[redacted] [redacted] **Summary Sheet**

Name : Kriuk Fedor Student ID : [redacted]

Objective

- To be familiar FS and frequency response

[redacted]

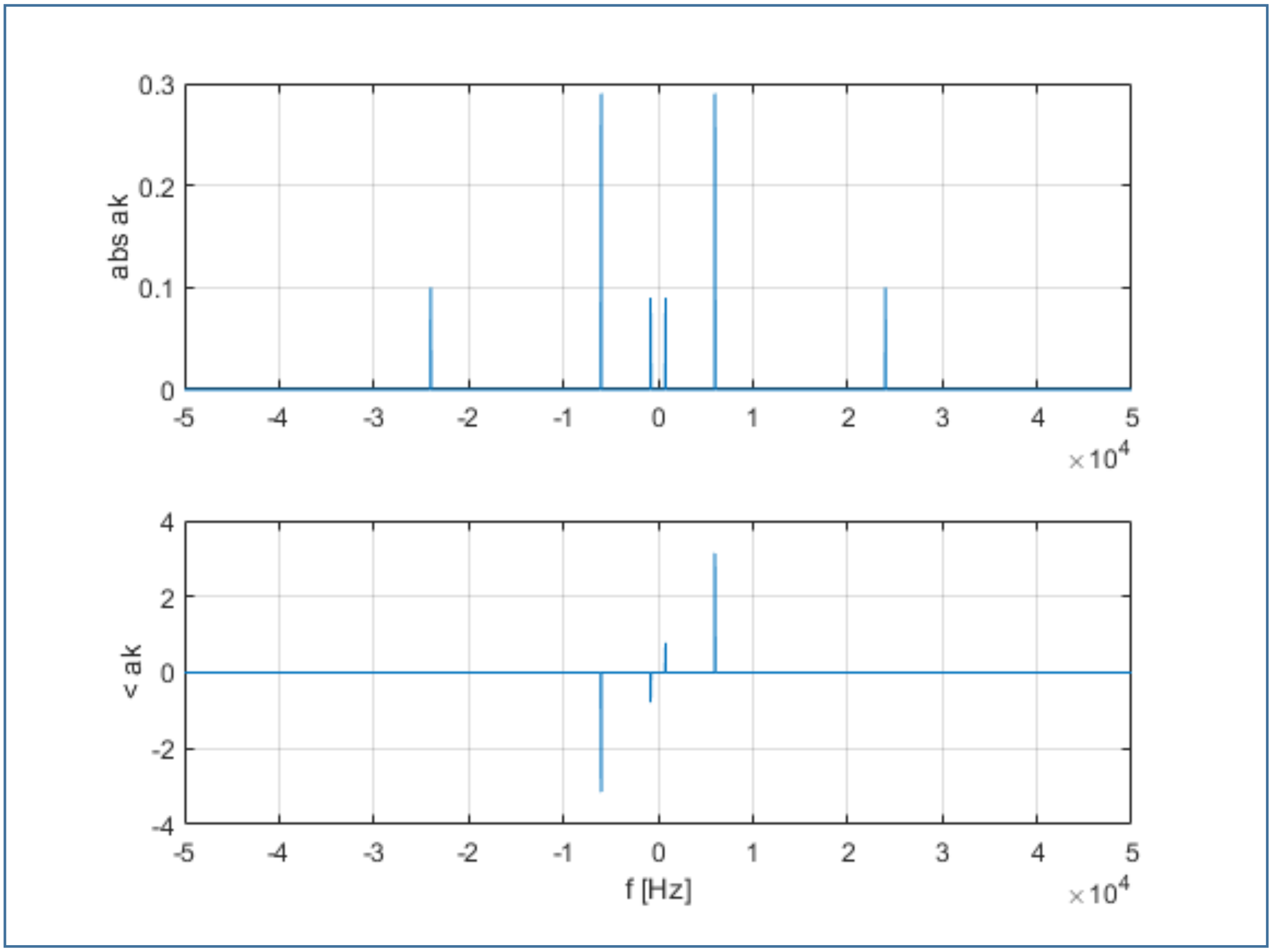
Part I **(10)**

a) Download the sample file (**sample3a.wav**) from [redacted]
b) Use “**audioread**” to read the file.
c) What is the sample frequency (in kHz) ?

