**TED UNIVERSITY**

**CMPE 468 / CS 568**

**Speech Processing**

**PROJECT REPORT**

*Pitch Detection*

*and*

*Analysis of Audio Signals*

Contents

[1. Introduction 3](#_Toc136781044)

[2. Purposed Study 4](#_Toc136781045)

[2.1. Processing 4](#_Toc136781046)

[2.2. Frame Segmentation 4](#_Toc136781047)

[2.3. Spectral Segmentation 4](#_Toc136781048)

[2.4. Pitch Estimation 4](#_Toc136781049)

[2.5. Pitch Refinement 5](#_Toc136781050)

[2.6. Pitch Tracking 5](#_Toc136781051)

[2.7. Visualization 5](#_Toc136781052)

[3. Results 6](#_Toc136781053)

[4. Conclusion and Discussion 8](#_Toc136781054)

[5. References 9](#_Toc136781055)

# Introduction

For this project we wanted to come up with a pitch detector to be used in various domains like speech recognition and speech recognition using digital speech signal processing. Pitch estimation is one of the significant challenges in speech processing. Pitch detectors can be used for analyzing and understanding the acoustic properties of sounds for different applications. Creating effective speech recognition and synthesis systems, accurate methods of pitch detection are important to implement and use. Our project includes converting audio files into digital signals, processing the data and estimating the pitch frequency. Our focus was creating an accurate and efficient pitch detector using MATLAB.

# Purposed Study

The purpose of this Pitch Detection and Analysis of Audio Signals project is to develop a pitch detection system that accurately detects and analyzes the pitch of audio signals using digital signal processing and track its changes over time. We aimed to involve the systems that are already being used for pitch detection and combine existing algorithms and libraries for pitch detection with our own while serving a reliable and improved system.

Our project goes through different parts after getting the audio input. Processing, Frame Segmentation, Spectral Segmentation, Pitch Estimation, Pitch Refinement, Pitch Tracking and lastly Visualization are the 7 total parts that combines our pitch detection project which implemented in our code in order.

## Processing

The first step is “processing” and in this step we read the audio file and store its sampling frequency and signal. Then we normalize the signal by dividing it by its maximum absolute value. By this way we are ensuring consistent amplitudes. For normalization of the signal, we used low-pass Butterworth filter which removes the high frequency noises during this process.

* 1. Frame Segmentation

The Frame Segmentation part uses a sliding window approach for dividing the denoised signal into frames. Sliding window approach is used in signal processing that divides signal into overlapping segments which are called windows. This method allowed us to analyze properties of the signal in a time-dependent manner and made it possible to perform calculations on each window independently. We set the frame duration and calculate the frame length in this part. Calculations were made based on the duration and sampling frequency. Lastly for this part we rephrased the denoised signal into frames and stored it. During the frame segmentation step we divide the denoised signal into frames of 30 milliseconds.

## Spectral Segmentation

For spectral segmentation we applied hamming window to each frame. We used hamming window function to minimize the spectral leakage during subsequent Fourier Transform while improving the accuracy. Using the Fast Fourier Transform (FFT) we compute magnitude spectra of the windowed frames.

## Pitch Estimation

In this pitch estimation part, we compute the autocorrelation of the frames by iterating through each frame. We get the index of the maximum autocorrelation lag and calculate the pitch estimation (in Hz) based on our sampling frequency. Lastly, we store the pitch estimation for each frame in an array.

## Pitch Refinement

The Pitch Refinement part obtains pitch estimation from the previous steps. In this part we used Harmonic Product Spectrum (HPS) for each frame which is a pitch refinement method generally used in pitch detection algorithms. After obtaining the initial pitch estimate, Magnitude spectrum of the frame is computed using the FFS and HPS computed for each frame by downsizing and multiplying the magnitude spectra. Then we used the index of the maximum value which we found in HPS to calculate the refined pitch estimate. Lastly we store each frame’s pitch estimation.

* 1. Pitch Tracking

In this part of our project, we calculated the starting and ending indices which are based on the frame length and assigned the refined pitch estimate to the corresponding indices in the pitch track. Lastly, we stored pitch track over time in an array.

## Visualization

This is the final output part of the code implementation of our pitch detection project. We plot the waveform of the audio signal and overlay the pitch track. We save the pitch track in a text file as well and display a message to inform that pitch track is saved.

# Results

Pitch Detection and Analysis of Audio Signal project implementation successfully processed the input audio signals from the user and produced accurate pitch estimation throughout the duration of the signal. We effectively capture the magnitude spectra of each frame using the hamming window function and get the pitch estimation using the autocorrelation method, Harmonic Product Spectrum. HPS successfully enhanced the accuracy of pitch estimation by leveraging the harmonics present in the signal. Overall, the pitch detector algorithm that we implanted successfully tracks the pitch throughout the signal and generates the pitch track as well as demonstrates a visualization of the waveform. Here are some example outcomes from our code implementation:

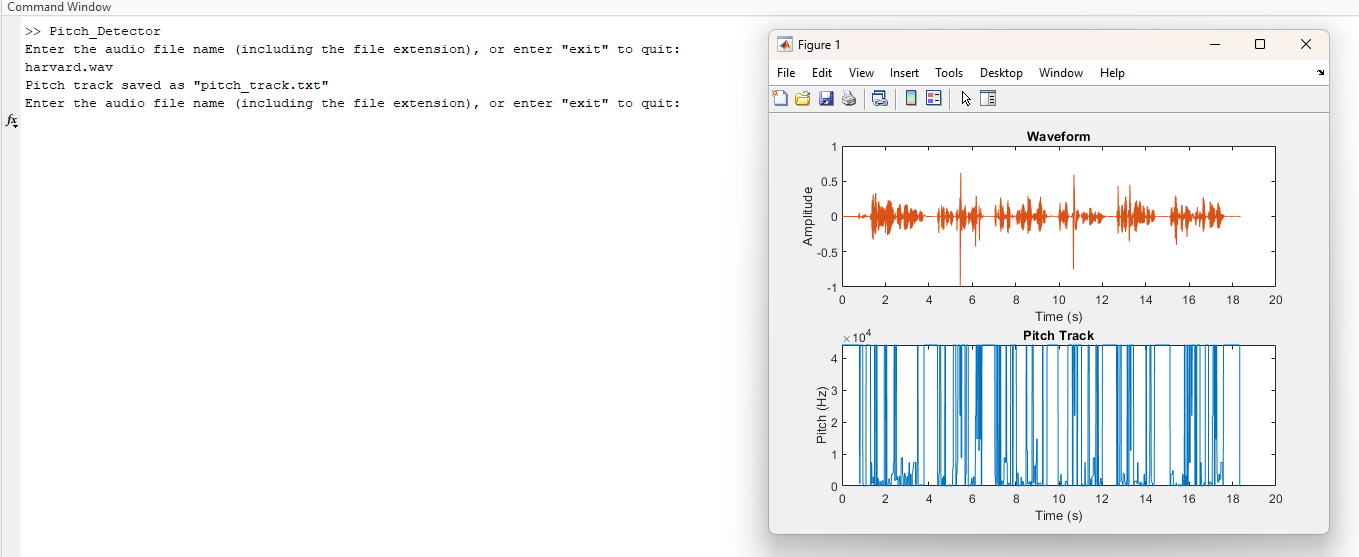


Figure 1: Pitch Detection of file "harvard.wav"

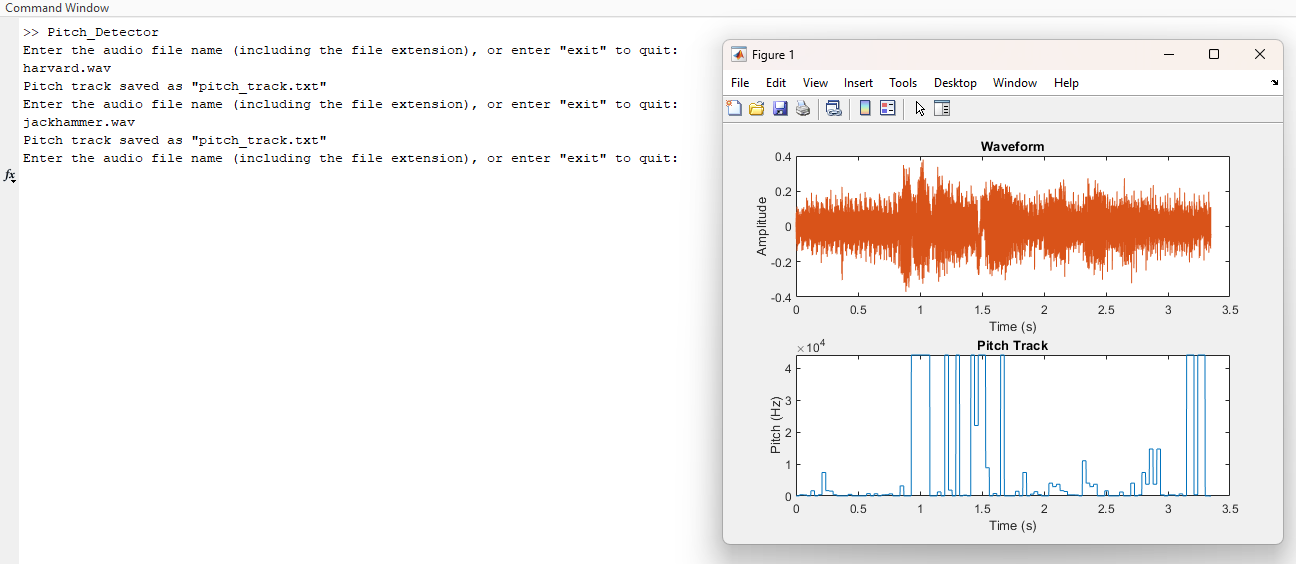


Figure 2: Pitch Detection of file "jackhammer.wav"

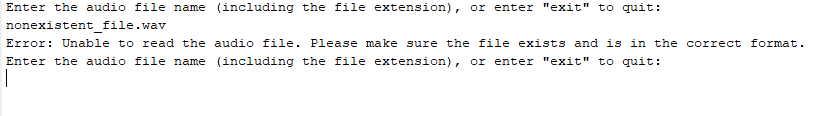


Figure 3: Scenario for non-existing input data.

