

Ödev7_Ornek3

May 7, 2021

```
[1]: # As usual, we begin with importing necessary libraries and functions
import numpy as np          # for using basic array functions
import matplotlib.pyplot as plt # for this example, it may not be necessary

# the main package for signal processing is called "scipy" and we will use
↳ "signal" sub-package
import scipy.signal as sgnl
from scipy.fftpack import fft, ifft
# alternative syntax: from scipy import signal as sgnl
%matplotlib notebook
```

```
[2]: N = 40
hn = np.ones(5)
Hw = np.fft.fftshift(fft(hn, N))

w = np.linspace(-np.pi, np.pi, N)

Hw_mag = abs(Hw)
Hw_phs = np.unwrap(np.angle(Hw))

fig, (ax1, ax2) = plt.subplots(2)
ax1.plot(w/np.pi, Hw_mag)
ax2.plot(w/np.pi, Hw_phs), plt.ylabel('phase (rad)'), plt.xlabel('frequency rad/
↳sample')
```

<IPython.core.display.Javascript object>

<IPython.core.display.HTML object>

```
[2]: ([<matplotlib.lines.Line2D at 0x216f02df5f8>],
      Text(0, 0.5, 'phase (rad)'),
      Text(0.5, 0, 'frequency rad/sample'))
```

```
[7]: tau_w = -np.diff(Hw_phs)*(N/(2*np.pi))

plt.figure()
plt.plot(w[1:]/np.pi, tau_w)
plt.xlabel('frequency (rad/sample)')
plt.ylabel('group delay (samples)')
```

<IPython.core.display.Javascript object>

<IPython.core.display.HTML object>

[7]: Text(0, 0.5, 'group delay (samples)')

Now, we will simulate a sinusoidal signal and apply it to the input of the system defined above. Then by comparing input and the output, we will observe and confirm the effects of the group delay.

```
[8]: n = np.arange(0, N, 1)      # first create the index vector at N points
     w0 = 0.1*np.pi             # frequency of the input signal
     xn = np.cos(w0*n)           # create the input signal at w0

     Xw = np.fft.fftshift(fft(xn, N))    # compute the frequency response (in the
     ↪ [-pi pi] interval)

     Yw_shifted = Hw * Xw           # compute the fft of the output signal
     Yw = np.fft.ifftshift(Yw_shifted)  # we shifted back to the 0-2*pi interval
     yn = np.real(ifft(Yw))
```

Now that we found the output signal in the time domain, we will compare it to the input and interpret the results.

```
[9]: fig, (ax1, ax2) = plt.subplots(2)
     ax1.stem(n, xn), ax1.title.set_text('input signal')
     ax2.stem(n, yn), ax2.title.set_text('output signal')
```

<IPython.core.display.Javascript object>

<IPython.core.display.HTML object>

[9]: (<StemContainer object of 3 artists>, None)

Now, by looking at the first peaks, we can confirm that, as we have predicted from the group delay, the output signal shifted exactly 2 samples with respect to the input.

[]: