Ampliación de Señales y Sistemas

Tema 4:

Fundamentos de la Transforma Discreta de Fourier

La presentación contiene material del OCW del MIT (6.003 Signals and Systems S10, F11), así como material de A. Guerrero, A. Wilsky y Antonio García Marqués.

Ubicándonos

- Tema 1: Señales y sistemas discretos en el dominio del tiempo
- Tema 2: Señales y sistemas discretos en el dominio de la frecuencia
- Tema 3: Muestreo
- **Tema 4: Fundamentos de la Transformada Discreta de Fourier**
  - 4.1 Definición: la DFT como el muestreo de la Transformada de Fourier
  - 4.2 Propiedades
  - 4.3 Convolución circular: definición y relación con la DFT
  - 4.4 La DFT en Matlab
- Tema 5: Transformada Z
- Tema 6: Introducción al diseño de filtros discretos

- Comentarios:
  - Tema muy importante (el más difícil de entender)
  - Resumen: los ordenadores sólo pueden calcular DFTs, la DFT puede interpretarse como el muestreo de la TF (no siempre la TF de la señal original, sino la de una versión enventanada de la señal original)
  - Trabajo previo: relación entre la TF de una SD y la TF de un segmento de esa SD

Ubicándonos

- **<u>Tema 4</u>: Fundamentos de la Transformada Discreta de Fourier**
  - **<u>4.1 Definición: la DFT como el muestreo de la TF</u>**
    - 4.1.0 Introducción
    - 4.1.1 Definición y ejemplos
    - 4.1.2 La DFT como el muestreo de la TF
    - 4.1.3 Problemas y aspectos prácticos
  - 4.2 Propiedades
  - 4.3 Convolución circular: definición y relación con la DFT
  - 4.4 La DFT en Matlab

- Comentarios:
  - Bibliografía básica y complementaria: [BB2: Opp&Sch] Cap. 8, Secs. 8.0-8.5; [BB3: McC&Sch&Yod] Cap. 13, Secs. 13.0-13.5

DFT ¿otra transformada más?

- Hasta ahora ya hemos visto 4 tipos de TF/DSF:
  - ¿Es la señal continua o discreta?
  - ¿Es la señal periódica o aperiódica?
  - Ecuaciones de análisis y síntesis ligeramente distintas

$$
\begin{tikzcd}
X(j\omega) \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & X(e^{j\Omega}) \arrow[d, leftrightarrow] \\
a_k & A_k
\end{tikzcd}
$$

- En este tema otra más: la DFT (Discrete Fourier Transform) → ¿por qué?
  - Las anteriores sólo podían calcularse con “papel y lápiz”, la DFT puede calcularse en un microprocesador (todos los *smart-phones* tienen chips que están calculando DFTs)
  - Se utiliza para calcular/estimar TF o espectros (analizador de espectros, osciloscopio, Matlab,…)
  - Es una pieza de muchos algoritmos de procesamiento de señal (codificación de imagen, modulaciones digitales, radar, estimación de armónicos en bioingeniería, …)

- ¿Por qué este empeño en las TF? → Expresar señales como suma de exponenciales complejas → ¿Por qué es esto importante? → Temas 0 y 1

Definición de la DFT

- ¿Se utiliza sobre SC o sobre SD? ¿periódicas o aperiódicas?
- ¿Ecuaciones de análisis y síntesis?
- La DFT es una transformación **discreto a discreto**, toma señales discretas de longitud finita y devuelve una señal discreta de longitud finita

$$X_N[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn}, \qquad k = 0, \ldots, N-1$$

Ec. de análisis de la DFT

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X_N[k] \cdot e^{+j\frac{2\pi}{N}nk}, \qquad n = 0, \ldots, N-1$$

Ec. de síntesis de la DFT

- $x[n]$ tiene longitud N (ya veremos qué pasa si no es así), $X_N[k]$ tiene longitud N
- La definición no es única, depende de N → Deberemos decir "DFT de longitud N"
- En ocasiones se utiliza la notación:

$$X_N[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \qquad , k=0, \ldots, N-1$$

donde $W_N := e^{-j\frac{2\pi}{N}} \Rightarrow W_N^{kn} := e^{-j\frac{2\pi}{N}kn}$

Cálculo de la DFT

- Son parecidas a las ecuaciones de la TF, pero más fáciles de calcular
  - ¿Cuánto vale la DFT en 0?

$$X_N[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn}, \qquad k = 0, ..., N-1$$

$$X_N[0] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}0n} = \sum_{n=0}^{N-1} x[n] \cdot e^{-j0} = \sum_{n=0}^{N-1} x[n] \longrightarrow$$

Suma de valores de x[n]!!

  - Supongamos que N=4 ¿cuánto vale la DFT en 0?¿cuánto en 1?

$$X_4[0] = \sum_{n=0}^{3} x[n] = x[0] + x[1] + x[2] + x[3]$$

$$\begin{aligned} X_4[1] &= \sum_{n=0}^{3} x[n] \cdot e^{-j\frac{2\pi}{4}1n} = \sum_{n=0}^{3} x[n] \cdot e^{-j\frac{\pi}{2}n} = \sum_{n=0}^{3} x[n] \cdot (e^{-j\frac{\pi}{2}})^n = \sum_{n=0}^{3} x[n] \cdot (-j)^n \\ &= x[0](-j)^0 + x[1](-j)^1 + x[2](-j)^2 + x[3](-j)^3 = x[0] - x[1]j - x[2] + x[3]j \end{aligned}$$

Cálculo de la DFT

$$X_N[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn}, \qquad k = 0, ..., N-1$$

- Supongamos que N=4 ¿cuánto vale la DFT de cualquier señal en 0, 1, 2, 3?

$$X_4[0] = \sum_{n=0}^{3} x[n] \cdot e^{-j\frac{2\pi}{4}0n} = \sum_{n=0}^{3} x[n] \cdot 1 = x[0] + x[1] + x[2] + x[3]$$

$$X_4[1] = \sum_{n=0}^{3} x[n] \cdot e^{-j\frac{2\pi}{4}1n} = \sum_{n=0}^{3} x[n] \cdot e^{-j\frac{\pi}{2}n} = \sum_{n=0}^{3} x[n] \cdot (e^{-j\frac{\pi}{2}})^n = \sum_{n=0}^{3} x[n] \cdot (-j)^n$$
$$= x[0](-j)^0 + x[1](-j)^1 + x[2](-j)^2 + x[3](-j)^3 = x[0] - x[1]j - x[2] + x[3]j$$

$$X_4[2] = \sum_{n=0}^{3} x[n] \cdot e^{-j\frac{2\pi}{4}2n} = \sum_{n=0}^{3} x[n] \cdot (e^{-j\pi})^n = x[0] + x[1](-1) + x[2](-1)^2 + x[3](-1)^3$$

$$X_4[3] = \sum_{n=0}^{3} x[n] \cdot e^{-j\frac{2\pi}{4}3n} = \sum_{n=0}^{3} x[n] \cdot (e^{-j\frac{3\pi}{2}})^n = x[0] + x[1]j + x[2]j^2 + x[3]j^3$$

Cada punto de la DFT se obtiene realizando N sumas y N multiplicaciones
→ En total necesitamos N^2 sumas y N^2 multiplicaciones

Cálculo de la DFT: Ejemplos

- DFT de longitud 4 de las siguientes señales:

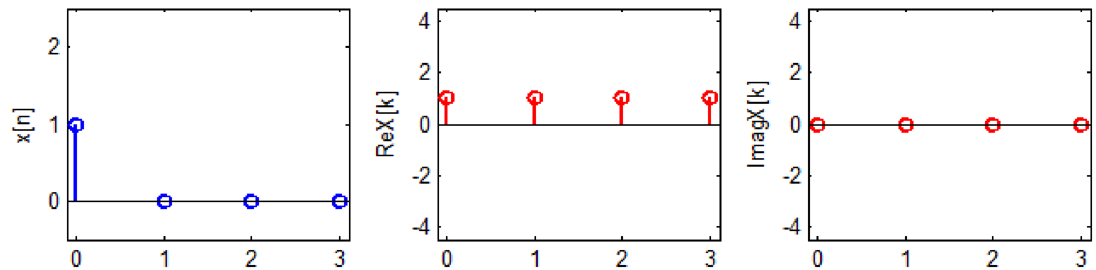
$$X_N[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn}, \qquad k = 0, ..., N-1$$

$$\begin{aligned} X_4[0] &= x[0] + x[1] + x[2] + x[3] \\ X_4[1] &= x[0] + x[1](-j) + x[2](-j)^2 + x[3](-j)^3 \\ X_4[2] &= x[0] + x[1](-1) + x[2](-1)^2 + x[3](-1)^3 \\ X_4[3] &= x[0] + x[1]j + x[2]j^2 + x[3]j^3 \end{aligned}$$

- a)

$$x[n] = \delta[n]$$

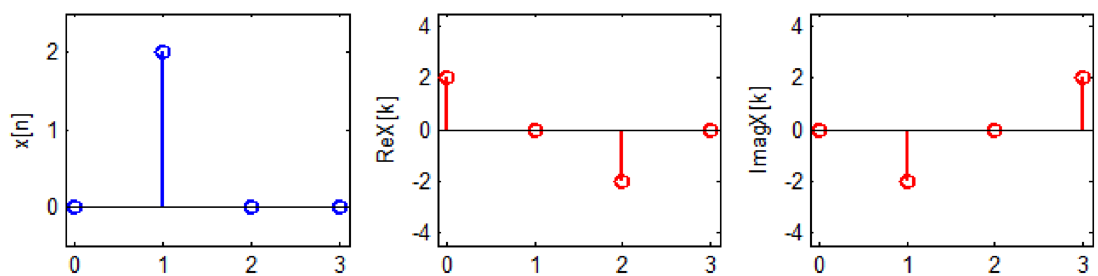
$$\begin{aligned} X_4[0] &= 1 \\ X_4[1] &= 1 \\ X_4[2] &= 1 \\ X_4[3] &= 1 \end{aligned}$$



- b)

$$x[n] = 2\delta[n-1]$$

$$\begin{aligned} X_4[0] &= 2 \\ X_4[1] &= -2j \\ X_4[2] &= -2 \\ X_4[3] &= 2j \end{aligned}$$



Cálculo de la DFT: Ejemplos

- DFT de longitud 4 de las siguientes señales:

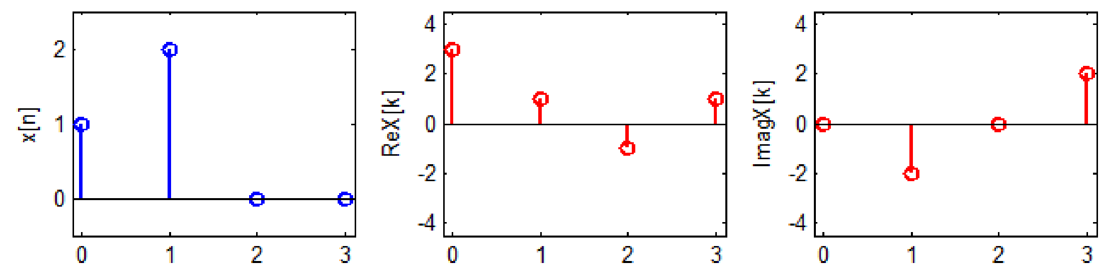
$$X_N[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn}, \qquad k = 0, ..., N-1$$

$$\begin{aligned} X_4[0] &= x[0] + x[1] + x[2] + x[3] \\ X_4[1] &= x[0] + x[1](-j) + x[2](-j)^2 + x[3](-j)^3 \\ X_4[2] &= x[0] + x[1](-1) + x[2](-1)^2 + x[3](-1)^3 \\ X_4[3] &= x[0] + x[1]j + x[2]j^2 + x[3]j^3 \end{aligned}$$

- c)

$$x[n] = \delta[n] + 2\delta[n-1]$$

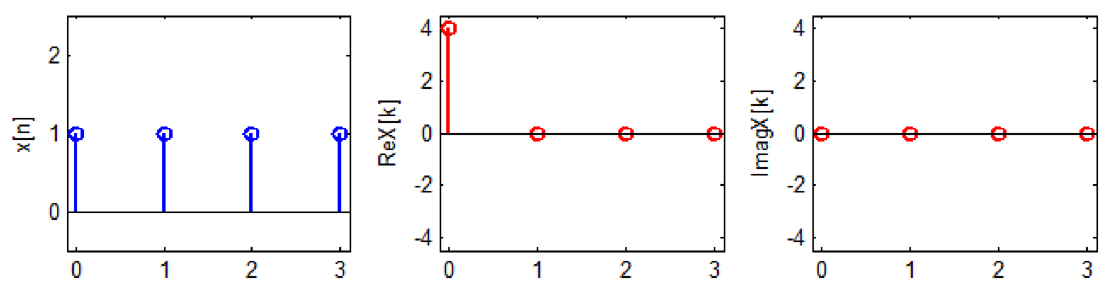
$$\begin{aligned} X_4[0] &= 1 + 2 = 3 \\ X_4[1] &= 1 - 2j \\ X_4[2] &= 1 - 2 = -1 \\ X_4[3] &= 1 + 2j \end{aligned}$$



- d)

$$x[n] = 1, \quad n = 0, 1, 2, 3$$

$$\begin{aligned} X_4[0] &= 1 + 1 + 1 + 1 = 4 \\ X_4[1] &= 1 - j - 1 + j = 0 \\ X_4[2] &= 1 - 1 + 1 - 1 = 0 \\ X_4[3] &= 1 + j - 1 - j = 0 \end{aligned}$$



Cálculo de la DFT

- Cada valor de la DFT N sumas y N multiplicaciones → Se puede calcular a través de una matriz → Matriz de DFT

Para el caso de N=4

$$\begin{bmatrix} X_4[0] \\ X_4[1] \\ X_4[2] \\ X_4[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}$$

Para el caso de N genérico

$$\mathbf{F} := \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & e^{-j\frac{2\pi}{N}} & e^{-j\frac{2\pi}{N}2} & \dots & e^{-j\frac{2\pi}{N}(N-1)} \\ 1 & e^{-j\frac{2\pi}{N}2} & e^{-j\frac{2\pi}{N}4} & \dots & e^{-j\frac{2\pi}{N}2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & e^{-j\frac{2\pi}{N}(N-1)} & e^{-j\frac{2\pi}{N}2(N-1)} & \dots & e^{-j\frac{2\pi}{N}(N-1)(N-1)} \end{bmatrix}_{N\times N}$$

¡¡La DFT es muy fácil de calcular en un ordenador!!

Cálculo de la DFT

- ¿Se puede calcular también con “papel y lápiz”? → Para casos fáciles sí

$$X_N[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn}, \qquad k = 0, ..., N-1$$

- a)

$$x[n] = 3\delta[n-2]$$

$$X_N[k] = \sum_{n=0}^{N-1} 3\delta[n-2] \cdot e^{-j\frac{2\pi}{N}kn} = 3\sum_{n=0}^{N-1} \delta[n-2] \cdot e^{-j\frac{2\pi}{N}k2} = 3e^{-j\frac{4\pi}{N}k}$$

- b)

$$x[n] = (-1)^n, \qquad n = 0, ..., N-1$$

$$X_N[k] = \sum_{n=0}^{N-1} (-1)^n \cdot e^{-j\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} (e^{-j\pi})^n \cdot (e^{-j\frac{2\pi}{N}k})^n = \sum_{n=0}^{N-1} (e^{-j(\frac{2\pi}{N}k+\pi)})^n$$

$$= \frac{1-(e^{-j(\frac{2\pi}{N}k+\pi)})^N}{1-(e^{-j(\frac{2\pi}{N}k+\pi)})} = \frac{1-e^{-jN\pi}}{1+e^{-j\frac{2\pi}{N}k}}$$

¿Qué pasa con el numerador si N es impar? ¿Y si es par?

