



INSTITUTO TECNOLÓGICO
DE
BUENOS AIRES

INGENIERÍA ELECTRÓNICA
22.01 - TEORÍA DE CIRCUITOS

TRABAJO PRÁCTICO N°1

Grupo 4:
Álvarez, Lisandro
Fogg, Matias
Diaz, Ian
Delgado, Milton
Dieguez, Manuel
Oh, Victor

Legajos:
57771
56252
57515
56451
56273
56679

APROBADO EN FECHA:

ENTREGADO EL 19 DE AGOSTO DE 2019

1. Filtro Notch

1.1. Análisis del circuito

En este ejercicio se busca crear un filtro notch a partir del circuito 1.

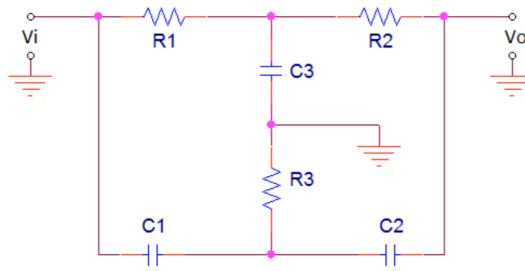


Figura 1: Filtro Notch Pasivo

En una primera instancia a través del uso de la herramienta matlab se logró conseguir la función transferencia a través de la resolución de un sistema de ecuaciones, en una segunda instancia se transformaron ambos circuitos tipo T en circuitos tipo PI mediante el teorema de Kennelly y se consiguió el circuito de la figura 2.

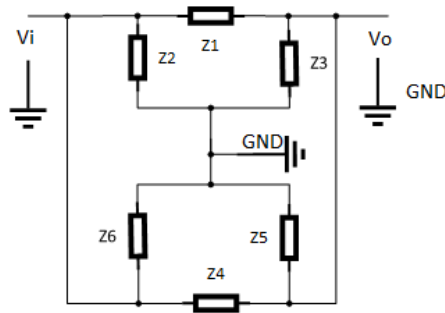


Figura 2: Mismo circuito con el teorema de Kenelly aplicado

Se notó que $Z2$ y $Z6$ se encontraban en paralelo, de la misma manera lo estaban $Z3$ con $Z5$ y $Z1$ con $Z4$, por lo que se prosiguió a calcular el equivalente y se obtuvo el circuito de la figura 3.

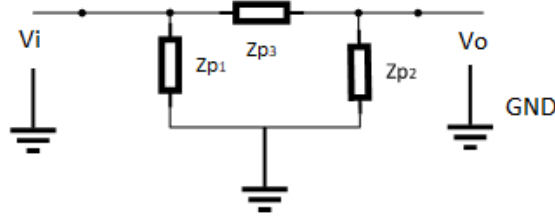


Figura 3: Circuito con el teorema de Kenelly aplicado simplificado

Con los valores:

$$Z_{p1} = Z_{p2} = \frac{C_3 R_3 s + 1}{C_3 s} ; Z_{p3} = \frac{4 R_3 (C_3 R_3 s + 1)}{C_3^2 R_3^2 s^2 + 1} \quad (1)$$

Una vez se llegó a este punto fue más fácil encontrar la función transferencia y se obtuvo lo detallado en la ecuación 2. Vale aclarar que se utilizó la ayuda propuesta por la cátedra y se consideró que $R_1 = R_2 = 2 R_3$ y $2 C_1 = 2 C_2 = C_3$.

$$H(s) = \frac{\left(\frac{s}{1/C_3 R_3}\right)^2 + 1}{\left(\frac{s}{1/C_3 R_3}\right)^2 + 4 \frac{s}{C_3 R_3} + 1} \quad (2)$$

Como se puede observar, la función transferencia describe un filtro Notch. Su frecuencia de corte es W_0 y su expresión se muestra en la ecuación 3.

$$W_0 = \frac{1}{C_3 R_3} \quad (3)$$

La frecuencia de corte pedida es $10,8kHz$, obtenemos así la relación que se puede ver en la ecuación 4.

$$R_3 = \frac{1}{C_3 2 \pi 10,8k} \quad (4)$$

Es posible dar valores al capacitor y así obtener un valor para las resistencias. Teniendo en cuenta los valores comerciales disponibles en el pañol, se tomo $C_3 = 10nF$ y se obtuvo $R_3 = 1,47k\Omega$. Al no haber disponibilidad de una resistencia fija de ese valor, se utilizo una $R_3 = 1,5k\Omega$. Tampoco se encontraron capacitores de $C = 5nF$ por lo que se utilizaron tanto $C_1 = 4,7nF$ como $C_2 = 4,7nF$. Estos valores se cargaron en LTspice y se obtuvo el bode de la figura 4.

