



به نام خدا



دانشگاه تهران
دانشکده‌ی مهندسی برق و کامپیوتر
پردازش سیگنال‌های زمان گسسته

گزارش پروژه سوم

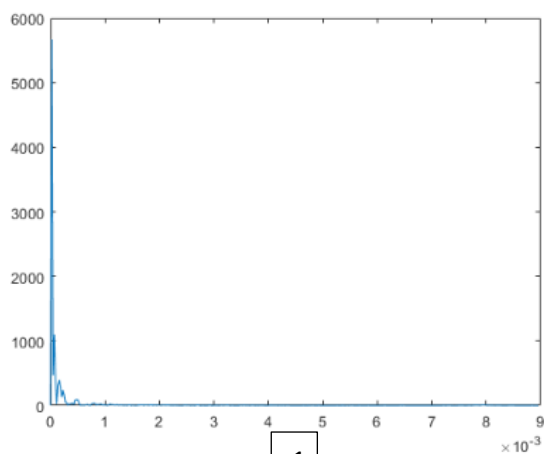
اشکان جعفری	نام و نام خانوادگی
810197483	شماره‌ی دانشجویی

سوال 1

الف) طبق خواسته سوال مطابق با فایل Q1.mlx تمامی نمودارها ترسیم شده‌اند.

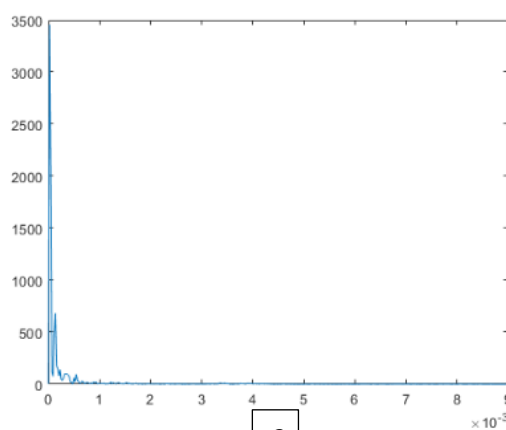
برای مثال یک نمونه از این نمودارهای cepstrum برای صدای مرد و زن برای یک مصوت خاص را اینجا می‌آوریم:

```
[y,Fs]=audioread("Male_voice\iii.mp3");
dt=1/Fs;
ninems = int16(0.009*Fs)-1;
t = 0:dt:0.009-dt;
c = abs(rceps(y));
plot(t,c(1:ninems));
```



1

```
[y,Fs]=audioread("Female_voice\iii.mp3");
dt=1/Fs;
ninems = int16(0.009*Fs)-1;
t = 0:dt:0.009-dt;
c = abs(rceps(y));
plot(t,c(1:ninems));
```



2

به خوبی براساس قله‌های متناوب بیشتر شکل یک مرد بودن جنس صدا پی برد و از طرفی سراسیابی در هر دو تصویر که بین ثانیه صفرم تا یک میلی ثانیه بعد از آن است نشان از تابع تبدیل است.

برای باقی صداها و تصاویر می‌توانید به فایل Q1.mlx مراجعه کنید.

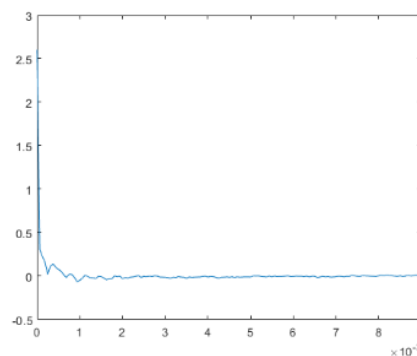
ب) چیزی که به ذهن بنده می‌رسد این است که زیر تر بودن صدای زنان نسبت به مردان باعث این می‌شود که فرکانس پایین بودن صدا در تعداد قله‌ها نمایان شود.

سوال 2

مطابق شکل 3، با پیدا کردن اولین قله بعد قله ناشی از نویز، و جدا سازی آن توسط یک فیلتر می‌توانیم به سیگنال اصلی دست پیدا کنیم:

```
[Echo, Fs] = audioread('echo_sound.wav');
dt=1/Fs;
ninems = int16(0.009*Fs);
t = 0:dt:0.009-dt;
c = rceps(Echo);
[pks, Max] = findpeaks(c,'Threshold',0.1);
plot(t,c(1:ninems));
%D = Delta;
D = Max(1)-1;

b=1;
a=[1 zeros(1, D-1) 0.8];
refirst_voice = filter(b, a, Echo);
audiowrite('refirst.wav', refirst_voice, Fs);
```



3

اما این فیلترینگ اساس ریاضی اش را میتوانیم در تبدیل Z ببینیم. مطابق محاسبات زیر، کافیهست جمله $\frac{1}{1+z^{-\Delta}\alpha}$ را از تبدیل Z صدای اکویافته برداریم.

$$\begin{aligned}
 y[n] &= x[n] + \alpha x[n-\Delta] \\
 \Rightarrow Y(z) &= X(z) + \alpha z^{-\Delta} X(z) \\
 &= \underline{X(z) (1 + \alpha z^{-\Delta})} \\
 \Rightarrow X(z) &= \frac{Y(z)}{(1 + \alpha z^{-\Delta})}
 \end{aligned}$$

سوال (3)

الف و ب) سیاست مورد نظر برای کم حجم سازی تصاویر این است که داده های جزئی را حذف و کلی ها را نگه داریم.

برای مثال اگر بخواهیم 5 درصد حجم را کم کنیم باید 5 درصد از ابعاد کم شود اما اینکه کجا کم شود مهم است که ترجیح ما داده سمت راست و پایین است.

برای اینکار یک ماتریس جدید تعریف میکنیم و براساس حجم مورد نیازمان با یک ضریب، درصدی از داده ها را منتقل میکنیم.

در شکل 4 به خوبی این روش مشهود است

پ) علت اینکه در حالت grayscale میتوان به هر خانه براساس شدت رنگ یک عدد اختصاص داد و اساسا حجم براساس حضور یا عدم حضور یک خانه است. اما در اشکال رنگی خودت رنگ هم یک علت برای حجم است.

```

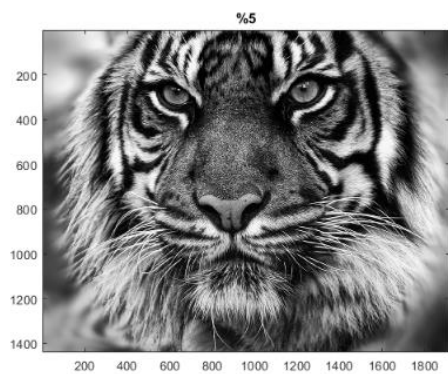
img = imread('tiger.jpg');
DCT_img = dct2(img);
[row , column] = size (DCT_img);
%%%%%%
fimg5P = zeros(row, column);
ROW=floor(0.4*row);
COLUMN=floor(0.4*column);
fimg5P(1:ROW, 1:COLUMN)= DCT_img(1:ROW, 1:COLUMN);
f5P = idct2(fimg5P);
imwrite(uint8(f5P), 'tiger5P.jpg', 'JPG');
%%%%%%
fimg15P = zeros(row, column);
ROW2=floor(0.3*row);
COLUMN2=floor(0.3*column);
fimg15P(1:ROW2, 1:COLUMN2) = DCT_img(1:ROW2, 1:COLUMN2);
f15P = idct2(fimg15P);
imwrite(uint8(f15P), 'tiger15P.jpg', 'JPG');

```

4

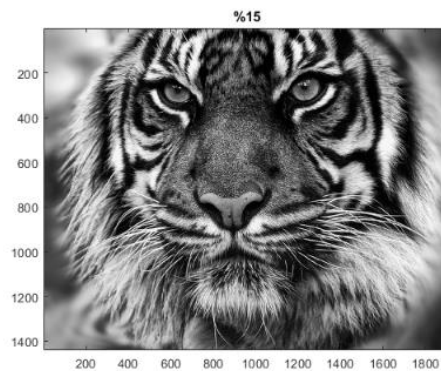
در شکل 5 و 6 می‌توانید کم حجم شده تصاویر را ببینید. (تصویر ها براساس درصد خواسته شده با نام های tiger5P و tiger15P ذخیره شده اند).

```
figure;  
image(fsp);  
colormap(gray(256));  
title('%5');
```



5

```
figure;  
image(f15P);  
colormap(gray(256));  
title('%15');
```



6