Cálculo de la DFT

- Papel y lápiz…

$$X_N[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn}, \qquad k = 0, ..., N-1$$

- c)

$$x[n] = 3\delta[n-2]$$

$$X_N[k] = \sum_{n=0}^{N-1} 3\delta[n-2] \cdot e^{-j\frac{2\pi}{N}kn} = 3\sum_{n=0}^{N-1} \delta[n-2] \cdot e^{-j\frac{2\pi}{N}k2} = 3e^{-j\frac{4\pi}{N}k}$$

- d)

$$x[n] = \begin{cases} 1 & n = 0, ..., L \\ 0 & resto \end{cases}$$

$$\begin{aligned} X_N[k] &= \sum_{n=0}^{L} 1 \cdot e^{-j\frac{2\pi}{N}kn} + \sum_{n=L+1}^{N-1} 0 \cdot e^{-j\frac{2\pi}{N}kn} = \sum_{n=0}^{L} (e^{-j\frac{2\pi}{N}k})^n = \frac{1 - e^{-j\frac{2\pi}{N}k(L+1)}}{1 - e^{-j\frac{2\pi}{N}k}} \\ &= \frac{e^{-j\frac{\pi}{N}k(L+1)}\left(e^{+j\frac{\pi}{N}k(L+1)} - e^{-j\frac{\pi}{N}k(L+1)}\right)}{e^{-j\frac{\pi}{N}k}\left(e^{+j\frac{\pi}{N}k} - e^{-j\frac{\pi}{N}k}\right)} = e^{-j\frac{\pi}{N}kL}\frac{2j\sin\left(\frac{\pi}{N}k(L+1)\right)}{2j\sin\left(\frac{\pi}{N}k\right)} \\ &= \frac{\sin\left(\frac{(L+1)\pi}{N}k\right)}{\sin\left(\frac{\pi}{N}k\right)} e^{-j\frac{L\pi}{N}k} \end{aligned}$$

Más ejemplos: hojas de transformadas y propiedades

Ubicándonos

- **Tema 4: Fundamentos de la Transformada Discreta de Fourier**
  - **4.1 Definición: la DFT como el muestreo de la TF**
    - 4.1.0 Introducción
    - 4.1.1 Definición y ejemplos
    - **4.1.2 La DFT como el muestreo de la TF**
    - 4.1.3 Problemas y aspectos prácticos
  - 4.2 Propiedades
  - 4.3 Convolución circular: definición y relación con la DFT
  - 4.4 La DFT en Matlab

- Comentarios:
  - Bibliografía básica y complementaria: [BB2: Opp&Sch] Cap. 8, Secs. 8.0-8.5; [BB3: McC&Sch&Yod] Cap. 13, Secs. 13.0-13.5

Relacionando la DFT con la TF

- La DFT se parece a la TF de secuencias, comparemos Ecs. Análisis

$$X_N[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn}, \qquad k = 0, ..., N-1$$

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{+\infty} x[n] \cdot e^{-j\Omega n}, \qquad -\infty < \Omega < \infty$$

- Diferencias:
  - La TF es continua y la DFT es discreta; la TF es de longitud infinita y la DFT es de longitud finita; la TF es periódica y la DFT es aperiódica
  - ¿Qué pasa si tomamos N muestreas equiespaciadas de la TF en el intervalo $[0,2\pi)$?

$$\hat{X}_N[k] := X(e^{j\Omega})\big|_{\Omega=\frac{2\pi}{N}k} = \sum_{n=-\infty}^{+\infty} x[n] \cdot e^{-j\frac{2\pi}{N}kn}, \qquad k = 0, ..., N-1$$

La DFT es algo parecido a la TF muestreada → estudiaremos esto con mayor detalle

Si $N = 20$: $\Omega = 0,\ 0.1\pi,\ 0.2\pi,\ 0.3\pi,\ \ldots,\ 1.9\pi$

Relacionando la DFT con la TF

- Vamos a estudiar la relación entre DFT y TF para tres casos:
  - a) Secuencias de longitud finita definidas entre 0 y N-1

$$x[n] = 0 \quad n \notin [N_i, N_f] \quad \text{con } N_i \geq 0 \text{ y } N_f \leq (N-1)$$

  - b) Secuencias de longitud infinita

$$x[n] \neq 0 \quad \text{en un número infinito de instantes}$$

  - c) Secuencias de longitud finita pero definidas fuera del intervalo [0,N-1]

$$x[n] = 0 \quad n \notin [N_i, N_f] \quad \text{con } N_i < 0 \text{ o } N_f > (N-1)$$

- Vamos a ver que:
  - Si tenemos “cuidado”, en los casos a) y c) la DFT puede utilizarse para obtener muestras de la TF
  - En el caso b) sólo podremos obtener los valores de forma aproximada (estaremos muestreando una TF que no es exactamente la de la señal, pero que sí está relacionada con la TF original)

Relacionando la DFT con la TF

- a) Secuencias de longitud finita definidas entre 0 y N-1

$$x[n] = 0 \quad n \notin [N_i, N_f] \quad \text{con } N_i \geq 0 \text{ y } N_f \leq (N-1)$$

$$X_N[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn}, \qquad k = 0, ..., N-1$$

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{+\infty} x[n] \cdot e^{-j\Omega n}, \qquad \infty < \Omega < \infty$$

- Si utilizamos el hecho de que la secuencia es finita

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{+\infty} x[n] \cdot e^{-j\Omega n} = \sum_{n=N_i}^{N_f} x[n] \cdot e^{-j\Omega n} = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\Omega n}$$

- Muestreando en los valores: $\Omega_k = \frac{2\pi}{N}k \quad \Rightarrow \quad \Omega = 0, \frac{2\pi}{N}, \frac{4\pi}{N}, ..., \frac{2(N-1)\pi}{N}$

$$\hat{X}_N[k] := X(e^{j\Omega})\Big|_{\Omega=\frac{2\pi}{N}k} = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn},$$

¡¡La DFT es la TF muestreada!!

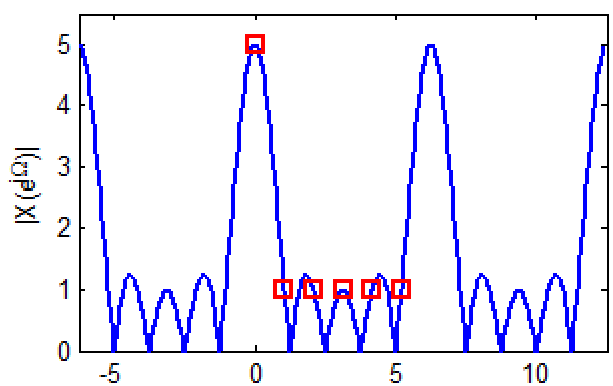
DFT como TF muestrada: Ejemplo

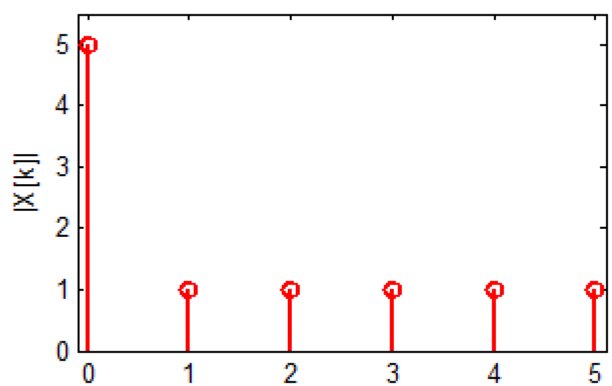
- Ej. 1: $N = 6$ y $x[n] = \begin{cases} 1 & n = 0, ..., 4 \\ 0 & n \leq -1 \text{ o } n \geq 5 \end{cases}$

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{+\infty} x[n] \cdot e^{-j\Omega n} = \sum_{n=0}^{4} e^{-j\Omega n} = \frac{1 - e^{-j5\Omega}}{1 - e^{-j\Omega}} = \frac{\sin(5\Omega/2)}{\sin(\Omega/2)} e^{-j2\Omega}$$

$$X_6[k] = \sum_{n=0}^{5} x[n] \cdot e^{-j\frac{2\pi}{6}kn} = \frac{\sin\left(\frac{5\pi}{6}k\right)}{\sin\left(\frac{\pi}{6}k\right)} e^{-j\frac{4\pi}{6}k}, \qquad k = 0, ..., 5$$

La hemos calculado antes





Es muy importante saber qué variable representa el eje x (horizontal) en cada gráfica

DFT como TF muestrada: Ejemplo

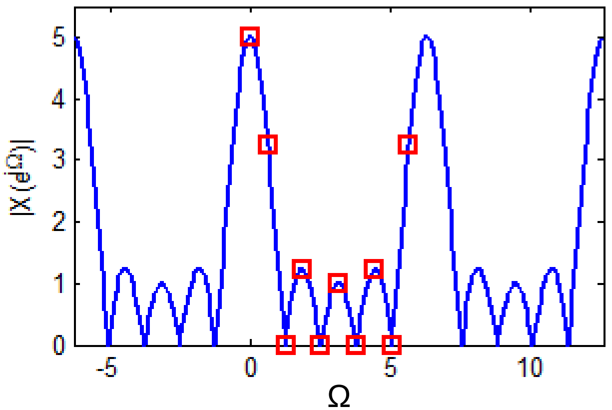
- Ej. 2: $N = 10$ y $x[n] = \begin{cases} 1 & n = 0, ..., 4 \\ 0 & n \leq -1 \text{ o } n \geq 5 \end{cases}$

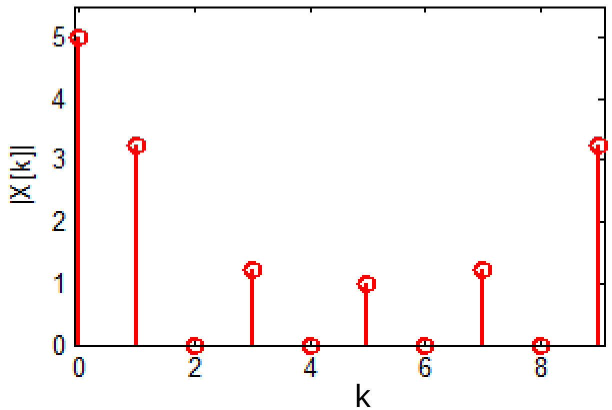
$$X(e^{j\Omega}) = \sum_{n=-\infty}^{+\infty} x[n] \cdot e^{-j\Omega n} = \sum_{n=0}^{4} e^{-j\Omega n} = \frac{1 - e^{-j5\Omega n}}{1 - e^{-j\Omega n}} = \frac{\sin(5\Omega/2)}{\sin(\Omega/2)} e^{-j2\Omega}$$

La hemos calculado antes

$$X_{10}[k] = \sum_{n=0}^{9} x[n] \cdot e^{-j\frac{2\pi}{10}kn} = \frac{\sin\left(\frac{5\pi}{10}k\right)}{\sin\left(\frac{\pi}{10}k\right)} e^{-j\frac{4\pi}{10}k}, \qquad k = 0, ..., 9$$

La hemos calculado antes





Es muy importante saber qué variable representa el eje x (horizontal) en cada gráfica

DFT como TF muestrada: Ejemplo

- Ej. 3: $N = 3$ y $\quad x[n] = \begin{cases} 1 & n = 0, ..., 4 \\ 0 & n \leq -1 \text{ o } n \geq 5 \end{cases}$

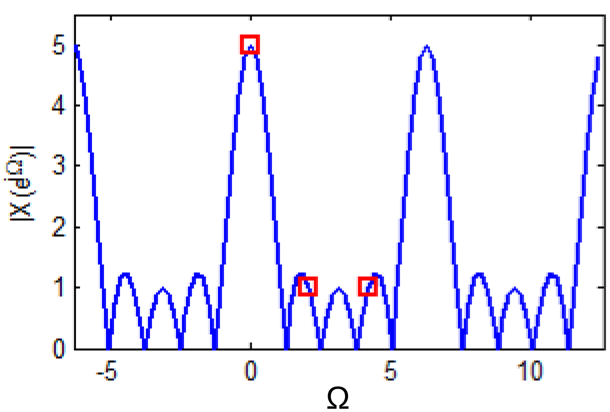
¡Ojo, la condición no se cumple!

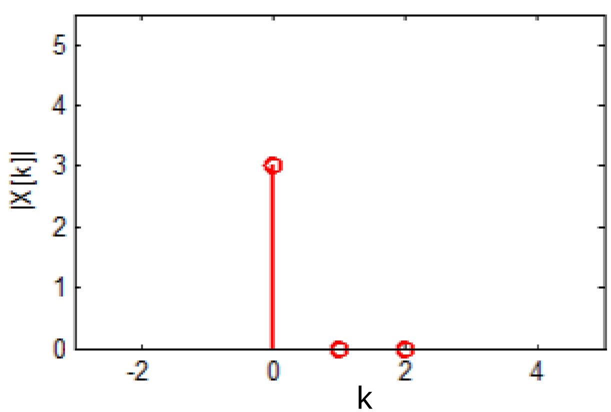
$$X(e^{j\Omega}) = \sum_{n=-\infty}^{+\infty} x[n] \cdot e^{-j\Omega n} = \sum_{n=0}^{4} e^{-j\Omega n} = \frac{1 - e^{-j5\Omega n}}{1 - e^{-j\Omega n}} = \frac{\sin(5\Omega/2)}{\sin(\Omega/2)} e^{-j2\Omega}$$

La hemos calculado antes

$$X_3[k] = \sum_{n=0}^{2} x[n] \cdot e^{-j\frac{2\pi}{3}kn} = \frac{\sin\left(\frac{3\pi}{3}k\right)}{\sin\left(\frac{\pi}{3}k\right)} e^{-j\frac{2\pi}{3}k}, \qquad k = 0, ..., 2$$

La hemos calculado antes





La condición no se cumple → la DFT no se corresponde con el muestreo de la TF

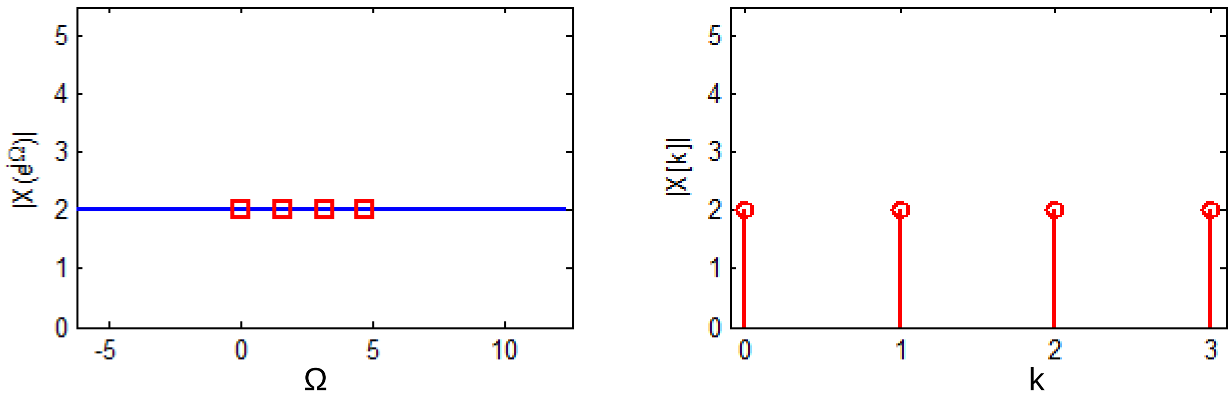
DFT como TF muestrada: Ejemplo

- Ej. 4: $N = 4$ y $x[n] = 2\delta[n]$

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{+\infty} x[n] \cdot e^{-j\Omega n} = \sum_{n=-\infty}^{\infty} 2\delta[n] = 2$$

$$X_4[k] = \sum_{n=0}^{3} x[n] \cdot e^{-j\frac{2\pi}{4}kn} = 2, \qquad k = 0, ..., 3$$

La hemos calculado antes



Relacionando la DFT con la TF

b) Secuencias aperiódicas de longitud infinita

$$x[n] \neq 0 \quad \text{en un número infinito de instantes}$$

$$X_N[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn}, \qquad k = 0, ..., N-1$$

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{+\infty} x[n] \cdot e^{-j\Omega n} = \sum_{n=N_i}^{N_f} x[n] \cdot e^{-j\Omega n}, \qquad -\infty < \Omega < \infty$$

- Ya hemos visto en el Ej. 3 que DFT y TF muestreada no coinciden:

$$\hat{X}_N[k] := X(e^{j\Omega})\Big|_{\Omega=\frac{2\pi}{N}k} = \sum_{n=-\infty}^{\infty} x[n] \cdot e^{-j\frac{2\pi}{N}kn} \neq \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn} = X_N[k]$$

- Evidentemente son distintas pero, ¿guardan algún parecido?
  - Sí
  - La clave para entenderlo es el enventanado

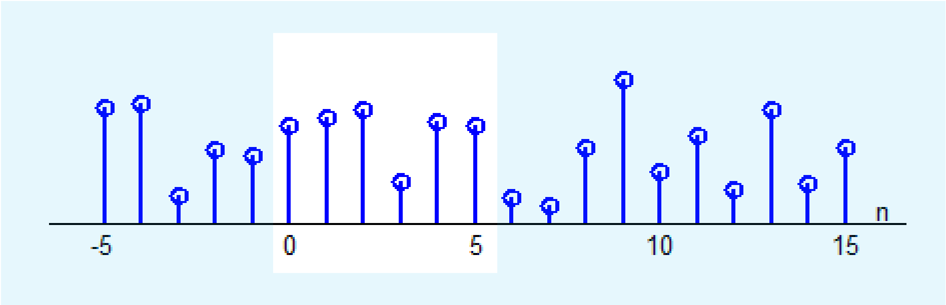
Enventanado

ventana de longitud L

$$x_{W,L}[n] = x[n] \cdot w_L[n]$$

- ¿Qué es el enventanado?
  - Multiplicar por una señal (ventana) de longitud finita (L)

$$w_L[n] = \begin{cases} 1 & n = 0, \ldots, L-1 \\ 0 & \text{resto} \end{cases}$$



- ¿Para qué enventanamos? Para convertir una señal de longitud infinita (o de longitud muy larga) en una señal de longitud más pequeña
  - En el dominio del tiempo perdemos información (“tiramos” muestras)
  - ¿Qué ocurre en frecuencia? → Distorsión y pérdida de resolución frecuencial

El enventanado es útil para entender qué pasa con la DFT de señales de longitud infinita, pero es útil para muchísimas cosas (diseño de filtros, algoritmos de codificación de fuente, procesamiento de señales aleatorias,…) → Hay varios tipos de ventanas (rectangulares, triangulares, gaussianas, …) → De hecho, hay libros enteros para ver cómo se diseñan ventanas (frames)

Enventanado y DFT

- ¿Cómo lo podemos utilizar para entender qué está pasando?
  - TF muestreada vs. DFT_N

$$x[n] \neq 0 \quad \text{en un número infinito de instantes}$$

$$\hat{X}_N[k] := X(e^{j\Omega})\big|_{\Omega=\frac{2\pi}{N}k} = \sum_{n=-\infty}^{\infty} x[n]\cdot e^{-j\frac{2\pi}{N}kn} \neq \sum_{n=0}^{N-1} x[n]\cdot e^{-j\frac{2\pi}{N}kn} = X_N[k]$$

- Si definimos la señal enventanada

$$x_{W,N}[n] = x[n]\cdot w_N[n] \Rightarrow$$
$$x_{W,N}[n] = x[n],\ n = 0, \ldots, N-1$$

$$TF\{x_{W,N}[n]\}\big|_{\Omega=\frac{2\pi}{N}k} = \sum_{n=-\infty}^{\infty} x_{W,N}[n]\cdot e^{-j\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} x_{W,N}[n]\cdot e^{-j\frac{2\pi}{N}kn} = DFT_N\{x_{W,N}[n]\}$$

¡Sí coinciden!