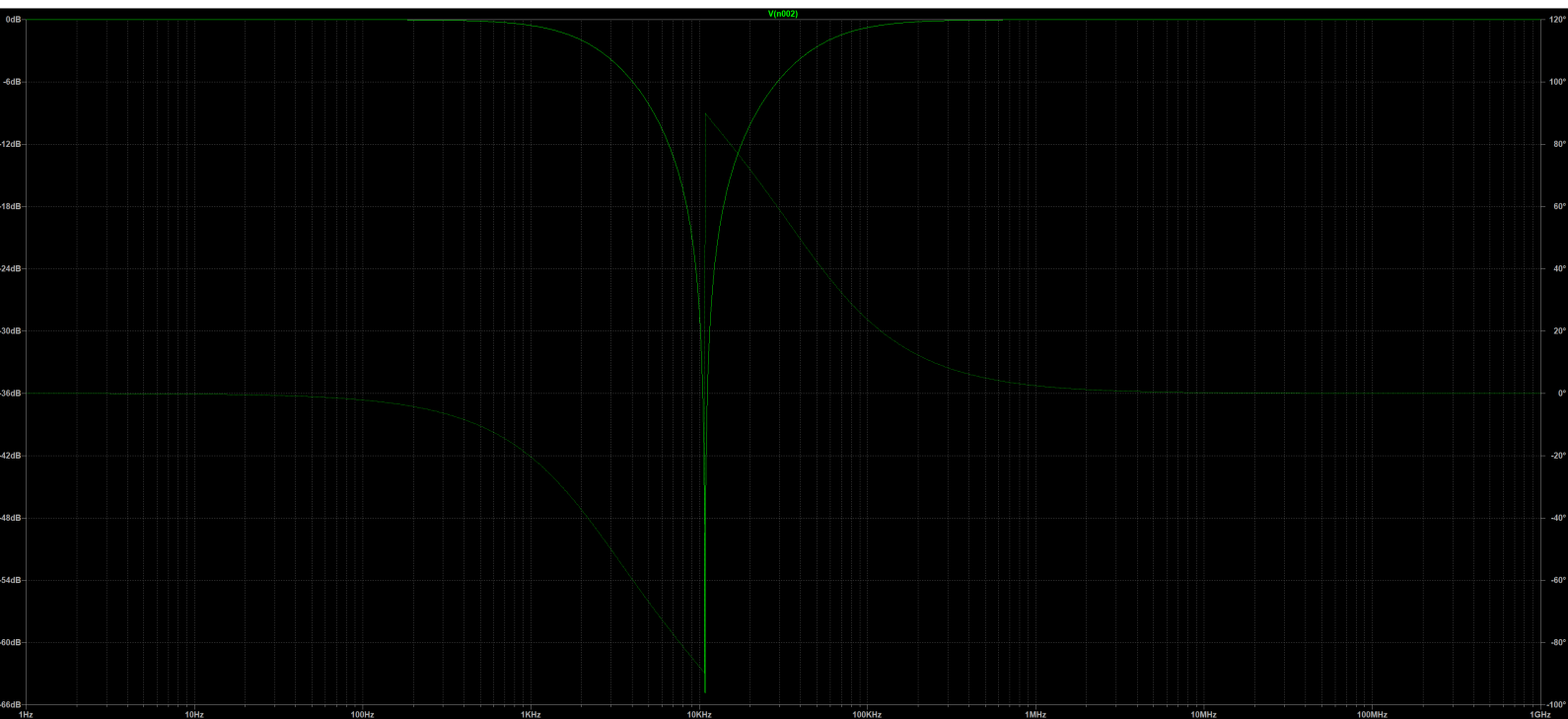


Figura 4: Circuito con los componentes definidos

Se puede observar que el comportamiento del bode describe un filtro notch y que la frecuencia de corte se ubica en $11,1kHz$. Si bien la frecuencia de corte pedida es $10,8kHz$ nos vemos obligados a tomar $11,1kHz$ por motivos de disponibilidad de componentes en el pañol. Luego las futuras mediciones se comparan respecto al bode obtenido en la figura 4.

Para obtener la respuesta impulsiva $h(t)$, se utilizó la antitransformada de Laplace conseguida mediante el uso de matlab. Esta resulto ser la que se muestra en la ecuación 5 y se puede ver graficada en la figura 5 .

$$y(t) = \delta(t) - \frac{4 e^{-\frac{2t}{C3R3}}}{C3R3} \left(\cosh\left(\frac{\sqrt{3}t}{C3R3}\right) - \frac{2\sqrt{3} \sinh\left(\frac{\sqrt{3}t}{C3R3}\right)}{3} \right) \quad (5)$$

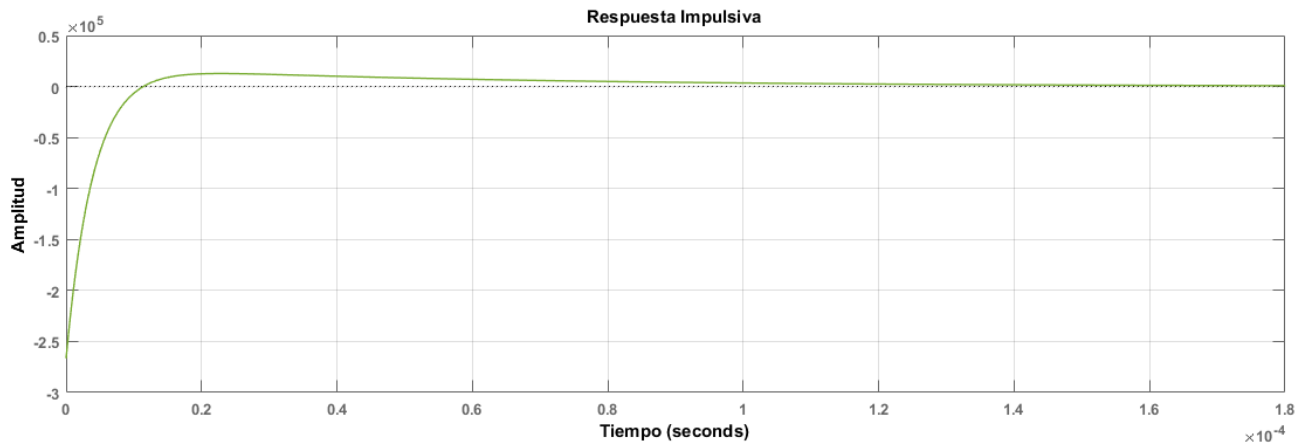


Figura 5: Respuesta impulsiva

Por último para poder terminar de caracterizar el sistema hace falta el diagrama de polos y ceros. Los polos y ceros se obtienen fácilmente si reordenamos la función transferencia como se ve continuación:

$$H(S) = \frac{(S - S_1)(S - S_2)}{(S - P_1)(S - P_2)S}$$

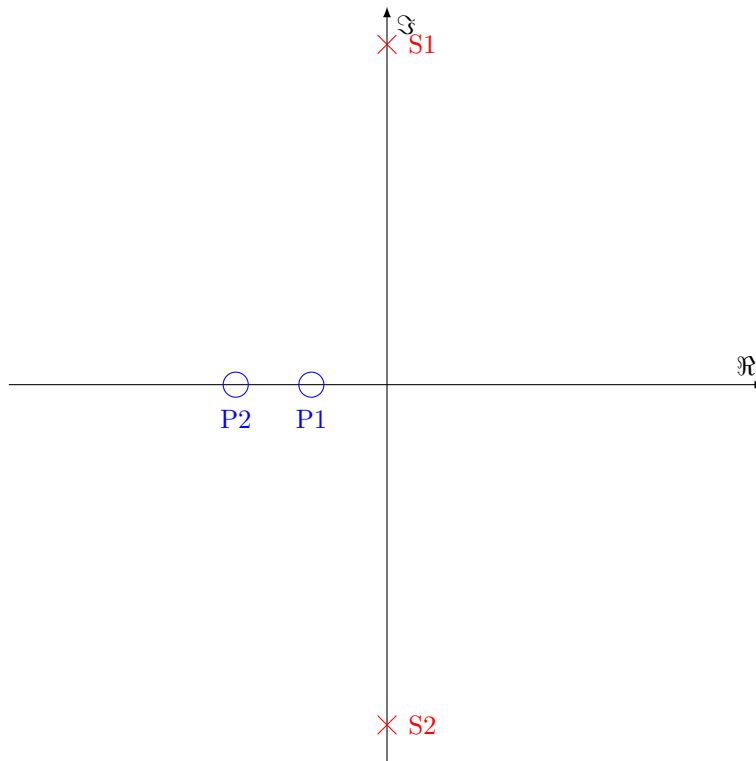
Hay dos ceros:

$$S_1 = 69743,35691j \quad S_2 = -69743,35691j$$

Hay dos polos:

$$P_1 = -18687,67616 \quad P_2 = -260285,7515$$

Como se puede ver los dos ceros se encuentran sobre el eje imaginario y los dos polos en el eje real del semiplano negativo



1.2. Respuesta en frecuencia

Con los valores de los componentes calculados anteriormente, se diseñó una placa en Altium. Su diseño se puede ver en la figura 6.

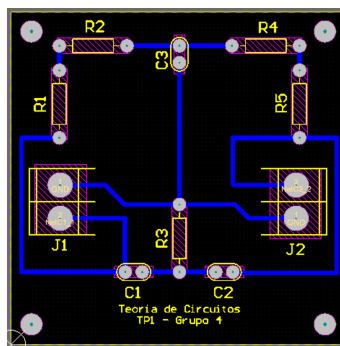


Figura 6: Placa diseñada en Altium

Notar que se utilizaron que utilizar dos resistencias en serie de $1,5k\Omega$ para obtener una resistencia de $3k\Omega$.

Para medir la respuesta en frecuencia se utilizó un generador de señales y se alimentó al circuito

con un senoide en V_{in} y se midió V_{out} con la ayuda de un osciloscopio.

En primer lugar se excitó el circuito con una señal senoidal de $10 V_{pp}$ a una frecuencia de $11,1 kHz$, que es la frecuencia de corte buscada con nuestros componentes. Se esperaba ver una señal totalmente atenuada ya que la señal de entrada está en la frecuencia de corte. Se realizó un barrido rápido en las frecuencias cercanas a $11,1 kHz$ y notamos que nuestro filtro tenía la mayor atenuación en $11,3 kHz$. Los resultados fueron los siguientes: $V_{in} = 9,73V$ y $V_{out} = 0,038V$ lo que da una atenuación de $20 \log(\frac{V_{out}}{V_{in}}) = -48,16db$, representando la salida tan sólo el 0,39% de la entrada.

