

Procesamiento de señales, fundamentos

Maestría en sistemas embebidos

Universidad de Buenos Aires

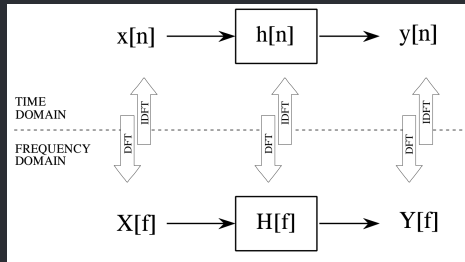
MSE 5Co2020

Clase 5 - Aplicaciones de DFT

Ing. Pablo Slavkin

slavkin.pablo@gmail.com

wapp:011-62433453



SAPI

Se aceptan pull request para la SAPI

SAPI DSP

Enuestas

Encuesta anónima clase a clase

Propiciamos este espacio para compartir sus sugerencias, críticas constructivas, oportunidades de mejora y cualquier tipo de comentario relacionado a la clase.

Encuesta anónima



<https://forms.gle/1j5dDTQ7qjVfRwYo8>

Link al material de la material



https://drive.google.com/drive/u/1/folders/1TIR2cgDPchL_4v7DxdpS7pZHtjKq38CK

Repaso Convolución

Multiplicación?!



55m50s

Algoritmo de Multiplicación de 2do grado

- Recordar la técnica de multiplicación aprendida
- Considerar que 1 2 es $h(n)$ y 3 4 $x(n)$
- Suponer que tenemos más que 10 símbolos y que no es necesario hacer acarreo
- Entran 2 números de 2 cifras y sale 1 de 3

$$\begin{array}{r} \begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array} \\ \hline \begin{array}{ccc} & 4 & 8 \\ 3 & 6 & 0 \end{array} \\ \hline \begin{array}{ccc} 3 & 10 & 8 \end{array} \end{array}$$

Repaso Convolución

Descomposición delta



56m57s

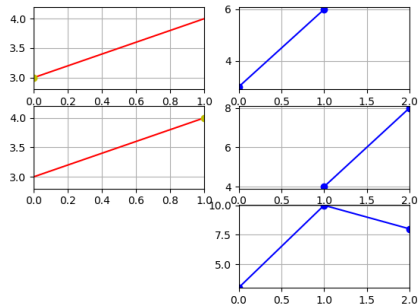
Suma deltas desplazadas

- Considerar escalar y desplazar $h(n)$ para hacer la convolucion
- 2 movimientos para obtener el resultado
- Entran 2 vectores de 2 elementos y sale 1 de 3

1	2	0
3	0	0
<hr/>		
3	6	0
<hr/>		
0	1	2
0	4	0
<hr/>		
3	6	0
0	4	8
<hr/>		
3	10	8

5_clase/conv_as_multiply1.py

pause



Repaso Convolución

Convolucion formal



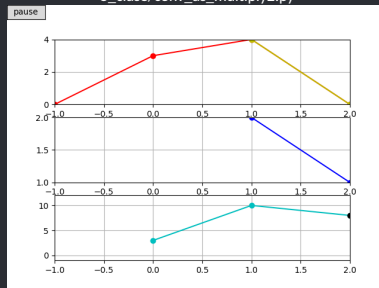
1h3m12s

Convolucion

2	1	0	0
0	3	4	0
<hr/>			
	3	0	0
<hr/>			
0	2	1	0
0	3	4	0
<hr/>			
	3	10	0
<hr/>			
0	0	2	1
0	3	4	0
<hr/>			
	3	10	8

- Invierto $h(n)$ desplazo, multiplico y sumo
- 3 movimientos para obtener el resultado
- Entran 2 vectores de 2 elementos y sale 1 de 3

5_clase/conv_as_multiply2.py



Repaso Convolución

Convolucion como producto de polinomios



1h10m50s

- Considerar cada elemento de $h(n)$ y $x(n)$ como coeficientes de un polinomio
- Multiplicar los 2 polinomios respetando exponentes
- Entren 2 vectores de 2 elementos y sale 1 de 3

$$\begin{aligned}(1x10^1 + 2x10^0) * (3x10^1 + 4x10^0) &= \\ (3x10^2 + 4x10^1 + 6x10^1 + 8x10^0) &= \\ (3x10^2 + 10x10^1 + 8x10^0)\end{aligned}$$

Repaso Convolución

Multiplicación?!



