

## تمرین امتیازی

هدف این تمرین پیدا کردن فرکانس pitch در فریم های مختلف با استفاده از روش autocorrelation می باشد.

چون تمام عملیات را بر روی دو فایل صوتی f.wav و a.wav با اعمال دو پنجره ی مختلف انجام می دهیم، لازم است برای سادگی کار تابعی با نام func تعریف کنیم که از تکرار کد جلوگیری شود.

```
13 % baraye inke chand bar amaliat zir ra anjam midahim yek function misazim
14 function [] = func(audio_str, is_rectangle)
```

برای اجرای عملیات لازم پنجره ها و فایل های مختلف را به این تابع دهیم:

```
6 - func("a.wav", 1); % file a , rectangle window
7 - func("a.wav", 0); % file a , hamming window
8
9 - func("f.wav", 1); % file f , rectangle window
10 - func("f.wav", 0); % file f , hamming window
```

در این تابع ابتدا فایل صوتی را به وسیله ی audioread میخوانیم و مقادیر y و فرکانس نمونه برداری را ذخیره می کنیم. سپس بر اساس عدد وارد شده به عنوان پارامتر پنجره، نوع آن را تشخیص می دهیم.

پس از آن نام فایل، نوع پنجره، فرکانس نمونه برداری، تعداد نمونه ها و فریم شیفت را بدست آورده و نمایش می دهیم.

```
23 - fprintf('~ %s - %s\n', audio_str, win_name);
24 % Fs sampling frequency
25 % Fs = 16KHz
26 - fprintf('Sampling Frequency (Fs): %d\n', Fs);
27
28 % N = 16,000 * 0.025 = 400 samples
29 - N = 0.025 * Fs;
30 - fprintf('Number of Samples (N): %d\n', N);
31
32 % Frame Shift = 40/100 * 400 = 160
33 - M = N * 40/100;
34 - fprintf('Frame Shift (M): %d\n', M);
35 - fprintf('\n\n');
36
37
38 - num_frames = round((length(y)- N)/M + 1) - 1; % tedad kole frame hara bedast miavarim
```

در این قسمت برای هر فریم میانگین ZCR و انرژی فریم را محاسبه می کنیم.

به گونه ای که ابتدا فریم را از  $\lambda$  جدا می کنیم و اسم آن را  $s$  می نامیم. سپس پنجره مورد نظر را بر روی  $s$  اعمال می کنیم و میانگین فریم را صفر می کنیم. پس از آن انرژی و ZCR آن فریم را از روابط داده شده بدست می آوریم.

• انرژی یک فریم:

$$E = \frac{1}{N} \sum_{n=0}^{N-1} S(n) * S(n)$$

• نرخ عبور از صفر:

✓ تعداد دفعات عبور از صفر در فریمی که دارای  $N$  نمونه است:

$$ZCR = \frac{1}{N} \sum_{n=1}^{N-1} \frac{|sgn[S(n)] - sgn[S(n-1)]|}{2}$$

```

53 - for i=1:num_frames % braye har frame amal zir ra anjam midahim
54 -     s = y((i-1)*M + 1 : (i-1)*M + N); %i th frame ra tashkhis midahim
55 -     s = s.*win; %window ra emal mikonim
56 -     s = s - mean(s); % meghdar signal ra az miangin (dc) kam mikonim
57 -
58 -     % Energy ra be dast miavarim bana bar ravabet
59 -     E(i) = sum(s.*s)/N;
60 -
61 -     %ZCR ra ham be komake rabete be sorate vectorized be dast miavarim
62 -     diff = sign(s(2:num_frames)) - sign(s(1:num_frames-1));
63 -     ZCR(i) = sum(abs(diff))/N;
64 - end

```

در ادامه برای هر فریم autocorrelation را بدست می آوریم.

