Text

Description automatically generated with medium confidenceDigital Communication Systems

**Laboratory Report**

Spring 2024

|  |  |
| --- | --- |
| Laboratory Number: | **2** |
| Laboratory Title: | **Fourier Decomposition** |
| Full Name: | **Leo Berman** |
| TUID: | **916027207** |

**Introduction:**

The purpose of this lab is to utilize MATLAB to generate some standard signals. We then want to perform Fourier Transforms on these signals to decompose them into their basic functions. Once we have these signals in the frequency domain, we can rebuild them by performing an inverse Fourier Transform which we can then plot to gauge the information that was lost by deconstructing and reconstructing the signal.

**Procedure:**

Before starting:

1. Define an amplitude and frequency to use based off of your TUID.

|  |
| --- |
| tuid = [9,1,6,0,2,7,2,0,7];  A = tuid(9) + 2; % amplitude in amps |

Task 1:

1. Generate a 3 second cosine wave with frequency *f* = 10, and plot.
2. Plot the amplitude spectrum
   1. Take the Fast Fourier Transform (FFT)
   2. Rescale it by dividing by the number of elements in the FFT
   3. Shift it to visualize the center at the zero frequency
   4. Plot a stem plot
3. Plot the frequency spectrum
   1. Take the amplitude spectrum as calculated above and plot it on the frequency spectrum
4. Plot the phase spectrum
   1. Take the amplitude spectrum and calculate the magnitude
   2. Then remove all elements that aren’t equal to the maximum value of this magnitude
   3. Calculate the angle between the imagery and real parts of these magnitudes
   4. Plot a stem plot
5. Plot the power spectral density
   1. Plot the magnitude of the Amplitude spectrum squared
6. Reconstruct the signal
   1. Take the number of samples in the FFT and multiply it by the shifted amplitude spectrum
   2. Then take the inverse transform of this
   3. Plot it on the same time scale as the original signal

Task 2:

1. Generate a 3 second cosine wave with frequency *f* = 10 and plot it
2. For each number of samples = 2­2 ,23, 24, 25, 26, 27, 28
   1. Generate the amplitude spectrum as mentioned above and plot
   2. Generate the Reconstruct signal as mentioned above and plot

Task 3:

1. Create a cosine wave of amplitude A and frequency *fc* = 10, and modulate it by a sine wave of amplitude A/2 and frequency *fm* = 20. Plot this signal
2. Generate the amplitude spectrum as mentioned above and plot
3. Generate the power spectral density as mentioned above and plot
4. Generate the phase spectrum as mentioned above and plot
5. Generate the frequency as mentioned above and plot
6. Generate the reconstructed wave as mentioned above and plot

Task 4:

1. Generate a triangular wave that is shifted by half a period and plot 3 periods
2. Generate the amplitude spectrum as mentioned above and plot
3. Generate the spectral density as mentioned above and plot
4. Generate the phase spectrum as mentioned above and plot
5. Generate the reconstructed signal as mentioned above and plot

**Results:**

Task 1:

|  |  |
| --- | --- |
| A graph of a graph of a number  Description automatically generated with medium confidence |  |
|  |  |
|  |  |

Task 2:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Task 3:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Task 4:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

**Descriptive Answers to Tasks:**

Task 1:

|  |
| --- |
| N/A |

Task 2:

|  |  |
| --- | --- |
| Q | A |
| Compare the amplitude spectrums and reconstructed signal spectrums for each case. Have you faced any issues with N is too small? Discuss the possible reasons. | For the first 3 cases (N = 2, 4 ,8), the two peaks at positive and negative aren’t able to be distinguished due to the very small sampling frequency. This ends of with an amplitude spectrum where the overlapping of these frequencies and amplitudes muddies the signal to the point where the tops and bottoms of the sine wavs are indistinguishable. We can also see this on our reconstructed waves where we don’t have a clear sine wave. After this, we see that actual split in the amplitude spectrum and we see the actual shape of the sine wave in the reconstructed signal. As we increase the sample rate, the amplitude peaks narrow and with the higher sampling rate we get closer to the actual frequency of the original wave, until we get to N = 256 where the original wave is almost perfectly reconstructed with the frequency being cut in half. |

Task 3:

|  |  |
| --- | --- |
| Q | A |
| Investigate the harmonics generated by this wave and compare the FFT results to the one obtained from the cosine signal. | The harmonics generated by a modulated signal look different than the original wave because while there are still the larger harmonics in place for the carrier wave, there are also smaller harmonics in place for the modulation wave. |
| Discuss the difference in the number of non-zero Fourier coefficients and their effect on signal reconstruction. | The higher the number of non-zero Fourier coefficients, the more accurately we are able to reconstruct a signal. |
| How does combining two signals affect the phase and amplitude spectrum? | The amplitude spectrum now has peaks at the positive and negative frequencies (*fm*) for the modulation wave. These peaks are significantly smaller than for the carrier wave since the amplitude of the modulation wave is also significantly smaller. The phase spectrum is affected by having the peaks of the amplitude spectrum be much larger since the two waves are offset by their differing frequencies. |

