ECE 2312

Dialing for DSP

Mini-Project

Bruce Huynh

February 25th, 2021

Homework #7

Objective

The objective of this assignment is to be able to take the frequency inputs of a land line phone and to be able to interpret it through the Discrete Fourier Transform.

Method

In order to decode the signal, we need to invoke another MATLAB function. Luckily, this one is provided for us and makes this process a lot easier. Using the "*hw7dial()*" function, we can input any set of inputs that contain one or more of these characters: *1, 2, 3, 4, 5, 6, 7, 8, 9, 0, *, #*. (Important! These characters have to be input as a string in one line without spaces). In addition to the string, we also need to provide a sampling frequency and optionally, some noise. Once that is done, we will get a single vector that contains the information of that string.

Next, we will create a new function to perform the transform. This function will take the output of the previous function (known as *dial_sig*), the frequency we are sampling at, and the amount of segments that we want. We will make one variable for the length of each segment by taking the length of *dial_sig* and dividing by the number of segments we want which is all rounded down to the nearest integer. We will make another variable that will be the empty array for the magnitudes of the DFT (called DFTmagnitudes).

Next, we start a for loop that goes from 1 to our number of segments. We will create variables for the beginning and end slices of our signals. The beginning slice is acquired by taking 1 plus the for loop index times the length of each segment minus the length of each segment. The end slice is acquired by multiplying the for loop index with the length of each segment. These are then put into a new variable we will call *dial_Column*. The FFT is then performed on *dial_Column*, then the magnitude of *dial_Column* is acquired and squared. This allows us to get the primary frequency content. Then, insert this new array into our empty array from earlier (Remember, this is performed all in the for loop).

Afterwards, we define *tBin* and *fBin* which will act as the time and frequency axes respectively. *tBin* is acquired by multiplying the number of segments with *n*, where *n* is a vector from 1 to the number of segments, all divided by the sampling frequency. *fBin* is acquired by multiplying the sampling frequency with *k*, where *k* is a vector from 1 to the length of each segment, all divided by the number of segments. The contour and surface plots are then plotted with *tBin* as our x axis, *fBin* as our y axis and our result from the for loop as our z axis.

Results

If performed correctly, you should get something similar to the following:

