BELMONT UNIVERSITY, CURB COLLEGE OF ENTERTAINMENT AND MUSIC BUSINESS

AET 5410 Homework 3

Digital Audio, Computer Programming, Signal Analysis

Due: November 6th

1 ANALYZING PITCH USING THE ZERO-CROSSING RATE

There are many different approaches audio engineers use for analyzing the pitch of a signal. One approach uses a measure called the **zero-crossing rate** (ZCR). This measure is the number of times the signal's amplitude crosses zero per second. In other words, it is the number of times the amplitude switches from positive to negative (or vice versa) per second.

Consider the waveform of the sine wave signal shown in Figure 1.1. The number of cycles per second can be counted by knowing where the start and end of each cycle occurs. During a cycle there are two points in time when the amplitude crosses zero. The first occurs at the phase of 0° and the second occurs at the phase of 180°. Therefore, the frequency in cycles per second can be calculated by dividing the zero-crossing rate by two, $f = \frac{ZCR}{2}$.

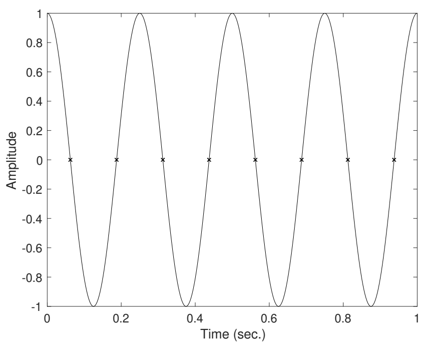

Figure 1.1: Marking a Signal's Zero Crossings

1.1 Analyzing Pitch Over Time

When analyzing an audio signal, it is common to determine the pitch at a specific point in time. This is because the pitch varies over time in many audio signals. Simply finding the ZCR of an entire signal would only determine the overall average pitch. A different approach would be to find the ZCR for short segments of the signal. A segment can also be called a frame or a buffer. It is assumed the pitch does not change during a single segment. This allows for the possibility that the pitch may be different from one segment to the next, indicating how pitch changes throughout a signal.

In order to have a smooth transition from one frame to the next, it is common to overlap part of one frame with the subsequent frame. For this problem, you should index the signal to create frames that overlap by 50%.

For this problem you will write a script to analyze the pitch of a signal. You may use some of the code from class for determining the zero-crossing rate for portions of the signal. A sound file is provided which alternates between silence, a 440 Hz sine wave, and a 500 Hz square wave. Create and save a **script** (m-file) in MATLAB that performs the following steps:

- Name the script: `pitchDetectionZCR.m`
- Initialize the following variables at the top of the script
  - *x* : input signal to be analyzed
  - *Fs* : sampling rate
  - *bufferSize* : number of samples to be included in each segment/frame
  - *overlap* : amount of overlap between buffers/frames
- Set up a loop to index segments of the input signal based on the *bufferSize* and *overlap*
- For each segment, find the ZCR
- Determine frequency (pitch) based on the ZCR
- Store this value as one element in the output array: *pitch*
- After the entire signal is analyzed, plot the *pitch* array to view the detected fundamental frequency

Remember to add comments to your code to explain what each command is accomplishing. For this problem, you will submit the script file - `pitchDetectionZCR.m`.

2 Introduction: Signal Correlation

Correlation is a measurement of the similarity in signals. One type of correlation is called: *cross-correlation*. It is a measurement of the similarity between two different signals. The cross-correlation between signals x and y is written: $r_{x,y}$.

Another type of correlation is called: *auto-correlation*. It is a measurement of how similar a signal is with itself at different points in time. The auto-correlation for signal x is written: $r_{x,x}$.

The concept of the measurement is to line two signals up, perform element-wise multiplication, and then sum over all the elements. Then, slide one of the signals along the other one and perform the same operation.

