

Respuesta Frecuencial Experimental de Sistemas Dinámicos Basada en la Transformada Rápida
de Fourier

Diego Andrés Corzo Ruíz

Trabajo de Grado para optar al título de Ingeniero Electrónico

Director

Ricardo Alzate Castaño

Doctorado en Ingeniería Informática y Automática

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones

Bucaramanga

2020

Dedicatoria

Este proyecto está dedicado a todas aquellas personas que apoyan y apoyaron el desarrollo y ejecución de este trabajo de grado.

En especial, a la escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones y a sus docentes, quienes dieron las herramientas necesarias para el desenvolvimiento del mismo.

Agradecimientos

Agradezco primeramente a Dios por permitir este importante momento y, de manera muy especial a mi madre, Carmen Cecilia Corzo Ruíz, que ha sido incondicional y, con su berraquera, ha logrado cumplir el sueño de muchos y el logro de pocos, graduar a su hijo de la Universidad Industrial de Santander. Agradezco a mi familia, que siempre ha sido una base sólida en los momentos difíciles. Un reconocimiento importante lo realizo a mi director de trabajo de grado, por dedicar su tiempo, experiencia y conocimiento en la guía de mi proyecto. Agradezco también a todos los amigos que han formado parte de este largo camino que es una carrera universitaria. A la pastoral Universitaria que siempre tiene un espacio para compartir experiencias con otras realidades. A mis primeros amigos en la universidad; Samuel Lopez y Ricardo Remolina. A aquellos amigos de gaseosa y comida; William Almeida, Carlos Lozano y María Alvarez. A aquellos amigos que durante las épocas más duras del semestre, siempre son apoyo y nos volvemos amigos de "bolsa"; Yuliana Pérez y Christian Orduz. A aquellos que durante mi vida universitaria hemos compartido alegrías y tristezas en noches de traspasado; Sergio Gil y Leonardo Florez. Agradezco a todas aquellas personas que, en esas "noches largas con amaneceres hermosos", han compartido su tiempo y espacio conmigo. Agradezco a los amigos de intercambio y, en especial, a los "Gados" de la "República Capella", quienes me acogieron en su casa, y desde sus actitudes y costumbres brasileñas, aprendí a aprovechar los momentos al máximo. Por último, un agradecimiento gigantesco a la entidad pública que hizo posible todo esto, la Universidad Industrial de Santander.

Tabla de Contenido

Introducción	13
1. Objetivos	17
2. Transformada de Fourier de un sistema	18
2.1. Transformada de Fourier	18
2.1.1. Relación entre las transformadas de Laplace y de Fourier	19
2.1.2. Relación entre la transformada de Fourier y el diagrama de Bode	21
2.1.2.1. Ejemplo 1	23
2.2. Cálculo computacional para la transformada de Fourier	25
2.2.1. TFTD para secuencias finitas	26
2.2.2. Cálculo rápido para la TDF	28
2.2.2.1. Ejemplo 2	29
2.2.2.2. Ejemplo 3	32
3. Diagrama de Bode experimental	37
3.1. Diseño del experimento	37
3.1.1. Circuito RC	37
3.1.2. Estímulo	38

3.1.3. Tarjeta de audio USB	38
3.1.4. Dispositivo de proceso	39
3.1.4.1. Configuración del dispositivo	40
3.1.4.2. Descripción de subrutina para lectura de tarjeta de audio	41
3.1.4.3. Descripción de subrutina para ajuste de formato de datos	41
3.1.4.4. Descripción de subrutina para cálculo de la FFT	41
3.1.4.5. Descripción de librería para visualización de datos	42
3.1.4.6. Flujograma para algoritmo implementado	42
3.1.5. Sistema final implementado	42
3.2. Visualización de resultados	43
4. Recomendaciones	47
5. Trabajo futuro	47
6. Conclusiones	48
Referencias Bibliográficas	50
Apéndices	53

Lista de Figuras

Figura 1.	Representación para el diagrama de Bode de un sistema	14
Figura 2.	Análisis experimental para una respuesta frecuencial	14
Figura 3.	Relación gráfica entre señales y sistemas	21
Figura 4.	Circuito RC	23
Figura 5.	Decimación en el tiempo para TDF	30
Figura 6.	Comparación gráfica entre TFTD y algoritmo FFT validando el uso de N=32 muestras.	32
Figura 7.	Excitación persistente y respuesta temporal del circuito	34
Figura 8.	Espectros de Fourier mediante FFT	35
Figura 9.	Diagramas de Bode teórico y numérico experimental	36
Figura 10.	Diagrama de bloques para sistema experimental	37
Figura 11.	Circuito RC implementado	38
Figura 12.	Tarjeta de audio USB empleada en el experimento	39
Figura 13.	Raspberry Pi 3 modelo B+	40
Figura 14.	Captura de interfaz gráfica para datos en Raspberry	43
Figura 15.	Flujograma para algoritmo de proceso de señal implementado	44
Figura 16.	Sistema final implementado en laboratorio	45

Figura 17. Superposición entre diagramas de Bode teórico y práctico experimental 46

Figura 18. Implementación Python para algoritmo de proceso 54

Lista de Apéndices

	pág.
Apéndice A. Algoritmo programado en Raspberry Pi	53

Resumen

Título: Respuesta Frecuencial Experimental de Sistemas Dinámicos Basada en la Transformada Rápida de Fourier

*

Autor: Diego Andrés Corzo Ruíz **

Palabras Clave: Cálculo Experimental, Diagrama de Bode, Prototipo de Laboratorio, Respuesta Frecuencial.

Descripción: El presente trabajo de grado aborda el estudio analítico, el análisis por simulación y la verificación experimental, para un procedimiento de cálculo de la respuesta frecuencial de sistemas dinámicos, a partir de algoritmos computacionales para determinar la transformada rápida de Fourier. Inicialmente, se muestra la relación conceptual entre las transformadas de Fourier y de Laplace de señales, para posteriormente relacionarlas con el concepto de respuesta frecuencial de sistemas mediante el diagrama de Bode. A partir de ello, se proponen experimentos de simulación que demuestran que las predicciones teóricas obtenidas empleando la función `bode(.)` de MATLAB pueden replicarse a partir del cociente entre las transformadas rápidas de Fourier de las señales de estímulo y respuesta de un sistema. Más aún, se procedió con la configuración en laboratorio para un circuito pasivo del tipo *RC* sobre el cual se aplicó una señal de excitación persistente y se analizó, empleando un sistema de computo basado en el *Raspberry Pi 3 modelo B+*, la estimación para su diagrama de Bode a partir del método mencionado. Los resultados experimentales mostraron una interesante similitud con las predicciones teóricas sugiriendo la efectividad del procedimiento para caracterizar la respuesta frecuencial en sistemas de difícil análisis. Lo anterior constituye la intención del trabajo futuro complementario al presente proyecto.

* Trabajo de grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y telecomunicaciones. Director: Ricardo Alzate Castaño, Doctorado en Ingeniería Informática y Automática.

Abstract

Title: Experimental Frequency Response of Dynamical Systems Based on the Fast Fourier Transform *

Author: Diego Andrés Corzo Ruíz **

Keywords: Experimental Calculation, Bode Diagram, Laboratory Prototype, Frequency Response.

Description: In this work the analytic study, the simulation analysis and the experimental validation for a calculation procedure of the frequency response in dynamical systems, employing algorithms for fast computation of Fourier transforms, is addressed. In particular, the conceptual relationship between the Fourier transform and the Laplace transform of signals is discussed, followed by its equivalence with the concept of frequency response of systems based on Bode diagrams. In accordance, simulation experiments were conducted to verify that theoretical predictions proposed by the `bode(.)` function of the MATLAB software agree with calculations performed on the quotient of fast Fourier transforms of stimulus and response signals of a given system. Moreover, an experimental prototype was configured in laboratory to validate the simulation results, composed of a simple passive RC circuit subjected to a persistent stimulus of spread bandwidth and corresponding analysis of its input / output relationship by online calculations of fast Fourier transform performed on a *Raspberry Pi 3 model B+* development system. The results obtained experimentally confirmed the validity of theoretical predictions, suggesting the applicability of the proposed method to characterize the frequency response of more complex practical systems with cumbersome modeling tasks, constituting the main topic for future and complimentary works.

* Bachelor Thesis

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y telecomunicaciones. Director: Ricardo Alzate Castaño, Doctorado en Ingeniería Informática y Automática.

Introducción

El *análisis de la respuesta frecuencial* constituye la base de la *teoría del control clásico* (OGATA, 2010). En particular, toda la *teoría de estabilidad* para sistemas *SISO-LTI* está construida a partir del teorema de estabilidad de *Nyquist* y los conceptos de *margen de fase* y *margen de ganancia* (KEEL and BHATTACHARYYA, 2008).

Un *diagrama de Bode* permite verificar de manera sencilla las características anteriormente mencionadas, siendo su concepción relativamente simple desde un punto de vista teórico. En particular, un *diagrama de Bode* se compone de dos gráficas (ver Fig. 1), una de las cuales indica la ganancia en decibeles para la relación entrada-salida del sistema en estado estacionario, ante un estímulo senoidal, punto a punto en un rango de frecuencias de interés; mientras que la otra realiza lo propio para la diferencia de fases en grados sexagesimales entre el estímulo y la respuesta del sistema (OGATA, 2010).

