fuente: Captura de pantalla en LTSpice.

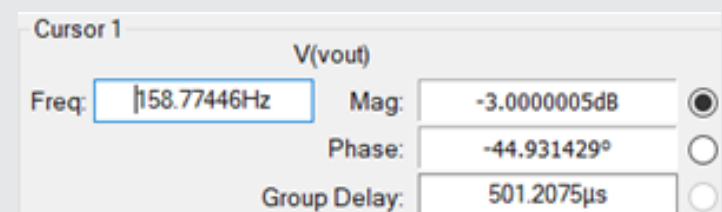


Figura 11. Valores obtenidos en simulación. Fuente: Captura de pantalla en LTSpice.

La siguiente tabla (Tabla 3) muestra una comparativa entre los resultados de cálculo teórico, simulado y experimental. (Los datos experimentales fueron estimados visualmente a partir de las gráficas obtenidas.)

Tabla 3. Comparativa de sistemas.

| Concepto | Teórico | Simulación | Experim. |
|---------------------|-----------|------------|----------|
| Frecuencia de corte | 159.15 Hz | 158.77 Hz | 159 Hz |
| Ganancia | -16.06 dB | -16 dB | -16 dB |

Fuente: Elaboración propia.

Para el barrido de frecuencias realizado en este experimento, el procedimiento manual implicaba aproximadamente 15 minutos, considerando el ajuste del generador, la captura de mediciones desde el osciloscopio y su anotación manual. A este tiempo se agregaban entre 5 y 10 minutos más para el procesamiento de los datos y la generación de los diagramas de magnitud y fase en una hoja de cálculo. En total, el proceso completo podía tomar entre 20 y 25 minutos por prueba. Con el sistema automatizado desarrollado, el mismo barrido se ejecuta en menos de un minuto, incluyendo la adquisición y graficación de los datos, lo que representa una reducción superior al 95% en el tiempo total de análisis. Esta optimización no solo permite realizar múltiples ensayos en menor tiempo, sino que también mejora la precisión y confiabilidad de los resultados.

RESULTADOS

Para validar el sistema automatizado, se eligió un circuito RC pasa-bajas de primer orden debido a su simplicidad y disponibilidad de componentes en el laboratorio escolar. Este tipo de circuitos son frecuentemente usados en cursos introductorios de electrónica y teoría de control. Esta elección permitió construir un caso práctico representativo, con materiales comunes y asequibles para los estudiantes.

Dado que, para este caso práctico, los valores del circuito ($R = 1\text{ k}\Omega$, $C = 1\text{ }\mu\text{F}$) generan una frecuencia de corte cercana a 159 Hz, se limitó el barrido experimental a un máximo de 1 kHz con el fin de observar con mayor resolución la transición en magnitud y fase, así como visualizar claramente el polo del sistema, característico de los filtros de primer orden. Esta decisión facilita el análisis y la validación del proceso automatizado de medición, especialmente en entornos educativos.

En filtros pasa-bajas de primer orden, una vez superada la frecuencia de corte, la magnitud de la respuesta disminuye a razón de 20 dB por década, acercándose asintóticamente a cero sin alcanzarlo completamente. La fase continúa descendiendo hasta aproximarse a -90° en frecuencias muy altas. Esta evolución, si bien no fue medida directamente, puede predecirse con alta precisión y se encuentra respaldada por las gráficas obtenidas dentro del rango evaluado.

Aunque la arquitectura del sistema permite operar hasta 100 MHz, este rango se reserva para futuros ensayos con circuitos que demanden análisis en bandas superiores. El sistema automatizado desarrollado es lo suficientemente versátil como para adaptarse al estudio de cualquier circuito eléctrico cuya respuesta en frecuencia se desee analizar, incluyendo configuraciones de segundo orden, redes resonantes o sistemas activos.

Las tolerancias de los componentes utilizados ($1\text{ k}\Omega \pm 5\%$, $1\text{ }\mu\text{F} \pm 10\%$) y el nivel de precisión del osciloscopio ($< 1\%$) permiten mantener los errores dentro de márgenes aceptables en contextos educativos y de laboratorio. Como trabajo futuro, se plantea extender la validación a circuitos más complejos, analizar un rango de frecuencias más amplio y

formalizar el tratamiento de la incertidumbre mediante herramientas estadísticas.

Los resultados obtenidos mediante el sistema automatizado muestran una buena correspondencia con el comportamiento teórico esperado para un filtro pasa-bajas de primer orden. La frecuencia de corte determinada experimentalmente se aproxima al valor calculado mediante la ecuación teórica, y tanto la magnitud como la fase presentan una respuesta típica de este tipo de circuito.

En comparación con la simulación realizada en LTSpice, las curvas generadas por el sistema en MATLAB presentan una forma similar, especialmente en lo que respecta al punto de transición y la pendiente de atenuación.

