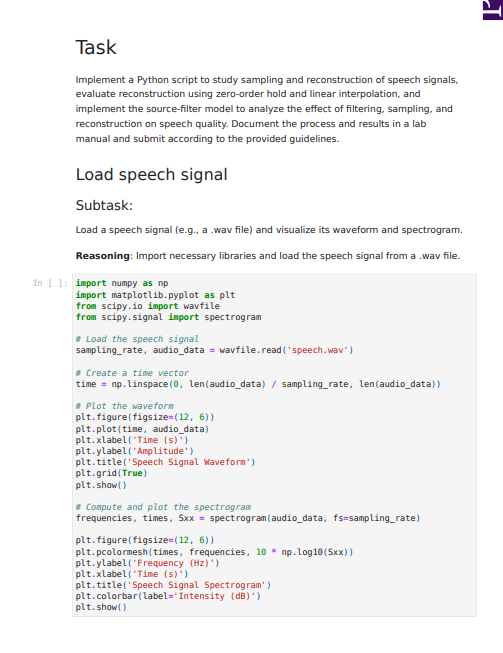
**Christ University**

**Name:** Joel Joseph Motha **Reg No:** 2448521

**Course:** SPR **Component:** Lab Manual

**Lab 1**

****

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**A screenshot of a document

AI-generated content may be incorrect.**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

A document with text on it

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Lab 2**

**A screenshot of a graph

AI-generated content may be incorrect.**

**A graph of a function

AI-generated content may be incorrect.**

**A screen shot of a graph

AI-generated content may be incorrect.**

**A graph of a diagram

AI-generated content may be incorrect.**

**A graph with a line drawn on it

AI-generated content may be incorrect.**

**A graph of a line graph

AI-generated content may be incorrect.**

**A screen shot of a graph

AI-generated content may be incorrect.**

**A graph of a graph of a graph

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Inference:**

This Python notebook analyses the frequency content of four different types of signals using the Fast Fourier Transform (FFT).

First, it generates and plots several basic signals in the time domain: a simple sine wave, a composite signal (two sine waves added together), a decaying exponential, and a rectangular pulse. For each of these, it then calculates and plots their frequency spectrum to show what "ingredients" (frequencies) they're made of.

The main point of the notebook is to demonstrate spectral leakage. This is a common issue in signal processing where the frequency analysis appears "blurry" because the signal is only observed for a short time. The final section clearly shows this problem using the composite signal and then demonstrates the solution: applying a Hamming window. The final plot compares the blurry spectrum with the much cleaner spectrum obtained after using the Hamming window.

**Lab 3**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Inference:**

* Both models are highly accurate in quiet environments with clear speech.
* Whisper is far more robust, successfully handling background noise where Google's API fails.
* Whisper better understands fast or soft speech, preventing critical command errors.
* Google is better at formatting data (e.g., "five" to "5"), while Whisper excels at inferring grammar.
* Whisper is the more reliable choice for accessibility due to its superior performance in varied, real-world conditions.

**Lab 4**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Inference:**

* The code successfully compares a 1.0-second sine wave (Signal 1) to a temporally shorter 0.7-second sine wave (Signal 2).
* The Dynamic Time Warping (DTW) algorithm finds the optimal, non-linear path to align the features of these two misaligned signals.
* The final Accumulated Cost Matrix shows the optimal path, visually demonstrating the non-linear warping required for alignment.
* Non-diagonal steps in the path are used to stretch the $0.7 \text{ s}$ signal, accommodating the $0.3 \text{ s}$ length difference.
* The DTW distance (e.g., $112.46$) is relatively low, indicating high structural similarity between the two sine wave patterns.
* The result confirms DTW's ability to measure shape similarity by neutralizing timing and duration differences.

**Lab 5**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a screen

AI-generated content may be incorrect.**

**A graph with lines and dots

AI-generated content may be incorrect.**

**Inference:**

* The LPC analysis on the high-sample rate audio failed due to a severe parameter mismatch.
* The 44100 Hz sampling rate and fixed LPC order of 18 caused the system to become unstable.
* This instability resulted in an explosive reconstructed signal and prevented the detection of any meaningful formant frequencies.
* The output correctly warned that the file must be downsampled to 16000 Hz for accurate analysis.

**Lab 6**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**A screenshot of a graph

AI-generated content may be incorrect.**

**A graph and diagram with numbers

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Inference:**

* The experiment compared two time series vectors exhibiting temporal misalignment and different lengths.
* The core DTW implementation successfully computed the full Accumulated Cost Matrix.
* The optimal warping path was determined, illustrating the non-linear alignment of the stretched vector.
* The DTW distance was finalized at approx. 1.73 indicating a low alignment cost.
* This low distance confirmed that the two signals were structurally highly similar despite their temporal differences.

**Lab 7**

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A graph of a wave

AI-generated content may be incorrect.

A screen shot of a graph

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer screen

AI-generated content may be incorrect.

A screen shot of a graph

AI-generated content may be incorrect.

**Inference:**

* The code implements Dynamic Time Warping (DTW) using both fastdtw and a custom algorithm to match signals of varying speeds.
* It processes synthetic "speech-like" sine waves by normalizing noisy, phase-shifted data for accurate feature comparison.
* The algorithm constructs an accumulated cost matrix to compute the optimal path that aligns sequences of differing lengths.
* Visualizations of the warping path and cost heatmap confirm the successful non-linear mapping of time indices.
* The results demonstrate DTW's effectiveness in speech recognition by quantifying high similarity despite temporal distortions.

**Lab 8**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Inference:**

* The notebook constructs a Hidden Markov Model (HMM) to represent speech phonemes as hidden states and acoustic features as observations.
* It utilizes defined Transition and Emission probability matrices to generate synthetic "Ground Truth" sequences for testing.
* The Viterbi algorithm is implemented to decode these noisy observations back into the most probable sequence of phonemes.
* The Forward algorithm computes the total likelihood of the observed sequence, providing a metric to evaluate how well the model fits the data.
* The results demonstrate the HMM's ability to recover hidden states, though high self-transition probabilities occasionally cause the decoder to linger on specific phonemes.

**Lab 9**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A graph showing the number of steps

AI-generated content may be incorrect.**

**Inference:**

* The Viterbi algorithm decodes the observation sequence [O1, O2, O3, O4] and correctly identifies the phoneme path ['/h/', '/e/', '/l/', '/o/'].
* This decoded path represents the maximum-likelihood sequence with an overall probability of 0.037044.
* At each step, the algorithm chooses the most probable transitions, efficiently discarding low-probability paths to avoid combinatorial explosion.
* The V-probability table shows a clear high-probability route from state S1 to state S4, forming the optimal phoneme sequence.
* The results highlight the effectiveness of Hidden Markov Models (HMMs) in extracting meaningful patterns from noisy acoustic data.
* Despite potential observation noise, the sequential phoneme structure of "hello" is accurately recovered, demonstrating the robustness of the model.