Apellidos:	**Nombre:**
Apellidos:	**Nombre:**
Apellidos:	**Nombre:**
Apellidos:	**Nombre:**

PRÁCTICA 2
CARACTERÍSTICAS Y PROPIEDADES DE LA DFT

- Al finalizar la práctica debes entregar las respuestas de este cuestionario en el formulario web que se proporcionará al principio de la sesión.
- Justifica adecuadamente tus cálculos y razonamientos.
- Usa al menos cuatro cifras significativas en tus operaciones

Ejercicio 2.1 La DFT y su relación con la DTFT

La DFT o transformada discreta de Fourier (Discrete Fourier Transform) es una operación definida sobre una señal en tiempo discreto cuyo objetivo es tomar muestras de la DTFT de dicha señal. Sin embargo, este proceso de muestreo en la frecuencia debe hacerse tomando ciertas precauciones, exactamente igual que cuando se hace en el dominio del tiempo. En este último caso sabemos que la frecuencia de muestreo debe tomar cierto valor mínimo para garantizar que la señal continua pueda recuperarse a partir de sus muestras. Del mismo modo, es necesario muestrear la DTFT por encima de cierto umbral para garantizar que la misma se pueda recuperar.

En este ejercicio vamos a realizar dos procesos diferentes: por un lado, calcularemos la DFT de una señal y luego trataremos de recuperar la señal original; por otro lado, tomaremos muestras de la DTFT y de nuevo trataremos de recuperar la señal original. El objetivo último es descubrir si ambos procesos son equivalentes y bajo qué condiciones.

Inicia Matlab y abre el programa `P2_1`. Este programa importa el fichero *Entrada.wav*, creando en tu espacio de trabajo una variable `Fs` correspondiente a la frecuencia de muestreo en Hz y un vector `x` correspondiente a los datos de sonido. A continuación, el programa `P2_1` toma la DFT de `N` puntos de `x` (a la que se denomina `X`) y luego la IDFT (DFT inversa) de `X`. Observa que los comandos correspondientes de Matlab son `fft` e `ifft`, respectivamente. El resultado se graba en un nuevo fichero de audio denominado *Salida_1.wav*.

Q2.1 Escoge un valor de `N` igual a la longitud de `x` y ejecuta el programa `P2_1`. Escucha el archivo *Salida_1.wav*. ¿Existe alguna diferencia con *Entrada.wav*? Si la hay, indica exactamente en qué consiste (incluyendo los posibles silencios). Te pueden ser útiles las representaciones de las señales en el tiempo.

no

Q2.2 Escoge un valor de `N` igual al doble de la longitud de `data` y ejecuta el programa `P2_1`. Escucha el archivo *Salida_1.wav*. ¿Existe alguna diferencia con *Entrada.wav*? Si la hay, indica exactamente en qué consiste (incluyendo los posibles silencios). Te pueden ser útiles las representaciones de las señales en el tiempo.

La diferencia es que esta vez la DFT toma muestras nulas, rellenando con ceros (silencio) hasta completar un audio con la duración que le hemos escogido. Es decir, si la señal original tiene N valores y muestreamos 2N valores, se crearán N ceros hasta llegar a 2N muestras.

Q2.3 Escoge un valor de `N` igual a la mitad de la longitud de `x` y ejecuta el programa `P2_1`. Escucha el archivo *Salida_1.wav*. ¿Existe alguna diferencia con *Entrada.wav*? Si la hay, indica exactamente en qué consiste (incluyendo los posibles silencios). Te pueden ser útiles las representaciones de las señales en el tiempo.

Al contrario que el último experimento, solo toma la mitad de las muestras que se podrían tomar de la señal original, esto se traduce en una duración menor (en este caso la mitad)

Abre el programa `P2_2`. Este programa toma `N` muestras de la DTFT de `x` (a la que se denomina `Xd`) y luego la IDFT de `Xd`. Los comandos correspondientes de Matlab son `freqz` e `ifft`, respectivamente. El resultado se graba en un nuevo fichero de audio llamado *Salida_2.wav*.

Q2.4 Escoge un valor de `N` igual a la longitud de `x` y ejecuta el programa `P2_2`. Escucha el archivo *Salida_2.wav*. ¿Existe alguna diferencia con *Entrada.wav*? Si la hay, indica exactamente en qué consiste (incluyendo los posibles silencios). Te pueden ser útiles las representaciones de las señales en el tiempo.

NO

Q2.5 Escoge un valor de `N` igual al doble de la longitud de `x` y ejecuta el programa `P2_2`. Escucha el archivo *Salida_2.wav*. ¿Existe alguna diferencia con *Entrada.wav*? Si la hay, indica exactamente en qué consiste (incluyendo los posibles silencios). Te pueden ser útiles las representaciones de las señales en el tiempo.

Igual que la anterior

Q2.6 Escoge un valor de `N` igual a la mitad de la longitud de `x` y ejecuta el programa `P2_2`. Escucha el archivo *Salida_2.wav*. ¿Existe alguna diferencia con *Entrada.wav*? Si la hay, indica qué se escucha exactamente (qué vocales, si hay silencios, etc.). Te pueden ser útiles las representaciones de las señales en el tiempo.

Solapamiento

Q2.7 Como es sabido, un muestreo inadecuado en el dominio del tiempo induce *aliasing* en el dominio de la frecuencia. De igual modo un muestreo insuficiente en la frecuencia produce solapamiento en el tiempo. Eso es lo que está ocurriendo en la cuestión anterior. ¿Qué segmentos temporales de la secuencia original `x` son los que están solapando entre sí exactamente?

La segunda mitad de la señal x se está solapando.

Q2.8 A partir de los resultados de las cuestiones anteriores, ¿qué condición será necesaria para que una DFT coincida con el muestreo de la DTFT? Si se da ese caso, ¿es posible recuperar la señal de audio original? (Esto implicaría que la señal se ha representado adecuadamente en el dominio de la frecuencia). ¿Dicha recuperación es posible si la condición no se cumple?

La condición para que coincida una DFT y una DTFT es que la longitud del muestreo N sea >= que la longitud de la señal original. Si es menor que N, se comportan de forma distinta: La DFT anula la segunda mitad, y la dftf lo solapa. Si esta condición se cumple (N>= la longitud de x(n))

Ejercicio 2.2 Propiedades de la DFT

Abre el programa `P2_3`. Este programa se puede utilizar para verificar la propiedad de *desplazamiento circular en el tiempo* de la DFT. Como vamos a comprobar, un desplazamiento circular se diferencia del habitual en que las muestras desplazadas hacia un lado se insertan por el lado contrario.

Q2.9 Ejecuta el programa `P2_3`, en el cual se efectúa un desplazamiento de `m` = 2 muestras. Escribe cuáles son la señal original y la señal desplazada circularmente (en forma de vector, no hace falta que las dibujes). ¿El desplazamiento circular de 2 muestras ha sido hacia la derecha o hacia la izquierda?

Se desplaza hacia la derecha. La señal desplazada quedaría: {7,8,0,1,2,3,5,6}

Q2.10 Modifica el programa `P2_3` para que el nuevo desplazamiento temporal sea –2. Ejecútalo y escribe cuál es la nueva señal desplazada (en forma de vector, no hace falta que la dibujes).

Se desplaza hacia la izquierda. La señal desplazada quedaría: {2,3,5,6,7,8,0,1}

Q2.11 Modifica el programa `P2_3` para que la señal a desplazar circularmente sean tus muestras de audio contenidas en `A.data` y para que grabe el resultado en un nuevo fichero de audio llamado *Salida_3.wav*. (Fíjate en cómo está hecho en el programa `P2_1`). Toma un valor del desplazamiento igual a 35000 muestras. Ejecuta el programa y escucha el resultado. Indica qué se escucha exactamente y justifícalo. Te pueden ser útiles las representaciones de las señales en el tiempo.

Se Se escucha el audio original empezando por la o, y cuando terminaría el original, vuelve a empezar hasta justo antes de la o. Lo que significa que ha habido un desplazamiento circular, en el que se ha atrasado el audio 35000 muestras, (justo la duración de la o y la u).

Abre el programa `P2_4`. Este programa se puede utilizar para verificar la propiedad de *inversión circular* de la DFT. Como vamos a ver, en este caso la inversión no implica una reflexión respecto del eje vertical.

Q2.12 Ejecuta el programa `P2_4`. Escribe cuáles son la señal original y la señal invertida (en forma de vector, no hace falta que las dibujes). Todas las muestras han seguido cierta regla general excepto una. ¿Cuál es esa regla general? ¿Cuál es la posición de la muestra que no la sigue?

Sabiendo que la señal invertida es x(-n) empieza en x(-0) que es igual que x(0).
Luego x(-1) es x(8) y así hasta x(-8) que es x(1).
Entonces la que no sigue la regla es x(-0), la primera muestra ya que invertida o no siempre será la misma al estar en cero.

Q2.13 Modifica el programa `P2_4` para que la señal a invertir sean tus muestras de audio contenidas en `A.data` y para que grabe el resultado en un nuevo fichero de audio llamado *Salida_4.wav*. (Fíjate en cómo está hecho en el programa `P2_1`). Ejecuta el programa y escucha el resultado. Indica qué se escucha exactamente y justifícalo. Te pueden ser útiles las representaciones de las señales en el tiempo.

Se escucha como si fuera el audio original escuchado del final hacia el principio. Esto es así, porque está sucediendo el mismo fenómeno que en el caso anterior. Y la primera muestra (x(0)) no se aprecia que no cambia porque es un silencio y es una muestra de 84986 que tiene el audio.

Abre ahora el programa P2_5. Este programa se puede utilizar para verificar la propiedad de *convolución circular* de la DFT. No vamos a analizar con detalle en qué consiste la convolución circular pero sí que nos interesa saber en qué casos es equivalente a la convolución "normal" (que en este contexto es llamada *convolución lineal*).

Lo que hace este programa es pasar la señal de entrada x a través de un filtro h. En primer lugar se calcula la salida del filtro mediante una convolución lineal directa en el dominio del tiempo. A continuación, se calcula la salida indirectamente a través del dominio de la frecuencia (tomando DFTs e IDFTs de longitud N), pero de este modo la convolución resultante es circular, no lineal. Los resultados se graban en sendos ficheros de audio denominados *Salida_lin.wav* y *Salida_cir.wav*, respectivamente.

Q2.14 ¿Cuál es el valor de N elegido inicialmente? ¿Es este valor adecuado para representar las señales x y h (por separado) en el dominio de la frecuencia? Justifícalo numéricamente de acuerdo a tu resultado de la cuestión Q2.8.

El N inicial es el número de muestras del audio. Si lo es, pq como seha dicho en el 2.8, para q dftf y dft sean equivalentes, N= latgo de señal.

Q2.15 Ejecuta el programa P2_5. ¿Cuál es la longitud de la convolución lineal, calculada en la variable y_lin? ¿Qué relación existe entre dicha longitud y la de las secuencias x y h?

124986. Es mayor que la longitud de x y h.

Q2.16 ¿Cuál es la longitud de la convolución circular, calculada en la variable y_cir? ¿Qué relación existe entre dicha longitud y el número de muestras que hemos tomado para las DFTs?

84986. Es la misma longitud.

Q2.17 ¿Son iguales la convolución circular y la lineal? Puedes comprobarlo observando las figuras y/o escuchando los archivos *Salida_lin.wav* y *Salida_cir.wav*. (No te preocupes de por qué ni cómo el filtro ha dado lugar a esta salida lineal concretamente, solo si es igual a la circular).

No son iguales.

Q2.18 A continuación, aumenta N en incrementos de 10000. Detente cuando *Salida_cir.wav* sea idéntico a *Salida_lin.wav* (puedes comprobarlo en la figura y/o en los audios). ¿Cuál es el valor de N para este caso?

124986

Q2.19 Sigue aumentando N en incrementos de 10000 un poco más. ¿Qué sucede ahora?

La convolución circular aumenta de muestras, pero solo con muestras nulas.

Q2.20 De acuerdo a los resultados anteriores, ¿qué condición será necesaria para que pueda recuperarse la convolución lineal a partir de la circular? ¿Es este resultado coherente con el de la cuestión Q2.8?

Que el numero de muestras de la conv lineal sea igual al numero de muestras de la conv circ.

Si es coherente, porque tanto en los dos procesos del 2.8 como en estas dos convoluciones, si el numero de muestras del proceso circular es menor que el otro, se produce aliasing. En cambio, la condición mínima en ambas, es que ambos procesos tengan la misma cantidad de muestras.