EE341 Discrete Time Linear System Lab4: Property of DTFT

Acknowledge: This slide is modified from those of previous years.

Introduction

- Frequency spectrum of sound signals
- Smooth the stock signal by filtering high frequency
- Equalizer filter (LP, HP, BP) for music

Background

- The DTFT of aperiodic signal

$$X\left(e^{jw}\right)=\sum_{n=-\infty}^{\infty} x[n]e^{-jwn}$$

  - Aperiodic in time domain --> continuous in frequency domain
- The filter system characterized by the difference equation

$$\sum_{k=0}^{N} a_k y[n-k]=\sum_{k=0}^{M} b_k x[n-k]$$

  - Coefficients of output $A=[a_0,\ldots,a_N]$
  - Coefficients of input $B=[b_0,\ldots,b_N]$
  - Transfer function of the filter $H(z)=\sum_{k=0}^{M} b_k z^{-k} \Big/ \sum_{k=0}^{N} a_k z^{-k}$
  - Let $z=e^{jw}$, then frequency response of the filter

$$H\left(e^{jw}\right)=\sum_{k=0}^{M} b_k e^{-jkw} \Big/ \sum_{k=0}^{N} a_k e^{-jkw}$$

Background

- If $A = [1,0,\ldots,0]$ and $B = [x_0,\ldots,x_M]$, then above equation reduces to

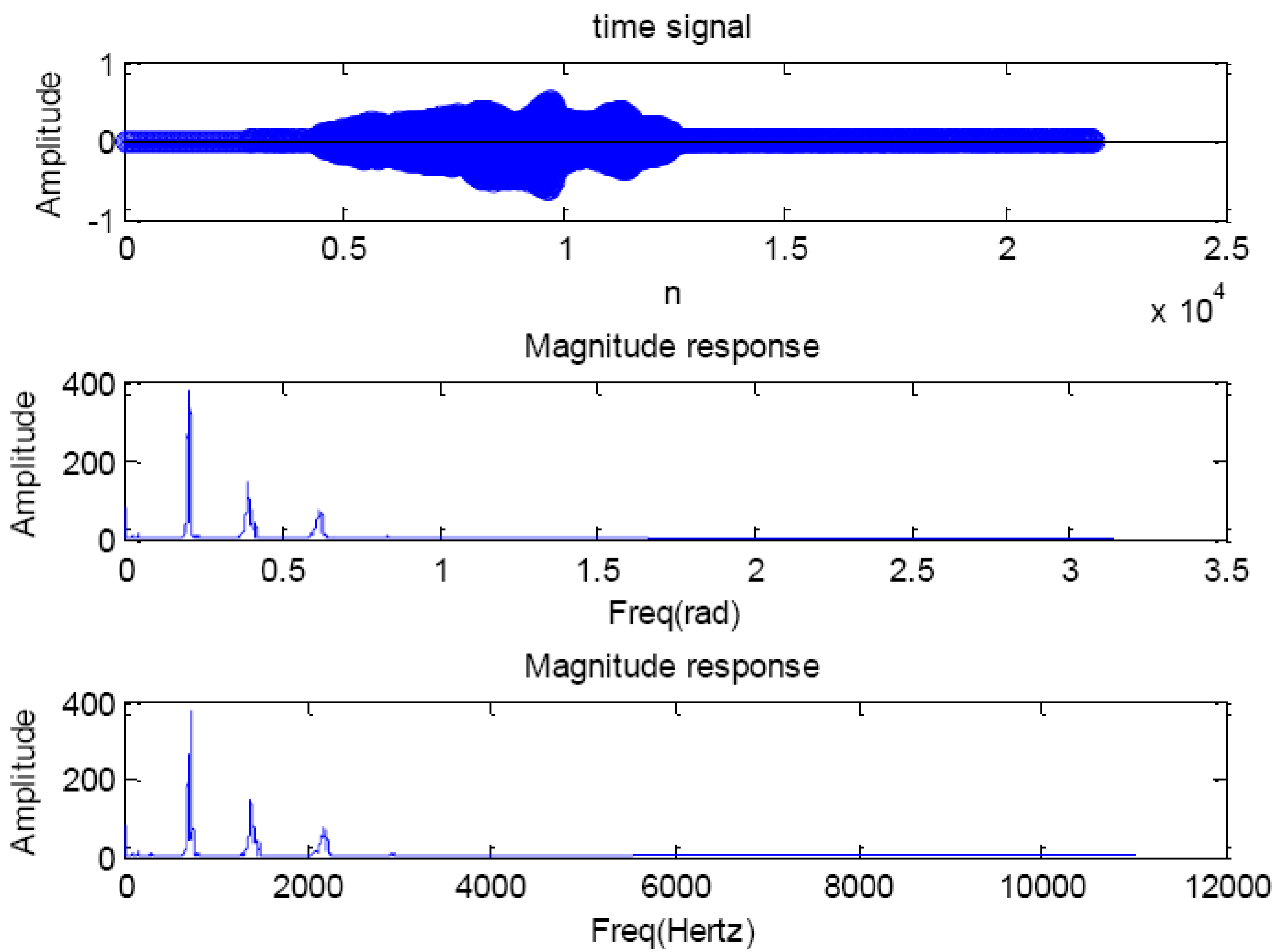
$$H\left(e^{jw}\right) = \sum_{k=0}^{M} x_k e^{-jkw} = X\left(e^{jw}\right)$$

