|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Digital Signal & Image Processing Laboratory** | **Semester:** | **VI** |
| **Date of Performance:** | **29 / 01 / 2025** | **Batch No.:** | **B - 2** |
| **Faculty Name:** | **Dr. Om Goswami** | **Roll No.:** | **16014022050** |
| **Faculty Sign & Date:** |  | **Grade/Marks:** | **\_\_\_ / 20** |

**Experiment No: 3**

**Title:** To find DTFT / DFT of given DT signal.

|  |
| --- |
| **Objective:**  To compute and analyze the Discrete-Time Fourier Transform (DTFT) and the Discrete Fourier Transform (DFT) of discrete signals. |

|  |
| --- |
| **COs to be achieved:** |
| **CO1:** Identify various discrete time signals and systems and perform signal manipulation |

|  |
| --- |
| **Materials Required:** MATLAB software  **Books/ Journals/ Websites referred:**   1. Nagoor Kani “Digital Signal Processing”, 2nd Edition, TMH Education. 2. Alan V. Oppenheim and Ronald W. Schafer, "Discrete-Time Signal Processing." 3. MATLAB Documentation: <https://www.mathworks.com/help/matlab/> |

|  |
| --- |
| Theory:Discrete-Time Fourier Transform (DTFT)The DTFT of a discrete-time signal is a continuous function of frequency defined as:X(ω) = Σ x[n] e^(-jωn)Where:• x[n] is the discrete-time signal.• ω is the normalized angular frequency (radians/sample). The DTFT provides a complete frequency-domain representation of a discrete-time signal. It is continuous and periodic in 2π.  **Discrete Fourier Transform (DFT)**  The DFT is a discrete counterpart of the DTFT, used for practical computation. For a finite sequence x[n] of length N, the DFT is defined as:  X[k] = Σ x[n] e^(-j(2π/N)kn)  Where:  • k is the frequency index (k = 0, 1, ..., N-1)  • X[k] is the sampled version of the DTFT  The relationship between the time-domain and frequency-domain representations is given by the inverse DFT:  x[n] = (1/N) Σ X[k] e^(j(2π/N)kn)  The DFT is computed efficiently using the Fast Fourier Transform (FFT) algorithm. |

|  |
| --- |
| **Stepwise-Procedure:** ****Compute DTFT of a Signal:******Define the Signal: Choose a discrete-time signal, e.g., x[n] = {1, 2, 3, 4}** **Generate Frequency Samples: Create a range of frequencies in for DTFT computation.****Compute DTFT: Use the formula for DTFT to compute for each frequency value.****Plot Results: Plot the magnitude and phase**  Write MATLAB/Python code for performing with and without inbuilt function  **Code:**  w = 0:0.01:pi;  Num = 1;  Den = [1, -0.5];  Xw = freqz(Num, Den, w);  MagX = abs(Xw); % Magnitude spectrum  PhX = angle(Xw); % Phase spectrum  RealX = real(Xw); % Real part  ImagX = imag(Xw); % Imaginary part  figure;  % Magnitude Spectrum  subplot(2,2,1);  plot(w/pi, MagX);  title('Magnitude Spectrum');  xlabel('\omega (rad/sample)');  ylabel('|X(\omega)|');  grid on;  % Phase Spectrum  subplot(2,2,2);  plot(w/pi, PhX);  title('Phase Spectrum');  xlabel('\omega (rad/sample)');  ylabel('\angleX(\omega) (rad)');  grid on;  % Real Part  subplot(2,2,3);  plot(w/pi, RealX);  title('Real Part');  xlabel('\omega (rad/sample)');  ylabel('Re\{X(\omega)\}');  grid on;  % Imaginary Part  subplot(2,2,4);  plot(w/pi, ImagX);  title('Imaginary Part');  xlabel('\omega (rad/sample)');  ylabel('Im\{X(\omega)\}');  grid on;  **Output:**     ****Compute DFT of a Signal:******Define the Signal: Use the same signal as in Part 1.****Compute DFT: Use the DFT formula or an in-built function (e.g., fft in Python/MATLAB):****Plot Results: Plot the magnitude and phase.****Compare with DTFT: Overlay the sampled DTFT with the DFT results to observe the relationship.** Write MATLAB/Python code for performing DFT with and without inbuilt function  **Code:**  % input signal x(n)  x = [2 1 3 2];  N = length(x);  X\_manual = zeros(1, N);  % DFT manually  for k = 0:N-1  for n = 0:N-1  X\_manual(k+1) = X\_manual(k+1) + x(n+1)\*exp(-1i\*2\*pi\*k\*n/N);  end  end  % using fft  X\_fft = fft(x);  disp('Manual DFT coordinates:');  disp(X\_manual);  disp('FFT DFT coordinates:');  disp(X\_fft);  subplot(2,1,1);  stem(0:N-1, abs(X\_manual));  title('Magnitude of DFT (Manual Calculation)');  xlabel('Frequency Index');  ylabel('Magnitude');  subplot(2,1,2);  stem(0:N-1, abs(X\_fft));  title('Magnitude of DFT (FFT)');  xlabel('Frequency Index');  ylabel('Magnitude');  **Output:** |

|  |
| --- |
| **Conclusion:**  To compute and analyze the Discrete-Time Fourier Transform (DTFT) and the Discrete Fourier Transform (DFT) of discrete signals. In this experiment, we analyzed signals using both the Discrete Fourier Transform (DFT) and the Discrete-Time Fourier Transform (DTFT). The DTFT provides a continuous frequency representation, while the DFT gives a discrete approximation, demonstrating how sampling affects frequency analysis in digital signal processing. |

|  |
| --- |
| **Post Lab Question:**   1. **What is the difference between DTFT and DFT?**  * The Discrete-Time Fourier Transform (DTFT) is a continuous function of frequency that represents the frequency content of a discrete-time signal over an infinite range. * The Discrete Fourier Transform (DFT) is a sampled version of the DTFT, computed at discrete frequency points, making it suitable for digital computation. DFT is typically used in practical applications since it can be efficiently computed using the Fast Fourier Transform (FFT).  1. **Why is DFT preferred for computational purposes?**  * DFT is preferred because it operates on a finite-length sequence, making it feasible for numerical computation. * It can be efficiently computed using the Fast Fourier Transform (FFT), significantly reducing computational complexity from O(N2) (for direct computation) to O(N logN). * Unlike DTFT, which is defined over a continuous range, DFT provides discrete frequency components that are easier to handle in digital systems.  1. **How does zero-padding affect the DFT computation?**  * Increases Frequency Resolution: Zero-padding a signal before computing the DFT increases the number of frequency points in the spectrum, providing a smoother and more detailed frequency representation. * Does Not Improve Actual Frequency Content: Zero-padding does not add new information but helps in visualizing spectral components more clearly. * Reduces Spectral Leakage: It can help in reducing spectral leakage by making the signal length fit better into the DFT framework.  1. **Explain the significance of magnitude and phase plots in frequency analysis.**  * Magnitude Plot: Represents the strength of different frequency components in a signal, helping to identify dominant frequencies. It is widely used in signal and image processing. * Phase Plot: Represents the phase shift of frequency components, which is crucial in signal reconstruction and systems where phase information affects performance (e.g., communication systems and filter design). * Together, magnitude and phase fully describe the frequency-domain characteristics of a signal, enabling complete reconstruction in the time domain. |

|  |
| --- |
| **Signature of faculty in-charge with Date:** |