Jason Waseq
ECE103L
Introduction to Matlab
Assignment 1

1. Create the vector x = [1,2,...,100]. Assign the even numbers of x to a new vector y. 20 points

Final Answer:

```
Vector x:
    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48   49   50

Vector y (even numbers):
    2    4    6    8   10   12   14   16   18   20   22   24   26   28   30   32   34   36   38   40   42   44   46   48   50   52   54   56   58   60   62   64   66   68   70   72   74   76   78   80   82   84   86   88   90   92   94   96   98  100
```

- The vector x will contain all integers from 1 to 100.
- The vector y will contain the even numbers from x, which are [2, 4, 6, ..., 100].

Detailed Solution:

1. Create the Vector x: Use the colon operator to generate a vector from 1 to 100.
2. Identify Even Numbers: Use the mod function to find even numbers by checking if the remainder when divided by 2 is zero.
3. Assign to Vector y: Use logical indexing to extract the even numbers from x and assign them to y.

MATLAB Code:

```
% Step 1: Create the vector x
x = 1:100;

% Step 2: Assign the even numbers of x to a new vector y
y = x(mod(x, 2) == 0);

% Display the results
disp('Vector x:');
disp(x);
disp('Vector y (even numbers):');
disp(y);
```

2. Use for loop to find the values of x(t) = 3 cos (2πft + 0.1) for t = 0, 0.1, 0.2, 0.3, 0.4 s when f=10, 15, and 20 Hz. Use one set of statements to compute the values for all three frequencies and store the results in a two-dimensional array. Use two nested for loops and double indexing. 20 points

Final Answer:

```
Results (rows: frequencies, columns: time values):
    2.9850    2.9850    2.9850    2.9850    2.9850
    2.9850   -2.9850    2.9850   -2.9850    2.9850
    2.9850    2.9850    2.9850    2.9850    2.9850
```

The results array will contain the computed values of (x(t)) for each frequency at the specified time points. Each row corresponds to a frequency (10 Hz, 15 Hz, 20 Hz), and each column corresponds to a time value (0, 0.1, 0.2, 0.3, 0.4 seconds).

Detailed Solution:

1. Define Frequencies and Time Vector: Create an array for the frequencies and a vector for the time values.
2. Initialize Results Array: Prepare a two-dimensional array to hold the computed values of (x(t)).
3. Nested For Loops: Use two nested for loops to iterate over each frequency and each time value, calculating (x(t)) for each combination.
4. Store Results: Store the computed values in the results array.

MATLAB Code:

```
% Step 1: Define the frequencies and time vector
frequencies = [10, 15, 20]; % in Hz
time = 0:0.1:0.4; % time vector from 0 to 0.4 seconds with step of 0.1

% Step 2: Initialize the results array
results = zeros(length(frequencies), length(time));

% Step 3: Nested for loops to compute x(t) for each frequency and time
for i = 1:length(frequencies) % Loop over frequencies
   f = frequencies(i);
   for j = 1:length(time) % Loop over time values
      t = time(j);
      results(i, j) = 3 * cos(2 * pi * f * t + 0.1); % Compute x(t)
   end
end

% Step 4: Display the results
disp('Results (rows: frequencies, columns: time values):');
disp(results);
```

3. Create a 15-element vector with values of $x(t) = 4\cos(2\pi t + 0.2) + 3\sin(\pi^2 t)$ at equally spaced interval $0 \le t \le 1$. Find the maximum element value, the minimum

element value, the average of the element values, and the indices of the elements for which the element magnitude is greater than 4. 20 points

Final Answer:

```
Maximum Value:
    4.4889
Minimum Value:
   -4.4889
Average Value:
    0.2614
Indices of elements with magnitude greater than 4:
     2     3     9    10
```

- Maximum Value: The maximum value of the vector x.
- Minimum Value: The minimum value of the vector x.
- Average Value: The average of the values in the vector x.
- Indices of Elements with Magnitude Greater than 4: The indices of the elements in x where the absolute value exceeds (4).