Podemos decir que se puede considerar que la señal de entrada fue totalmente atenuada a esta frecuencia y que hay que tener en cuenta el ruido, el osciloscopio es susceptible al ruido. Esto explica porque V_{out} no es cero en la frecuencia de corte del Notch. Lo que se está midiendo en esta situación es prácticamente ruido ya que este tiende a aumentar al ser tan pequeña la amplitud de la señal.

Si tenemos en cuenta que la frecuencia de corte que se planeaba obtener con los componentes elegidos y la que obtuvimos, el filtro fabricado tiene un error un 1,82%. Y si se compara con los $10,8 kHz$ que se buscaban en un comienzo, este error es de un 4,63%.

Se prosiguió a realizar el bode completo. Para esto se mantuvo una senoide de $10 V_{pp}$ pico a pico y se fue modificando la frecuencia de esta. Sabiendo de antemano como es la curva que describe el bode, se tomaron mas puntos en las áreas mas características del bode. Como lo es el área cercana a la frecuencia de corte. Los resultados del bode completo se pueden ver en la figura 7.

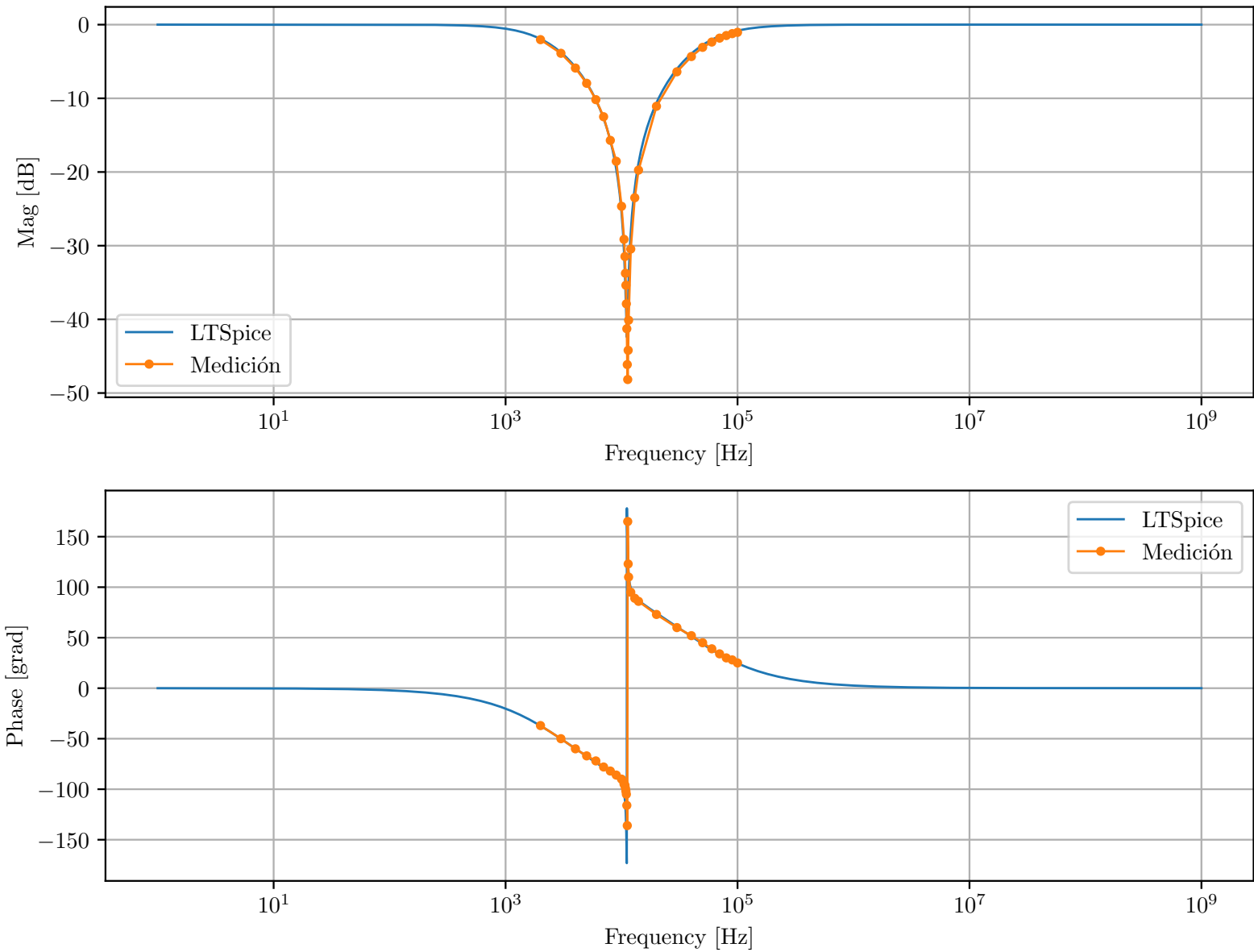


Figura 7: Filtro Notch Pasivo

Como se puede observar, la figura 7 tiene superpuesto el bode de LTSpice (curva azul) que se mostró en la figura 4 y el bode que se obtuvo de forma experimental (curva naranja). Los resultados son sumamente satisfactorios. En el bode de la medición se puede apreciar la frecuencia de corte y como el resto de los puntos se asemejan a la curva teórica calculada en LTSpice.

1.3. Respuesta al escalón

Para la obtención de la respuesta teórica al escalón se realizó lo mismo que se mencionó anteriormente para la respuesta al impulso y se muestra a continuación en la ecuación 6 su resultado y en la figura 8 su gráfico.

$$y(t) = 1 - \frac{4}{3} \sqrt{3} e^{-\frac{2t}{C3R3}} \sinh\left(\frac{\sqrt{3}t}{C3R3}\right) \quad (6)$$

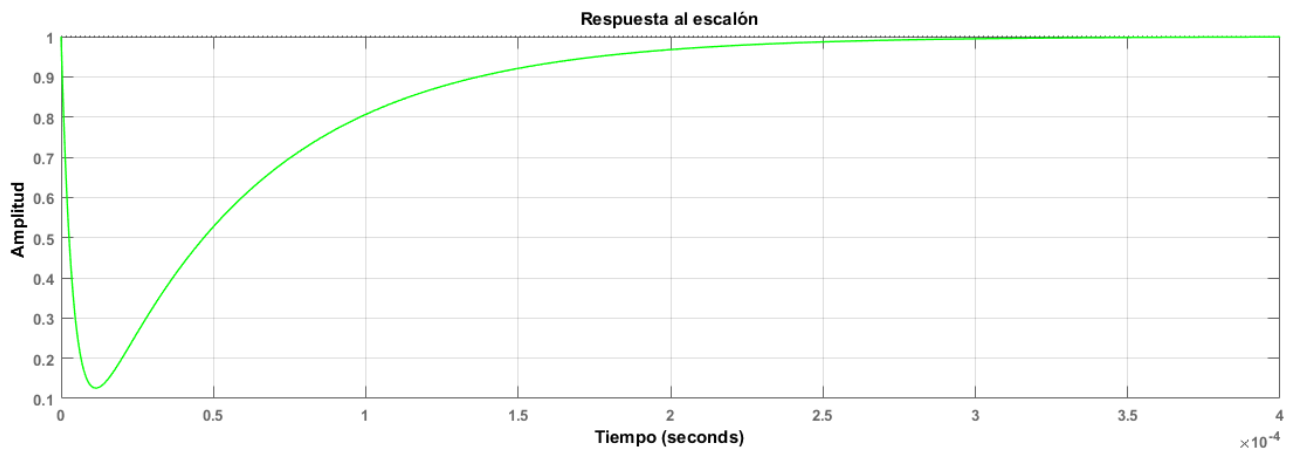


Figura 8: Respuesta al escalón

A continuación en la figura 9 se muestra la imagen tomada del osciloscopio al momento de excitar el circuito con el escalón:

