ADAPTIVE PITCH DETECTION METHOD EMPLOYING THE USE OF FAST FOURIER TRANSFORM AND AUTOCORRELATION FUNCTON

Osborn Fong 450430390

Final Written Review for Digital Audio Systems, DESC9115, 2016
Graduate Program in Audio and Acoustics
Faculty of Architecture, Design and Planning, University of Sydney

1. PROBLEM DESCRIPTION

For a pitch detection algorithm to track the pitch of a signal well in real time, it must be able to detect changes in pitch over time quickly and accurately using code that is computationally lightweight. However, there is trade-off typically in pitch detection algorithms between time and frequency resolution; since in order to obtain better frequency resolution, a longer sampling window must be taken, which results in a decrease in time resolution (and vice versa). As different algorithms have good frequency resolution across different frequency regions, a solution to increase frequency resolution may be to run multiple processing algorithms concurrently and to select the best answer as the detected pitch. However, running multiple processing methods concurrently is computationally intensive and may not be the best solution for a real-time system. Alternatively, an adaptive algorithm can be employed, which uses the most recently detected pitch to anticipate the frequency of the next incoming frame. By anticipating the pitch of the incoming frame, the algorithm that has the best frequency resolution for the anticipated frequency range can be selected to process the oncoming signal to provide the most accurate frequency prediction.

The solution presented in this paper is a program coded in MATLAB 2015b. The solution employs the adaptive method to select between fast fourier transform (FFT) processing in the frequency domain and autocorrelation function (ACF) processing in the time domain to process frequencies anticipated to be in the high and low frequency range respectively.

2. SPECIFICATION

2.1. Inputs

The developed program is able to take live input from the default on-board microphone on the code-running computer for real-time audio processing. As a secondary function, the program is also able to read in .wav files.

2.2. Performance Goals

As the program is designed for audio containing frequencies within the range of typical human hearing, it is designed to be able to detect pitches ranging from approximately 20Hz to 20kHz.

To ensure pitch that sound subjectively different are not classified as the same, the program aims to have pitch detected by the algorithm not deviate from the Just Noticeable Difference (JND) suggested by Kollmeier et al. [1], which is approximately 3 Hz for frequencies below 500 Hz and 0.6% for frequencies above 1 kHz. For the unspecified region between 500 Hz to 1 kHz, a maximum deviation of 3Hz is taken as the design goal.

Under default settings, the program is also able to track pitch changes at a rate faster than 10 times per second.

2.3. Outputs

As the program is designed to process audio signals in real-time, the system provides real-time feedback of the detected pitch so that users are able to receive instantaneous feedback in the form of live plots and text output via the MATLAB command window interface. Output variables are also provided containing an array of the detected pitch and other information for further potential post-processing.

3. IMPLEMENTATION

Figure 1 shows an overview of the implemented adaptive procedure, which will be described in detail below.

3.1. Input and Audio Buffer

In order to process signals in real-time, code from the MATLAB Digital Signal Processing Toolbox is employed to hold live audio data in a buffer between the processing of each frame of the signal.

3.2. Method Selection

At the start of processing each frame of the signal, either FFT or ACF processing is selected based on whether the most recent determined frequency is above or below a determined crossover frequency. The crossover frequency is determined by finding the corresponding frequency difference between two ACF lags that equal or is closest to the frequency resolution given by the FFT bin size, which is the inverse of the sampling period (e.g. a sampling period of 2 seconds correspond to a FFT bin size of 0.5 Hz).

For frequencies below the crossover frequency, ACF processing will be applied to the frame since ACF provides better frequency resolution in low frequencies. For frequencies above the crossover frequency, FFT processing will be applied for better frequency resolution.