1h20m49s

Algoritmo de Multiplicación

	1	2
	3	4
<hr/>		
	4	8
3	6	0
<hr/>		
3	10	8

Multiplicación de polinomios

$$\begin{aligned}(1x10^1 + 2x10^0) * (3x10^1 + 4x10^0) &= \\ (3x10^2 + 4x10^1 + 6x10^1 + 8x10^0) &= \\ (3x10^2 + 10x10^1 + 8x10^0) &\end{aligned}$$

Suma deltas desplazadas

1	2	0
3	0	0
<hr/>		
3	6	0
<hr/>		
0	1	2
0	4	0
<hr/>		
3	6	0
0	4	8
<hr/>		
3	10	8

Convolución

2	1	0	0
0	3	4	0
<hr/>			
	3	0	0
<hr/>			
0	2	1	0
0	3	4	0
<hr/>			
	3	10	0
<hr/>			
0	0	2	1
0	3	4	0
<hr/>			
	3	10	8

Convolucion en tiempo

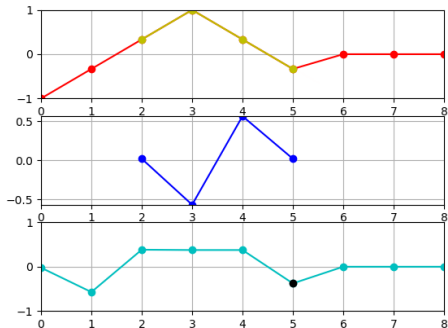
filtrado



1h21m19s

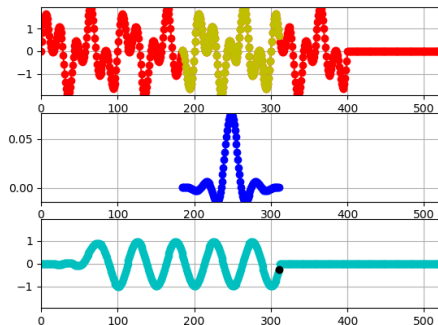
4_clase/conv1.py

pause



4_clase/conv2.py

pause



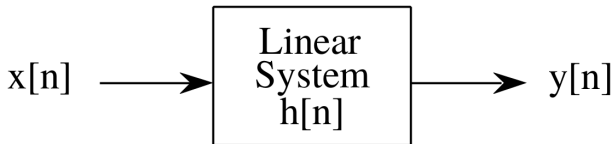
Repaso Convolucion

Propiedades



1h37m55s

- Conmutativa
- Distributiva
- Asociativa



$$x[n] * h[n] = y[n]$$

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n - k]$$

Repaso Multiplicacion

Propiedad conmutativa



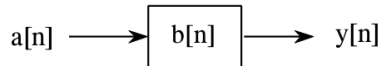
1h39m58s

Commutative Property

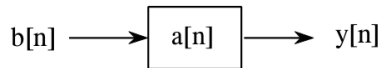
You can multiply the factors in any order, and the product will be the same.

$$\begin{array}{c} \text{Green Car} \\ \text{5} \times \text{4} \\ \hline 20 \end{array} = \begin{array}{c} \text{Yellow Car} \\ \text{4} \times \text{5} \\ \hline 20 \end{array}$$

IF



THEN



Repaso Multiplicacion

Propiedad asociativa



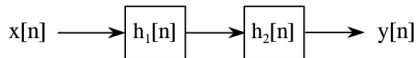
1h40m49s

Associative Property

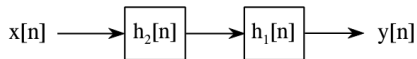
You can **group** the factors in different ways, and the product will be the same.

$$(3 \times 4) \times 2 = 24$$
$$3 \times (4 \times 2) = 24$$

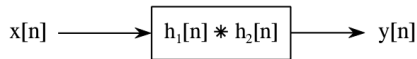
IF



THEN



ALSO



Repaso Multiplicacion

Propiedad distributiva



1h42m40s

Distributive Property

A multiplication fact can be broken up into the sum of two other multiplication facts.

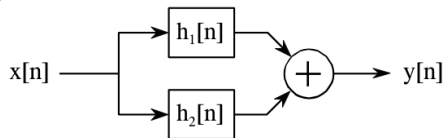
$$23 \times 2 = ?$$

$$(20 + 3) \times 2 =$$

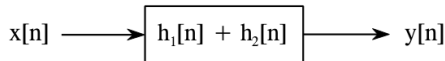
$$(20 \times 2) + (3 \times 2) = 40 + 6 = 46$$