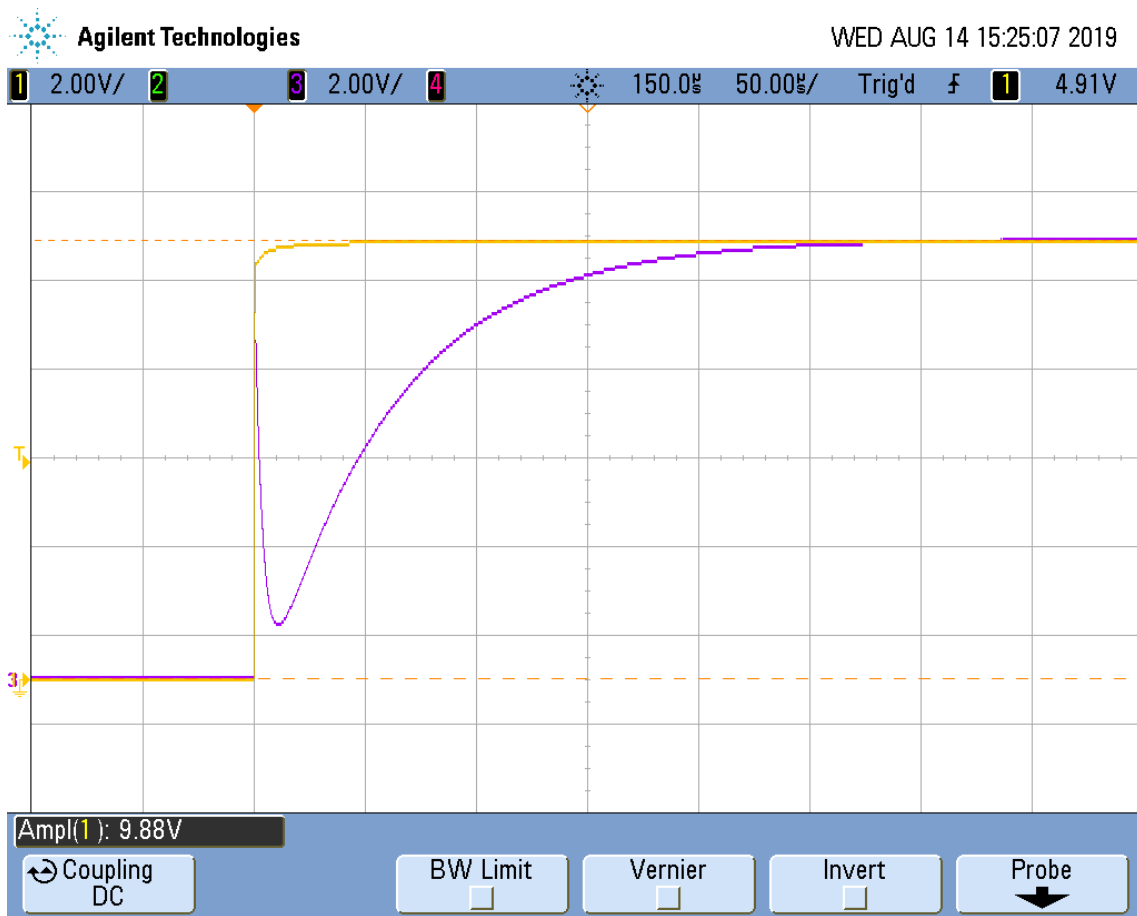


Figura 9: Respuesta al escalón medida

La línea violeta es la repuesta al escalón medida y la amarilla es el escalón. Podemos decir entonces que se corresponde de manera adecuada la respuesta medida con la teórica y que el circuito se comportó como se esperaba.

2. Filtro pasabajos

Se pidió aplicar a un filtro RC de frecuencia de corte $f_0 = 64 \text{ (kHz)}$ una onda cuadrada de $10 V_{pp}$ con frecuencia de $f = 32 \text{ (kHz)}$. Los resultados obtenidos empíricamente fueron los que se muestran en las figuras 10 y 11. A su vez, se calculó la transferencia del circuito idealmente resultando ser:

$$H(s) = \frac{1}{1 + sRC} \quad (7)$$

Si simulamos la transferencia del circuito en LTSpice, el resultado es el que se ve en las figuras 11 y 12.

