Digital Signal Processing

Section 1

Name of Lecturer:

Assoc Prof Dr Syed Abdul Rahman b. Syed Abu Bakar

Name of group members:

Ho Chun Yeen (A12KE0148)

Hong Sing Min (A12KE149)

Kelvin Ling Chia Hiik (A12KE0153)

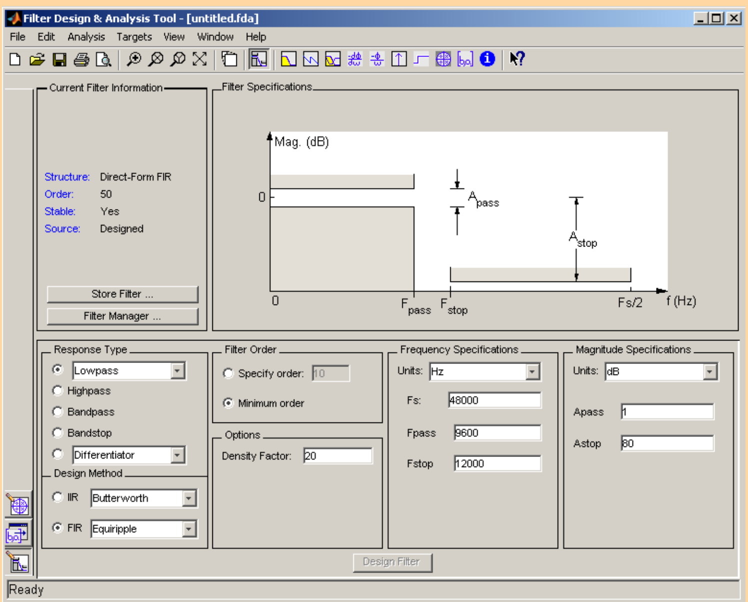
Toh Jun Hau (A12KE0235)

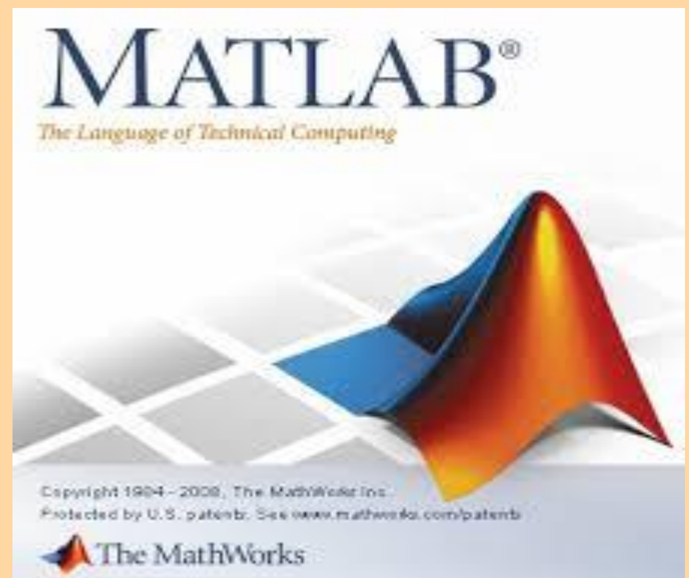
I) Introduction

- We are given a sound signal that comes from 2 sources.
- In this sound, we can hear two distinct sounds.
- The first one is a human voice while the other one is the background windy noise.

II) Objectives

- To separate the human voice and the background noise which is the wind noise using a filter by MATLAB simulation





III) Presentation Outline

- To show the method to achieve our task.
- To state the specification of the filter used such as the order of the filter and cut off frequency used.
- To show the MATLAB code used
- To provide the original unfiltered sample
- To provide the filtered sample
- Comparison between the two sample

Advantages and Disadvantages of FIR

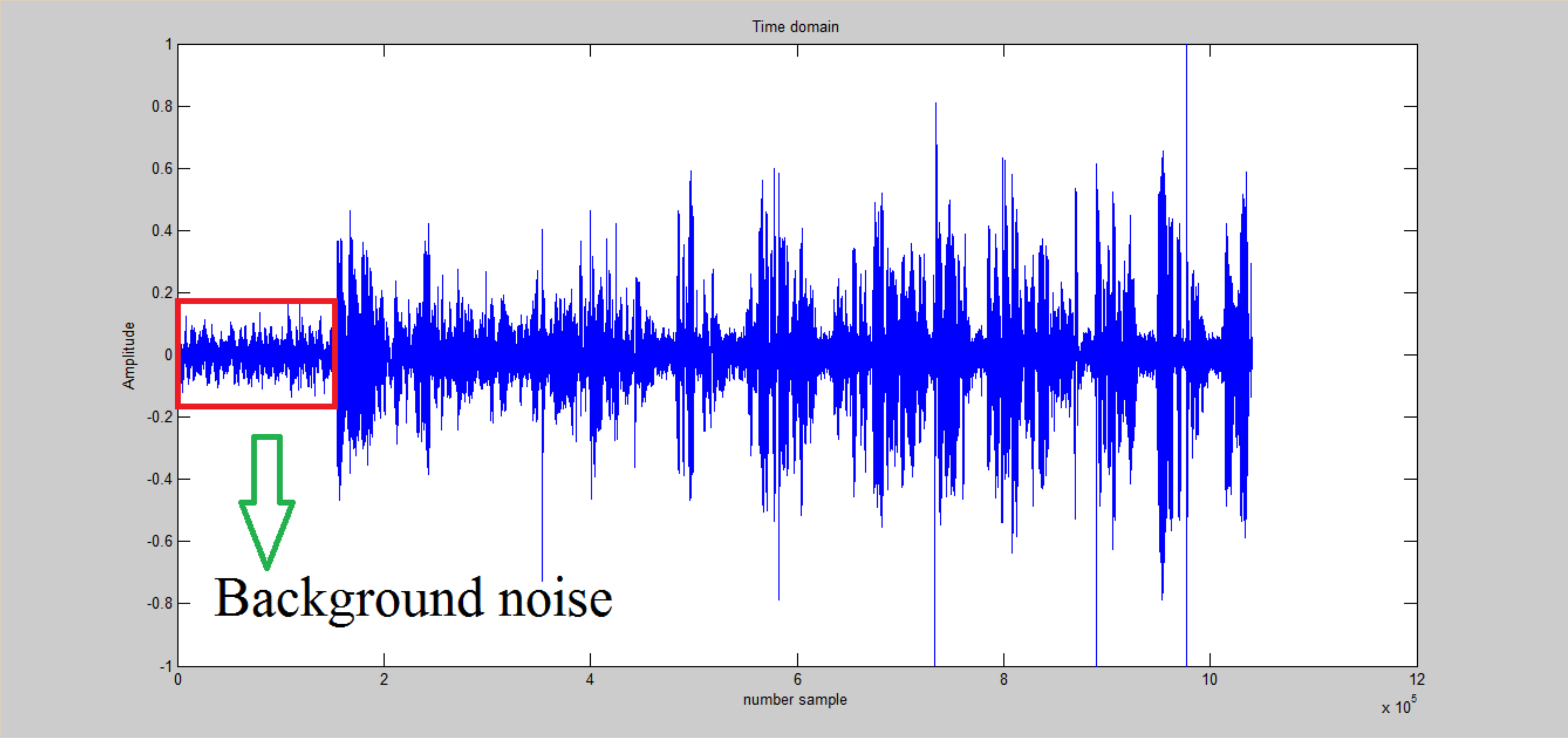
- Advantages
- -Require no feedback
- -Stable
- -Easily be designed to be linear phase
- Disadvantage
- -More computation power