Task 4:

|  |  |
| --- | --- |
| Q | A |
| Is the number of nonzero Fourier coefficients the same (or smaller or higher) as task 3? | Higher because they are symmetric about the origin and the fact that the triangular wave is an odd function. This results in much slower decay which results in way more non-zero Fourier coefficients. |
| Does the reconstructed signal match with the original one? If not, what could be the possible reason? | Yes |

**Conclusion:**

The FFT can be a really great way to compress a signal if you are more worried about things like amplitude instead of frequency. During an FFT, it’s very difficult to not lose information, and if you know when to use it, it can be a useful tool. It’s important to take into consideration symmetry and harmonics due to the ability of those two things to impact the amount of non-zero coefficients.

|  |
| --- |
| Lab 02 Fourier Decomposition  % clear the workspace  clear  close all  clc  clf  % set tuid and amplitude  tuid = [9,1,6,0,2,7,2,0,7];  A = tuid(9) + 2; % amplitude [Volts]  **Part 1**  %% generating original cosine signal  fc = 10; % Frequency  fs = 32\*fc; % Sampling frequecny  T = 1/fc; % Time period  Ts = 1/fs; % Sampling Interval  phase = 0; % Phase  t = 0:Ts:3-Ts; % 2 seconds duartion  x = A\*cos(2\*pi\*fc\*t+phase);  plot(t,x)  set(gca,'fontsize',16)  title('$x(t)=9cos(2 \pi 10 t)$','Interpreter','latex')  xlabel('$t$','Interpreter','latex')  ylabel('$x(t)$','Interpreter','latex')  % Generating the amplitude spectrum  % FFTShift  N = 512;  X = (1/N)\*fft(x,N);  df = fs/N;  X1 = fftshift(X);  sampleIndex = -N/2:1:N/2-1;  f = sampleIndex\*df;  % Generating the frequency spectrum  stem(f,abs(X))  title('Frequency Spectrum (X[f])')  xlabel('frequencies (f)')  ylabel('|X[f]|')  % Generating Phase spectrum  X2 = X1;  tol = max(abs(X1));  X2(abs(X1)<tol) = 0;  phase1 = atan2(imag(X2),real(X2));  stem(f,rad2deg(phase1))  ylabel('X[k]')  xlabel('f (HZ)')  title('Phase Spectrum')  % Generating Power Spectral Density  stem(f,abs(X1).^2)  xlabel('f(HZ)')  ylabel('[X[f]|^2')  title('Power Spectral Density')  plot(f,abs(X1))  title('Amplitude Spectrum')  xlabel('Frequencies (f)')  ylabel('|X(k)|')  % Reconstructing the signal  X = N\*ifftshift(X1);  x\_recon = ifft(X,N);  T = linspace(0,3,N);  plot(T,x\_recon)  set(gca,'fontsize',16)  title('$x(t)\ =\ Reconstructed\ 9cos(2 \pi 10 t)$','Interpreter','latex')  xlabel('$t$','Interpreter','latex')  ylabel('$x(t)$','Interpreter','latex')  **Part 2**  %% generating original cosine signal  fc = 10; % Frequency  fs = 32\*fc; % Sampling frequecny  T = 1/fc; % Time period  Ts = 1/fs; % Sampling Interval  phase = 0; % Phase  t = 0:Ts:3-Ts; % 2 seconds duartion  x = A\*cos(2\*pi\*fc\*t+phase);  plot(t,x)  set(gca,'fontsize',16)  title('$x(t)=9cos(2 \pi 10 t)$','Interpreter','latex')  xlabel('$t$','Interpreter','latex')  ylabel('$x(t)$','Interpreter','latex')  % Generating the amplitude spectrum  % FFTShift  N = 4;  X = (1/N)\*fft(x,N);  df = fs/N;  X1 = fftshift(X);  sampleIndex = -N/2:1:N/2-1;  f = sampleIndex\*df;  plot(f,abs(X1))  title('Amplitude Spectrum N = 4')  xlabel('Frequencies (f)')  ylabel('|X(k)|')  % Reconstructing the signal  X = N\*ifftshift(X1);  x\_recon = ifft(X,N);  T = linspace(0,3,N);  plot(T,x\_recon)  set(gca,'fontsize',16)  title('$x(t)\ =\ Reconstructed\ 9cos(2 \pi 10 t)\ N = 4$','Interpreter','latex')  xlabel('$t$','Interpreter','latex')  ylabel('$x(t)$','Interpreter','latex')  % Generating the amplitude spectrum  % FFTShift  N = 8;  X = (1/N)\*fft(x,N);  df = fs/N;  X1 = fftshift(X);  sampleIndex = -N/2:1:N/2-1;  f = sampleIndex\*df;  plot(f,abs(X1))  title('Amplitude Spectrum N = 8')  xlabel('Frequencies (f)')  ylabel('|X(k)|')  % Reconstructing the signal  X = N\*ifftshift(X1);  x\_recon = ifft(X,N);  T = linspace(0,3,N);  plot(T,x\_recon)  set(gca,'fontsize',16)  title('$x(t)\ =\ Reconstructed\ 9cos(2 \pi 10 t)\ N = 8$','Interpreter','latex')  xlabel('$t$','Interpreter','latex')  ylabel('$x(t)$','Interpreter','latex')  % Generating the amplitude spectrum  % FFTShift  N = 16;  X = (1/N)\*fft(x,N);  df = fs/N;  X1 = fftshift(X);  sampleIndex = -N/2:1:N/2-1;  f = sampleIndex\*df;  plot(f,abs(X1))  title('Amplitude Spectrum N = 16')  xlabel('Frequencies (f)')  ylabel('|X(k)|')  % Reconstructing the signal  X = N\*ifftshift(X1);  x\_recon = ifft(X,N);  T = linspace(0,3,N);  plot(T,x\_recon)  set(gca,'fontsize',16)  title('$x(t)\ =\ Reconstructed\ 9cos(2 \pi 10 t)\ N = 16$','Interpreter','latex')  xlabel('$t$','Interpreter','latex')  ylabel('$x(t)$','Interpreter','latex')  % Generating the amplitude spectrum  % FFTShift  N = 32;  X = (1/N)\*fft(x,N);  df = fs/N;  X1 = fftshift(X);  sampleIndex = -N/2:1:N/2-1;  f = sampleIndex\*df;  plot(f,abs(X1))  title('Amplitude Spectrum N = 32')  xlabel('Frequencies (f)')  ylabel('|X(k)|')  % Reconstructing the signal  X = N\*ifftshift(X1);  x\_recon = ifft(X,N);  T = linspace(0,3,N);  plot(T,x\_recon)  set(gca,'fontsize',16)  title('$x(t)\ =\ Reconstructed\ 9cos(2 \pi 10 t)\ N = 32$','Interpreter','latex')  xlabel('$t$','Interpreter','latex')  ylabel('$x(t)$','Interpreter','latex')  % Generating the amplitude spectrum  % FFTShift  N = 64;  X = (1/N)\*fft(x,N);  df = fs/N;  X1 = fftshift(X);  sampleIndex = -N/2:1:N/2-1;  f = sampleIndex\*df;  plot(f,abs(X1))  title('Amplitude Spectrum N = 64')  xlabel('Frequencies (f)')  ylabel('|X(k)|')  % Reconstructing the signal  X = N\*ifftshift(X1);  x\_recon = ifft(X,N);  T = linspace(0,3,N);  plot(T,x\_recon)  set(gca,'fontsize',16)  title('$x(t)\ =\ Reconstructed\ 9cos(2 \pi 10 t)\ N = 64$','Interpreter','latex')  xlabel('$t$','Interpreter','latex')  ylabel('$x(t)$','Interpreter','latex')  % Generating the amplitude spectrum  % FFTShift  N = 128;  X = (1/N)\*fft(x,N);  df = fs/N;  X1 = fftshift(X);  sampleIndex = -N/2:1:N/2-1;  f = sampleIndex\*df;  plot(f,abs(X1))  title('Amplitude Spectrum N = 128')  xlabel('Frequencies (f)')  ylabel('|X(k)|')  % Generating the phase spectrum  X2 = X1;  tol = max(abs(X1));  X2(abs(X1)<tol) = 0;  phase1 = atan2(imag(X2),real(X2));  figure(6)  stem(f,rad2deg(phase1))  ylabel('X[k]')  xlabel('f (HZ)')  title('Phase Spectrum')  % Reconstructing the signal  X = N\*ifftshift(X1);  x\_recon = ifft(X,N);  T = linspace(0,3,N);  plot(T,x\_recon)  set(gca,'fontsize',16)  title('$x(t)\ =\ Reconstructed\ 9cos(2 \pi 10 t)\ N = 128$','Interpreter','latex')  xlabel('$t$','Interpreter','latex')  ylabel('$x(t)$','Interpreter','latex')  % Generating the amplitude spectrum  % FFTShift  N = 256;  X = (1/N)\*fft(x,N);  df = fs/N;  X1 = fftshift(X);  sampleIndex = -N/2:1:N/2-1;  f = sampleIndex\*df;  plot(f,abs(X1))  title('Amplitude Spectrum N = 256')  xlabel('Frequencies (f)')  ylabel('|X(k)|')  % Reconstructing the signal  X = N\*ifftshift(X1);  x\_recon = ifft(X,N);  T = linspace(0,3,N);  plot(T,x\_recon)  set(gca,'fontsize',16)  title('$x(t)\ =\ Reconstructed\ 9cos(2 \pi 10 t)\ N = 256$','Interpreter','latex')  xlabel('$t$','Interpreter','latex')  ylabel('$x(t)$','Interpreter','latex')  **Part 3**  %% generating original cosine signal  fc = 10; % Frequency  fs = 32\*fc; % Sampling frequecny  fm = 20;  T = 1/fc; % Time period  Ts = 1/fs; % Sampling Interval  phase = pi/6; % Phase  t = 0:Ts:2-Ts; % 2 seconds duartion  x = A \* cos(2\*pi\*fc\*t+phase) + .5 \* A \* sin(2\*pi\*fm\*t+phase);  plot(t,x)  set(gca,'fontsize',16)  title('$x(t)=9cos(2 \pi 10 t) \* 4.5sin(2 \pi 20 t)$','Interpreter','latex')  xlabel('$t$','Interpreter','latex')  ylabel('$x(t)$','Interpreter','latex')  % Generating the amplitude spectrum  % FFTShift  N = 512;  X = (1/N)\*fft(x,N);  df = fs/N;  X1 = fftshift(X);  sampleIndex = -N/2:1:N/2-1;  f = sampleIndex\*df;  plot(f,abs(X1))  title('Amplitude Spectrum')  xlabel('Frequencies (f)')  % Generating the power spectral density  stem(f,abs(X1).^2)  xlabel('f(HZ)')  ylabel('[X[f]|^2')  title('Power Spectral Density')  % Generating the phase spectrum  X2 = X1;  tol = max(abs(X1));  X2(abs(X1)<tol) = 0;  phase1 = atan2(imag(X2),real(X2));  stem(f,rad2deg(phase1))  ylabel('X[k]')  xlabel('f (HZ)')  title('Phase Spectrum')  % Generating the frequency spectrum  stem(f,abs(X))  title('Frequency Spectrum (X[f])')  xlabel('frequencies (f)')  ylabel('|X[f]|')  % Reconstructing the signal  X = N\*ifftshift(X1);  x\_recon = ifft(X,N);  T = linspace(0,3,N);  plot(T,x\_recon)  set(gca,'fontsize',16)  title('$x(t)\ =\ Reconstructed\ 9cos(2 \pi 10 t) \* 4.5sin(2 \pi 20 t)$','Interpreter','latex')  xlabel('$t$','Interpreter','latex')  ylabel('$x(t)$','Interpreter','latex')  **Part 4**  %% Generating the original signal  vec\_size = 500;  f = 1;  fs = 32 \* f;  t = linspace(0,(1/f)\*3,vec\_size); % x axis  x = A\*sawtooth(2\*pi\*(t + f/4)\* f,.5) ;  plot(t,x)  xlabel('Time');  ylabel('Amplitude');  title('Triangle Wave')  grid on  % Generating the amplitude spectrum  % FFTShift  N = 512;  X = (1/N)\*fft(x,N);  df = fs/N;  X1 = fftshift(X);  sampleIndex = -N/2:1:N/2-1;  f = sampleIndex\*df;  plot(f,abs(X1))  title('Amplitude Spectrum')  xlabel('Frequencies (f)')  ylabel('|X(k)|')  % Generating the power spectral density  stem(f,abs(X1).^2)  xlabel('f(HZ)')  ylabel('[X[f]|^2')  title('Power Spectral Density')  % Generating the phase spectrum  X2 = X1;  tol = max(abs(X1));  X2(abs(X1)<tol) = 0;  phase1 = atan2(imag(X2),real(X2));  stem(f,rad2deg(phase1))  ylabel('X[k]')  xlabel('f (HZ)')  title('Phase Spectrum')  % Reconstructing the signal  X = N\*ifftshift(X1);  x\_recon = ifft(X,N);  T = linspace(0,3,N);  plot(T,x\_recon)  set(gca,'fontsize',16)  title('$x(t)\ =\ Reconstructed\ Triangle Wave$','Interpreter','latex')  xlabel('$t$','Interpreter','latex')  ylabel('$x(t)$','Interpreter','latex') |