Otros detalles sobre el algoritmo FFT

Para el algoritmo radix-2 se asume que $N = 2^M$

de modo que el # de operaciones $= N \log_2(N)$

$= 2^M (\log_2(2^M))$

$= M 2^M$

¿Qué pasa si esto no se cumple?

1) Calcular DFT directo
2) usar divide y conquistar (si es divisible por 2)
3) usar zeropadding hasta llegar a $N = 2^M$
4) si es divisible por otro factor, de modo que

$N = P \cdot 2^M$ OBS) $P = 3$ (si es entero)

decomponer em 3 FFTs-radix 2 de 2^M puntos (M etapas)

¿Qué pasa si N es un número primo?

- calcular DFT directa
- Se debe hacer zero padding hasta llegar al caso más eficiente de los anteriores

¿Hay otras descomposiciones posibles?

Sí, el caso más general es $N = L \cdot M$
el cual da origen a múltiples versiones de FFTs

¿Porqué MATLAB tiene un fft tan eficiente?

Porque busca el mejor algoritmo para el largo de la señal ... y además aprende !!!

↳ ver fftw en MATLAB

DESCOMPOSICIÓN para $N = 3 \cdot 2^M$

$$X^{(N)}(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi kn/N} \qquad k = 0, 1, \dots, N-1$$

$$= \sum_{m=0}^{\frac{N}{3}-1} \underbrace{x(3m)}_{x_0(n)}\, e^{-j2\pi km/(N/3)}$$

$$+ \sum_{m=0}^{\frac{N}{3}-1} \underbrace{x[3m+1]}_{x_1(n)}\, \underbrace{e^{-j2\pi k(3m+1)/N}}_{e^{-j2\pi km/(N/3)} \cdot e^{-j2\pi k/N}}$$

$$+ \sum_{m=0}^{\frac{N}{3}-1} \underbrace{x[3m+2]}_{x_2(n)} \cdot \underbrace{e^{-j2\pi k(3m+2)/N}}_{e^{-j2\pi km/(N/3)} \cdot e^{-j2\pi k2/N}}$$

$$X^{N}(k) = X_0^{(\frac{N}{3})}(k) + X_1^{(\frac{N}{3})}(k) \cdot W_N^{k} + X_2^{(\frac{N}{3})}(k) W_N^{2k}$$

$k = 0, 1, \dots, N-1$

$X_0^{(N/3)}(k)$, $X_1^{(N/3)}(k)$ y $X_2^{(N/3)}(k)$ tienen $\left(\frac{N}{3}\right)\log_2\left(\frac{N}{3}\right)$ cos

+ 2 multiplicaciones por W_N^k y W_N^{2k}, N veces

$$\Rightarrow \# \text{ operaciones} = 3 \cdot \left(\frac{N}{3}\right) \log_2\left(\frac{N}{3}\right) + 2N$$

$$= N \log_2\left(\frac{N}{3}\right) + 2N$$

Ejemplo: $N = 12 = 3 \cdot 4 = 3 \cdot 2^2$

1) $\Rightarrow$ i) # CO DFT directo $= (12)^2 = 144$!!

3) ii) # CO FFT zeropadding $12 \longrightarrow 16 = 2^4$

$\Rightarrow 4 \cdot 2^4 = 4 \cdot 16 = 64$!!

4) iii) # CO Special decoup

$N = 3 \cdot 4 = 3 \cdot 2^2 \Rightarrow$ 3 DFTs de $2^2 = N_2 \Rightarrow 3 \cdot N_2$

donde cada una tiene $2^2 \cdot 2 = 8$ CO

$$X^{(N)}(K) = X_0^{(N/3)}(K) + W_N^K X_1^{(N/3)}(K) + W_N^K X_2^{(N/3)}(K)$$

$K = 0, \dots, N-1 \Rightarrow 3 \cdot \left(\frac{N}{3}\right) \log\left(\frac{N}{3}\right) + 2 \cdot N$

$\Rightarrow$ # CO totales $= 3 \cdot 8 \Rightarrow 24$ CO.

$24 + 2 \cdot 12 = 48$ COs!

2) iv) # CO divide y conquista: 2 DFTs de 6 puntos directos

a) $\Rightarrow \underbrace{2 \cdot 36}_{} + 12$

$72 + 12 = 84$!!

b) 4 DFTs de 3 puntos
(usando divide y conquista 2 veces)

$\Rightarrow (3)^2 \cdot 4 + 2 \cdot 12 = 60$ COs !!

