|  |  |
| --- | --- |
| **Name** | Shubham Golwal |
| **UID no.** | 2020300015 |
| **Experiment No.** | EXP 1 |

|  |  |
| --- | --- |
| **AIM:** | To sample and reconstruct the given continuous time signal using MATLAB |
| **Program A** | |
| **Objective:** | Consider three cases: Fs > 2Fmax, Fs = 2Fmax, Fs < 2Fmax. Draw your conclusion in all three cases by comparing original CT signal and Reconstructed CT Signal. |
| **Software Tool Used:** | MATLAB - Mathworks  MATLAB R2022b |
| **Code:** | n = 10; %used to generate 10 cycles  f = 1e6;  fc = 5e3; %Frequency for cosine part of the signal  fs = 10e3; %Frequency for sine part of the signal  t = 0:1/f:n\*1/fc; %Time scale for the graph  %Signal to be sampled  x = 1 + 13\*cos(2\*pi\*fc\*t) + 05\*sin(2\*pi\*fs\*t);  subplot(5,1,1);  plot(t,x);  title('Input Signal: 1 + 13\*cos(2\*pi\*fc\*t) + 05\*sin(2\*pi\*fs\*t)');  xlabel('Time');  ylabel('Amplitude');  hold on;  % Sampling and Reconstruction 1  fs1 = 8e3; % Fs<2Fm @8kHz sampling rate  %Time scale for the graph using sampling frequency  ts1 = 0:1/fs1:n\*1/fc;  x1 = 1 + 13\*cos(2\*pi\*fc\*ts1) + 05\*sin(2\*pi\*fs\*ts1);  subplot(5,1,3);  title('Sampling and Reconstruction at fs<2f (fs = 8kHz)');  xlabel('Time');  ylabel('Amplitude');  hold on;  reconst1 = 1 + 13\*cos(2\*pi\*(fc-fs1)\*t) + 05\*sin(2\*pi\*(fs-fs1)\*t);  stem(ts1,x1); %to show samples  plot(t,reconst1); %reconstructed wave from the samples  % Sampling and Reconstruction 2  fs2 = 20e3; % Fs=2Fm @20kHz sampling rate  ts2 = 0:1/fs2:n\*1/fc;  x2 = 1 + 13\*cos(2\*pi\*fc\*ts2) + 05\*sin(2\*pi\*fs\*ts2);  subplot(5,1,4);  title('Sampling and Reconstruction at fs=2f (fs = 20kHz)');  xlabel('Time');  ylabel('Amplitude');  hold on;  recon2 = 1 + 13\*cos((2\*pi\*fc)\*t) + 05\*sin(2\*pi\*fs\*t);  stem(ts2,x2);  plot(t,recon2);  % Sampling and Reconstruction 3  fs3 = 50e3; % Fs>2Fm @50kHz sampling rate  ts3 = 0:1/fs3:n\*1/fc;  x3 = 1 + 13\*cos(2\*pi\*fc\*ts3) + 05\*sin(2\*pi\*fs\*ts3);  subplot(5,1,5);  title('Sampling and Reconstruction at fs>2f (fs = 20kHz)');  xlabel('Time');  ylabel('Amplitude');  hold on;  recon3 = 1 + 13\*cos(2\*pi\*fc\*t) + 05\*sin(2\*pi\*fs\*t);  stem(ts3,x3);  plot(t,recon3); |
| **Output** |  |
| **Conclusion**  In this experiment, I was able to reconstruct the given signal for different sampling frequencies which were as follow:  Fs < 2Fmax  Fs = 2Fmax  Fs > 2Fmax  With the output I was able to understand how sampling frequency affects the signal against time in all the three cases. | |

|  |  |
| --- | --- |
| **AIM:** | To perform signal operation on the given L point sequence x[n] using MATLAB |
| **Program B** | |
| **Software Tool Used:** | MATLAB R2022b |
| **Code:** | clc  clear all  close all  n=0:3  x=[1 2 3 4]  subplot(2,1,1);  stem(n,x);  title('Original Signal x(n)');  y=x;  subplot(2,1,2);  stem(n-2,y);  xlim([-5 5])  title('Operated Signal: x(n+2)'); |
| **Output** |  |
| **Conclusion**  In conclusion, performing signal operations on a given L-point sequence x[n] requires a thorough understanding of signal processing techniques and concepts. In this assignment, we have explored various signal processing techniques such as time shifting, scaling, addition, and convolution. By applying these operations to the given L-point sequence, we were able to analyze and manipulate the signal in meaningful ways, revealing important information about its properties and characteristics. Overall, this assignment has provided a valuable opportunity to gain hands-on experience with signal processing techniques and their applications, helping to deepen our understanding of this important field. | |

|  |  |
| --- | --- |
| **AIM:** | To obtain energy and power of the given L point sequence x[n] using MATLAB |
| **Program C** | |
| **Software Tool Used:** | MATLAB R2022b |
| **Code:** | clc  close all  clear all  disp('Energy and Power Evaluation ')  x=input('Enter input signal ')  % for i=1:1e3  % x(i)=1;  % end  subplot(3,1,1)  stem(x);  title('Original Sequence')  disp('Individual Energy of the sequence at every frequency is ')  ind\_energy=x.\*x  subplot(3,1,2)  stem(ind\_energy)  title('Sequence Energy at Frequency')  energy=sum(x.\*x);  % if (energy<1e5)  % disp('The given sequence is an Energy Signal');  % else  % disp('The given sequence is an NOT an Energy Signal');  % end  disp('The Energy of the sequence is ')  energy  F = fft(x);  disp('Individual Power of the sequence at every frequency is ')  pow = F.\*conj(F)  subplot(3,1,3)  stem(pow),  title('Sequence Power at Frequency')  Total\_Power=sum(pow);  disp('The total energy of the sequence is ')  Total\_Power |
| **Output** |  |
| **Conclusion**    In conclusion, obtaining the energy and power of a given L-point sequence x[n] is an important task in signal processing. In this assignment, we have learned how to calculate the energy and power of a discrete-time signal using mathematical formulas and numerical methods. By applying these techniques to the given L-point sequence, we were able to analyze and characterize the signal in terms of its total energy and average power. This information is crucial for understanding the signal's behavior and potential applications in various fields. Overall, this assignment has provided a valuable opportunity to deepen our knowledge and skills in signal processing, helping to prepare us for future challenges in this exciting and rapidly evolving field. | |