Ayudantía 7 - Procesamiento Digital de Señales

1. Sea la secuencia $x[n] = \{1\ 1\ 0\ 3\}$

   (a) Calcule la TFD $X[k]$ utilizando la matriz $\boldsymbol{W_4}$.

Solución: Recordemos que la TFD se define por:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \tag{1}$$

Y definiendo $W_N = e^{-j\frac{2\pi}{N}}$:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \tag{2}$$

Esta suma se puede reescribir como un producto entre una matriz $\boldsymbol{W_N}$ y un vector $\boldsymbol{x}$. De forma general:

$$\boldsymbol{X} = \boldsymbol{W_N x} \tag{3}$$

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & W_N^1 & W_N^2 & \dots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & \dots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \dots & W_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix} \tag{4}$$

Para la matriz $\boldsymbol{W_N}$:

- Tiene dimensiones $N \times N$.
- La primera fila y columna son solo unos.
- Para la fila k, los exponentes de W_N aumentan de $k-1$ en $k-1$. Ej: La segunda fila los exponentes van de uno en uno, para la tercera fila van de dos en dos, etc...
- Es simétrica ($\boldsymbol{W_N} = \boldsymbol{W_N^T}$)

En el caso de la secuencia $x[n] = \{1\ 1\ 0\ 3\}$ ($N = 4$) la transformada es:

$$X[k] = \sum_{n=0}^{3} x[n] W_4^{kn} \tag{5}$$

y en forma matricial:

$$\boldsymbol{X} = \boldsymbol{W_4 x} \tag{6}$$

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & W_4^3 \\ 1 & W_4^2 & W_4^4 & W_4^6 \\ 1 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix} \tag{7}$$

Como $W_N^n = W_N^{n+N}$, entonces $W_4^4 = W_4^0$, $W_4^6 = W_4^2$ y $W_4^9 = W_4^1$ se escribe:

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & W_4^3 \\ 1 & W_4^2 & W_4^0 & W_4^2 \\ 1 & W_4^3 & W_4^2 & W_4^1 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix} \tag{8}$$

Notando que $W_4^0 = 1$, $W_4^1 = e^{-j\frac{2\pi}{4}} = -j$, $W_4^2 = -1$ y $W_4^3 = j$, y reemplazando por los valores de $x[n]$:

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 3 \end{bmatrix} = \begin{bmatrix} 1+1+0+3 \\ 1-j+0+3j \\ 1-1+0-3 \\ 1+j+0-3j \end{bmatrix} = \begin{bmatrix} 5 \\ 1+2j \\ -3 \\ 1-2j \end{bmatrix} \tag{9}$$

Es decir: $X[k] = \{5 \quad 1+2j \quad -3 \quad 1-2j\}$

(b) Determine la la TFD inversa $z[n]$ de $W_4^{2k}X[k]$. Grafique $z[n]$ y $x[n]$ para su comparación.

Solución: Para el cálculo de la transformada inversa de $W_4^{2k}X[k] = e^{-j\frac{2\pi}{4}2k}X[k]$ debemos tener en cuenta la propiedad de desplazamiento circular. Sea una señal $x[n]$ y $X[k]$ su TFD de N puntos:

$$\boxed{x[\langle n-m\rangle_N] \overset{TFD}{\longleftrightarrow} W_N^{mk}X[k] = e^{-j\frac{2\pi}{N}mk}X[k]} \tag{10}$$

Con $\langle * \rangle_N$ la operación módulo N. Notar que la exponencial debe estar en forma $e^{-j\frac{2\pi}{N}mk}$.

Al aplicar la propiedad en nuestro caso ($N = 4$, $m = 2$):