  - La DFT de $x_{W,N}[n]$ coincide con el muestreo de la TF de $x_{W,N}[n]$
  - Pero … la DFT de $x[n]$ y la DFT de $x_{W,N}[n]$ es la misma $\quad DFT_N\{x[n]\} = DFT_N\{x_{W,N}[n]\}$
  - La DFT de $x[n]$ corresponde a muestrear la TF de $x_{W,N}[n]$ →

$$\begin{aligned} X_N[k] = DFT_N\{x[n]\} &= \sum_{n=0}^{N-1} x[n]\cdot e^{-j\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} x_{W,N}[n]\cdot e^{-j\frac{2\pi}{N}kn} \\ &= \sum_{n=-\infty}^{\infty} x_{W,N}[n]\cdot e^{-j\frac{2\pi}{N}kn} = TF\{x_{W,N}[n]\}\big|_{\Omega=\frac{2\pi}{N}k} \end{aligned}$$

Enventanado y DFT

- Acabamos de concluir que la DFT de x[n] corresponde a muestrear la TF de $x_{W,N}[n]$

$$X_N[k] = DFT_N\{x[n]\} = \sum_{n=0}^{N-1} x[n]\cdot e^{-j\frac{2\pi}{N}kn} = \sum_{n=-\infty}^{\infty} x_{W,N}[n]\cdot e^{-j\frac{2\pi}{N}kn} = TF\{x_{W,N}[n]\}\big|_{\Omega=\frac{2\pi}{N}k}$$

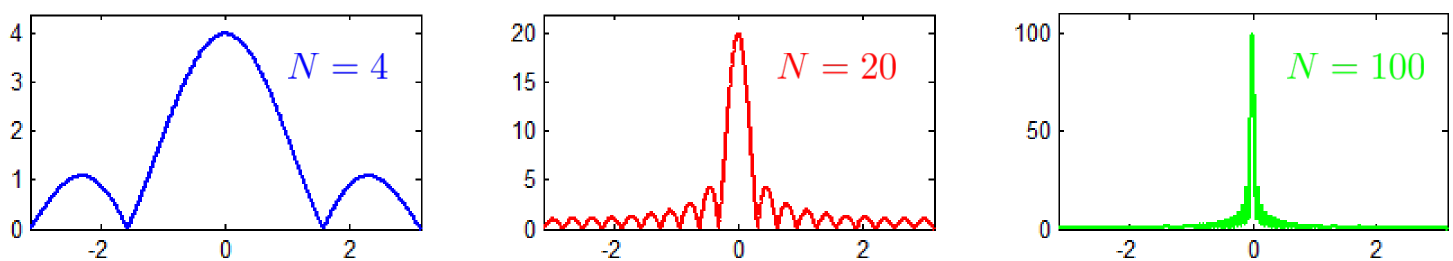
- ¿Cómo se relaciona la TF de $x_{W,N}[n]$ con la TF de x[n]? → Convolución

$$x_{W,N}[n] = x[n]\cdot w_N[n] \xleftrightarrow{\text{TF}} X_{W,N}(e^{j\Omega}) = \frac{1}{2\pi}X(e^{j\Omega}) *_{2\pi} W_N(e^{j\Omega})$$

$$w_N[n] = \begin{cases} 1 & n = 0, \dots, N-1 \\ 0 & \text{resto} \end{cases} \qquad W_N(e^{j\Omega}) = \frac{\sin\left(\frac{N\Omega}{2}\right)}{\sin\left(\frac{\Omega}{2}\right)} e^{-j\frac{(N-1)\Omega}{2}}$$

- La DFT de x[n] corresponde a muestrear la señal resultante de convolucionar la TF original con la TF de la ventana (sinc con primer cero en $2\pi/N$ → anchura $4\pi/N$)

$|W_N(e^{j\Omega})|$

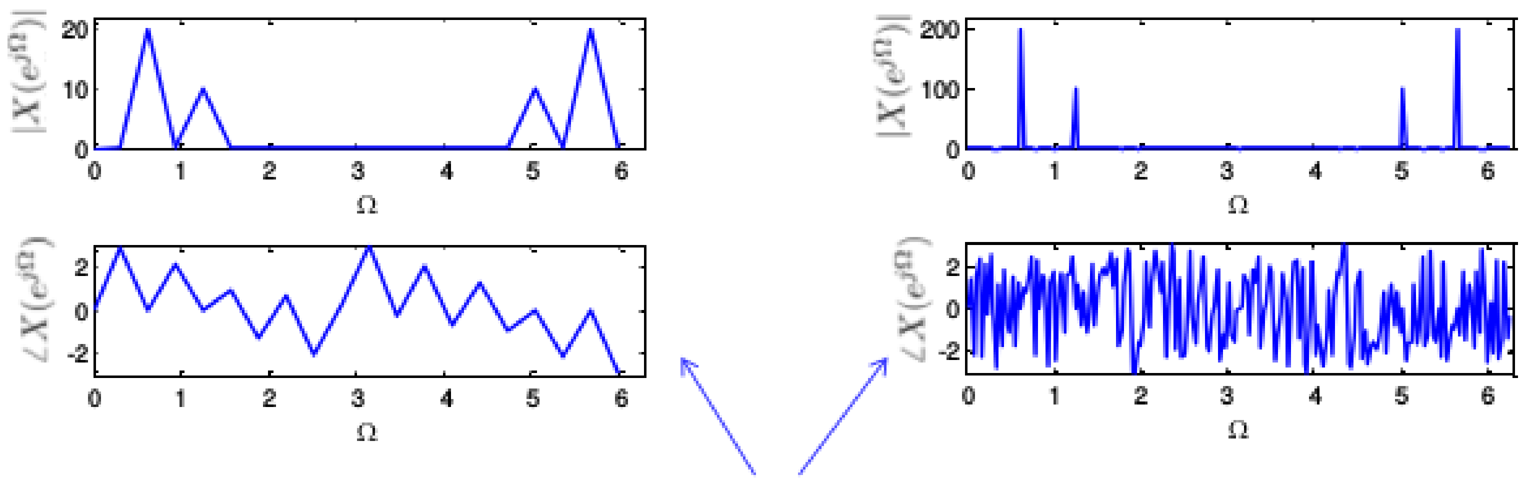


Gráficamente

- Ej. 1) Suma de sinusoides separadas $x[n] = 2\cos\left(2\pi n/10\right) + \cos\left(4\pi n/10\right)$

$$X(e^{j\Omega}) = 2\pi\delta(\Omega-2\pi/10)+\pi\delta(\Omega-4\pi/10)+\pi\delta(\Omega-16\pi/10)+2\pi\delta(\Omega-18\pi/10) \qquad \text{period. } 2\pi$$

  - Calculamos la DFT de N puntos, obtenemos la TF (señal continua en frecuencia) por interpolación lineal a partir de los N valores de la DFT, aplicando lo visto en el tema de muestreo y estudiamos cuánto se parece a la TF teórica.



N=20

Cuidado con las fases

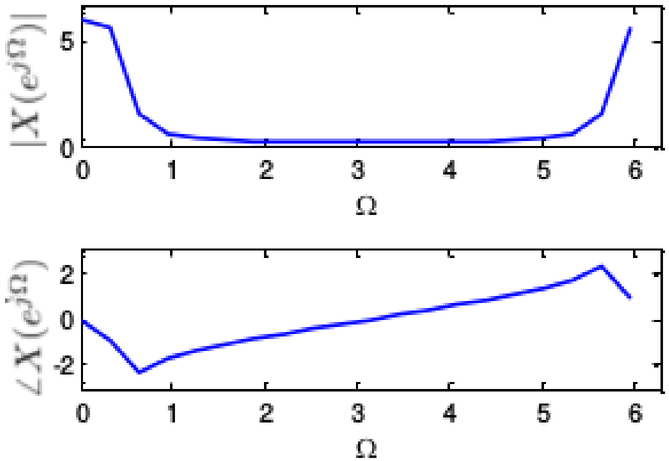
N=200

Gráficamente

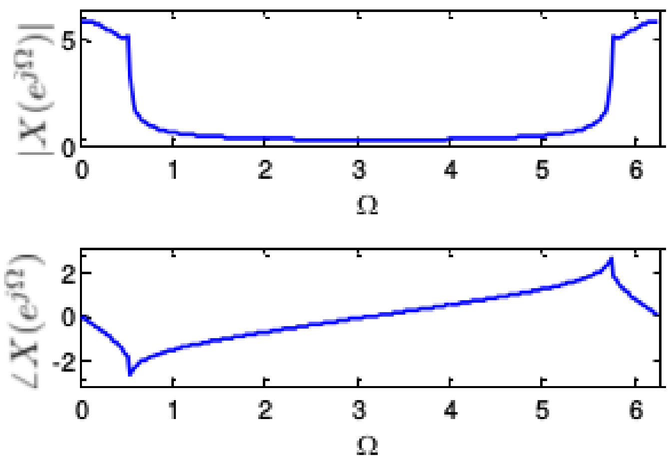
- Ej. 2) Sinc desplazada

$$x[n] = \frac{\sin\left(\frac{\pi(n-3)}{6}\right)}{\frac{\pi(n-3)}{6}} \qquad X(e^{j\Omega}) = \begin{cases} 6e^{-j3\Omega}, & |\Omega| \leq \frac{\pi}{10} \\ 0, & \frac{\pi}{10} < |\Omega| \leq \pi \end{cases} \quad \text{period. } 2\pi$$

  - Calculamos la DFT de N puntos, obtenemos la TF (señal continua en frecuencia) por interpolación lineal a partir de los N valores de la DFT, aplicando lo visto en el tema de muestreo y estudiamos cuánto se parece a la TF teórica.



N=20



N=200

Gráficamente

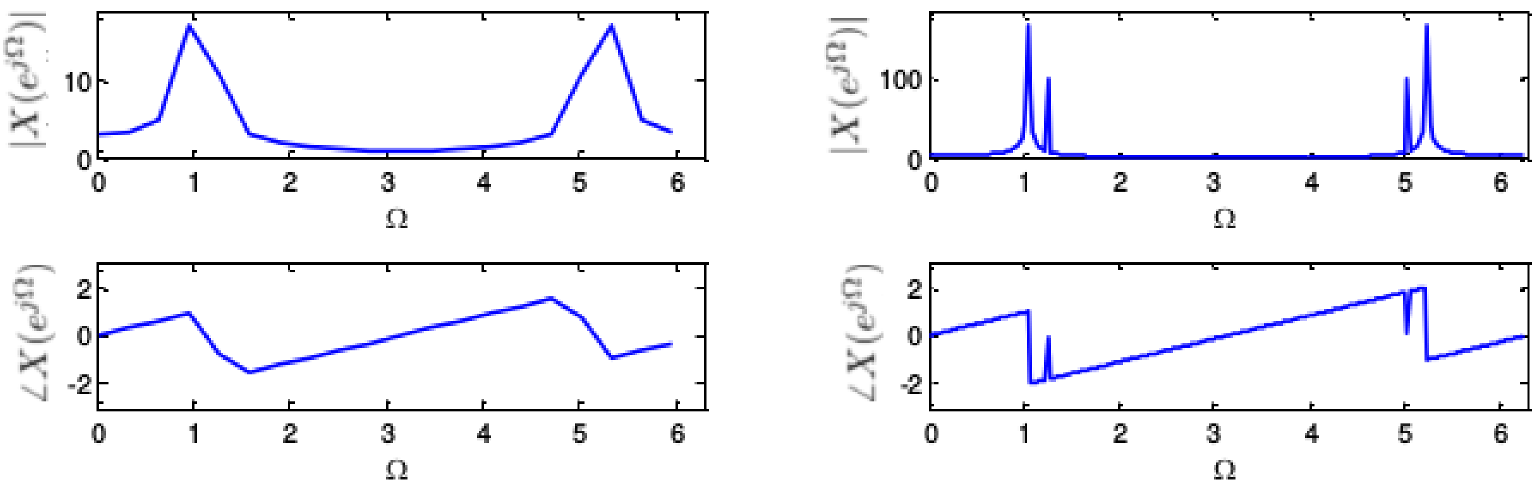
- Ej. 3) Suma de sinusoides "cercanas" (pérdida de resolución frecuencial)

$$x[n] = 2\cos\left(2\pi n/6\right) + \cos\left(2\pi n/5\right)$$

$$X(e^{j\Omega}) = 2\pi\delta(\Omega - 10\pi/30) + \pi\delta(\Omega - 12\pi/30) + \pi\delta(\Omega - 48\pi/30) + 2\pi\delta(\Omega - 50\pi/30)$$

$$\text{period. } 2\pi$$

Separadas en frecuencia: 2π/5- 2π/6 = **2π/30 = π/15**



N=20 ➔ **ΔΩ=2π/20=π/10**

N=200 ➔ **ΔΩ=2π/200= π/100**

π/10 > π/15 → ¡Pérdida de resolución espectral! → No se distinguen las dos deltas

Relacionando la DFT con la TF

- c) Secuencias de longitud finita definidas fuera del intervalo [0, N-1]

$$x[n] = 0 \quad n \notin [N_i, N_f] \quad \text{pero } N_i < 0 \text{ o } N_f \geq N$$

$$X_N[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn}, \qquad k = 0, ..., N-1$$

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{+\infty} x[n] \cdot e^{-j\Omega n} = \sum_{n=N_i}^{N_f} x[n] \cdot e^{-j\Omega n}, \qquad \infty < \Omega < \infty$$

- Ya hemos visto en el Ej. 3 (Transp. 19) que DFT y TF muestreada no coinciden:

$$\hat{X}_N[k] := X(e^{j\Omega})\Big|_{\Omega=\frac{2\pi}{N}k} = \sum_{n=N_i}^{N_f} x[n] \cdot e^{-j\frac{2\pi}{N}kn} \neq \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn} = X_N[k]$$

- Surgen dos preguntas:
  - Sin hacer nada ¿guardan algún parecido? → Sí, como en al apartado b)
  - ¿Hay alguna manera de hacer que coincidan? → Sí, vamos a verlo a continuación

Relacionando la DFT con la TF

- Caso 1: ¿Qué hacer si $N_i=0$ (la DFT si empieza en cero) pero $N_f>N-1$?

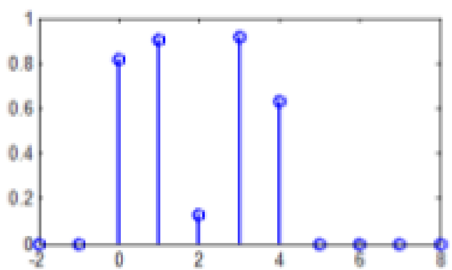
$$x[n]=0 \quad n<0 \qquad \text{pero} \quad x[n]\neq 0 \quad N\leq n\leq N_f$$

  - Para que la DFT corresponda a muestras de la TF, incrementamos el número de puntos N en los que calculamos la DFT hasta como mínimo la longitud de la señal x[n]:

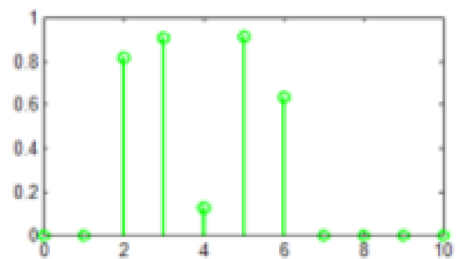
$$N \geqslant N_f + 1 \qquad X_N[k] = \sum_{n=0}^{N-1} x[n]\cdot e^{-j\frac{2\pi}{N}kn}, \qquad k=0,\ldots,N-1$$

→ ***"La longitud de la DFT tiene que ser igual o mayor que la de la señal"***

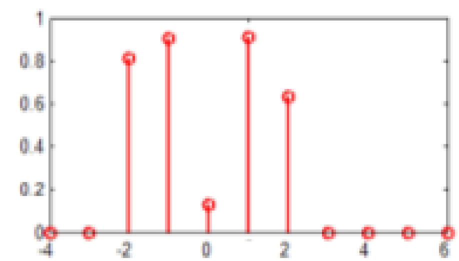
  - Importante: para que esto sea verdad, se asume que la señal empieza en cero



OK si N>4



OK si N>6



KO (indep. del valor de N)

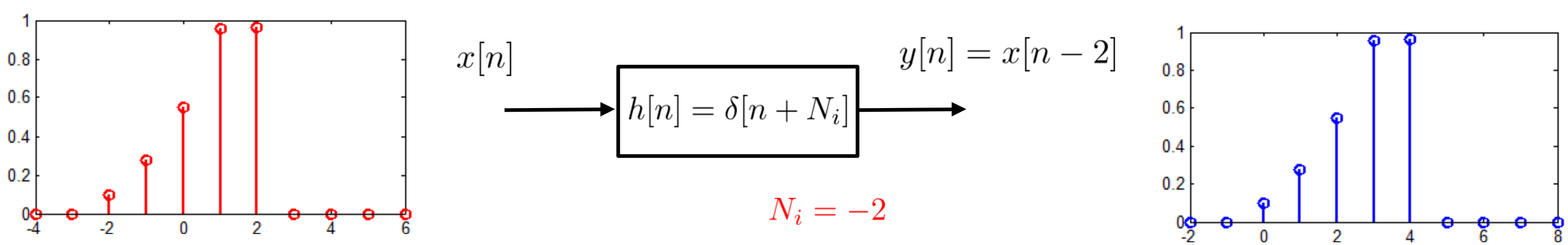
- Caso 2: ¿Qué hacer si la señal empieza antes de cero ($N_i<0$)?
  - La DFT "ignora" las muestras antes de cero → Desplazamos la señal a la derecha para que la señal comience en cero

Relacionando la DFT con la TF

- Pero… si desplazamos la señal, ¿la TF es la misma?
  - No, desplazar en tiempo supone multiplicar por una exponencial en frecuencia

$$x[n] \stackrel{\text{TF}}{\longleftrightarrow} X(e^{j\Omega}) \quad \Rightarrow \quad x[n+N_i] \stackrel{\text{TF}}{\longleftrightarrow} X(e^{j\Omega})e^{jN_i\Omega}$$