(1) **The sampling frequency = 100 kHz**

d) Define a time index using the sample frequency.
e) Use “**fft**” to obtain the FS coefficients.
f) Define a frequency index from – fs/2 to fs/2 (in Hz).
g) Use “**subplot**” , “**plot**”, “**fftshift**” , “**abs**” and “**angle**” to plot the magnitude and the phase of FS versus the actual frequency (Hz) in figure(1).

(1) **figure(1)**



h) What is the fundamental frequency (in Hz) ?

(1) **Fundamental frequency = 2 Hz**

i) Use Data Tips to observe the values and fill in the following table. Only write down the FS for the positive value of k.

(2)

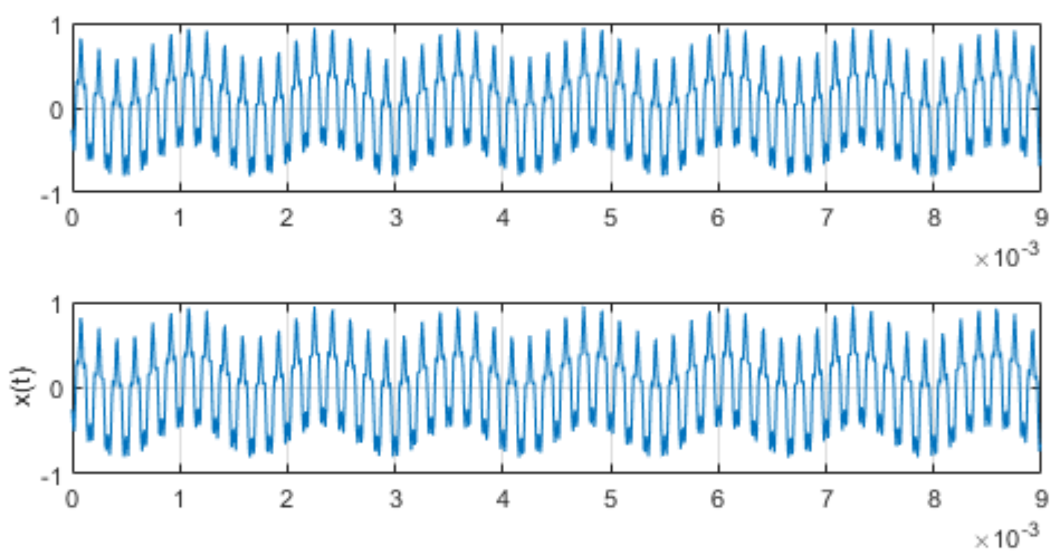
k	Actual frequency (in Hz)	$\vert a_k \vert$	$\angle a_k$ (in terms of π)
1	**806**	**0.0899**	**pi/4**
2	**6000**	**0.29**	**pi**
3	**24000**	**0.1**	**0**

j) Write down the time-domain expression of the audio file as the sum of real **<u>cosine</u>** signals.

(2) $\boldsymbol{sample3c(t)}$ = **2*(0.0899cos((806)*2pi*t+pi/4) + 0.29cos((6000)*2pi*t+pi) + 0.1cos((24000)*2pi*t))**

k) Use "**subplot**" and "**plot**" to plot the audio file versus time and the equation provided in (j) versus time in figure(2). Only show the first **900** points.

(1) **figure(2)**



(2) **Screenshot of Matlab code for Part I**

lab3_1.m

```
[x, fs] = audioread('sample3a.wav');

t = [0:length(x)-1]/fs;
ak = fft(x)/length(x);
f = [-length(x)/2 : length(x)/2 - 1]*fs/length(x);
akp = (abs(ak)>0.001).*angle(ak);

figure(1);
subplot(211); plot(f, abs(fftshift(ak))); ylabel('abs ak'); grid;
subplot(212); plot(f, fftshift(a)); ylabel('< ak'); grid;
xlabel('f [Hz]');

figure(2);
y = 2*(0.0899*cos((806)*2*pi*t+pi/4) + 0.29*cos((6000)*2*pi*t+pi) + 0.1*cos((24000)*2*pi*t));
subplot(311); plot(t(1:900), x(1:900)); grid;
subplot(312); plot(t(1:900), y(1:900)); grid;
ylabel('x(t)');

```

Part II **(10)**

Design a Butterworth **bandpass** filter to complete the following task.