مانند بخش قبل  $s$  را انتخاب، پنجره را اعمال و میانگین را صفر می کنیم.

```

66 - %autocorr ra dar har frame be dast miavarim
67 - for i=1:num_frames
68 -     s = y((i-1)*M + 1 : (i-1)*M + N); %i th frame ra tashkhis midahim
69 -     s = s.*win; %window ra emal mikonim
70 -     s = s - mean(s); % meghdar signal ra az miangin (dc) kam mikonim

```

برای بدست آوردن فرکانس pitch و عملکرد بهتر، فقط مقدار پیچ و اتوکورولیشن فریم هایی که مقدار انرژی نسبتا بالایی دارند، را انتخاب می کنیم و این فریم ها را اینگونه انتخاب می کنیم که اگر مقدار انرژی آنها از یک چهارم بالاترین انرژی بیشتر بود انتخاب می شود و مقدار اتوکورولیشن آن فریم را به ازای  $\eta$  های مختلف بدست می آوریم.

```

72 - if E(i) > max(E)/4 % meghdar pitch ra dar frame haE ke energy bala daran hesab mikonim
73 -     corrs = zeros(1,N-1); % meghdar correlation hara inja zakhire mikonim

```

در غیر این صورت مقدار pitch آن فریم را برابر NaN قرار می دهیم.

```
97 - else % frame haE ke energy paEni daran megdhar pitch ra NaN migozarim ke f_pitch daghigh be dast biayad
98 -     pitch(i) = NaN;
99 - end
```

برای هر اتا از 0 تا N-1 (یک کمتر از اندازه ی فریم) مقدار اتوکورولیشن را محاسبه می کنیم و مقدار آن را در اندیسی برابر با eta+1 در آرایه ی corrs ذخیره می کنیم.

$$r(\eta) = \frac{1}{N} \sum_{n=0}^{N-1} S(n) * S(n - \eta)$$

```
73 - corrs = zeros(1,N-1); % megdhar correlation hara inja zakhire mikonim
74 - for eta=0:N-1 % baraye eta az 0 ta N-1 bar asase ravabet moadele ra hesab mikonim
75 -     res=0;
76 -     for n=1:N
77 -         if n - eta <= 0 % agar n-eta dar frame mojud nabood az tahe frame dar nazar migirim
78 -             % tahe frame ra be khudash motasel
79 -             % mikonim va maghadir ra be in sorat
80 -             % hesab mikonim
81 -             res = res + s(n)*s(n-eta+N);
82 -         else % dar halat adi ham s(n)*s(n-eta) ra hesab mikonim
83 -             res = res + s(n)*s(n-eta);
84 -         end
85 -     end
86 -     corrs(eta+1)=res/N; % dar nahayat bar tedad (N) taghsim mikonim
87 - end
```

اگر مقدار  $n - \eta$  کمتر از صفر بود به دلیل اینکه اندیس های مربوط به فریم حداقل صفر هستند، فریم را به گونه ای در نظر میگیریم که ابتدای آن به انتهای آن وصل شده باشد بنابراین مقدار  $n - \eta$  در آن حالت برابر  $n - \eta + N$  می باشد.

```
if n - eta <= 0 % agar n-eta dar frame mojud nabood az tahe frame dar nazar migirim
    % tahe frame ra be khudash motasel
    % mikonim va maghadir ra be in sorat
    % hesab mikonim
    res = res + s(n)*s(n-eta+N);
else % dar halat adi ham s(n)*s(n-eta) ra hesab mikonim
    res = res + s(n)*s(n-eta);
end
```

حال مقدار اتوکورولیشن در مقادیر مختلف اتا بدست آمده است، لازم است که  $I_{pos}$  را از طریق پیدا کردن peak مقادیر بدست آوریم. به همین دلیل از تابع findpeaks برای پیدا کردن peak در نصف مقادیر (مقادیر قرینه هستند نصف آنها برای بررسی کافی می باشد) corrs استفاده می کنیم. بیشترین مقدار peak ها را به کمک تابع max بدست می آوریم، اما ما با مقدار آن کاری نداریم بنابر این از مقدار argmax که اندیس بیشترین مقدار در بین peak ها می باشد، استفاده می کنیم.

تابع findpeaks علاوه بر peak ها یک نگاشت از اندیس آنها در آرایه ی pks به اندیس حقیقی آنها در آرایه ی corrs به ما می دهد. بنابراین برای بدست آوردن اندیس واقعی آن (مقدار اتای که اتوکورولیشن آن بیشترین مقدار شده است) از locs استفاده می کنیم و argmax را به آن می دهیم و مقدار اتای با بیشترین مقدار اتوکورولیشن را در index ذخیره می کنیم.

```
89 - [pks,locs]=findpeaks(corrs(1:N/2)); % dar inja peak haye maghadir correlation bedast amade ra hesab mikonim
90 - [arg, argmax] = max(pks); % bishtarin megdhar nesfe baze ra bedast miavarim (maghadir motagharen hastan)
91 - index = locs(argmax); % index max megdhar ra bedast miavarim ke I_pos mibashad
92 -
```

```
89 - [pks,locs]=findpeaks(corrs(1:N/2));
90 - [arg, argmax] = max(pks); % bishtari
91 - index = locs(argmax); % index max me
92 -
```

مقدار index همان  $I_{pos}$  ای می باشد که به دنبال آن بودیم.  
بنابراین برای بدست آوردن فرکانس پیچ در آن فریم از  $I_{pos}$  (index) و رابطه ی زیر استفاده می کنیم.

