EE432 Project 05: Speech Processing (Due: 10/3/08)
(Work in groups of no more than 2, one report per group)

Introduction

In this project, you will write some MATLAB code that will be used to process speech signals. Part 1 computes overall average power and number of zero crossings for single spoken letters. In Part 2, signals will be broken into short intervals, then each interval will be identified as belonging to one of three basic segments of speech: voiced, unvoiced or silence.

I. Speech Processing, Part 1.

1. Create a speech signal by recording yourself saying "e", a pause, then "s" in a .wav file, sampled at 8000 samples/sec. You can call it "es.wav". Normalize the signal using your *normalize432* function.

   Create a 2nd speech signal by recording yourself saying "oo" (as in "moon"), a pause, then "ch" as in "chin". Call this file "ooch.wav". Normalize it using *normalize432*.

2. Use your *FindSignalStart* and *FindSignalEnd* functions to get rid of the beginning noise and ending noise, so that the signal starts just at the beginning of "e" and ends at the end of "s".

3. Use your *FindSignalStart* and *FindSignalEnd* functions to get rid of the noise that separates the "e" and the "s" and use that to create two signals, one for "e" and one for "s". I suggest you use variable names *e* for "e" and *s* for "s".

4. Repeat step 3 for "oo" and "ch".

5. Using the code contained on page 72 of Section 2.6 of the Etter/Ives notes, compute the average power and number of zero crossings of each signal and fill in the following table.

Metric	**Voiced "e"**	**Voiced "oo"**	**Unvoiced "s"**	**Unvoiced "ch"**
Average Power				
Zero Crossings				

6. Is the information you obtained in the table above consistent with what you would expect of voiced and unvoiced signals?

7. Create a 3x1 subplot with your "es" signal plotted in the top window versus time in seconds, the "e" by itself in the middle window, and the "s" in the bottom window. Be sure to put in labels and a title. Add a text box to the "e" plot, and another to the "s" plot that specifies the computed average power and the number of zero crossings for the signal.

8. Repeat step 6 for the "ooch" signal.

(turn this page over)

II. Speech Processing, Part 2.

1. Implement the speech processing algorithm in Example 2.6 of the Etter/Ives notes. Note that the notes contain code you can enter into your own m-file, except that it uses a file of data called speech1.txt, which you don't need and won't use…you will use your own recorded signals.

   Specifically, write a MATLAB program(s) that will allow you to recreate Figures 2.40 and 2.41 of the text for (1) your "es.wav" file, and (2) one of your "Should we chase?" files. That is, turn in a total of 4 plots for this exercise.

   Important: the choice of values for the thresholds used in the code from the Etter/Ives notes may not work for you. These thresholds are used to determine whether a 30 msec segment of the signal is voiced, unvoiced or silence. You will probably be forced to alter these in your code to have a reasonable result. You should be able to look at a signal plot and pick out which portions are voiced, unvoiced and silence…so you can judge if your algorithm works well or not.

   Note that Figure 2.40 is a 3x1 subplot with top plot being the original signal, middle plot the normalized average power at 30 msec intervals, and the bottom plot the normalized zero-crossings at 30 msec intervals.

   Figure 2.41 is a 2x1 subplot, with the top plot being the original signal and the bottom being the classification of each 30 msec interval into either voiced, unvoiced or silence.

2. Comment on how well your algorithm was able to distinguish between voiced, unvoiced and silence portions of the signals.

For this project, turn in a formal report. I will count it twice as much as any of the projects we have done so far. Be sure to include:

-your figures
-answer questions asked above
-include table from Part 1
-comment on how well your algorithm worked in Part 2