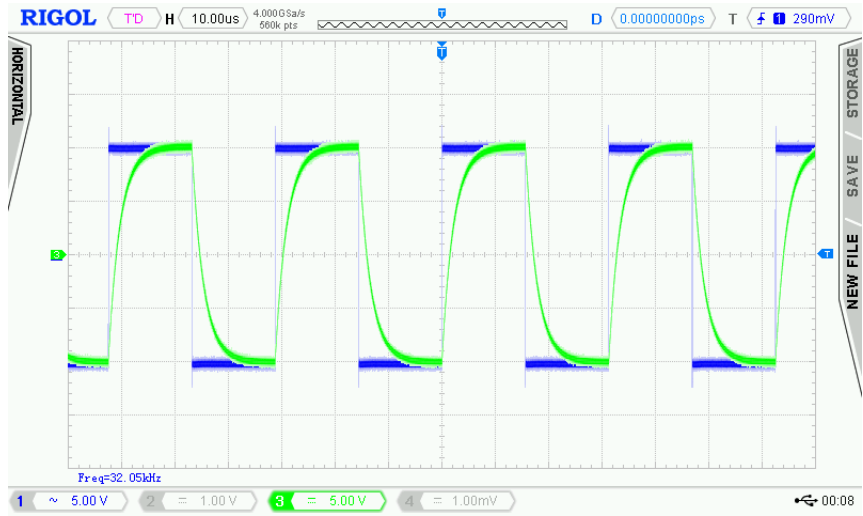


Figura 10: Resultados

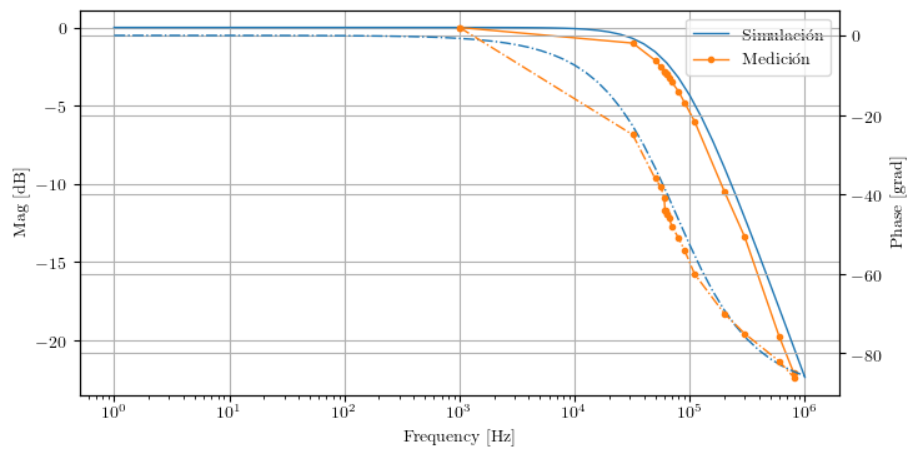


Figura 11: Simulacion LTSpice y Mediciones

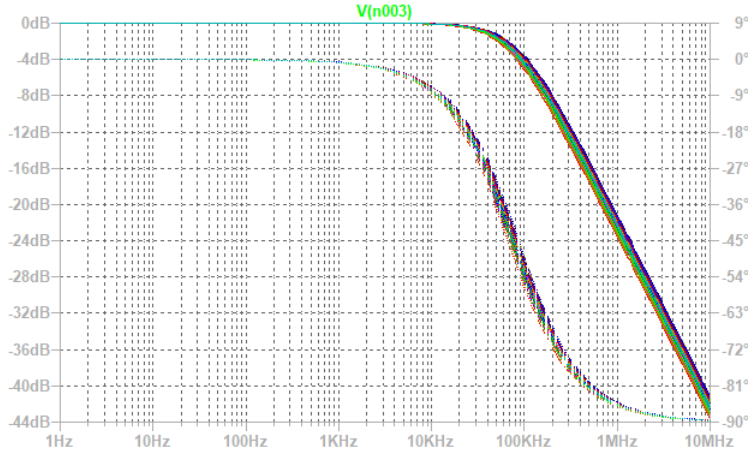


Figura 12: Análisis montecarlo

2.1. Cálculo de armónicos

Si queremos ver como reacciona el circuito a una señal cuadrada, debemos calcular primero como afecta nuestro circuito a la onda de entrada. Como sabemos que la onda es una cuadrada, haciendo su descomposición de suma de señales trigonométricas, los coeficientes de Fourier resultan ser:

$$a_0 = 0$$

$$a_n = 0$$

$$b_{2n-1} = \frac{20}{(2n-1)\pi}$$

$$b_{2n} = 0$$

Por lo tanto, la onda cuadrada se puede expresar en el espacio temporal como se ve en la expresión 8. Sin embargo, podemos ver, por temas de idealización, que una señal cuadrada ideal, se puede aproximar por una suma de términos finitos de senoidales, por lo tanto, si aproximamos la cuadrada con 10 términos, podemos ver como la aproximación van quedando mas parecidas, esto se muestra en la figura 13. A medida que agreguemos mas términos a nuestra suma, menos sera la diferencia con una onda cuadrada ideal. No obstante, hay que tener en cuenta que, como fue visto en Matemática V, al tener una discontinuidad no evitable cada $\frac{T}{2}$, siendo T el período de la señal, se generaran sobrepicos en los puntos de discontinuidad. por ende, si llamamos $x(t)$ a la función cuadrada ideal e $y(t)$ a su aproximación por senoides, $x(t)$ será igual a $y(t)$ en todos los números reales exceptuando los puntos de discontinuidad. Esto quiere decir, que es posible que al trabajar con ondas cuadradas, se encuentren sobrepicos.

$$x(t) \sim \sum_{n=1}^{\infty} \frac{20}{(2n-1)\pi} \sin(2\pi(2n-1)f_0 t) \quad (8)$$

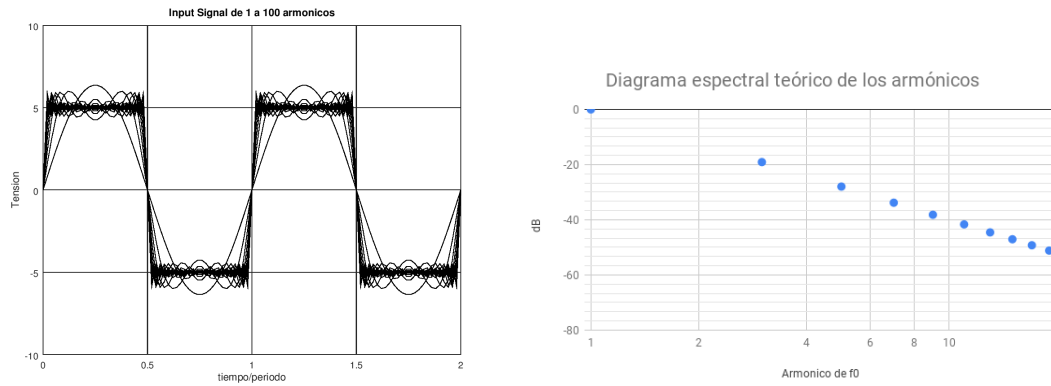


Figura 13: Representación de onda cuadrada mediante suma de senoidales y su diagrama espectral teórico

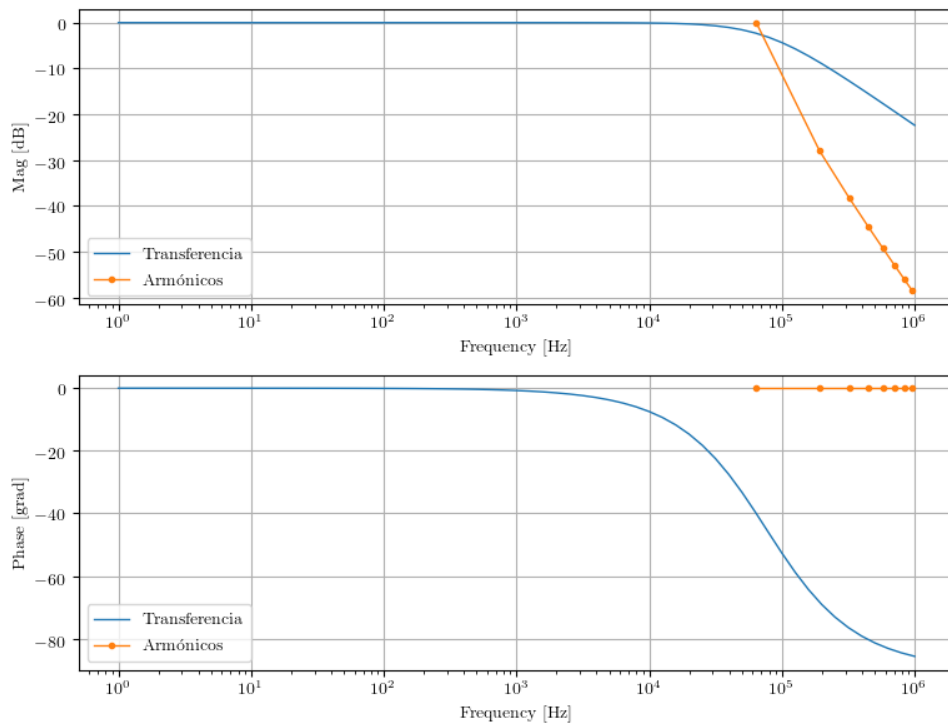


Figura 14: Transferencia superpuesta con los armónicos (los pto de los armónicos están unidos por líneas)

Debido a que la entrada es una función continua a trozos de período T , entonces:

$$y(t) \sim \sum_{n=1}^{\infty} |X_{2n-1}| |H((2n-1)f_0)| \cos[2\pi(2n-1)f_0t + \phi((2n-1)f_0) + \theta_{2n-1}]$$

$$X_{2n-1} = i \frac{b_{2n-1}}{2} \quad (9)$$

$$H(f) = |H(f)| e^{i\phi(f)}$$

A su vez;

$$H(f) = \frac{1}{1 + i2\pi fRC} \quad (10)$$

Por lo tanto, haciendo calculos de las ecuaciones 9 y 10, concluimos que:

$$\phi(f) = -\arctg(2\pi fRC)$$

$$H(f) = \frac{1}{\sqrt{(2\pi fRC)^2 + 1}}$$

$$\theta_n = \frac{\pi}{2}$$

Finalmente, podemos escribir la salida como la ecuacion 11 y graficamente se ve como la figura 15.

