LABORATORY MANUAL

Subject code: 221LEC001

Subject Name: SIGNAL PROCESSING LAB 1

2022-2023

DEPARTMENT OF ELECTRONICS AND COMMUNICATION
COLLEGE OF ENGINEERING TRIVANDRUM
Thiruvananthapuram

Course PO mapping

	221LEC001 Signal Processing Lab-1		PO1	PO2	PO3	PO4	PO5	PO6
CO1	Apply the principles of linear algebra and random processes in signal processing applications, analyze observations of experiments/simulations and infer.		0	0	3	0	0	3
CO2	Implement algorithms in machine learning.		0	0	3	0	0	3
CO3	Design and implement a real world applications of signal processing.		0	3	0	0	0	3
		Course PO mapping	0	3	3	0	0	3

Syllabus

No	Topics
1	**Linear Algebra**
1.1	Row Reduced Echelon Form: To reduce the given mxn matrix into Row reduced Echelon form
1.2	Gram-Schmidt Orthogonalization: To find orthogonal basis vectors for the given set of vectors. Also find orthonormal basis.
1.3	Least Squares Fit to a Sinusoidal function
1.4	Least Squares fit to a quadratic polynomial
1.5	Eigen Value Decomposition
1.6	Singular Value Decomposition
1.7	Karhunen- Loeve Transform
2	**Advanced DSP**
2.1	Sampling rate conversion: To implement Down sampler and Up sampler and study their characteristics
2.2	Two channel Quadrature Mirror Filterbank: Design and implement a two channel Quadrature Mirror Filterbank
3	**Random Processes**
3.1	To generate random variables having the following probability distributions (a) Bernoulli(b) Binomial(c) Geometric(d) Poisson(e)Uniform,(f) Gaussian(g)Exponential (h) Laplacian
3.2	Central Limit Theorem: To verify the sum of sufficiently large number of Uniformly distributed random variables is approximately Gaussian distributed and to estimate the probability density function of the random variable.
4	**Machine Learning**
4.1	Implementation of K Nearest Neighbours Algorithm with decision region plots
4.2	Implementation of K Means Algorithm with decision region plots
4.3	Implementation of Perceptron Learning Algorithms with decision region plots
4.4	Implementation of SVM algorithm for classification applications
5	**Implement a mini project pertaining to an application of Signal Processing in real life, make a presentation and submit a report**

Experiment 1:

Reduction of a Matrix to Row Reduced Echelon Form

Objective: The objective of this experiment is to familiarize students with the process of reducing a given m*n matrix to its row reduced echelon form (RREF).

Steps:

1. Initialize Variables:
   - Set variables for the number of rows (m) and columns (n) in thematrix.
   - Set lead to 0.
2. Loop Over Rows:
   - For each row in the matrix:
     - Check if lead is greater than or equal to n. If true, break theloop.
     - Find the pivot index (first non-zero entry) in the currentcolumn.
     - If a pivot is found:
       - Swap the current row with the row containing the pivot.
       - Scale the pivot row so that the pivot element is 1.
       - Eliminate non-zero entries above and below the pivotin the current column.
     - Move to the next column by incrementing lead.
3. Scale Row:
   - Implement a function to scale a row by a constant factor.
4. Add Scaled Row:
   - Implement a function to add a scaled row to another row.
5. Swap Rows:
   - Implement a function to swap two rows.
6. End:
   - The algorithm ends when the loop is completed for all rows or whenlead is greater than or equal to n.

Result: The row reduced Echelon form of the given m*n matrix is determined.

Experiment 2:

Gram-Schmidt Orthogonalization

Aim: To find orthonormal basis vectors for a given set of vectors using the Gram-Schmidt orthogonalization process.

Steps:

1. Initialize:
    - Set the number of vectors in the set: n
    - Initialize an array to store orthogonalized vectors: orthogonal_set
    - Initialize an array to store intermediate orthogonalized vectors:temp_set
2. Input Vectors:
    - Input the set of vectors: v1, v2, ..., vn
3. Gram-Schmidt Process:
    - For each vector vi in the set:
        - Set u_i equal to v_i
        - For each previously orthogonalized vector u_j (where $j < i$):
            - Calculate the projection of u_i onto u_j and subtract it from u_i
        - Normalize u_i (divide by its magnitude)
        - Add u_i to the orthogonal_set
        - Store u_i in temp_set
4. Output Orthogonal Set:
    - The orthogonal_set now contains the orthogonalized vectors.
5. End:
    - The process ends.

Result: The orthonormal basis vectors for the given set of vectors is obtained.

Experiment 3:

Least Squares Fit to a Sinusoidal Function

Aim:

To determine the least squares fit to a sinusoidal function for a given set of data points.

Procedure:

- Define a function that takes an array of data points (x, y) and a guess for the sinusoidal function parameters (amplitude, frequency, phase shift)

- Calculate the residuals (differences between the predicted y-values from the sinusoidal function and the actual y-values)
- Calculate the Jacobian matrix (matrix of partial derivatives of the residuals with respect to the sinusoidal function parameters)
- Use an optimization algorithm (e.g., gradient descent, Newton's method) to minimize the sum of squared residuals by iteratively updating the sinusoidal function parameters
- Return the updated sinusoidal function parameters as the least square fit to the data points

Conclusion: Least square fit to a sinusoidal function is obtained.

