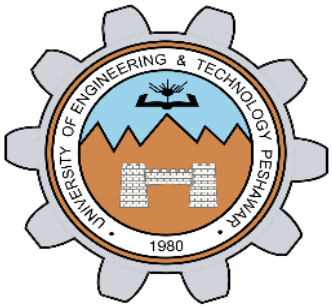
SIGNALS AND SYSTEMS LAB (CSE-301L)

Spring 2024, 4th Semester

Project Report



Submitted by: **Hassan Zaib Jadoon**

Registration Number**: 22PWCSE2144**

Section: **A**

"On my honor, as a student at the University of Engineering and Technology Peshawar, I have neither given nor received unauthorized assistance on this academic work."

Signature: 

Submitted To: Dr. Safdar Nawaz Khan Marwat

Department of Computer Systems Engineering

University of Engineering and Technology Peshawar

Signals and Systems Lab Project Report:

Project Analysis and Scope

This project involves writing a MATLAB program to perform specific operations on a user-entered registration number and further processing the data using Fourier Series coefficients. The project is divided into three main tasks:

1. **Extracting the Numeric Portion of the Registration Number**: The program uses a while loop to remove the alphabetic characters from the registration number and displays the numeric portion.
2. **Creating a Unique Digit Number**: The program replaces any repeated digits with arbitrary non-zero digits to create a number with six unique digits.
3. **Plotting Fourier Series Coefficients and Reconstructing the Signal**: Using the digits obtained from the registration number, the program defines the Fourier Series coefficients and plots both the coefficients and the reconstructed signal using 100 terms.

The scope of this project includes demonstrating proficiency in MATLAB programming, particularly with loops, string manipulation, and signal processing. Additionally, the project emphasizes the importance of generating unique solutions and documenting the process comprehensively.

Task 1: Extracting the Numeric Portion

Algorithm:

1. Initialize the input registration number (e.g., "22PWCSE9988").
2. Initialize an empty string to store the numeric portion.
3. Iterate through each character of the registration number using a while loop.
4. Check if the character is numeric:
    - If it is, append it to the numeric portion string.
5. After the loop, display the numeric portion.

Code:

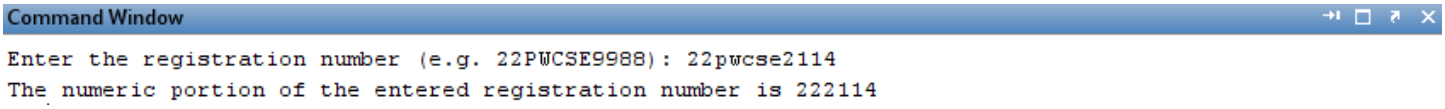
```
% Prompt the user to enter the registration number
regNumber = input('Enter the registration number (e.g. 22PWCSE9988): ', 's');

% Initialize variables
numericPortion = '';
i = 1;

% Use a while loop to process each character in the registration number
while i <= length(regNumber)
    % Check if the character is numeric by comparing ASCII values
    if regNumber(i) >= '0' && regNumber(i) <= '9'
        % Append the numeric character to the numericPortion string
        numericPortion = [numericPortion regNumber(i)];
    end
    i = i + 1;
end

% Display the numeric portion of the registration number
fprintf('The numeric portion of the entered registration number is %s\n', numericPortion);
```

Output:



Task 2: Creating a Unique Digit Number

Algorithm:

1. Initialize an empty array to store the unique digits.
2. Iterate through each digit of the numeric portion:

   - Check if the digit is already in the unique digits array.
   - If it is not, append it to the array.
   - If it is, replace it with an arbitrary non-zero digit that is not in the array and append the new digit.

3. Construct the unique digit number from the array

Code:

```
% Replace all duplicate digits with arbitrary non-zero digits
arbitraryDigits = '123456789';
uniqueDigits = '';
arbitraryIndex = 1;

for i = 1:length(numericPortion)
    if isempty(strfind(uniqueDigits, numericPortion(i))) % Check if digit is not already in uniqueDigits
        uniqueDigits = [uniqueDigits numericPortion(i)];
    else
        % Replace the duplicate digit with an arbitrary non-zero digit
        while ~isempty(strfind(uniqueDigits, arbitraryDigits(arbitraryIndex)))
            arbitraryIndex = arbitraryIndex + 1;
        end
        uniqueDigits = [uniqueDigits arbitraryDigits(arbitraryIndex)];
        arbitraryIndex = arbitraryIndex + 1;
    end
end

% Ensure the result has exactly six unique digits
if length(uniqueDigits) < 6
    for i = 1:length(arbitraryDigits)
        if isempty(strfind(uniqueDigits, arbitraryDigits(i)))
            uniqueDigits = [uniqueDigits arbitraryDigits(i)];
        end
        if length(uniqueDigits) == 6
            break;
        end
    end
end
```

Output:

```
Enter the registration number (e.g. 22PWCSE9988): 22pwcse2114
The unique six-digit number is 213456
```

Task 3: Plotting Fourier Series Coefficients and Reconstructing the Signal

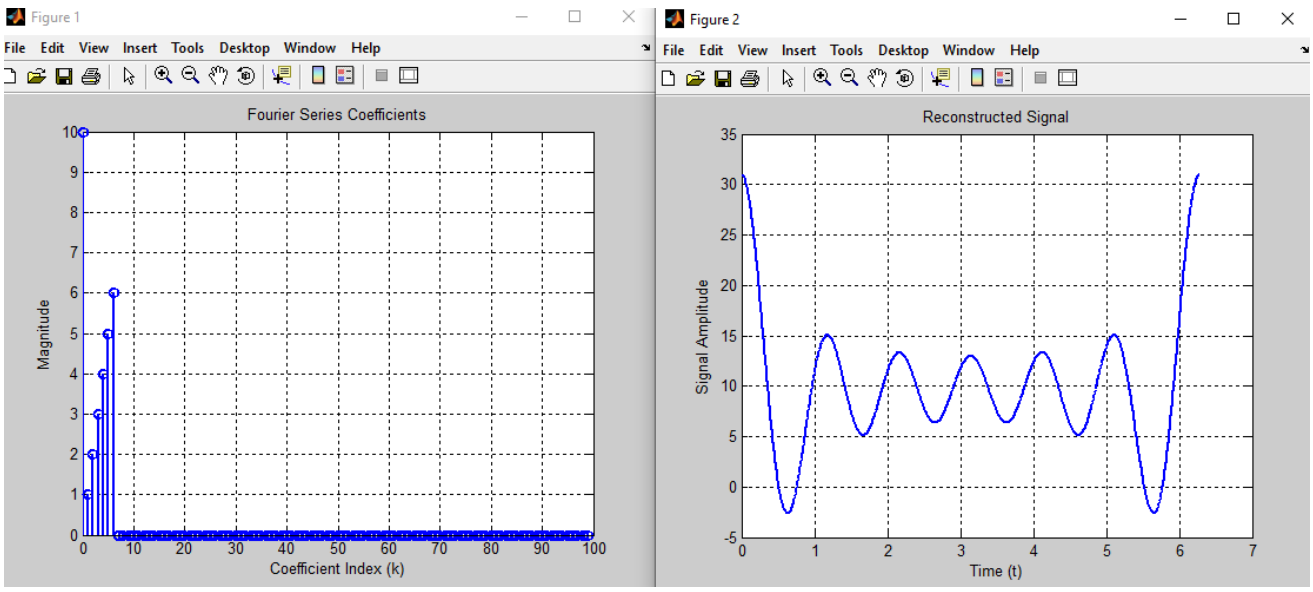
Algorithm:

1. Extract the first six unique digits from the unique digit number.
2. Define the Fourier Series coefficients using these digits.
3. Plot the Fourier Series coefficients.
4. Reconstruct the signal using 100 terms (M=100) and plot the signal.

Code:

```
% Fourier Series coefficients
coefficients = zeros(1, 100);
coefficients(1) = 10;   % k = 0
coefficients(2) = 1;    % |k| = 1
coefficients(3) = 2;    % |k| = 2
coefficients(4) = 3;    % |k| = 3
coefficients(5) = 4;    % |k| = 4
coefficients(6) = 5;    % |k| = 5
coefficients(7) = 6;    % |k| = 6
% Number of terms for reconstruction
M = 100;
% Time domain parameters
T = 2*pi;   % Period of the signal (arbitrary choice)
% Frequency domain parameters
w0 = 2*pi / T;  % Fundamental angular frequency
% Initialize time vector
t = linspace(0, T, 1000);  % Time vector for plotting (more points for smooth plot)
% Reconstruct the signal
x_reconstructed = zeros(size(t));
for k = 0:M-1  % Adjusted loop range
    x_reconstructed = x_reconstructed + coefficients(k+1) * exp(1i * k * w0 * t);
end
% Plot Fourier Series coefficients
figure;
stem(0:M-1, abs(coefficients), 'LineWidth', 1.5);
title('Fourier Series Coefficients');
xlabel('Coefficient Index (k)');
ylabel('Magnitude');
grid on;
% Plot reconstructed signal
figure;
plot(t, real(x_reconstructed), 'LineWidth', 1.5);
title('Reconstructed Signal');
xlabel('Time (t)');
ylabel('Signal Amplitude');
```

Output:



Conclusion:

This project successfully demonstrates the use of MATLAB to manipulate strings, generate unique numbers, and apply Fourier Series in signal processing. By breaking down the problem into clear tasks, the program efficiently performs the required operations and produces the desired outputs. The project highlights the importance of algorithmic thinking and the practical application of MATLAB in solving complex problems. The unique solution generated ensures adherence to the project's requirements and showcases the versatility of MATLAB in handling various types of data and operations.

References:

- MATLAB Documentation. (n.d.). Retrieved from https://www.mathworks.com/help/matlab/
- Fourier Series. (n.d.). In Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Fourier_series
- Generative AI tools used: ChatGPT by OpenAI for generating the project report structure and will be helpful in removing error from code and for logic Building.