

گزارش کار پروژه نهایی

البرز محمودیان

قسمت 1

برای قسمت اول فقط باید با یک حلقه نت ها را از آرایه استخراج می‌کردم. بعد هم نت مربوط را در یک آرایه قرار می‌دهم و در نهایت هم آنها را با یکسری صفر به عنوان سکوت به یک آرایه کلی که کل موسیقی بود اضافه می‌کردم و با استفاده از `audiowrite` در یک فایل مینوشتم.

برای مشخص کردن نت موسیقی ابتدا چنین آرایه هایی ساختم:

```
frequencies = [ 16.352 , 17.324 , 18.354 , 19.445, 20.602, 21.827, 23.125, 24.500 , 25.957, 27.500 , 29.135 ,30.868 ];  
note_names = [ "C" ; "C#" ; "D"; "D#"; "E" ; "F" ; "F#" ; "G" ; "G#" ; "A" ; "A#" ; "B" ];
```

بعد هم با یک حلقه فور دیگر در بین این ها بررسی می‌کردم و وقتی پیدا می‌کرد فرکانس انرا محاسبه کردم. البته لازم نبود که تمام نت ها را داشته باشم. تنها اکتاو اول را ذخیره در کردم و برای اکتاو های بالاتر فرکانس اینها را در دو به توان اکتاو ضرب کردم.

```
timevec = 0:1/fs:time;  
chunk = [cos(2*pi*frequencies(j)*(2^(octave)) * timevec) , zeros(1,floor(0.025*fs))];
```

به صورت بالا

```
audiowrite("noteHarryPoter.wav",music,fs)
```

به این ترتیب هم نت ها را در یک فایل نوشته میشود.

نحوه شناسایی نت:

```
for i = 1:lengthofMusic  
    part = someNotes(i);  
    seperate = split(part);  
    type = seperate(1);  
    octave = str2num(seperate(2));  
    time = str2num(seperate(3));  
  
    for j= 1:12  
  
        if(note_names(j)==type)  
            break  
        end  
    end  
end
```

خروجی برنامه :

 noteHarryPoter.wav	1/8/2025 2:29 PM	WAV File	3,446 KB
--	------------------	----------	----------

قسمت 2

در فایل calculateFrequencies.m

برای پیدا کردن هارمونی های نت های اکتاو 5ام ابتدا صدای مربوط که نت ها را از یک شبیه ساز آنلاین دانلود کردم.

بعد هم آنها را در پوشه octave5 قرار دادم و با audioread مقادیر آنها را در متلب وارد کردم. سپس از تک تک آنها تبدیل فوریه گرفتم .

برای اینکه فرکانس ها و دامنه آنها را بدست بیاورم به صورت زیر عمل کردم:

- ابتدا بزرگ ترین مقدار موجود در تبدیل فوریه را به همراه فرکانس مرتبط با آن پیدا کردم.(با کمک max() میتوان هم مقدار و هم اندیس آنرا پیدا کرد. سپس از اندیس برای یافتن فرکانس مربوط استفاده میشود.)
- بعد چون میدانیم که فرکانس هارمونی های بعدی به صورت $2f_0, 3f_0, 4f_0, 5f_0, 6f_0$ است میتوان اندیس های مربوط به هارمونی ها را محاسبه کرد. البته چون ممکن است این مقادیر دقیق نباشند علاوه بر nf_0 چندین خانه قبل و بعد آنرا هم مورد نظر قرار دادم و بزرگ ترین مولفه را استخراج کردم.
- همه این موارد را در یک آرایه سه بعدی ریختم.(یک بعد نوع نت یعنی A B C ... ، یک بعد برای اینکه کدام هارمونی است یعنی هارمونی اول تا ششم. و یک بعد دیگر هم برای اینکه فرکانس را میخواهیم یا دامنه را)

سپس بقیه کد خیلی شبیه به قسمت قبل است و تنها میبایست بجای یک فرکانس خالص ، با یک حلقه فور این شش مولفه را برای هر نت استفاده کرد.


حلقه ای که هارمونی ها را میسازد و با هم جمع میکند:

```
for k = 1:6
    theNote = theNote + harmonies(j,k,2)*cos(2*pi*harmonies(j,k,1)*(2^(octave - 5))*timevec);
end
```

برای عبارت دمپ شونده هم میتوانستیم از یک نمایی استفاده کنم بنابراین هر نت تولید شده را در یک نمایی میرا با ثابت زمانی 0.04 ثانیه ضرب کردم

برای خود موسیقی هم تعدادی نت را در یک آرایه مانند قسمت قبل قرار دادم .