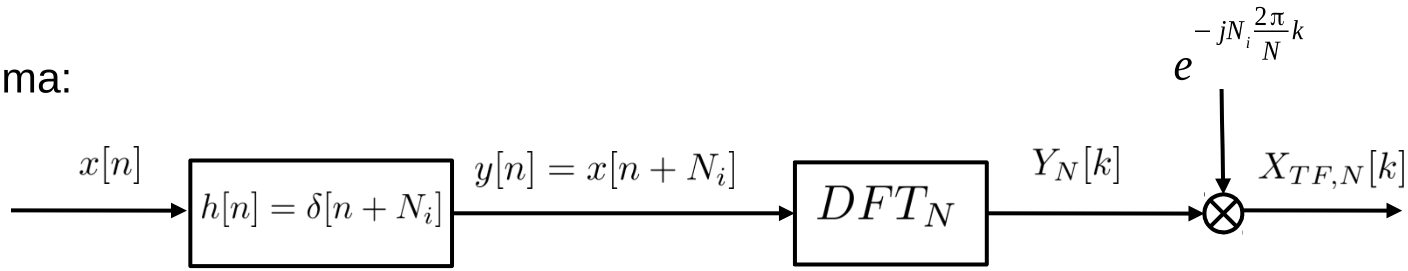
- Si, por ejemplo, queremos utilizar la DFT para estimar la TF tendremos que dividir la DFT por la exponencial muestreada



$$\begin{aligned} Y[k] &= DFT_N\Big\{y[n]\Big\} = DFT_N\Big\{x[n+N_i]\Big\} = X(e^{j\Omega})e^{jN_i\Omega}\Big|_{\Omega=\frac{2\pi}{N}k} \Rightarrow \\ Y[k] &= X(e^{j\frac{2\pi}{N}k}) \cdot e^{jN_i\frac{2\pi}{N}k} \Rightarrow X(e^{j\frac{2\pi}{N}k}) = Y[k] \cdot e^{-jN_i\frac{2\pi}{N}k} \end{aligned}$$

Relacionando la DFT con la TF

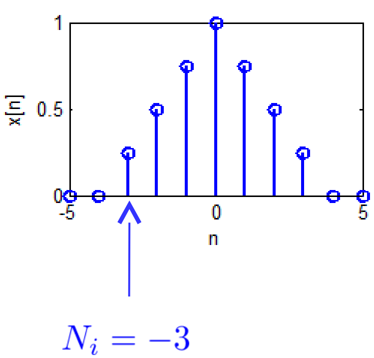
- Esquema:

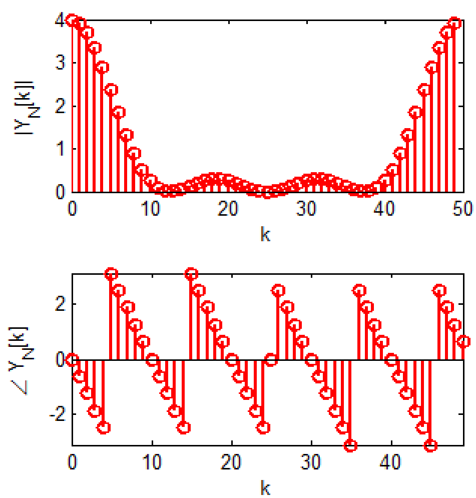


  - Si x[n]=0 para n<N_i y N>(N_f-N_i) → $X_{TF,N}[k] = X(e^{j\Omega})\big|_{\Omega=\frac{2\pi}{N}k}, \quad k=0,\ldots,N-1$

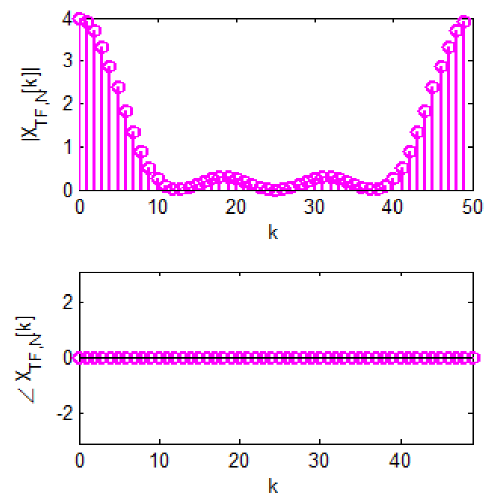
- Ejemplo:

Triángulo centrado en cero → TF es una sinc al cuadrado → Como la TF es real y positiva, tiene fase cero





Fase distinta de cero



Fase cero!!

Relación entre DFT y TF muestreada: resumen

- La DFT está relacionada con muestrear la TF en el intervalo $[0,2\pi)$
  - En concreto, muestreamos en las frecuencias $\Omega = 0, \frac{2\pi}{N}, \frac{2\pi}{N}2, \frac{2\pi}{N}3, \ldots, \frac{2\pi}{N}(N-1)$
- Si la señal original es cero fuera del intervalo $[0,N-1]$ → la DFT_N coincide exactamente con la TF muestreada
- Si la señal es de longitud infinita → la DFT coincide con una versión distorsionada de la TF (TF convolucionada con la TF de la ventana)
  - La distorsión en menor cuanto mayor sea la longitud de la DFT
  - Pérdida de resolución frecuencial
- Si la señal es de longitud finita pero definida fuera del intervalo $[0,N-1]$ → Si hacemos la DFT directamente no coincide con la TF muestreada, pero
  - Si la desplazamos para que empiece en cero
  - Hacemos que la longitud de la DFT sea como mínimo la de la señal

¡¡Coincide!!

Solapamiento al muestrear la TF

- Hemos visto que si la longitud de la DFT es suficientemente larga, la DFT puede verse como la TF muestreada → Si queremos muestrear la TF correctamente tenemos que tomar suficientes muestras → Parecido al teorema de muestreo

- ¿Significa esto que cuando la DFT no coincide con la TF de muestreada podemos decir que hay solapamiento (aliasing)? → Sí → ¿dónde, en **tiempo** o en frecuencia?

$$X(e^{j\Omega}) = TF\left\{x[n]\right\}$$

$$\hat{X}_N[k] = X(e^{j\Omega})\Big|_{\Omega=\frac{2\pi}{N}k}, \quad k = 0, ..., N-1$$

$$\hat{x}_N[n] = IDFT_N\left\{\hat{X}_N[k]\right\}, \quad n = 0, ..., N-1$$

$\boxed{\text{Se parecen } x[n] \text{ y } \hat{x}_N[n]\,? \Rightarrow \text{Sí}}$

<u>Caso particular</u>: $x[n] = 0 \;\; n \notin [0, N-1]$

$$\Downarrow$$

$$x[n] = \hat{x}_N[n]$$

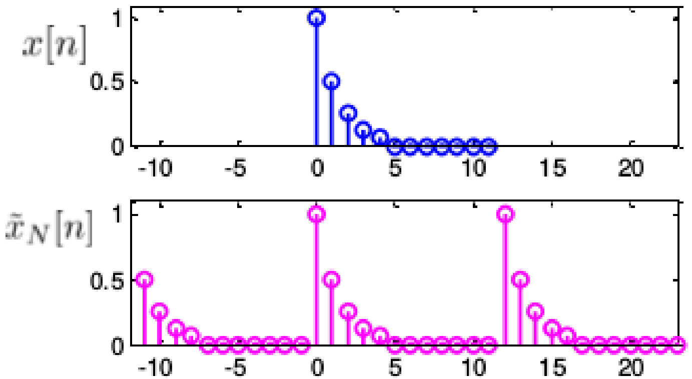
<u>Caso general</u>: $\tilde{x}_N[n] = \sum_{r=-\infty}^{\infty} x[n - rN] \quad \Rightarrow \quad \hat{x}_N[n] = \tilde{x}_N[n] w_N[n]$

Nos quedamos con 1 periodo

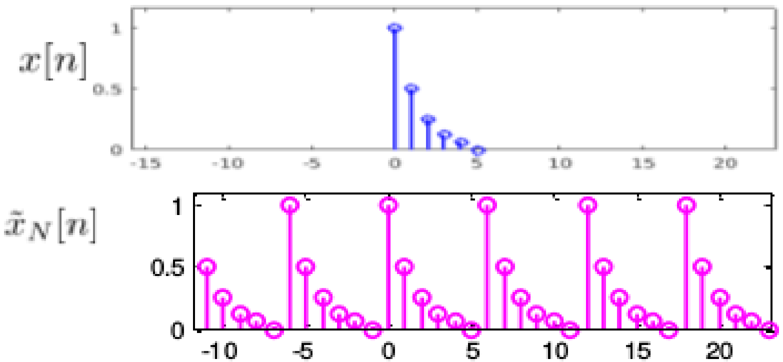
*Extensión periódica (**aliasing si la longitud de x[n] es mayor que N**)*

Ejemplos de extensión periódica

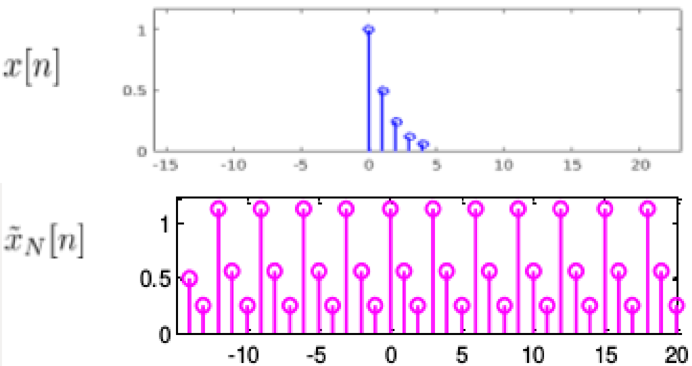
- Partimos de la señal: $x[n]=2^{-n}\,(u[n]-u[n-5]) \rightarrow$ longitud = 5



Perióica con periodo 12, no hay solape



Periódica con periodo 6, no hay solape



Periódica con periodo 3, <u>sí hay solape</u>

La extensión periódica será muy útil a lo largo de este tema (sobre todo para realizar operaciones circulares)

$$\tilde{x}_N[n] = \sum_{r=-\infty}^{\infty} x[n - rN]$$

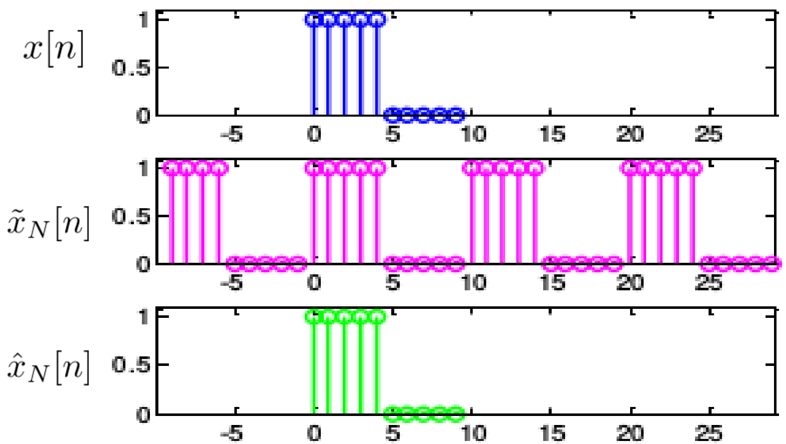
Se construye una señal periódica con periodo N

$$\begin{aligned}
\tilde{x}_N[0] &= \ldots + x[2N] + x[N] + x[0] + x[-N] + x[-2N] + \ldots \\
\tilde{x}_N[N] &= \ldots + x[3N] + x[2N] + x[N] + x[0] + x[-N] + \ldots
\end{aligned}$$

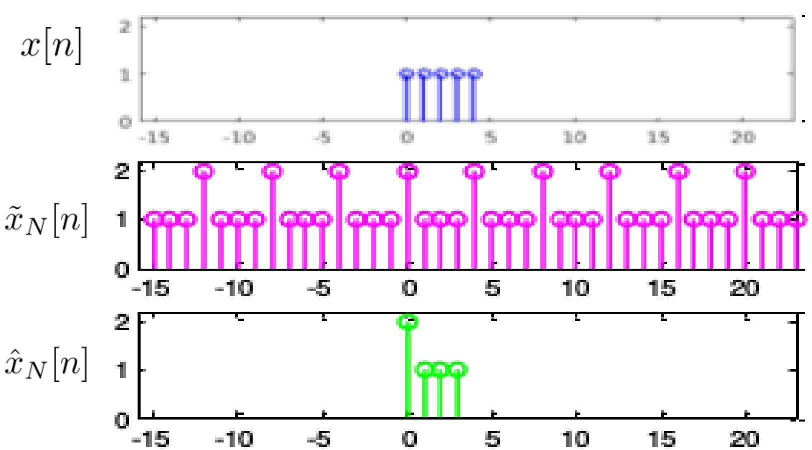
Solapamiento al muestrear la TF: Ejemplo

- Comprobemos que:
  - Camino A: $$X(e^{j\Omega}) = TF\left\{x[n]\right\} \Rightarrow \hat{X}_N[k] = X(e^{j\Omega})\big|_{\Omega=\frac{2\pi}{N}k} \Rightarrow \hat{x}_N[n] = IDFT_N\left\{\hat{X}_N[k]\right\}$$
  - *Camino B*: $$\tilde{x}_N[n] = \sum_{r=-\infty}^{\infty} x[n-rN] \Rightarrow \hat{x}_N[n] = \tilde{x}_N[n]w_N[n]$$

Empezamos viendo el Camino B: Construimos la señal periódica $\tilde{x}_N[n]$ y nos quedamos con un periodo $\hat{x}_N[n]$ (enventanando)



L=5, N=10: *N≥L*, *no hay solape*

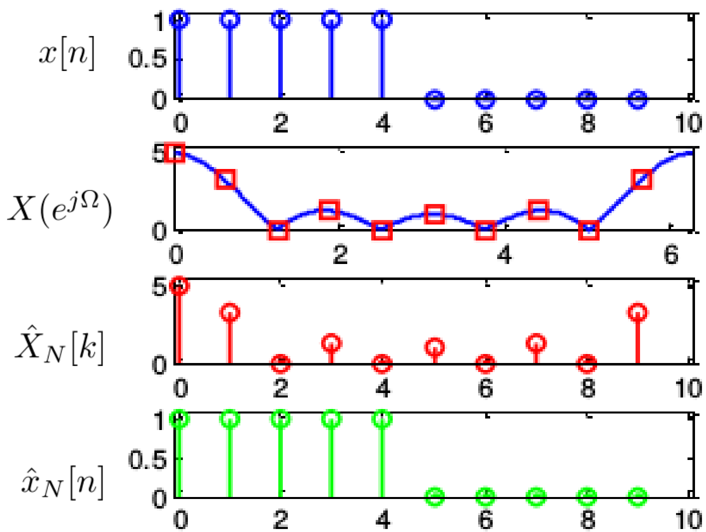


L=5, N=4: *N<L*, *hay solape*

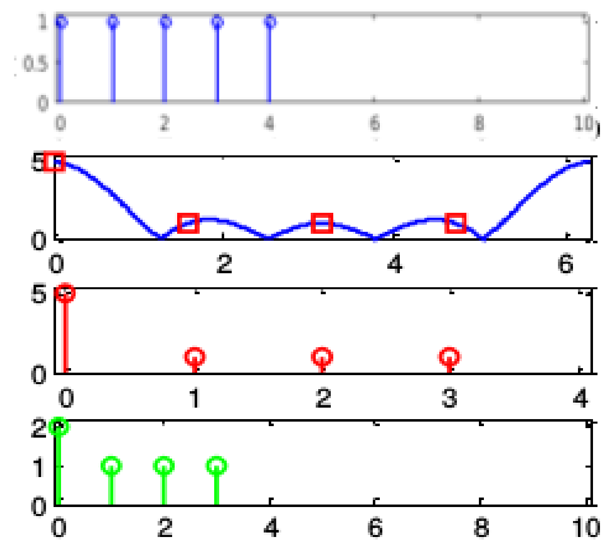
Solapamiento al muestrear la TF: Ejemplo

<u>Pasamos a ver el Camino A</u>: Calculamos la TF, muestreamos, nos quedamos con un periodo y calculamos la TF inversa

$$X(e^{j\Omega}) = TF\left\{x[n]\right\} \Rightarrow \hat{X}_N[k] = X(e^{j\Omega})\big|_{\Omega=\frac{2\pi}{N}k} \Rightarrow \hat{x}_N[n] = IDFT_N\left\{\hat{X}_N[k]\right\}$$



$\Delta\Omega = 2\pi/10$



$\Delta\Omega = 2\pi/4$

L=5, N=10: *N≥L, no hay solape*

L=5, N=4: *N<L, hay solape*

- Efectivamente nos da lo mismo (el camino B es más sencillo de calcular)
- Cuando L=5 y N=4 → *Intuición*: no estamos muestreando la TF con la suficiente resolución (intervalo de muestreo muy grande) → hay solapamiento en el otro dominio, es decir, en el dominio del tiempo.