Convolución periódica

El filtrado de señales finitas $\to x[n], n = 0, 1, \ldots, N$ mediante una respuesta en impulso conocida $h[n], n = 0, 1, \ldots, M$, donde $M < N$ (Filtro FIR) está dada por

$$y[n] = \sum_{m=0}^{M-1} h[m]\, x(n-m)$$

Cada salida $y[n]$ requiere M multiplicaciones y $M-1$ sumas.

Usando la FFT podemos filtrar la señal de modo que

1) Calcular $X^{(N)}(K) \longrightarrow N \log_2(N)$ operaciones
2) Extender $h[n]$ con ceros de modo que tenga largo N y calcular $H^{(N)}(K) \longrightarrow N \log_2(N)$ operaciones
3) Multiplicar $X^{(N)}(K) \cdot H^{(N)}(K) = Y^{(N)}(K) \to N$ multiplicaciones
4) Calcular la inversa de $Y^{(N)}(K) = y[n] = DFT^{-1}\{Y^{(N)}(K)\}$ $\longrightarrow N \log_2(N)$ operaciones

OBS) ¿Cómo se calcula la DFT^{-1}? Usar propiedad de reciprocidad de la DFT.

$$x[n] \xleftrightarrow{DFT} X(K)$$
$$X[n] \xleftrightarrow{DFT} N \cdot x(-K)$$

Es decir tomar $X(K)$ como entrada, e invertirla en el tiempo y normalizarla por N una vez transformada

$$X^{(2)}(0) = x[0] + x[1]$$
$$X^{(2)}(1) = x[0] - x[1]$$

DFT 2 puntos

$$X^{(2)}(0) + X^{(2)}(1) = 2x[0]$$

$$\Rightarrow \quad x[0] = \frac{1}{2}\left(X^{(2)}(0) + X^{(2)}(1)\right)$$
$$x[1] = \frac{1}{2}\left(X^{(2)}(0) - X^{(2)}(1)\right)$$

IDFT 2 puntos

OBS] · $x[-n] = x[N-n]$

· Para $N=2$ $\quad X^{(N)}(k) = X^{N}(-k)$ ((caso especial))

vs $2M-1$

⟶ El total de operaciones 1)– 4) ⇒ $3N\log_2 N + N$ ↓

y la cantidad de operaciones por cada salida = $3\log_2 N + 1$
(en la DFT)

⇒ Se necesitan N puntos para reconstrucción perfecta

⇒ cuando el largo del filtro es igual largo de la señal (i.e, $M=L$) es mucho más eficiente usar el filtrado vía FFT

⇒ Mientras mayor sea M (el largo del filtro), mejor es filtrar vía FFT. Para M muy pequeños $N<5$ la convolución puede ser más eficiente especialmente si N es muy grande

* ver gráfico de ambos métodos para varios N y M

Al tomar la inversa de $Y^{(N)}(K) = H^{(N)}(K) \cdot X^{(N)}(K)$ se obtiene convolución circular donde el resultado es periódico con periodo N.

¿De dónde viene esto?

Considere la inversa IDFT de $Y^{(N)}(K) = H^{(N)}(K) X^{(N)}(K)$

$$y(n) = \frac{1}{N} \sum_{K=0}^{N-1} H^{(N)}(K) X^{(N)}(K) e^{j2\pi Kn/N}$$

$$= \frac{1}{N} \sum_{K=0}^{N-1} \sum_{m=0}^{N-1} x(m) e^{-j2\pi m K/N} H^{(N)}(K) e^{j2\pi Kn/N}$$

$$= \frac{1}{N} \sum_{m=0}^{N-1} x(m) \sum_{K=0}^{N-1} H^{(N)}(K) e^{j2\pi (n-m)K/N}$$

$$= \sum_{m=0}^{N-1} x(m) \underbrace{\left(\frac{1}{N} \sum_{K=0}^{N-1} H^{(N)}(K) e^{j2\pi (n-m)K/N} \right)}_{h[n-m]}$$

$$= \sum_{m=0}^{N-1} x[m] h[n-m]$$

por periodicidad DFT

pero $h[n-m] = h[n-m+N] = h[n-m] \bmod N$

$\Rightarrow$

$y[n] = x[n] \, (N) \, h[n]$ convolución circular

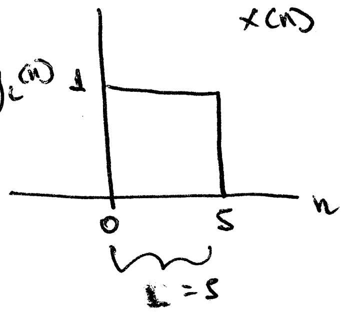
$y[n] = x[n] \circledast h[n]$ (ambas notaciones son comunes)