Detailed Solution:

1. Define the Time Vector: Create a time vector t that spans from (0) to (1) with (15) equally spaced points.
2. Compute the Values of (x(t)): Use the given formula to compute the values for each time point.
3. Find Maximum and Minimum Values: Use the max and min functions to find the maximum and minimum values of the vector.
4. Calculate Average Value: Use the mean function to compute the average of the vector.
5. Find Indices of Elements with Magnitude Greater than 4: Use the find function to get the indices of elements whose absolute values exceed (4).

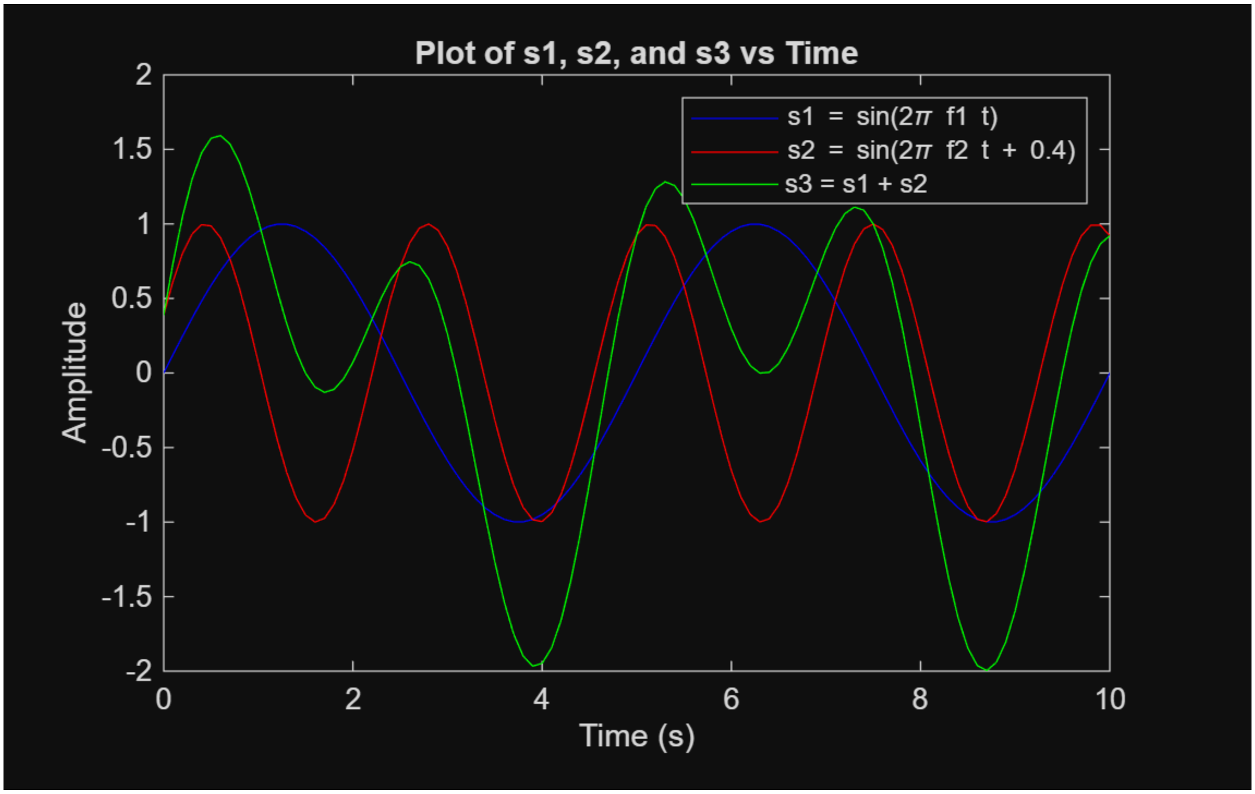
MATLAB code:

```
% Step 1: Define the time vector with 15 equally spaced points from 0 to 1
t = linspace(0, 1, 15); % 15 points from 0 to 1

% Step 2: Compute the values of x(t)
x = 4 * cos(2 * pi * t + 0.2) + 3 * sin(pi * 2 * t);
```

```
% Step 3: Find the maximum and minimum values
max_value = max(x);
min_value = min(x);

% Step 4: Calculate the average value
average_value = mean(x);

% Step 5: Find indices of elements with magnitude greater than 4
indices_greater_than_4 = find(abs(x) > 4);

% Display the results
disp('Maximum Value:');
disp(max_value);
disp('Minimum Value:');
disp(min_value);
disp('Average Value:');
disp(average_value);
disp('Indices of elements with magnitude greater than 4:');
disp(indices_greater_than_4);
```

