Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería Eléctrica
IEE2103 – Señales y Sistemas

Tarea 2

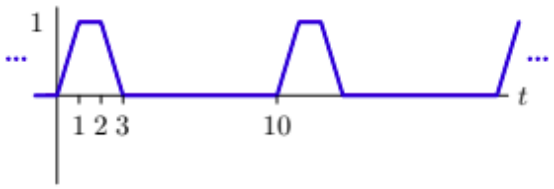
Entrega: 25 de septiembre a las 23:59 hrs.

Instrucciones generales

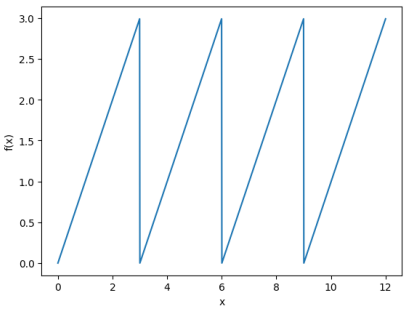
1. Para la parte de implementación, debe adjuntar un código desarrollado en Python (*.ipynb*).
2. Deben entregar un PDF con el desarrollo llamado *Apellido*_Tarea1.pdf (donde *Apellido* es su apellido).
3. La tarea es individual, pero pueden comentar las preguntas con sus compañeros.
4. Cada eje de los gráfico debe incluir un nombre descriptivo y su respectiva unidad.

Ejercicios Teóricos

1. **[2 puntos]** Determine la serie de Fourier de las siguientes señales periódicas:



(a) Primera Señal Periódica



(b) Segunda Señal Periódica

2. **[2 puntos]** Utilizando transformada de Fourier muestre si las siguientes expresiones son verdaderas o falsas:

i)

$$\int_{-\infty}^{\infty} \frac{\sin^2(x-a)}{(x-a)^2} dx = \pi \quad \forall a \in \mathbb{R}$$

II)

$$\int_0^\infty \frac{\cos(\omega t)}{1+\omega^2} d\omega = \frac{\pi}{2} e^{-|t|}$$

3. **[2 puntos]** Determine la transformada de Fourier de las siguientes funciones:

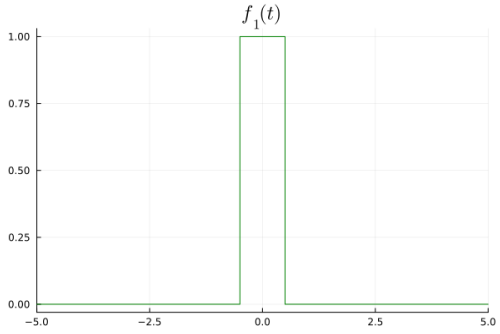
I)

$$f(t) = t \cdot e^{i\omega t}$$

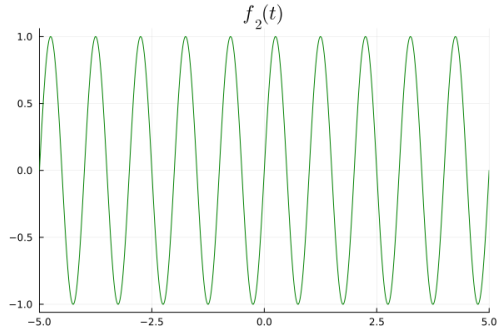
II)

$$f(t) = t \cdot e^{-\pi t^2}$$

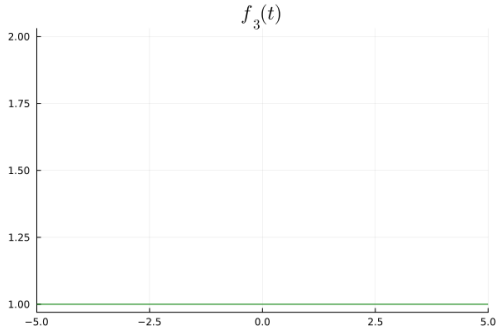
4. **[2 puntos]** A continuación se presentan 6 gráficos: tres son funciones en el tiempo y los otros tres son las transformadas de Fourier de estas funciones, calculadas numéricamente en código. Indique la correspondencia de cada función en el tiempo a cada función en frecuencia, teniendo en mente que una obtención numérica implica resultados no exactos. Justifique sus respuestas.