Ubicándonos

- **<u>Tema 4</u>: Fundamentos de la Transformada Discreta de Fourier**
  - 4.1 Definición: la DFT como el muestreo de la TF
    - 4.1.0 Introducción
    - 4.1.1 Definición y ejemplos
    - 4.1.2 La DFT como el muestreo de la TF
    - **<u>4.1.3 Problemas y aspectos prácticos</u>**
  - 4.2 Propiedades
  - 4.3 Convolución circular: definición y relación con la DFT
  - 4.4 La DFT en Matlab

- Comentarios:
  - Bibliografía básica y complementaria: [BB2: Opp&Sch] Cap. 8, Secs. 8.0-8.5; [BB3: McC&Sch&Yod] Cap. 13, Secs. 13.0-13.5

Problemas: resolución espectral

- Cuando la DFT se utiliza para estimar la TF de señales de longitud infinita (=muy largas) el mayor problema es la resolución espectral
  - La solución "evidente" consiste en incrementar la longitud de la DFT
  - Si la DFT es de longitud N, muestreamos el intervalo $[0,2\pi)$ en N puntos → resolución espectral $\Delta\Omega=2\pi/N$
    - Por ejemplo, si $N=1000$ → $\Delta\Omega=2\pi/N=2\pi/1000=0.00628$ rad
    - Si necesitamos $\Delta\Omega_{min}=0.001$ rad → $N>2\pi/\Delta\Omega_{min}=2\pi/0.001$ → $N>6283$

- ¿Y si necesitamos una resolución en continua (e.g., señales de radar)?
  - La resolución depende del periodo de muestreo y de la longitud de la DFT
    - Muestreada a Ts → El intervalo de longitud 2π rad corresponde a $2\pi/Ts$ rad/seg
    - N muestras → resolución $\Delta\omega= 2\pi/(Ts\cdot N)$ rad/seg → resolución $\Delta f=1/(Ts\cdot N)$ Hz
    - Ejemplo: $fs=1MHz$, $N=2048$ → resolución $\Delta f\approx 500Hz$

- En la práctica se escoge al menos el doble de resolución de la necesaria

Problemas: complejidad computacional

- Hemos visto que la DFT_N puede calcularse a como el producto de un vector de longitud N por una matriz de tamaño $N \times N$
  - Coste computacional: N^2 sumas y N^2 multiplicaciones complejas
  - Si N es muy grande (e.g. 10.000) es demasiado costoso

$$\begin{bmatrix} X_4[0] \\ X_4[1] \\ X_4[2] \\ X_4[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}$$

- Solución: FFT (Fast Fourier Transform)
  - Es un algoritmo que permite calcular la DFT de forma más eficiente
  - ¿Cómo? Aprovechando la estructura de la matriz de DFT
  - N tiene que ser una potencia de 2
  - Coste computacional: $\log_2(N)\cdot N$ sumas y $\log_2(N/2)\cdot N$ multiplicaciones complejas

- Comparación: $N=8192 \rightarrow$ 134 millones (DFT) vs. 0.2 millones (FFT)

- En la práctica N de hasta $2^{16}=65536$, las DFT siempre suelen hacerse de potencias de 2 (telefonía celular 4G, telescopios, etc.)

Ubicándonos

- **<u>Tema 4</u>: Fundamentos de la Transformada Discreta de Fourier**
  - 4.1 Definición: la DFT como el muestreo de la TF
  - **<u>4.2 Propiedades</u>**
  - 4.3 Convolución circular: definición y relación con la DFT
  - 4.4 La DFT en Matlab

- Comentarios:
  - Bibliografía básica y complementaria: [BB2: Opp&Sch] Cap. 8, Secs. 8.0-8.5; [BB3: McC&Sch&Yod] Cap. 13, Secs. 13.0-13.5

Propiedades

- ¿Para qué? ¿Por qué son útiles?
  - Para calcular DFTs de señales que se pueden expresar a partir de otras
  - Para entender qué ocurre cuando realizamos ciertas operaciones (e.g., desplazar en el tiempo equivale a …)
  - ¡¡Para calcular convoluciones!! → Apartado 4.3
  - La DFT se parece a una TF muestreada → Parecidas a las de la TF

- Propiedad 1: linealidad

$$x_1[n]\text{: long. } N_1 \quad \overset{DFT}{\leftrightarrow} \quad X_1[k]$$

$$x_2[n]\text{: long. } N_2 \quad \overset{DFT}{\leftrightarrow} \quad X_2[k]$$

$$x_3[n] = a x_1[n] + b x_2[n] \rightarrow \text{ long } N_3=\max\{N_1,N_2\} \quad \overset{DFT}{\leftrightarrow} \quad X_3[k] = a X_1[k] + b X_2[k]$$

¡¡Las DFTs de $x_1[n]$ y $x_2[n]$ se tienen que calcular con la misma longitud $N \geq N_3$!!

Para $N > N_3$, se aumentan con ceros las longitudes de las secuencias

Propiedades

- Propiedad 2: desplazamiento circular
  - Si desplazamos en el tiempo ¿multiplicamos la DFT por una exponencial?
    - $x[n]=\delta[n]$, $N=4 \rightarrow X[k]=1,\ k=0,1,2,3$
    - $y[n]=x[n-6]=\delta[n-6]$, $N=4 \rightarrow Y[k]=0,\ k=0,1,2,3$

Nos da cero porque y[n] "se sale" del intervalo [0,3]

  - Tendríamos que desplazar pero conseguir que la señal siguiera estando definida en el intervalo $[0, N-1] \rightarrow$ desplazamiento circular de orden N

Aritmética módulo N:
$((q))_N$ es el resto de dividir q entre N
Ejs: $((2))_{10}=2$; $((13))_{10}=3$;
$((21))_{10}=1$; $((-1))_{10}=9$

$$x[n]\text{: long. } N \quad \overset{DFT}{\leftrightarrow} \quad X[k]$$

$$x[((n-n_0))_N],\ \ 0 \le n \le N-1 \quad \overset{DFT}{\leftrightarrow} \quad X[k]e^{-j(2\pi k/N)n_0}$$

  - En vez de utilizar la aritmética módulo N, las operaciones circulares son más fáciles si utilizamos la extensión periódica de la señal

Recordemos que la extensión periódica de periodo N se define como

$$\tilde{x}_N[n] = \sum_{r=-\infty}^{\infty} x[n-rN]$$

Propiedades

- ¿Cómo podemos utilizarla para hacer un desplazamiento circular de orden N?

- Objetivo: $z[n] = x[((n-n_0))_N]$

- Paso 1: hacemos la extensión periódica de periodo N

$$\tilde{x}_N[n] = \sum_{r=-\infty}^{\infty} x[n - rN]$$

- Paso 2: desplazamos la señal periódica linealmente un valor n_0

$$y[n] = \tilde{x}_N[n-n_0]$$

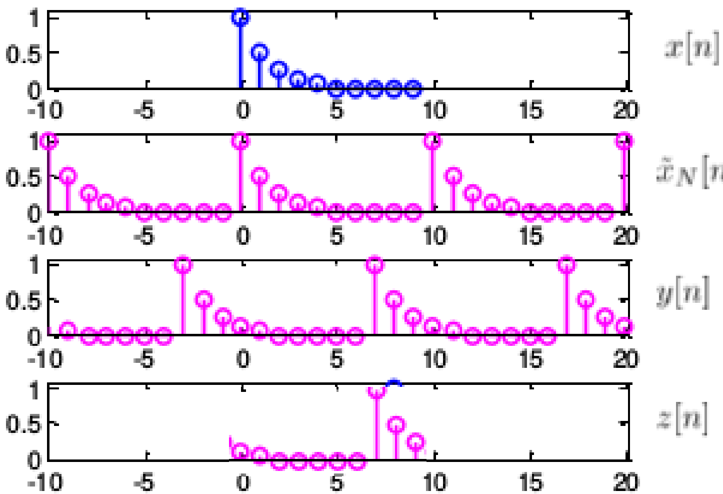
- Paso 3: nos quedamos con los valores de la señal entre 0 y N-1, es decir, enventanamos.

$$z[n] = y[n] \cdot w_N[n] = \tilde{x}_N[n-n_0] \cdot w_N[n]$$

$$w_N[n] = \begin{cases} 1 & n = 0, ..., N-1 \\ 0 & \text{resto} \end{cases}$$

Ejemplo:

$$z[n]=x[((n-7))_{10}]$$



Propiedades

- Propiedad 3: dualidad

x[n]: long. N $\overset{DFT}{\leftrightarrow}$ X[k]

X[n], $\overset{DFT}{\leftrightarrow}$ $N \cdot x[((-k))_N]$, $0 \le k \le N-1$

- Propiedad(es) 4: simetrías

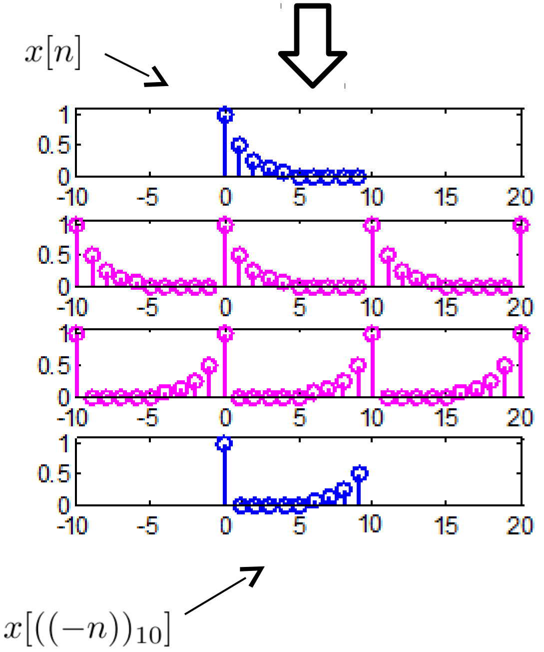
$x^*[n]$ $\overset{DFT}{\leftrightarrow}$ $X^*[((-k))_N]$, $0 \le k \le N-1$

$x^*[((-n))_N]$, $0 \le n \le N-1$ $\overset{DFT}{\leftrightarrow}$ $X^*[k]$, $0 \le k \le N-1$

si x[n] es real: $X[k] = X^*[((-k))_N]$

Otra vez aritmética módulo: $((-1))_8=7$; $((-3))_8=5$; …

Al igual que antes, si nos resulta más sencillo, podemos hacer la extensión periódica, abatir y quedarnos con la señal resultante en el intervalo [0,N-1]



Propiedades

- Propiedad 5: convolución circular

$$x_1[n]\text{: long. N} \quad \overset{DFT}{\leftrightarrow} \quad X_1[k]$$

$$x_2[n]\text{: long. N} \quad \overset{DFT}{\leftrightarrow} \quad X_2[k]$$

$$x_3[n] = x_1[n] \; \textcircled{N} \; x_2[n] \quad \overset{DFT}{\leftrightarrow} \quad X_3[k] = X_1[k].\, X_2[k]$$

(long. N)

Convolución circular de N puntos (ya veremos cómo se define → 4.3)

- Propiedad 6: multiplicación (modulación)

$$x_1[n]\text{: long. N} \quad \overset{DFT}{\leftrightarrow} \quad X_1[k]$$
$$x_2[n]\text{: long. N} \quad \overset{DFT}{\leftrightarrow} \quad X_2[k]$$

$$x_3[n] = x_1[n]\cdot x_2[n] \quad \overset{DFT}{\leftrightarrow} \quad X_3[k] = (1/N)\, X_1[k] \; \textcircled{N} \; X_2[k]$$

- La conv. circular es tan importante que se le dedica el siguiente apartado (4.3)

Ubicándonos

- **<u>Tema 4</u>: Fundamentos de la Transformada Discreta de Fourier**
  - 4.1 Definición: la DFT como el muestreo de la TF
  - 4.2 Propiedades
  - **<u>4.3 Convolución circular: definición y relación con la DFT</u>**
  - 4.4 La DFT en Matlab

- Comentarios:
  - Bibliografía básica y complementaria: [BB2: Opp&Sch] Cap. 8, Secs. 8.0-8.5; [BB3: McC&Sch&Yod] Cap. 13, Secs. 13.0-13.5

Convolución circular definición

Nótese que para $0 \leq n \leq N-1$:
$((-n))_N = N-n$
$((n))_N = n$,

- Definición: la convolución circular de longitud N es

$$x_3[n] = x_1[n] \circledast_N x_2[n] \rightarrow \quad x_3[n] = \sum_{m=0}^{N-1} x_2[m] x_1[((n-m))_N], \quad 0 \leq n \leq N-1$$

  - Importante: (1) hay que fijar la longitud, (2) solo está definida en el intervalo [0,N-1]
  - Propiedades: conmutativa, asociativa, distributiva,…

- ¿Otra convolución más? ¿Por qué?
  - Porque es importante para entender lo que está pasando con las DFTs, que como hemos dicho es la única transformada que se puede calcular de forma eficiente con un ordenador.
  - Porque en determinadas ocasiones es equivalente a una convolución lineal (que es la clásica).
  - Porque se puede calcular de forma muy eficiente en un ordenador.

Calculando convoluciones circulares

- La aritmética módulo es complicada, ¿podemos calcular convoluciones circulares de forma más sencilla utilizando la extensión periódica? → Sí
  - Objetivo: $z[n] = v[n] \, (N) \, x[n]$
  - Paso 1: enventanamos v[n] y x[n] entre 0 y N-1

$$v_N[n] = v[n] \cdot w_N[n] \qquad x_N[n] = x[n] \cdot w_N[n]$$

$$w_N[n] = \begin{cases} 1 & n = 0, ..., N-1 \\ 0 & \text{resto} \end{cases}$$

  - Paso 2: hacemos la extensión periódica de periodo N de una de las dos señales

$$\tilde{x}_N[n] = \sum_{r=-\infty}^{\infty} x_N[n - rN]$$

  - Paso 3: la convolucionamos linealmente con la otra señal

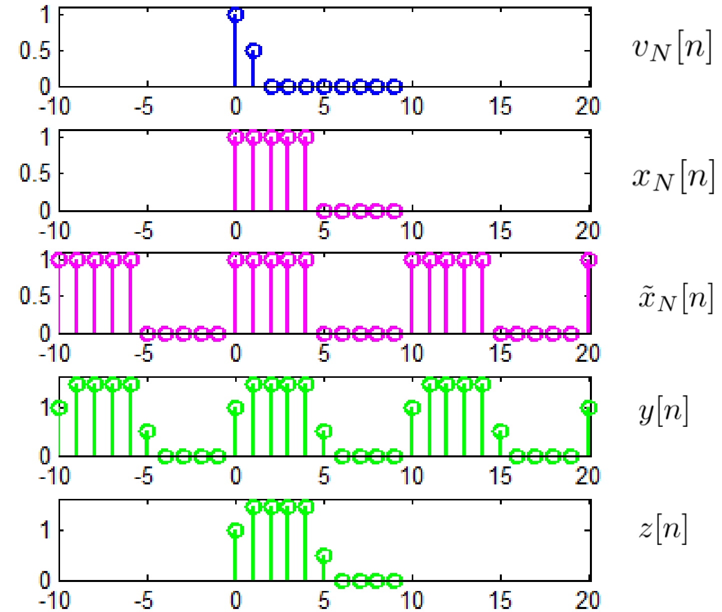
$$y[n] = v_N[n] * \tilde{x}_N[n]$$

  - Paso 4: nos quedamos con los valores de la señal entre 0 y N-1

$$z[n] = y[n] \cdot w_N[n] = \left(v[n] * \tilde{x}_N[n]\right) \cdot w_N[n]$$

Ejemplo:

$$v[n] = \delta[n] + 0.5\delta[n-1]$$
$$x[n] = u[n] - u[n-5]$$

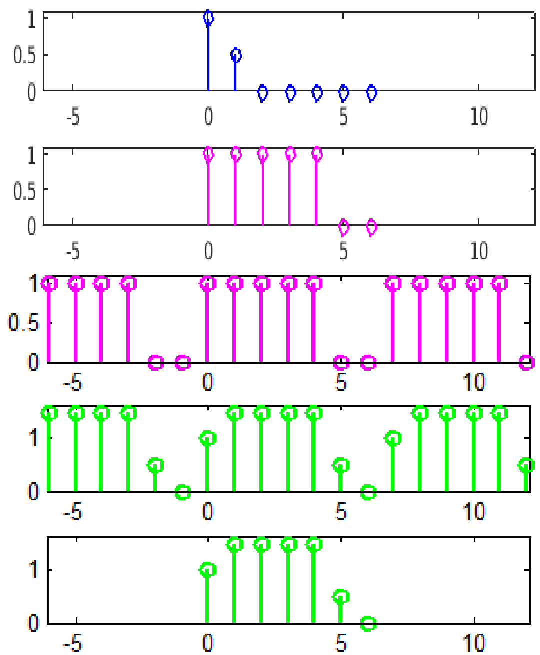


N=10

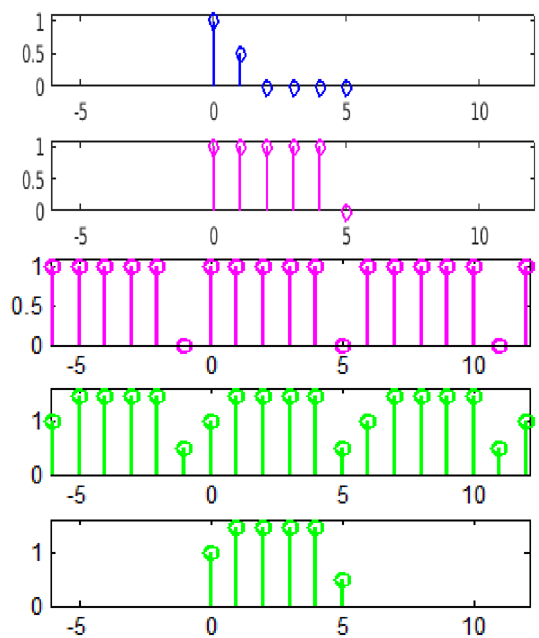
Calculando convoluciones circulares

- En el ejemplo anterior, la convolución circular y la convolución “normal” (llamada convolución lineal) coinciden , ¿esto es siempre así? No → Repitamos el ejemplo con otros valores de N

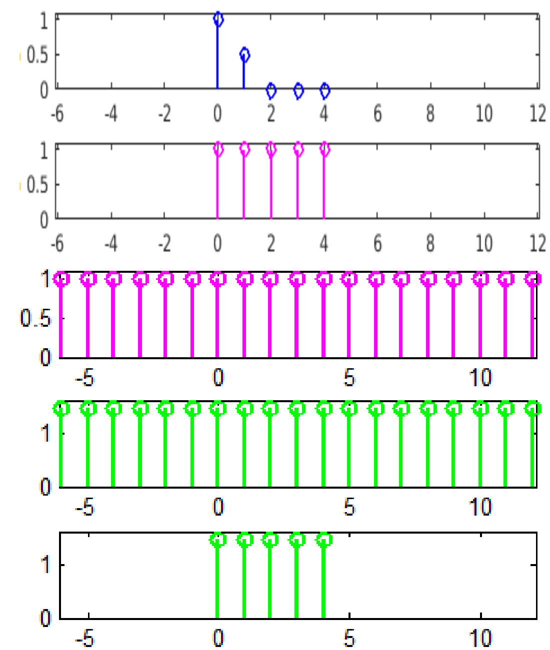
$$z[n] = v[n] \; N \; x[n]$$



N=7



N=6



N=5

Convolución circular vs. convolución lineal

- ¿Diferencias entre convolución circular y lineal?
  - La lineal puede dar lugar a señales de longitud infinitas
  - La circular únicamente da resultados entre 0 y N-1

- Evidentemente no son iguales pero ¿cuándo coinciden?

$$z[n] = v[n] \; N \; x[n] \qquad\qquad y[n] = v[n] * x[n]$$

Paso 1

- Condición 1: v[n] y x[n] tienen que ser finitas y definidas entre 0 y N-1

Paso 4

- Condición 2: el resultado de la lineal (y[n]) tiene que estar definido entre 0 y N-1

- La condición 1 es fácil de comprobar pero ¿cómo comprobamos la condición 2? ¿Tenemos que calcular y[n] para saber si la condición 2 se cumple?