Sin embargo, inyectar estímulos senoidales individuales, en múltiples frecuencias, para construir puntos de un diagrama en estado estacionario, se hace *impráctico* e implica conocimiento *a priori* de las características dinámicas del sistema y/o de su modelo (al menos para definir el rango de frecuencias de interés (KEEL and BHATTACHARYYA, 2008)). Por tanto, el cálculo para respuestas frecuenciales de sistemas reales hace uso de métodos de procesamiento de señales que aprovechan la descomposición espectral de *Fourier*. La Fig 2 ilustra a manera de ejemplo los

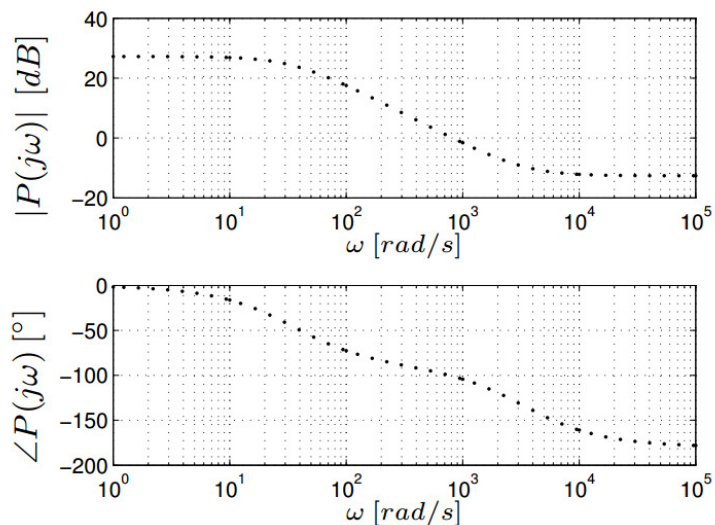


Figura 1. Representación para el diagrama de Bode de un sistema

elementos que interactúan en un análisis experimental de vibraciones mecánicas, basado en la respuesta frecuencial.

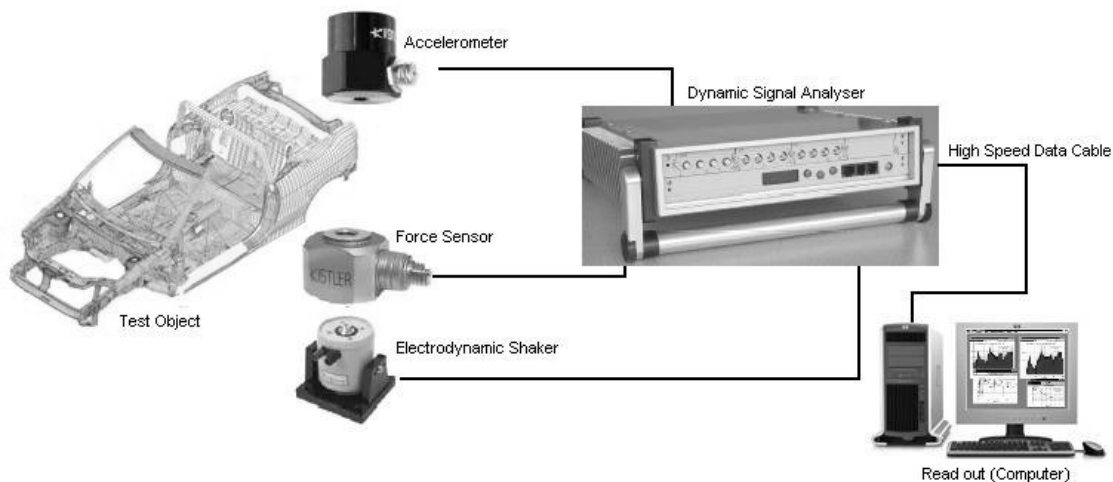


Figura 2. Análisis experimental para una respuesta frecuencial (tomado de: www.wikipedia.org)

De la teoría de señales se sabe que cualquier señal (periódica o aperiódica de periodo infi-

nito) puede descomponerse en una serie ponderada de funciones seno (entendiendo que la función coseno es una función seno desplazada) (OPPENHEIM et al., 1998). Por tanto, una señal con una cantidad suficiente de componentes espectrales puede ser empleada como estímulo para excitar la mayor cantidad de dinámicas en un procedimiento de identificación para la respuesta frecuencial de un sistema. A través de ello, pueden aproximarse representaciones matemáticas (como la función de transferencia) y valores de parámetros, basándose exclusivamente en datos experimentales (LJUNG, 1999).

Ahora bien, algoritmos de cálculo rápido (como la transformada rápida de Fourier, FFT: *fast Fourier transform* (MITRA, 2010)) permiten determinar en modo eficiente la descomposición espectral de señales a través de dispositivos electrónicos configurados para realizar tareas dedicadas de cálculo computacional. A pesar de ello, no existe en la literatura mayor documentación respecto a la relación existente entre la descomposición espectral de una señal y el diagrama de Bode del sistema que la genera.

Una aproximación trivial llevaría a pensar que el diagrama de Bode debería relacionar el cociente entre la transformada rápida de Fourier de la respuesta del sistema respecto a la transformada rápida de Fourier de su estímulo. Sin embargo existen consideraciones de escala que no permiten total certeza de lo anterior (WANG et al., 1997). Algunos desarrollos introducen el concepto de chirrido (*chirp* en inglés), como una señal experimental con amplias componentes frecuenciales que facilita el análisis frecuencial (SAUPE and KNOBLACH, 2015; CHAN, 2010). En sistemas

de control es popular el método denominado realimentación de relé (relay feedback) (WANG et al., 1997, 1999), para obtener la función de transferencia de la planta a partir de la respuesta frecuencial del sistema realimentado. En (MARCHETTI et al., 2001) el método se extiende en lazo abierto a sistemas inestables.

A partir de lo anterior es posible formular las siguientes inquietudes: ¿Es posible calcular en tiempo real el diagrama de Bode de un sistema? ¿Cómo emplear la transformada rápida de Fourier para aproximar en línea la respuesta frecuencial de un sistema? ¿Qué consideraciones experimentales deben tomarse en cuenta al momento de identificar la respuesta frecuencial de un sistema en laboratorio?

El presente trabajo de grado busca realizar aportes en la temática, permitiendo abordar y resolver dichas inquietudes.

1. Objetivos

Objetivo general

Desarrollar un procedimiento para determinar la respuesta frecuencial de un sistema dinámico a partir del cálculo experimental de su transformada rápida de Fourier.

Objetivos específicos

Analizar la relación conceptual entre la transformada de Fourier de una señal y el diagrama de Bode de un sistema SISO-LTI;

Determinar mediante un experimento de simulación la respuesta frecuencial de un sistema SISO-LTI empleando como algoritmo de cálculo a la transformada rápida de Fourier sobre sus señales de entrada y salida;

Verificar experimentalmente el procedimiento anterior aplicando la técnica sobre un circuito eléctrico pasivo y empleando un dispositivo dedicado para cálculo de la respuesta frecuencial del sistema en tiempo real.

2. Transformada de Fourier de un sistema

La respuesta frecuencial de un sistema dinámico lineal, suele asociarse con la relación logarítmica ante un estímulo senoidal conocida como el *estado senoidal permanente* y definida formalmente mediante un *diagrama de Bode*. Sin embargo, no es clara la relación existente entre una respuesta frecuencial y la descomposición espectral para las señales de entrada y salida de un sistema, en términos de sus correspondientes transformadas de Fourier.

El presente Capítulo aborda dicha temática desde una perspectiva teórica conceptual, que permitirá posteriormente justificar una técnica experimental para el cálculo de diagramas de Bode.

2.1. Transformada de Fourier

La transformada de Fourier, es una transformación matemática que mapea una señal desde el dominio del tiempo continuo hacia el dominio de la frecuencia compleja (OPPENHEIM et al., 1998). Dicha transformación se utiliza para extraer y analizar información espectral, útil en diferentes aplicaciones de la teoría de la información, como el procesamiento de señales y las telecomunicaciones.

Matemáticamente, siendo x una señal del tiempo continuo t , su transformada de Fourier se simboliza y define en el modo siguiente (OPPENHEIM et al., 1998):

$$\begin{aligned} X(j\omega) &= \mathcal{F}\{x(t)\} \\ &= \int_{-\infty}^{+\infty} x(t)e^{-j\omega t} dt, \end{aligned}$$

siendo $\omega = 2\pi f$ la frecuencia angular en *rad/s*.

A continuación, se realizará una discusión temática respecto a la interpretación del anterior resultado desde diferentes puntos de vista.

2.1.1. Relación entre las transformadas de Laplace y de Fourier. Otro de los resultados importantes en matemáticas de variable compleja es la *transformada de Laplace* (OPENHEIM et al., 1998). Esta herramienta es ampliamente empleada como instrumento para reducir la complejidad de las ecuaciones integro-diferenciales lineales, a representaciones algebraicas equivalentes.

En el contexto de las ingenierías eléctrica y electrónica, la transformada de Laplace permite definir el dominio de la frecuencia compleja:

$$s = \sigma + j\omega,$$

y a través de ello explicar diferentes propiedades de los circuitos y las señales eléctricas.

Matemáticamente, la transformada (bilateral) de Laplace para una señal x del tiempo continuo t , se define como (OPPENHEIM et al., 1998):

$$\begin{aligned} X(s) &= \mathcal{L}\{x(t)\} \\ &= \int_{-\infty}^{+\infty} e^{-st} x(t) dt. \end{aligned} \quad (1)$$

Se observa por tanto la manera cómo la transformada de Fourier de una señal $x(t)$ corresponde por definición con su transformada de Laplace para $s = j\omega$; es decir (OPPENHEIM et al., 1998):

$$\mathcal{F}\{x(t)\} \equiv \mathcal{L}\{x(t)\}|_{\sigma=0}.$$

Desde un punto de vista físico, la condición $\sigma = 0$ implica una respuesta sin amortiguamiento, o dicho de otra forma, en *estado senoidal permanente*.

