UNIVERSITY OF TEHRAN

Engineering Mathematics

Report for Computer Assignment 1

Fourier Transform and its application

Danial Saeedi

Student Number : 810198571

Table of Contents

Page #

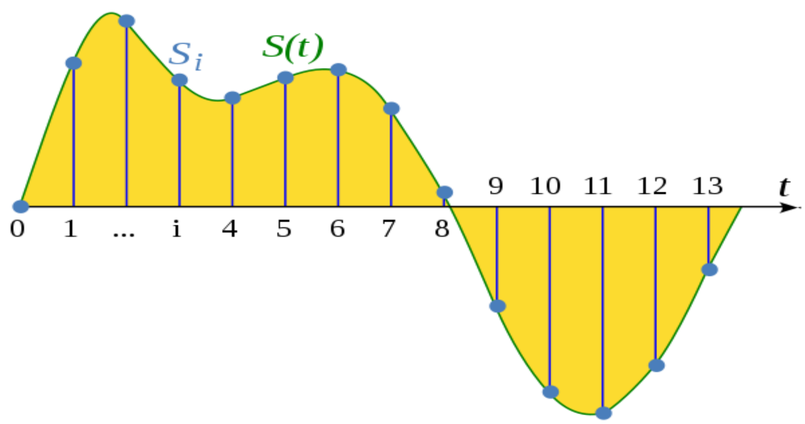
Part 1 : Getting started with Matlab

Question 1 : What is frequency sampling?

In signal processing, **sampling** is the reduction of a continuous-time signal to a discrete-time signal. A common example is the conversion of a sound wave (a continuous signal) to a sequence of samples (a discrete-time signal).

For functions that vary with time, let $s(t)$ be a continuous function (or "signal") to be sampled, and let sampling be performed by measuring the value of the continuous function every T seconds, which is called the **sampling interval** or **the sampling period**.

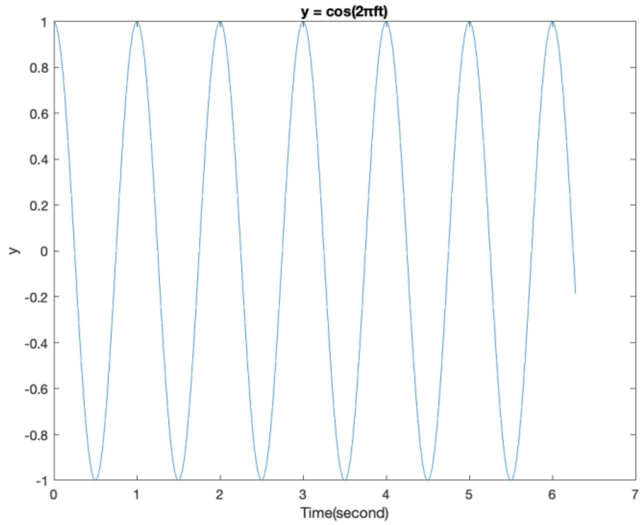
The **sampling frequency** or **sampling rate, fs,** is the average number of samples obtained in one second (*samples per second*), thus **fs = 1/T**.



The green curve indicates continuous signal and blue dots indicate the samples.

Question 2 : Draw y = cos(2πft) in time domain using plot function

```
f = 1;
time_step = 0.01;
%% Sample Frequency
fs = 1/time_step;
t = 0:time_step:2*pi;
y = cos(2*pi*f*t);
plot(t,y)
xlabel("Time(second)");
ylabel("y");
title("y = cos(2πft)");
```