Cabe destacar que las ligeras diferencias observadas entre los resultados de la simulación y los experimentales se deben a la presencia de factores físicos que no siempre se modelan con exactitud. Entre ellos se encuentran las capacitancias parásitas, la tolerancia de los componentes electrónicos y otros efectos del entorno que pueden alterar levemente las mediciones reales.

En comparación con los sistemas previos, como el de [16], que integraba MATLAB y GPIB para diagramas de Bode, el presente trabajo ofrece una solución más robusta al consolidar en un solo entorno la adquisición, el procesamiento y la visualización gráfica de magnitud y fase, con mínima intervención manual.

Este tipo de integración ha sido explorado en trabajos recientes como el de [17], quien automatizó la adquisición de diagramas de Bode utilizando MATLAB y equipos físicos mediante comandos SCPI. Sin embargo, su implementación se centró en tareas de adquisición secuencial sin incluir herramientas de visualización en tiempo real ni una interfaz adaptable por parte del usuario. En contraste, el sistema propuesto en este trabajo integra adquisición, procesamiento y graficación simultánea en un entorno único, lo que incrementa su utilidad tanto para la caracterización como para fines didácticos.

Tabla 4. Comparativa de sistemas.

| Atributo | Singh et al. | BodePlotA | Este trabajo |
|--|---------------|------------------|---------------|
| Control de instrumentos físicos. | Si | No | Si |
| Plataforma usada | MATLAB + GPIB | MATLAB (teórico) | MATLAB + SCPI |
| Visualización en tiempo real | Limitada | Si simulada | Si (real) |
| Automatización sin intervención manual | Parcial | N/A | Completa |
| Enfoque educativo y replicabilidad | No explícito | Limitado | Alto |

Fuente: Elaboración propia.



De acuerdo con [18], los sistemas de medición electrónica deben ser flexibles y adaptables, permitiendo su aplicación tanto en entornos educativos como industriales.

Aunque el sistema aún no ha sido implementado formalmente en una asignatura, el prototipo fue presentado a estudiantes de nivel ingeniería en el Tecnológico Nacional de México, campus Ciudad Guzmán, como parte de una sesión demostrativa (Figura 12). Esta experiencia permitió observar un mayor interés y una mejor comprensión de conceptos clave como la frecuencia de corte, el comportamiento de filtros, la respuesta en frecuencia y la interpretación de diagramas de Bode. La posibilidad de visualizar resultados en tiempo real, sin recurrir a procedimientos manuales tediosos, convierte a esta herramienta en un recurso valioso para complementar la teoría de control con experiencias prácticas tangibles. Con el objetivo de fomentar la replicabilidad y el uso didáctico de esta herramienta, el código fuente completo del sistema automatizado, así como las instrucciones básicas para su configuración, han sido publicados en un repositorio en línea [19]. Esto permite a otros docentes, investigadores o estudiantes implementar la solución en sus propios laboratorios sin depender de licencias adicionales ni configuraciones complejas. El acceso al repositorio se encuentra detallado en la sección de referencias.



Figura 12. Sesión demostrativa del sistema automatizado con estudiantes de ingeniería en el TecNM, campus Ciudad Guzmán.

Fuente: Captura propia

CONCLUSIONES

El sistema automatizado desarrollado en este trabajo ha demostrado ser una herramienta eficaz para la caracterización de circuitos electrónicos mediante barridos de frecuencia. Su implementación en MATLAB, con control simultáneo de un generador de funciones y un osciloscopio, permitió la obtención automática de diagramas de Bode precisos, reflejando un comportamiento coherente con los modelos teóricos y las simulaciones realizadas.

A diferencia de otros trabajos reportados en la literatura, como BodePlotA [20], que se enfocan en el análisis gráfico de funciones de transferencia modeladas matemáticamente, el presente sistema extiende dicho enfoque hacia la automatización experimental sobre plataformas físicas reales, utilizando instrumentación externa controlada por comandos SCPI. Este aspecto representa un aporte original, al combinar automatización, adquisición experimental y visualización integrada dentro de un mismo entorno de programación.

Asimismo, se atiende un vacío técnico en la literatura nacional: la carencia de plataformas accesibles que permitan a estudiantes o investigadores automatizar pruebas sobre circuitos reales sin requerir software propietario o licencias costosas. Nuestra solución, al basarse únicamente en MATLAB y drivers gratuitos, ofrece una alternativa reproducible, escalable y educativa.

Los resultados confirman que este procedimiento no solo mejora la precisión de las mediciones, sino que elimina la necesidad de realizar procesos manuales repetitivos, reduciendo los tiempos de ejecución y análisis en más del 90 %. Además, permite repetir las pruebas de forma flexible, lo que facilita el aprendizaje, la validación experimental y el análisis de sistemas lineales desde una perspectiva más práctica. De este modo, el usuario puede concentrarse en la interpretación de resultados y no en la ejecución técnica del proceso.