Así entonces, otra forma de enunciar la equivalencia anterior es diciendo que la transformada de Fourier de una señal corresponde con su transformada de Laplace en el estado senoidal permanente, lo cual cobra sentido si la señal en mención es la respuesta de un sistema ante un estímulo sinusoidal, constituyendo la esencia de un diagrama de Bode.

2.1.2. Relación entre la transformada de Fourier y el diagrama de Bode. Las transformadas de Laplace y Fourier son conceptos exclusivos de señales. De otro lado, el diagrama de Bode se aplica sobre sistemas.

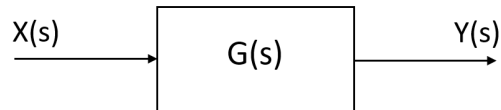


Figura 3. Relación gráfica entre señales y sistemas

Por tanto, haciendo hincapié en la relación conceptual entre señales y sistemas (OPPENHEIM et al., 1998), es claro que no tiene mucho sentido hablar de la transformada de Fourier de un sistema. Sin embargo, es posible relacionar lo anterior con la equivalencia entre \mathcal{F} y \mathcal{L} , si se recuerda que para un sistema como el ilustrado en la Fig.3, el diagrama de Bode se determina a partir de los gráficos de magnitud y fase como función de la frecuencia en el estado senoidal

permanente; es decir (OGATA, 2010):

$$G(s) = \frac{Y(s)}{X(s)};$$

$$\begin{aligned} G(j\omega) &= \frac{Y(j\omega)}{X(j\omega)}, \\ &= Me^{j\theta}; \end{aligned}$$

$$M = |G(j\omega)|,$$

$$\theta = \angle G(j\omega).$$

Así entonces, es posible escribir:

$$\begin{aligned} G(j\omega) &\equiv \mathcal{F}\{g(t)\} \\ &= \frac{\mathcal{F}\{y(t)\}}{\mathcal{F}\{x(t)\}}, \end{aligned} \tag{2}$$

y con base en ello:

$$\begin{aligned} M &= \frac{|\mathcal{F}\{y(t)\}|}{|\mathcal{F}\{x(t)\}|}, \\ \theta &= \angle \mathcal{F}\{y(t)\} - \angle \mathcal{F}\{x(t)\}, \end{aligned}$$

permitiendo relacionar la transformada de Fourier de una señal con el diagrama de Bode de un

sistema.

2.1.2.1. Ejemplo 1. Considere el circuito RC de la Fig. 4. La función de transferencia para la relación de voltajes viene dada como:

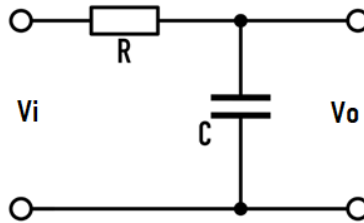


Figura 4. Circuito RC

$$\begin{aligned}
 G(s) &= \frac{V_o}{V_i} \\
 &= \frac{1}{RCs + 1} \\
 &= \frac{1/RC}{s + (1/RC)}.
 \end{aligned} \tag{3}$$

Asumiendo $v_i(t) = \delta(t)$, el voltaje de salida $v_o(t)$ corresponderá con la respuesta al impulso del sistema, misma que puede ser calculada en el modo siguiente:

$$\begin{aligned}
 v_o(t) &= \mathcal{L}^{-1}\{V_i(s)G(s)\} \\
 &= \mathcal{L}^{-1}\{G(s)\} \\
 &= \left(\frac{1}{RC}\right) e^{-(\frac{1}{RC})t}.
 \end{aligned}$$

Ahora bien, utilizando la expresión (2) es posible determinar la respuesta frecuencial del circuito RC en el modo siguiente:

$$\begin{aligned}
 G(j\omega) &\equiv \frac{\mathcal{F}\{v_o(t)\}}{\mathcal{F}\{v_i(t)\}} \\
 &= \frac{\mathcal{F}\left\{\left(\frac{1}{RC}\right)e^{-\left(\frac{1}{RC}\right)t}\right\}}{\mathcal{F}\{\delta(t)\}} \\
 &= \frac{\left(\frac{1}{RC}\right)\left(\frac{1}{j\omega + \frac{1}{RC}}\right)}{1} \\
 &= \frac{1/RC}{j\omega + (1/RC)}, \tag{4}
 \end{aligned}$$

lo cual coincide con reemplazar s por $j\omega$ en (3).

Asimismo, considerando una entrada $v_i(t)$ arbitraria, el cálculo para $v_o(t)$ implica obtener una expresión que preserve el resultado dado en (4), precisamente porque la respuesta frecuencial de un sistema es independiente de las señales que se le estén aplicando.

La pregunta que surge en este punto es: ¿Cómo aplicar la equivalencia dada por (2) cuando no se dispone de un modelo analítico del sistema?

Como respuesta a dicho interrogante, se presentan los métodos de cálculo algorítmico para la transformada de Fourier, a partir de datos.

2.2. Cálculo computacional para la transformada de Fourier

La definición dada en (1) para la transformada de Fourier de una señal, implica la resolución de una integral en un intervalo infinito de valores del tiempo continuo.

Lo anterior representa una formulación analítica que puede ser aproximada en la práctica al procesamiento de una cantidad finita de datos a partir de una cuantización de la señal $x(t)$ en muestras discretas en el tiempo.

A partir de ello, la transformada de Fourier puede redefinirse en el modo siguiente:

$$\begin{aligned} X(j\omega) &= \int_{-\infty}^{+\infty} x[nT]e^{-j\omega t} dt \\ &\sim \sum_{n=-\infty}^{\infty} x[nT]e^{-j\omega nT}, \end{aligned} \quad (5)$$

Siendo $x[nT]$ la secuencia discretizada para $x(t)$ con intervalo de muestreo T .

Hay que tener en cuenta que esta igualdad sólo será posible si T tiende a cero y que $x[nT]$ cumpla el criterio de Nyquist, de modo que la señal no pierda información importante.

A la expresión (5) se le conoce como la *transformada de Fourier de tiempo discreto* (TFTD). Es interesante notar que dicho resultado asume un codominio continuo (rango continuo de frecuencias), a pesar de aplicarse a señales en el tiempo discreto.

2.2.1. TFTD para secuencias finitas. Desde un punto de vista computacional (i.e. práctico), el cálculo de una TFTD implica secuencias del tiempo discreto de longitud finita y por tanto, es posible asociar su determinación a un conjunto finito de frecuencias (MITRA, 2010).

A partir de ello considere la secuencia $x[nT]$ definida para $0 \leq n \leq (N - 1)$, siendo N un número entero positivo que representa la longitud (número de muestras) de la secuencia. En estos términos, la TFTD en (5) puede reescribirse como:

$$X(j\omega) = \sum_{n=0}^{N-1} x[nT]e^{-j\omega nT}.$$

Más aún, asumiendo un muestreo uniforme en el dominio de la frecuencia, de manera tal que:

$$\begin{aligned} X[k] &\approx X(j\omega)|_{\omega=\omega_k} \\ &= \sum_{n=0}^{N-1} x[nT]e^{-j\omega_k nT}, \end{aligned} \quad (6)$$

siendo $\omega_k \in [0, 2\pi] = \frac{2\pi}{N}k \quad \forall \quad k = 0, 1, 2, \dots, (N - 1)$, es posible asociar una secuencia $X[k]$ de N elementos en el dominio de la frecuencia a una secuencia $x[nT]$ de N elementos en el dominio del tiempo.

A la expresión (6) se le denomina *Transformada Discreta de Fourier* (TDF).

Un análisis detallado para la TDF permite observar que el cálculo para cada elemento de $X[k]$ implica N productos complejos y $(N - 1)$ sumas complejas. De esta manera, la totalidad de la secuencia implica N^2 productos complejos que representan un costo computacional importante para resoluciones de escala (cantidad de puntos) que satisfagan ampliamente el *Teorema del Muestreo*.

Así entonces, para alivianar la carga computacional implicada en el cálculo de la TDF se explota la periodicidad y simetría del término:

$$\begin{aligned}
 e^{-j\omega_k nT} &= \cos(\omega_k nT) - j \sin(\omega_k nT) \\
 &= \cos\left(\frac{2\pi}{N} knT\right) - j \sin\left(\frac{2\pi}{N} knT\right) \\
 &= W_N^{kn},
 \end{aligned} \tag{7}$$

Donde $W_N = e^{-j\frac{2\pi}{N}}$.

Ahora la TDF puede, alternativamente, escribirse como:

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[nT]W_N^{kn} \\ &= x[0]W_N^0 + x[T]W_N^k + x[2T]W_N^{2k} + \cdots + x[(N-1)T]W_N^{(N-1)k}. \end{aligned} \quad (8)$$

2.2.2. Cálculo rápido para la TDF. Asumiendo la relación: $z = W_N^{-k}$, la expresión (8) corresponde con la *Transformada z Unilateral* para $x[n]$, brindando todas las ventajas estructurales derivadas de la teoría de implementación (realizaciones) para filtros digitales (OPPENHEIM et al., 1998).