$$y(t) \sim \sum_{n=1}^{\infty} \frac{20}{(2n-1)\pi \sqrt{(2\pi(2n-1)f_0RC)^2 + 1}} \cos\left(2\pi(2n-1)f_0t - \arctg(2\pi(2n-1)f_0RC) + \frac{\pi}{2}\right) \quad (11)$$

Teniendo en cuenta que $f_0 = \frac{1}{2\pi RC}$, y llamando $k=2n-1$, entonces

$$y(t) \sim \sum_{k \text{ impares}} \frac{20}{k\pi \sqrt{k^2 + 1}} \cos\left(2\pi k f_0 t - \arctg(k) + \frac{\pi}{2}\right)$$

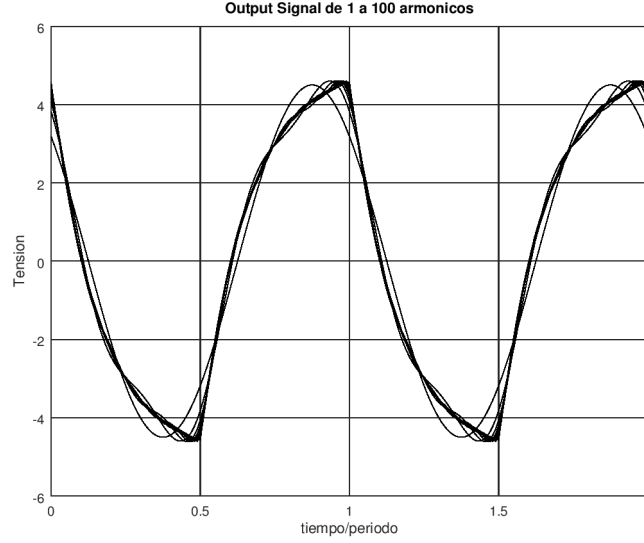


Figura 15: Salida del RC en armónicos

2.2. Circuito como Integrador

Como ya sabemos, un circuito integrador es aquel que cumple que su funcion transferencia sea $H(s) = \frac{1}{s}$, como vemos en la ecuacion 7, nuestro circuito no posee esa transferencia, sin embargo, si procuramos movernos en un intervalo donde sRC sea lo suficientemente grande comparado con 1, podremos aproximar a una funcion transferencia integradora. Es decir;

$$Si \ sRC \gg 1 \implies H(s) = \frac{1}{1 + sRC} \approx \frac{1}{RC} \frac{1}{s}$$

Por ende, si cambiamos al espacio de frecuencias, procurando que $2\pi fRC \gg 1$ podemos obtener la transferencia de un circuito integrador. En particular, para $R = 3,68(k\Omega)$ y $C = 560(pF)$ debemos concluir que para una frecuencia $f_a \gg 63(kHz)$, nuestro circuito se comportará como un integrador.

En particular, si tenemos una frecuencia $f_i = 300(kHz)$, $2\pi f_a RC \gg 1$, con lo cual podemos aproximar el denominador de la transferencia y usar nuestro circuito como integrador, como se puede observar en la figura 16.

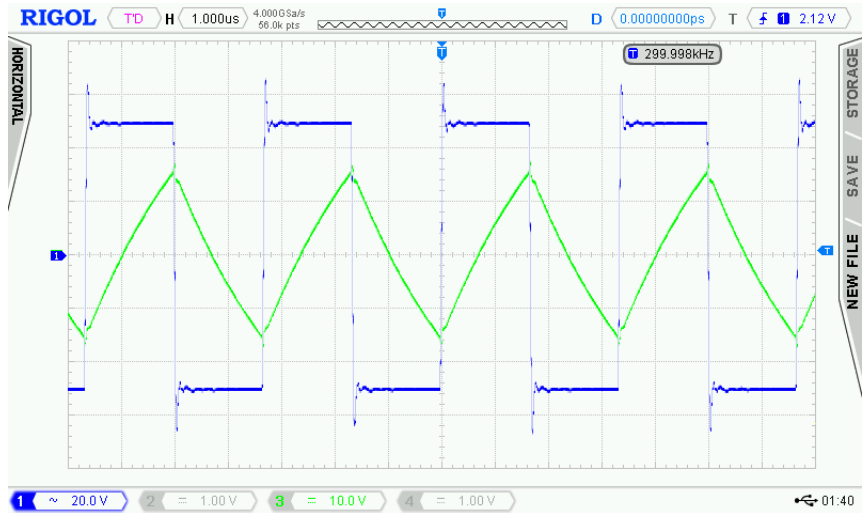


Figura 16: Circuito como integrador a 300 (kHz)

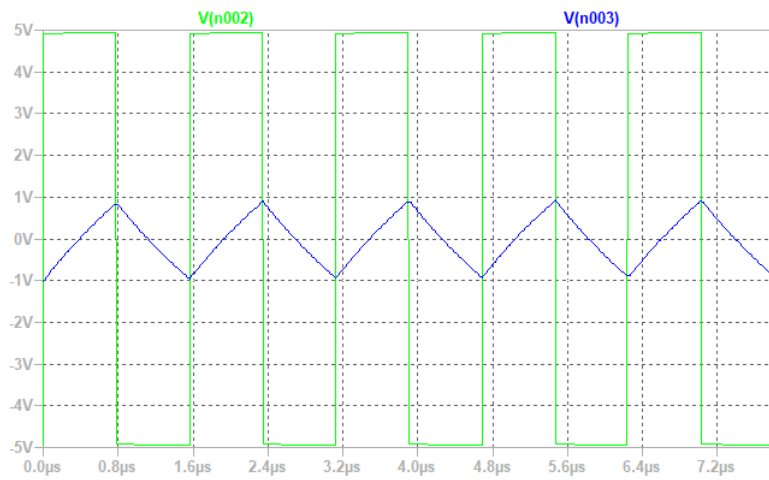


Figura 17: Simulacion del circuito a 300 (kHz)

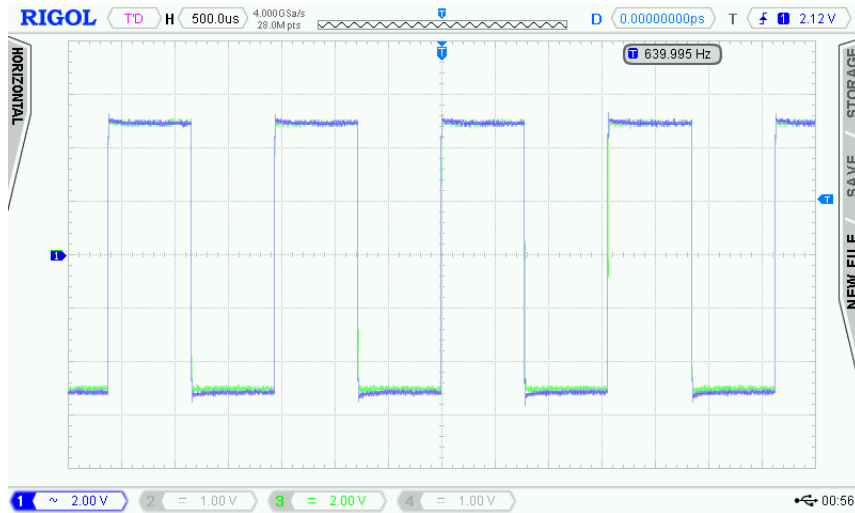


Figura 18: Circuito andando a 640(Hz)

3. Plot Tool

Se pidió desarrollar un programa con interfaz gráfica de usuario, cuyo propósito fuera el de facilitar la confección de gráficos, mediante un método de ploteo automático. El programa debe implementarse en Python, y debe cumplir con los siguientes requisitos básicos. En primer lugar, respecto a fuentes desde las cuales se puede ingresar datos a graficar, los mismos deben ser:

- Función transferencia
- Archivo LTspice
- Plantilla de Excel o csv con mediciones

En segundo lugar, se debe otorgar al usuario las siguientes capacidades:

- Especificar la etiqueta para el eje x.
- Especificar la etiqueta para el eje y.
- Guardar el gráfico como imagen.
- Borrar todos los gráficos anteriores.
- Graficar más de tres señales simultáneamente.

