Image and Video Processing Lab

Name: - Yash Rajput

Registration Number : 211060042 **Date: -** 09.02.2024

Experiment 1
Custom Discrete Time Signal

Aim: - To make Matlab program that generates custom discrete-time signals based on user inputs through the console. The program should prompt the user to input parameters such as the number of data points, the starting point of the signal on the axis, and the values of the signal.

Software Used: - MATLAB

Code:

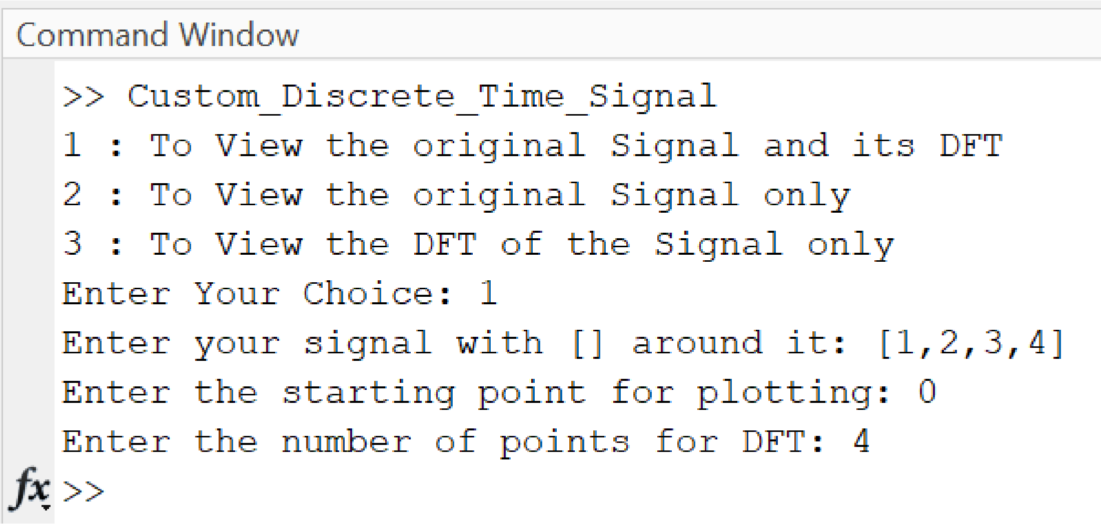
```
% IVP LAB - Experiment No-1 - CUSTOM DISCRETE TIME SIGNAL
% Yash Rajput - TY EC - 211060042

input_signal = input("1 : To View the original Signal and its DFT\n2 : To View the
original Signal only\n3 : To View the DFT of the Signal only\nEnter Your Choice: ");


if input_signal == 1
    signal_values = input("Enter your signal with [] around it: ");
    start_x = input("Enter the starting point for plotting: ");
    N = input("Enter the number of points for DFT: ");

    subplot(3,1,1);
    plot_signal(signal_values, start_x); % Call the plot_signal function to plot the input
sequence
    title('Original Signal'); % Add title to the subplot
    xlabel('Data points'); % Label the x-axis
    ylabel('Values'); % Label the y-axis
    grid on; % Display grid lines

    % Compute the Discrete Fourier Transform (DFT) of the input sequence
    dft_result = dft(signal_values, N);

    % Plot the magnitude of the DFT sequence
    subplot(3,1,2);
    plot_signal(abs(dft_result), start_x); % Plot the magnitude of the DFT sequence
    title('Magnitude of DFT Sequence'); % Add title to the subplot
    xlabel('Frequency (k)'); % Label the x-axis
    ylabel('Magnitude'); % Label the y-axis
    grid on; % Display grid lines

    % Plot the phase of the DFT sequence
    subplot(3,1,3);
    plot_signal(angle(dft_result), start_x); % Plot the phase of the DFT sequence
    title('Phase of DFT Sequence'); % Add title to the subplot
    xlabel('Frequency (k)'); % Label the x-axis
    ylabel('Phase in Radians)'); % Label the y-axis
    grid on; % Display grid lines

elseif input_signal == 2
```

```
    signal_values = input("Enter your signal with [] around it: ");
    start_x = input("Enter the starting point for plotting: ");