IV) Unfiltered Signal

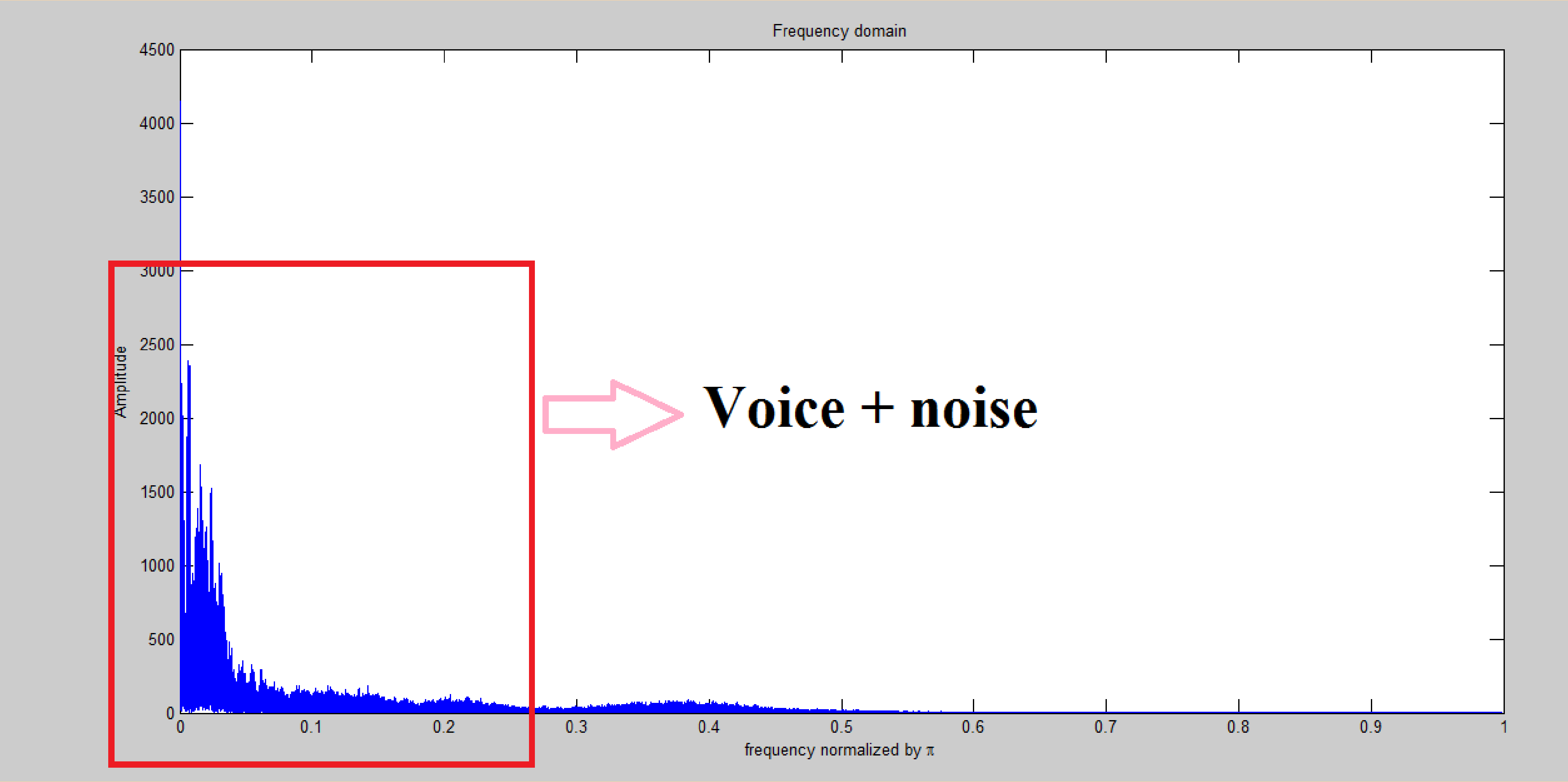
- A) Matlab Code

```
[y,fs] = wavread('Matlab'); %acquire sample
L = length (y);             %acquire sample of y
Y = fft(y,L);               %fast fourier transform
step = 2*pi/L;              %normalize the x-axis
w = 0:step:2*pi-step;

figure(1);                  %plot time domain
plot(y);
title('Time domain');
xlabel('number sample');
ylabel('Amplitude');

figure(2);                  %plot frequency domain
plot(w(1:L/2)./pi,abs(Y(1:L/2)));
title('Frequency domain');
xlabel('frequency normalized by \pi');
ylabel('Amplitude');
```

B) Graph



Time domain



Frequency domain

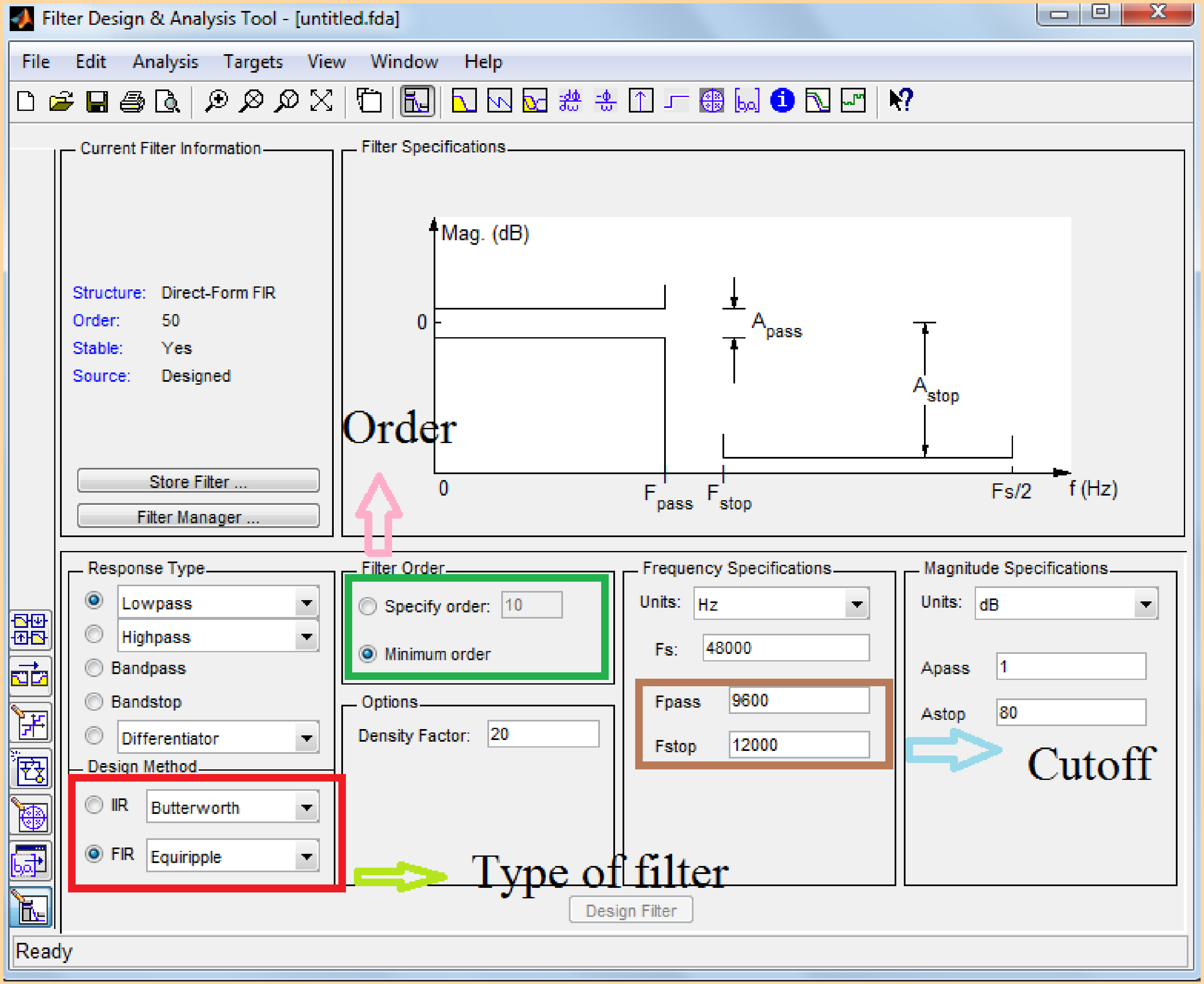
C) Unfiltered sound

V) Design filter

- A)Method

1. Tool use : FDA tool and Matlab
2. Type of filter : Kaiser (FIR digital filter)
3. Type of band : Band pass
4. Cutoff frequency : 1300 ~ 11500 Hz
5. Filter order : 200
6. Beta value : 1

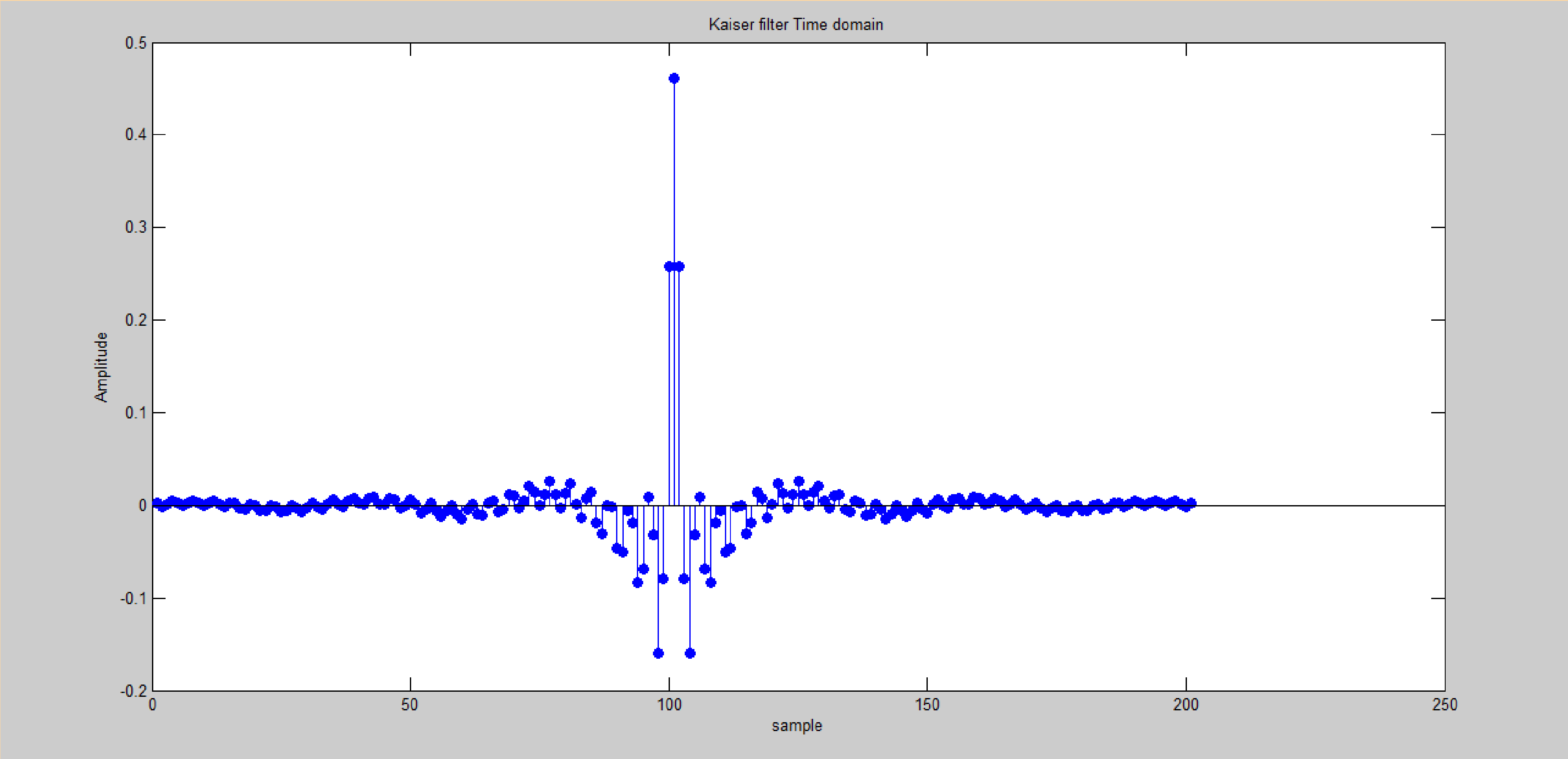
Layout of FDA tool



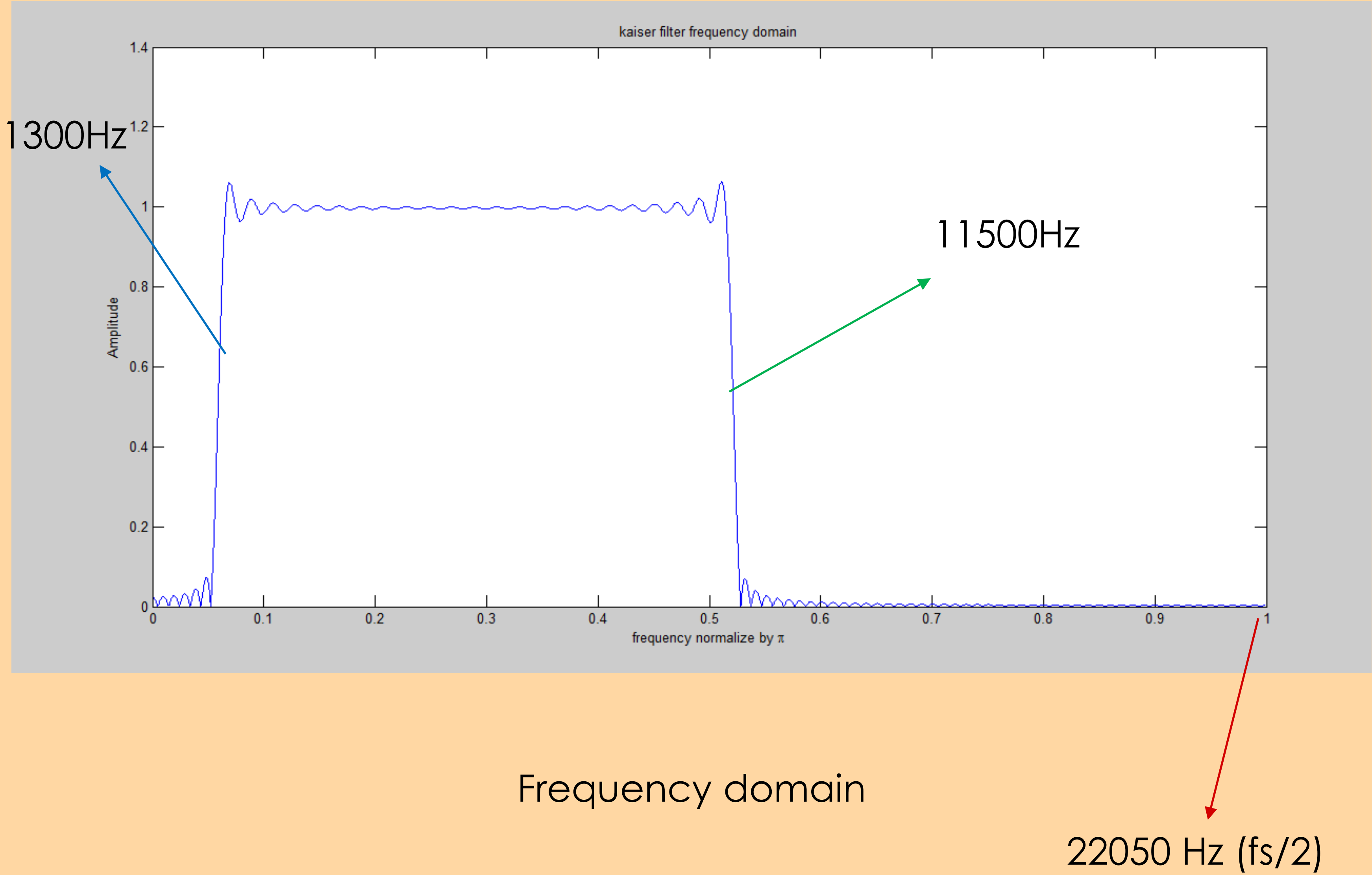
B) Matlab code (filter only)

```
N=200;              %order 200
L = 1040384;        %number of sample
fs = 44100;         %sample frequency
fc1=1300;           %low cutoff
fc2=11500;          %high cutoff
Beta = 1;           %beta value

win = kaiser(N+1,Beta);     %window filter : kaiser
b  = fir1(N,[fc1 fc2]/(fs/2), 'bandpass', win);  %FIR method
B = fft(b,L);             %fast fourier transform the filter with L sample

step = 2*pi/L;        %normalize x axis
w = 0:step:2*pi-step;

figure(1);            %plot time domain
stem(b,'filled');
title('Kaiser filter Time domain');
xlabel('sample');
ylabel('Amplitude');

figure(2);            %plot frequency domain
plot(w(1:L/2)./pi,abs(B(1:L/2)));
title('kaiser filter frequency domain');
xlabel('frequency normalize by \pi');
ylabel('Amplitude');
```