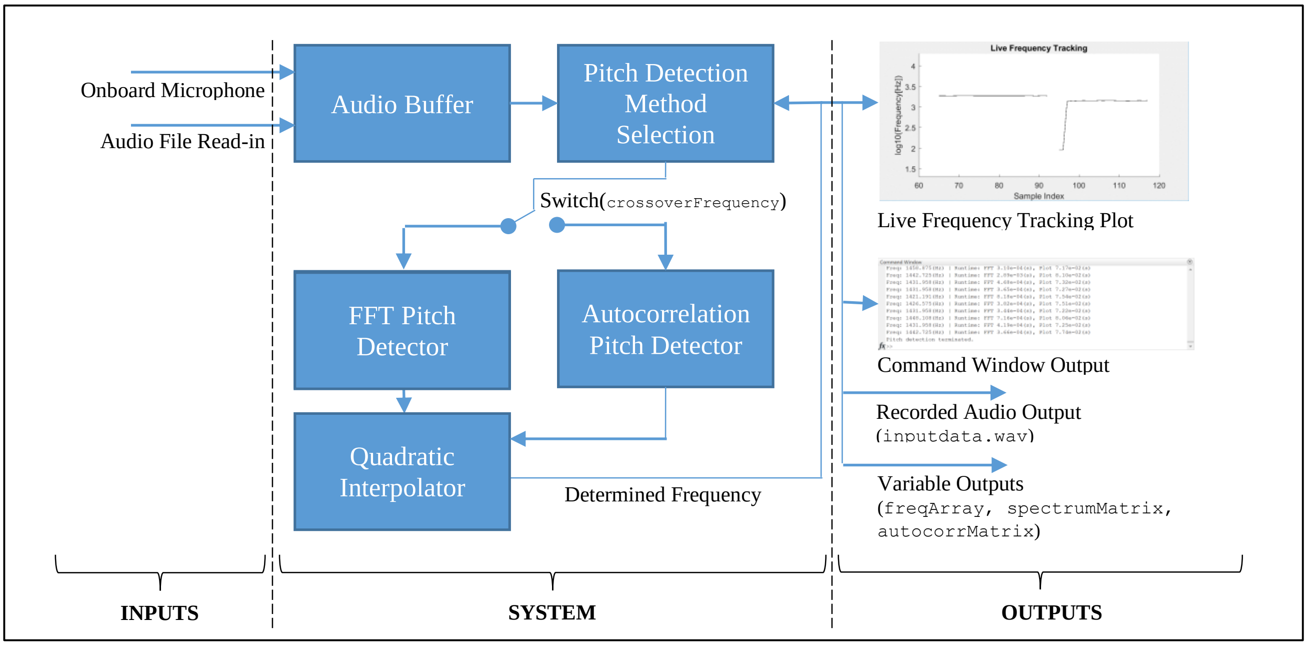

Figure 1. *Diagrammatic representation of implemented pitch detection algorithm*

3.3. Pitch Identification

3.3.1. *Fast Fourier Transform (FFT) Procedure*

If the FFT procedure is selected, the discrete fourier transform of the frame is computed via FFT and the magnitude spectrum is obtained (see Figure 1 for example). The peak of the magnitude spectrum is then found and quadratic interpolation (see Section 3.4) is applied to the peak and adjacent points on the spectrum to find the interpolated frequency. The interpolated frequency is output as the determined frequency of the algorithm.

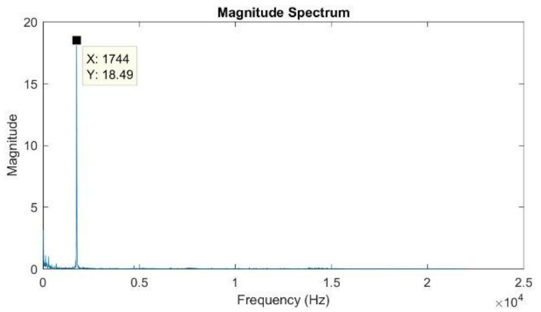

Figure 1. *Magnitude spectrum obtained from FFT of an audio signal computed by the program*

3.3.2. *Autocorrelation Function (ACF) Procedure*

If the ACF procedure is selected, the ACF of the frame is computed using unbiased correlation [2] as shown below.

$$R(\tau) = \frac{W}{W-\tau}\sum_{j=0}^{W-1-\tau} x_j x_{j+\tau}, \qquad 0 \le \tau < W \qquad (1)$$

The delay of the peaks of the autocorrelation function are then obtained and the delay at which the first peak is found is considered to be the corresponding pitch of the frame (see Figure 2 for example).

