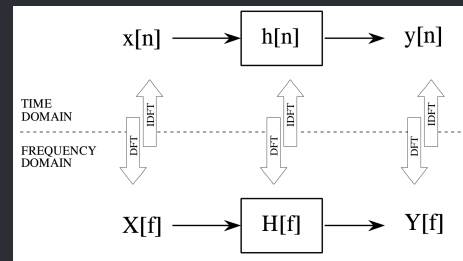


2020-05-29

Procesamiento de señales, fundamentos

Maestría en sistemas embebidos
Universidad de Buenos Aires
MSE 5Co2020

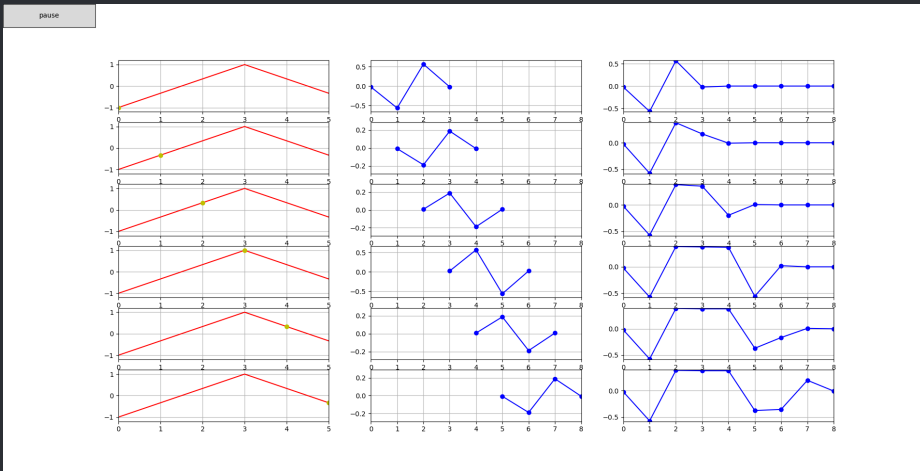
Clase 5 - Aplicaciones de DFT



Ing. Pablo Slavkin
slavkin.pablo@gmail.com
wapp:011-62433453

Repaso Convolución

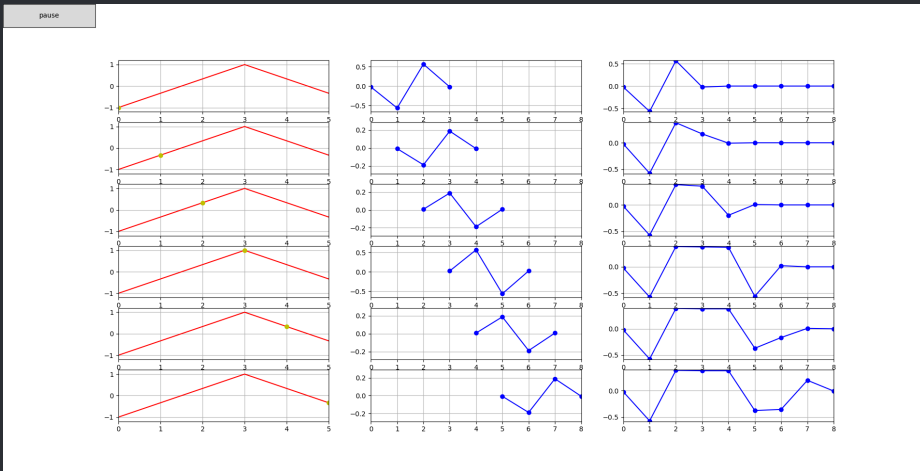
Descomposición delta



- lanzar conv3
- sumo delta a delta y voy acumulando..
- la salida es $N+M-1$

Repaso Convolución

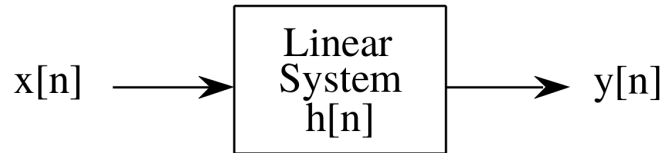
Descomposición delta



- lanzar conv3
- sumo delta a delta y voy acumulando..
- la salida es $N+M-1$

Repaso Funcion delta

Respuesta al impulso



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

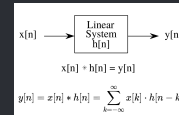
$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n - k]$$

2020-05-29

Procesamiento de señales, fundamentos

Repaso Funcion delta

Repaso Funcion delta
Respuesta al impulso



Convolución

Respuesta al impulso

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sc
from matplotlib.animation import FuncAnimation
from buttons import buttonOnFigure

#-----
fig = plt.figure()
fs = 6
N = 6
xFrec = 1

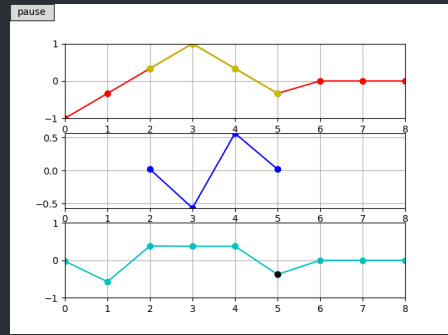
hData=np.load("4_clase/low_pass_short.npy").astype
(float)
hData=np.insert(hData,0,hData[-1]) #ojo que pydfa
me guarda 1 dato menos...
M=len(hData)
#-----
def x(f,n):
    return 1*sc.sawtooth(2*np.pi*xFrec*n/fs,0.5)

tData = np.arange(-M+1,M+1,1)
xData = np.zeros(N+2*(M-1))
xData[M-1:N+M-1] = x(xFrec,tData[M-1:N+M-1])
xAxis = fig.add_subplot(3,1,1)
xLn,xHighLn = plt.plot(tData,xData,'r-o',
, [], [], 'y-o')
xAxis.grid(True)
xAxis.set_xlim(0,M+N-2)
xAxis.set_ylim(np.min(xData),np.max(xData))
#-----

hAxis = fig.add_subplot(3,1,2)
hLn, = plt.plot([], [], 'b-o')
hAxis.grid(True)
hAxis.set_xlim(0,M+N-2)
hAxis.set_ylim(np.min(hData),np.max(hData))
#-----
yAxis = fig.add_subplot(3,1,3)
yLn,yDotLn = plt.plot([], [], 'c-o', [], [], 'ko')
yAxis.grid(True)
yAxis.set_xlim(0,M+N-2)
yAxis.set_ylim(np.min(xData),np.max(xData))
yData=[]
#-----
def init():
    global yData
    yData=np.zeros(N+M-1)
    return hLn,xLn,xHighLn,yLn,yDotLn

def update(i):
    global yData
    t=np.linspace(-M+1+i,M,endpoint=True)
    yData[i]=np.sum(xData[t:t+M]*hData[t:t+M])
    xHighLn.set_data(t,xData[t:t+M])
    hLn.set_data(t,hData[t:t+M])
    yLn.set_data(tData[M-1:],yData)
    yDotLn.set_data(tData[M-1+i],yData[i])
    return hLn,xLn,xHighLn,yLn,yDotLn

ani=FuncAnimation(fig,update,M+N-1,init,interval
=1000 ,blit=True,repeat=True)
plt.get_current_fig_manager().window.showMaximized
()
b=buttonOnFigure(fig,ani)
plt.show()
```

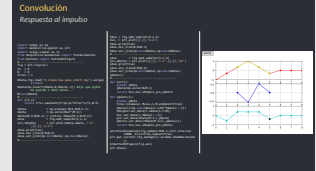


2020-05-29

Procesamiento de señales, fundamentos

- Repaso Convolución
- Convolución

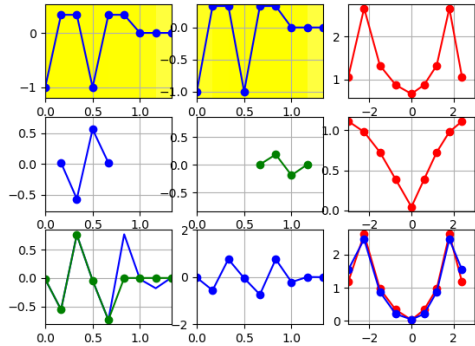
- lanzar conv1
- hacer las cuentas a mano



Convolución vs Multiplicacion

Tiempo vs Frecuencia

pause

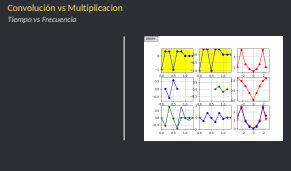


2020-05-29

Procesamiento de señales, fundamentos

└─ Convolucion vs Multiplicacion

└─ Convolución vs Multiplicacion



- explicar el teorema de la convolucion a mano

Bibliografía

Libros, links y otro material

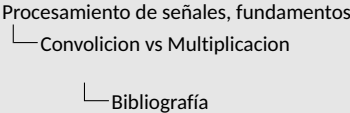
[1] ARM CMSIS DSP.
https://arm-software.github.io/CMSIS_5/DSP/html/index.html

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

[3] Grant Sanderson
<https://youtu.be/spUNpyF58BY>

[4] Interactive Mathematics Site Info.
<https://www.intmath.com/fourier-series/fourier-intro.php>

2020-05-29



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. <i>The Scientist and Engineer's Guide to Digital Signal Processing</i> . Second Edition, 1999.
[3] Grant Sanderson https://youtu.be/spUNpyF58BY
[4] Interactive Mathematics Site Info. https://www.intmath.com/fourier-series/fourier-intro.php