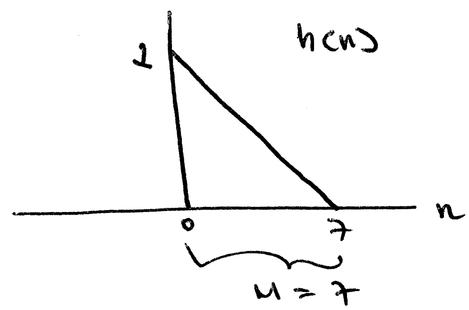
Convolución circular

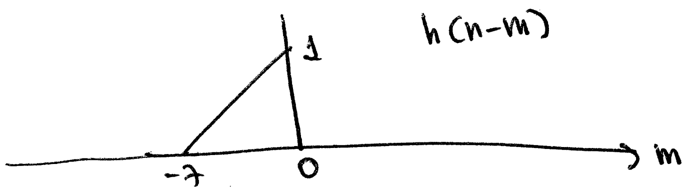
length(x) = L
length(h) = M
length(y) = N

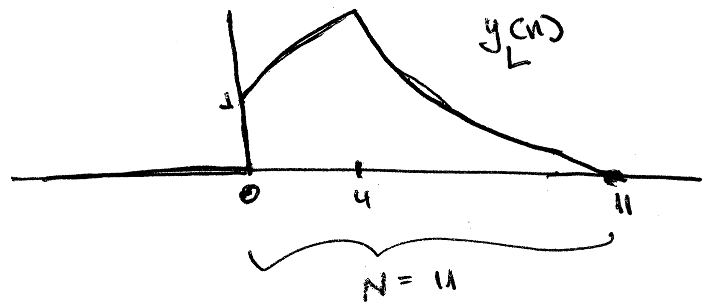
1) lineal

$x(n) * h(n) = y_L(n)$



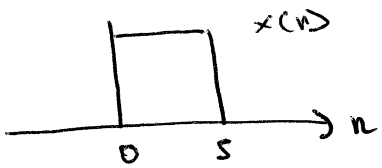


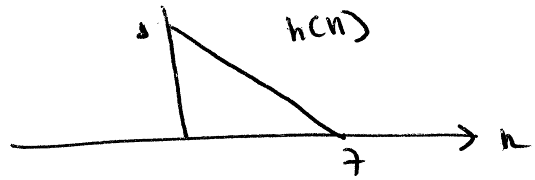




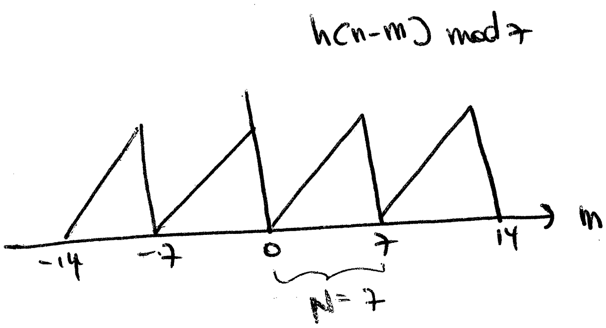
2) circular

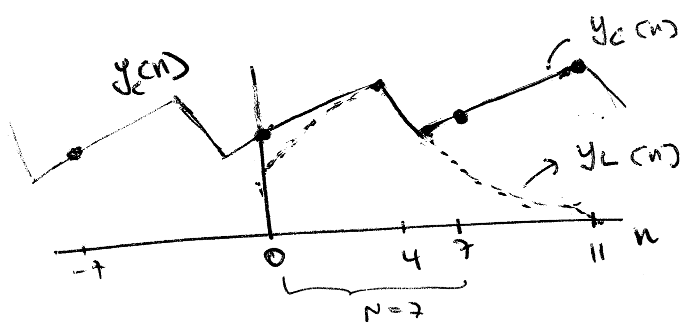
$y_c(n) = x(n) \circledast_N h(n)$



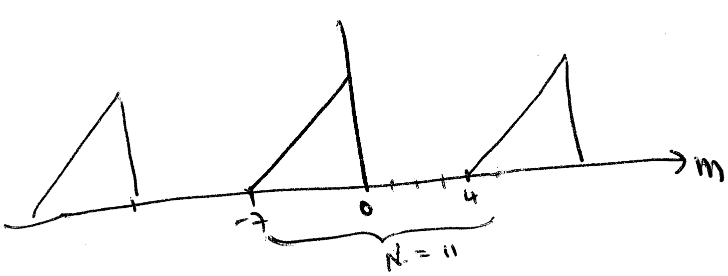


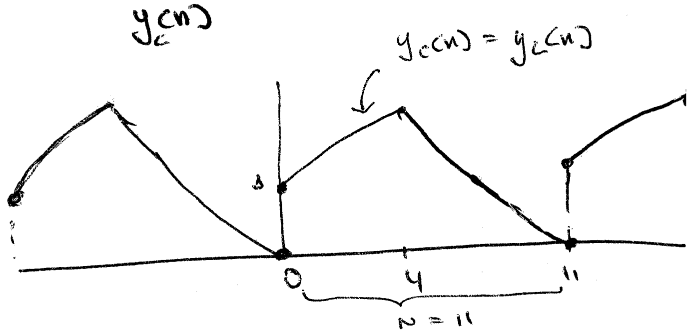
N = 7





N: 5+7-1 = 11
(L+M-1)





OBS) Para lograr la misma señal (dentro de un período) con convolución circular se requiere que ésta tenga L+M-1 puntos o más.

Ejemplo.

dado $x_1(n) = \{ \underset{\uparrow}{2}, 1, 2, 1 \}$

$x_2(n) = \{ \underset{\uparrow}{1}, 2, 3, 4 \}$

encontrar $y_L = x_1(n) * x_2(n)$ y $x_1(n) \overset{4}{\circledast} x_2(n) = y_c$

i) $y_L \Rightarrow$ 0, 0, 0, 2, 1, 2, 1, 0, 0, 0 ← $x(n)$

$m = 0$ 4 3 2 1 → $y_L(0) = 2$

$m = 1$ 4 3 2 1 → $y_L(1) = 4+1 = 5$

$x_2(n-m)$ 4 3 2 1 → $y_L(2) = 6+2+2 = 10$

4 3 2 1 → $y_L(3) = 8+3+4+1 = 16$

4 3 2 1 → $y_L(4) = 4+6+2 = 12$

4 3 2 1 → $y_L(5) = 8+3 = 11$

$m = 6$ 4 3 2 1 → $y_L(6) = 4$

$y_L(n) = \{ \underset{\uparrow}{2}, 5, 10, 16, 12, 11, 4 \}$ (largo $7 = 4+4-1$)

ii) y_c 2 1 2 1 ← $x(n)$