$$W_4^{2k}X[k] \xrightarrow{TFDI} z[n] = x[\langle n-2\rangle_4] \tag{11}$$

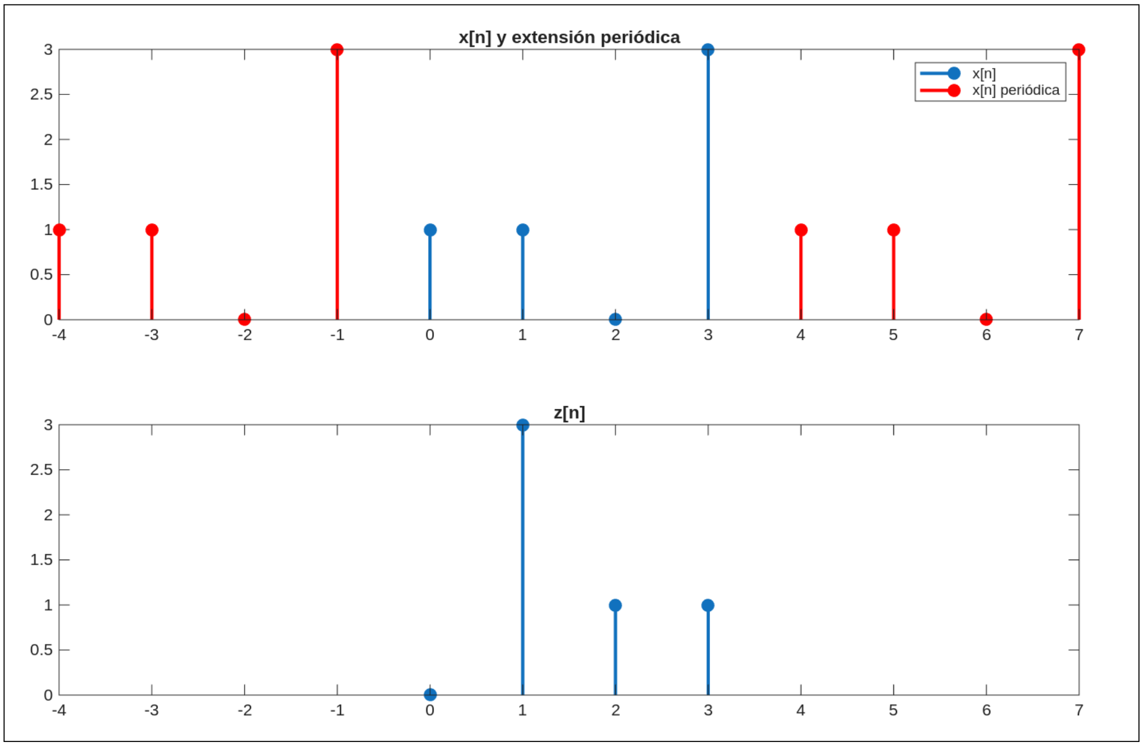
Para encontrar los valores de la señal para $n = 0,\ 1,\ 2,\ 3$ se retarda la señal en $m = 2$ rotando los valores que se salen del rango $0 \leq n \leq 3 = N - 1$:

$$x[n] = \{1\ 1\ 0\ 3\} \tag{12}$$

$$\downarrow$$

$$z[n] = x[\langle n-2\rangle_4] = \{0\ 3\ 1\ 1\} \tag{13}$$

Al graficar $x[n]$, su extensión periódica y $z[n]$ vemos como la TFD no "ve" la señal $x[n]$ sino su extensión periódica al aplicar la propiedad de desplazamiento:



(c) Realice la convolución circular $x[n] \circledast h[n]$, con $h[n] = \{1\ 0\ 1\ 0\}$ en el plano temporal.

Solución: La convolución circular se define por:

$$y[n] = x[n] \circledast h[n] = \sum_{m=0}^{N-1} h[n]x[\langle n-m\rangle_N] \tag{14}$$

$$= \sum_{m=0}^{3} h[n]x[\langle n-m\rangle_4] \tag{15}$$

Lo que se puede expresar en forma matricial:

$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \end{bmatrix} = \begin{bmatrix} x[0] & x[3] & x[2] & x[1] \\ x[1] & x[0] & x[3] & x[2] \\ x[2] & x[1] & x[0] & x[3] \\ x[3] & x[2] & x[1] & x[0] \end{bmatrix} \begin{bmatrix} h[0] \\ h[1] \\ h[2] \\ h[3] \end{bmatrix} \tag{16}$$

$$= \begin{bmatrix} 1 & 3 & 0 & 1 \\ 1 & 1 & 3 & 0 \\ 0 & 1 & 1 & 3 \\ 3 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1+3 \\ 1 \\ 3+1 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 1 \\ 4 \end{bmatrix} \tag{17}$$

Por lo tanto: $y[n] = \{1\ 4\ 1\ 4\}$

(d) Repita (c) usando las TFD $X[k]$ y $H[k]$.