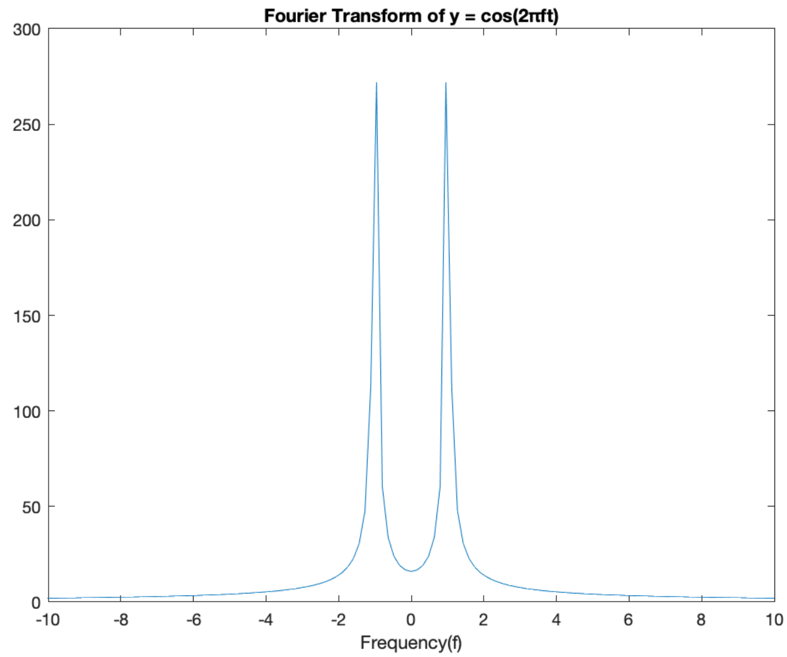


Question 3: Build frequency vector

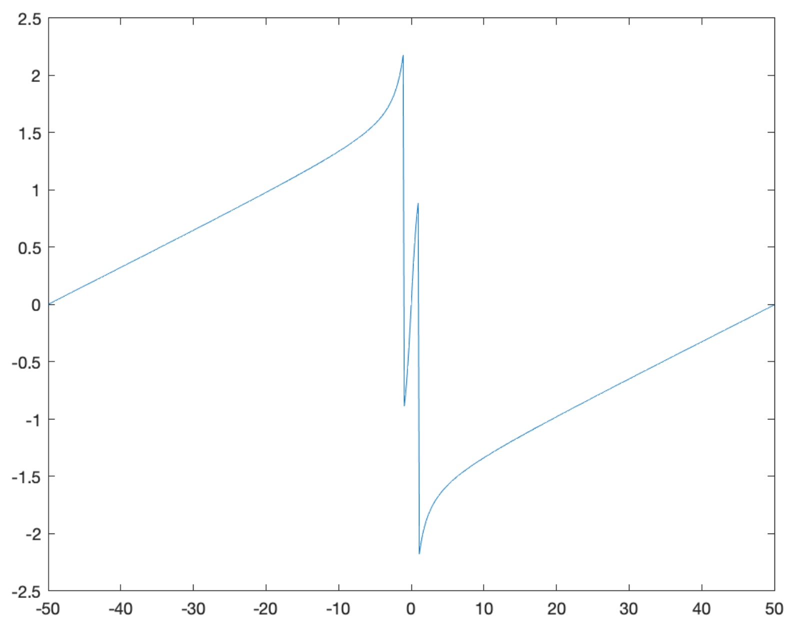
numel(y) calculates number of elements in y vector.

The second line generates numel(y) points and the space between the points is $\frac{\text{fs}}{\text{numel}(y) - 1}$.

```
f = linspace(-fs/2,fs/2,numel(y));
yfft = fft(y);
yfft_shitft = fftshift(yfft);
plot(f,abs(yfft_shitft));
title("Fourier Transform of y = cos(2πft)");
xlabel("Frequency(f)");
xlim([-10 10]);
```



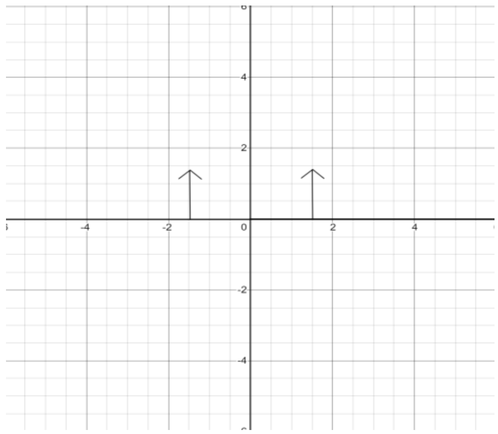

```
plot(f,angle(yfft_shitft))
```



Question 4 : Calculate the Fourier Transform of f(t) theoretically

$$F(w) = \int_{-\mathrm{inf}}^{+\mathrm{inf}} \cos(2\pi \mathrm{ft}) \, . \, e^{-\mathrm{iwt}} . \, \mathrm{dt} = \pi(\delta(w - 2\pi f) + \delta(w + 2\pi f))$$

the diagram of question 3 and 4 are similar.



Part 2 : Processing Sleep Patterns of New Born Babies

New born baby sleep cycle :

```
load EEG/data.mat
```

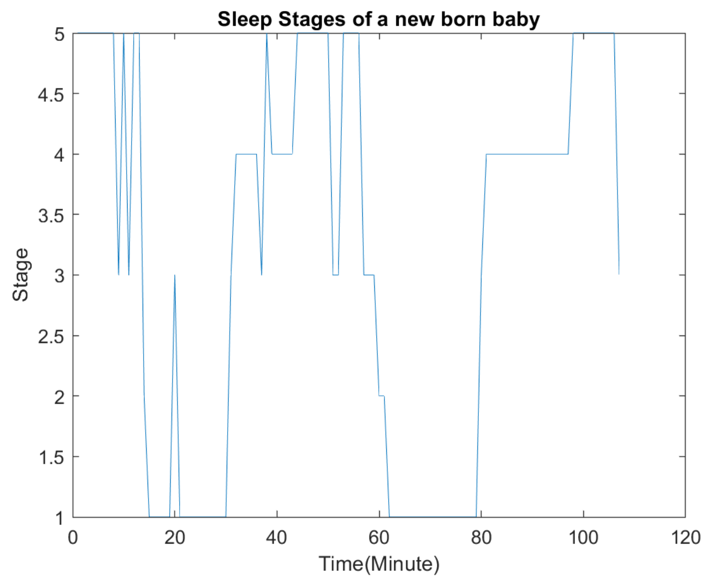
Assigning index to each sleep stage

The REM sleep stage is frequently called “active sleep” and NREM is called “quiet sleep.” During “active sleep,” or REM, a baby can be seen making small movements. The baby’s eyes move around (while closed), their limbs and fingers might twitch or jerk, their breathing might speed up, and they might move their mouths. During “quiet sleep,” or NREM, the baby is still and does not make these movements.(Source)

1. qt - Quiet sleep, trace alternant(NERM)
2. qh - Quiet sleep, high voltage(NREM)
3. tr - Transitional sleep
4. al - Active sleep, low voltag(REM)
5. ah - Active sleep, high voltage(REM)
6. aw - Awake

```
len = length(data);
data2 = zeros(1,len);
for i = 1:len
    if data(i) == "qt"
        data2(i) = 1;
    elseif data(i) == "qh"
        data2(i) = 2;
    elseif data(i) == "tr"
        data2(i) = 3;
    elseif data(i) == "al"
        data2(i) = 4;
    elseif data(i) == "ah"
        data2(i) = 5;
    elseif data(i) == "aw"
        data2(i) = 6;
    end
end

plot(data2);
title("Sleep Stages of a new born baby");
ylabel("Stage");
xlabel("Time(Minute)");
```



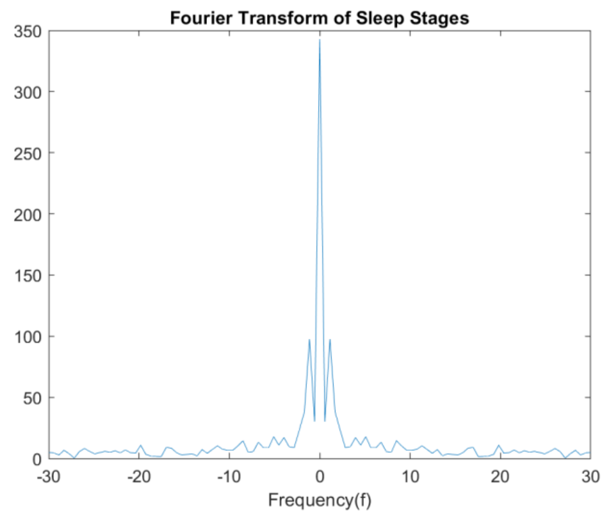
Newborn babies spend roughly 50% of their time asleep in REM. In the diagram above, roughly 50% od new born baby is in REM stage.

Fourier Transform of Sleep Stages

```
fft_sleep_stages = fft(data2);
```

Graph of Fourier Transform

```
fs = 60;
% Frequency Vector
f = linspace(-fs/2,fs/2,numel(data2));
fft_sleep_stages_shift = fftshift(fft_sleep_stages);
plot(f,abs(fft_sleep_stages_shift));
title("Fourier Transform of Sleep Stages");
xlabel("Frequency(f)");
```



Question : How many times baby’s sleep cycle is completed during an hour?

