LAB 2 - TASK 4 (1 point possible)

In this task, you will learn how to approximate the signal within one frame using only the most significant frequency components. You will observe how the quality of the approximation improves as the number of frequency components used increases. However, a reasonable approximation can be obtained using a fairly small number of frequency components.

```
% load a speech signal
[x,Fs] = audioread('test.wav');

% define the frame length
frame_length = 256;

% compute power of each frame
frame_power = get_frame_power(x,frame_length);
% plot original waveform and power in each frame
figure(1);
plot_speech_power(x,frame_power,frame_length);

% find index of frame with maximum power (fnum)
[mxpow,fnum] = max(frame_power);

% extract frame fnum
```

Unanswered

Figure 1

tutorials

- MATLAB Sandbox

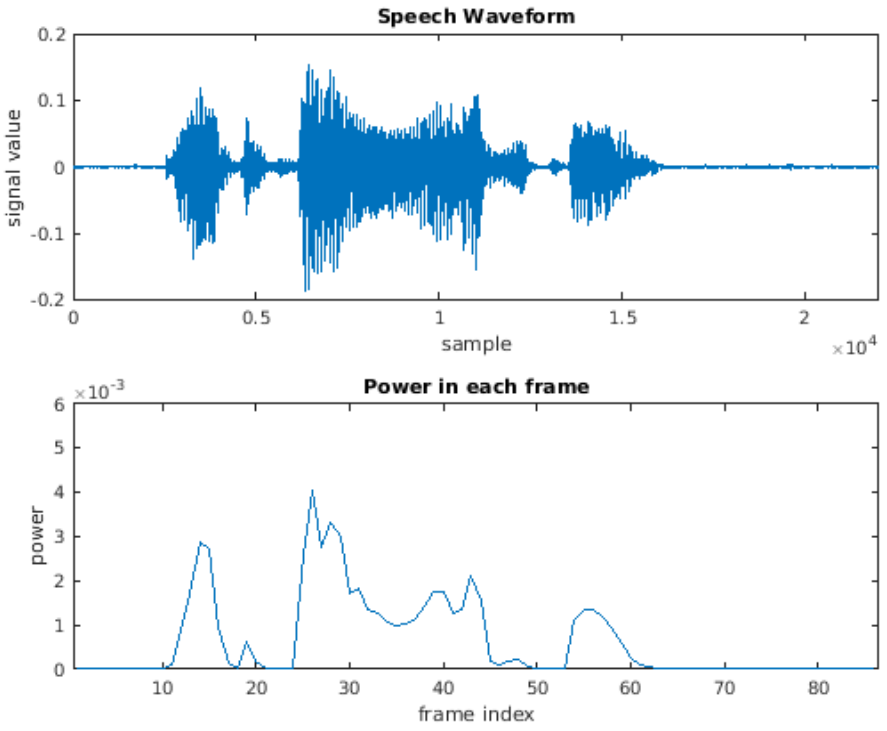


Figure 2

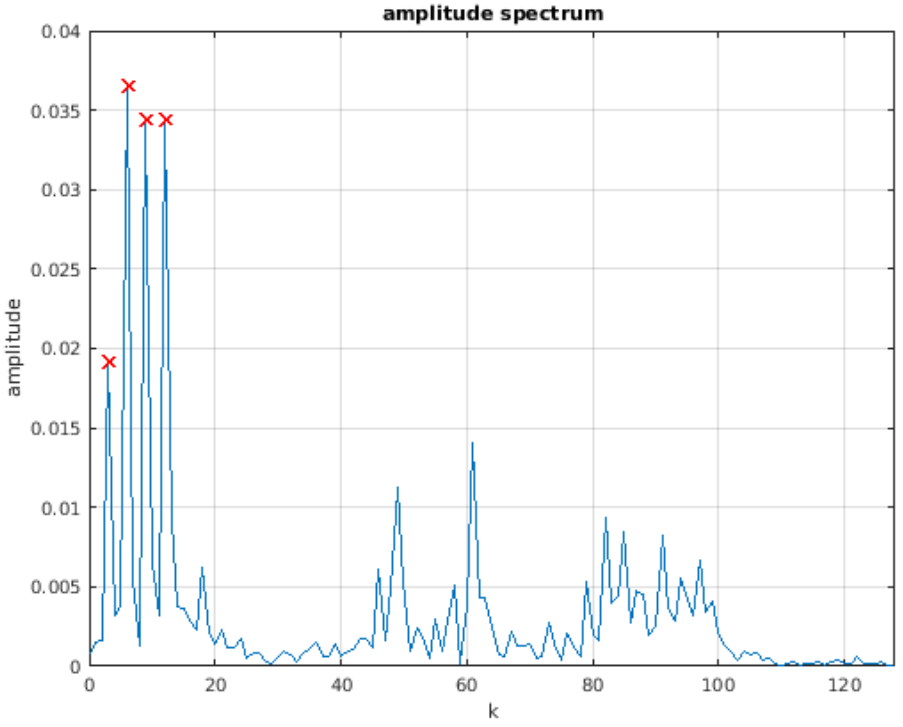
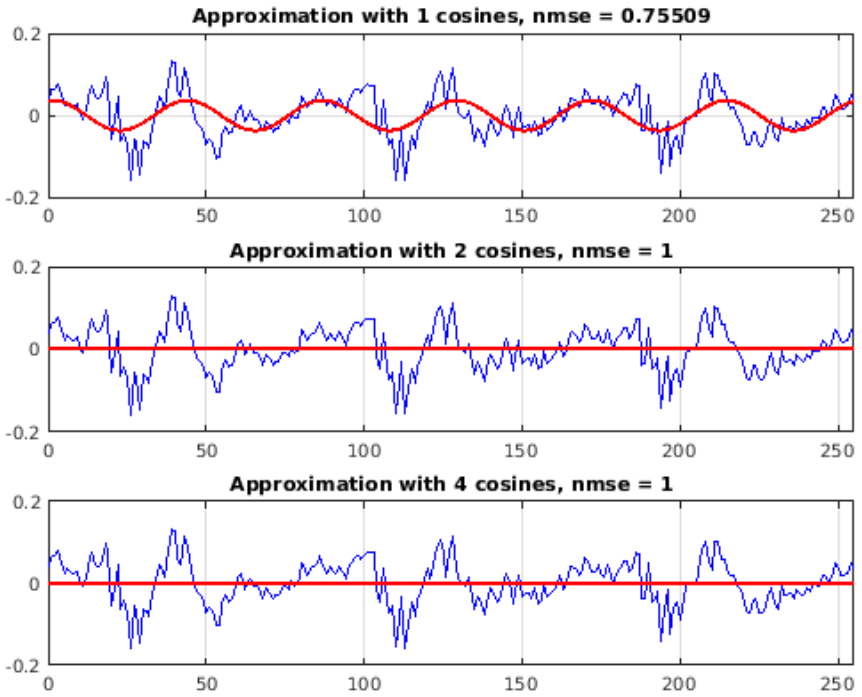


Figure 3



Run Code

INSTRUCTIONS

The initial MATLAB code in the above window analyzes the amplitude spectrum and then try to approximate the selected frame using a faulty implementation of the function **approximate_frame** that you used in Task 1. In this task, you will need to correct this function.

If you click the **Run Code** button, the initial code will generate three figures. Figure 1 shows the waveform of the entire speech signal, **x**, and the power of each frame. Figure 2 shows the amplitude spectrum of the frame **fnum** and the location of the first four most important components. The code automatically chooses **fnum** so that the frame with the highest power is shown. Figure 3 contains three subplots, each one showing a different approximation of that frame obtained by using 1, 2 and 4 frequency components. Due to some mistakes in the code, the approximations are not very good. Indeed, you can observe that the normalized mean squared error, **nmse**, between the original frame and its