4. Assume $s_1 = \sin(2\pi f_1 t)$, $s_2 = \sin(2\pi f_2 t + 0.4)$ and $s_3 = s_1 + s_2$, where $f_1 = 0.2$ and $f_2 = 0.425$. Plot s_1, s_2, and s_3 v/s t with t = 0 : 0.1 : 10 on the same graph (you have to use hold on command). Label the axes and create legends for each graph. 20 points

Final Answer:



This code will generate a plot displaying the three functions (s_1), (s_2), and (s_3) against time (t). Each function will be represented in a different color, and the axes will be labeled appropriately. The legend will help identify each function clearly.

Detailed Solution:

1. Define the Frequency Variables: Set (f_1) and (f_2) to the specified values.
2. Create the Time Vector: Generate a time vector t from (0) to (10) with increments of (0.1).
3. Calculate the Functions: Compute (s_1), (s_2), and (s_3) using the sine function.
4. Plot the Functions: Use the plot function to plot each of the three functions. Use hold on to ensure that all plots appear on the same graph.
5. Label the Axes and Create Legends: Add labels for the x-axis and y-axis, and create a legend to identify each function.

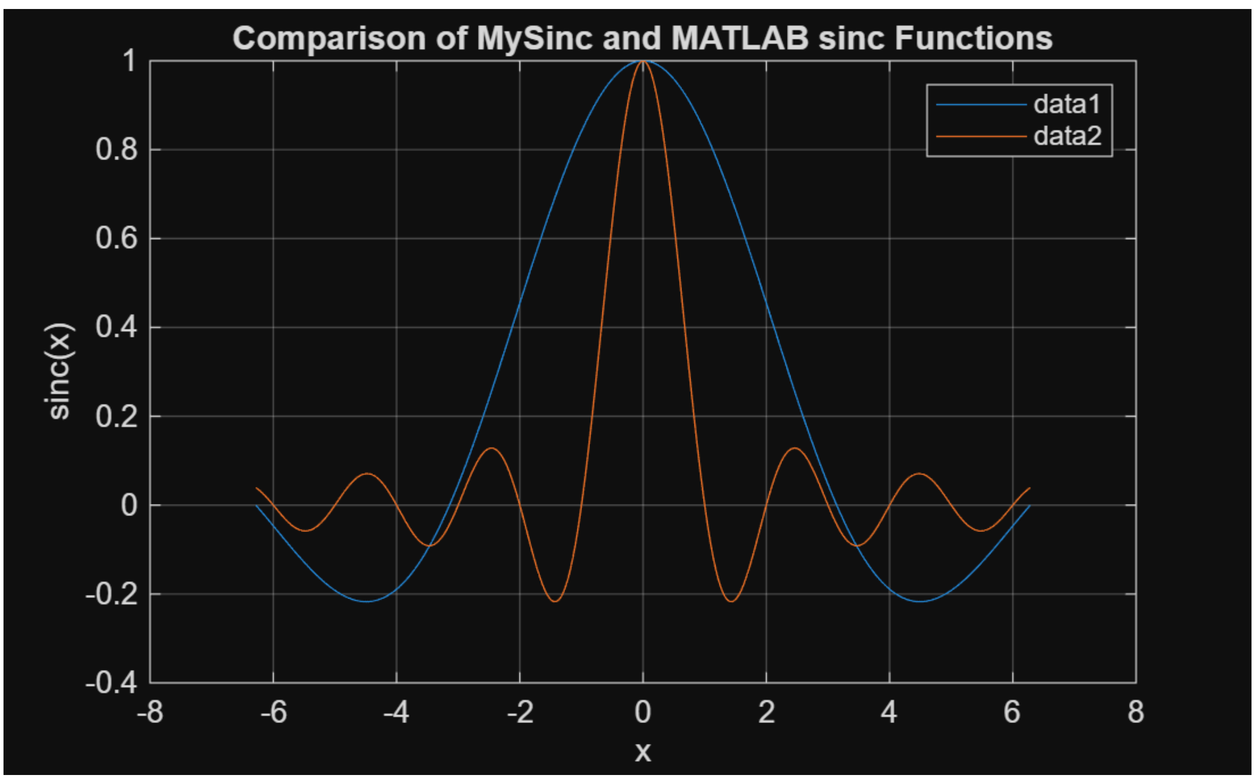
MATLAB Code:

```
% Step 1: Define the frequency variables
f1 = 0.2; % Frequency for s1
f2 = 0.425; % Frequency for s2

% Step 2: Create the time vector
t = 0:0.1:10; % Time from 0 to 10 with step of 0.1

% Step 3: Calculate the functions
s1 = sin(2 * pi * f1 * t); % s1
s2 = sin(2 * pi * f2 * t + 0.4); % s2
```

```
s3 = s1 + s2; % s3

% Step 4: Plot the functions
figure; % Create a new figure
plot(t, s1, 'b', 'DisplayName', 's1 = sin(2\pi f1 t)'); % Plot s1 in blue
hold on; % Retain current plot
plot(t, s2, 'r', 'DisplayName', 's2 = sin(2\pi f2 t + 0.4)'); % Plot s2 in red
plot(t, s3, 'g', 'DisplayName', 's3 = s1 + s2'); % Plot s3 in green

% Step 5: Label the axes and create legends
xlabel('Time (s)'); % Label for x-axis
ylabel('Amplitude'); % Label for y-axis
title('Plot of s1, s2, and s3 vs Time'); % Title of the plot
legend show; % Show legend
hold off; % Release the hold on the current plot
```