Inicialmente, considere la descomposición de una secuencia $x[n]$ en dos partes: x_0 (muestras pares) y x_1 (muestras impares). La transformada z de $x[n]$ puede expresarse en el modo siguiente:

$$\begin{aligned} X(z) &= X_0(z) + z^{-1}X_1(z) \\ &= X[k] \\ &= X_0[k] + W_N^k X_1[k], \end{aligned}$$

según ilustrado en la Fig. 5(a). A este método se le denomina *decimación en el tiempo* y facilita el cálculo de muestras en paralelo a partir de la generalización para secuencias de $N = 2^m$ muestras, como se muestra en la Fig. 5(b) para el caso de $N = 8$. Observe el caracter circular del algoritmo

propuesto, en el cual:

$$X[0] = X_0[0] + W_N^0 X_1[0]$$

$$X[1] = X_0[1] + W_N^1 X_1[1]$$

$$X[2] = X_0[2] + W_N^2 X_1[2]$$

$$X[3] = X_0[3] + W_N^3 X_1[3]$$

$$X[4] = X_0[0] + W_N^4 X_1[0]$$

$$X[5] = X_0[1] + W_N^5 X_1[1]$$

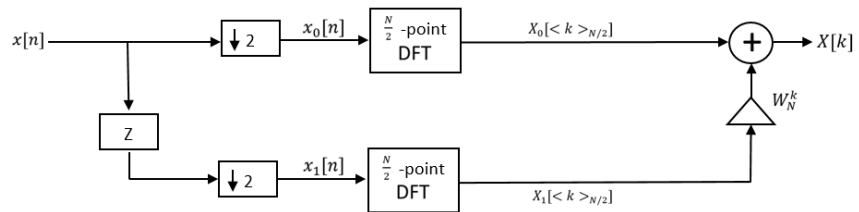
$$X[6] = X_0[2] + W_N^6 X_1[2]$$

$$X[7] = X_0[3] + W_N^7 X_1[3],$$

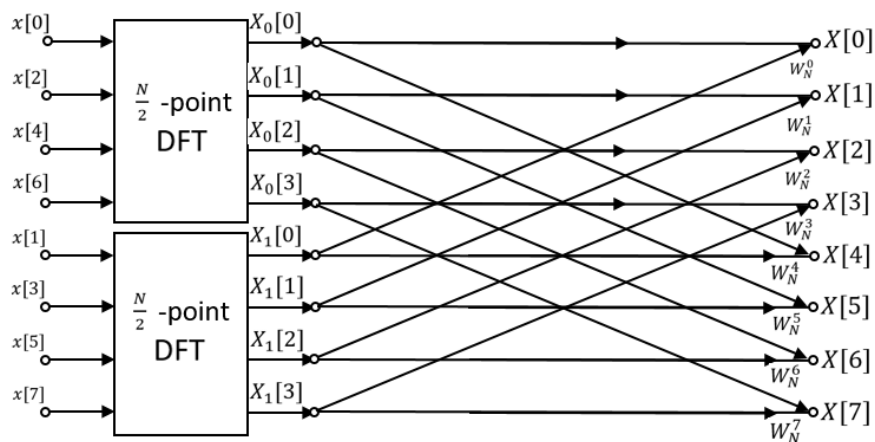
repetiendo la secuencia para $k = N/2$. La forma del algoritmo justifica su nombre de *cálculo ma-riposa*. La misma idea puede emplearse para aplicar subdivisiones adicionales, configurando un método denominado *radix-2*. Se recomienda al lector interesado consultar la Sección 8.3 de (MITRA, 2010), para una discusión más detallada de este y otros métodos de cálculo rápido para la TDF (FFT - *fast Fourier transform*).

2.2.2.1. Ejemplo 2. Considere la función del tiempo continuo:

$$x(t) = te^{-t}u(t). \quad (9)$$



(a)



(b)

Figura 5. Decimación en el tiempo para TDF (adaptado de (MITRA, 2010)): (a) descomposición de secuencia en dos partes iguales, (b) cálculo paralelo para 8 muestras

Acudiendo a una tabla de transformadas o bien desarrollando cálculos analíticos, es posible obtener su correspondiente transformada de Fourier:

$$\mathcal{F}\{x(t)\} = \frac{1}{(1 + j\omega)^2}. \quad (10)$$

Asimismo, el algoritmo *radix-2* para el cálculo rápido de la transformada viene implementado en MATLAB a través de la función: $X = \text{fft}(x, N)$, siendo X la secuencia de N puntos correspondiente con la transformada discreta de Fourier de la secuencia x . De esta manera, ejecutando el comando anterior ante una versión discreta de (9) constituida de 51 puntos, para $t = 10\text{ s}$ y $T = 0.2\text{ s}$ (suficientes para preservar la información relevante de la señal en el tiempo), la cantidad N de muestras para la secuencia de salida $X[k]$ se determina de forma empírica, es decir, a partir de la contraposición de las gráficas de la TFTD y la FFT, se varía los valores de N y se encuentra que, con una resolución de 32 muestras como mínimo, se garantiza que la transformada rápida de Fourier (FFT) y la transformada de Fourier de tiempo discreto (TFTD) son similares.

La Fig. 6 muestra los resultados comparativos (de magnitud) para la transformada de Fourier analítica dada en la expresión (10), evaluada para un rango de valores de ω y su correspondiente cálculo en MATLAB empleando la función mencionada. Como se observa, el método de cálculo computacional es bastante acertado en la aproximación para el espectro de la señal (9). Es importante mencionar que para efectos de escala, el algoritmo $\text{fft}(\cdot)$ de MATLAB requiere

consideraciones especiales (ajustes) en los ejes de frecuencia y amplitud para coincidir cuantitativamente con las predicciones analíticas de referencia. Se recomienda al lector interesado consultar estos detalles en la Sección 8.3 de (MITRA, 2010).

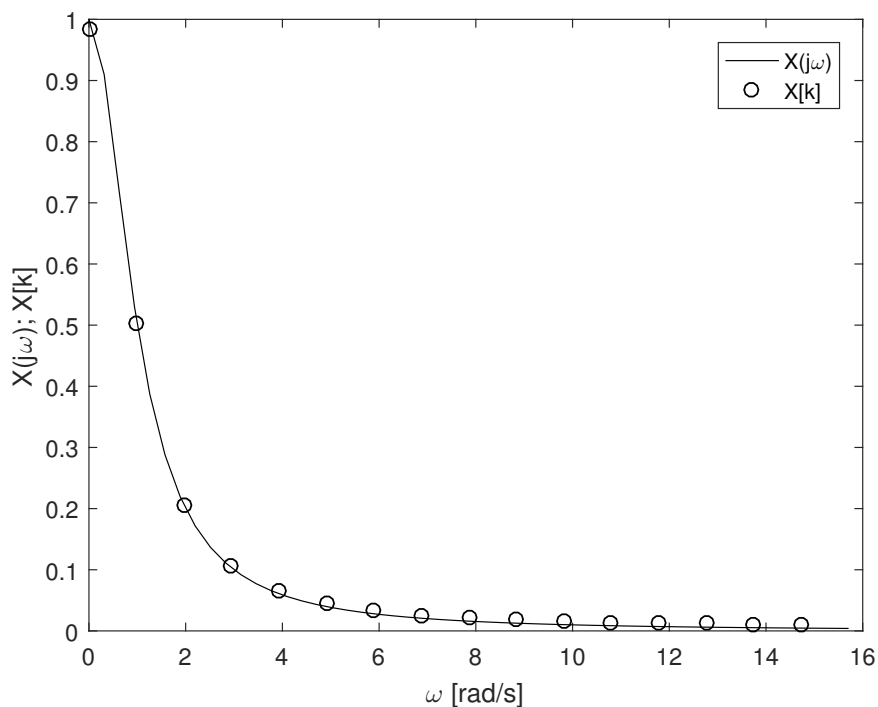


Figura 6. Comparación gráfica entre TFTD y algoritmo FFT validando el uso de $N=32$ muestras.

2.2.2.2. Ejemplo 3. La respuesta frecuencial para el circuito del Ejemplo 1, puede calcularse a partir de información obtenida de los datos de sus señales de estímulo y respuesta.

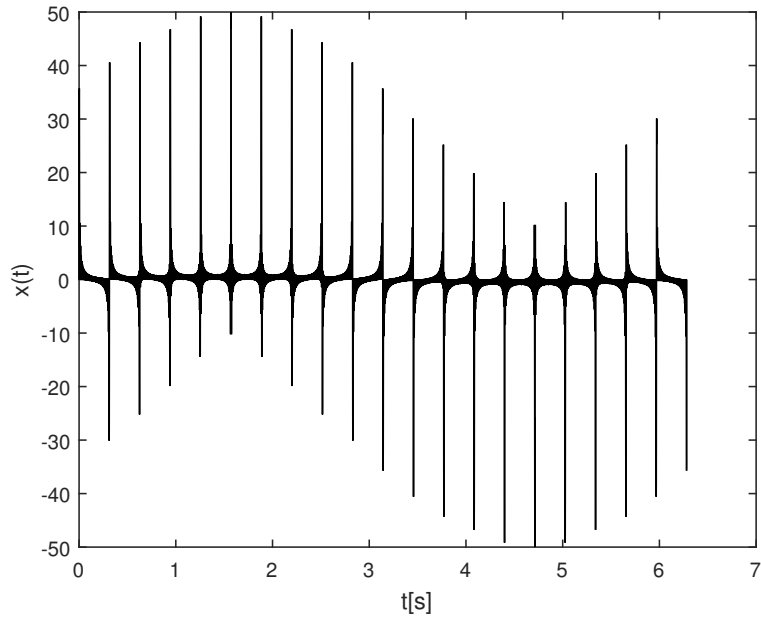
Precisamente, se requiere de una excitación suficiente de componentes frecuenciales a través de la aplicación de un estímulo de *excitación persistente* (LJUNG, 1999), que puede aproximarse mediante la superposición de señales senoidales en un rango amplio de frecuencias.

