

2.프로그램은 Python을 썼으며,Python은 FFT에 대해 간단한 하고 강력한 함수를 제공하고, 이함수를 이용하여 프로그래밍 하면

#python programming by Kim Gi Soek

import matplotlib.pyplot as plt

import numpy as np

Fs = 1000.0; # sampling rate

Ts = 1.0/Fs; # sampling interval

t = np.arange(0,4,Ts) # time vector

y = []

for temp in t: # $0 < x < 4, V_x = 1$

if abs(temp) < 2:

y.append(1.)

else:

y.append((0.))

n = len(y) # length of the signal

k = np.arange(n)

T = n/Fs

frq = k/T # two sides frequency range

frq = frq[range(int(n/100))] # one side frequency range

Y = np.fft.fft(y,n= 40000)/n # fft computing and normalization

Y = Y[range(int(n/100))]

fig, ax = plt.subplots(2, 1)

ax[0].plot(t,y)

ax[0].set_xlabel('Time')

ax[0].set_ylabel('Amplitude')

ax[0].grid(True)

ax[1].plot(frq,Y,'r') # plotting the spectrum

ax[1].set_xlabel('Freq (Hz)')

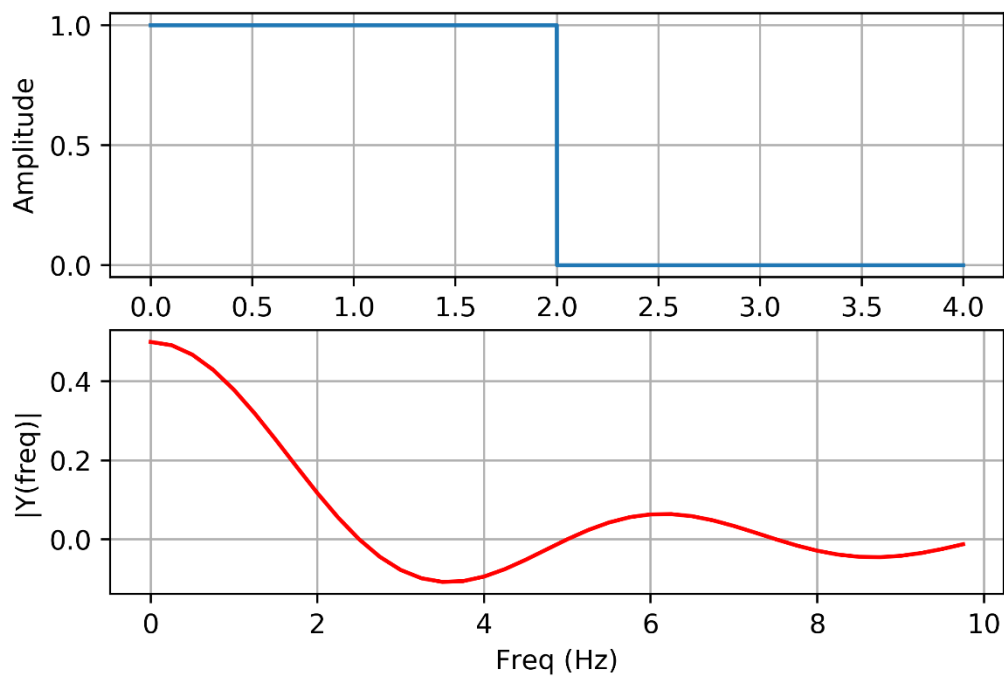
ax[1].set_ylabel('|Y(freq)|')

ax[1].grid(True)

plt.savefig('save.png',dpi = 500)

이며, $0 < x < 2$ 일때 $f(x) = 1$, $2 < x < 4$ 일때 $f(x) = 0$ 인 함수를 주파수 1000(1000번의 구간)으로 FFT 하는 프로그램이며, FFT 함수 특정상 x축 -구간 까지는 포함이 안되므로 +구간만 일단 프로그래밍하였다

3. 프로그램 실행 결과



원래 $-2 < x < 2$ 에서 $f(x)=1$ 인 함수였지만 FFT 함수는 -구간을 생각안하고 계산하므로 +부분만 계산한후에 -부분은 +부분과 대칭되기 때문에 저값에서 대칭되는값이 -부분이다.