$m=0$ 4 3 2 1 4 3 2 → $y_c(0) = 2+4+6+2 = 14$

$m=1$ 2 1 4 3 → $y_c(1) = 4+1+8+3 = 16$

$x_2(n-m)_{\text{mod } 4}$ 3 2 1 4 → $y_c(2) = 6+2+2+4 = 14$

$m=$ 4 3 2 1 → $y_c(3) = 8+3+4+1 = 16$

$y_c(n) = \{ \underset{\uparrow}{14}, 16, 14, 16 \}$

Filtrado de señales de larga duración usando overlap-save y overlap-add

- Consideremos una señal de largo L que es filtrada por un filtro FIR de largo M donde $L \gg M$.

- Aunque un filtrado en frecuencia vía DFT sea más eficiente usando la señal completa, es conveniente, y a veces necesario, que el filtrado produzca un vector de salida de manera más frecuente.

  (OBS: Mientras más larga la señal, menos frecuentemente tendremos una salida de la IDFT, es decir el filtrado vía DFT es en función a cuadros y no muestras)

  muestras ↳ se refiere a muestras individuales.

  cuadros ↳ se refiere a ventanas de muestras

recordar que por cada muestra:

→ convolución directa # $M + M - 1$ operaciones complejas

→ DFT para filtrar # $3 \log_2 N + 1$

¿Dónde se cruzan?

$$2M - 1 = 3 \log_2 N + 1$$

$$\frac{2}{3}(M-1) = \log_2 N \Rightarrow N = 2^{\frac{2}{3}(M-1)}$$

$$\Rightarrow M = \frac{3}{2} \log_2(N) + 1$$

entonces si $M < 4$ es mejor filtrar con FIR y convolución directa, pero si $M > 4$ existe una región en la cual es más conveniente usar filtrado DFT (ver gráfico).