Solución: Del ejercicio 1(a): $X[k] = \{5 \quad 1+2j \quad -3 \quad 1-2j\}$. $H[k]$ se puede obtener de forma similar con la matriz $\boldsymbol{W}_4$ y un vector $\boldsymbol{h}$:

$$\boldsymbol{H} = \boldsymbol{W}_4\boldsymbol{h} \tag{18}$$

$$\begin{bmatrix} H[0] \\ H[1] \\ H[2] \\ H[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} h[0] \\ h[1] \\ h[2] \\ h[3] \end{bmatrix} \tag{19}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1+1 \\ 1-1 \\ 1+1 \\ 1-1 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 2 \\ 0 \end{bmatrix} \tag{20}$$

Es decir: $H[k] = \{2\ 0\ 2\ 0\}$.

Luego aplicamos la propiedad que relaciona la convolución cíclica con la TFD:

$$\boxed{y[n] = x[n] \circledast h[n] \overset{TFD}{\longleftrightarrow} Y[k] = X[k]H[k]} \tag{21}$$

Es decir, multiplicando las transformadas y aplicando la TFDI a $Y[k]$ obtenemos $y[n]$. La transformada $Y[k]$ es:

$$\begin{aligned} X[k] &= \{5 \quad 1+2j \quad -3 \quad 1-2j\} \\ H[k] &= \{2 \quad 0 \quad 2 \quad 0\} \\ Y[k] = X[k]H[k] &= \{10 \quad 0 \quad -6 \quad 0\} \end{aligned} \tag{22}$$

Aplicamos la transformada inversa. Matricialmente:

$$\boldsymbol{y} = \boldsymbol{W}_4^{-1}\boldsymbol{Y} \tag{23}$$

Recordando que en general $\boldsymbol{W}_{\boldsymbol{N}}^{-1} = \frac{1}{N}\boldsymbol{W}_{\boldsymbol{N}}^{*}$, tenemos:

$$
\begin{aligned}
\boldsymbol{y} &= \boldsymbol{W}_4^{-1}\boldsymbol{Y} && (24)\\
&= \frac{1}{4}\boldsymbol{W}_4^{*}\boldsymbol{Y} && (25)\\
\begin{bmatrix} y[0]\\ y[1]\\ y[2]\\ y[3] \end{bmatrix} &= \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1\\ 1 & -j & -1 & j\\ 1 & -1 & 1 & -1\\ 1 & j & -1 & -j \end{bmatrix}^{*}\begin{bmatrix} Y[0]\\ Y[1]\\ Y[2]\\ Y[3] \end{bmatrix} && (26)\\
&= \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1\\ 1 & j & -1 & -j\\ 1 & -1 & 1 & -1\\ 1 & -j & -1 & j \end{bmatrix}\begin{bmatrix} 10\\ 0\\ -6\\ 0 \end{bmatrix} && (27)\\
&= \frac{1}{4}\begin{bmatrix} 10-6\\ 10+6\\ 10-6\\ 10+6 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 4\\ 16\\ 4\\ 16 \end{bmatrix} = \begin{bmatrix} 1\\ 4\\ 1\\ 4 \end{bmatrix} && (28)
\end{aligned}
$$

Es decir: $y[n] = \{1\ 4\ 1\ 4\}$. Coincide con la convolución en el tiempo.

2. Muestre brevemente como se realizaría el calculo de la TFD de $x[n] = \{0\ \ 1\ \ 2\ \ 3\}$ usando el algoritmo de Transformada Rápida de Fourier visto en clases.

Solución: Un algoritmo de Transformada Rápida de Fourier divide la señal en el tiempo en muestras pares e impares ($\boldsymbol{x_p}$ y $\boldsymbol{x_i}$), aplica la TFD (es recursivo!) a cada una y las vuelve a unir de acuerdo al siguiente esquema:

$$x[n] \rightarrow \overset{FFT}{\boxed{ \boldsymbol{x} = \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ \vdots \\ x[N-1] \end{bmatrix} \begin{array}{c} \nearrow \\ \\ \\ \searrow \end{array} \begin{array}{c} \boldsymbol{x_p} = \begin{bmatrix} x[0] \\ x[2] \\ x[4] \\ \vdots \end{bmatrix} \xrightarrow{FFT} \boldsymbol{X_p} \\ \\ \boldsymbol{x_i} = \begin{bmatrix} x[1] \\ x[3] \\ x[6] \\ \vdots \end{bmatrix} \xrightarrow{FFT} \boldsymbol{X_i} \end{array} \begin{array}{c} \searrow \\ \\ \\ \nearrow \end{array} \begin{array}{c} \boldsymbol{X_p} + \boldsymbol{D_N X_i} = \boldsymbol{X_T} \longrightarrow \\ \boldsymbol{X_p} - \boldsymbol{D_N X_i} = \boldsymbol{X_B} \longrightarrow \end{array} \begin{bmatrix} \boldsymbol{X_T} \\ \boldsymbol{X_B} \end{bmatrix} = \boldsymbol{X} }} \rightarrow X[k] \tag{29}$$