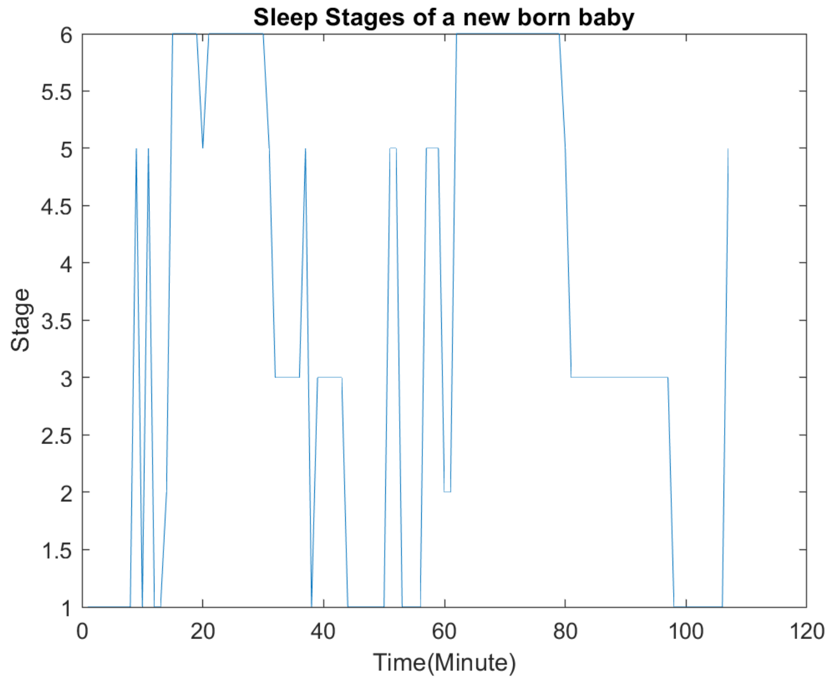
The baby’s sleep cycle is completed only 1 time during an hour.

Question : If we change order of indexes, is it still analysable?

If the consecutive sleep stages have different indexes, then this is not **analysable**.

```
len = length(data);
data3 = zeros(1,len);
for i = 1:len
    if data(i) == "qt"
        data3(i) = 6;
    elseif data(i) == "qh"
        data3(i) = 2;
    elseif data(i) == "tr"
        data3(i) = 5;
    elseif data(i) == "al"
        data3(i) = 3;
    elseif data(i) == "ah"
        data3(i) = 1;
    elseif data(i) == "aw"
        data3(i) = 4;
    end
end

plot(data3);
title("Sleep Stages of a new born baby");
ylabel("Stage");
xlabel("Time(Minute)");
```



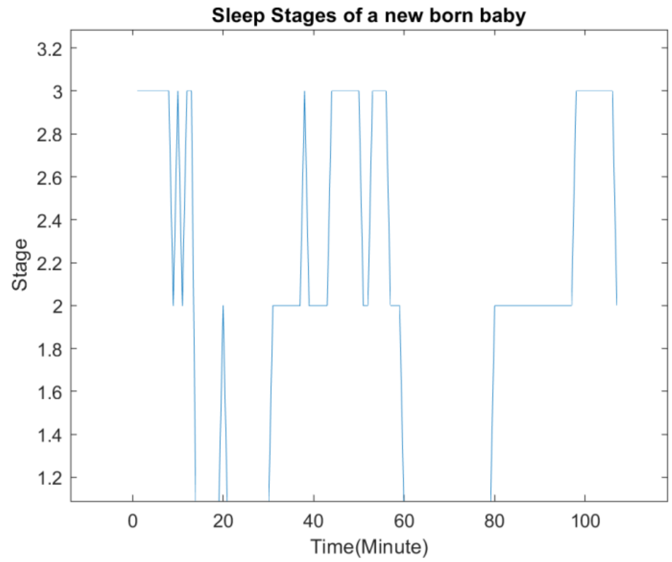
As you can see the diagram is kind of random.

Assigning different indexes to sleep stages

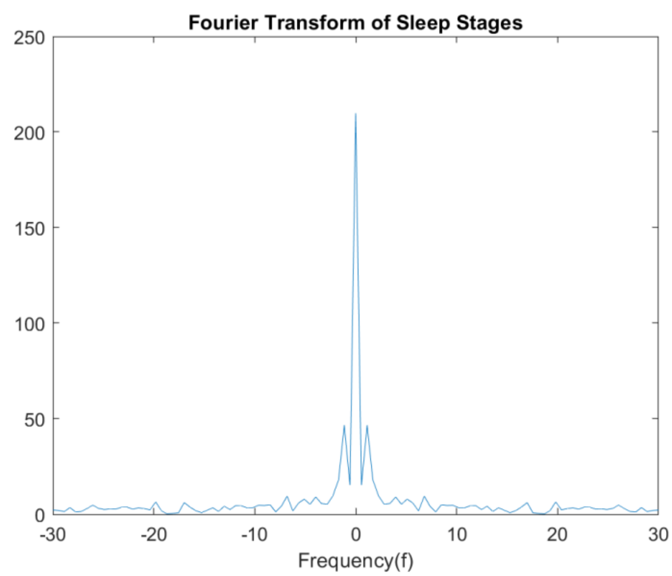
order of indexes : 1 1 2 2 3 3 -> qt qh tr al ah aw

```
for i = 1:len
    if data(i) == "qt" || data(i) == "qh"
        data2(i) = 1;
    elseif data(i) == "tr" || data(i) == "al"
        data2(i) = 2;
    elseif data(i) == "ah" || data(i) == "aw"
        data2(i) = 3;
    end
end

plot(data2);
title("Sleep Stages of a new born baby");
ylabel("Stage");
xlabel("Time(Minute)");
```




```
fft_sleep_stages = fft(data2);
% Frequency Vector
fft_sleep_stages_shift = fftshift(fft_sleep_stages);
plot(f,abs(fft_sleep_stages_shift));
title("Fourier Transform of Sleep Stages");
xlabel("Frequency(f)");
```