Break it into numbers that are easier to multiply with mental math

IF



THEN

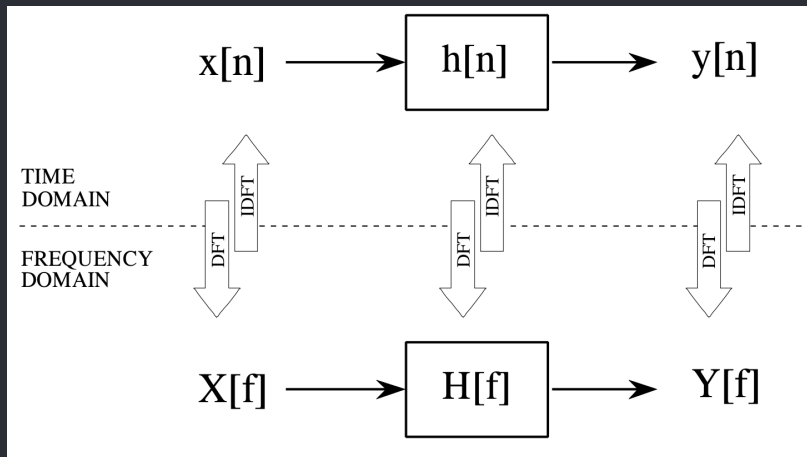


Convolución vs Multiplicación

Teorema de la convolución



2h07m18s



Multiplicación con DFT

Tiempo vs Frecuencia

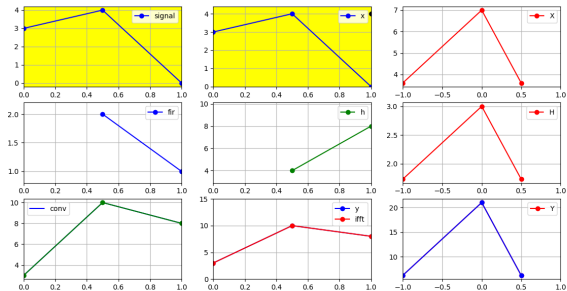


2h18m24s

- Ejemplo de calcular 12 conv 3 4 utilizando:
 - Convolución x desplazamiento de $h(n)$
 - Convolución invirtiendo $h(n)$
 - Convolución usando $\text{iFFT}(\text{fft}(h) * \text{fft}(x))$
- Notar que todos los resultados son iguales
- Notar que siempre la salida con $N+M-1$ datos
- En este ej. $2+2-1=3$

5_clase/conv_vs_dft1.py

pause



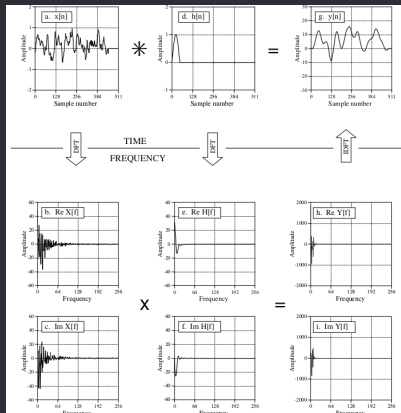
Convolución vs Multiplicación

Teorema de la convolución



2h32m53s

- Imagen en donde se destaca el camino de la ifft para calcular la convolución
- Notar la representación en parte real e imaginaria



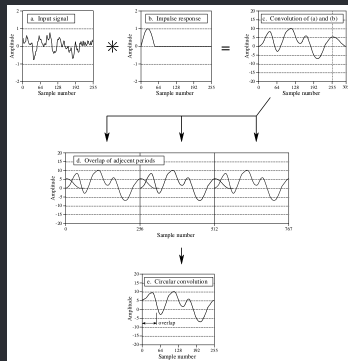
Convolución vs Multiplicación

Convolución circular



2h45m10s

- Efecto de la convolución circular
- De la definición de la DFT se supone que $x(n)$ es periódica
- Dado que en la practica cortamos en algún pinto $x(n)$ para procesar, cuando involucionamos con otra señal debemos agregar ceros para evitar el efecto de solapamiento en la salida y asegurarnos que la salida tenga cuando menos $N+M-1$ valores



Convolución vs Multiplicación

Teorema de la convolución



2h50m00s

- Ecuación para obtener la salida de un sistema usando DFT
- El resultado es el mismo que convolucionar en el tiempo
- A partir de unos 64 puntos para $h(n)$ la velocidad de calculo de la DFT es superior a la convolución en el tiempo

$$x * y = \text{DTFT}^{-1} \left[\text{DTFT}\{x\} \cdot \text{DTFT}\{y\} \right]$$