Con la matriz $\boldsymbol{D_N}$ dada por:

$$\boldsymbol{D_N} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & W_N^1 & 0 & \dots & 0 \\ 0 & 0 & W_N^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & W_N^{\frac{N}{2}-1} \end{bmatrix} \tag{30}$$

Al aplicar recursivamente el algoritmo las secuencias $\boldsymbol{x_p}$ y $\boldsymbol{x_i}$ se vuelven a dividir, cada una en dos sub-secuencias, hasta llegar a una secuencia de largo 1. Al aplicar la TFD a una secuencia de largo 1, se obtiene la misma secuencia, y se comienza a construir la TFD.

En el caso de la señal $x[n] = \{0 \ 1 \ 2 \ 3\}$ $(N = 4)$, al aplicar recursivamente el algoritmo, y notando que $\boldsymbol{D_2} = [1]$ y $\boldsymbol{D_4} = \begin{bmatrix} 1 & 0 \\ 0 & W_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -j \end{bmatrix}$:

$$
\begin{bmatrix}x[0]\\x[1]\\x[2]\\x[3]\end{bmatrix} = \begin{bmatrix}0\\1\\2\\3\end{bmatrix}
\begin{array}{l}
\nearrow \begin{bmatrix}0\\2\end{bmatrix} \begin{array}{l} \nearrow [0] \xrightarrow{TFD} [0] \searrow \\ \searrow [2] \xrightarrow{TFD} [2] \nearrow \end{array} \begin{bmatrix}0+(1)2\\0-(1)2\end{bmatrix} = \begin{bmatrix}2\\-2\end{bmatrix} \searrow \\
\\
\searrow \begin{bmatrix}1\\3\end{bmatrix} \begin{array}{l} \nearrow [1] \xrightarrow{TFD} [1] \searrow \\ \searrow [3] \xrightarrow{TFD} [3] \nearrow \end{array} \begin{bmatrix}1+(1)3\\1-(1)3\end{bmatrix} = \begin{bmatrix}4\\-2\end{bmatrix} \nearrow
\end{array}
\begin{array}{l}
\boldsymbol{X}_T = \begin{bmatrix}2\\-2\end{bmatrix} + \begin{bmatrix}1 & 0\\0 & -j\end{bmatrix} \begin{bmatrix}4\\-2\end{bmatrix} \\
\\
\boldsymbol{X}_B = \begin{bmatrix}2\\-2\end{bmatrix} - \begin{bmatrix}1 & 0\\0 & -j\end{bmatrix} \begin{bmatrix}4\\-2\end{bmatrix}
\end{array}
\tag{31}
$$

$$
\begin{array}{l}
\boldsymbol{X}_T = \begin{bmatrix}2\\-2\end{bmatrix} + \begin{bmatrix}4\\2j\end{bmatrix} = \begin{bmatrix}6\\-2+2j\end{bmatrix} \searrow \\
\\
\boldsymbol{X}_B = \begin{bmatrix}2\\-2\end{bmatrix} - \begin{bmatrix}4\\2j\end{bmatrix} = \begin{bmatrix}-2\\-2-2j\end{bmatrix} \nearrow
\end{array}
\begin{bmatrix}\boldsymbol{X}_T\\ \boldsymbol{X}_B\end{bmatrix} = \begin{bmatrix}6\\-2+2j\\-2\\-2-2j\end{bmatrix} = \begin{bmatrix}X[0]\\X[1]\\X[2]\\X[3]\end{bmatrix}
\tag{32}
$$

3. Implemente en Matlab el algoritmo de Transformada Rápida de Fourier en Matlab. Puede asumir que el vector de entrada es de $N = 2^m$ una potencia de dos.
   (a) Compruebe la validez calculando la TFD $x[n]$ de la pregunta 1.