C)Graph (filter only)



Time domain



Frequency domain

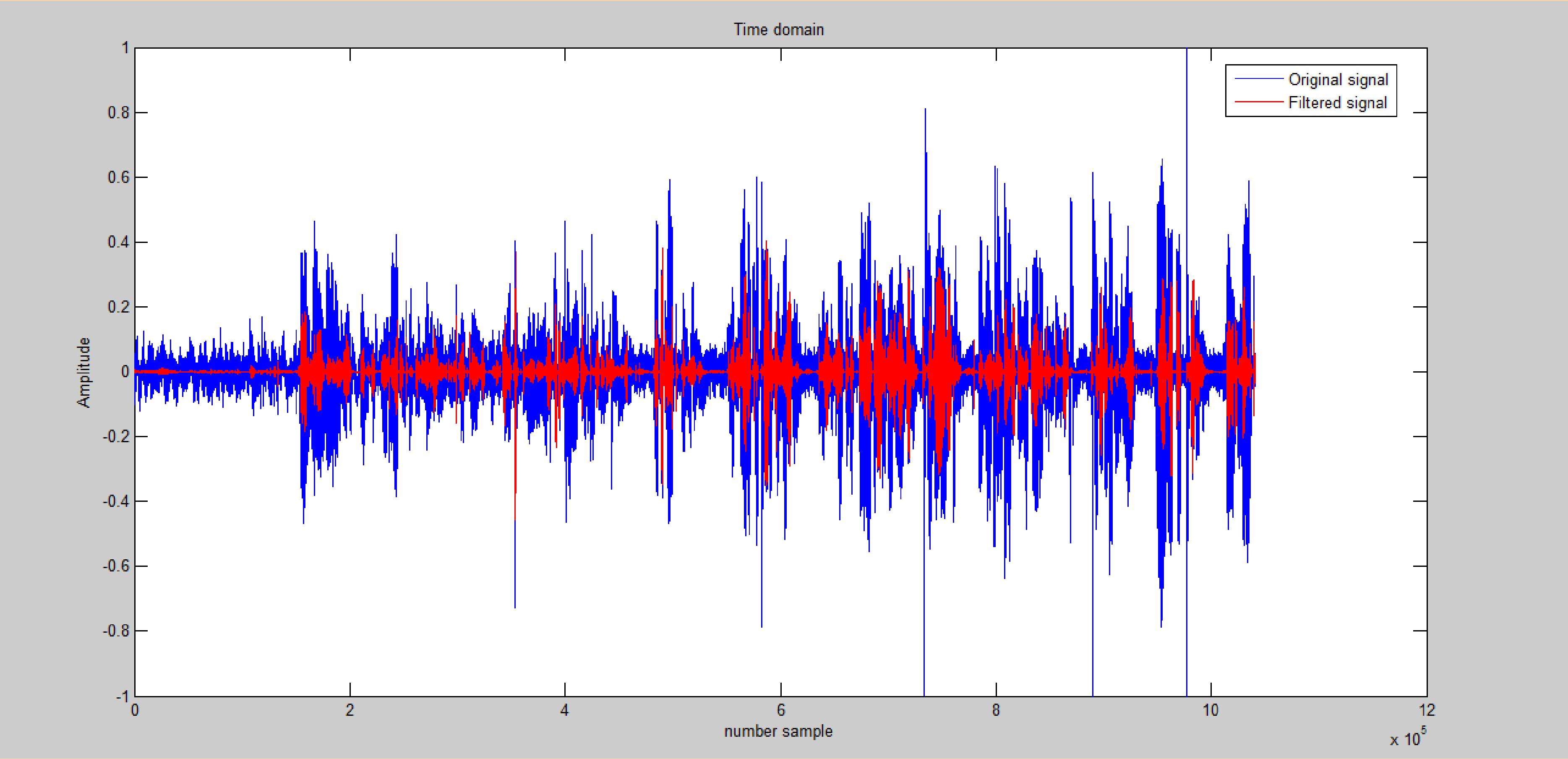
VI) Filtered signal

- A) Matlab code

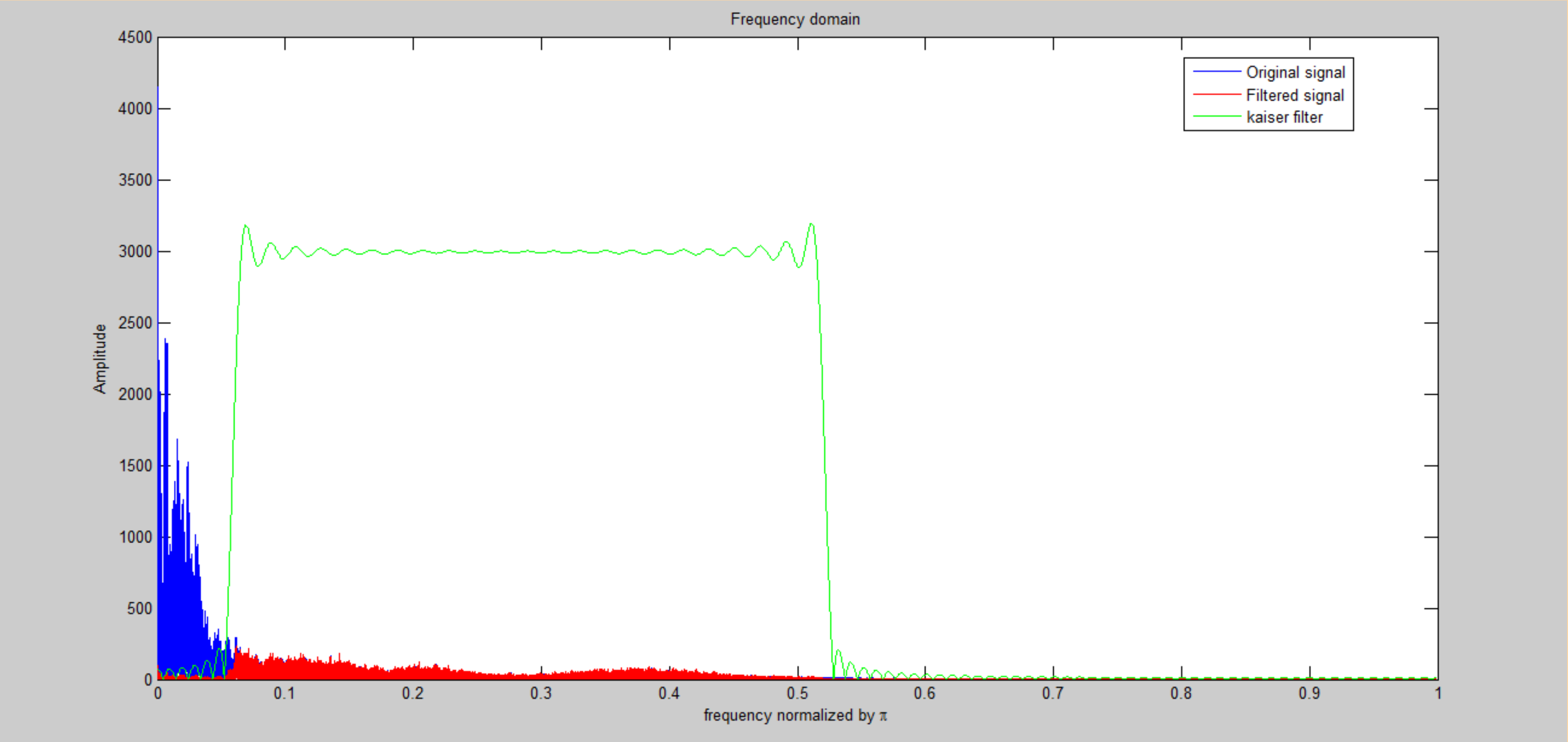
```
%Step 1: take sample
[y,fs] = wavread('Matlab'); %acquire sample
L = length (y);             %acquire sample of y
Y = fft(y,L);               %fast fourier transform

%Step 2: using FDAtool create kaiser code
N=200;          %order 200
fc1=1300;       %low cutoff
fc2=11500;      %high cutoff
Beta = 1;       %beta value

win = kaiser(N+1,Beta);    %window filter : kaiser
b  = fir1(N,[fc1 fc2]/(fs/2), 'bandpass', win);  %FIR method
B = fft(b,L);          %fast fourier transform the filter with L sample

%Step 3: filter signal
filt = dfilt.dffir(b);    %create the filter
f=filter(filt,y);         %filter signal
F = fft(f,L);             %fast fourier transform the filter with L sample

step = 2*pi/L;       %normalize x axis
w = 0:step:2*pi-step;

%Step 4: plot graph
figure(1);                  %plot time domain
plot(y);
hold;
plot(f,'red');
title('Time domain');
xlabel('number sample');
ylabel('Amplitude');
```

```
figure(2);                      %plot frequency domain
plot(w(1:L/2)./pi,abs(Y(1:L/2)));
hold;
plot(w(1:L/2)./pi,abs(F(1:L/2)),'red');
plot(w(1:L/2)./pi,3000*abs(B(1:L/2)),'green');
title('Frequency domain');
xlabel('frequency normalized by \pi');
ylabel('Amplitude');

%Step 5: write wav file
wavwrite(f,fs,'filtered');
```

