BRI509: Introduction to Brain Signal Processing
Assignment No. 4

CANOY RAYMART JAY
Student ID #: **2020021376**

June 21, 2021

1 Explain the following terms briefly.

(a) Sampling theorem The process of sampling involves multiplying the original signal by a pulse train, with each pulse having an area of one, that is,

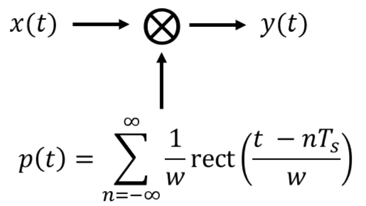


Figure 1: The sampling process

Therefore,

$$y(t) = \sum_{n=-\infty}^{+\infty} x(t) \frac{1}{w} \texttt{rect}\left(\frac{t - nT_s}{w}\right). \tag{1}$$

To get the original value of the signal, the width of the pulse should approach zero:

$$\begin{aligned} x_\delta(t) &= \lim_{w \to 0} \sum_{n=-\infty}^{+\infty} x(t) \frac{1}{w} \texttt{rect}\left(\frac{t - nT_s}{w}\right), \\ x_\delta(t) &= \sum_{n=-\infty}^{+\infty} x(t) \delta\left(t - nT_s\right) \end{aligned} \tag{2}$$

By taking the CTFT of this impulse-sampled signal, we get

$$\begin{aligned} X(f) &= \sum_{n=-\infty}^{+\infty} X(f) f_s \delta\left(f - nf_s\right), \\ X(f) &= f_s X(f) * \delta_{f_s}(f) \end{aligned} \tag{3}$$

The CTFT of this impulse-sampled signal is just the replicas (aliases) of the CTFT of the original signal. The idea in sampling theorem is that the signal should be bandlimited, which means that the non-zero values should be confined to a finite range of frequency, so that the original signal could be recovered from the samples when a filter is applied.

(b) Anti-aliasing Aliasing is the phenomenon where the aliases (replicas of the CTFT of the original signal) overlap with each other. To avoid aliasing, the sampling rate f_s should be greater than twice the highest frequency f_m in a lowpass bandlimited signal. But more generally, the minimum sampling rate should be

$$|f_{s,\mathtt{min}}| > \frac{2f_H}{[f_H/B]}, \tag{4}$$

where f_H and f_L are the highest and lowest frequencies, respectively, in the finite range where the CTFT of the original signal is non zero. B is the bandwidth of the CTFT, i.e., $B = f_H - f_L$.

(c) Bandlimited signal Bandlimited signals are signals which are zero for all values greater than the highest frequency f_m.

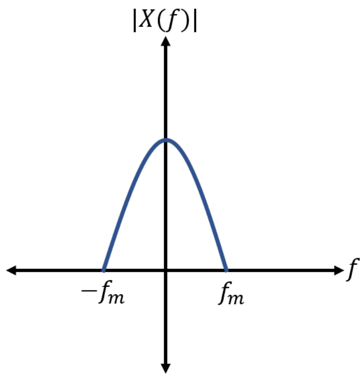


Figure 2: Lowpass bandlimited signal.

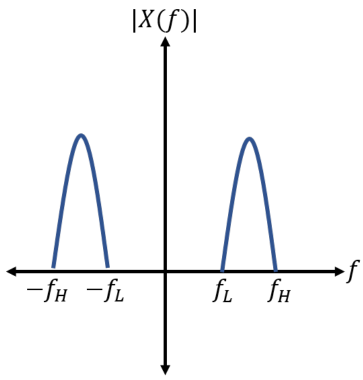


Figure 3: Bandlimited signal with a bandwidth of $B = |f_H| - |f_L|$.

(d) Distortion Distortion is a phenomenon where the shape of the signal is being changed when a filter is used. It should be noted, however, that the multiplication of a signal by a constant and the time shifts are not considered as distortion.

(e) Causal filters Causal filters are filters with nonzero response only after the impulse is applied at time $t = 0$.

