

May 7, 2018

EE 356/CS 376

Project 02 – C#, MATLAB, sound files

Due: September 17, 2018

MATLAB is a programming language which has extensive libraries that are applicable to electrical engineering. For this project you will call a number of MATLAB libraries from C# to do calculations on sound files.

Your C# program will do the following:

1. Allow the user to choose a .wav file.
2. Provide user options to do the following:
 - A) Call a MATLAB function which will return information about the wav file including the following: sample frequency, number of channels (mono or stereo), number of samples, bits per sample, and duration in seconds.
 - B) Call a MATLAB function to play the wavfile.
 - C) Plot the wav file in time and frequency (two plots) using the forms charting tool. (You will need to call a MATLAB function to read the wav file which will provide the time domain data and a second MATLAB function to get the Fourier transform frequency domain data.)
 - D) Allow the user to modify either the time axis or the frequency axis by enter new start and stop values.
 - E) Allow the user to create a filtered version of the wavfile after entering a filter type (low pass or high pass), a cutoff frequency ($fs/10 \leq fc \leq 4fs/10$), a sample frequency (fs), and a file name for the filtered wav file. The user should be able to repeat step B and play the filtered wav file.

The following MATLAB functions on the following page have been written for you. Your C# program will call these functions to get the required data.

Before you can use the MATLAB function you will need to add a reference to the MATLAB application. Right click on References. Click on Add reference. Under COM locate Matlab Application and select that as a new reference. In your code add the line:

```
MApp.MLApp matlab = new MApp.MLApp();  
// Change to the directory where the functions are located  
matlab.Execute(@"cd c:\courses\EE356\2018\MLFunction");
```

Once this is done you can use the functions listed below as shown.

MATLAB Functions

Function	Comments
MLWavInfo	<p>Sample usage:</p> <pre>object result1 = null; matlab.Feval("MLWavInfo", 5, out result1, fileName); object[] res1 = result1 as object[]; fs = (double)res1[0]; ch = (double)res1[1]; totSamp = (double)res1[2]; dur = (double)res1[3]; bits = (double)res1[4];</pre> <p>fileName is a string and the name of the wav file.</p>
MLReadWavFile	<p>Sample usage:</p> <pre>object result = null; matlab.Feval("MLReadWavFile", 2, out result, fileName); object[] res = result as object[]; double [,]d = (double[,])res[0]; fs = (double)res[1];</pre> <p>The data is in d which is a 2-D array but the second column is always zero.</p>
MLPlayWavFile	<p>Sample usage:</p> <pre>matlab.Feval("MLPlayWavFile", 0, out p, fileName);</pre> <p>filename is a string as in doorbell.wav. The out variable is empty.</p>
MLFreqResp	<p>Sample usage:</p> <pre>object result2 = null; matlab.Feval("MLFreqResp", 2, out result2, fileName); object[] res2 = result2 as object[]; double[,] fResp = (double[,])res2[0]; double[,] freq = (double[,])res2[1];</pre> <p>Note that fResp and freq come back as 2D arrays but one dimension is always 0. The freq array goes from 0 to fs (with corresponding values in fResp) but it is only necessary to plot data from 0 to fs/2.</p>
MLFilter	<p>Sample usage:</p> <pre>object dummy; double fc = fs/5; string wavIn = fileName; string wavOut = "FilteredWav.wav"; matlab.Feval("MLFilter", 0, out dummy, wavIn, wavOut, 0, fc);</pre> <p>The first 0 in the arguments indicates that there are no output parameters. The second 0 is for a low pass filter. Change this to 1 for a high pass filter.</p>

Minimum Requirements:

1. Your project must use C# WPF.
2. You should have a menu and a filechooser option that allows the user to browse to a wav file location.
3. Provide working options for items A, B, C, D, and E above.

May 7, 2018

Extras:

1. Allow the user to add other menu options such as About and eXit.
2. Allow the user to change the background and foreground color of the plots.
3. Hide options that are not available such as filtering when out of range cutoff frequencies are chosen.

Turn in the following electronically to your instructor

1. A Word document titled `EE356P2XXX.docx` (where `XXX` is your three initials) which contains:
 - Cover Sheet – The cover sheet should include your name, course number, project number, project title, and date handed in.
 - A short description of your project. Include an overview and any special features that you added that are not in the specifications.
2. A complete code file that can be executed. Your code file must have well documented source code.

Compress these two items into a zip file should have the title `EE356P2XXX.zip` where `XXX` are your three initials and email them to your instructor.