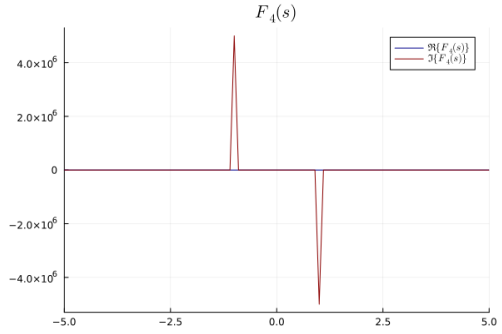
f_1



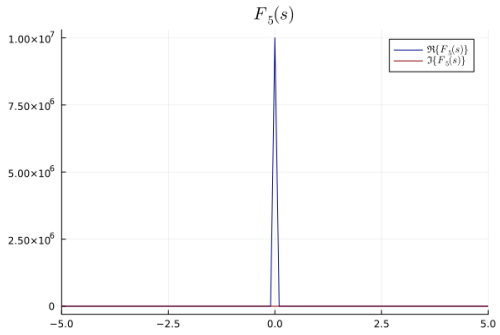
f_2



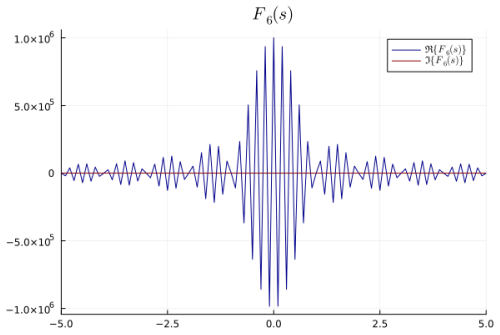
f_3



F_4



F_5



F_6

Ejercicios de Implementación

1. *a*) **[1 punto]** Considere la función $f(x)$ de la tarea anterior:

$$f(x) = \left(3 \cdot Gauss\left(\frac{x}{3}\right) - \sqcap\left(x - \frac{3}{2}\right) - \sqcap\left(x + \frac{3}{2}\right)\right) \cdot ramp\,(x+3)$$

Para el mismo intervalo de x ($x \in [-6, 6]$). Investigue sobre la función convolve de numpy para calcular:

$$g(x) = f(x) * \sqcap(2 \cdot x)$$

Luego grafique $g(x)$ para $x \in [-6, 6]$.

Consideración adiconales:

- Para el vector de x utilice un total de 12000 muestras.
- Para tener un resultado fidedigno con lo que se vería en el mundo continuo, pondere el resultado de la convolución por 0.001.
- Al usar la función convolve, utilice el modo *'same'* en la última entrada.

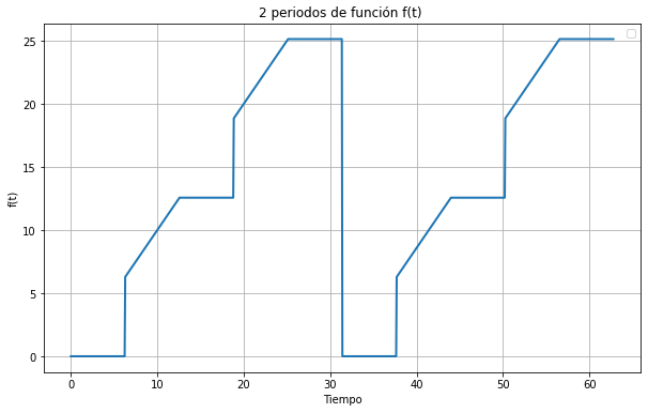
b) **[1 punto]** Repita el procedimiento para un rect de ancho $x = 1$, $x = 1.5$ y 2. Grafique las cuatro convoluciones realizadas en esta pregunta en una misma figura, y comente sobre el efecto de convolucionar una función con un rect. ¿Qué está pasando a medida que aumentamos el largo del rect?

2. Cree un código en Python en el que vaya siguiendo los pasos a continuación para poder obtener de manera numérica la serie de Fourier de una señal.

a) **[1 punto]** Cree usando la libreria de Python numpy la siguiente función a tramos. Considere que f(t) es una señal periódica de periodo 10π.

$$f(t) = \begin{cases} 0, & \text{si t} \in [0\pi : 2\pi] \\ t, & \text{si t } \in [2\pi : 4\pi] \\ 4\pi, & \text{si t} \in [4\pi : 6\pi] \\ t, & \text{si t} \in [6\pi : 8\pi] \\ 8\pi, & \text{si t} \in [8\pi : 10\pi] \end{cases}$$

A continuación, grafique f(t) por dos periodos usando matplotlib, es decir, de 0 a 20π. Su gráfica debería quedar como la siguiente:



Incluya fotos de su gráfica resultante así como del código usado.

b) **[2 puntos]** La siguiente parte tiene como finalidad que usted aprenda a integrar de forma numérica. A continuación se muestra un código de ejemplo que integra la siguiente función:

$$IntegralDeseada = \int_0^2 10t\, dt$$

```python
import numpy as np

num_muestras = 1000
periodo = 2
t = np.linspace(0, periodo, num_muestras, endpoint=False)
funcion = np.zeros_like(t)
funcion = 10*t

suma_acumulada = 0
for i in range (num_muestras):
    suma_acumulada += funcion[i]*periodo/num_muestras
print(suma_acumulada)

```

El anterior código realiza una aproximación numérica a la integral deseada. El resultado que se imprimirá con este código será de 19.8, pero sabemos que el

valor correcto es de 20. Vemos que en la línea 3 se define que la aproximación se realizará tomando 1000 muestras. Si este número se aumenta, el error cometido entre el valor obtenido y el valor real disminuirá.