(f) Linear phase The impulse response of a distortionless system can be generally described as

$$h(t) = A\delta(t - t_0), \tag{5}$$

and its CTFT can be written as

$$H(f) = Ae^{-j2\pi f t_0}. \tag{6}$$

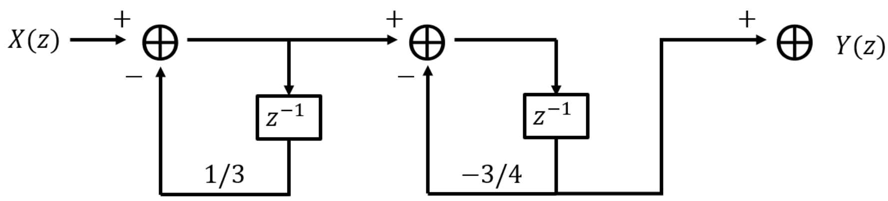
This frequency responses has a constant magnitude and a linear phase, that is, $|H(f)| = A$ and $\angle H(f) = 2\pi f t_0$.

2 Solve the following problems.

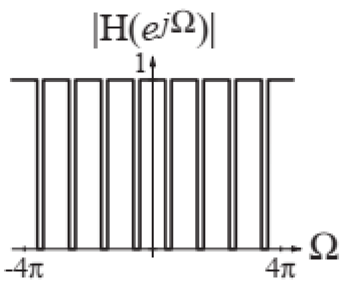
(a) Draw a cascade-form block diagram for the system transfer function.

$$H(z) = \frac{z}{(z + 1/3)(z - 3/4)} \tag{7}$$

Solution:



(b) Classify the frequency responses in the figure as being lowpass, highpass or bandstrop.



Solution:
This is a frequency reponse of a **bandstop filter**. The transfer function can be represented as

$$H(e^{j\Omega}) = Ae^{-j\Omega n_0}\left\{1 - \left[\texttt{rect}\left(\frac{\Omega - \Omega_0}{\Delta\Omega}\right) + \texttt{rect}\left(\frac{\Omega - \Omega_0}{\Delta\Omega}\right)\right] * \delta_{2\pi}(\Omega)\right\} \tag{8}$$

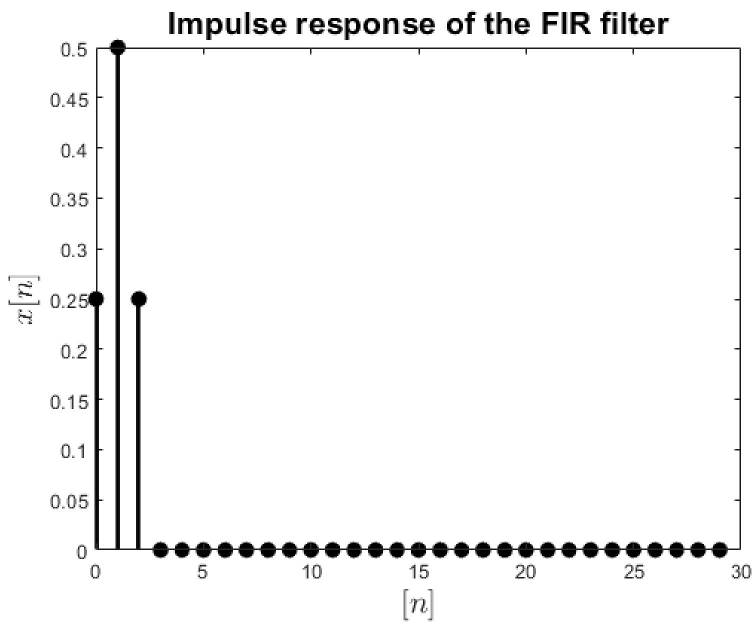
(c) Calculate the filter responses of the following FIR filter.

$$h_N[n] = \sum_{m=0}^{N-1} a_m \delta[n-m], \tag{9}$$

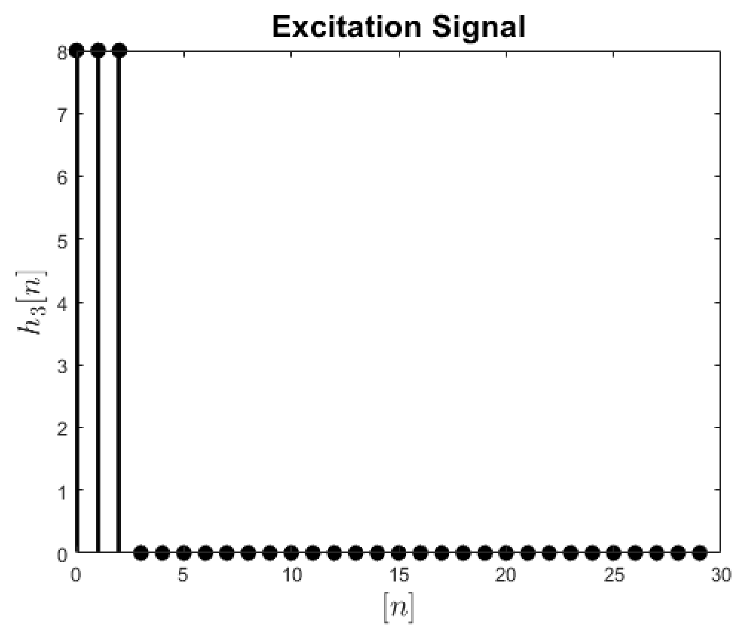
where $N = 0$, $a_0 = 0.25$, $a_1 = 0.50$, $a_2 = 0.25$.

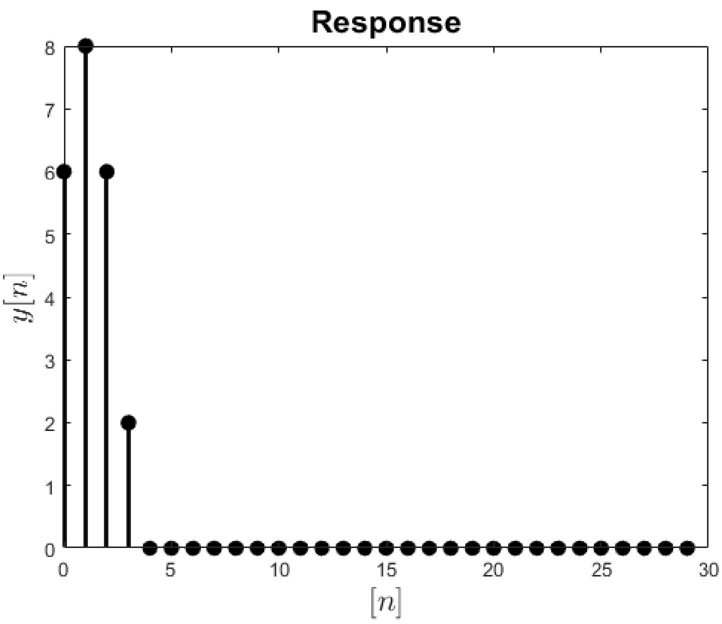
1. Impulse response

$$\begin{aligned} h_N[n] &= a_0\delta[n] + a_1\delta[n-1] + a_2\delta[n-2] \\ h_N[n] &= 0.25\delta[n] + 0.50\delta[n-1] + 0.25\delta[n-2] \end{aligned} \tag{10}$$



2. Excitation: {..., 0, 0, 8, 8, 8, 0, 0, ...}





(3) MATLAB coding. Design the FIR filters to separate do, mi, sol from do-mi-sol chord, respectively.

