Midterm 1: Speech Recognition

Suppose that we want to design a system to recognize the words for the ten digits: “zero,” “one,” “two,” …, “nine.” One of the first things that we might do is analyze data values collected with a microphone for the ten corresponding sequences (or signals) to see if there are some statistical measurements that would allow us to distinguish these digits. The MATLAB data analysis functions allow us to compute these measurements easily. We could then print a table of the measurements and look for those that allow us to distinguish values. For example, one measurement might allow us to narrow the possible digits to three, and another might allow us to identify the specific digit from the three possible digits.

Write a MATLAB program (midterm1a.m) to read and plot the Microsoft® wave files 3.wav, 4.wav, and 5.wav that contain the utterance of the word “three”, “four”, and “five”, respectively. Compute the statistical measurements of each voice and add tolerances to the statistical measurements to form a set of acceptance criteria for each voice.

MATLAB can read any WMA, MP3, MPEG-4 AAC, WAV, FLAC audio files.

```
[y,Fs] = audioread('fileName.mp3'); % save data to y at speed Fs
sound(y,Fs); % plays the file
```

Use the subplot function to generate 3x1 subplots in one figure. Plot the utterance of “three”, “four”, and “five” in each plot. All graphs should be titled clearly, for example, “Utterance of the word THREE”. The x-axis for all graphs should be labeled as “Index”. The y-axes for all graphs should be labeled as “Speech Utterance”.

The program should also compute the following statistical measurements for the utterance data from the 3.wav, 4.wav, and 5.wav files:

1. Mean: **mean**(x)
2. Standard deviation: **std**(x)
3. Variance (the squared value of standard deviation): **std**(x) .^ 2
4. Average magnitude (the average of the absolute value of each data): **mean**(**abs**(x))
5. The number of zero crossings: The number of times that the values transition from a negative to a positive value or from a positive to a negative value.

Apply the following tolerances to the statistical measurements. Compute the lower and upper tolerance limits for the voice of “three”, “four”, and “five”.

Measurement	**3.wav**	**4.wav**	**5.wav**
Mean	**+/- 0.001**	**+/- 0.001**	**+/- 0.001**
Standard deviation	**+/- 0.008**	**+/- 0.004**	**+/- 0.015**
Variance	**+/- 0.0015**	**+/- 0.001**	**+/- 0.0015**
Average magnitude	**+/- 0.006**	**+/- 0.003**	**+/- 0.015**
The number of zero crossings	**+/- 35**	**+/- 80**	**+/- 35**

Write the computation results to a file, voiceSignature.txt, in the following format. Use the format controls to control display. For **floating point** values, print **5 digits** after the decimal point. For **integer** values print **no digit** behind the decimal point.

An example of the voiceSignature.txt file should contain the following information:

```
Acceptance criteria for the word THREE:
mean: -0.00178 (-0.00278 - -0.00078)
standard deviation:  0.07729 ( 0.06929 -  0.08529)
variance:  0.00597 ( 0.00447 -  0.00747)
average magnitude:  0.04077 ( 0.03477 -  0.04677)
zero crossings: 167 (132 - 202)

Acceptance criteria for the word FOUR:
. . .

Acceptance criteria for the word FIVE:
. . .
```

You may like to combine the acceptance criteria for each voice into a minimum vector and a maximum vector so that the computation can be written in loops and functions.

The program should also **save all the computed result in a .mat file**, so that other MATLAB programs may utilize the computation results.

(Optional)

- You may record your own .wma files by going to Start -> All Programs -> Accessories -> Sound Recorder, and click on the “Record” button (require sound card and mic).
- To record .wav files on Windows, go to Start, type **soundrecorder /file outputfile.wav** in the Start Search box, and then press ENTER.

Write a second MATLAB program (midterm1b.m) to simulate voices being received into one particular voice file. Analyze the voice in the file and utilize the statistical measurement computed earlier to determine what the voice says.

Load all the variables saved by the first program to this program**.**

Write a loop to let the user input as many .wav file names as the user wishes from the command window to simulate the voices being received into o particular voice file, “receiveVoice.wav”.

An example of the command window may look like the following:

```
Voice Recognition Program

Input a voice file name (or press q to quit):
```

The program should wait for the user input. If the user wishes to quit, the program should end. If the user types in a file name, the program should check if the file exists in the current folder. If the user input file name does not exist, print an error message on the command window. If the file is available, copy the voice file to the "receiveVoice.wav" file.

The following MATLAB built-in functions may be useful.

- exist
- copyfile

An example of the command window dialog for the file does not exist and quit may look like the following:

```
Voice Recognition Program

Input a voice file name (or press q to quit):  10.wav
The 10.wav file does not exist!

Voice Recognition Program

Input a voice file name (or press q to quit):  q

Goodbye!
```

Whenever a new sound file is copied to the "receiveVoice.wav" file, the program should play the sound from the "receiveVoice.wav" file.

Generate three (3) Figure Windows. In the first figure, **use the subplot function to generate 2x1 subplots. Plot the utterance of three** on top **and the utterance from the "receiveVoice.wav" file** at the bottom. In the second figure, **use the subplot function to plot the utterance of four** on top **and the utterance from the "receiveVoice.wav" file** at the bottom**.** In the third figure, **use the subplot function to plot the utterance of five** on top **and the utterance from the "receiveVoice.wav" file** at the bottom**.** The graphs should be titled and labeled clearly.

Compute the mean, standard deviation, variance, average magnitude, and the number of zero crossings from the data in the "receiveVoice.wav" file.

Compare each of the statistical measurements obtained from the "receiveVoice.wav" file against each of the acceptance criteria (the lower and upper tolerance limits) for the voice of "three", "four", and "five". A voice is matched when all 5 statistical measurements are within the tolerance range.

Your program should do the actual mathematical comparison of the acceptance criteria for "three", "four", and "five". If a voice file for the digit "one" is input by the user, your program should recognize that the number is not "three", "four", nor "five".

You may like to utilize loops and functions to significantly reduce the number of lines in your program.

If the voice in the "receiveVoice.wav" is the voice of "three", "four", or "five", print the result on the command window. At the same time, **append the numeric digit (i.e., 3, 4, or 5) into a "voiceCommand.dat" file**, so that other programs may read the input from this file and perform other actions.

An example of the command window dialog may look like the following:

```
Voice Recognition Program

Input a voice file name (or press q to quit):  test3.wav
The voice says 3!

Voice Recognition Program

Input a voice file name (or press q to quit):  test1.wav
The voice is not 3, 4, nor 5!

Voice Recognition Program

Input a voice file name (or press q to quit):  test4.wav
The voice says 4!

Voice Recognition Program

Input a voice file name (or press q to quit):  q
Goodbye!
```

The corresponding test sequence above should append the following data in the "voiceCommand.dat" file:

```
3
4
```