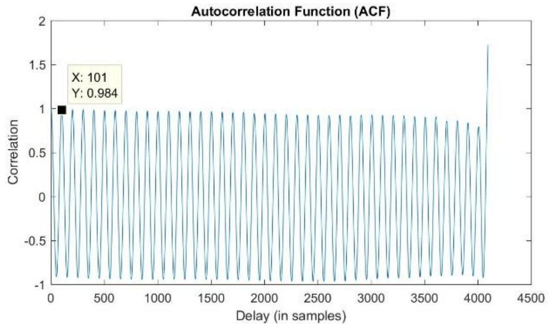

Figure 2. *Autocorrelation function of a sinusoidal tone computed by the program.*

The frequency corresponding to the first peak found is then further refined using quadratic interpolation (see Section 3.4) to obtain the interpolated frequency, which is output as the determined frequency of the algorithm.

3.4. Quadratic Interpolation

3.4.1. *Quadratic Interpolation in FFT procedure*

If the frequency bin of the peak and adjacent frequency bins (f_{p-1}, f_p, f_{p+1}) and their corresponding magnitudes (m_{p-1}, m_p, m_{p+1}) are known along with the spacing of the frequency bins (Δf), then the interpolated frequency $f_{interpolated,fft}$ follows the equation below:

$$f_{interpolated,fft} = f_p + \Delta f * \frac{m_{p+1} - m_{p-1}}{2 * (2 * m_p - m_{p-1} - m_{p+1})} \qquad (2)$$

3.4.2. Quadratic Interpolation in ACF procedure

If the delay of the first peak correlation and adjacent delays $(\tau_{p-1}, \tau_p, \tau_{p+1})$ and their corresponding magnitudes (m_{p-1}, m_p, m_{p+1}) are known along with the sampling frequency (Fs), then the interpolated frequency $f_{interpolated,acf}$ follows the equation below:

$$f_{interpolated,acf} = \frac{Fs}{\tau_p + \frac{m_{p+1} - m_{p-1}}{2 * (2 * m_p - m_{p-1} - m_{p+1})}} \quad (2)$$

4. EVALUATION

To evaluate the performance of the implemented algorithm, signals of known frequencies were processed through the algorithm to compare the output with the known input.

4.1. Exponential Sweep Test

An exponential sweep from 20 Hz to 20 kHz was processed by the adaptive algorithm and by single method processing using FFT and autocorrelation. A plot of the results is presented in Figure 3 appended to the end of this paper. Results show that the adaptive algorithm tracks the pitch of the signal well from above approximately 100 Hz and is able to produce a linear line on a logarithmic y-axis, demonstrating smooth tracking of the expected frequency. However, in the low frequency region of below 100 Hz, the adaptive algorithm was found to occasionally producing erroneous results caused by the ACF producing an unwanted initial peak of high correlation near zero delay. By comparing the adaptive results with the FFT and ACF results, it is observed that the use of ACF in the low frequencies and FFT in the high frequencies has enabled better pitch detection accuracy, which can be seen as a straighter line plotted by the adaptive process in Figure 3.

4.2. Octave-step Tone Sequence Test

An octave-step tone sequence from 20 Hz to 20 kHz was processed by the adaptive algorithm and the results compared against the theoretical values. A plot of the results is presented in Figure 4 appended to the end of this paper. The table below comparing the detected values against the theoretical values shows that all but the last two octaves of this test signal meets the JND design goal.