De esta manera, siendo $\omega = \{\omega_1, \omega_2, \dots, \omega_L\}$ para $\omega_k = k \times \left(\frac{\omega_L}{L}\right) \forall k = 2, 3, \dots, (L-1)$, y valores extremos $\omega_1 = \omega_{min}$ y $\omega_L = 10 \times \frac{1}{RC}$ (considerando el criterio de la máxima frecuencia de esquina (OGATA, 2010)), es posible constituir un vector de frecuencias para excitar el sistema asumiendo $R = 10 k\Omega$, $C = 1 \mu F$, $L = 50$ y $\omega_1 = 1 rad/s$. A partir de ello, la Fig. 7 ilustra el estímulo aplicado como entrada al circuito, con su correspondiente respuesta temporal a través de simulación numérica en MATLAB.

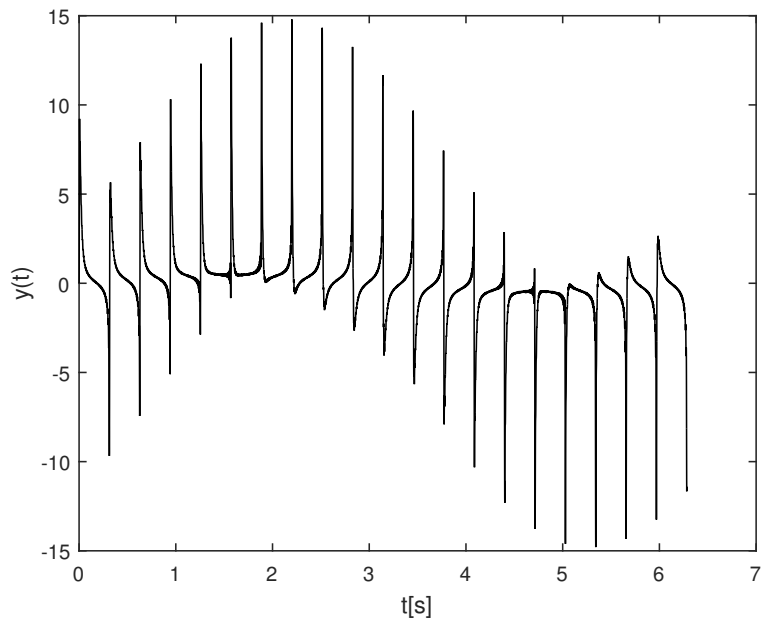
Por tanto, empleando la técnica de cálculo computacional ilustrada en el Ejemplo 2, para $N = 256$ fue posible obtener las correspondientes transformadas de Fourier para las señales de entrada y salida del circuito presentadas en la Fig. 8.

Sobre la base de los conceptos presentados en la Sección 2.1.2 para relación entre la transformada de Fourier de una señal y el diagrama de Bode de un sistema, se construye la respuesta frecuencial mostrada en la Fig. 9, donde la línea sólida corresponde con las predicciones teóricas calculadas sobre el modelo del circuito empleando la función `bode(.)` de MATLAB.

Como se observa, la predicción respecto a la forma de la componente de magnitud de la respuesta frecuencial del sistema obtenida a partir del análisis de datos mediante la FFT, es bastante acertada respecto a las predicciones teóricas, sugiriendo un interesante potencial para desarrollar cálculo rápido de diagramas de Bode en sistemas desconocidos, utilizando métodos experimenta-



(a)



(b)

Figura 7. Excitación persistente y respuesta temporal del circuito: (a) estímulo, (b) respuesta

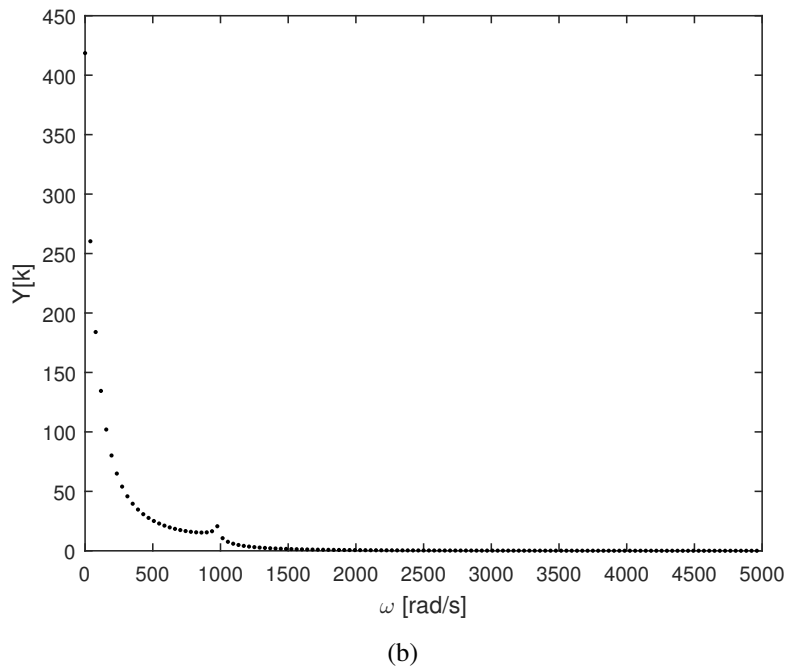
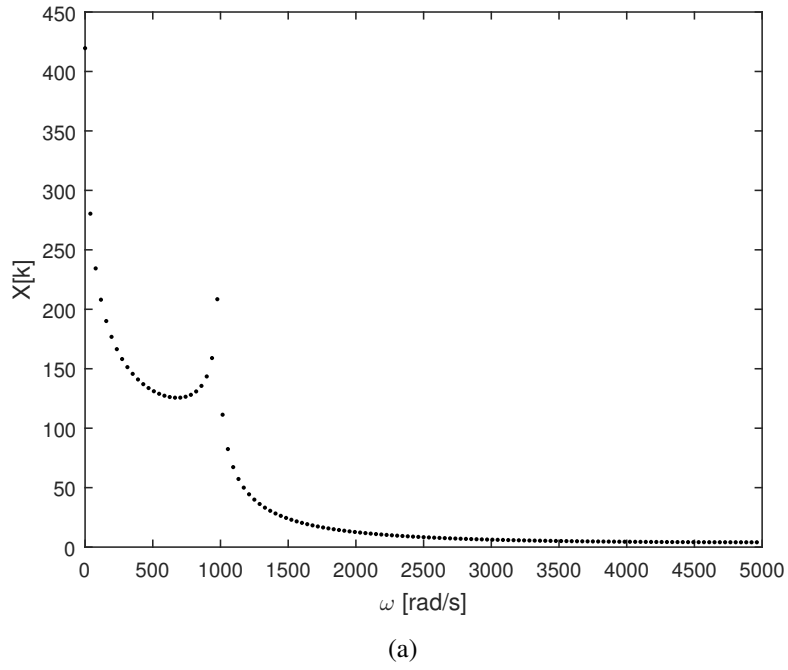


Figura 8. Espectros de Fourier mediante FFT para: (a) estímulo, (b) respuesta

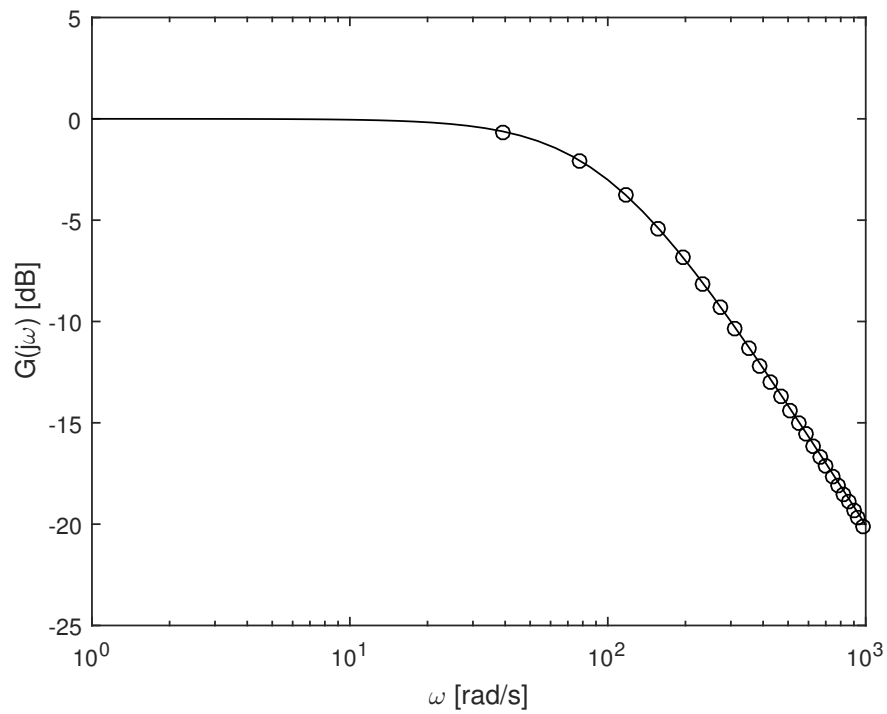


Figura 9. Diagramas de Bode teórico (trazo continuo) y numérico experimental (círculos)

les, según será presentado en el siguiente *Capítulo*.

3. Diagrama de Bode experimental

En complemento a los desarrollos analíticos y de simulación anteriormente presentados, el presente *Capítulo* aborda la implementación de un experimento de laboratorio para realizar el cálculo en tiempo real para el diagrama de Bode de un circuito eléctrico, empleando algoritmos de cálculo rápido para la transformada de Fourier ejecutados en un sistema embebido.

3.1. Diseño del experimento

Tomando como base el Ejemplo 3 del *Capítulo 2*, se configuró en laboratorio el esquema presentado en el diagrama de bloques de la Fig. 10, cuyas etapas constitutivas se describen a continuación.