As you can see the results are **similar**.

Question : Why changing the indexes didn't affect the results?

Because the order of consecutive sleep didn't change. If we assign random indexes to each sleep stage, the result will not be analysable.

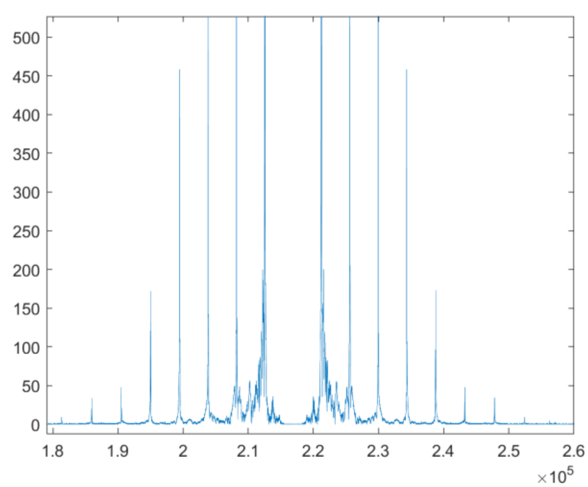
Part 3 : Processing sound waves using Fourier Transform

Reading piano notes from Data/Piano/*.mp3 :

```
cd C:\Users\saeed\Desktop\EngMath\
[y1,Fs1] = audioread('Data/Piano/piano_A.mp3');
[y2,Fs2] = audioread('Data/Piano/piano_A_sharp.mp3');
[y3,Fs3] = audioread('Data/Piano/piano_B.mp3');
[y4,Fs4] = audioread('Data/Piano/piano_C_sharp.mp3');
[y5,Fs5] = audioread('Data/Piano/piano_D.mp3');
[y6,Fs6] = audioread('Data/Piano/piano_D_sharp.mp3');
[y7,Fs7] = audioread('Data/Piano/piano_E.mp3');
[y8,Fs8] = audioread('Data/Piano/piano_F.mp3');
[y9,Fs9] = audioread('Data/Piano/piano_F_sharp.mp3');
[y10,Fs10] = audioread('Data/Piano/piano_G.mp3');
[y11,Fs11] = audioread('Data/Piano/piano_G_sharp.mp3');
[y12,Fs12] = audioread('Data/Piano/piano_middle_C.mp3');
```

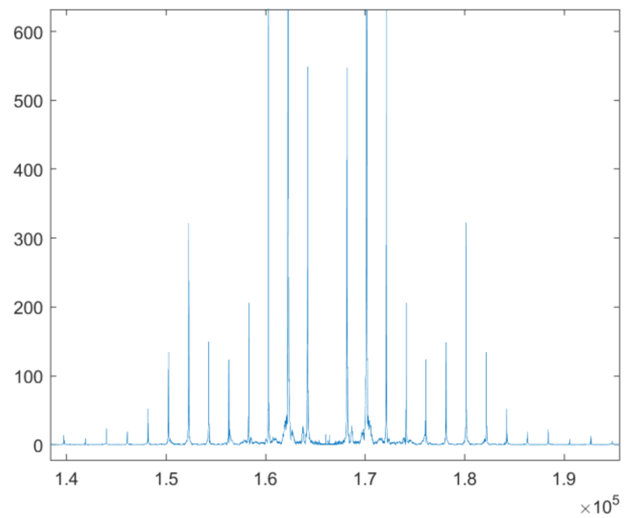
Fourier Transform of sound waves

```
y1_fft = fft(y1);
y1_fft_shift = fftshift(y1_fft);
plot(abs(y1_fft_shift));
xlim([179067 260004])
ylim([-12 527])
```




```
y12_fft = fft(y12);
y12_fft_shift = fftshift(y12_fft);
plot(abs(y12_fft_shift));
```

```
xlim([138360 195588])
ylim([-22 632])
```




```
sound(y1,Fs1);
sound(y12,Fs12);
```

Question : Explain the differences between these two notes