    subplot(1,1,1);
    plot_signal(signal_values, start_x); % Call the plot_signal function to plot the input sequence
    title('Original Signal'); % Add title to the subplot
    xlabel('Data points'); % Label the x-axis
    ylabel('Values'); % Label the y-axis
    grid on; % Display grid lines

elseif input_signal == 3

    signal_values = input("Enter your signal with [] around it: ");
    start_x = input("Enter the starting point for plotting: ");
    N = input("Enter the number of points for DFT: ");

    subplot(2,1,1);
    plot_signal(abs(dft_result), start_x); % Plot the magnitude of the DFT sequence
    title('Magnitude of DFT Sequence'); % Add title to the subplot
    xlabel('Frequency (k)'); % Label the x-axis
    ylabel('Magnitude'); % Label the y-axis
    grid on; % Display grid lines

    % Plot the phase of the DFT sequence
    subplot(2,1,2);
    plot_signal(angle(dft_result), start_x); % Plot the phase of the DFT sequence
    title('Phase of DFT Sequence'); % Add title to the subplot
    xlabel('Frequency (k)'); % Label the x-axis
    ylabel('Phase in Radians)'); % Label the y-axis
    grid on; % Display grid lines
else
    disp("Please enter a valid number.")
end

% Function to compute the Discrete Fourier Transform (DFT) of a signal
function dft_result = dft(signal, N)
    dft_result = zeros(1, N); % Initialize output sequence as zero
    n = 0:length(signal)-1; % Generate a vector of time indices
    k = 0:N-1; % Generate a vector of frequency indices
    [N_mat, n_mat] = meshgrid(k, n); % Create matrices of frequency and time indices
    W = exp(-1j * 2 * pi / N * N_mat .* n_mat); % Compute the DFT matrix
    dft_result = signal * W; % Compute the DFT
end

% Function to plot the input sequence
function plots = plot_signal(signal, start_x)
    x_values = start_x:start_x+length(signal)-1; % Generate the x-coordinates for plotting
    plots = stem(x_values, signal, 'MarkerFaceColor', 'blue'); % Plot the input sequence
    xlim([x_values(1)-1, x_values(end)+1]); % Set the x-axis limits
    yline(0, 'b'); % Add a blue horizontal line at y=0
    xline(0, 'r'); % Add a red vertical line at x=0
    grid on; % Display grid lines
end
```