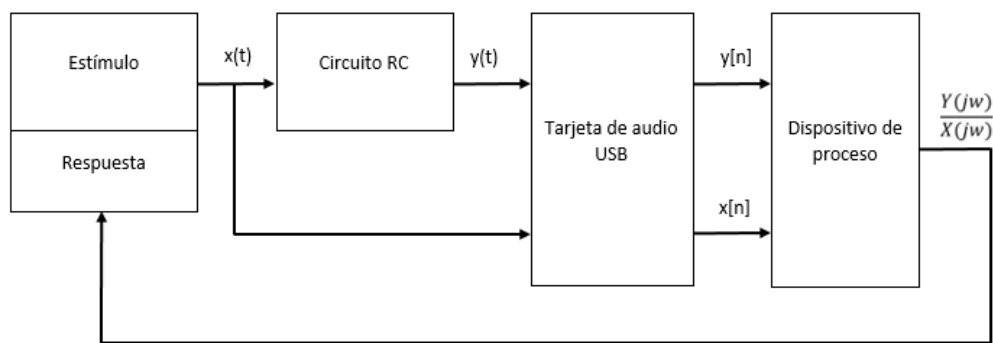


Figura 10. Diagrama de bloques para sistema experimental

3.1.1. Circuito RC. El esquema RC ilustrado en la Fig. 4 fue implementado en una tarjeta de circuito impreso, según presentado en la Fig. 11. Como parte de esta adecuación, se emplearon borneras y terminales que permitieran acoplar el dispositivo a la señal aplicada como entrada y al dispositivo que recibirá su correspondiente respuesta de salida.

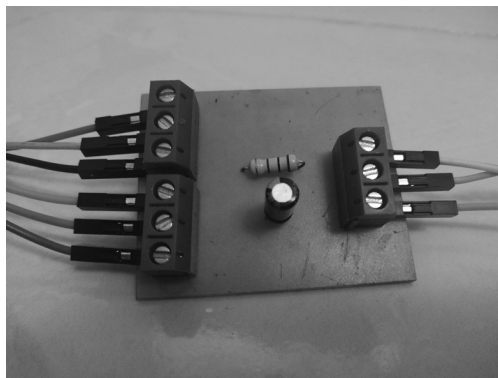


Figura 11. Circuito RC implementado

3.1.2. Estímulo. Para crear el estímulo del Circuito RC, se hace uso de la señal de *excitación persistente* descrita en el *Ejemplo 3*. De esta manera, empleando la función `audiowrite(.)` de MATLAB, es posible construir un archivo de audio de extensión `*.wav` que posteriormente se reproduce como señal analógica a través de la tarjeta de audio de un PC, actuando como generador de señales.

Las características de la función `audiowrite(.)` de MATLAB, implican la selección de una frecuencia de muestreo para la señal a ser creada a manera de archivo de audio. Esta frecuencia de muestreo fue asignada como $f_s = 44100 \text{ Hz}$, teniendo en cuenta que es la misma frecuencia a la cual se comunica la tarjeta de audio USB con el dispositivo de proceso, según explicado a continuación.

3.1.3. Tarjeta de audio USB. El dispositivo de proceso de señales elegido no posee entradas analógicas. Por tanto, se requiere de una interfaz que permita digitalizar la señal de respuesta $y(t)$ del Circuito RC, para poder ser procesada de manera acorde.

Para ello se hizo uso de una tarjeta de audio USB de marca 3D Sound como la mostrada en la Fig.12. Básicamente, dicho dispositivo se encuentra compuesto por un puerto USB, convertidores digital/analógico (para salida de audífonos) y analógico/digital (para entrada de micrófono), además de elementos de amplificación.

Tomando en cuenta que también se requiere digitalizar el estímulo $x(t)$ aplicado al Circuito RC para poder realizar su correspondiente procesamiento, y que para efectos prácticos sólo se empleará la entrada de micrófono de la tarjeta, es necesario emplear dos tarjetas de audio USB (una para $x(t)$ produciendo $x[n]$ y otra para $y(t)$ produciendo $y[n]$).



Figura 12. Tarjeta de audio USB empleada en el experimento

3.1.4. Dispositivo de proceso. Como dispositivo de proceso de datos se eligió la tarjeta de desarrollo *Raspberry Pi 3 modelo B+*, ilustrada en la Fig. 13. Entre las razones que motivaron la elección del dispositivo se encuentran su facilidad de programación y la amplia disponibilidad de librerías, hardware e información técnica.

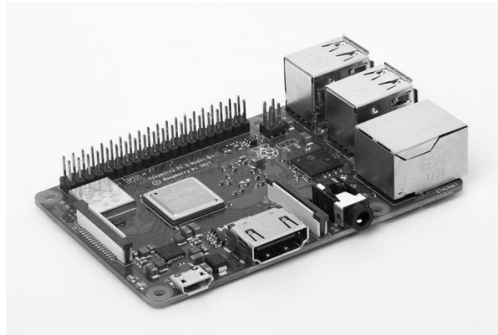


Figura 13. Raspberry Pi 3 modelo B+

En palabras simples, el *Raspberry Pi 3 modelo B+* es un minicomputador ejecutado a través del sistema operativo *Raspbian*, como versión de *Linux-Debian* para esta clase de dispositivos. Características técnicas y detalles adicionales pueden consultarse en la página web del fabricante (<https://www.raspberrypi.org>).

A continuación se discutirá brevemente el procedimiento realizado para ejecutar el cálculo de la FFT y la visualización de resultados para la respuesta frecuencial del Circuito RC.

3.1.4.1. Configuración del dispositivo. Inicialmente, en el código del programa se definen las librerías a ser empleadas durante el procesamiento. En particular, se requieren: 1) *pyaudio* para captura de señal de tarjetas de audio; 2) *wave* para convertir datos de tarjetas de audio en valores de representación numérica; 3) *numpy* para manejo matricial de datos; 4) *math* para acceso a funciones matemáticas y 5) *pyp1ot* para realizar gráficos en tiempo de ejecución.

Posteriormente, se definen valores de parámetros importantes en la configuración del sis-

tema destacándose la tasa de adquisición de datos de la tarjeta de audio USB ($f_s = 44100$) y el tamaño de trama de datos en el puerto de lectura ($N = 256$).

Por último, la rutina de proceso cae en un ciclo infinito que realiza de manera secuencial el llamado a las subrutinas descritas a continuación.

3.1.4.2. Descripción de subrutina para lectura de tarjeta de audio. El primer paso del proceso consiste en adquirir los datos provenientes de las tarjetas de audio USB. Para ello se emplea la librería `pyaudio`, misma que deberá ejecutarse sobre *Python 3*. A su vez, es necesario inicializar objetos ($p1$ y $p2$ para señales de entrada y salida al circuito RC) y crear variables con características de datos a ser adquiridos (números enteros de 16 bits), canales (lectura entrada de micrófono), tasa de muestreo (f_s), número de puerto detectado para el dispositivo (1 para $p1$ y 2 para $p2$) y la cantidad de datos por paquete de captura (N).

3.1.4.3. Descripción de subrutina para ajuste de formato de datos. Los datos adquiridos de puerto, deben ser ajustados a un formato numérico compatible con operaciones aritméticas y lógicas a través de la librería `wave` empleando comandos como: `wave.struct.unpack()`, que se encarga de extraer la información contenida en la trama de datos adquirida. Posteriormente, se hace uso de la librería `numpy` para construir y procesar vectores de datos almacenados.

3.1.4.4. Descripción de subrutina para cálculo de la FFT. La librería `numpy` posee una función especial para realizar el cálculo de la FFT. De esta manera, se aplica dicha función a los vectores de datos almacenados, ajustando posteriormente la escala de sus ejes de frecuencia y amplitud en aras de su apropiada visualización. Al respecto es importante mencionar que la

subrutina procesa la transformada en modo vectorial y por tanto el cálculo de las magnitudes en decibeles se realiza a través de un ciclo *for* para aplicar la operación logaritmo sobre los elementos del vector empleando funciones de la librería *math*. Siguiendo un procedimiento similar se calcula el cociente de las magnitudes de entrada/salida en el dominio de la frecuencia, correspondiente con la aproximación numérica del diagrama de Bode.

3.1.4.5. Descripción de librería para visualización de datos. Para visualizar el diagrama de Bode se utiliza la librería *pyplot*. Para ello se debe definir en primera instancia el tamaño de la imagen a graficar (configurada como 10 pulgadas de ancho por 8 de alto). Posteriormente, se procede a crear el objeto de figura en la ubicación deseada y realizar la visualización correspondiente de los datos empleando comandos como: *fig.add_subplot()* y *ax.plot()*. La Fig. 14 presenta una ilustración de una captura de datos para el diagrama de Bode calculado en tiempo real a partir de la interfaz desarrollada.

3.1.4.6. Flujograma para algoritmo implementado. La Fig. 15 resume las ideas anteriormente expuestas a través de un flujograma para el algoritmo de proceso de datos implementado para ejecutar el cálculo del diagrama de Bode experimental, a partir de la determinación de la FFT en el dispositivo de proceso (*Raspberry Pi 3 modelo B+*).

De otro lado, en el *Anexo 1* se incluyen las líneas de código correspondientes para el programa realizado en lenguaje Python.

