Assignment: Homework 2

How to Hand It In

1. Put all your solutions in one folder. Compress this folder and name it <firstname>_<lastname>_HW2.zip. For example, "Zhiyao_Duan_HW2.zip".
2. Submit to the corresponding entry on Blackboard.

When to Hand It In

It is due at 11:59 PM on the date specified on the course calendar. **Late assignments will receive a 20% deduction of the full grade each day.**

Problems (10 points in total)

1. (7 points) Implement the YIN algorithm (Step 2-5) [1] and test your implementation on some recordings.
    a. (2.5 points) Write a function named "myYin" to implement the YIN algorithm (Step 2-5), following the input/output specifications given in the provided "myYin.m". The points break down as follows. Your program should be well commented to receive the points.
        i. (0.5 points) Step 2.
        ii. (0.5 points) Step 3.
        iii. (0.5 points) Step 4.
        iv. (0.5 points) Step 5.
        v. (0.5 points) Your program runs. You should print out some information (e.g., which frame is being processed) on the screen during the run so that we know your program is not dead.

    Hint 1: To make your program run faster, consider reusing the autocorrelation and difference function calculations in the previous frame. This is described in Section V of [1].
    Hint 2: You could skip Step 5 and still get your program run (and lose 0.5 points), but I don't think you can skip Step 2-4 to get it run. If you want to implement Step 5, the formulae of parabola interpolation can be found here (you can derive them by yourself too):
    https://ccrma.stanford.edu/~jos/parshl/Peak_Detection_Steps_3.html.

    b. (1point) Write an evaluation function "evalSinglePitch" to calculate Accuracy, Precision, and Recall of single pitch estimation. The specifications of input/output are given in the provided "evalSinglePitch.m".

    c. (1 point) Run your myYin function on the provided "violin.wav", with the following parameters:
    [pitch, ap_pwr, rms] = myYin(wavData, fs, 0.01, 0.0464, 40, 2000, 0.1);
    Plot three figures: 1) pitch vs. time; 2) ap_pwr vs. time; 3) rms vs. time. Label the axis of your figures properly. By looking at the figures and listening to the audio, choose appropriate thresholds for ap_pwr and/or rms to decide if there is a pitch or

not in each frame of the audio. If you think there is no pitch in an audio frame, then you should modify your pitch estimate to 0 Hz in that frame. Plot a new figure of the final pitch estimate vs. time.

d. (1 point) Evaluate your pitch estimates against the provided ground-truth pitch values stored in "violin_gt_pitch.mat", using your evaluation function. Report your accuracy, precision, and recall.
   Hint 3: Your estimated pitch vector might be longer or shorter than the ground-truth pitch vector. This won't affect the evaluation results since the two vectors are aligned in the beginning, i.e., the first element of each vector is aligned to the first frame of the audio (which is 46.4ms long), and their hop sizes are the same (which is 10ms). The last couple of frames are usually silent. You can simply discard the last several elements of the longer vector to make the two vectors have the equal length.

e. (0.5 points) Run your myYin function on the provided "violin_noise.wav". Keep the tHop and tW parameters same as before, but you may change the other parameters. Again, choose appropriate thresholds for ap_pwr and/or rms to determine the pitch/nonpitch frames. Plot your final pitch estimate vs. time. Evaluate your pitch estimates against the ground-truth and report the accuracy, precision, and recall. Does the performance of your myYin function drop significantly from the noiseless violin recording to the noisy recording?

f. (1 points) Listen to "violin_noise.wav". Can you hear the melody clearly? Plot the spectrogram of "violin_nois.wav". Can you see/guess where the pitches are? From your hearing and observation, do you have any idea on how to get better pitch estimation on this file? Your idea does not have to have any relation with YIN. It could be any crazy idea.

2. (3 points) In this problem, you will attempt to estimate pitches of a polyphonic music piece. Please describe your thoughts and also write code to implement your thoughts.
   a. (1 point) Run your myYin function on the provided "violin_bassoon.wav". Choose appropriate parameters. Plot your final pitch estimate vs. time. Are the pitch contours reasonable? Which instrument is your algorithm detecting? Could you change parameters to make myYin detect different instruments?
   b. (1 point) Could you design a way to use your myYIN function to estimate pitches from both instruments? For example, perhaps you could run myYin twice and set different pitch ranges as suggested in the previous question. Or perhaps you could modify myYin to output multiple pitches at a time? But you may run to the issue that the multiple pitches you output in each time frame always have an octave relation.
   c. (1 point) Suppose you did successfully estimate two pitches in each time frame, now please figure out a way to connect the pitches over time into pitch contours so that

each contour corresponds to one instrument. Perhaps you could use the continuity cue, or perhaps some other cues?

References

[1] de Cheveigné, A., and Kawahara, H. (2002). "YIN, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Am*., 111, 1917-1930.