2.1 Auto-correlation Calculation

One auto-correlation value is calculated for each time lag, l. Each value is stored in an the array, $r_{x,x}$, for integer time lags, $l = (-N+1), (-N+2), \ldots, -1, 0, 1, 2, \ldots, N-1$, where N is the length of signal $x[n]$, with sample numbers, n. The result is stored in an array with MATLAB indexing, $r_{x,x}[1], \ldots, r_{x,x}[2N-1]$. The following is an equation to calculate one correlation value for a single time lag:

$$r_{x,x}[N+l] = \sum_{n=1}^{N} x[n] \cdot x[n+l] \tag{2.1}$$

One approach to performing this calculation is to zero-pad the beginning or end of array, $x[n]$, to line up the appropriate values for each time lag. This approach ensures the arrays are identical dimensions, a necessary requirement of element-wise multiplication. The drawback of this approach is it uses a large number of multiplications by zero, which is unnecessary and inefficient.

A second approach is to perform this calculation by indexing only the elements of the arrays which overlap, and exclude the non-overlapping values in each calculation. In this case, the number of multiplications is minimized to only those necessary.

2.2 Problem: Correlation Matlab Script

For this problem, write a Matlab **script** to perform the correlation analysis for two signals. You should use the script provided with the homework as a starting point: `userCorrelation.m`. The script should calculate two arrays, r and l, representing the array with correlation values and an array of time lags, repectively.

The script should use the approach of indexing only the necessary elements for each time lag. Additionally, the calculation should use the *inner product* to perform the element-wise multiplication and addition in the measurement. This will require one horizontal row vector and one vertical column vector when using the operator, `x * y`.

You should verify the performance of your scipt by comparing your variables, r and l, to the output of the built-in function:

```
[R,L] = xcorr(x,y)
```

Remember to add comments to your code to explain what each command is accomplishing. For this problem, you will submit the script file - `userCorrelation.m`

3 ANALYZING PITCH USING AUTO-CORRELATION

Correlation is a fundamental and important measurement used in many analyses of audio signals. As an example, auto-correlation can be used to determine the pitch of a signal.

The autocorrelation function can be used to determine a signal's pitch by finding the first peak (i.e. local maximum) for a positive time lag in the auto-correlation function. This local maximum occurs at a time lag equal to the period of the signal. The period can then be converted to the frequency, $f = \frac{1}{\tau}$, representing pitch.

For audio signals which are non-stationary (i.e. they do not stay the same the whole time), the auto-correlation function can be analyzed in short segments (buffers). It is assumed the pitch does not change during a single segment. However, the pitch may be different from one segment to the next.

3.1 PROBLEM: PITCH DETECTION

For this problem, you will recreate a similar analysis performed in problem 1. Instead of using the zero-crossing rate, you will now use an auto-correlation function.

For this problem, you may use the built-in MATLAB function for calculating correlation. A sound file is provided which alternates between silence, a 440 Hz sine wave, and a 500 Hz square wave. Create and save a **script** (m-file) in MATLAB that performs the following steps:

- Name the script: pitchDetectionCorr.m
- Initialize the following variables at the top of the script
  - x : input signal to be analyzed
  - Fs : sampling rate
  - *bufferSize* : number of samples to be included in each segment/frame
  - *overlap* : amount of overlap between buffers/frames
- Set up a loop to index segments of the input signal based on the *bufferSize* and *overlap*
- For each segment, find the auto-correction
- Determine frequency (pitch) based on the location of a peak
- Store this value as one element in the output array: *pitch*
- After the entire signal is analyzed, plot the *pitch* array to view the detected fundamental frequency

Remember to add comments to your code to explain what each command is accomplishing. For this problem, you will submit the script file - `pitchAutoCorr.m`.

4 SUBMISSION

To submit your homework, create a single zip file that contains the MATLAB m-file scripts and the sound files you used in the problems. Name the zip file: xxxxx_AET5410_HW3.zip, where *xxxxx* is your last name. Email the zip file to: eric.tarr@belmont.edu before 5 pm on November 6th.