  → La clave está en conocer la longitud de la señal resultante de la convolución lineal, es decir, de y[n]

Convolución circular vs. convolución lineal

- **Recapitulando:** $z[n] = v[n] \, \textcircled{N} \, x[n]$ $y[n] = v[n] * x[n]$

  - Supongamos que v[n] y x[n] son de longitud finita y que empiezan y acaban en…

$$v[n] \neq 0 \text{ solo si } n \in [N_{i,v}, N_{f,v}] \qquad x[n] \neq 0 \text{ solo si } n \in [N_{i,x}, N_{f,x}]$$

→ *long de v[n] =* $N_{f,v} - N_{i,v} + 1$

→ *long de x[n] =* $N_{f,x} - N_{i,x} + 1$

Si: $N_{i,v} \geq 0 \text{ y } N_{i,x} \geq 0$, $N_{f,v} \leq N-1 \text{ y } N_{f,x} \leq N-1$ → *Se cumple la condición 1*

  - En este caso, y[n] sería finita y empezaría y acabaría en:

long de y[n] = long de v[n] + long x[n] – 1 = $N_{f,v} + N_{f,x} - N_{i,v} - N_{i,x} + 1$*, es decir,*

$$y[n] \neq 0 \text{ solo si } n \in [N_{i,y}, N_{f,y}], \qquad \text{con } N_{i,y} = N_{i,v} + N_{i,x} \text{ y } N_{f,y} = N_{f,v} + N_{f,x}$$

$$\boxed{\text{Si} \quad N_{i,v} \geq 0 \quad N_{i,x} \geq 0 \quad N_{f,v} + N_{f,x} \leq N-1 \quad \Rightarrow \quad z[n] = y[n]}$$

Es decir,

Si N ≥ long de y[n] → z[n] = y[n]

Convolución circular vs. convolución lineal

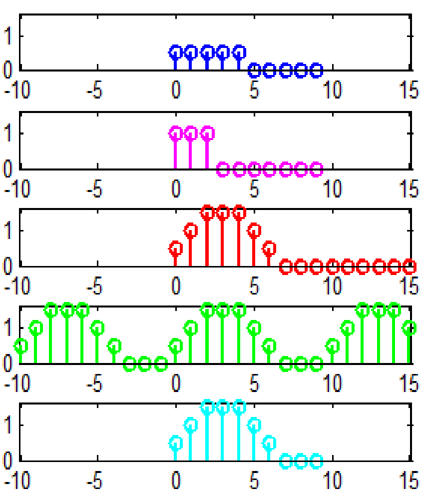
- **Conclusión:** Si N ≥ long de la convolución lineal entre v[n] y x[n] $\Rightarrow$ $v[n] \textcircled{N} x[n] = v[n] * x[n]$

- **Ejemplos:** $v[n] = 0.5(u[n] - u[n-5])$ $\quad x[n] = u[n] - u[n-3]$

long de v[n] = 5 *long de x[n] = 3* → *long de la Conv. Lineal = 5 + 3 - 1 = 7*

N=10

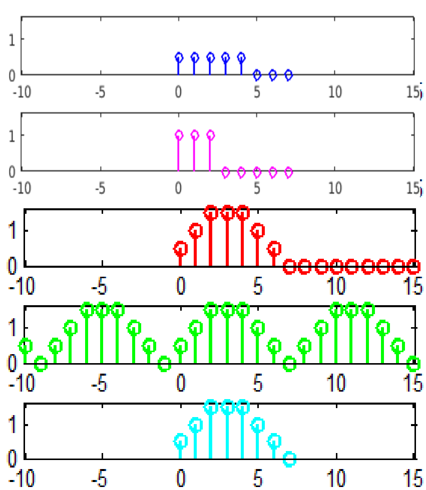
¿10 ≥ 7? → **OK**



Conv. Circ = Conv. Lineal

N=8

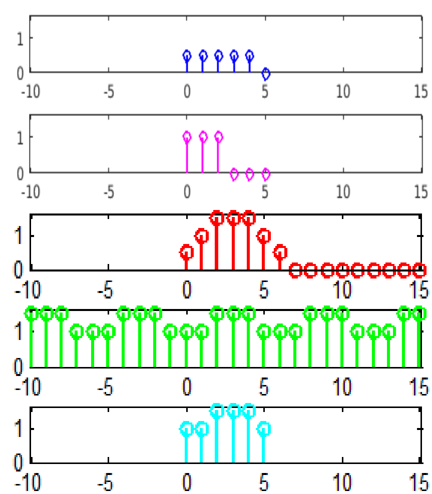
¿8 ≥ 7? → **OK**



Conv. Circ = Conv. Lineal

N=6

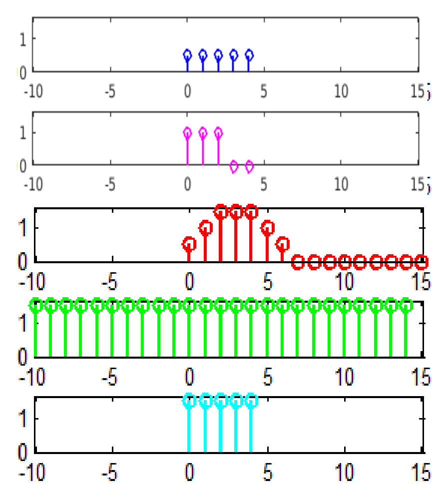
¿6 ≥ 7? → **KO**



Conv. Circ ≠ Conv. Lineal

N=5

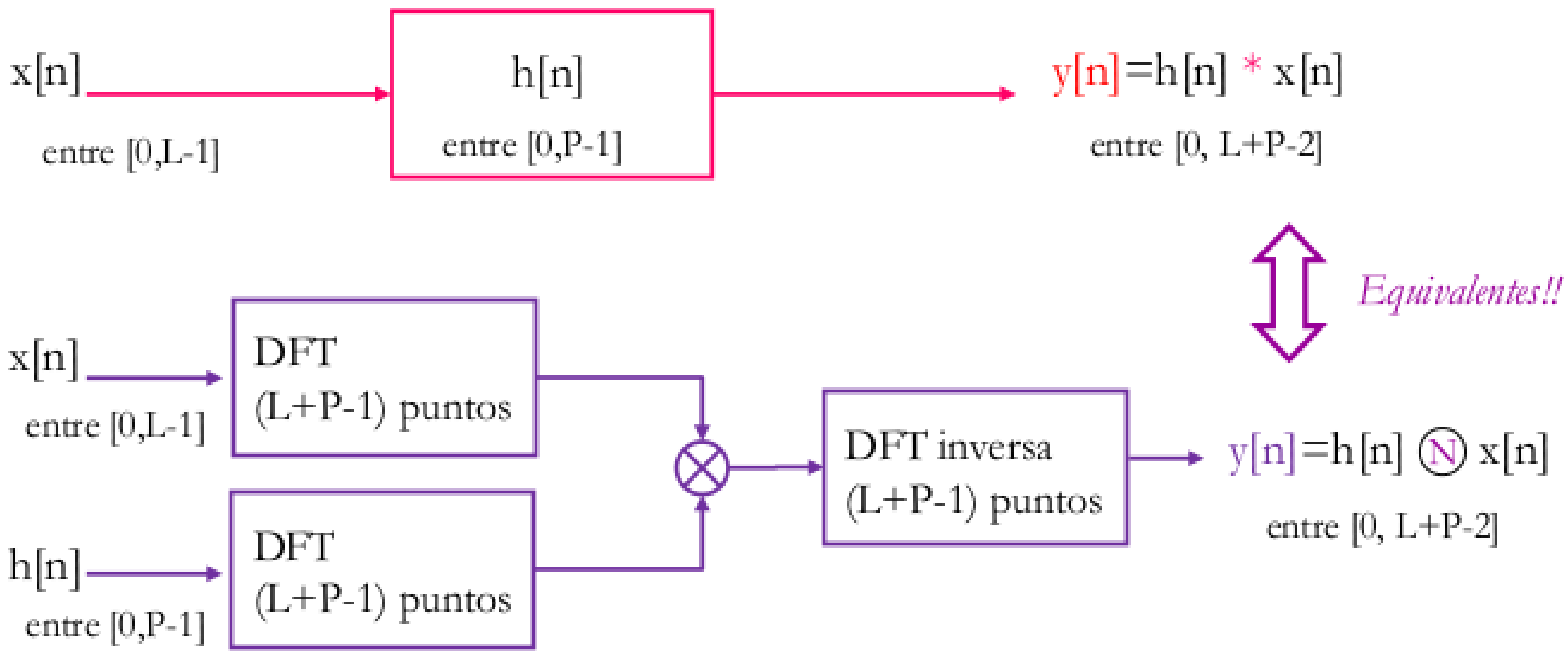
¿5 ≥ 7? → **KO**



Conv. Circ ≠ Conv. Lineal

Convolución circular a través de DFTs

- Si convlin=concirc y convcirc se puede calcular a través de DFTs → convlin también se puede calcular a través de DTFs. Veamos cómo:



- ¿Por qué es útil el 2º esquema? Parece más complicado → Complej.computacional
  - Convlin: complejidad proporcional a N^2
  - Convcirc a través de la fórmula: complejidad prop. a N^2 $x_3[n] = \sum_{m=0}^{N-1} x_2[m]x_1[((n-m))_N],\ 0 \le n \le N-1$
  - Convcirc a través de FFTs: complejidad prop. a $\log_2(N)\cdot N$

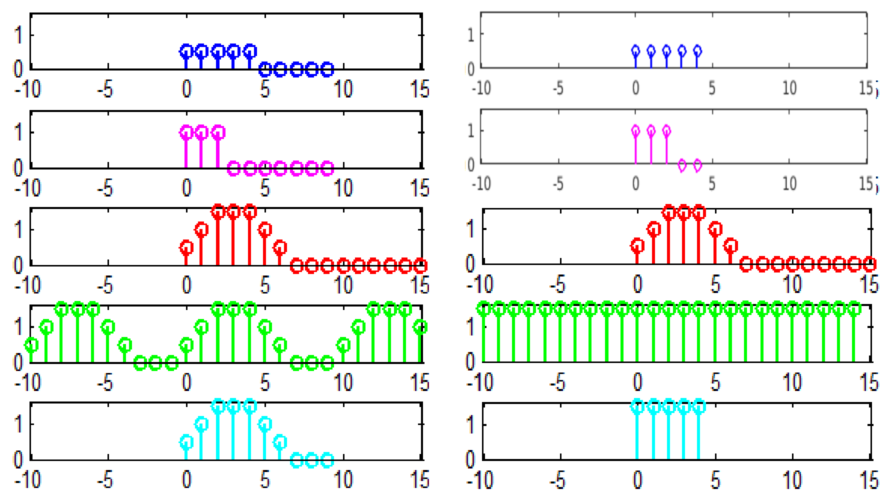
Convolución circular a través de DFTs

- Ejemplo:
  - Habíamos visto que para las señales:

$$v[n] = 0.5(u[n] - u[n-5]) \qquad x[n] = u[n] - u[n-3]$$

→ convolución circular = convolución lineal para N ≥ 7

  - El resultado para $N=10$ y para $N=5$ fue



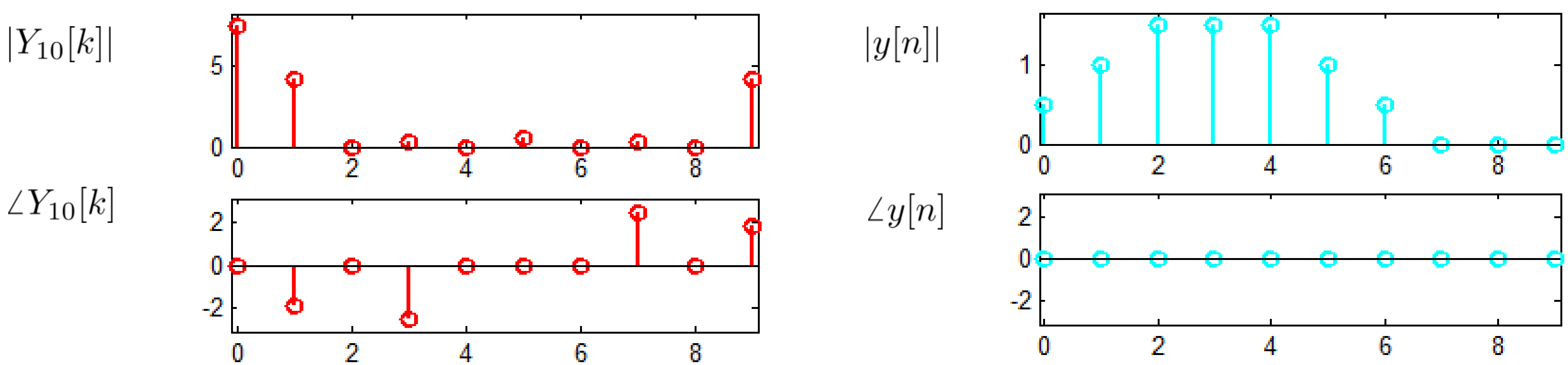
- Comprobaremos que al hacer las DFTs, multiplicarlas y hacer la DFT inversa obtenemos la convolución circular.

Convolución circular a través de DFTs

- Caso 1: DFT de longitud N=10
  - Las señales son $v[n] = 0.5(u[n] - u[n-5])$ $\quad x[n] = u[n] - u[n-3]$
  - En ambos casos son pulsos, ya habíamos calculado la DFT de un pulso

$$V_{10}[k] = 0.5\frac{\sin\left(\frac{5\pi}{10}k\right)}{\sin\left(\frac{\pi}{10}k\right)}e^{-j\frac{4\pi}{10}k}, \quad k = 0, \ldots, 9 \qquad X_{10}[k] = \frac{\sin\left(\frac{3\pi}{10}k\right)}{\sin\left(\frac{\pi}{10}k\right)}e^{-j\frac{2\pi}{10}k}, \quad k = 0, \ldots, 9$$

$$Y_{10}[k] = V_{10}[k] \cdot X_{10}[k], \quad k = 0, \ldots, 9 \qquad y[n] = \frac{1}{10}\sum_{k=0}^{9} Y_{10}[k]e^{j\frac{2\pi}{10}kn}, \quad n = 0, \ldots, 9$$

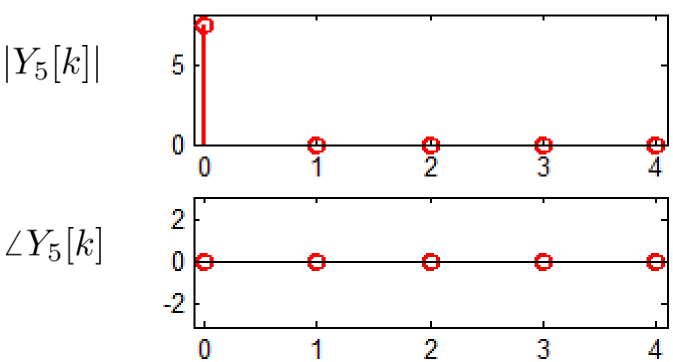


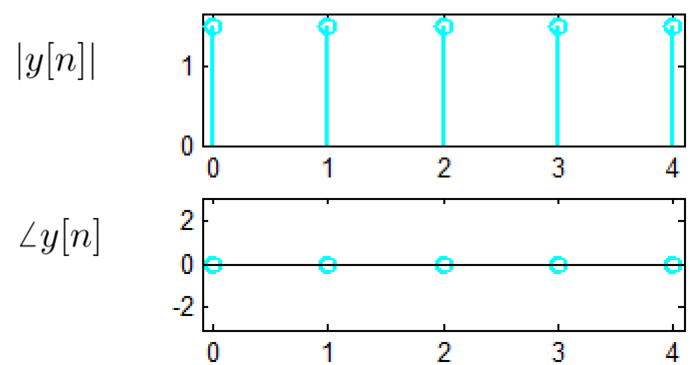
Convolución circular a través de DFTs

- Caso 2: DFT de longitud N=5
  - Las señales son pulsos $v[n] = 0.5(u[n] - u[n-5])$ $\quad x[n] = u[n] - u[n-3]$
  - Y las DFTs sincs con fase lineal

$$V_5[k] = 0.5\frac{\sin\left(\frac{5\pi}{5}k\right)}{\sin\left(\frac{\pi}{5}k\right)}e^{-j\frac{4\pi}{5}k} = 2.5\delta[k], \quad k = 0, \ldots, 4 \qquad X_5[k] = \frac{\sin\left(\frac{3\pi}{5}k\right)}{\sin\left(\frac{\pi}{5}k\right)}e^{-j\frac{2\pi}{5}k}, \quad k = 0, \ldots, 4$$

$$Y_5[k] = V_5[k] \cdot X_5[k], \quad k = 0, \ldots, 4 \qquad y[n] = \frac{1}{5}\sum_{k=0}^{4} Y_5[k]e^{j\frac{2\pi}{5}kn}, \quad n = 0, \ldots, 4$$