3.1. Guía de usuario

A continuación se explicará detalladamente cada paso referente a la utilización de la GUI implementada. Los pasos a desarrollar son el ingreso de datos desde los distintos tipos de fuentes, edición de las etiquetas, seleccionar la visualización del gráfico en una o dos figuras, eliminar los gráficos anteriores y por ultimo, la exportación de los gráficos.

Previo a desarrollar los pasos se muestra una figura con el diseño de la interfaz gráfica, tal cual se muestra una vez que el usuario inicia el programa

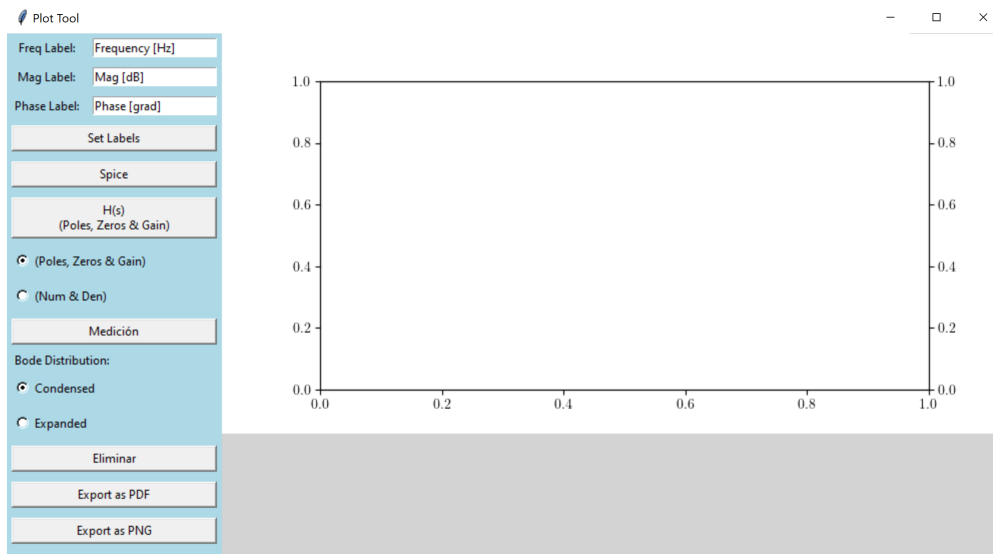


Figura 19: GUI al iniciar el programa

A la izquierda se encuentra el panel de controles, cuyo funcionamiento se explicara a continuación, mientras que a la derecha se encuentra la figura donde se graficarán nuestros datos.

3.1.1. Ingresar datos

El programa nos permite ingresar datos desde tres fuentes distintas, archivos de LTspice, transferencias analíticas o mediciones(en un formato determinado que se explicara posteriormente). Los tres botones para ingresar los mismos se muestran en la Figura 20.



Figura 20: Elementos del panel de controles con referencias

El botón 'Spice' abrirá un explorador de archivos, que nos permitirá seleccionar un archivo .txt que previamente habremos exportado desde una simulación de LTspice. El archivo debe corresponder a una simulación del tipo 'AC Analysis', que cuente con la transferencia de un solo nodo. Una vez seleccionado el archivo correspondiente aparecerá el gráfico correspondiente en la figura.

Para ingresar una transferencia analítica contamos con dos posibilidades, puede ser ingresada mediante sus polos, ceros y ganancia ('Poles, Zeros & Gain') o bien mediante los polinomios del numerador y denominador('Num & Den'). La opción se selecciona con los controles que se encuentran debajo del botón 'H(s)', y se debe indicar previo a pulsar el botón. Si seleccionamos la opción 'Poles, Zeros & Gain', al pulsar el botón 'H(s)' se abrirá una ventana (Figura 21) en la cual se deben ingresar los datos. El formato de los polos y ceros es el siguiente: suponiendo que la función transferencia tiene n polos $p1, p2, \dots, pn$, se deberán ingresar separados por un espacio: $p1\ p2\ \dots\ pn$. De la misma forma para los ceros, si la transferencia cuenta con m ceros $z1, z2, \dots, zm$, se deberán ingresar separados por un espacio: $z1\ z2\ \dots\ zm$. El campo de la ganancia se llena con el valor de la ganancia correspondiente.

Figura 21: Prompt modo 'Poles, Zeros & Gain'

Al seleccionar el modo 'Num & Den' y pulsar el botón 'H(s)' se abrirá un prompt como el que se muestra en la Figura 22. Los datos se deben ingresar como un conjunto de coeficientes de polinomios ordenados (estilo polinomio de Matlab). Por ejemplo, si quisieramos ingresar el polinomio $ax^2 + bx + c$, se deberán ingresar los coeficientes a, b y c en forma ordenada separados por un espacio, de la siguiente forma: $a\ b\ c$. Los campos Numerador y Denominador admiten el mismo formato.

Figura 22: Prompt modo 'Num & Den'

La última opción de ingresar datos admite dos tipos de archivos, .csv y .xlsx. El formato de los mismos deberá ser el siguiente:

- **.csv:** Los datos deberán contener headers, el carácter de separación de columnas debe ser ; y el orden de las mismas debe ser *frecuencia;magnitud;fase*; frecuencia en Hz, magnitud en dB y fase en grados.
- **.xlsx:** Los archivos de Excel deberán estar formateados de la siguiente manera: los datos *frecuencia, magnitud y fase* deberán colocarse en las columnas A, B y C, respectivamente, el archivo no deben contener headers, y el primer dato deberá colocarse en la fila 0

Cada vez que se cargue un dato a graficar, el programa abrirá automáticamente un prompt en el cual se podrá ingresar una etiqueta que describa el gráfico correspondiente.

3.1.2. Editar etiquetas

En la parte superior del panel de controles se encuentran tres cuadros de texto en los cuales podemos ingresar las etiquetas de los ejes, como se muestra en la Figura 23. Por default estas son *Frequency [Hz]*, *Mag [dB]* y *Phase [grad]* para frecuencia, magnitud y fase respectivamente. Para modificarlas, debemos ingresar la etiqueta deseada y hacer click en 'Set Labels'.

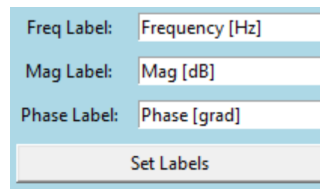


Figura 23: Controles para modificar etiquetas de ejes

3.1.3. Distribución de los gráficos

La interfaz cuenta con una herramienta para poder seleccionar la disposición de las curvas. Mediante este control podemos disponer las curvas de magnitud y fase en una misma figura ('Condensed'), o separar las curvas de magnitud y fase en dos figuras diferentes ('Expanded'). Las Figuras 24 y 25 muestran los gráficos en modo 'Condensed' y 'Expanded', respectivamente.