approximation using 2 and 4 components is bigger that the one obtained using just one component. Your job is to correct the code to generate the correct approximations.

The initial code is similar to the code in task 1. It loads a speech signal and obtains the index of the frame with the highest power. Then, the frame with the highest power is extracted using the function **extract_frame** and function **get_sig_fc** provides the frequency value (**ksig**), the amplitude (**Asig**) and the phase (**phisig**) of the 4 most important frequency components.

The next part of the code should compute approximations to the original frame with one, two and four sinusoidal functions. These approximated signals are to be stored inside the matrix **fa**, where each row of the matrix contains a different approximation. The first row should contain the approximation with the one most significant cosine. The second and third rows should contain the approximations with the two and four most significant cosines. However, due to mistakes in the code, the reconstructed signals with two and four components (red lines in the plots) do not approximate the original signal very well (blue line). A quantitative evaluation of the approximation error is provided by the **nmse**, which should decrease when the number of components increases. If you correct the code, then the red lines will approximate the blue line and the **nmse** will decrease. Your task is to revise the code between the lines

% % % % Revise the following code % % % %

and

% % % % Do not change the code below % % % %

so that **fa(2,:)** and **fa(3,:)** will better approximate the original frame. The remainder of the code is used to generate the plots.

Please, revise the code without using the function **approximate_frame** and without changing the varibles **x**, **Fs**, **frame_length**, **nsig**, **numcom** and **fnum**. Once you are done with the code, you can change **fnum** to see how well different

frames in the signal can be approximated. However, before you submit your code for checking, be sure that **fnum** is set to the frame with highest power.