Theoretical	Detected	JND	Meets JND?
20 Hz	20 Hz	± 3 Hz	Yes
40 Hz	40 Hz	± 3 Hz	Yes
80 Hz	80 Hz	± 3 Hz	Yes
160 Hz	160 Hz	± 3 Hz	Yes
320 Hz	320 Hz	± 3 Hz	Yes
640 Hz	639 Hz	± 38.4 Hz	Yes
1280 Hz	1297 Hz	± 76.8 Hz	Yes
2560 Hz	2594 Hz	± 153.6 Hz	Yes
5120 Hz	4899 Hz	± 307.2 Hz	Yes
10240 Hz	11025 Hz	± 614.4 Hz	No
20480 Hz	22062 Hz	± 1228.8 Hz	No

Table 1. *Comparison of the detected frequency against the theoretical frequency of the octave stepped tone sequence from 20 Hz to 20 kHz. All but the last two octaves show compliance with the JND design goal*

5. CONCLUSION AND DISCUSSION

This paper presented a solution in developing a lightweight pitch detector with good frequency and time resolution. By switching adaptively between ACF and FFT processing based on the anticipated incoming signal frequency, greater accuracy in frequency determination is achieved over using a single method of processing, as shown in Section 4.1. In meeting the specification, the program has demonstrated its ability in providing accurate pitch detection within the JND goal for most frequencies.

However, it is noted that the peak detection method in its current form for the ACF is occasionally producing erroneous results for frequencies observed to be below approximately 100 Hz, and the FFT with quadratic interpolation may not be able to provide adequate accuracy to meet the JND design goal in around the 10 kHz octave band and above. It is anticipated that these issues can be resolved by further refining the peak detection method of the ACF procedure and by tweaking the sampling window size or using a different interpolating method for the FFT procedure.

Further to the requirement of meeting the design specification, it is proposed that the JND threshold be verified with further human perception studies to ascertain that the threshold currently set is not too stringent, and that a more relaxed design goal may not change the perception of the program's performance experienced by the end user.

6. REFERENCES

[1] Kollmeier, B. , Brand, T., & Meyer, B. (2008). *Perception of Speech and Sound*. In Jacob Benesty, M. Mohan Sondhi, Yiteng Huang. Springer handbook of speech processing. Springer.

[2] McLeod, P. G. (2008). *Fast, Accurate Pitch Detection Tools for Music Analysis*, University of Otago, New Zealand. Retrieved from http://miracle.otago.ac.nz/tartini/papers/Philip_McLeod_PhD.pdf