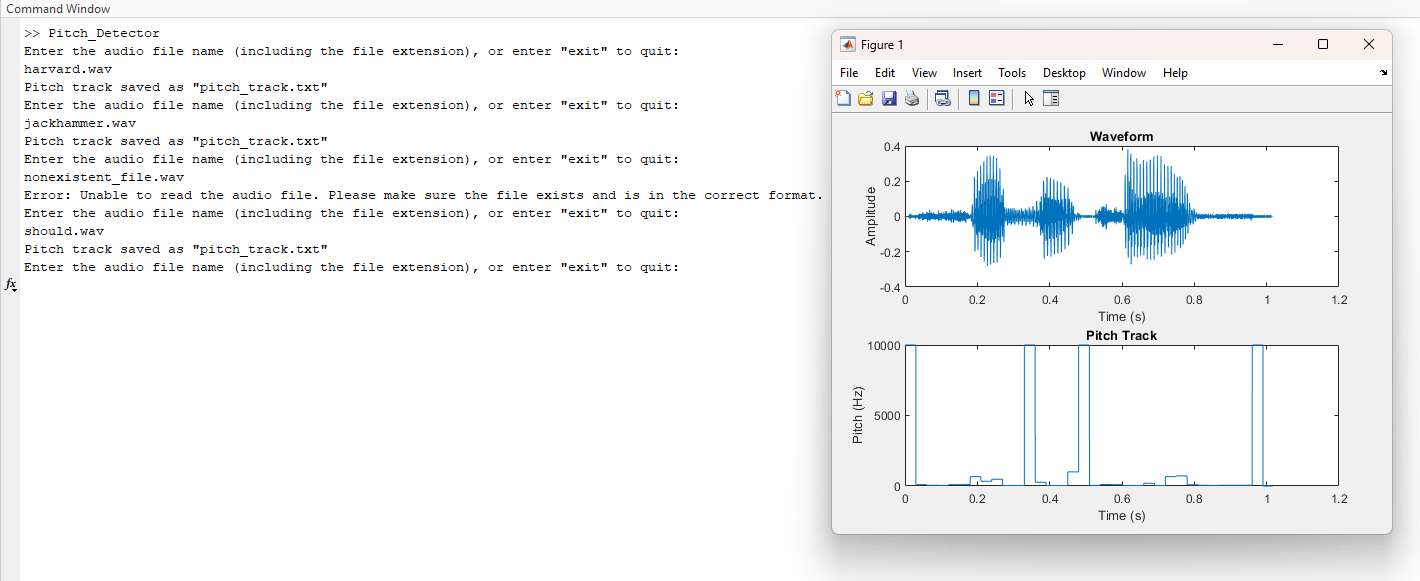


Figure 4: Pitch Detection of file "should.wav"

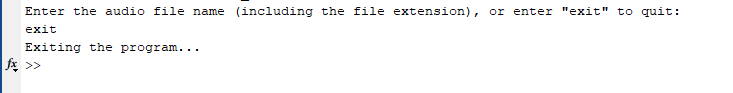


Figure 5: Exiting the pitch detector.

# Conclusion and Discussion

In conclusion, implementation of our pitch detection and analysis of audio signals project successfully works for pitch estimation and tracking in audio signals. We effectively utilized preprocessing techniques like normalization, low-pass filtering, noise reduction to enhance the quality of the given input audio signal. The results we obtained from the pitch detector implementation show its efficiency and accuracy when it comes to estimation and tracking the pitch of the input audio signals. The visualization of the waveform and pick track provided us with a valuable insight into the dynamics of the pitch. Due to external factors such as background noise and pitch variability, performance of our implementation can vary and may face challenges to accurate pitch estimation. Overall, our pitch detection project demonstrates successful implementation of pitch estimation and tracking. We believe we serve as a solid foundation for pitch detection systems and this project can be improved in the future for later projects and research.

# References

Important point: We gathered our example audio files for this project from several sources such as online datasets and publicly available audio files online.

[1] Roadmap for pitch estimation: A review of pitch estimation techniques. (2021).

[2] Smith, J. O., Abel, J. S., & Berners-Price, T. (2007). Spectral Audio Signal Processing. W3K Publishing.

[3] Wikipedia, Pitch Detection Algorithm, <https://en.wikipedia.org/wiki/Pitch_detection_algorithm>

[4] Center for Computer Research in Music and Acoustics, Pitch Detection Methods, <https://ccrma.stanford.edu/~pdelac/154/m154paper.htm>

[5] <http://research.google.com/audioset/>

[6] <https://www.signalogic.com/index.pl?page=speech_codec_wav_samples>

[7]<https://www.kaggle.com/datasets/pavanelisetty/sample-audio-files-for-speech-recognition?select=jackhammer.wav>