<u>Experiment 4:</u>

Least Squares Fit to a Quadratic Polynomial

Aim:

To determine the least squares fit to a quadratic polynomial for a given set of data points.

Procedure:

- Create a matrix containing the x-values and squared x-values.
- Solve the least squares problem.
- Plot the fitted quadratic polynomial.

Conclusion: Least square fit to a quadratic polynomial is obtained**.**

<u>Experiment 5:</u>

Eigenvalue Decomposition

Aim:

- Import the necessary libraries.
- Define a function to perform eigenvalue decomposition.
- Call the eigenvalue decomposition function.
- Print the eigenvalues and eigenvectors.

Conclusion: Eigen values of the given m*n matrix is determined.

RANDOM PROCESSES

Experiment 1:

Random Variables for Probability Distribution

Aim: To generate random variables following various probability distributions, including Bernoulli, Geometric, Poisson, Exponential, Uniform, and Binomial.

Steps:

- Set seed for reproducibility
- Number of random variables to generate
- Bernoulli Distribution (p=0.5, one trial)
- Binomial Distribution (n=5, p=0.3, five trials)
- Geometric Distribution (p=0.2, probability of success)
- Poisson Distribution (lambda=3, average rate)
- Uniform Distribution (low=0, high=1)
- Exponential Distribution (beta=0.5, inverse of the rate parameter)
- Plotting histograms for each distribution

Conclusion: Generated random variables having the following probability distributions Bernoulli, Binomial, Geometric, Poisson, Uniform, Exponential.

Experiment 2:

Central Limit Theorem

Aim: To verify the Central Limit Theorem by demonstrating that the sum of a sufficiently large number of uniformly distributed random variables is approximately Gaussian distributed. Additionally, estimate the probability density function of the random variable.

Procedure:

- Parameters for the original distribution, size of each sample, number of samples to generate
- Generate samples from the original distribution (e.g., uniform, exponential, etc.)
- Calculate the means of each sample
- Plot the histogram of the sample means
- Plot the theoretical normal distribution based on the Central Limit Theorem

Conclusion: Verified that the sum of sufficiently large number of Uniformly distributed random variables is approximately Gaussian distributed and estimated the probability density function of the random variable.

ADVANCED DSP

Experiment 1:

Sampling Rate Conversion

Aim: To implement a Down Sampler and Up Sampler and study their characteristics in the context of sampling rate conversion.

Procedure:

- Create a sample signal (e.g., a sine wave)
- Down Sample the signal
- Up Sample the down-sampled signal
- Plot the original, down-sampled, and up-sampled signals

Conclusion: Implemented Down sampler and Up sampler and studied their characteristics.

Experiment 2:

Two-Channel Quadrature Mirror Filterbank

Aim: To design and implement a two-channel Quadrature Mirror Filterbank (QMF).

Procedure:

- Design the analysis filters (e.g., simple half-band filters)
- Apply the analysis filters
- Design the synthesis filters (mirror of the analysis filters)
- Apply the synthesis filters
- Create a sample signal
- Apply the QMF analysis
- Apply the QMF synthesis
- Plot the original and reconstructed signals

Conclusion: Designed and implemented a two channel Quadrature Mirror Filterbank.

MACHINE LEARNING

EXPERIMENT NO:1

K MEANS ALGORITHM

AIM : To Implement K Means Algorithm with decision region plots.

- Generate synthetic data
- Apply K-Means algorithm
- Get cluster centers and labels
- Plot the original data and cluster centers
- Plot original data points
- Plot cluster centers

RESULT: Implemented K Means Algorithm with decision region plots.

EXPERIMENT NO:2

PERCEPTRON LEARNING

AIM: To implement Perceptron Learning Algorithms with decision region plots.

STEPS:

- Generate synthetic data
- Apply Perceptron learning algorithm
- Make predictions on the same data for visualization
- Plot the decision region
- Plot data points
- Plot decision boundary
- Highlight misclassified points
- Evaluate accuracy on the training set

RESULT: Implemented perceptron learning algorithm with decision region plots.

EXPERIMENT NO:3

SUPPORT VECTOR MACHINE

AIM : To implement SVM algorithm for classification applications.

STEPS:

- Generate synthetic data for binary classification
- Split the data into training and testing sets
- Apply SVM for classification
- Make predictions on the test set
- Plot the decision boundary
- Plot decision boundary on training set
- Evaluate accuracy on the test set

RESULT: Implemented SVM algorithm with decision region plots.

EXPERIMENT NO:4

K NEAREST NEIGHBOUR

AIM: To implement K Nearest Neighbours Algorithm with decision region plots.

STEPS:

- Generate synthetic data for binary classification
- Split the data into training and testing sets
- Apply KNN for classification
- Make predictions on the test set
- Plot the decision boundary
- Plot decision boundary on training set

- Evaluate accuracy on the test set

RESULT: Implemented K Nearest Neighbours Algorithm with decision region plots.