→ caso voz, $N = 256$ (~ 25 ms @ $f_s = 8$ KHz)

si el filtro tiene orden 12 o más conviene usar filtrado DFT
($M = 13$)

Overlap-Save y overlap-add

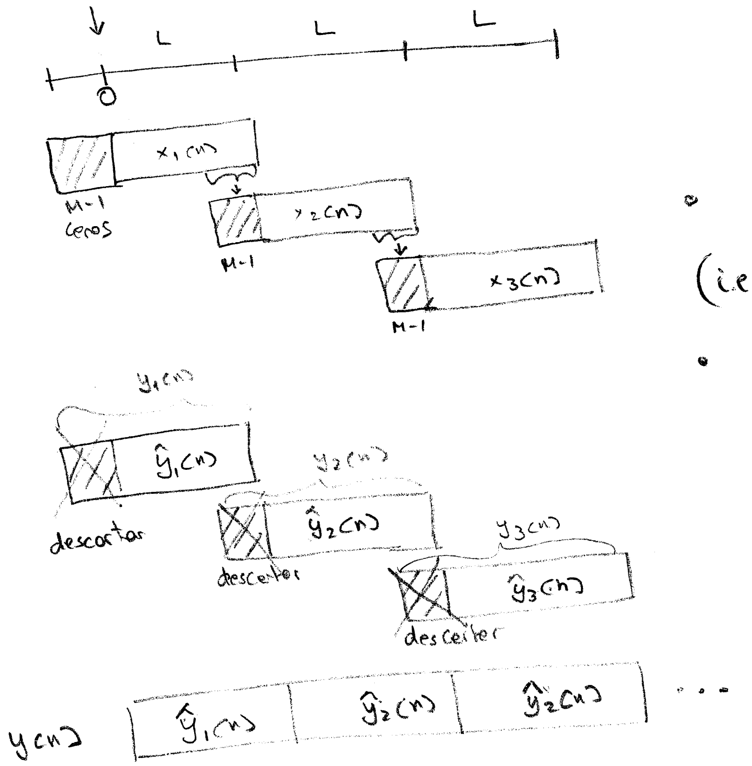
la Convolución $y[n] = \sum_{K=0}^{M-1} h[K]\, x[n-K]$

donde $x(n)$ tiene largo L y $h(n)$ largo M, entonces $y(n)$ tiene largo $N = L + M - 1$ (assumiendo que $x[n] = 0$ para $0 > n > L$ y $h(n) = 0$ para $0 > n > M$)

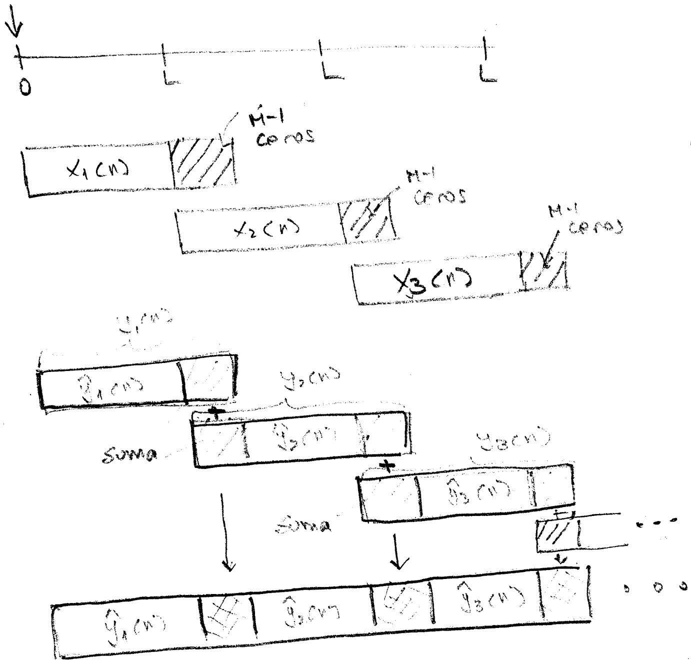
$\Rightarrow$ el largo de la DFT debe ser entonces $N = L + M - 1$, pero queremos dividir la señal $x(n)$ en segmentos de largo L.
hay dos formas de hacer esto

i) overlap-save:

- toma muestras del cuadro anterior ($M-1$ muestras) y las agrega adelante del cuadro para tener el largo $N = L + M - 1$
- Realiza el Filtrado en Frecuencia (i.e., $y_i[n] = IFFT(FFT(x_i, N) \cdot FFT(h, N))$)
- Descarta los puntos repetidos en $y_i(n)$ que se usaron en dos cuadros y junta todos los $y[n]$



ii) Overlap-add :



- Agrega ceros a cada cuadro para obtener el largo $N = L+M-1$ (zero-padding)
- Realiza el filtrado en frecuencia

$$y_i(n) = \text{ifft}(\text{fft}(x,N) \cdot \text{fft}(h,N))$$

- Suma las porciones que se cruzan entre cuadros y juntando todas las demás porciones para obtener $y(n)$

OBS) Ambas técnicas son eficientes cuando se realiza filtrado de señales largas $L \gg M$ y/o cuando se desea procesamiento en tiempo real