خروجی برنامه:

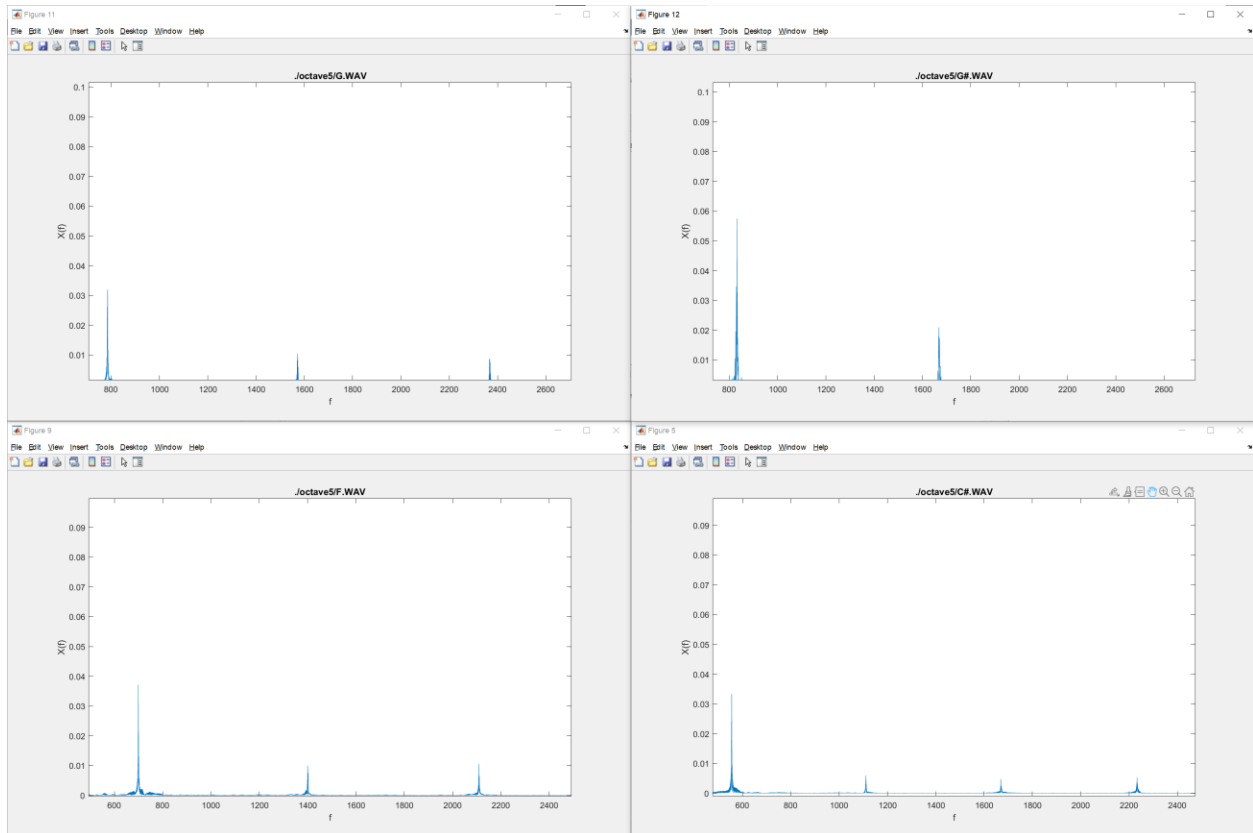
 noteOptimized.wav

1/8/2025 7:31 PM

WAV File

345 KB

کدی که نوشتم به صورت زیر تغییر دادم تا برای تبدیل فوریه هر نت یک نمودار میکشد در اینجا چند نمونه را نشان میدهم



البته چون کشیدن این نمودار ها زمان میبرد برای اجرای سریع تر میتوان این قسمت را حذف کرد و فقط در صورت نیاز آنرا کشید بنابراین من هم آن قسمت ها را کامنت میکنم تا سرعت اجرا پایین نیاید. در صورت لزوم میتوانید آن قسمت ها را از کامنت بودن خارج کنید و تمام شکل موج ها در حوزه فرکانس ببینید.

ضرایب بدست آمده هم در یک فایل اکسل قرار دادم و در همین پوشه قرار دارد. البته مقدار نمایش داده شده در واقع دقت بالاتری دارد دلیل آن هم این است که هنگامی که اینها را از متلب کپی میکردم متلب تنها با سه رقم اعشار اینها را نمایش میداد.

برای قسمت امتیازی تنها فایل emtiaz.m وجود دارد که تمام کد ها در آن قرار دارد.

این فایل دارای دو بخش است. بخش اول فایل ساخته شده در مساله اول را وارد میکند و یک آرایه مشابه آرایه داده شده که نت ها و مدت زمان آنرا در خود دارد تولید میکند. بخش دوم هم دقیقاً کد مساله دوم است که باید آرایه تولید شده را به صدای پیانو تبدیل کند. این قسمت را کامل از سوال دوم کپی کردم.

نحوه تبدیل صدای موسیقی به نت ها به صورت زیر است.

اگر سکوت در حال پخش باشد انقدر جلو میرود تا به جایی که صدا وجود دارد برسد و بعد از آن به اندازه 0.1 ثانیه از صوت را جدا میکند و در آرایه chunk قرار میگیرد.

