

Tarea 1: Entrando en calor...

En esta tarea usted deberá trabajar los fundamentos de GNU/Octave. Aunque puede utilizar Matlab para realizar la tarea, el profesor utilizará GNU/Octave para su revisión, por lo que debe asegurarse que el código implementado se ejecute correctamente en GNU/Octave.

Además, utilizará Git para registrar el avance, **paso a paso**, de la solución de la tarea. Para ello utilice [esta invitación a Github Classroom](#) para realizar el *fork* del repositorio base. Realice commits con bastante frecuencia y resuelva la tarea con tiempo.

15 pts

1. Familiarícese con los comandos `plot`, `stem`, `subplot`, `hold` y `figure` de GNU/Octave. Para ello investigue cómo cambiar colores y representaciones de los puntos en las curvas. El comando `help comando` le será de utilidad.
2. Utilice GNU/Octave para graficar las primeras 50 muestras de la función iniciando con $n = 0$

$$y[n] = \text{sen}(2\pi f n)$$

con $f = 0,01$, $f = 0,02$, $f = 0,05$ y $f = 0,1$. Investigue cómo graficar las 4 funciones:

- cada una en su propia ventana,
- todas en la misma figura, pero cada una por separado (la figura dividida en 4 secciones),
y
- las cuatro funciones traslapadas en una misma gráfica.

Indique cuál es la diferencia entre los comandos `stem` y `plot`. Indique cuántas muestras tiene el periodo de cada función.

Para este punto, cree un *script* `tarea2.m` de GNU/Octave que realice las figuras. En ese mismo script deberá llamar las funciones de los puntos subsiguientes.

10 pts

3. Edite un archivo de GNU/Octave (`.m`) para definir una función de nombre `frec4`, que recibe un argumento N y que muestra en una sola imagen las primeras N muestras de las cuatro funciones del punto anterior, cada una en su propia subfigura. Asegúrese de colocar los comentarios necesarios para que “`help frec4`” indique cómo utilizar su función, y además explique en ese comentario qué es lo que ella realiza.
4. Calcule el alias positivo de la frecuencia $f = 0,05$ más próximo a ella, y muestre gráficamente que ambas frecuencias son equivalentes.
5. Investigue cómo leer, salvar y reproducir en GNU/Octave archivos con señales de audio con los comandos `audioread`, `audioplayer`, `play`, `playblocking`, etc. Utilice el formato `.wav` de los archivos de audio.
6. Investigue cómo se puede controlar la frecuencia de reproducción de una señal almacenada en un vector.

10 pts

10 pts

7. Utilizando lo encontrado en el punto anterior, escriba una función `playtone` que reproduzca una señal senoidal de alguna frecuencia dada (por ejemplo 440 Hz para la nota “la”), muestreada utilizando una frecuencia de muestreo F_s , que debe tener algún valor por defecto. El tono debe sonar durante 1 s, a menos que el usuario indique un tercer parámetro, que debe ser la duración del tono. Así, `playtone(440)` toca un tono de 440 Hz durante un segundo, muestreado con alguna frecuencia de muestreo por defecto, o `playtone(880,11025,3.5)` toca un tono de 880 Hz muestreado a 11025 Hz durante 3,5 s. 15 pts

8. Realice una función `barrido` que reproduzca una señal en la que la frecuencia sube lentamente desde 0 hasta un número de veces la frecuencia de muestreo, dado como primer argumento. La frecuencia de muestreo debe ser el tercer argumento, con algún valor adecuado por defecto. El barrido debe tardar un número de segundos dado como segundo argumento, que por defecto debe ser 5 s. Así `barrido(3,6,44100)` debe recorrer las frecuencias desde 0 hasta 3×44100 Hz en 6 s. El objetivo de este punto es *escuchar* el efecto del aliasing. ¿Qué causa lo que se escucha? ¿Es el número de ciclos esperado? ¿Cómo se podría justificar lo escuchado? (responda con comentarios en su código) 20 pts

9. Realice una función `sobreponer` que sobreponga en una figura una señal senoidal en tiempo continuo con una frecuencia analógica dada, con las primeras N muestras tomadas con una frecuencia de muestreo dada, y que además sobreponga lo anterior con el k -ésimo alias senoidal. En otras palabras, la entrada a la función son cuatro argumentos: la frecuencia de la señal en tiempo continuo, el período de muestreo (o la frecuencia de muestreo si lo desea), el k para el alias y el número de muestras N . 10 pts

Toda tarea debe ser resultado del trabajo intelectual propio de la persona o personas que la entregan. Además de la literatura de referencia, solo puede utilizarse el material expresamente así indicado en la tarea, lo que incluye código brindado por el profesor o indicado en los enunciados a ser utilizado como base de la tarea. Expresamente quedan excluidos como material de referencia los trabajos entregados por otros estudiantes en el mismo semestre o semestres anteriores.

Nótese que esto no elimina la posibilidad de discutir estrategias de solución o ideas entre personas y grupos, lo cual es incluso recomendado, pero la generación concreta de cada solución, derivación o programa debe hacerse para cada entrega de forma independiente.

Para toda referencia de código o bibliografía externa deben respetarse los derechos de autor, indicando expresamente de dónde se tomó código, derivaciones, etc. Obsérvese que el código entregado por el profesor usualmente ya incluye encabezados con la autoría correspondiente. Si un estudiante agrega código a un archivo, debe agregar su nombre a los encabezados si la modificación es de más del 50 % del archivo, o indicar expresamente en el código, con comentarios claros, la autoría de las nuevas líneas de código, pues a la autora o al autor del archivo original no se le debe atribuir código que no es suyo.

Si se detecta código o deducciones teóricas iguales o muy cercanas a trabajos de otros estudiantes del mismo semestre o de semestres anteriores, se aplicará lo establecido por la reglamentación vigente, en particular el Artículo 75 del Reglamento de Régimen de Enseñanza y Aprendizaje.

Modificaciones de comentarios, cadenas alfanuméricas, nombres de variables, orden de estructuras independientes, y otras modificaciones menores de código se siguen considerando como clones de código, y las herramientas automatizadas de detección reportarán la similitud correspondiente.

Los estudiantes que provean a otros estudiantes del mismo o futuros semestres soluciones de sus tareas, también son sujetos a las sanciones especificadas en la reglamentación institucional. Por lo tanto, se advierte no poner a disposición soluciones de las tareas a otros estudiantes.