B) Graph



Time domain



Frequency domain

C) filtered sound

VII) Finding

A)Kaiser formula and beta

$$N = \frac{-20 \log \sqrt{dp * ds} - 13}{14.6 \frac{(Fa - Fp)}{Fs}}$$

Fa = Stop band frequency

Fp = pass band frequency

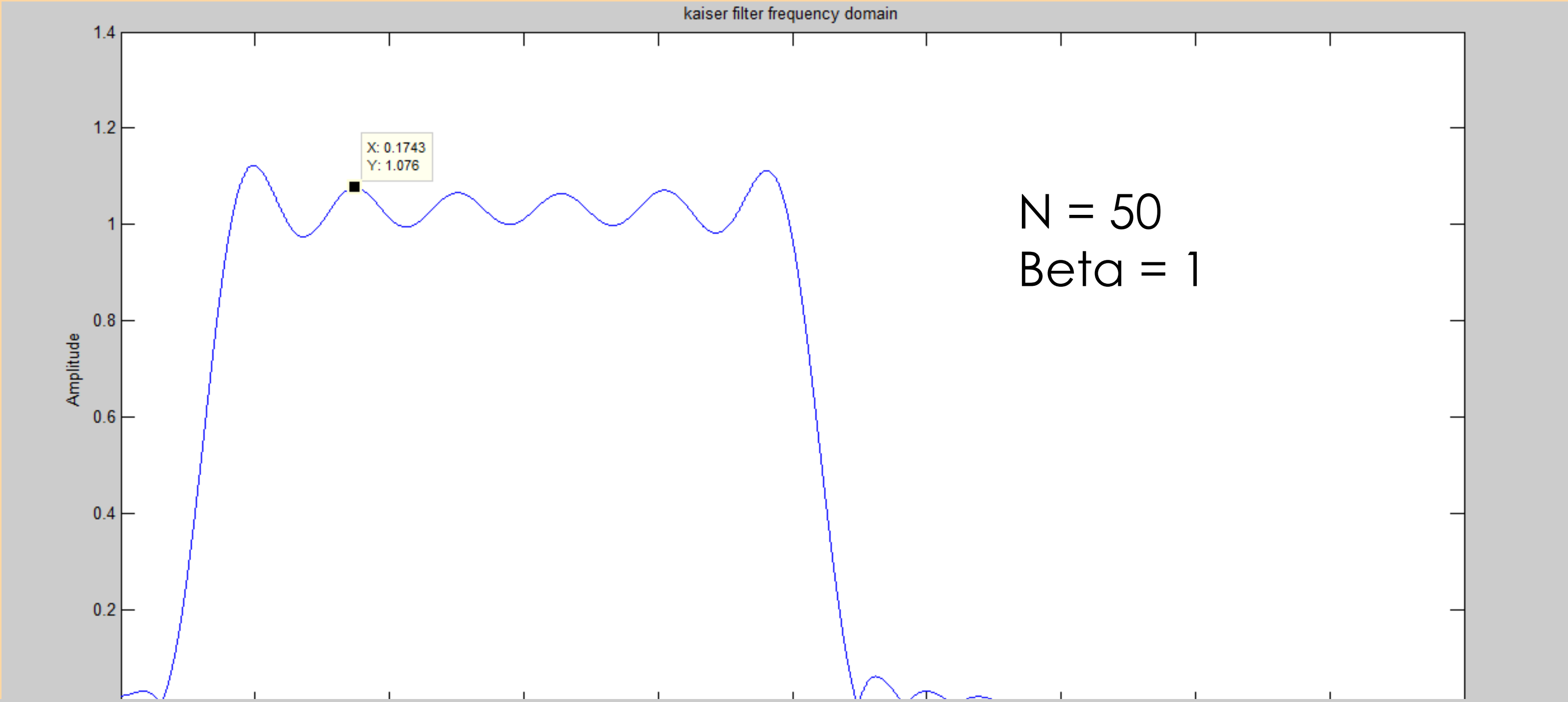
Fs = sample frequency

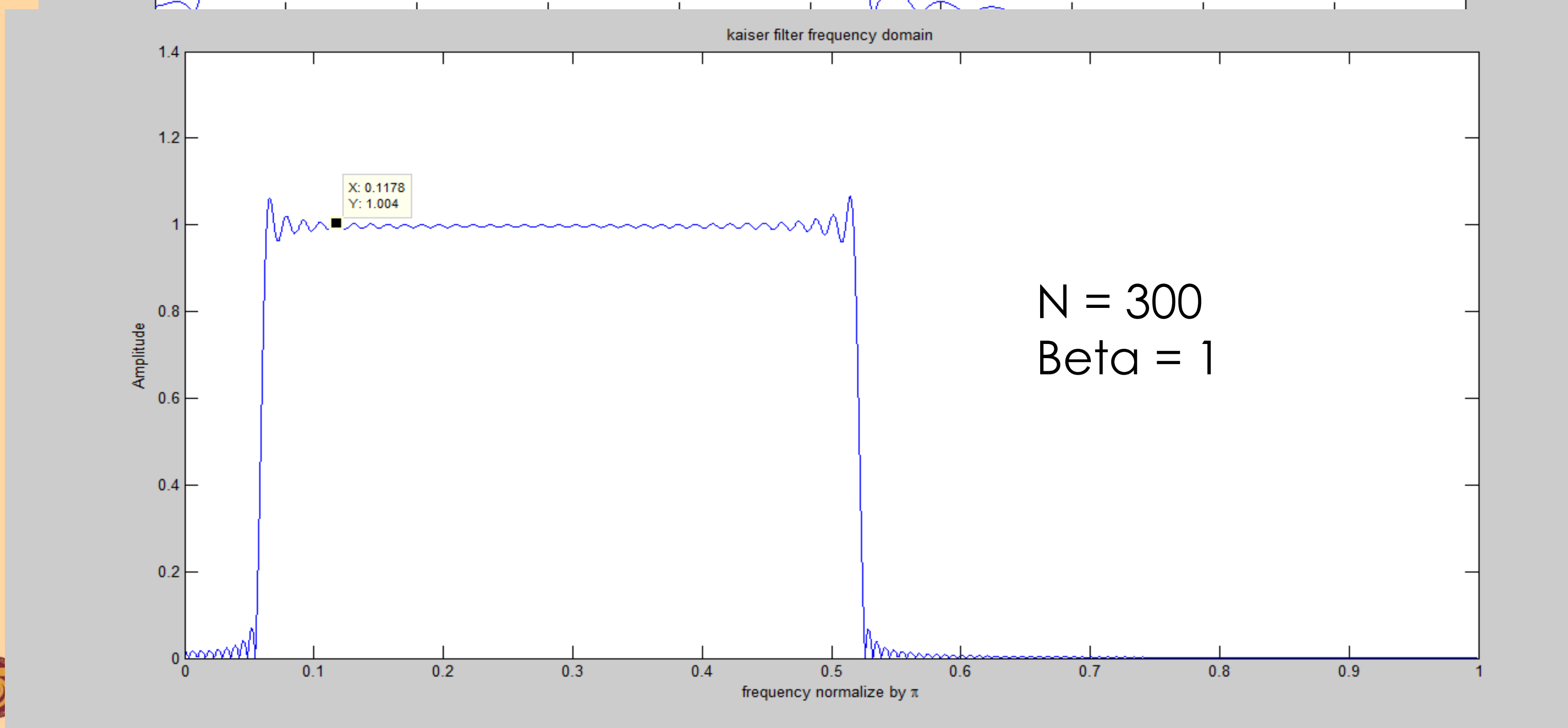
dp = maximum pass band ripple

ds = maximum stop band ripple

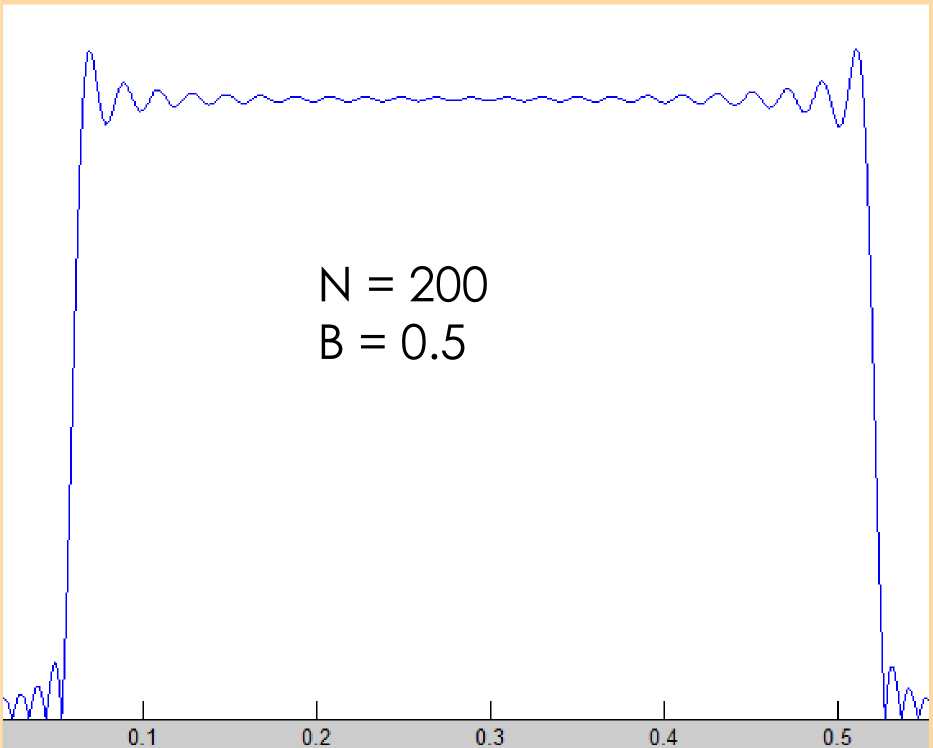
When order(N) increase:

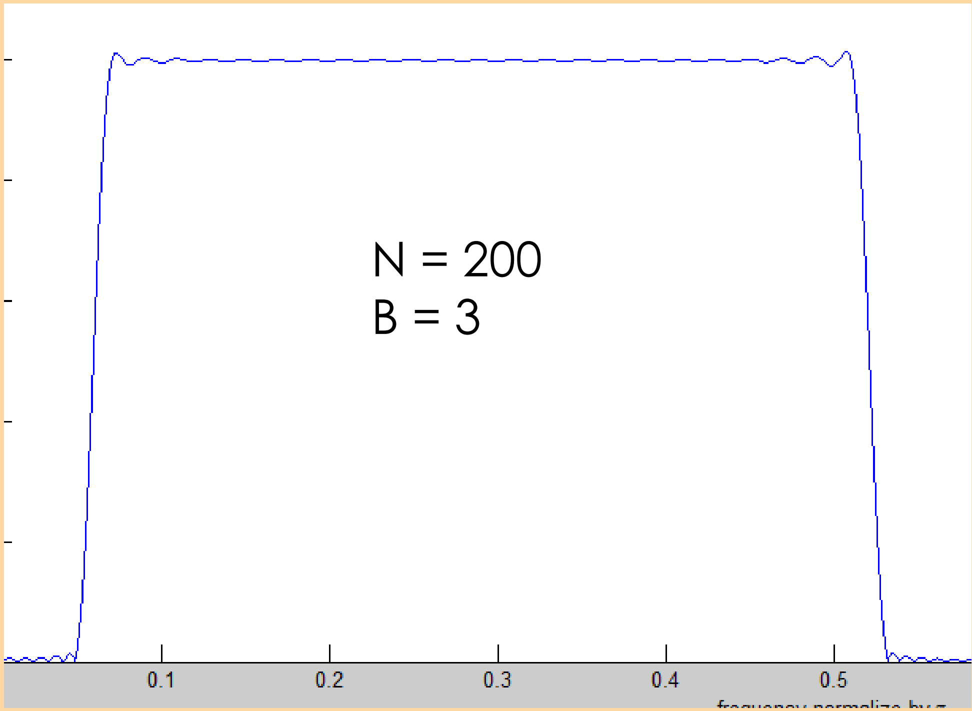
- dp*ds increase cause less distortion
- Transition band (Fa-Fp) decrease , capture less sample which causes decrease in volume
- Filter more noise


kaiser filter frequency domain
Amplitude
X: 0.1743
Y: 1.076
N = 50
Beta = 1


kaiser filter frequency domain
Amplitude
X: 0.1178
Y: 1.004
N = 300
Beta = 1
frequency normalize by π

$$\beta = \begin{cases} 0.1102(\alpha - 8.7), & \alpha > 50 \\ 0.5842(\alpha - 21)^{0.4} + 0.07886(\alpha - 21), & 50 \geq \alpha \geq 21 \\ 0, & \alpha < 21 \end{cases}$$





Alpha is side loop attenuation (maximum height of ripple)


Q
&
A


THANK YOU