سپس تبدیل فوریه این آرایه محاسبه میشود از تبدیل فوریه بدست آمده استفاده میشود تا بزرگ ترین مولفه فرکانسی پیدا شود:

```
while i<length
    j =0;
    while(signal(i+j)==0 && i+j<length-1)
        j=j+1;

    end
    if(j>1000)
        notes_f = [notes_f , 0];
    end
    i = i +j;
    if(i+0.1*fs>length)
        break
    end
    chunk = signal(i:i+0.1*fs);
    N = 0.1*fs +1;
    Y = abs(fft(chunk))/N;
    frequencies = (0:N-1)*(fs/N);
    [amplitude , baseFrequencyIndex] = max(Y);
    baseFrequency = frequencies(baseFrequencyIndex);
    notes_f = [notes_f , baseFrequency];
    i = 1+i+0.1*fs;
end
```

آرایه note_f فرکانس هم نت را ذخیره میکند(هر 0.1 ثانیه را جدا میکند) و اگر هم سکوت داشتیم یک صفر درج میکند.

حالا برای اینکه از روی فرکانس بتوانیم اکتاو و نام نت را بفهمیم به این صورت عمل شده که در یک حلقه ابتدا این فرکانس را با فرکانس های اکتاو صفر مقایسه میکند و اگر اختلاف کمتر از یک هرتز بود(چون فرکانس محاسبه شده خیلی دقیق نیست) آن نت را با اکتاو صفر در نظر میگیرد. اگر هیچ کدام نبودند یعنی این نت در اکتاو صفر نیست. بنابراین فرکانس را نصف میکند و اکتاو را

یکی افزایش میدهد. اگر با یکی از نت ها مشابه بود آن نت را به عنوان نت این chunk در نظر میگیرد دوباره اگر نبود اکتاو را یکی افزایش داده و فرکانس chunk را نصف میکند. به همین ترتیب وقتی از حلقه خارج شود فرکانس آن نت را فهمیده.

```
while(abs(note_freq(numberOfNote)-notes_f(x))>.5)
    if(notes_f(x)==0)
        break
    end
    numberOfNote = numberOfNote +1;
    if(numberOfNote ==13)
        numberOfNote = 1;
        octave = octave +1;
        notes_f(x) = notes_f(x)/2;
    end
end
```

برای همه نت ها همین کار را انجام میدهد. وقتی نت تکرار میشود یعنی مدت زمان نواختن نت باید 0.1 بیشتر شود. وقتی هم که به فرکانس صفر میرسد یعنی نواختن نت تمام شده و مدت زمان پخش آن و نام نت را به همان فرمت خواسته شده در آرایه دیگری ذخیره میکند.

```
while(x<endOf(2))
    numberOfNote = 1;
    octave = 0;
    while(abs(note_freq(numberOfNote)-notes_f(x))>.5)
        if(notes_f(x)==0)
            break
        end
        numberOfNote = numberOfNote +1;
        if(numberOfNote ==13)
            numberOfNote = 1;
            octave = octave +1;
            notes_f(x) = notes_f(x)/2;
        end
    end
    if(notes_f(x)~=0)
        duration = duration+0.1;
        prevNote = numberOfNote;
        prevoct = octave;
    else
        outputStr = [outputStr ,note_names(prevNote)+" "+string(prevoct)+" "+string(duration)];
        duration =0;
    end

    x = x+1;
end
display(outputStr)
```

نتیجه:

```
Command Window

outputStr =
1x60 string array
Columns 1 through 11
    "B 4 0.3"    "E 5 0.6"    "G 5 0.2"    "F# 5 0.3"    "E 5 0.6"    "B 5 0.4"    "A 5 0.8"    "F# 5 0.8"    "E 5 0.6"    "G 5 0.2"    "F# 5 0.3"
Columns 12 through 22
    "D# 5 0.7"    "F 5 0.4"    "B 4 1.6"    "B 4 0.3"    "E 5 0.6"    "G 5 0.2"    "F# 5 0.3"    "E 5 0.6"    "B 5 0.4"    "D 6 0.6"    "C# 6 0.3"
Columns 23 through 33
    "C 6 0.6"    "G# 5 0.3"    "C 5 0.5"    "B 5 0.2"    "A# 5 0.3"    "A# 4 0.6"    "G 5 0.3"    "E 5 1.6"    "G 5 0.3"    "B 5 0.6"    "G 5 0.3"
Columns 34 through 44
    "B 5 0.6"    "G 5 0.3"    "C 6 0.6"    "B 5 0.3"    "A# 5 0.6"    "F# 5 0.3"    "G 5 0.5"    "B 5 0.2"    "A# 5 0.3"    "A# 4 0.6"    "B 4 0.4"
Columns 45 through 55
    "B 5 1.6"    "G 5 0.3"    "B 5 0.7"    "G 5 0.3"    "B 5 0.7"    "G 5 0.3"    "D 6 0.7"    "C# 6 0.3"    "C 6 0.8"    "G# 5 0.3"    "C 6 0.6"
Columns 56 through 60
    "B 5 0.2"    "A# 5 0.3"    "A# 4 0.6"    "G 5 0.4"    "E 5 1"

fx >>
```

همین آرایه در بخش دوم کد استفاده میشود تا فایل newOutputFromListening.wav ساخته شود.

```
audiowrite("newOutputFromListening.wav",music,fs);
```



newOutputFromListening.wav

که مشابه همان فایل ورودی است.