Modifique el código anterior para que calcule la siguiente integral:

$$\int_0^{5\pi} t^2 \cdot \cos(t)\, dt$$

Muestre una foto del código escrito, y especifique el valor obtenido cuando el número de muestras es de 1000, 10000 y de 100000. ¿Qué puede decir al respecto?

c) **[3 puntos]** Ahora se le pide implementar un código que calcule la Serie de Fourier de la función f(t) del ítem a).

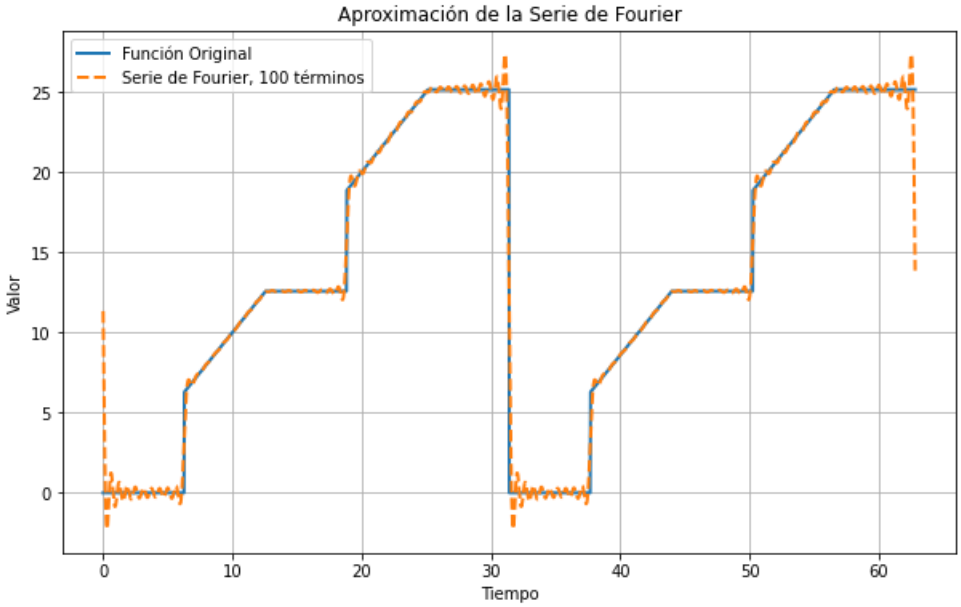
Para calcular los coeficientes tendrá que integrar numéricamente. Hágalo parecido a lo que aprendió a hacer en item b). Dentro de la integral de los coeficientes, se encuentra la función ponderada por una exponencial compleja. Para la función considere usar el código ya creado en item a), y para la exponencial compleja, la librería numpy puede serle útil para tratar con números complejos.

Se recomienda fuertemente usar la formulación compleja de la Serie de Fourier. De esta manera, usted sólo deberá calcular una constante (c_k) y no dos (a_k y b_k). Para efectos de la correción, se exigirá que cada vez que se pida truncar la serie al "k-ésimo"término, se consideren todos los coeficientes que hayan desde c_{-k} hasta c_k. Por ejemplo, si se pide encontrar la Serie de Fourier considerando los primeros 100 términos, el código debe calcular el valor de los 201 pesos que hay entre c_{-100} y c_{100}. Todos los demás pesos serán cero para llevar a cabo la aproximación.

Hint: Investigue cómo puede serle útil la información de que la función a aproximar sea puramente real. Además cuenta con la función np.real() que permite extraer la parte real de un número complejo. Si usted desarrolla correctamente la matemática se dará cuenta de que la combinación de ambas cosas le permitirá olvidarse de los c_k con k negativos, y así simplificar la serie.

1) En primer lugar, investigue qué es el fenómeno de Gibbs y descríbalo brevemente.

2) Se muestra como ejemplo a continuación una aproximación a la función f(t) considerando 1000 muestras por periodo, considerando los primeros 100 términos de la Serie de Fourier.



Cree su propio código y calcule con él la misma Serie de Fourier, pero para los primeros 5, 20 y 50 términos. Debe dejar fija la cantidad de muestras por periodo en 1000. Explique qué está pasando con el error entre la función aproximante y la función real. ¿Cómo se manifiesta el fenómeno de Gibbs en cada caso?, ¿En qué puntos del dominio se hace más notable?
Adjunte fotos de su código así como los gráficos que considere interesantes para su explicación.

3) Disminuya la cantidad de muestras por período a 50. Vuelva a graficar para 5, 20 y 50 términos de la serie. ¿Qué diferecias nota?
4) ¿Cuál cree usted que son las ventajas y desventajas de trabajar con 1000 muestras por periodo v/s trabajar con 50 muestras?

d) **[2 puntos]** **[Bonus]** Cree usted su propia función a tramos (como la desarrollada durante todo este ejercicio, pero una original) y calcule usando el mismo método su Serie de Fourier para 1000 muestras por periodo para los primeros 100 términos. Considere que el puntaje bonus que se le asignará dependerá de la creatividad que tenga a la hora de elegir la función que aproximará.