AGRADECIMIENTOS

Los autores agradecen al Tecnológico Nacional de México, campus Ciudad Guzmán, por las facilidades otorgadas para el uso del equipo de laboratorio necesario en el desarrollo de este trabajo.

BIBLIOGRAFÍA

- [1] A. Goswami, "Performance comparison of instrument automation pipelines using different programming languages," **Scientific Reports**, vol. 13, art. 18579, 2023. [En línea]. Disponible en: <https://pubmed.ncbi.nlm.nih.gov/37903822/>. [Accedido: 05-abr-2025].
- [2] I. Burda, "Quartz crystal microbalance with impedance analysis based on virtual instruments: Experimental study," *Sensors*, vol. 22, art. 1506, 2022.
- [3] A. D. Helfrick and W. D. Cooper, *Instrumentación electrónica moderna y técnicas de medición*, 5ª ed. Englewood Cliffs, NJ, USA: Prentice Hall, 1991.
- [4] B. Sindhu et al., "Digital control of electronic instruments over SCPI," in *Futuristic Trends in Network & Communication Technologies (IIP Series)*, 2024.
- [5] L. Zheng and J. R. Smith, "Influence of instrument selection on measurement reliability in auto-

mated systems,” *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–10, 2023.

[6] MathWorks, “GPIB Interface Overview.” [En línea]. Disponible en: https://www.mathworks.com/help/instrument/g_pib-interface.html [Accedido: 04-abr-2025].

[7] Tektronix, “AFG3251 Function Generator.” [En línea]. Disponible en: <https://www.axiomtest.com/Signal-Generators/ARB,-Data,-Function,-Pattern-and-Pulse/Tektronix/AFG3251/Arbitrary-Function-Generator,-240-MHz,-1-Ch./> [Accedido: 03-abr-2025].

[8] Keysight Technologies, “82357B USB/GPIB Interface High-Speed USB 2.0.” [En línea]. Disponible en: https://www.keysight.com/us/en/product/82357_B/usb-gpib-interface-high-speed-usb-2-0.html [Accedido: 01-abr-2025].

[9] DSO7034A Oscilloscope: 350 MHz, 4 analog channels. [En línea]. Disponible en: https://www.keysight.com/us/en/product/DSO7_034A/oscilloscope-350-mhz-4-analog-channels.html [Accedido: 04-abr-2025].

[10] MathWorks, “Get Started with GPIB Interface.” [En línea]. Disponible en: <https://www.mathworks.com/help/instrument/working-with-the-gpib-interface.html> [Accedido: 01-abr-2025].

[11] Keysight Technologies, “IO Libraries Suite Downloads.” [En línea]. Disponible en: <https://www.keysight.com/us/en/lib/software-detail/computer-software/io-libraries-suite-downloads-2175637.html>.

[12] National Instruments, *Automating Instrument Control with LabVIEW and NI-VISA: User Guide and Reference*. Austin, TX, USA: National Instruments, 2025. [En línea]. Disponible en: <https://www.ni.com/en-us/support/documentation.html> [Accedido: 02-abr-2025].

[13] PyVISA Developers, “PyVISA: Control your instruments with Python.” [En línea]. Disponible en: <https://pyvisa.readthedocs.io> [Accedido: 25-mar-2025].

[14] MathWorks, “Troubleshooting GPIB Interface.” [En línea]. Disponible en: <https://www.mathworks.com/help/instrument/troubleshooting-gpib-interface.html> [Accedido: 01-abr-2025].

[15] H. W. Bode, *Network Analysis and Feedback Amplifier Design*. New York, NY, USA: Van Nostrand, 1945.

[16] R. P., R. Kumar, y R. Prasad, “Automated Bode plot measurements using MATLAB and GPIB instruments,” en *Proc. 2021 IEEE Int. Conf. on Intelligent Technologies (CONIT)*, pp. 1–5, 2021. [En línea]. Disponible en: <https://ieeexplore.ieee.org/document/9568467> [Accedido: 01-abr-2025].

[17] A. Jajaho, “Bode Plots using MATLAB® & Rigol DS1000Z and DG1000Z,” GitHub repository, 2025. [En línea]. Disponible en: <https://github.com/Jajaho/BodePlot> [Accedido: 01-abr-2025].

[18] E. J. Marín García, *Sistemas de medición electrónica: Generalidades y algo más*. Cali, Colombia: Universidad del Valle, Programa Editorial, 2023.

[19] W. Torres, “Código fuente para automatización de mediciones con MATLAB y GPIB,” GitHub repository, 2025. [En línea]. Disponible en: <https://github.com/willytecnm/BodePlot> [Accedido: 04-abr-2025].

[20] F. Barroso, “BodePlotA: Herramienta para el análisis gráfico de sistemas lineales en MATLAB,” *Revista de Aplicaciones en Ingeniería Electrónica*, vol. 12, no. 1, pp. 45–52, 2025.