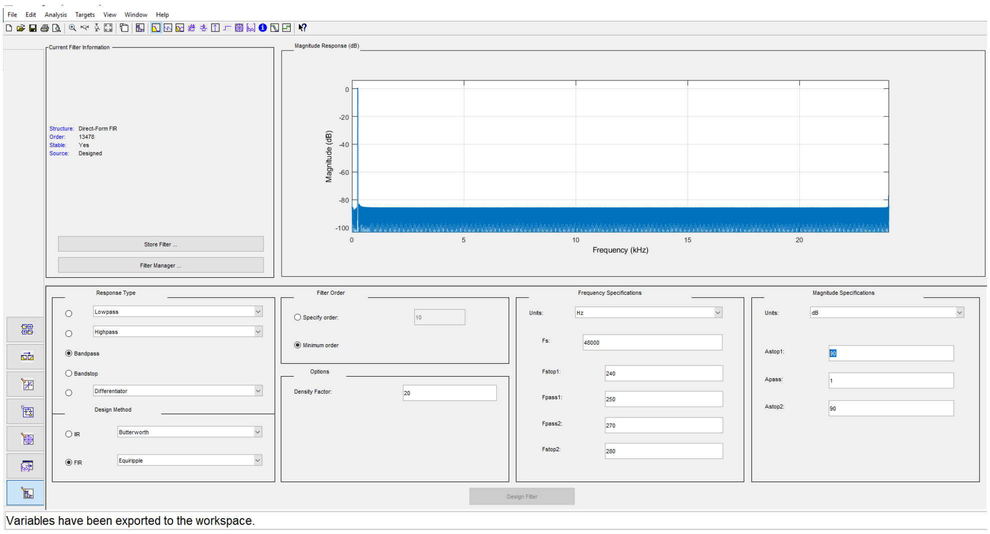
- Source code
- Filter coefficients for do, mi, sol, respectively
- Plot the Bode diagram of the designed filter
- Attach the output MP3 files of the designed filters

Source Code

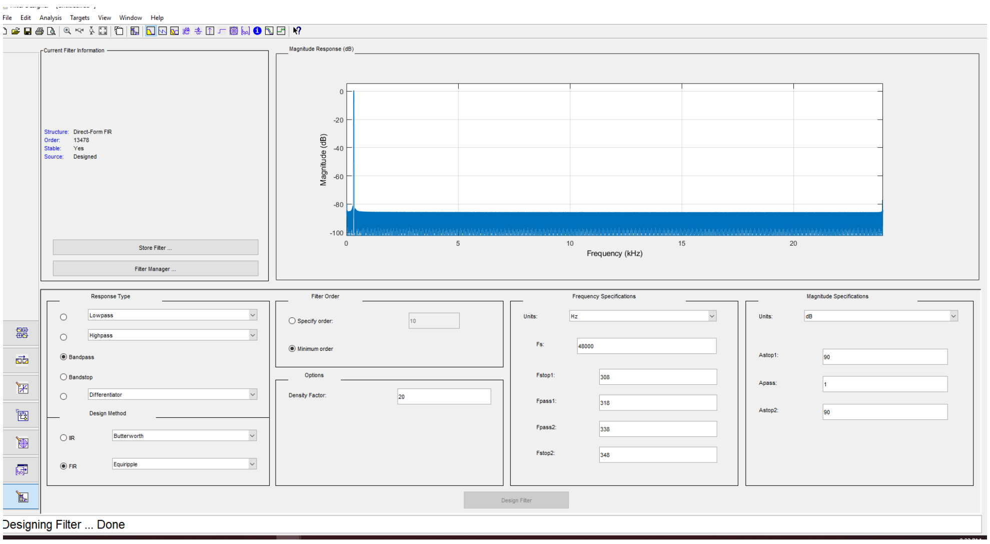
```
% BRI509 (Introduction to Brain Signal Processing)
% Assignment # 4
% Author: Raymart Jay E. Canoy
% Student ID #: 2020021376

% Notes:
% "The frequencies 440 Hz and 880 Hz both correspond to the musical note A,
% but one octave apart. In the western musical scale, there are 12 notes in
% every octave. These notes are evenly distributed (geometrically), so the
% next note above A, which is B flat, has frequency (440 x beta), where
% beta is the twelfth root of two."
% Source: https://ptolemy.berkeley.edu/eecs20/week8/scale.html

%% Draw the Bode diagrams of:
%  (a) Do(C4)
%  (b) Mi(E4)
%  (c) Sol(G4)
%  (d) Chord(C4+E4+G4)

% (00) Initialization
clear; clc;

fs = 48000;;
T = 1/fs;
dur = 1;
L = dur*fs;
t = (0:L-1)*T;

% (01) Sinusoid signal
beta = 2^(1/12);
beta_exponents = [-9, -7, -5, -4, -2, 0, 2, 3];
note = @(index) sin(2*pi*(440*(beta^(index)))*t);

% C major
musical_scale_keys = {'C4', 'D4', 'E4', 'F4', 'G4', 'A4', 'B4', 'C5'};
musical_scale_values = cell(length(musical_scale_keys), 1);

for ind = 1 : length(musical_scale_keys)
    musical_scale_values{ind} = note(beta_exponents(ind));
end

% (02) Sofege for C major
music_octave = containers.Map(musical_scale_keys, musical_scale_values);

%% (03) Chord
chord = music_octave('C4') + music_octave('E4') + music_octave('G4');

%% (04) Filter
path = uigetdir();
```

```
cd(path)
bpf_do_num = load('
    Assignment4_2020021376_CanoyRaymartJay_do_coefficients.mat'
    );
bpf_mi_num = load('
    Assignment4_2020021376_CanoyRaymartJay_mi_coefficients.mat'
    );
bpf_so_num = load('
    Assignment4_2020021376_CanoyRaymartJay_so_coefficients.mat'
    );

%% (05) Individual musical tone

% Do
do = filter(bpf_do_num.bpf_do_num, 1, chord);
audiowrite('Assigment4_2020021376_CanoyRaymartJay_do.mp4', do,
     fs)

% Mi
mi = filter(bpf_mi_num.bpf_mi_num, 1, chord);
audiowrite('Assigment4_2020021376_CanoyRaymartJay_mi.mp4', mi,
     fs)

% So
so = filter(bpf_so_num.bpf_so_num, 1, chord);
audiowrite('Assigment4_2020021376_CanoyRaymartJay_so.mp4', so,
     fs)
```