- Matlab command `freqz`
  - Frequency response of digital filter
    - `[H, W] = freqz(B, A, N)`
      - H is the frequency response vector
      - W is angular frequncy vector between 0 to pi
      - B and A are coefficient vectors mentioned above
      - N is length of H and W, usually 512 or 1024 or length of signal
    - `[H, W] = freqz(B, A, N, 'whole')`
      - n samples points around the entire unite circle
      - W is from 0 to 2*pi
    - `[H, F] = freqz(B, A, N, Fs)`
      - Gives the frequency response vector H corresponding to Hertz frequency F
      - f = w*fs/(2*pi)

Task 1: Animal sounds

- Load each sound using `wavread`
  - Eg: `[x, Fs, Nbits] = wavread('cat.wav');`
  - x is sound signals, Fs is sampling frequency, Nbits is number of bits per samples
  - Number of samples = length(x)
  - Duration of signals = Number of samples / sampling frequency
- Plot time sequence, DTFT to angular frequency, DTFT to Hertz frequency
  - Eg: `[Hw W] = freqz(x, 1, length(x));`
  - `[Hf F] = freqz(x, 1, length(x), Fs);`
- Commend on low/medium frequency sound, the fundamental frequency.
- Defining a function may simplify your code

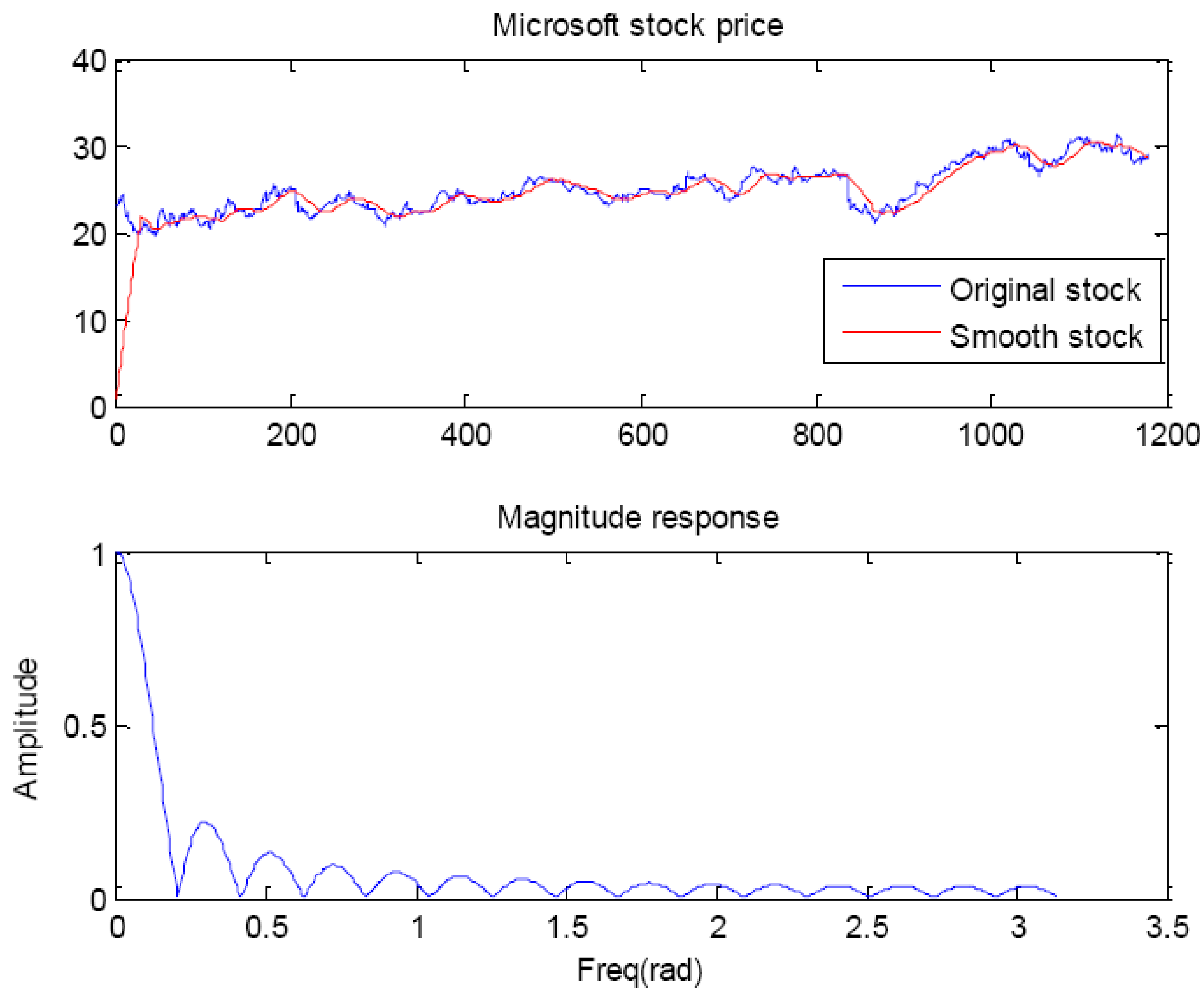
Task 1: Animal sounds



Task 2: Stock

- Load the text file
  - Eg: `load microsoftstock.txt`
- Generate a 30-point moving average filter
  - Eg: `h = 1/30*ones(1, 30);`
  - This is a FIR filter, B = h, A = [1,0, …, 0]
- Filter the stock price by using `filter` command
  - Eg: `y = filter(h, 1, microsoftstock);`
- Compare the change of stock signals
- Plot the frequency response of the moving average filter

