Homework 2 - Problems 5-6

October 17, 2019

1 Homework 2: *Problems 5 - 6*

[52]:
```
import numpy as np
import math
import matplotlib.pyplot as plt
import IPython.display as ipd
from scipy import signal

%matplotlib inline
```

1.1 Problem 5 - Basic filtering

Consider the filter corresponding to the following difference equation:

$$y[n] = \frac{1}{3}x[n] + \frac{1}{3}x[n-2] + \frac{1}{3}y[n-1]$$

Plot the magnitude of its frequency response. What type of filter is this?

Next, using a sampling rate of 8000Hz, create a linear chirp signal whose frequency starts at 50Hz and ends after 5 seconds at 2kHz.

Apply this filter to the chirp signal. Plot this new signal's magnitude frequency spectrum and listen to it. How does it sound compared to the original chirp?

1.2 Your Answer

$$Y(z) = \frac{1}{3}X(z) + \frac{1}{3}z^{-2}X(z) + \frac{1}{3}z^{-1}Y(z)$$

$$Y(z)(1 - \frac{1}{3}z^{-1}) = X(z)(\frac{1}{3} + \frac{1}{3}z^{-2})$$

$$\frac{Y(z)}{X(z)} = \frac{\frac{1}{3} + \frac{1}{3}z^{-2}}{1 - \frac{1}{3}z^{-1}}$$

$$H(z) = \frac{1}{3}(\frac{1}{1 - \frac{1}{3}z^{-1}} + z^{-2}\frac{1}{1 - \frac{1}{3}z^{-1}})$$

$$h[n] = (\frac{1}{3})^{n+1}u[n] + (\frac{1}{3})^{n-2}u[n-2]$$

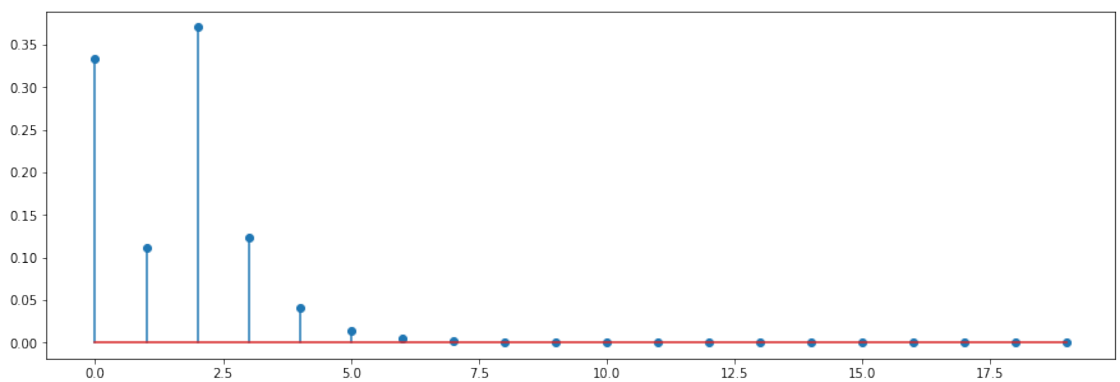
This is a low-pass filter

[2]:
```
##### to do #####
# determine the transfer function coefficients
numH = [1/3, 0, 1/3]
denH = [1, -1/3, 0]
#################

# compute the impulse response
systemH = signal.dlti(numH, denH)
n, h = signal.dimpulse(systemH, n=20)

# plot the impulse response
plt.figure(figsize = (15,5))
plt.stem(n,np.squeeze(h));
```

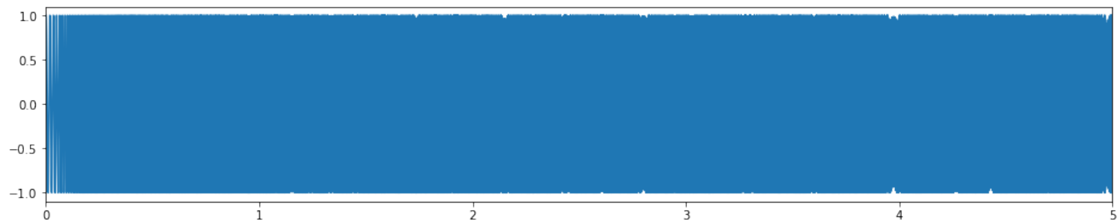
```
/home/snapple/2-coursework/ecec434/labs/lib/python3.6/site-
packages/ipykernel_launcher.py:13: UserWarning: In Matplotlib 3.3 individual
lines on a stem plot will be added as a LineCollection instead of individual
lines. This significantly improves the performance of a stem plot. To remove
this warning and switch to the new behaviour, set the "use_line_collection"
keyword argument to True.
  del sys.path[0]
```



[3]:
```
f1 = 50
f2 = 2000
fs = 8000
dur = 5
t = np.linspace(0, dur, fs*dur)
x = signal.chirp(t, f1, dur, f2)

plt.figure(figsize=(16,3))
plt.xlim(0, 5)
plt.plot(t,x)
```

[3]: [<matplotlib.lines.Line2D at 0x7f3ad8617208>]

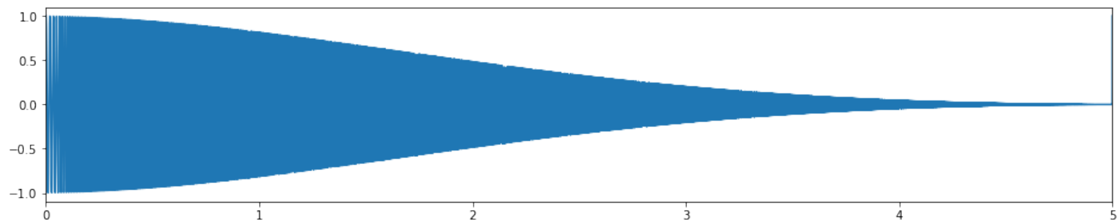


[4]:
```
ipd.Audio(x, rate=fs)
```

[4]: <IPython.lib.display.Audio object>

[5]:
```
# b, a = scipy.signal.butter(N, Wn, 'low')
o = signal.filtfilt(numH, denH, x)
plt.figure(figsize=(16,3))
plt.xlim(0, 5)
plt.plot(t,o)
```

[5]: [<matplotlib.lines.Line2D at 0x7f3ad8727080>]



[6]:
```
ipd.Audio(o, rate=fs)
```

[6]: <IPython.lib.display.Audio object>

The output signal's amplitude is much lower at higher frequencies than the input signal.

1.3 Problem 6 - System Convergence

Write a Python implemetation of the discrete-time system below with initial condition $y[-1] = 1$. Show that for an input $x[n] = au[n]$, the system converges to $\sqrt{a}$ as $n \to \infty$.

$$y[n] = \frac{1}{2}\left(y[n-1] + \frac{x[n]}{y[n-1]}\right)$$

1.4 Your Answer

[44]:
```
n = 10000
t = np.arange(-1,n,1)
```

[54]:
```
def iir(i, x, y):
    if (i == -1):
        return y
    if (i == 0):
        return 1
    else:
        return 0.5*(iir(i-1,x,y) + x[i]/iir(i-1,x,y))



a = 21

x = a*np.ones((t.shape[0]))
x[0] = 0

print(iir(n-1, x, 1))
print(math.sqrt(a))
```

```
4.58257569495584
4.58257569495584
```

[ ]: