MAI 574- SPEECH PROCESSING AND RECOGNITION

Total Teaching Hours for Semester: 75 (3+4)
Max Marks:150 **Credits: 5**

Course Objectives

This course enables the learners to understand fundamentals of speech recognition, speech production and representation. It also enables the learners to impart knowledge on automatic speech recognition and pattern comparison techniques. This course helps the learners to develop automatic speech recognition model for different applications.

Course Outcomes

After successful completion of this course students will be able to

CO1: Understand the speech signals and represent the signal in time and frequency domain.
CO2: Analyze different signal processing and speech recognition methods.
CO3: Implement pattern comparison techniques and Hidden Markov Models (HMM)
CO4: Develop speech recognition system for real time problems.

Lab Exercises 0– Fundamentals of Signal Processing

Objective: To have basic knowledge of signals and understand the libraries for signal processing .

Learning outcomes (by the end of doing this lab the expectation is you will be able to….
LO1: Generate and visualize standard signals.
LO2: Apply and analyze sampling concepts, including aliasing.
LO3: Compare continuous and discrete signals.
LO4: Demonstrate time shifting and scaling of signals.
LO5: Combine and scale signals to observe superposition.
LO6: Simulate noise effects and apply filtering techniques.

Question:

(1) Generate and plot the following signals,
(a) A unit step function.
(b) A unit impulse function.
(c) A ramp function.
(d) An exponential signal (decaying and growing).
(e) A sinusoidal signal.
Write Python code to generate and plot each signal using matplotlib and numpy.

(2) You are asked to visualize the effects of sampling and reconstructing a continuous-time signal.

(a) Generate a continuous sinusoidal signal.

(b) Sample the signal at different rates (Nyquist rate, above, and below Nyquist).

(c) Reconstruct the sampled signal and observe the aliasing effect when undersampled.

(d) Plot the continuous signal, sampled points, and the reconstructed signal.

(3) Generate and plot a sinusoidal signal with amplitude = 1, frequency = 5 Hz, and duration = 1 second. Plot both the continuous and discrete versions of the signal.

(4) Write a Python program to demonstrate the effects of time shifting and time scaling on a signal.

(a) Generate a unit step function.

(b) Perform time shifting (delaying or advancing the signal).

(c) Perform time scaling (compressing or expanding the signal).

(d) Plot the original and transformed signals.

(5) Write a Python program to perform the following:

(a) Generate two sinusoidal signals with different frequencies and amplitudes.

(b) Add the signals together and plot the result.

(c) Scale one of the signals and observe the effect.

(6) Write a Python program to perform noise addition and filtering.

(a) Generate a clean sinusoidal signal.

(b) Add random Gaussian noise to the signal.

(c) Apply a low-pass filter and plot the filtered signal.

- Reference: Text book 1

  https://youtube.com/playlist?list=PLcumQJsBYq9GrRnMtxeif2EDtlKDAXZ1B&si=lU1KJIFK2CUSHVho

Deliverables: submit your colab file / Jupiter notebook in class room .Create a Git repo and add each SPR lab in it .

Lab 1

Lab Exercise I: Sampling and Reconstruction of Speech Signals

Aim

To study sampling and reconstruction of speech signals at different sampling rates, evaluate reconstruction using zero-order hold and linear interpolation, and implement the source-filter model to analyze the effect of filtering, sampling, and reconstruction on speech quality.

Question:

(1) Implement sampling and quantization techniques for the given speech signals.

(a) Plot the time domain representation of the original speech signal.

(b) Sample the speech signal at different sampling rates (e.g., 8kHz, 16kHz, and 44.1kHz).

(c) Plot sampled speech signal for each of these sampling rates.

(d) Using the sampled signals from above, reconstruct the signal using:

(i) Zero-order hold (nearest-neighbor interpolation)

(ii) Linear interpolation.

(e) Calculate the Mean Squared Error (MSE) between the original and the reconstructed signals for both methods.

Write an inference on how sampling rates affect the quality and accuracy of the reconstructed speech signal.

(2) Implement the source-filter model for a given speech signal and analyze the impact of sampling and reconstruction on the quality of the speech signal.

(a) Generate a synthetic speech signal using the source-filter model.

(i) Create a source signal (e.g., a glottal pulse train for voiced sounds or white noise for unvoiced sounds).

(ii) Apply a filter that models the vocal tract, represented by an all-pole filter or an FIR filter with formants (resonances of the vocal tract).

(b) Plot the generated speech signal and analyze the effect of the filter on the original source.

(c) Sample the speech signal generated above at different sampling rates (e.g., 8 kHz, 16 kHz, 44.1 kHz).

(d) Reconstruct the signal using a suitable interpolation method (e.g., zero-order hold, linear interpolation).

(e) Compute the Mean Squared Error (MSE) between the original and reconstructed speech signals.

Write an inference on tasks such as creating the source-filter model, different sampling rates, and reconstruction of the sampled signals.

Evaluation Rubrics:

(1) Implementation: 5 marks.

(2) Complexity and Validation: 3 marks.

(3) Documentation & Writing the inference: 2 marks.

Submission Guidelines:

- Make a copy of the lab manual template with your <name_reg:no_subject name >,
- Copy the given question and the answer (lab code) with results, followed by the conclusion of that lab. Title the lab as lab number.
- Keep updating your lab manual and show the lab manual of that particular lab for evaluation.
- Create a Git Repository in your profile <SPR lab-reg no> . Follow a different branch for each lab <Lab 1, Lab 2…>, and push the code to Git. The link should be provided in Google Classroom along with the PDF of the lab manual.
- Upload the PDF to Google Classroom before the deadline.

Lab Exercise 2:

Fourier Transform and Frequency Spectrum Analysis of Signals

Aim

To study the Fourier Transform and analyze the frequency spectrum of different signals (sinusoidal, composite, exponential, and rectangular). To compare their time-domain representation with their frequency-domain characteristics using both the Discrete-Time Fourier Transform (DTFT) and the Discrete Fourier Transform (DFT).

Questions

Question 1

(a) Generate a basic sinusoidal signal in the time domain. (For example, generate a sine wave with a frequency of 5 Hz, sampled at 1000 Hz.)
(b) Plot the time-domain waveform of the signal.
(c) Compute the Discrete-Time Fourier Transform (DTFT) and plot the continuous frequency spectrum.
(d) Compute the Discrete Fourier Transform (DFT) and plot the discrete frequency spectrum.

Question 2

(a) Generate a composite signal by adding two or more sinusoidal signals of different frequencies and amplitudes.
(b) Plot the time-domain waveform of the composite signal.
(c) Compute the Discrete-Time Fourier Transform (DTFT) and plot the continuous frequency spectrum.
(d) Compute the Discrete Fourier Transform (DFT) and plot the discrete frequency spectrum.

Question 3

(a) Generate an exponentially decaying signal.
(b) Plot the time-domain waveform.
(c) Compute the Discrete-Time Fourier Transform (DTFT) and plot the continuous frequency spectrum.
(d) Compute the Discrete Fourier Transform (DFT) and plot the discrete frequency spectrum.
(e) Analyze the relationship between the time-domain waveform and the frequency-domain representation.

Question 4

(a) Generate a **rectangular pulse signal** of finite duration in the time domain.
(b) Plot the time-domain waveform.
(c) Compute the Discrete-Time Fourier Transform (DTFT) and plot the continuous frequency spectrum.
(d) Compute the Discrete Fourier Transform (DFT) and plot the discrete frequency spectrum.
(e) Analyze the relationship between the time-domain waveform and the frequency-domain representation.

Evaluation Rubrics

1. **Implementation**: 5 marks

2. **Complexity and Validation**: 3 marks

3. **Documentation & Writing the inference**: 2 marks

Submission Guidelines

- Prepare a **single PDF** containing answers for all questions with proper plots, results, and inferences.

- Each answer must include **Python/Matlab code, output plots, and a conclusion**.

- Title the file as: **Lab_2_<YourName_RegNo>**.

- Upload the PDF to **Google Classroom** before the deadline.

- Maintain a **Git repository** with separate branches for each lab (Lab1, Lab2, …). Push your code and include the repo link in your submission.