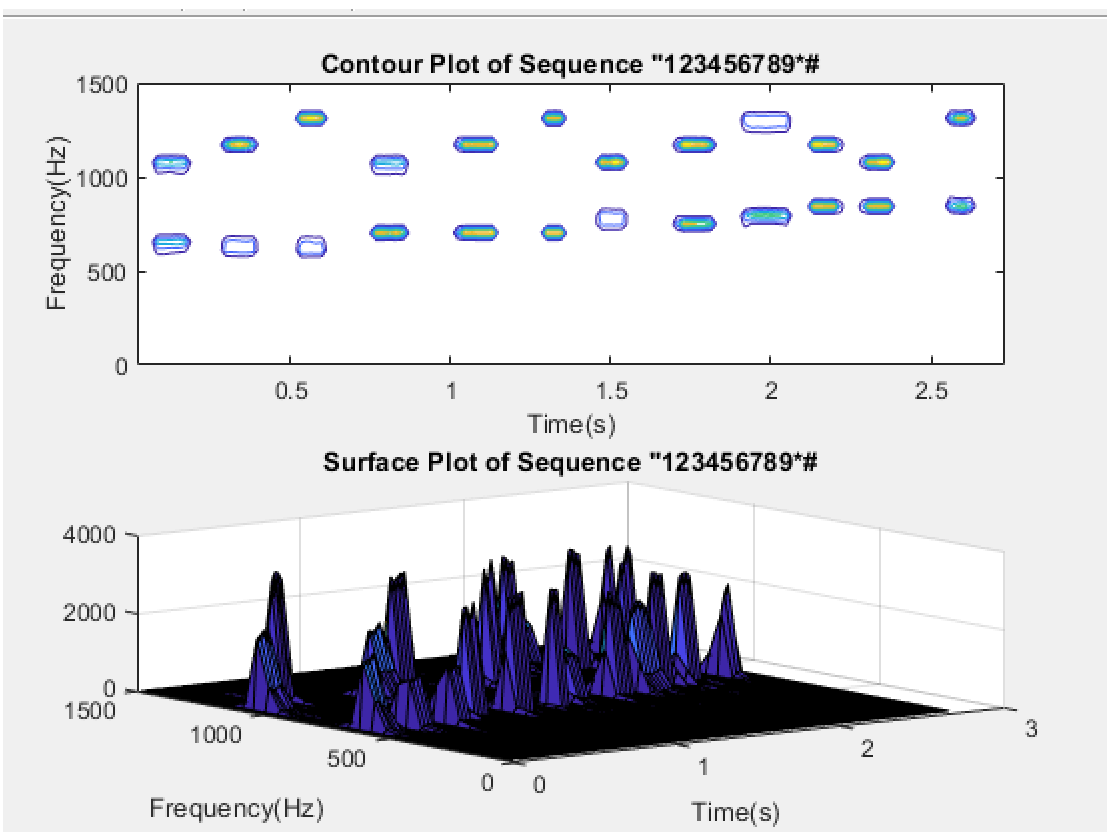

Figure A. Sequence dialed: ‘1234567890*#” Nseg = 128 Fs = 6000

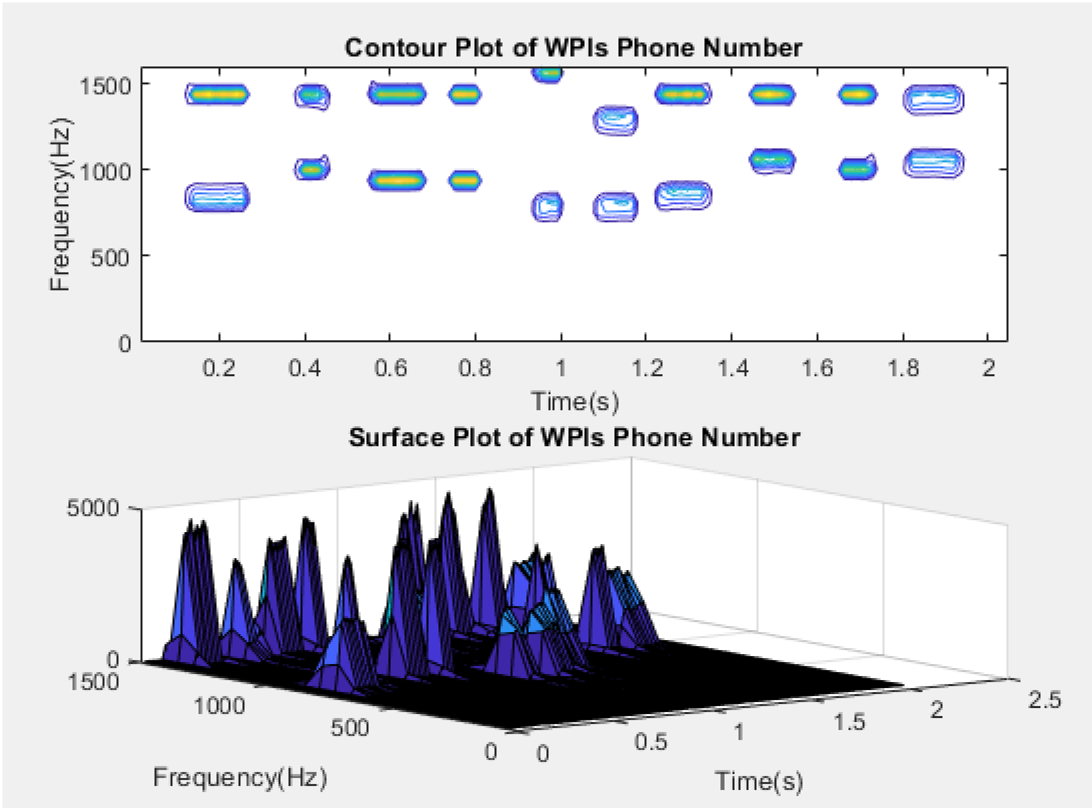

Figure B. Sequence dialed: ‘5088315000” Nseg = 128 Fs = 8000

Discussion

______As you can see from the results above, we were able to very accurately take the land line signal, transform it, and then convert it back to its original signal. When you compare the contour graph to the table that Professor Clancy provided, the numbers match up exactly at both frequency locations. When repeated, the MATLAB function works with both the example string and WPI’s main phone number.

Conclusion

This assignment is a more practical use of the DFT when compared to HW4. This gave us a real world example in which we could apply signal processes to derive what we want. The process that we used today may not have been the most optimal, but it performs accurately to what we need. This same type of code can be used in the future to evaluate other even more difficult problems, some of which may change many people's lives.

Appendix

```
function stftplot = hw7ece2312(dial_sig,Fs,Nseg)

% dial_sig = array of values from the entered phone number
% Fs   = Frequency we are sampling at
% Nseg = number of segments

Ssignallength = floor(length(dial_sig)/Nseg); % Amount of samples in each segment
DFTmagnitudes = zeros(Ssignallength,Nseg);  % Empty array for the magnitudes of the STFT



for x = 1:Nseg

    sliceStart = 1 + x*Ssignallength - Ssignallength;
    sliceEnd = x*Ssignallength;
```

```
    dialColumn = dial_sig(sliceStart:sliceEnd); % Defines where we start and where we end our slices
    magFFTsquared = (abs(fft(dialColumn))).^2; % Performing the fourier transform and taking only the positive values
    DFTmagnitudes(:,x) = magFFTsquared; % Inserting the values into our empty array
end

k = 1:Ssignallength;
n = 1:Nseg;
fBins = (k*Fs)/Nseg;
tBins = (n*Nseg)/Fs;

subplot(2,1,1)  % plotting
contour(tBins,fBins,DFTmagnitudes);
ylim([0 1600]);
xlabel('Time(s)');
ylabel('Frequency(Hz)');
title('Contour Plot of WPIs Phone Number')

subplot(2,1,2)
surf(tBins,fBins,DFTmagnitudes);
ylim([0 1500]);
xlabel('Time(s)');
ylabel('Frequency(Hz)');
title('Surface Plot of WPIs Phone Number')




end
```