Test Case 1: Selecting the first option by pressing 1

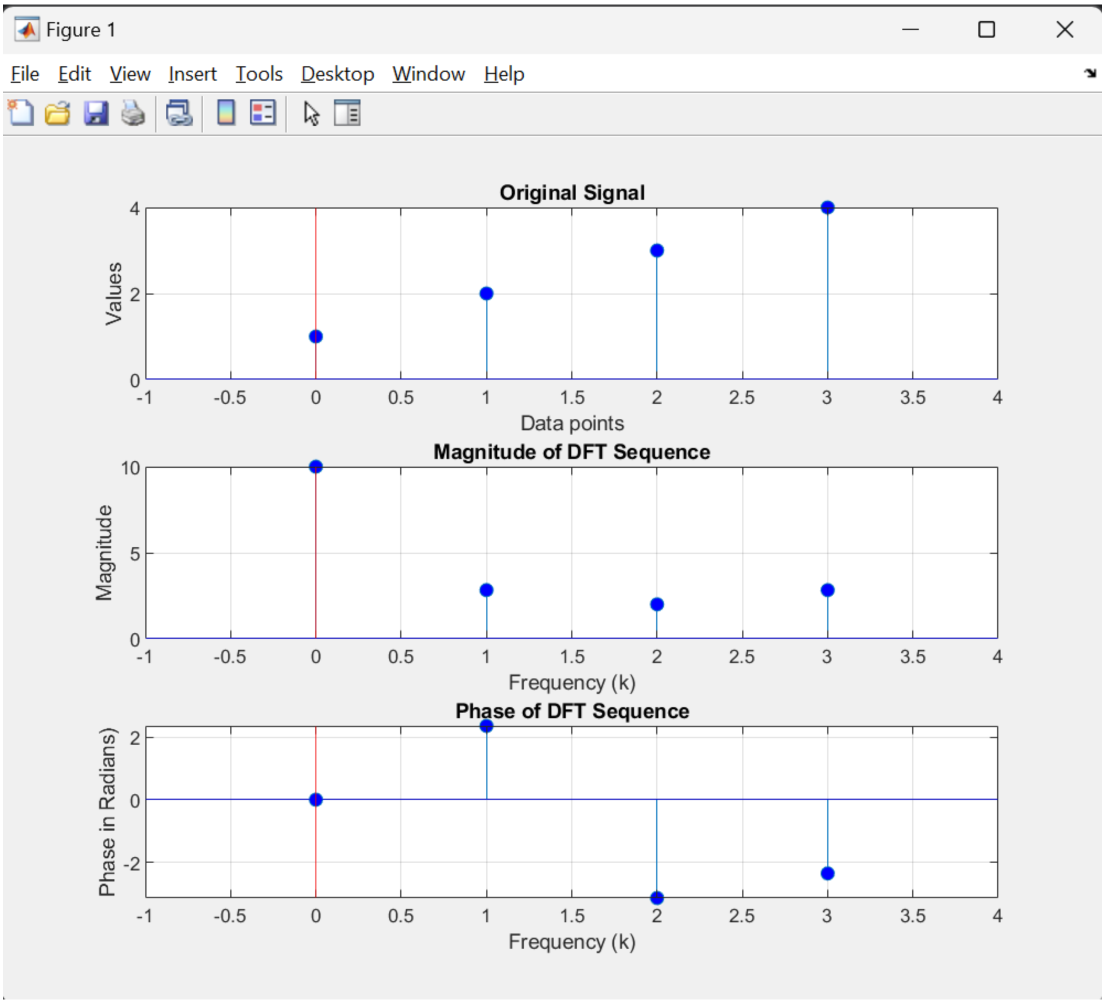
Here we are trying to see the original signal as well as its DFT which can be seen in the output

Input:



```
>> Custom_Discrete_Time_Signal
1 : To View the original Signal and its DFT
2 : To View the original Signal only
3 : To View the DFT of the Signal only
Enter Your Choice: 1
Enter your signal with [] around it: [1,2,3,4]
Enter the starting point for plotting: 0
Enter the number of points for DFT: 4
fx >>
```

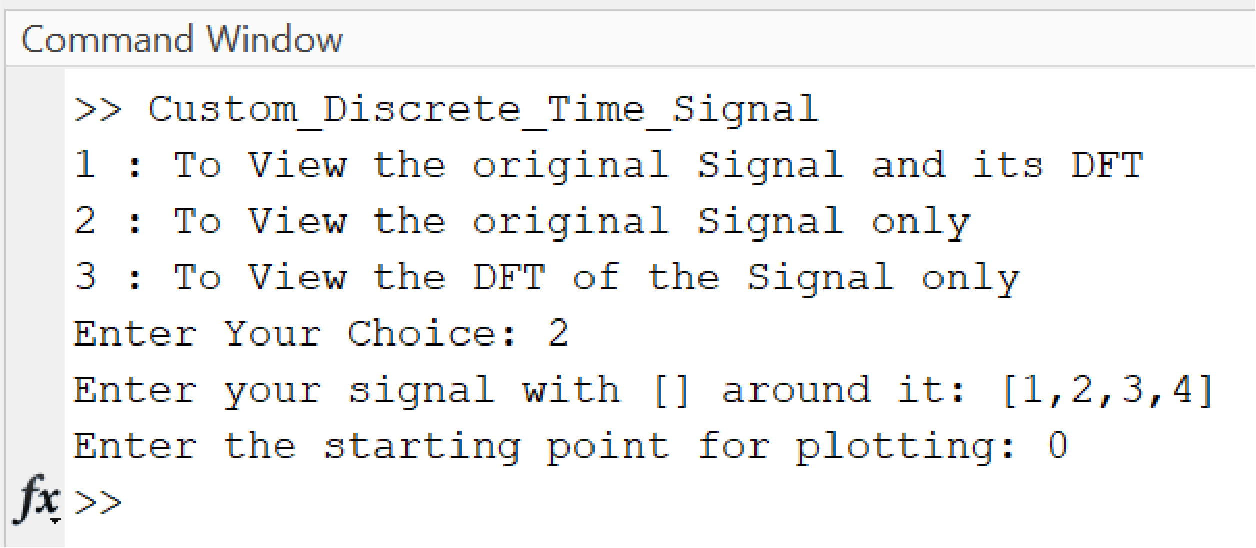
Output:



Test Case 2: Selecting the second option by pressing 2

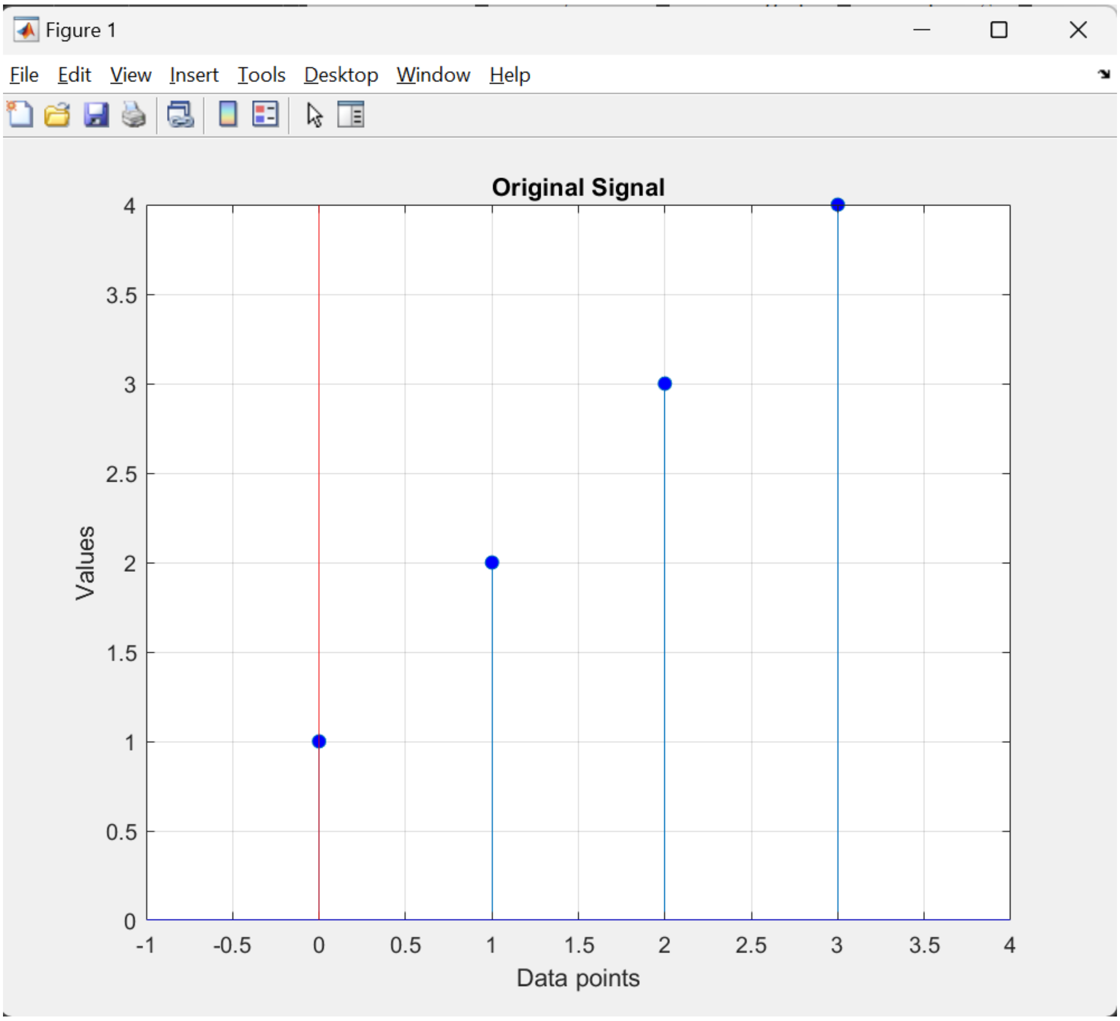
Here we are trying to see the original signal only which can be seen in the output

Input:



```
>> Custom_Discrete_Time_Signal
1 : To View the original Signal and its DFT
2 : To View the original Signal only
3 : To View the DFT of the Signal only
Enter Your Choice: 2
Enter your signal with [] around it: [1,2,3,4]
Enter the starting point for plotting: 0
>>
```

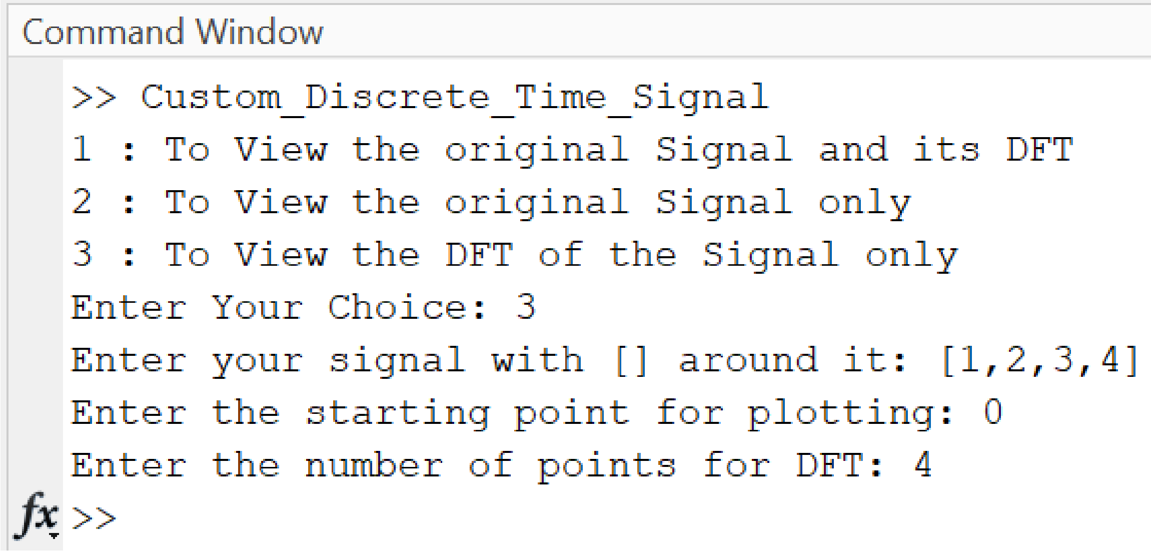
Output:



Test Case 3: Selecting the third option by pressing 3

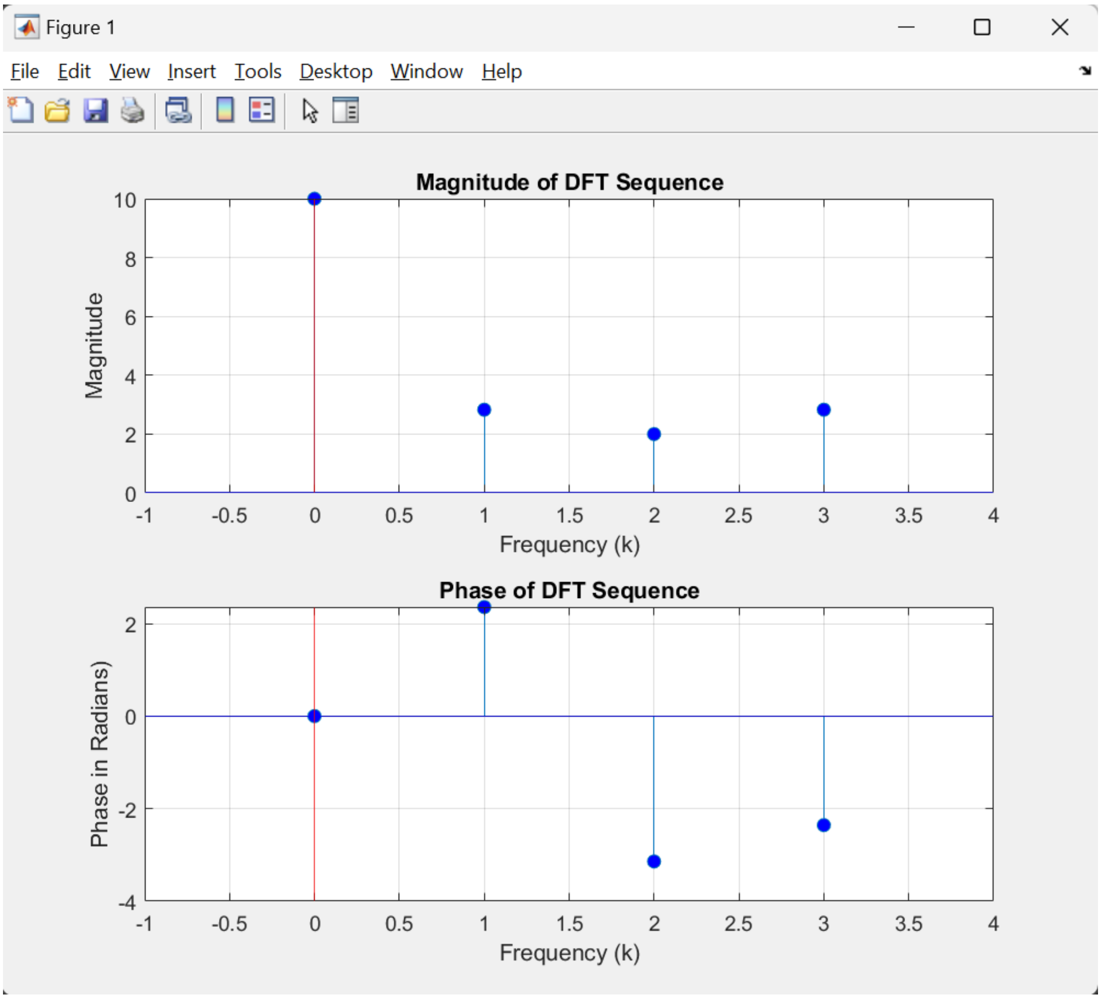
Here we are trying to see the DFT of the original signal only which can be seen in the output

Input:



```
>> Custom_Discrete_Time_Signal
1 : To View the original Signal and its DFT
2 : To View the original Signal only
3 : To View the DFT of the Signal only
Enter Your Choice: 3
Enter your signal with [] around it: [1,2,3,4]
Enter the starting point for plotting: 0
Enter the number of points for DFT: 4
fx >>
```

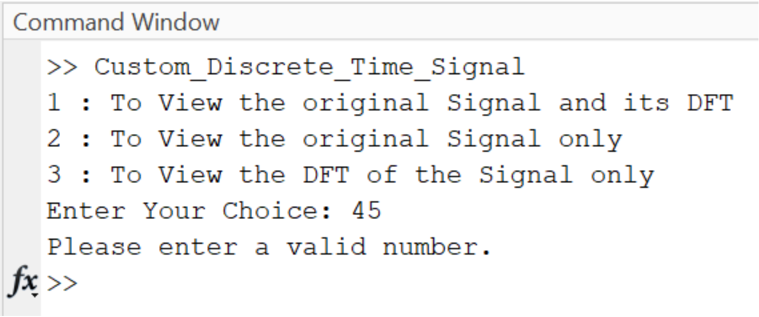
Output:



Test Case 4: Selecting a random number

Here we are trying to select a number which is not in the given option to see how the code deals with it

Input:



```
>> Custom_Discrete_Time_Signal
1 : To View the original Signal and its DFT
2 : To View the original Signal only
3 : To View the DFT of the Signal only
Enter Your Choice: 45
Please enter a valid number.
>>
```

The code asks the user to enter a valid number indicating that the user’s option has no functionality

Conclusion: -

This MATLAB code allows users to analyze discrete time signals and their Discrete Fourier Transform (DFT) through an interactive interface. Users can choose to view the original signal, its DFT, or both. They input signal values and parameters like the starting point for plotting and the number of points for the DFT. The code then plots the selected components accordingly. It calculates the DFT using a custom function and plots the signals with appropriate labeling and grid lines. Overall, it's a useful tool for signal processing tasks, providing insights into the frequency domain characteristics of discrete time signals.