Let N = 8 Sample frequency = fs given in Part I (i.e. audio file)

a) Determine the cutoff frequencies (in Hz) to **completely** remove the **lowest** and the **highest** frequency components of the audio file.

(1) **Lower cutoff in Hz = 24000 Hz** **Higher cutoff in Hz = 12100 Hz**

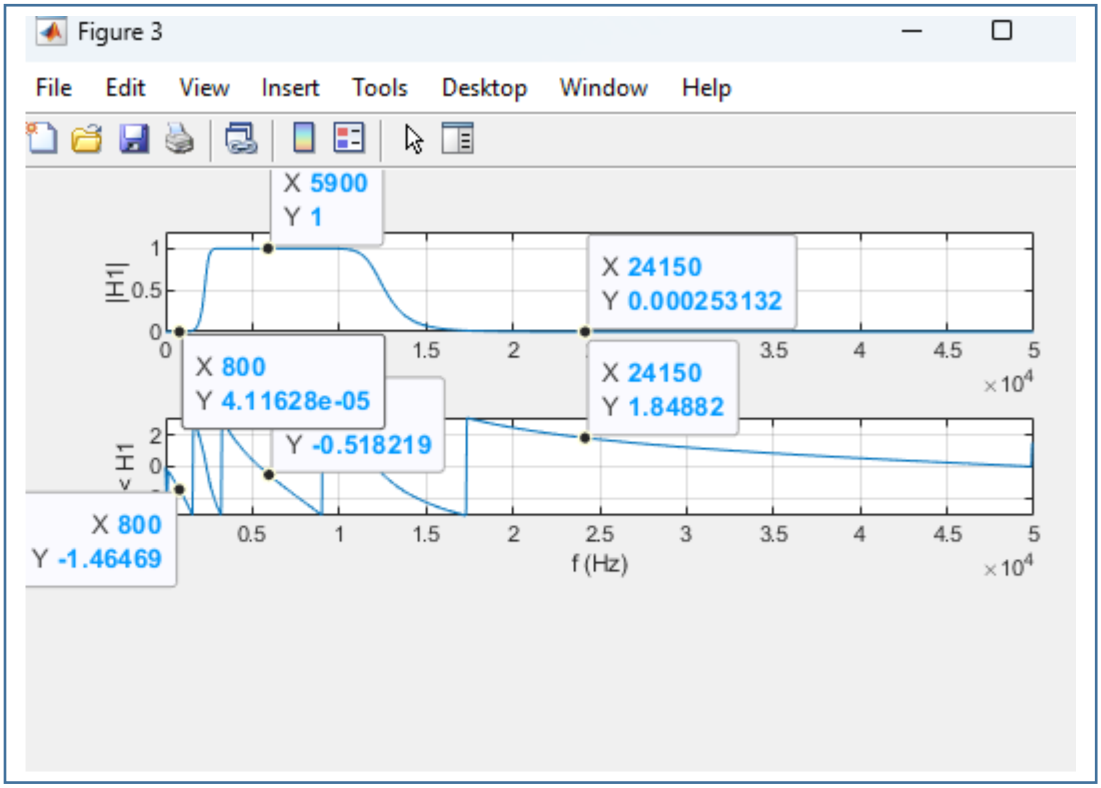
b) Write down the values of Wn according to (a).

(1) **Normalized lower cutoff = 0.016** **Normalized higher cutoff = 0.47**

c) Use "**butter**" and "**freqz**" to generate the required frequency response.
d) Use "**subplot**" , "**plot**" , "**abs**" and "**angle**" to plot the magnitude response and phase response versus frequency (Hz) in figure(3).

If the audio file is applied to your designed bandpass filter to give the output y,
e) Use Data Tips to show the magnitude response and phase response introduced to each frequency component of the audio file.

