Sunday, November 14, 2021

Fundamental Frequency Detection/Pitch Tracking Results Report

---

A. Maximum spectral peak based pitch tracker

If the block size = 1024, what is the exact time resolution of your pitch tracker?

Can this be improved without changing the block-size? If yes, how?

Time Resolution = Hop Size / Sample Rate (Assuming Hop Size = Block Size / 2, we get Time Resolution = Block Size / 2 * Sample Rate)

Time Resolution = 1024 / 44100 = 0.0232 s

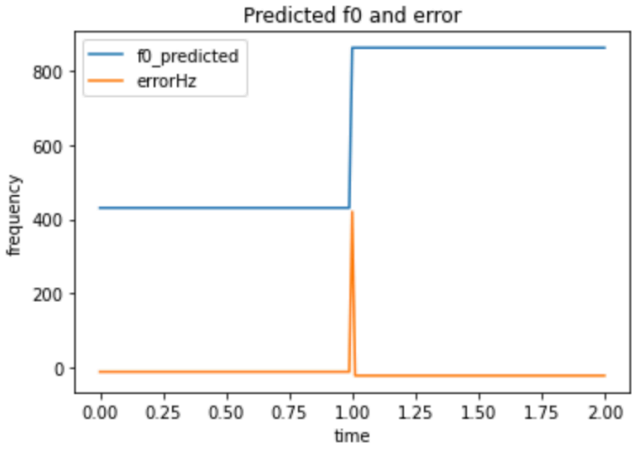
The resolution can be adjusted by either reducing hop size, or by increasing the sample rate.

Alternatively it can be increased by padding zeros to the end of each block to double the length of each FFT block. This will increase time resolution as well.

E. Evaluation

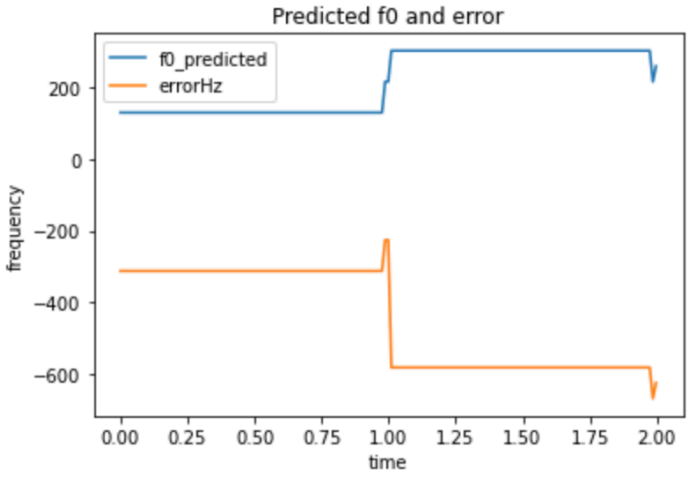
1. Apply your track_pitch_fftmax(), plot the absolute error per block and discuss the possible causes for the deviation.

The discontinuity at time=1s occurs in the middle of a block, and hence even though the signal is 441Hz for some part and 882Hz for some part, the output of the block is not correct at this point. After the discontinuity, the error is negative. (since the nearest frequency bin centre is lower than the actual frequency)



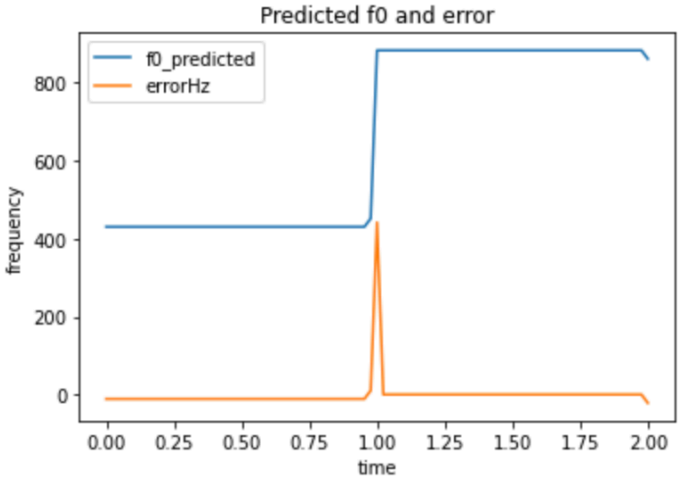
1. Repeat for track_pitch_hps(), why does the HPS method fail with this signal?

Since the test signal is a pure sine wave, without any harmonics, the HPS is unable to match the peaks after downsampling, and hence fails.



2. Next use (blockSize = 2048, hopSize = 512) and repeat the above experiment (only for the max spectra method). Do you see any improvement in performance?

When doubling block size, the frequency resolution increases, and as a result the error after the discontinuity is a very small positive value (since the nearest frequency bin centre is just above the actual frequency)



3. Evaluate your track_pitch_fftmax() using the development set(see assignment 1) and the eval_pitchtrack_v2() method (use blockSize = 1024, hopSize = 512). Report the average performance metrics across the development set.

File1 - 01-D_AMairena.wav

RMS Error = 1705.5855, PFP = 0.8417, PFN = 0.0010

File 2 - 24-M1_AMairena-Martinete.wav

RMS Error = 1988.9020, PFP = 0.9206, PFN = 0.0

File 3 - 63-M2_AMairena.wav

RMS Error = 2087.3317, PFP = 0.6897, PFN = 0.0008

Average Error = 1934.0551

4. Evaluate your track_pitch_hps() using the development set and the eval_pitchtrack_v2() method (use blockSize = 1024, hopSize = 512). Report the average performance metrics across the development set.

File1 - 01-D_AMairena.wav

RMS Error = 507.0356, PFP = 1.0, PFN = 0.0

File 2 - 24-M1_AMairena-Martinete.wav

RMS Error = 854.0372, PFP = 1.0, PFN = 0.0

File 3 - 63-M2_AMairena.wav

RMS Error = 541.6398, PFP = 1.0, PFN = 0.0

Average Error = 653.1560

6. Evaluate your track_pitch() using the development set and the eval_pitchtrack_v2() method (use blockSize = 1024, hopSize = 512) over all 3 pitch trackers (acf, max and hps) and report the results with two values of threshold (threshold = -40, -20)

AVERAGE VALUES: Threshold=-40

HPS rms error: 572.56 cents

ACF rms error: 588.42 cents

FFT Max rms error: 798.92 cents

AVERAGE VALUES: Threshold=-20

HPS rms error: 3721.49 cents

ACF rms error: 3720.37 cents

FFT Max rms error: 3740.09 cents

It is evident that a threshold of -20dB RMS is too high, and many parts of the voiced signal are being masked as unvoiced, causing the rms error to increase dramatically.

Bonus: Improving your Pitch Trackers

Implement a function: [f0, timeInSec] = track_pitch_mod(x, blockSize, hopSize, fs) that combines ideas from different pitch trackers you have tried thus far and thereby provides better f0 estimations. You may include voicing detection within this method with parameters of your choosing. Please explain your approach in the report.

To help mitigate the octave errors that are present in the output of fftmax and hps, we first compare every element with the previous element, then the element/2 (1 octave error) with previous element and then element/4 with previous element (2 octave error), and keep whichever value is closest to the previous element (within a threshold that was fine-tuned)

Next this smoothed f0 was compared to the mean of f0 (since this has the least, if any octave errors), and if it is outside a second threshold range, is replaced by f0/2.

This processing was applied to both the fftmax as well as hps, then all three methods were combined using a weighted average to give the below results for 01-A-D_AMairena:

The average error across all 3 test files is 576.41 cents.