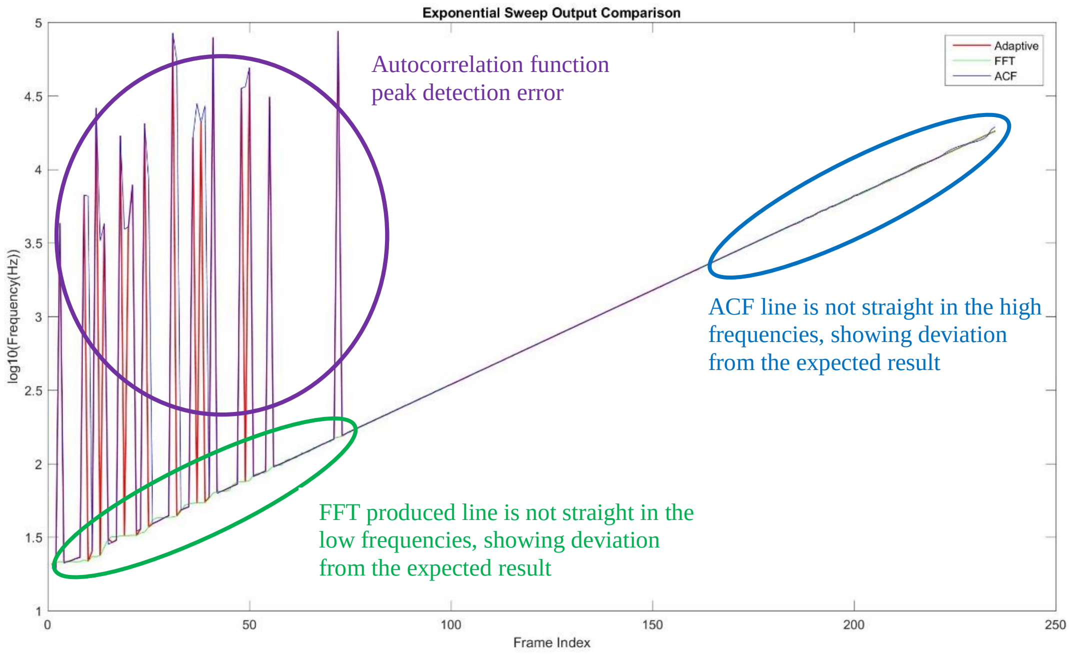

Figure 3. Comparison of an *exponential sweep processed by the adaptive algorithm (red), fast fourier transform (green) and autocorrelation function (blue)*

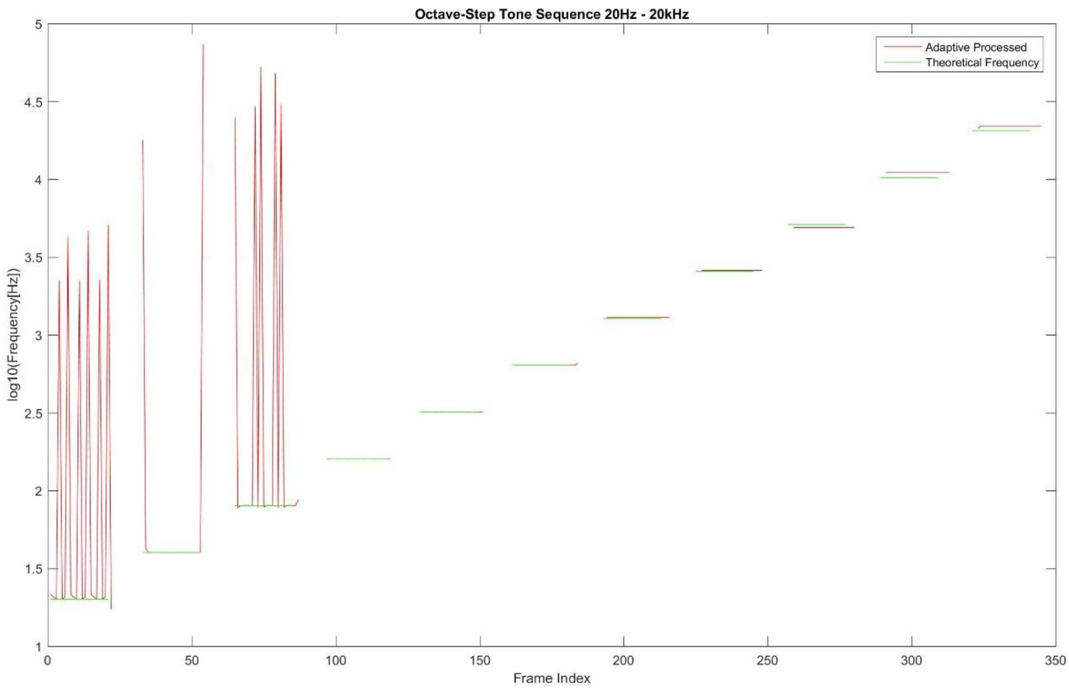

Figure 4. *Comparison of an octave-stepped tone sequence from 20 Hz to 20 kHz processed by the adaptive algorithm (red) against theoretical values (green)*