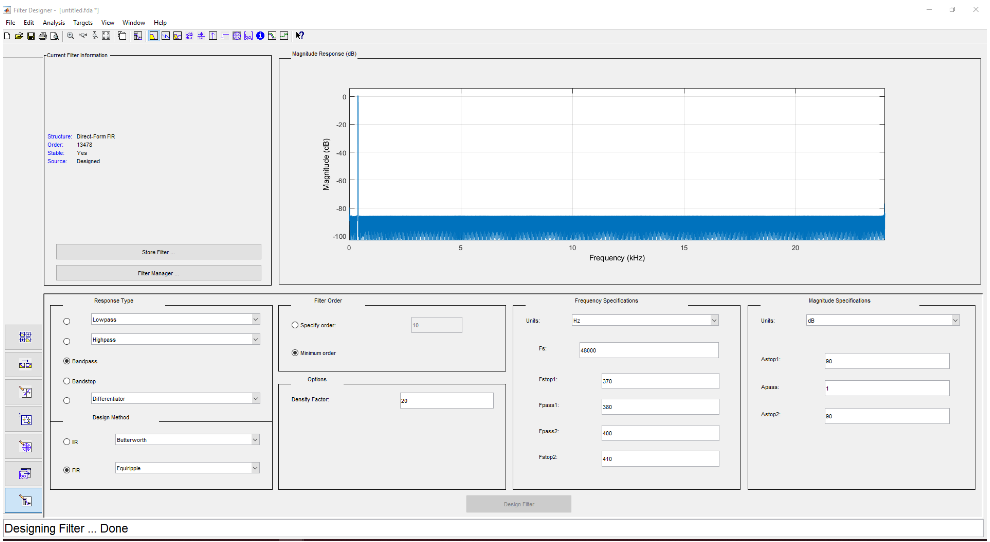
Bode diagram of the designed filter for do:



Bode diagram of the designed filter for mi:



Bode diagram of the designed filter for so:



Note: All the files were uploaded on GitHub

All the files in this document were uploaded on Github, and can be accessed at:

`https://github.com/rjcanoy03/BRI509/tree/Assignment%233`

If there are errors in the solution or codes kindly email, recanoy@korea.ac.kr.