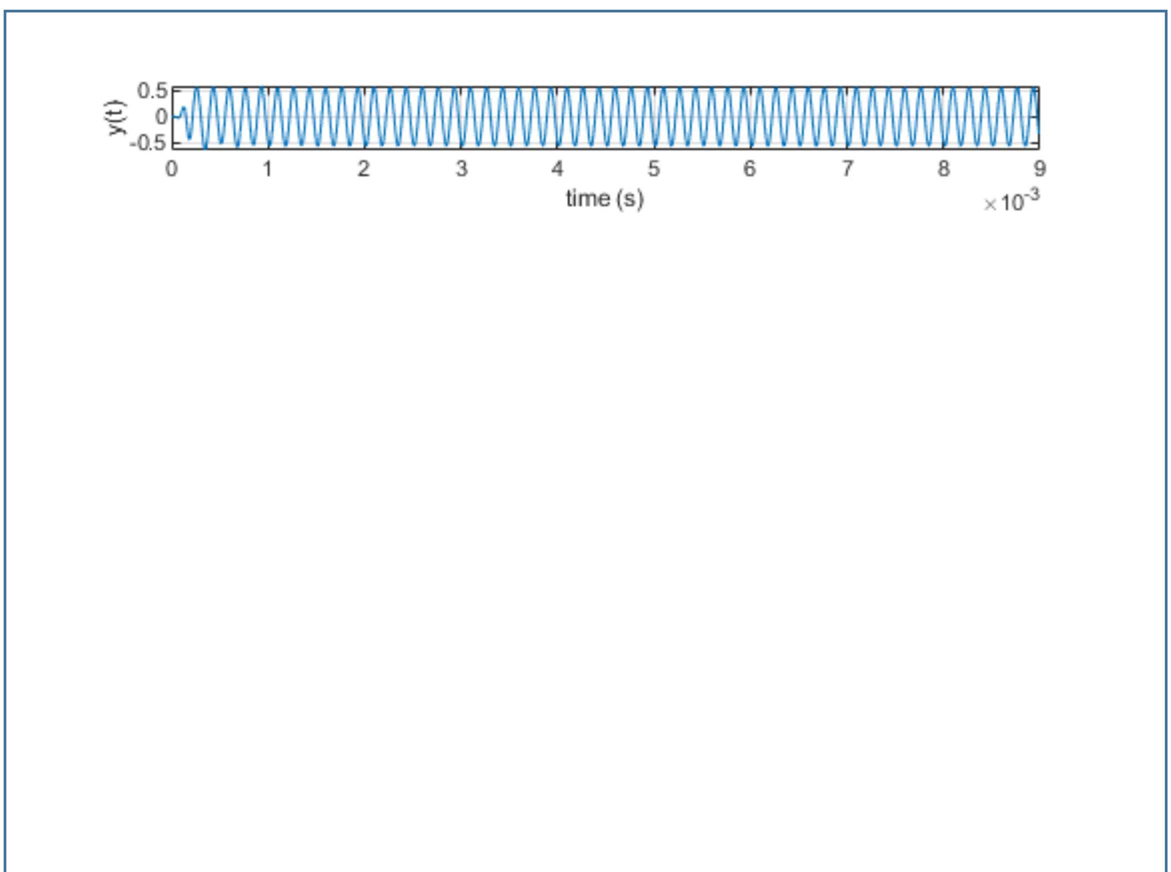
(2) **figure(3) with 6 Data Tips (3 shown in the magnitude response and 3 shown in the phase response)**



f) Perform filtering using "**filter**" to obtain the output y.

g) Plot the first **900** points of the output y versus time in figure(4).

(1) **figure(4)**



h) Write down the expression of the output y as **one** real **cosine** signal.

(1) $y(t) =$ 2*(0+0.29cos((6000) * 2pi * t + pi) + 0)

i) Describe the difference on the output y (time domain) between using butterworth filter and theoretical ideal filter.

(2)

Theoretical ideal filer has a sharp cutoff with no imperfections, while a butterworth filer has a smoother transition and some imperfections.

(2) **Screenshot of Matlab code for Part II**

lab3_1_code.m | lab3_2.m | +

```
N = 8;
t = 0:1/fs:1-1/fs;

[B1, A1] = butter(N, [0.047, 0.242]);
[H1, fh] = freqz(B1, A1, 1e3, fs);

figure(3)
subplot(411); plot(fh,abs(H1)); axis([0 fs/2 0 1.2]); grid; ylabel('|H1|');
subplot(412); plot(fh,angle(H1)); axis([0 fs/2 -pi pi]); grid; ylabel('< H1')
xlabel('f (Hz)');

figure(4);
y = filter(B1, A1, x);
subplot(511); plot(t(1:900), y(1:900)); grid;
ylabel('y(t)'); xlabel('time (s)');

```