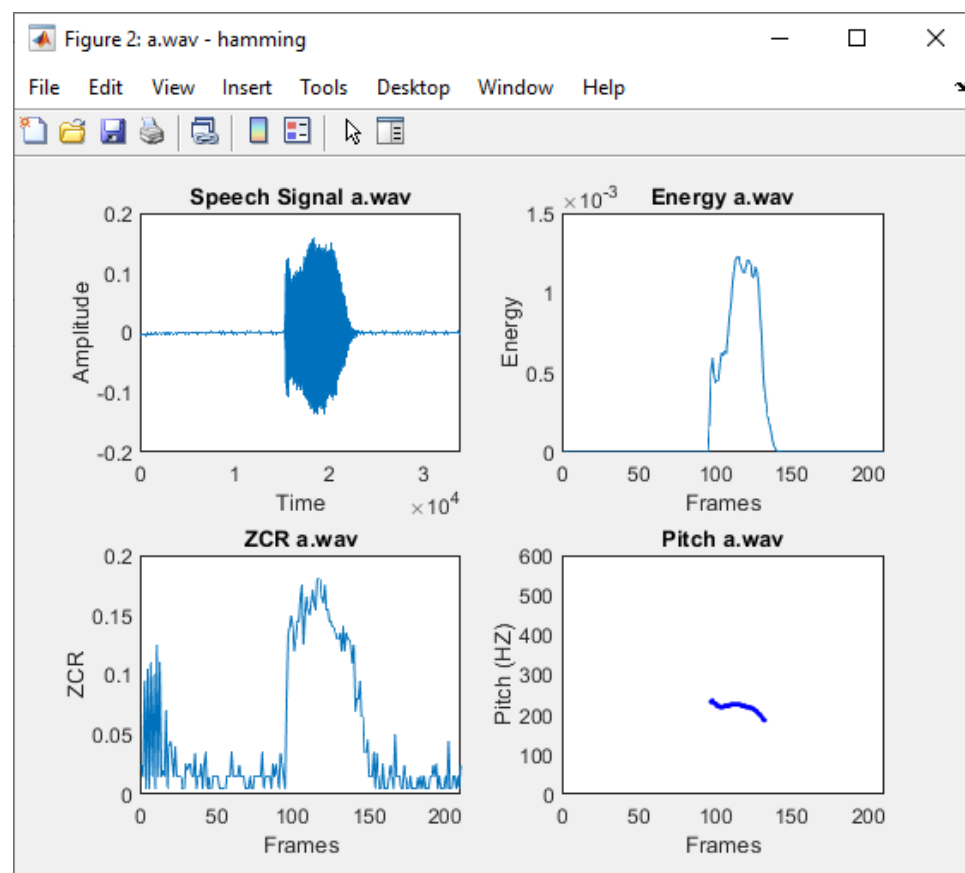
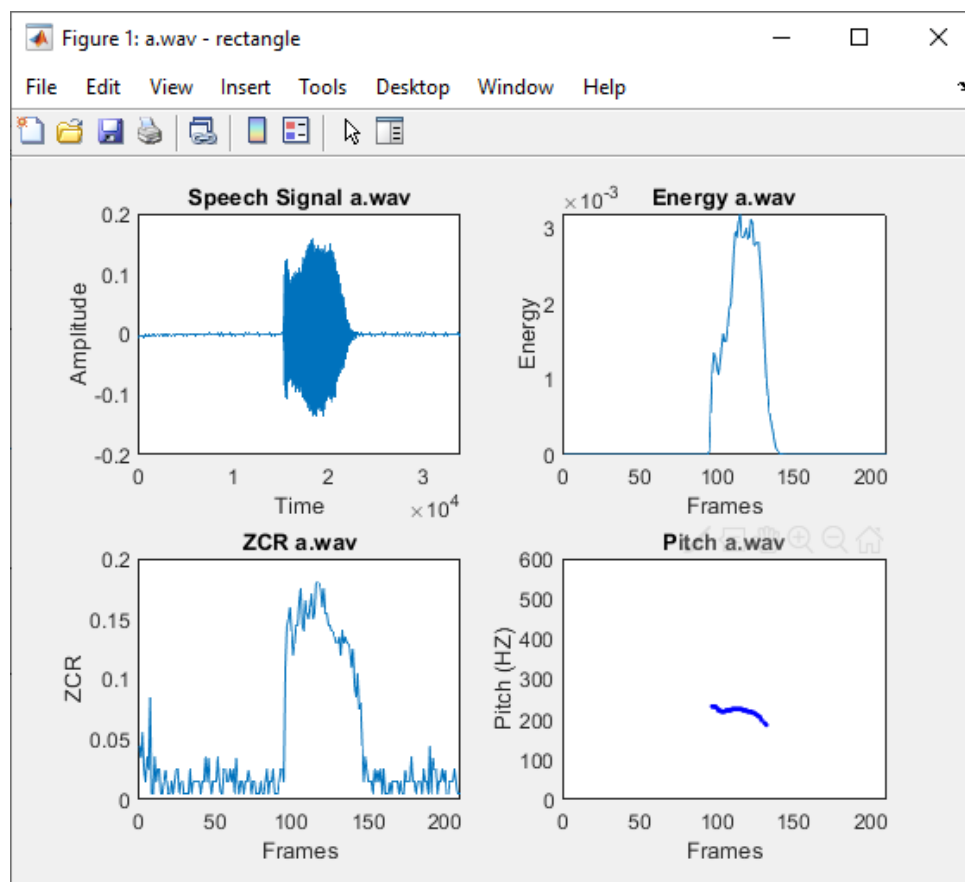
- فاصله اولین قله یعنی  $I_{pos}$  ، مقدار پریود گام خواهد بود.
- فرکانس گام برابر است با:

$$\bullet F_{pitch} = \frac{F_s}{I_{pos}}$$

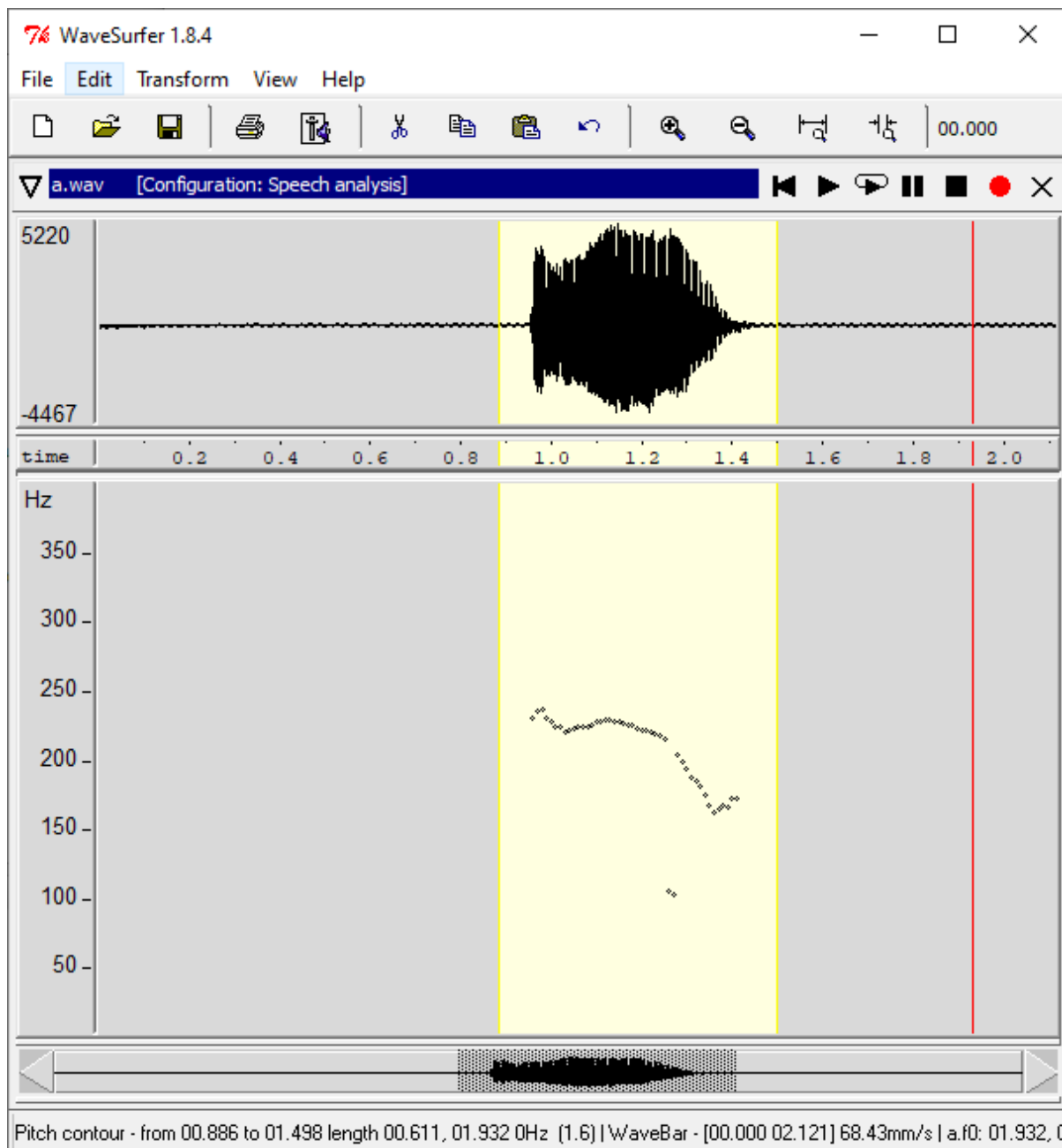
```
if ~isempty(index) % dar baazi frame ha megdhar max peak vojod nadarad
    % dar sorate vojod pitch ra az tarigh rabete hesab mikonim
    pitch(i) = Fs/index;
else
    pitch(i) = NaN;
end
```

