**Instrucciones para el llenado de práctica**

1. La fecha de entrega es antes de las 9 horas del día 31 de Octubre
2. La entrega es vía correo electrónico a la dirección: [**ana\_cris814@hotmail.com**](mailto:ana_cris814@hotmail.com)
3. El asunto del correo debe escribir**: PROCESAMIENTO DE SEÑALES \_PRACTICA(No.Practica)**
4. La entrega de la práctica debe ser en **formato PDF**
5. En el **archivo PDF** no se debe incluir este instructivo
6. Debe incluir el **número de lista y nombre completo de cada integrante**.
7. Dentro del desarrollo de las prácticas se deben **incluir el código y los gráficos** obtenidos.
8. En la práctica por equipo se debe incluir **conclusión por participante.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Respuestas del sistema | | | | | |
| No. de Practica: | 6 | **Fecha de entrega:** | | **13 de octubre de 2023** | |
| Integrantes de equipo:  19 NAVIL PINEDA RUGERIO | | **Practica:** | **Individual** | | **En equipo**  **2 personas** |
| OBJETIVO: | | | | | |
| MATERIAL Y EQUIPO: **Computadora con MATLAB** | | | | | |
| Desarrollo: | | | | | |
| Paso 1. Calcula la Transformada Discreta de Fourier (DFT) de cada una de las siguientes secuencias x[n]      Para calcular la Transformada Discreta de Fourier, se utiliza la función de Matlab “fft”, que recibe como parámetros una secuencia x[n], por ello cada una de las secuencias anteriores se escriben como vectores:  x1 = [1,-1];  x2 = [1,1,-1,-1];  x3 = [3,-1,4,2];  x4 = [1,0,0,-1,0,0];  Y se grafican las señales.  subplot(4,1,1)  stem(x1)  title("Señal " + num2str(1) + " en el dominio del tiempo")  subplot(4,1,2)  stem(x2)  title("Señal " + num2str(2) + " en el dominio del tiempo")  subplot(4,1,3)  stem(x3)  title("Señal " + num2str(3) + " en el dominio del tiempo")  subplot(4,1,4)  stem(x4)  title("Señal " + num2str(4) + " en el dominio del tiempo")    Después, se calcula su transformada con la función de Matlab, y por cada transformada se calcula su magnitud, aplicando el valor absoluto al resultado de la transformada:  % Señal 1---------------------------------------------------------------  y1 = fft(x1);  m1 = abs(y1); % magnitud  % Señal 2---------------------------------------------------------------  y2 = fft(x2);  m2 = abs(y2); % magnitud  % Señal 3---------------------------------------------------------------  y3 = fft(x3);  m3 = abs(y3); % magnitud  % Señal 4---------------------------------------------------------------  y4 = fft(x4);  m4 = abs(y4); % magnitud  Y su fase, para ello se eliminan las frecuencias muy pequeñas, que pueden considerarse ruido, están se transforman en 0, aquellas que son menores a 1e-6, después se calcula el ángulo de la transformada en radianes.  % Señal 1---------------------------------------------------------------  y1(m1<1e-6) = 0;  p1 = unwrap(angle(y1)); % fase  % Señal 2---------------------------------------------------------------  y2(m2<1e-6) = 0;  p2 = unwrap(angle(y2)); % fase  % Señal 3---------------------------------------------------------------  y3(m3<1e-6) = 0;  p3 = unwrap(angle(y3)); % fase  % Señal 4---------------------------------------------------------------  y4(m4<1e-6) = 0;  p4 = unwrap(angle(y4)); % fase  Y se genera un vector de la nueva longitud de “muestras”, que serán las frecuencias resultantes de la transformada, esto hace que el vector de muestras en el tiempo se ajuste al vector de muestras en la frecuencia y se pueda graficar.  