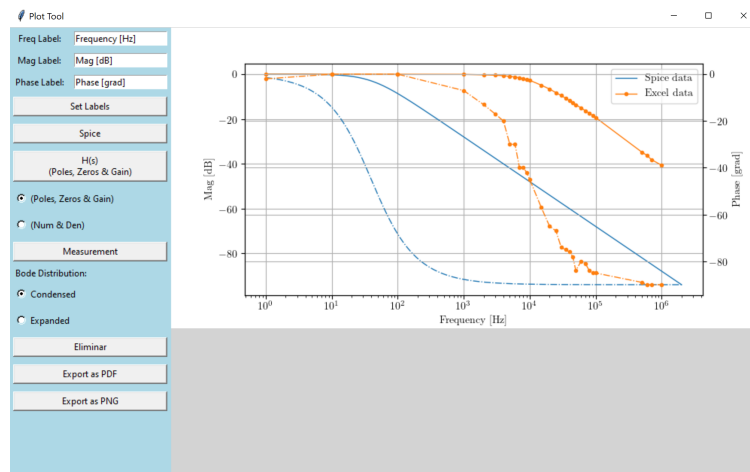


Figura 24: Gráficos en modo 'Condensed'

3.1.4. Eliminar gráficos

Para eliminar los gráficos cargados, es necesario hacer click en la opción 'Remove Plots'. Esta herramienta borrará todos los gráficos que estén graficados en la figura. No es posible remover los gráficos individualmente

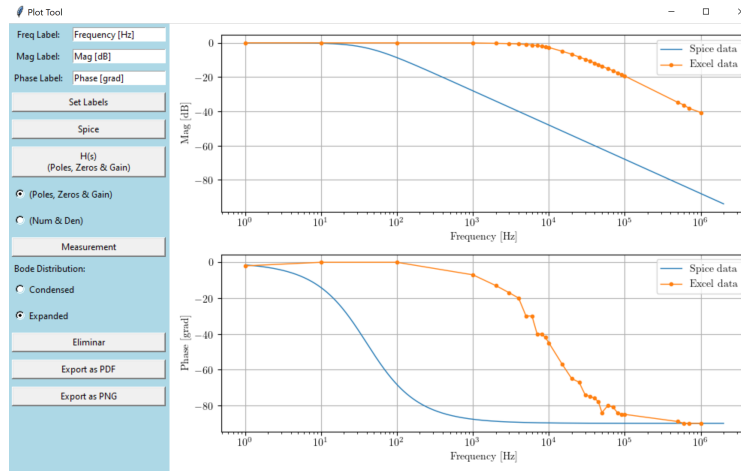


Figura 25: Gráficos en modo 'Expanded'

3.1.5. Exportar gráficos

La ultima herramienta de la interfaz permite exportar los gráficos en dos formatos: PDF y PNG. El archivo exportado tendrá el formato que se presente en la GUI según el modo seleccionado (Condensed o Expanded). Al seleccionar el botón de *Export as...* se abrirá un explorador de archivos para indicar la ubicación y nombre de nuestro archivo exportado. Las Figuras 26 y 27 muestran las figuras generadas en ambos modos:

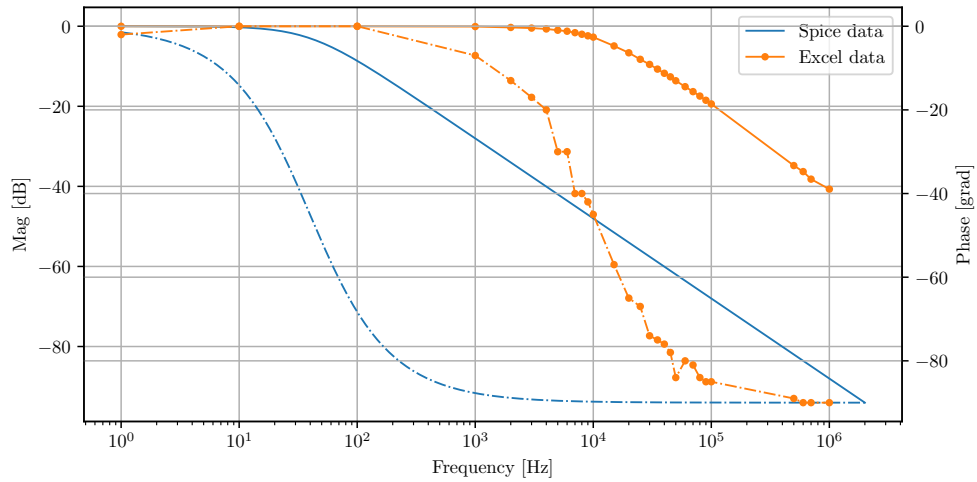


Figura 26: Resultado exportado en modo 'Condensed'

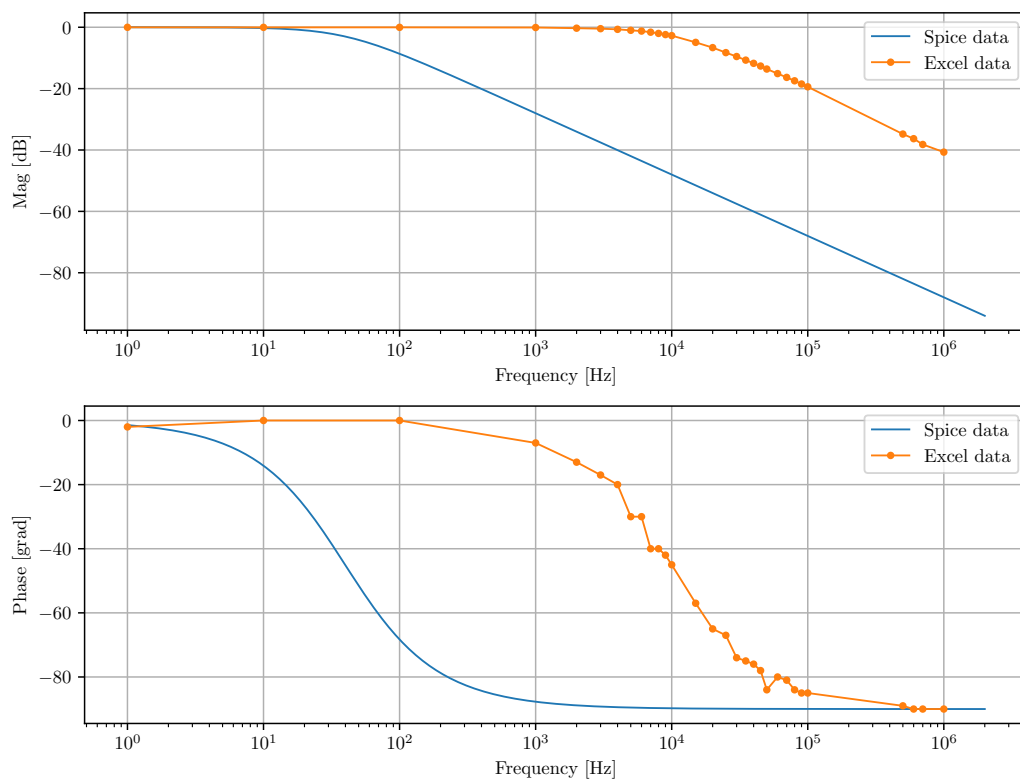


Figura 27: Resultado exportado en modo 'Expanded'