Solución: En la pag. 439 del libro se encuentra una implementación del algoritmo recién visto. Sin embargo la implementación del libro tiene dos errores. Arreglando los errores se obtiene:

```
function Xdft = fftrecur(x)
% Recursive computation of the DFT using divide & conquer
% N should be a power of 2
N = length(x);
if N == 1
    Xdft = x;
else
    m = N/2;
    Xp = fftrecur(x(1:2:N));
    Xi = fftrecur(x(2:2:N));
    W = exp(-2j*pi/N).^((0:m-1)'); % <- Aquí había un error
    temp = W.*Xi;
    Xdft = [ Xp+temp ; Xp-temp ]; % <- Aquí había otro
end
end
```

Y se comprueba que la implementación funciona.

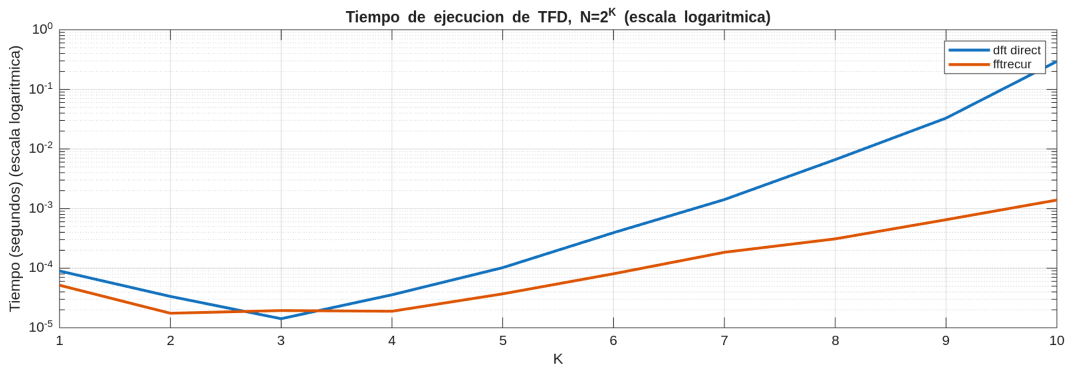
```
x = [1 1 0 3];
X = fftrecur(x)

X =

   5.0000 + 0.0000i
   1.0000 + 2.0000i
  -3.0000 + 0.0000i
   1.0000 - 2.0000i
```

(b) Considere la siguiente implementación directa (por definición) de la TFD:

```
function Xdft=dftdirect(x)
% Direct computation of the DFT
N=length(x); Q=2*pi/N;
for k=1:N
    S=0;
    for n=1:N
        W(k,n)=exp(-1j*Q*(k-1)*(n-1));
        S=S+W(k,n)*x(n);
    end
    Xdft(k)=S;
end
end
```

Usando las funciones `tic` y `toc` de Matlab, compare el rendimiento de su implementación y la de `dftdirect`. Para la generación del vector de entrada de prueba puede utilizar la función `rand`.

Solución: Al comparar el tiempo que demora en ejecutarse un algoritmo de TFD por definición, y otro de Transformada Rápida de Fourier hay una diferencia notoria:




```
%% 3 (b)

K = 10; % N=2^1 hasta 2^10
reps = 50; % Cada prueba se repite 10 veces y se promedia

k_vec = 1:K;
tiempos_direct = zeros(1,K);
tiempos_fft = zeros(1,K);

for k = k_vec
    for r = 1:reps
        x = rand(1,2^k);

        tic;
        dftdirect(x);
        tiempos_direct(k) = tiempos_direct(k) + toc;

        tic;
        fftrecur(x);
        tiempos_fft(k) = tiempos_fft(k) + toc;
    end
end

figure
subplot(2,1,1)
plot(k_vec, tiempos_direct./reps, 'LineWidth', 2); hold on
```

```
plot(k_vec, tiempos_fft./reps, 'LineWidth', 2)
legend("dft direct", "fftrecur")
xlabel("K")
ylabel("Tiempo (segundos)")
title("Tiempo de ejecucion de TFD, N=2^K")
subplot(2,1,2)
semilogy(k_vec, tiempos_direct./reps, 'LineWidth', 2); hold on
semilogy(k_vec, tiempos_fft./reps, 'LineWidth', 2)
legend("dft direct", "fftrecur")
xlabel("K")
ylabel("Tiempo (segundos) (escala logaritmica)")
title("Tiempo de ejecucion de TFD, N=2^K (escala logaritmica)")
```