3.1.5. Sistema final implementado. La apariencia física del prototipo implementado se ilustra en la Fig. 16, permitiendo identificar cada uno de los elementos que constituyen las

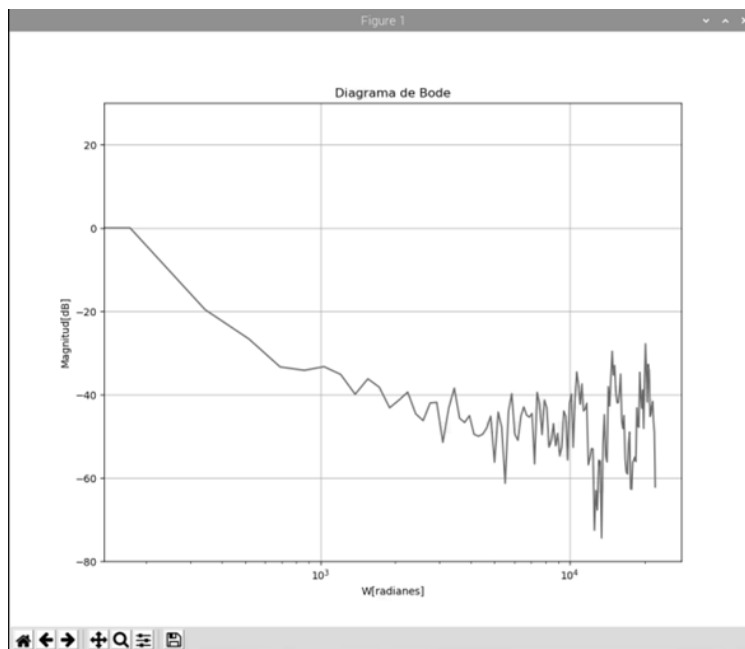


Figura 14. Captura de interfaz gráfica para datos en Raspberry

etapas del esquema funcional definido al principio del *Capítulo* en el diagrama de bloques de la Fig. 10.

3.2. Visualización de resultados

Finalmente, se observa en la Fig. 17 la superposición entre el diagrama de Bode teórico definido previamente en la Fig. 9 y aquel obtenido experimentalmente a partir del procedimiento descrito en la *Sección* anterior. Para ello, los datos del dispositivo de proceso se exportan a MATLAB a manera de archivos de texto.

Como se observa, los resultados experimentales replican de manera favorable las predicciones teóricas, confirmando la validez de los cálculos y procedimientos realizados y la respuesta frecuencial (en tiempo real) obtenida, a un costo computacional considerablemente bajo.

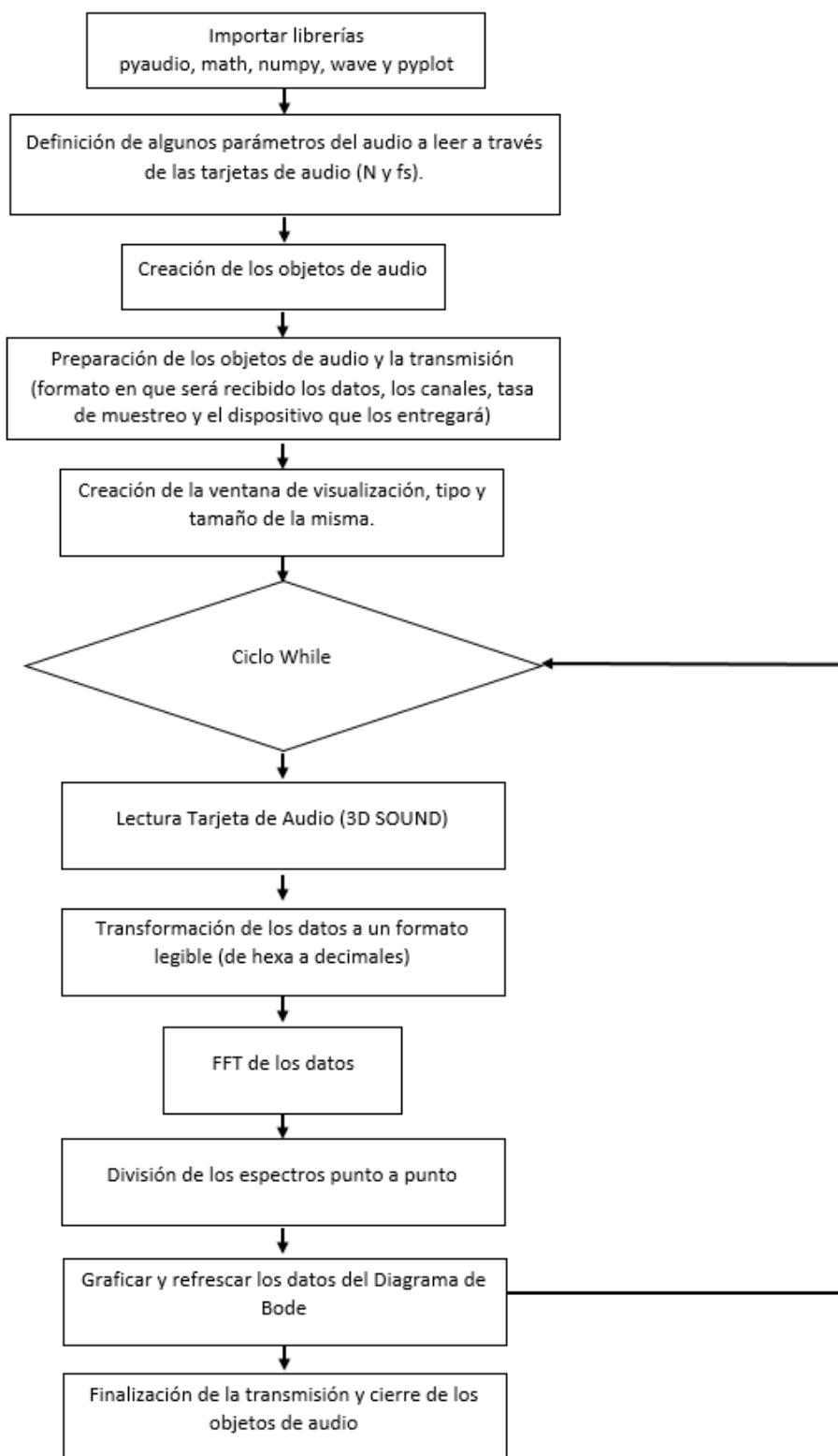


Figura 15. Flujograma para algoritmo de proceso de señal implementado

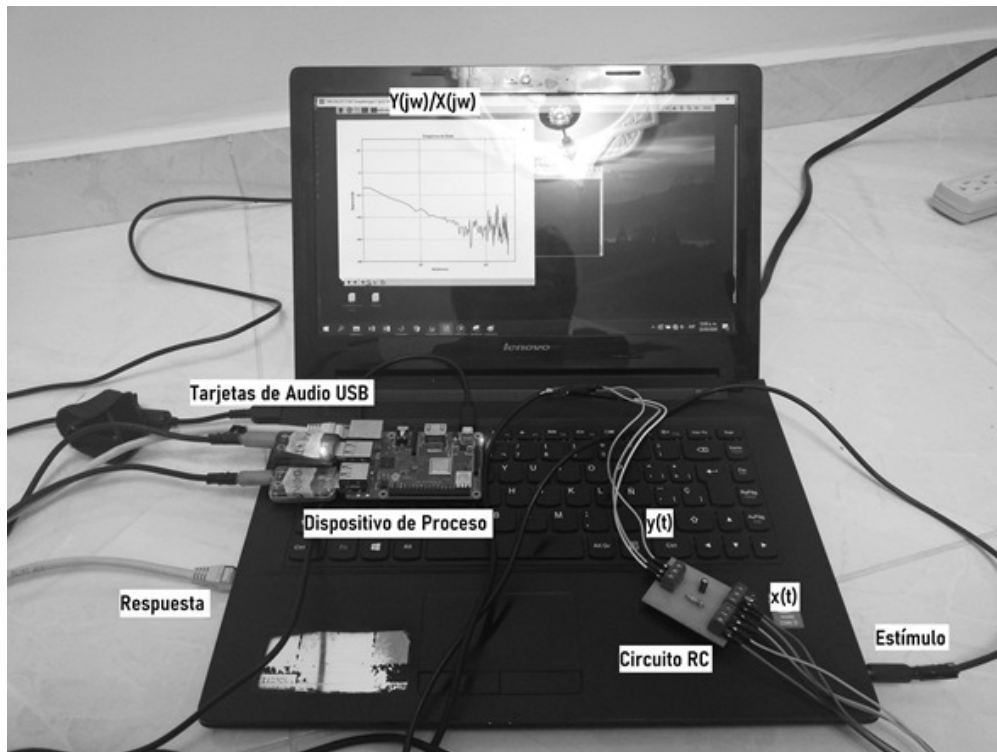


Figura 16. Sistema final implementado en laboratorio

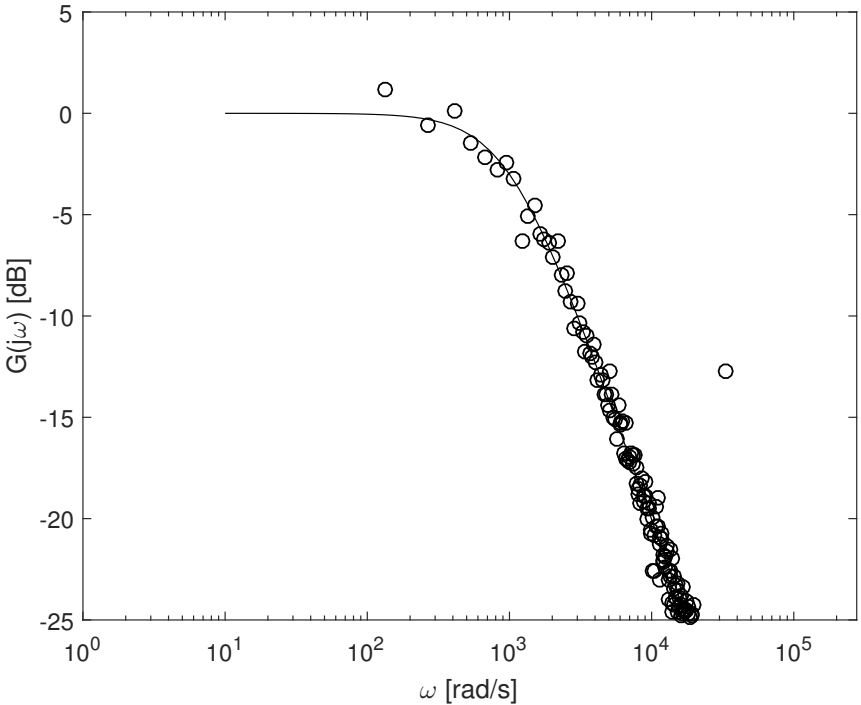


Figura 17. Superposición entre diagramas de Bode teórico (trazo continuo) y práctico experimental (círculos)

4. Recomendaciones

En aras de excitar una cantidad suficiente de dinámicas por parte de la señal de excitación persistente, se recomienda tener presente las limitaciones de ancho de banda de los dispositivos de acople de señal entre el entorno y el dispositivo de proceso (*Raspberry Pi 3 modelo B+*).