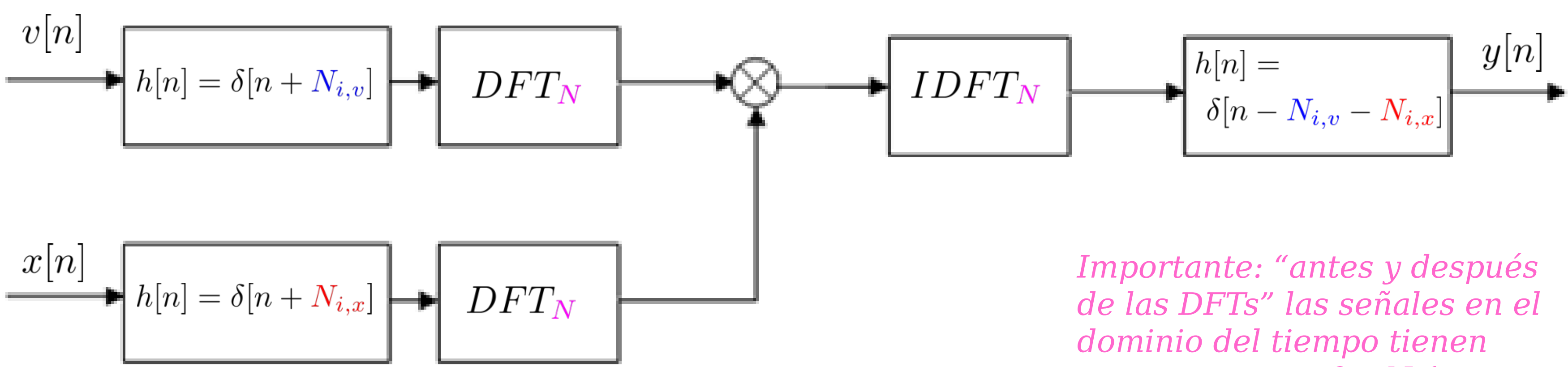




Convolución circular&lineal a través de DFTs

- ¿Y qué pasa si v[n] y/o x[n] son de longitud finita pero empiezan antes de cero? → Estrictamente lineal y circular no coinciden, pero…

$v[n] \neq 0$ en $n \in [N_{i,v}, N_{f,v}]$, con $N_{i,v} < 0$ $\quad$ $x[n] \neq 0$ en $n \in [N_{i,x}, N_{f,x}]$, con $N_{i,x} < 0$



Importante: "antes y después de las DFTs" las señales en el dominio del tiempo tienen que estar entre 0 y N-1

Tras desplazarlas, las señales empiezan en cero y acaban en

$$N_{i,v} + N_{f,v}$$
$$N_{i,v} + N_{f,v}$$

La convolución empieza en cero y acaba en

$$N_{i,v} + N_{f,v} + N_{i,v} + N_{f,v}$$

La longitud de la DFT debe satisfacer

$$N \geq (N_{i,v} + N_{f,v} + N_{i,v} + N_{f,v} + 1)$$

La longitud de la DFT debe ser mayor que la suma de las longitudes de las señales

Convolución por bloques

- Consideremos una señal $x[n]$ definida entre 0 y $N\text{-}1$ a la entrada de un sistema LTI, caracterizado por una $h[n]$ definida entre 0 y $P\text{-}1$ → La salida vendrá dada por la convolución lineal entre ambas y estará entre 0 y $(N+P\text{-}2)$.

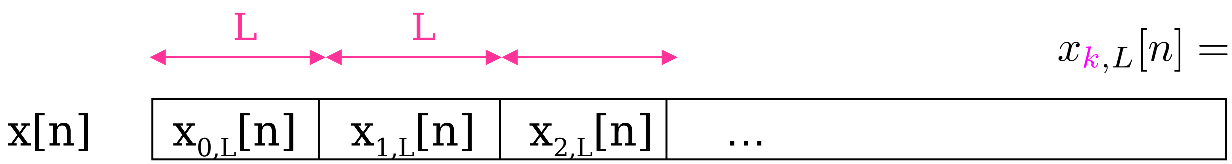
- Hemos visto que para poder calcular dicha convolución a través de DFTs necesitamos DFTs de long. $(N+P\text{-}1)$

- Problema: en muchas aplicaciones (p.e., procesado de voz), la long. de $x[n]$ es indefinida (muy larga) ⇒ no podemos calcular la DFT de toda la señal $x[n]$, y si almacenamos $x[n]$ para calcular después su DFT, introduciríamos retardos grandes de procesamiento, además de tener un tiempo de respuesta del sistema muy elevado.

- Solución: tranformar la convolución inicial en una secuencia de convoluciones más cortas → convolución por bloques

Convolución por bloques

- Si dividimos la señal en bloques



$$x_{k,L}[n] = x[n] \cdot w_L[n - kL]$$

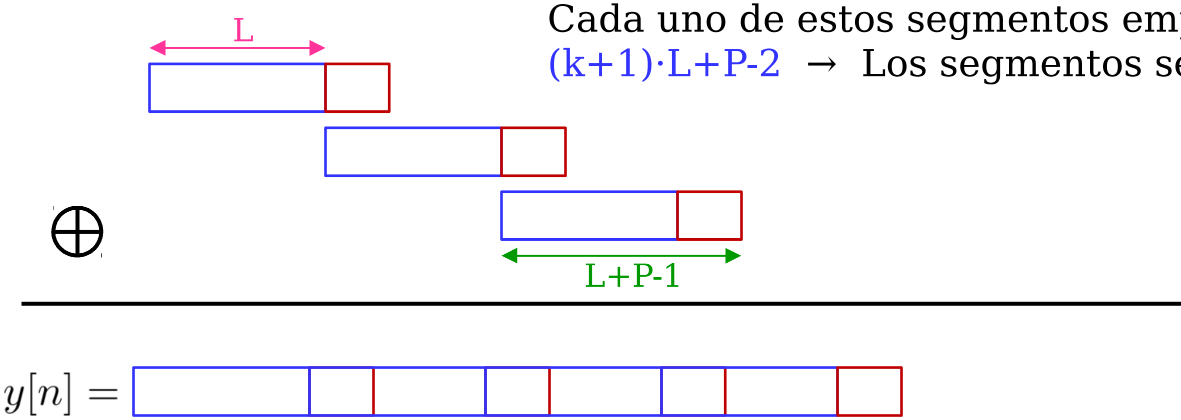
ventana

  - Podemos expresarla como $x[n] = \sum_{k=0}^{\infty} x_{k,L}[n]$

- Para calcular $y[n] = h[n] * x[n]$ utilizamos que la convolución es distributiva

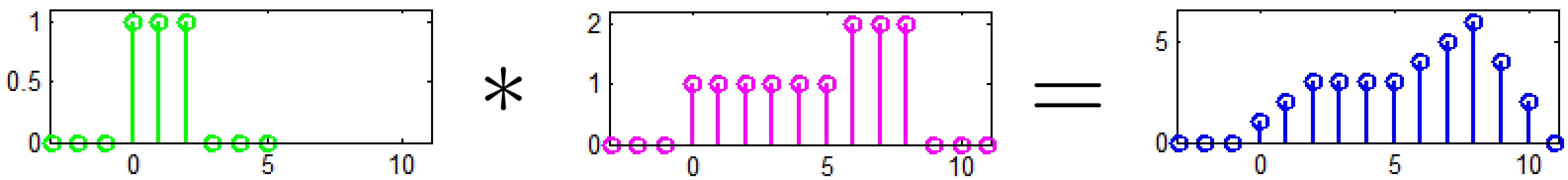
$$y[n] = h[n] * x[n] = h[n] * \left(\sum_{k=0}^{\infty} x_{k,L}[n] \right) = \sum_{k=0}^{\infty} \underbrace{\left(h[n] * x_{k,L}[n] \right)}$$

Cada uno de estos segmentos empieza en k·L y acaba en (k+1)·L+P-2 → Los segmentos se solapan

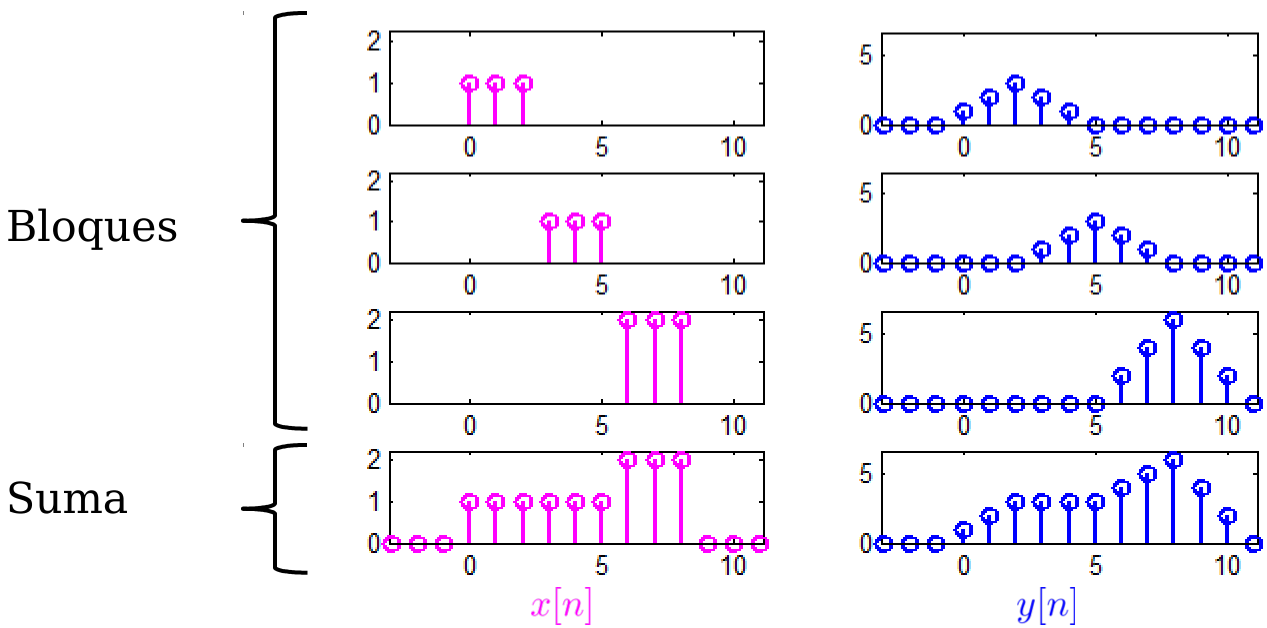


Esquema convolución por bloques

- Ejemplo: $y[n] = h[n] * x[n]$

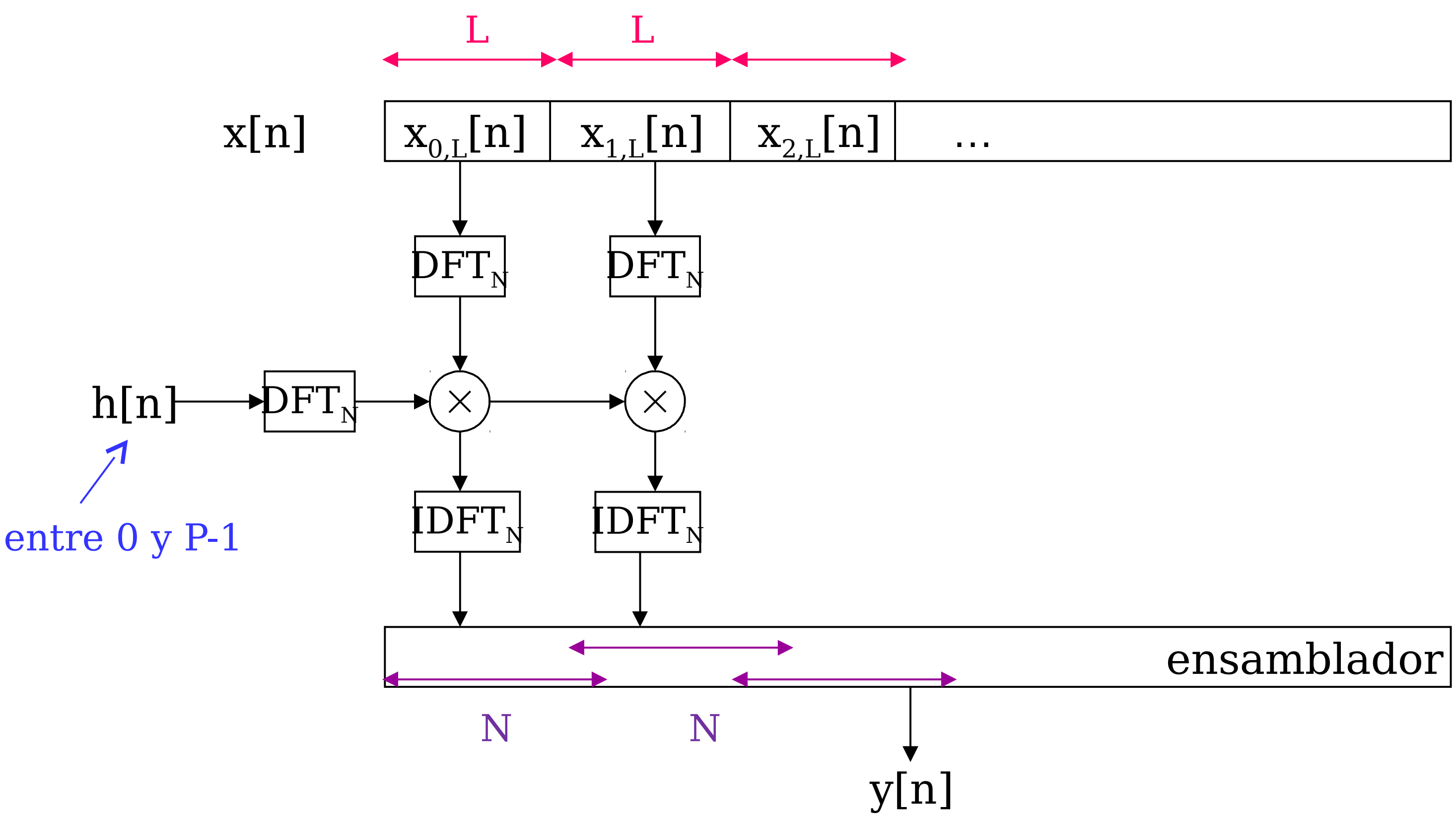


- Si lo hacemos utilizando bloques de tamaño $L=3$



Convolución por bloques

- Para ahorrar coste computacional, las convoluciones se hacen con DFTs



- Tamaño segmento L; tamaño respuesta al impulso P; tamaño DFT $N \geq L+P-1$
- Los bloques deben separarse L puntos → deben solaparse $N-L$ puntos

Ubicándonos

- **Tema 4: Fundamentos de la Transformada Discreta de Fourier**
  - 4.1 Definición: la DFT como el muestreo de la TF
  - 4.2 Propiedades
  - 4.3 Convolución circular: definición y relación con la DFT
  - **4.4 La DFT en Matlab**

- Comentarios:
  - Bibliografía básica y complementaria: [BB2: Opp&Sch] Cap. 8, Secs. 8.0-8.5; [BB3: McC&Sch&Yod] Cap. 13, Secs. 13.0-13.5

Función en Matlab que calcula la DFT

- No existen la funciones `dft,idft`, sino las funciones `fft`, `ifft`

```
  FFT(x,N) is the N-point FFT, padded with zeros if x has less
than N points and truncated if it has more.

For length N input vector x, the DFT is a length N vector X,
with elements
                 N
   X(k) =       sum  x(n)*exp(-j*2*pi*(k-1)*(n-1)/N), 1 <= k <= N.
                n=1
The inverse DFT (computed by IFFT) is given by
                 N
   x(n) = (1/N) sum  X(k)*exp( j*2*pi*(k-1)*(n-1)/N), 1 <= n <= N.
                k=1
```

- Le damos el valor de las amplitudes y nos devuelve valor de las amplitudes
  - ¿En qué intervalo de tiempo supone Matlab que está definida la señal x?
  - ¿Cuáles son los ejes de las amplitudes que nos devuelve fft?