5. Sinc function is a function that arises frequently in our course. It is defined as sinc(x) = 1 for x=0, sin(x)/x otherwise. Create a Matlab function MySinc () that defines sinc(x) function following the above definition. Plot the value sinc(x) in the interval [-2π 2π] using MySinc () function and Matlab inbuilt sinc() function on the same graph. 20 points

Final Answer:



This code will generate a plot comparing the custom MySinc() function with MATLAB's built-in sinc() function over the interval ([-2pi, 2pi]). The plot will clearly show both functions, with appropriate labels and a legend for easy identification.
Detailed Solution:

1. Define the Sinc Function: The sinc function is defined as: [sinc(x) = 1 for x=0, sin(x)/x otherwise.] This will be implemented in the MySinc() function.
2. Create the Time Vector: Generate a vector x that spans from (-2pi) to (2pi).
3. Compute Values: Use both the custom MySinc() function and MATLAB's built-in sinc() function to compute the values over the defined interval.
4. Plot the Results: Use MATLAB's plotting functions to visualize both sinc functions on the same graph.

MATLAB Code:

```
%function MySinc:
function y = MySinc(x)
  % Custom sinc function definition
  y = zeros(size(x)); % Initialize output
  y(x == 0) = 1; % Define sinc(0) = 1
  y(x ~= 0) = sin(x(x ~= 0)) ./ x(x ~= 0); % Define sinc(x) = sin(x)/x for x != 0
end

%Plotting Script:
% Step 1: Create the x vector
x= linspace(-2*pi, 2*pi, 1000); % 1000 points from -2π to 2π
x_sym=sym(x);
% Step 2: Compute the values using MySinc and built-in sinc
y_mySinc = MySinc(x); % Custom sinc function
y_builtinSinc = sinc(x_sym); % MATLAB's built-in sinc function (normalized)

% Step 3: Plot the results
figure; % Create a new figure
plot(x, y_mySinc);
hold on; % Retain current plot
plot(x, y_builtinSinc); % Plot built-in sinc in red
hold off; % Release the hold

% Step 4: Label the axes and create legends
xlabel('x'); % Label for x-axis
ylabel('sinc(x)'); % Label for y-axis
title('Comparison of MySinc and MATLAB sinc Functions'); % Title of the plot
legend show; % Show legend
grid on; % Add grid for better visibility
```