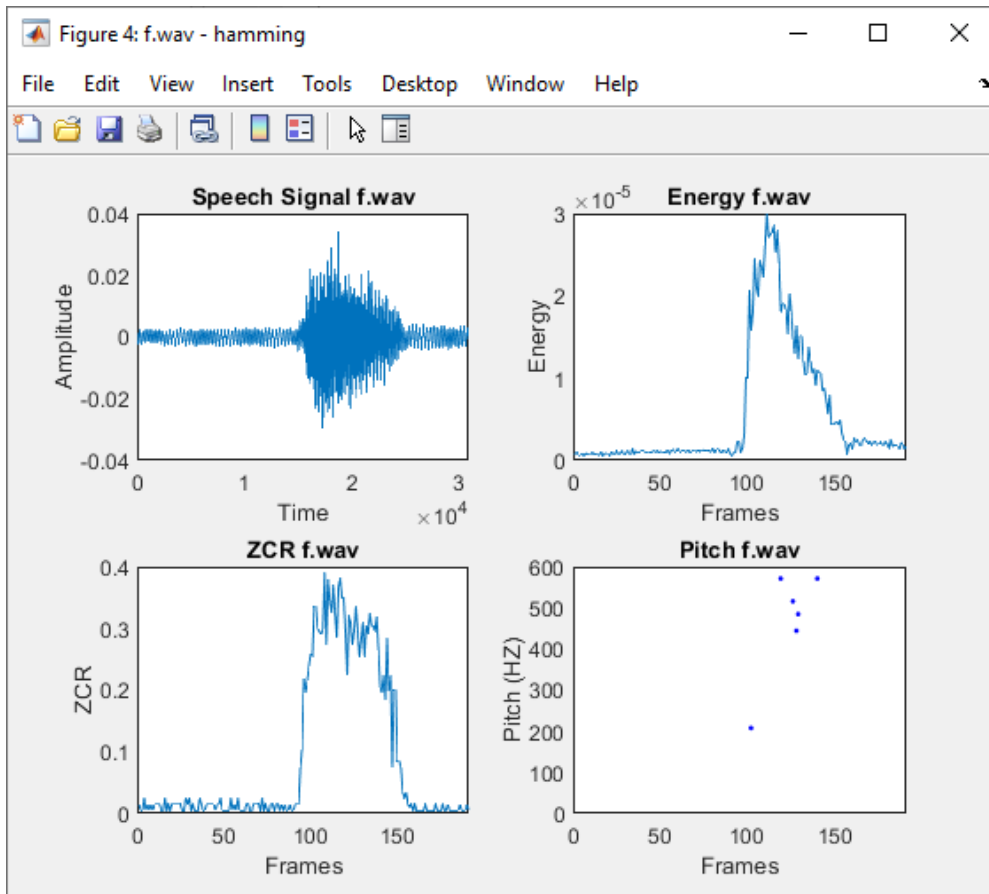
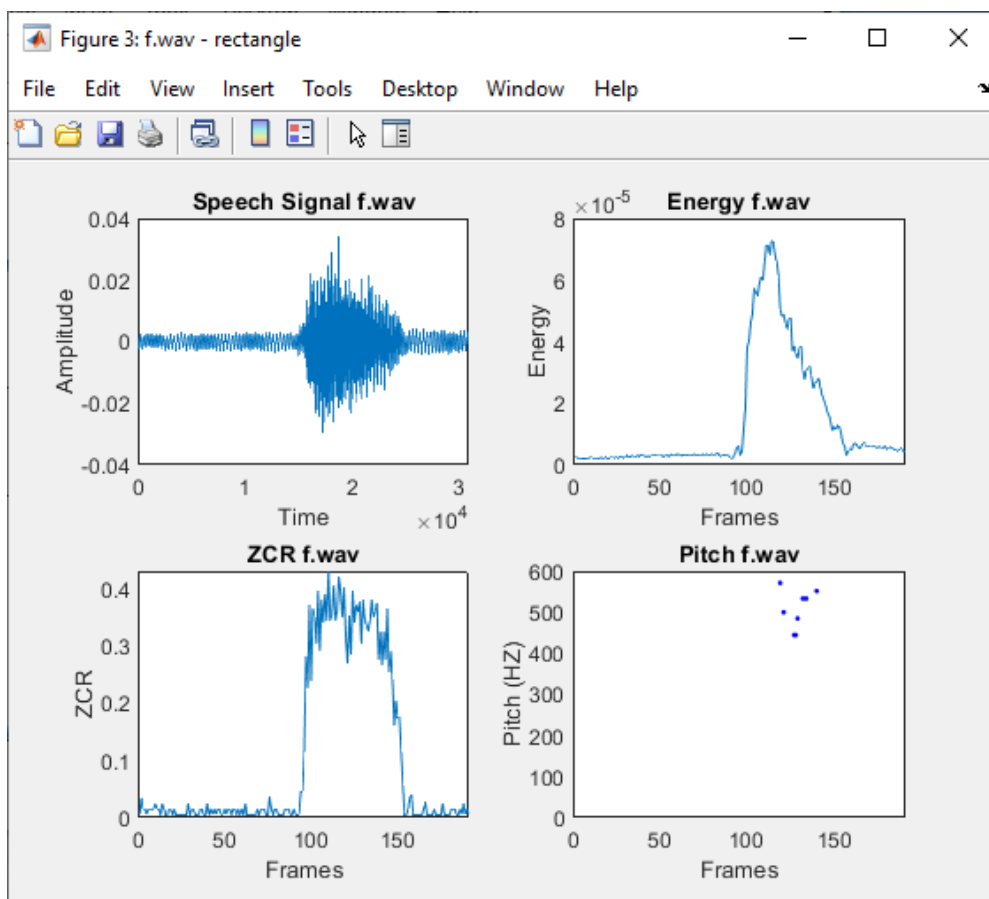
بعضی مواقع امکان این وجود دارد که اندیسی برای peak بدست نیاید (index مقداری نداشته باشد)  
بنابراین از مقدار پیچ در آن فریم چشم پوشی می کنیم و مقدار آن را NaN می گذاریم.

در ادامه نمودار های مختلف را برای هر حالت نمایش می دهیم.



در حالتی که فایل صوتی مربوط به /a/ را بررسی می کنیم چون واکدار می باشد فرکانس پیچ را می توان بدست آورد و در مقایسه با مقدار واقعی آن مشاهده می شود که به صورت درست محاسبه شده است.





در مورد فایل مربوط به /f/ چون بی واک می باشد. فرکانس پیچ را نمیتوان بدست آورد. همانگونه که مشاهده می شود مقادیری نامنظم و در چند فریم محدود نمایش داده شده است. در حالی که با اعمال فیلتر بر روی ZCR این فایل می توانستیم تشخیص دهیم که مربوط به یک بی واک می باشد و بخش اتوکورولیشن چشم پوشی کرد.

اما برای مشاهده نتیجه ی محاسبه ی اتوکورولیشن برای یک بی واک فیلتری قرار داده نشده است.