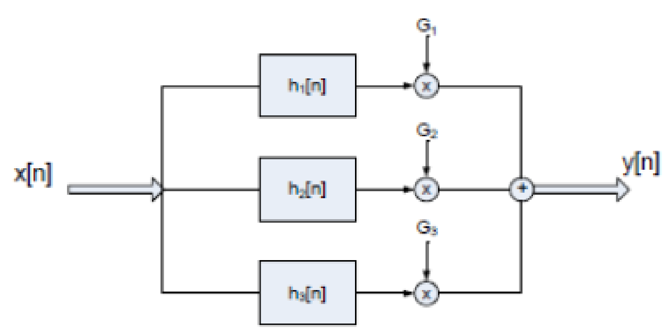
Task 2: Stock



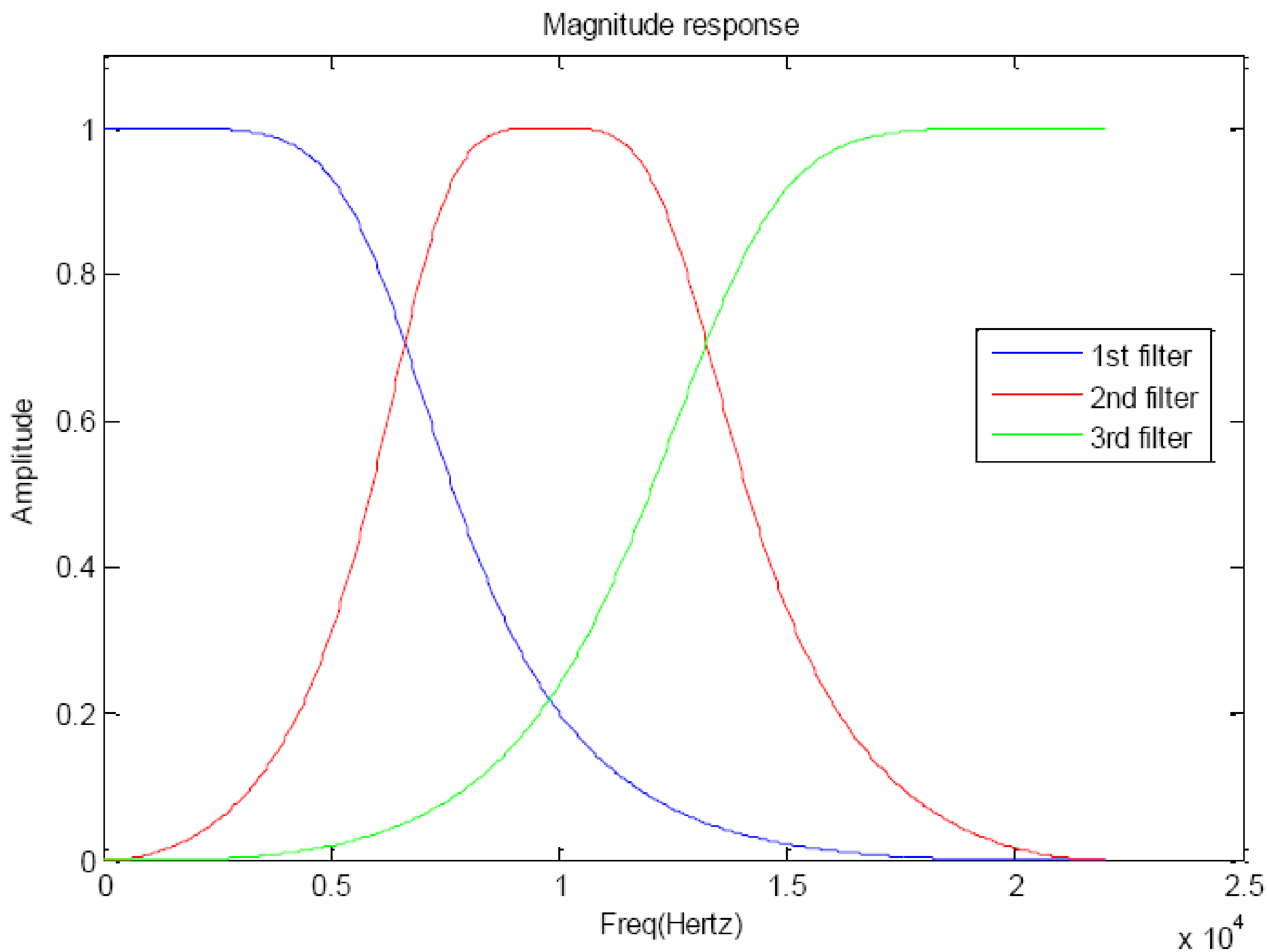
Microsoft stock price
40
30
20
10
0
0
200
400
600
800
1000
1200
Original stock
Smooth stock
Magnitude response
Amplitude
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
Freq(rad)

Task 3: Simple 3-bands equalizer

- Load the music file
  - Fs = 44.1KHz, 16 bit/sample/channel, 2 channel, how many samples of 10 second music?
  - Eg: `[x, Fs, Nbits] = wavread('music.wav'); x_left = x(:,1);x_right = x(:,2);`
- Plot the frequency response of three filters(LPF, BPF, HPF)
- Apply each filter to the signal relatively(3 filters->3channels), and weight each channel by gain, and then sum up
  - Check the sample code in the lab assignment



Task 3: Simple 3-bands equalizer



Updated