f1 = (0:length(y1)-1)\*100/length(y1);  f2 = (0:length(y2)-1)\*100/length(y2);  f3 = (0:length(y3)-1)\*100/length(y3);  f4 = (0:length(y4)-1)\*100/length(y4);  Se grafican los vectores de magnitud de todas las señales.  % Señal 1---------------------------------------------------------------  subplot(4,1,1)  plot(f1,m1)  title("señal " + num2str(1) + " en el dominio de la frecuencia: magnitud")  % Señal 2---------------------------------------------------------------  subplot(4,1,2)  plot(f2,m2)  title("señal " + num2str(2) + " en el dominio de la frecuencia: magnitud")  % Señal 3---------------------------------------------------------------  subplot(4,1,3)  plot(f3,m3)  title("señal " + num2str(3) + " en el dominio de la frecuencia: magnitud")  % Señal 4---------------------------------------------------------------  subplot(4,1,4)  plot(f4,m4)  title("señal " + num2str(4) + " en el dominio de la frecuencia: magnitud")    Y se grafican sus vectores de fase.  % Señal 1---------------------------------------------------------------  subplot(4,1,1)  plot(f1,p1\*180/pi)  title("señal " + num2str(1) + " en el dominio de la frecuencia: fase")  % Señal 2---------------------------------------------------------------  subplot(4,1,2)  plot(f2,p2\*180/pi)  title("señal " + num2str(2) + " en el dominio de la frecuencia: fase")  % Señal 3---------------------------------------------------------------  subplot(4,1,3)  plot(f3,p3\*180/pi)  title("señal " + num2str(3) + " en el dominio de la frecuencia: fase")  % Señal 4---------------------------------------------------------------  subplot(4,1,4)  plot(f4,p4\*180/pi)  title("señal " + num2str(4) + " en el dominio de la frecuencia: fase")    Paso 2. Dada la secuencia de longitud 𝑁=4, definida por para 𝑛= 0,1,2,3 calcula su Transformada Discreta de Fourier (DFT).  Primero se escribe el vector de muestras para evaluar la función:  % Vector de tiempo  t = [0,1,2,3];  Y se escribe la señal coseno:  x5 = cos(pi\*t);  Se grafica utilizando la función stem:  subplot(3,1,1)  stem(t,x5)  title("Señal " + num2str(5) + " en el dominio del tiempo")    Al igual que el paso anterior se calcula su transformada utilizando la función de matlab, que también puede recibir una función, como la definida anteriormente, en vez de una secuencia.  y5 = fft(x5);  Asimismo, se calcula su magnitud y su fase tal como en el paso anterior.  m5 = abs(y5); % magnitud  y5(m5<1e-6) = 0;  p5 = unwrap(angle(y5)); % fase  f5 = (0:length(y5)-1)\*100/length(y5);  Y las graficamos:  subplot(3,1,2)  plot(f5,m5)  title('magnitud de la señal coseno')  ax = gca;  ax.xtick = [15 40 60 85];  subplot(3,1,3)  plot(f,p\*180/pi)  title('fase de la señal coseno')  ax = gca;  ax.xtick = [15 40 60 85];    Paso 3. Calcula la Transformada Discreta Inversa de Fourier (IDFT) de las secuencias:      Ahora se realiza el proceso inverso, utilizando la función de Matlab “ifft” para calcular la transformada de Fourier Inversa  Paso 3. Complete la definición de la función *dtft* que acepta las siguientes entradas:  f 🡪 una función anónima de tiempo continuo que se va a muestrear  T 🡪 un período de muestreo  N 🡪 el número de muestras que se incluirán a cada lado de 0 en la suma DTFT (es decir, las muestras realmente están involucradas en la suma)  W 🡪 un vector de frecuencias en las que muestrear el DTFT  y devuelve la salida:  F 🡪 un vector complejo correspondiente a la DTFT muestreada en los puntos en w. | | | | | |
| Conclusión: | | | | | |
|  | | | | | |
| Fecha final de entrega: lunes, 30 de octubre de 2023 | | | | | |