**  
Digital Signal Processing Lab**

Adamson University Computer Engineering Department

**Laboratory Exercise 2**Vector and Matrices using Numpy

Submitted by:

**Group #**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **SO** | **PI** | **Category** | **Exceptional  4** | **Acceptable  3** | **Marginal 2** | **Unacceptable  1** | **Score** |
| b | 1 | **Compliance**  30% | All procedures were followed, the output is as expected, additional related functionalities were augmented. | All procedures were followed and the output is as expected. | All procedures were followed but the output is not as expected. | Did not follow the set procedures |  |
| b | 1 | **Analysis**  20% | Data interpretation is professionally written with appropriate and clear illustrations. | Data is clearly and correctly explained with proper illustrations. | Data is not clearly explained, has minor flaws, or no illustrations | There is no data explanation about the data or results. |  |
| b | 1 | **Validity** 20% | The implementation uses the concepts and principles of the experiment as well as advanced topics. | The implementation uses the concepts and principles of the theory discussed for the experiment. | Implementation did not clearly express the use of theory discussed for the experiment. | There is no implementation. |  |
| b | 1 | **Interpretation**  20% | The conclusion is professionally written and points the theories in the experiment and its implications in engineering. | The conclusion points to the main ideas and applications of the theory in the experiment. | The conclusion does not point out the main ideas and applications of the theory in the experiment. | There is no conclusion. |  |
|  |  | **Format and Clarity** 10% | Follows the prescribed format, observes proper and technical grammar, and observes proper citation and referencing according to IEEE journal standards. | Follows the prescribed format, observes proper and technical grammar, and observes proper IEEE citation and referencing. | Did not follow the prescribed format, has poor grammar, or incorrect citations and references scheme. | Did not follow the prescribed format, has poor grammar, and has no citations and references. |  |
|  |  | **TOTAL SCORE** | | | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Group Members** | | | |
| **STUDENT NUMBER** | **NAME** | **CONTRIBUTION** | **SCORE** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Submitted to:

Engr. Zaldy Enrico Louis A. Mejia

Date:

mm/dd/year

**OBJECTIVES**

* This laboratory exercise aims to help students in reviewing their python programming for Vector and Matrices using Numpy

**DISCUSSION**

Signals are naturally time-sampled values collected from nature or from any physical entity or observation in the real world. Linear Algebra posits a representation scheme for easy computation, manipulation, and modeling for simple to complex signal processes. In this notebook, we try to understand how linear algebra works as a mathematical tool and a computational tool for analysis. Aside from the mathematical concepts for linearity, vectors, and linear transformation, we will also look at how we could model such mathematics in code.

An advantage of using software tools for linear algebra or with any applied field of mathematics is that it abstracts the processes or hastens computations so that engineers or scientists can interpret results faster. The use of software-based mathematics could also help develop better intuition in how math works and how it reacts to certain stimuli or as part of a system. In this notebook, we will use Python, specifically the numpy library for computation and matplotlib for visualization.

A screenshot of a computer

Description automatically generated

A graph of a function

Description automatically generatedA screenshot of a math problem

Description automatically generatedA screenshot of a computer program

Description automatically generated

A graph with colored lines and arrows

Description automatically generatedA screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

A graph and line graph

Description automatically generated

A screenshot of a math problem

Description automatically generated

A white rectangular object with a black border

Description automatically generated

**MATERIALS**

**Software:**

* Python
* Anaconda
* Jupyter Notebook

**PROCEDURES**

**Solve the problem sets given below manually and by using python programming.** (Show your complete solution)

1. Solve the system using elimination method in matrix form

x + y + z + w = 10  
9x – 6y + 3z + 4w = 8  
−3x + 2y − 2z + 5w = −3  
x – y + 4z − w = 5

1. A diagram of a circuit

   Description automatically generated
2. Find the inverse of matrix A:

A =

1. DSP Concept review

For a DSP review, a vector will be used to represent and manipulate a simple signal. A digital signal is a series of number that measures a quantity (e.g loudness of a sound, brightness of a picture, etc.). In DSP, we treat these sequences of numbers as vectors. Linear algebra allows us to analyze and change the signals.

For simulation, the code below will use vector addition to simulate a distorted or noisy signal. In real life, audio recordings are never perfect. They almost always contain unwanted sounds. We can represent both the desired signal and unwanted signal noise as vectors and do vector addition

Inner product operation will measure the similarity between two vectors. This will allow us to "filter" out the noise and recover our original signal.

Run the python code below:

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  t = np.linspace(0, 1, 100) # Time vector  clean\_signal = np.sin(2 \* np.pi \* 5 \* t)  noise = 0.5 \* np.random.randn(100)  noisy\_signal = clean\_signal + noise  plt.figure(figsize=(10, 6))  plt.subplot(3, 1, 1)  plt.plot(t, clean\_signal, label='Clean Signal')  plt.title('Clean Signal')  plt.legend()  plt.subplot(3, 1, 2)  plt.plot(t, noisy\_signal, color='orange', label='Noisy Signal')  plt.title('Noisy Signal')  plt.legend()  correlation\_scalar = np.dot(noisy\_signal, clean\_signal) / np.dot(clean\_signal, clean\_signal)  recovered\_signal = correlation\_scalar \* clean\_signal  plt.subplot(3, 1, 3)  plt.plot(t, recovered\_signal, color='green', label='Recovered Signal')  plt.plot(t, clean\_signal, '--', color='red', label='Original Clean Signal')  plt.title('Recovered Signal vs Original Clean Signal')  plt.legend()  plt.tight\_layout()  plt.show() |

Discuss the results of the code and the functions used and answer the following:

* How can vectors represent a real-world signal like an audio recording?
* How does the inner product "filter" out noise and help you recover the original signal?

**CONCLUSION**