

Teoría de Circuitos 2019

Trabajo Práctico de Laboratorio N°1

Filtros pasivos y análisis computacional

Consideraciones generales:

- Se utilizará la letra **N** para denotar al número del grupo.
- Cuando se indiquen valores de resistencias, se deberá sintetizar este valor con combinaciones de **a lo sumo** un par en serie o paralelo para obtener el menor error posible.
- Está permitido asistir el desarrollo matemático mediante programas algebraicos, sin embargo todos los resultados y pasos intermedios más importantes deberán quedar asentados en el informe.
- Se realizarán todas las simulaciones de circuitos mediante LTSpice.
- Todas las gráficas de respuesta en frecuencia deberán expresarse en escala semi-logarítmica.
- Se espera coherencia en las cifras significativas a lo largo de las mediciones, cálculos y resultados, tanto en escala lineal como logarítmica.
- Se les recuerda a los alumnos que la política de Fraude y Plagio del Instituto rige sobre este trabajo.

Pautas para la evaluación del informe (en orden de importancia):

- Contenido y capacidad de síntesis.
 - Se penalizarán contenidos irrelevantes.
 - Se valorará la presentación clara, concisa, específica y sin redundancias.
 - Se esperan conclusiones relevantes dentro del desarrollo de cada tema y del trabajo práctico en general.
- Adecuado manejo y presentación de magnitudes numéricas.
- Organización grupal del trabajo.
 - Se espera el mayor grado de cohesión y homogeneidad en la resolución de los distintos enunciados. Se deben respetar un estándar y objetivos comunes.
- Originalidad e Inventiva
- Presentación, redacción y ortografía.
- Aportes no obligatorios

Entrega:

- Versión digital: Lunes 19 de Agosto de 2019 a las 23:59 Hs.
- Fecha del oral: Jueves 22 de Agosto de 2019 a las 9 Hs.

1 Filtro notch pasivo

Objetivos:

- Aprender a hacer placas
- Resolución de circuitos con métodos alternativos (Kenelly, cuadripolos, Mason, etcétera)
- Comparación entre teoría, simulación y práctica

A partir del circuito de la figura 1 se pide atender los siguientes incisos:

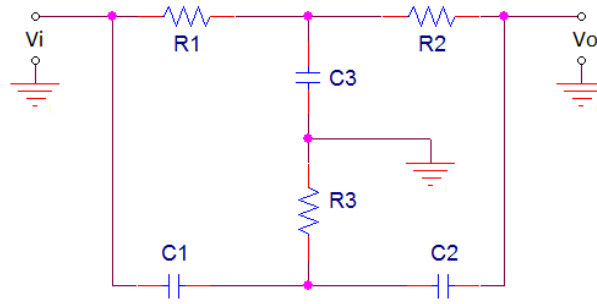


Figure 1: Filtro Notch Pasivo

- Diseñar un filtro rechaza banda de segundo orden con $f_0 = 2.7 \text{ kHz} \cdot N$.
- Caracterizar el sistema. Calcular la respuesta impulsiva.
- Calcular, simular y medir la respuesta en frecuencia del circuito comparando los resultados y **explicando diferencias**.
- Calcular la expresión analítica de la respuesta al escalón del circuito, simularla, medirla y comparar resultados. ¿Hay algún instante característico? *Pista:* $\frac{d}{dt} = 0$.

Ayuda: Considerar $R_1 = R_2 = 2 \cdot R_3$ y $C_1 = C_2 = \frac{C_3}{2}$.

2 Filtro pasa-bajos pasivo

Objetivo del ejercicio:

- Utilizar una protoboard.
- Experimentar las limitaciones de dicho circuito en alta frecuencia.
- Aplicación de análisis matemático de Fourier a circuitos.

Experiencia de laboratorio: Aplicar a un filtro RC pasabajos de $f_0 = 10\text{kHz}$ una señal cuadrada de $10V_{pp}$ @ 8kHz . N sin valor medio.

- a. Analizar y explicar los resultados obtenidos.
- b. Realizar un diagrama espectral teórico de las componentes de la señal de entrada y superponerlo con la respuesta en frecuencia del filtro.
- c. Calcular de forma teórica como afecta el filtro a cada una de las componentes de la señal.
- d. Aplicar una señal de las mismas características pero con frecuencia $160\text{Hz} \cdot N$ y obtener conclusiones comparando los resultados con ambas señales.
- e. ¿Se puede utilizar el sistema como integrador?

Recomendación: Repasar series trigonométricas de Fourier.

3 Plot Tool 2019

Objetivos del ejercicio:

- Aprender a utilizar Python, realizando un programa con una interfaz visual (GUI).
- Desarrollar un método de ploteo automático para realizar los gráficos de los tps.

Implementar una GUI en Python que permita al usuario superponer gráficos de respuesta en frecuencia. Los mismos pueden ser ingresados de tres formas distintas:

- Función transferencia.
- Archivo de LTspice.
- Plantilla de excel o csv con mediciones (en un formato estándar a elección). En caso de extrapolar la curva entre los puntos, debe ser claro qué puntos corresponden a las mediciones reales.

El usuario debe poder:

- 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.

Cada función del gráfico debe estar acompañada por una etiqueta pertinente. Puede ser predeterminada o (opcional) ingresada por el usuario.

Los archivos de simulaciones y mediciones se deben poder seleccionar mediante el sistema de archivos de la computadora. La transferencia teórica debe poder ingresarse a través de la GUI, ya sea a través de un archivo o con una interfaz específica en el programa (no línea de comando, no editar el código).

Opcional:

- Respuesta al escalón, al impulso, a senoidal, a señales arbitrarias...

- Marcar puntos y/o áreas de interés
- “Togglear” (mostrar/eliminar) cada curva
- Permitir guardar módulo y fase en imágenes separadas o en la misma imagen
- Otras funcionalidades extra

A continuación se muestra un esquema de una posible interfaz gráfica a implementar, según los requerimientos mencionados anteriormente:

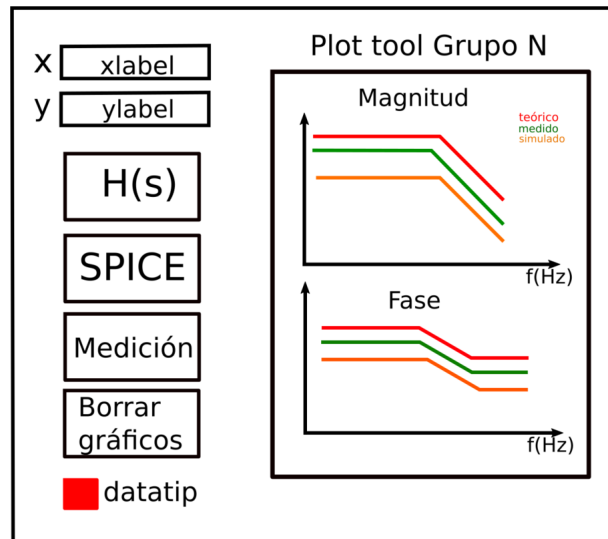


Figure 2: GUI a implementar

Sugerencias:

- Utilizar las librerías Tkinter (GUI), Pandas (csv y excel), Matplotlib (gráficos), Scipy.signal, Numpy (álgebra)
- Una opción para ingresar la transferencia es usar un txt que tenga los coeficientes del numerador y denominador, o ceros y polos.
- Ejemplos de programación en Python para Electrónica: <https://github.com/newtonis/EjemplosElectrotecnia>