Función en Matlab que calcula la DFT

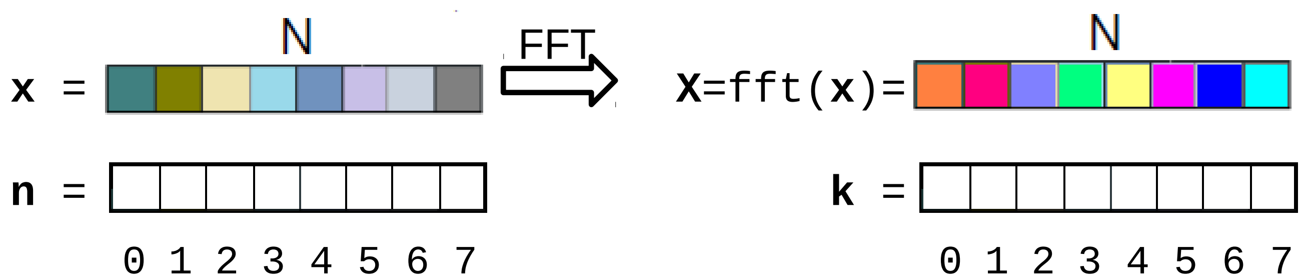
- ¿Cuáles son los ejes?
  - Matlab

```
                 N
X(k) =          sum  x(n)*exp(-j*2*pi*(k-1)*(n-1)/N), 1 <= k <= N.
                n=1
```

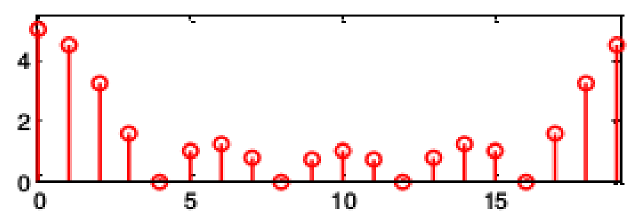
  - Form. teórica

```
                N-1
X[k] =          sum  x[n]*exp(-j*2*pi*k*n/N), 0 <= k <= N-1.
                n=0
```

  - Conclusión



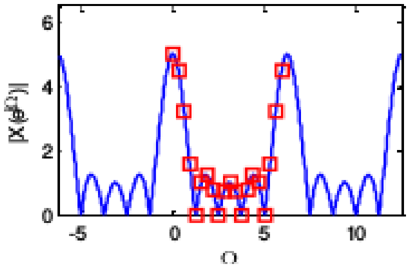

```
x=[1 1 1 1 1 zeros(1,15)];
X=fft(x,N);
k=0:(N-1);
figure; stem(k,abs(X));
```

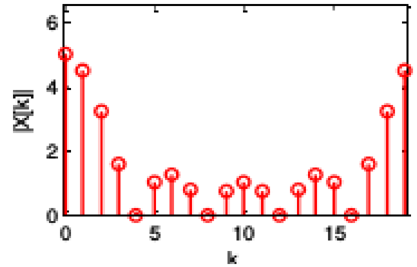


Estimando la TF de una señal discreta

- Asumimos que la señal está definida en $[0,N-1]$ *(If not → enventanado)*

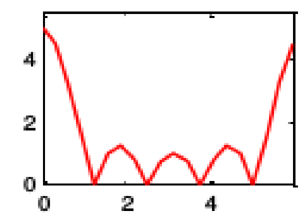
  - La DFT equivale a tomar N muestras equiespaciadas en el intervalo $[0,2\pi)$





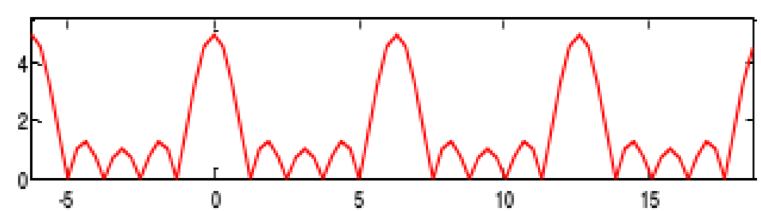
  - Por tanto:

```
X=fft(x,N);
Omega=(0:(N-1))*(2*pi/N);
figure; plot(Omega,abs(X));
```



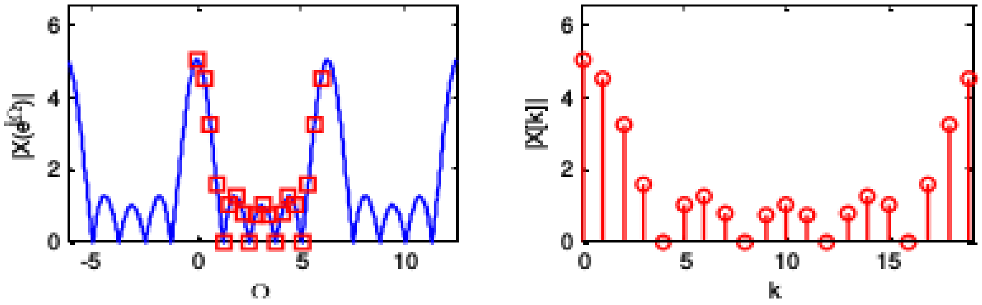
  - ¿Qué pasa si queremos dibujar varios periodos?

```
X=fft(x,N); X_per=[X, X, X, X]
Omega=(-N:(3*N-1))*(2*pi/N);
figure; plot(Omega,abs(X_per));
```

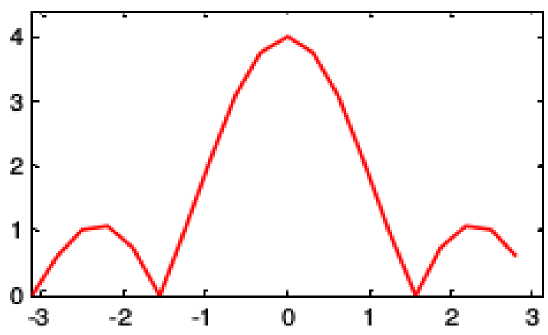


%Utilizamos plot (continua) y no stem

Estimando la TF de una señal discreta



- ¿Qué pasa si queremos dibujarla en el intervalo [-π,π)?
  - `X=fft(x,N); X_cent=fftshift(X);`
  - `Omega = ((-N/2):(N/2-1))*(2*pi/N);`
  - `figure; plot(Omega,abs(X_cent));`



Estimando la TF de una señal continua

- ¿Y qué ocurre con la TF de señales continuas?
  - Paso 1: Utilizamos tema muestreo para relacionar TF de señal continua con TF de señal discreta → 3 efectos: 1) amplitud modificada por $1/T_s$, 2) expansión por T_s del eje de frecuencias, 3) réplicas cada 2π
  - Paso 2: Utilizamos las transparencia anteriores para relacionar TF de señal discreta con DFT de señal discreta (misma amplitud, eje frec. muestreado)
- Por tanto:
  - A) Eje vertical (amplitud):
    Paso 1 multiplica por $1/T_s$, Paso 2 no hace nada → Hay que multiplicar la amplitud de la DFT por T_s
  - B) Eje horizontal (frecuencia):
    - Paso 1: al muestrear las frecuencias entre 0 y $\omega_s/2$ pasan a estar entre 0 y π
    - Paso 2: Al hacer la DFT las frecuencias entre 0 y π constituyen los primeros $N/2$ puntos de la DFT
    - Paso 1 + 2: las $N/2$ primeros puntos de la DFT corresponden a tomar $N/2$ muestras de la TF de la señal continua entre 0 y $\omega_s/2$

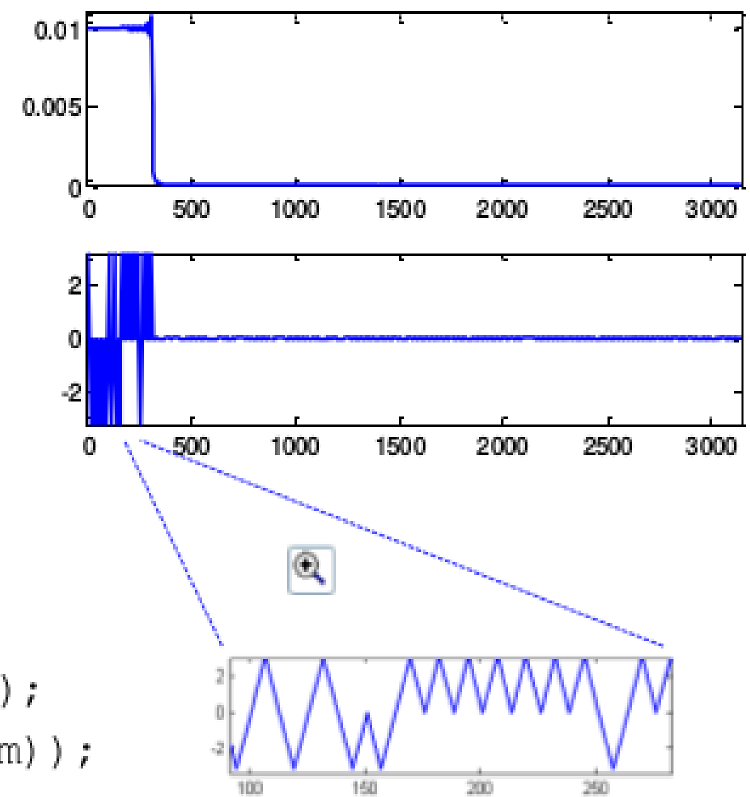
Estimando la TF de una señal continua

- Si queremos estimar la TF de la señal continua

```
➢ Ts=.001; t=0:Ts:.999; N=length(t); x=sinc((t-.5)/.01);

➢ X=fft(x,N);
➢ Omega=(0:(N-1))*(2*pi/N);
➢ %Esta sería la frec. discreta

➢ Fs=1/Ts; Ws=2*pi*Fs;
➢ X_m = Ts*X(1:N/2);
➢ omega = Omega(1:N/2)/Ts;
➢ %También omega = Omega(1:N/2)*fs;
➢ %También omega = (0:(N/2-1))*Ws/N;
```

Resolución espectral (rads/seg)

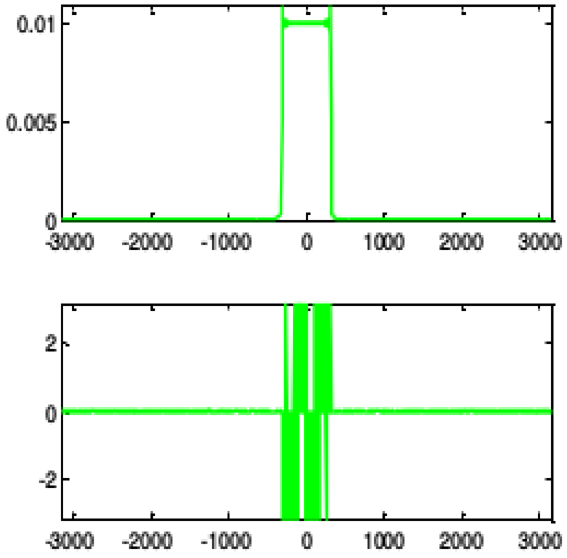
```
➢ figure;
➢ subplot(2,1,1); plot(omega,abs(X_m));
➢ subplot(2,1,2); plot(omega,angle(X_m));
➢ %Ojo, pintamos la mitad
```



Estimando la TF de una señal continua

- Si queremos estimar también la parte negativa de la TF

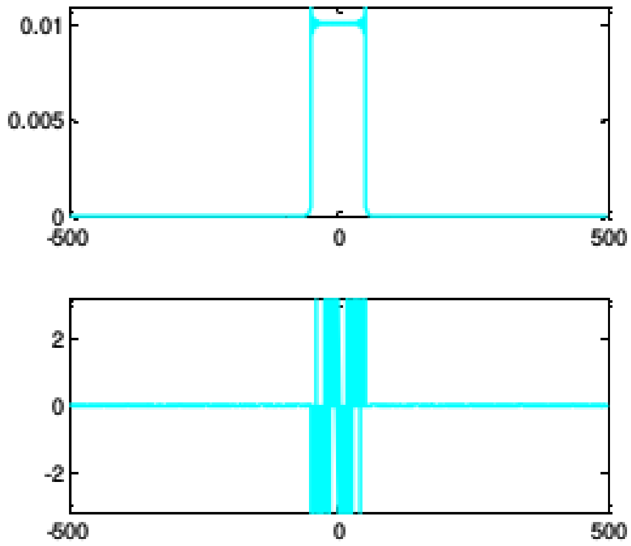
```
Ts=.001; t=0:Ts:.999; N=length(t); x=sinc((t-.5)/.01);

X=fft(x,N); X_c = fftshift(X);
Omega=(-N/2:(N/2-1))*(2*pi/N);
%También Omega=-pi:(2*pi/N):pi;
%(sigue) Omega=Omega(1:end-1);

X_c = Ts*X_c; omega = Omega/Ts;
%También omega = -(Ws/2):(Ws/N):(Ws/2);
% (sigue) omega=omega(1:end-1);

figure;
subplot(2,1,1);
plot(omega,abs(X_c));
subplot(2,1,2);
plot(omega,angle(X_c));
```



Estimando la TF de una señal continua

- Si queremos que las unidades de la TF sean Hz

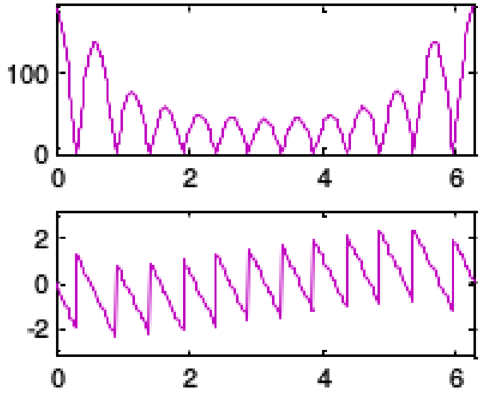
```
➢ Ts=.001; t=0:Ts:.999; N=length(t); x=sinc((t-.5)/.01);

➢ X=fft(x,N);
➢ X_c = Ts*fftshift(X);

➢ Fs=1/Ts;

➢ f=(-Fs/2):(Fs/N):(Fs/2);
➢ f=f(1:end-1);

➢ figure;
➢ subplot(2,1,1);
➢ plot(f,abs(X_c));
➢ axis([-Fs/2 Fs/2 0 max(abs(X_m))])
➢ subplot(2,1,2);
➢ plot(f,angle(X_c));
➢ axis([-Fs/2 Fs/2 -pi pi])
```



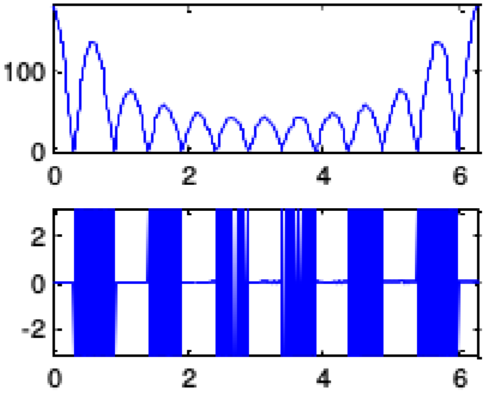
DFT en Matlab: otros apectos

- ¿Qué ocurre si tenemos una señal fuera del intervalo $[0,N-1]$?
  - Tenemos que corregirlo nosotros, desplazándola para llevarla al intervalo $[0,N-1]$
  - La TF cambia (desplazar en tiempo = multiplicar por una exponencial en frecuencia)

```
n=-6:6; x=n.^2; N=1000; Omega=0:(2*pi/N):(2*pi-2*pi/N);
X=fft(x,N); %Matlab siempre asume que empieza en 0 (no hay que hacer nada)
X_correg = X.*exp(j*6*Omega); %Como no empieza en 0 sino en ==> corregimos
```



¿Cuál es X y cuál es X_correg?




```
figure; subplot(2,1,1); plot(abs(X));
subplot(2,1,2); plot(angle(X));
figure; subplot(2,1,1); plot(abs(X_correg));
subplot(2,1,2); plot(angle(X_correg));
```

DFT en Matlab: otros apectos

- Tanto FFT como IFFT generan por defecto señales complejas → necesitamos dos gráficas para representarlas
  - `subplot(2,1,1); plot(real(X)); subplot(2,1,2); plot(imag (X));`
  - `subplot(2,1,1); plot(abs(X)); subplot(2,1,2); plot(angle(X));`

- Es un algoritmo numérico, así que hay que tener cuidado con los errores numéricos. Para corregirlos:
  - La parte real debería ser cero pero sale un número muy pequeño → nos la "cargamos" y dejamos sólo la parte imaginaria
    - `k_borr=(abs(real(X))<1e-5);`
    - `X(k_borr) = 0 + j*imag(X(k_borr));`
  - La parte imaginaria debería ser cero pero sale un número muy pequeño →dejamos solo parte real
    - `k_borr=(abs(imag(X))<1e-5);`
    - `X(k_borr) = real(X(k_borr)) + 0*j;`

Resumen del tema

- La DFT es una transformación que transforma una señal discreta de longitud finita en otra señal discreta de longitud finita

- La longitud siempre debe especificarse ("DFT de longitud N"), porque su definición depende de N

- Se puede calcular en un ordenador de forma trivial con N^2 multiplicaciones y sumas (si se usa el algoritmo FFT basta con $\log_2(N)\cdot N$)

- Se utiliza para codificación de fuente (JPG), transmisión de información (OFDM) y para estimar la TF de señales
  - Si la DFT tiene longitud N → Se muestrea el intervalo $[0,2\pi)$ con resolución $2\pi/N$
  - La DFT coincide exactamente con la TF muestreada si: (1) la señal empieza en cero y (2) la longitud de la DFT es mayor o igual que la de la señal
  - Si no se cumplen las dos condiciones anteriores (p.e. porque la señal tiene longitud infinita), la DFT coincide con la TF de la señal enventanada

Resumen del tema

- Enventanado
  - Una ventana convierte una señal de longitud infinita en una de longitud finita
  - Modelado: multiplicación en el tiempo, convolución en frecuencia
  - Produce una pérdida de resolución espectral
  - Existen varios tipos de ventanas, en este tema sólo hemos visto la rectangular

- Propiedades de la DFT
  - Muy parecidas a las de la TF
  - Diferencia fundamental: para garantizar que las señales están definidas en el intervalo $[0,N-1]$ hay que utilizar la versión circular de las distintas operaciones (desplazamiento, abatimiento y convolución circular)

- Operaciones circulares
  - Matemáticamente se definen a través de la aritmética módulo
  - Los ordenadores las implementan usando aritmética módulo
  - Nosotros utilizamos la extensión periódica de la señal (más fácil de entender)

Resumen del tema

- Relación entre convolución circular y convolución lineal
  - La convolución circular se calcula habitualmente calculando las DFTs de las señales, multiplicandos las DFTs y calculando la DFT inversa
  - Si tenemos señales de longitud finita definidas a partir de cero → la convolución circular coincide con la lineal si su longitud es mayor que la suma de longitudes de las señales
  - Si una de las señales es de longitud muy larga (teóricamente incluso infinita) → se utiliza la convolución por bloques
    - Se divide la señal larga en bloques/segmentos de longitud fija
    - Se convoluciona cada uno de los segmentos con la otra señal segmentos → más largos
    - Los segmentos resultantes se concatenan con el solapamiento (suma) adecuado

- La DFT en Matlab
  - Utiliza el algoritmo FFT, las funciones se llaman `fft` e `ifft`
  - Asumen siempre que la señal empieza en cero
  - Si no especificamos la longitud, la hace de la misma que la de la señal; si ponemos una menor, ignora las muestras finales; y si ponemos una mayor, hace *zero padding*
  - Si se quiere mostrar el intervalo $[-\pi, \pi)$, hay que hacer `fftshift(fft(x))`