En particular, pudo observarse que las tarjetas de audio USB presentaban un comportamiento de tipo paso banda con una frecuencia de corte inferior correspondiente con la frecuencia del protocolo de comunicación USB para transferencia de datos al dispositivo de proceso ($f_s = 44100 Hz$). A partir de ello, se recomienda utilizar este valor de frecuencia de muestreo para construir los estímulos **.wav* con la función `audiowrite(.)` de MATLAB.

Es preciso recomendar que, para mejorar la gráfica de visualización del Diagrama de Bode, investigar el concepto de Periodogramas.

5. Trabajo futuro

Actividades complementarias al presente trabajo de grado pueden orientarse hacia la verificación experimental del cálculo de diagramas de Bode en sistemas de ingeniería complejos, como máquinas y circuitos electrónicos de potencia. También pueden estar enfocados en implementar tecono-

logías IoT para la visualización del Diagrama de Bode desde un sitio web.

6. Conclusiones

A partir de los desarrollos presentados y los resultados obtenidos en el presente trabajo de grado, es posible enunciar la siguiente conclusión general:

Se desarrolló un procedimiento para determinar la respuesta frecuencial de un sistema dinámico a partir del cálculo experimental de su transformada rápida de Fourier y se verificó su desempeño mediante la construcción de un prototipo de laboratorio basado en el sistema de cómputo *Raspberry Pi 3 modelo B+*, que permitió obtener en tiempo de ejecución el cálculo aproximado para el diagrama de Bode de un *circuito RC* y con ello constatar las predicciones teóricas y de simulación analizadas para el mismo sistema de ejemplo. Lo anterior permite asumir que sistemas más complejos (para los cuales no sea posible derivar de manera simple modelos analíticos) puedan ser caracterizados mediante el método propuesto.

De manera más puntual:

Se analizó la relación conceptual entre la transformada de Fourier de una señal y el diagrama de Bode de un sistema SISO-LTI. Tal y como fue abordado en la *Sección 2.1*, se discutieron las bases conceptuales para las transformadas de Laplace y de Fourier, al igual que la relación entre ellas. Asimismo, en la *Subsección 2.1.2* se abordó la equivalencia entre la transformada

de Fourier de una señal y la técnica del diagrama de Bode para la respuesta frecuencial de sistemas, ilustrando los resultados analíticos mediante un ejemplo.

Se determinó mediante un experimento de simulación la respuesta frecuencial de un sistema SISO-LTI empleando como algoritmo de cálculo a la transformada rápida de Fourier sobre sus señales de entrada y salida. En particular, la *Sección 2.2* abordó inicialmente la relación entre la transformada de Fourier y la transformada de Fourier de tiempo discreto. Posteriormente, se realizó la discusión respecto a la transformada discreta de Fourier y los algoritmos de cálculo rápido para la misma. A partir de ello se realizaron dos experimentos de simulación: en el primero de ellos (*Subsección 2.2.2.1*) se comparó el cálculo computacional para la transformada rápida de Fourier de una señal en tiempo continuo con su correspondiente solución analítica; posteriormente en un segundo ejemplo (*Subsección 2.2.2.2*) se aplicó el cociente numérico entre las transformadas rápidas de Fourier de las señales de salida y entrada de un *circuito RC* y se comparó el resultado con los vectores proporcionados por la función `bode(.)` de MATLAB. Esto último permitió verificar la equivalencia práctica entre ambos resultados.

Se verificó experimentalmente el procedimiento anterior aplicando la técnica sobre un circuito eléctrico pasivo y empleando un dispositivo dedicado para cálculo de la respuesta frecuencial del sistema en tiempo real. Más precisamente, el *Capítulo 3* fue dedicado a describir la concepción para un prototipo de laboratorio que permitió ejecutar el procedimiento estudiado para obtener una aproximación del diagrama de Bode de un sistema mediante el cálculo

computacional del cociente entre las transformadas rápidas de Fourier de sus señales de estímulo y respuesta. Para ello se construyó un circuito RC e interfases de proceso de señal a través de un sistema de cómputo *Raspberry Pi 3 modelo B+*. El sistema final implementado se ilustra en la Fig. 16. Un contraste entre las predicciones analíticas y la respuesta frecuencial experimental puede observarse en la Fig. 17. Finalmente el código en Python para el algoritmo de proceso de señal implementado se describe en el *Anexo 1*.

Referencias Bibliográficas

- CHAN, I. H. (2010). Swept Sine Chirp for Measuring Impulse. *Stanford Research Systems, Inc*, pages 1–6.
- KEEL, L. and BHATTACHARYYA, S. (2008). Controller synthesis free of analytical models: Three term controllers. *IEEE Transactions on Automatic Control*, 53(6):1353–1369.
- LJUNG, L. (1999.). *System Identification: Theory for the User*. Prentice Hall, 2nd. edition.
- MARCHETTI, G., SCALI, C., and LEWIN, D. R. (2001). Identification and control of open-loop unstable processes by relay methods. *ELSEVIER Automatica*, 37(12):2049–2055.
- MITRA, S. K. (2010). *Digital Signal Processing*. McGraw-Hill, 4th edition.
- OGATA, K. (2010.). *Ingeniería de Control Moderna*. Pearson Education.
- OPPENHEIM, A. V., WILLSKY, A. S., and NAWAB, S. H. (1998). *Señales y Sistemas*. Prentice Hall, 2nd edition.
- SAUPE, F. and KNOBLACH, A. (2015). Experimental determination of frequency response function estimates for flexible joint industrial manipulators with serial kinematics. *ELSEVIER Mechanical System and Signal Processing*, 53-53:60–72.
- WANG, Q., BI, Q., and ZOU, B. (1997). Use of FFT in relay feedback systems. *IET Electronics Letters*, 33(2):1099–1100.

WANG, Q., HANG, C., and BI, Q. (1999). A technique for frequency response identification from relay feedback. *IEEE Transactions on Control Systems Technology*, 7(1):122–128.

Apéndices

Apéndice A. Algoritmo programado en Raspberry Pi

A continuación se presenta el código implementado en Python para constituir el algoritmo de proceso de señal descrito previamente en la Fig. 15 y que permite efectuar una aproximación para el diagrama de Bode de un sistema, mediante cálculo en tiempo de ejecución (tiempo real) de la FFT de sus señales de estímulo y respuesta.

En este orden de ideas, la Fig. 18 incluye la secuencia de líneas del programa correspondiente.

```

pi@raspberrypi: ~/Desktop/Graficas_bode
File Edit Tabs Help
GNU nano 3.2 proyecto_grado.py

import pyaudio
import math
import numpy as np
import wave
import matplotlib.pyplot as plt
fs=44100
N=256
x=0
p1=pyaudio.PyAudio()
p2=pyaudio.PyAudio()
stream1=p1.open(format=pyaudio.paInt16,channels=1,rate=fs,input_device_index=3,input=True,frames_per_buffer=N)
stream2=p2.open(format=pyaudio.paInt16,channels=1,rate=fs,input_device_index=2,input=True,frames_per_buffer=N)
plt.ion()
fig2=plt.figure(figsize=(10,8))
ax3=fig2.add_subplot(111)
contador=0
while True:
    data1=stream1.read(N, exception_on_overflow=False)
    data2=stream2.read(N, exception_on_overflow=False)
    waveData1=wave.struct.unpack("%dh"%(N),data1)
    waveData2=wave.struct.unpack("%dh"%(N),data2)
    npArrayData1=np.array(waveData1)
    npArrayData2=np.array(waveData2)
    indata1=npArrayData1
    indata2=npArrayData2
    fftData1=np.abs(np.fft.rfft(indata1))*(1./N)
    fftData2=np.abs(np.fft.rfft(indata2))*(1./N)
    fftTime=np.fft.rfftfreq(N,1./fs)
    div_espectros=[]
    for z in range(0,len(fftData1)):
        a=20*(math.log10((fftData2[z])/(fftData1[z])))
        div_espectros.append(a)
    ax3.cla()
    ax3.plot(fftTime,div_espectros)
    ax3.set_xscale('log')
    ax3.set_title("Diagrama de Bode")
    ax3.set_xlabel("W[radianes]")
    ax3.set_ylabel("Magnitud[dB]")
    ax3.grid()
    ax3.set_ylim(-80,30)
    plt.pause(0.0001)

stream1.stop_stream()
stream2.stop_stream()
stream1.close()
stream2.close()
p1.terminate()
p2.terminate()

```

Figura 18. Implementación Python para algoritmo de proceso