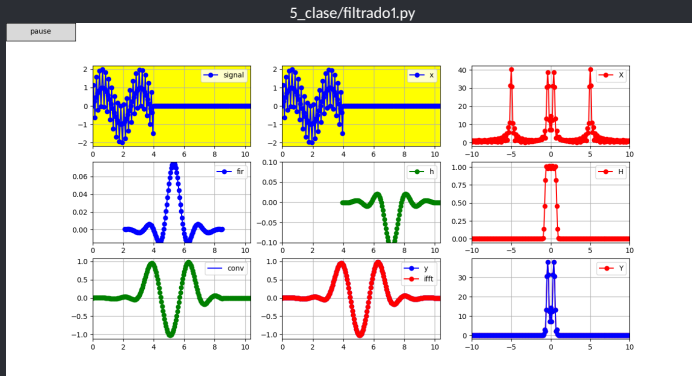
Filtrado

Pasabajos



2h54m45s

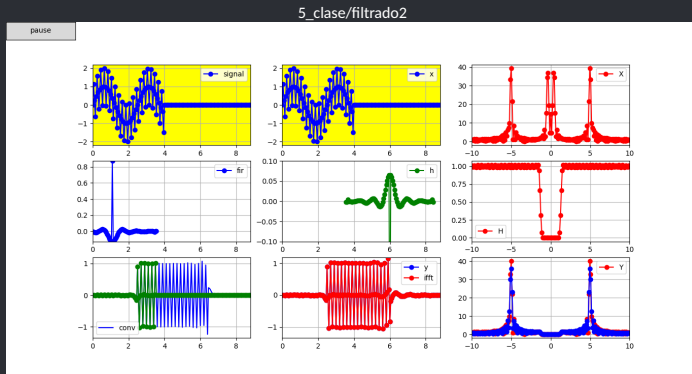
- Ejemplo de como tomar tramos de x , rellenar con ceros, tomar h , rellenar con ceros y luego convulocionar
- Al mismo tiempo, se calcula $\text{FFT}(x\text{-padd})$ $\text{FFT}(h\text{-padd})$, se multiplican entre si y luego se hace la IFFT para obtener el mismo resultado que la convolución





3h04m50s

- Ejemplo de como tomar tramos de x , rellenar con ceros, tomar h , rellenar con ceros y luego convolucionar
- Al mismo tiempo, se calcula $\text{FFT}(x\text{-padd})$ $\text{FFT}(h\text{-padd})$, se multiplican entre si y luego se hace la IFFT para obtener el mismo resultado que la convolucion



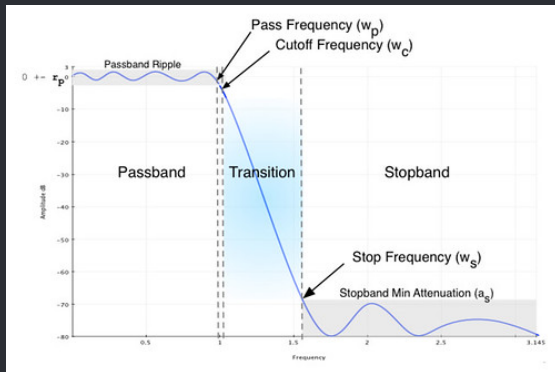
Filtrado

Definición



3h09m58s

- Plantilla de diseño de un filtro
- En el ejemplo se aprecia un pasabajos pero se destacan las zonas de interés y los niveles de la banda de paso y de rechazo
- Cuanto mas exigente se la plantilla del filtro, mas puntos tendrá nuestra $h(n)$ y mas lenta y compleja su convolución
- El objetivo es llegar a un compromiso entre los requisitos y la performance



PyFDA /opt/anaconda3/bin/pyfdax



-

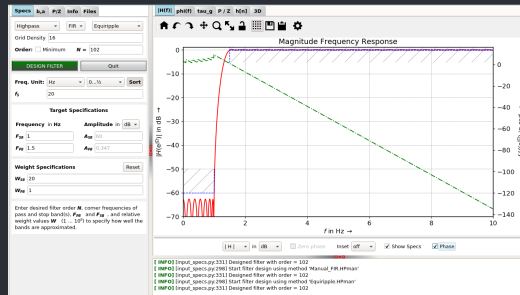
Filtrado

Pyfda `/opt/anaconda3/bin/pyfdax`



3h14m00s

- Uso de PyFDA como herramienta para diseño de filtros
- Inicialmente solo nos concentramos en la $H(f)$ para visualizar de manera practica las zonas de paso y de rechazo
- Notar la variedad de opciones disponibles y la respuesta en fase en esta imagen



Bibliografía

Libros, links y otro material

[1] *ARM CMSIS DSP.*

https://arm-software.github.io/CMSIS_5/DSP/html/index.html

[2] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. Second Edition, 1999.

[3] *Wikipedia.*

https://en.wikipedia.org/wiki/Convolution_theorem

[4] *PyFDA doc.*

<https://buildmedia.readthedocs.org/media/pdf/pyfda/latest/pyfda.pdf>