Dhirubhai Ambani Institute of Information and Communication Technology

ADSP

LAB – 3 : Investigate Phase and Magnitude

Name: Manish Manojkumar Prajapati

Student ID: 202411012

Branch: MTech ICT ML (2024 - 26)

Date: 30/01/2025

EXERCISES

1.) Plot 2 image signals.

CODE:

```
% Reading the Image file

% imread function reads the image file and stores it as a matrix in MATLAB
% For grayscale images: 2D matrix (height x width)
% For RGB images: 3D matrix (height x width x 3, where the third dimension
% representing color channels: Red, Green and Blue)

image_manish = imread("My_pic_tie.png");
image_ms_dhoni = imread("ms-dhoni-image.jpg");

% imshow function is used for displaying the image
% imshow(image_manish);
% imshow(image_ms_dhoni);

% Displaying images uploaded

% Creating a figure window
figure;

% Displaying image of myself
subplot(1, 2, 1); % means 1 row, 2 columns, 1st position
imshow(image_manish);
title('Manish Prajapati');
```

```
% Displaying image of MS Dhoni
subplot(1, 2, 2); % means 1 row, 2 columns, 2nd position
imshow(image_ms_dhoni);
title('MS Dhoni');
```

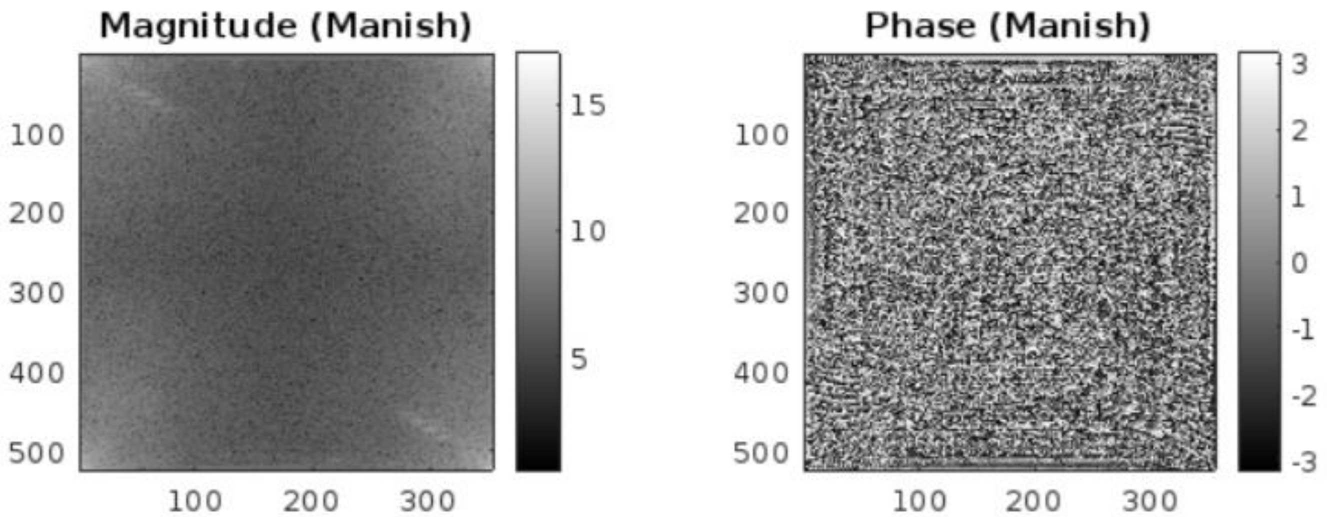
2.) Plot Magnitude and Phase of image signals.

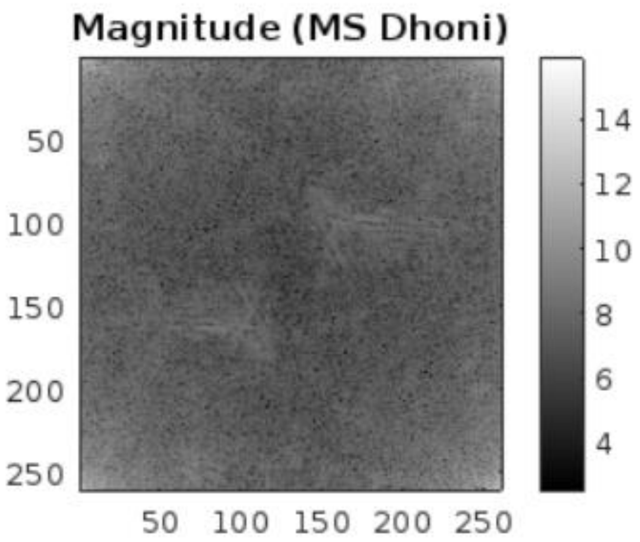
Answer:

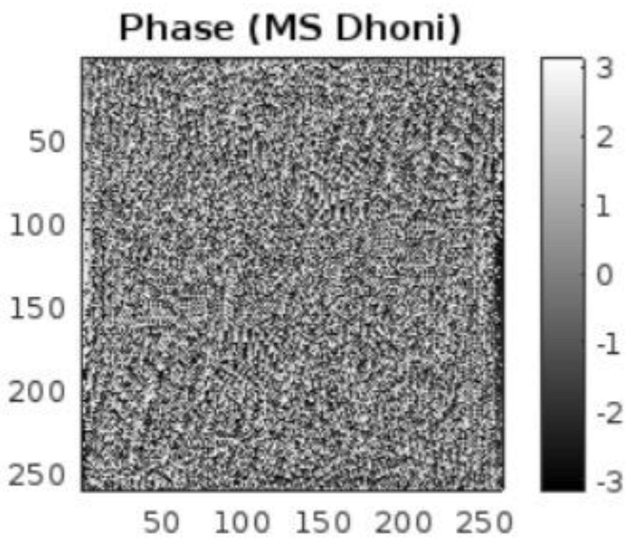
To plot the magnitude and phase of the images, we need to convert them into grayscale, as we will require to calculate the 2D Fourier Transform of the images.

```
% PLOTTING MAGNITUDE AND PHASE OF 2 IMAGES
% FIRSTLY, we need to convert them to grayscale if they are RGB images

if size(image_manish, 3)==3
    image_manish = rgb2gray(image_manish);
end
if size(image_ms_dhoni, 3) == 3
    image_ms_dhoni = rgb2gray(image_ms_dhoni);
end


% Step - 2: Computing the 2D Fourier Transform

fft_manish = fft2(double(image_manish));
fft_ms_dhoni = fft2(double(image_ms_dhoni));


% Step - 3: Computing the magnitude and phase
```

```
magnitude_manish = abs(fft_manish);
phase_manish = angle(fft_manish);

magnitude_ms_dhoni = abs(fft_ms_dhoni);
phase_ms_dhoni = angle(fft_ms_dhoni);


% Step - 4: Plot Magnitude and Phase

% Plotting for myself
figure;

subplot(2, 2, 1);

% Using log-scale for better visualization
imagesc(log(1+magnitude_manish));
colormap gray;
colorbar;
title('Magnitude (Manish)');

subplot(2, 2, 2);
imagesc(phase_manish);
colormap gray;
colorbar;
title('Phase (Manish)');


% Plotting for MS Dhoni
subplot(2, 2, 3);
imagesc(log(1+magnitude_ms_dhoni));
colormap gray;
colorbar;
title("Magnitude (MS Dhoni)");

subplot(2, 2, 4);
imagesc(phase_ms_dhoni);
colormap gray;
colorbar;
title('Phase (MS Dhoni)');
```







3.) Exchanging Magnitude and Phase of both images and creating new ones.

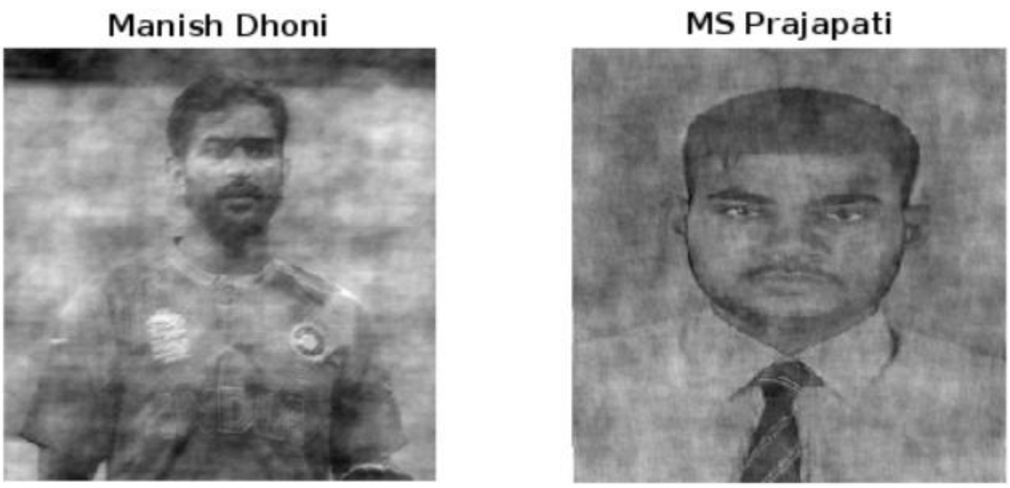
Answer:

a.) To create these images, we need to combine the magnitude of one with the phase of the other, **using polar form of complex numbers.**

b.) Then we compute the inverse fourier transform to reconstruct back the new images.

```
% EXCHANGING THE MAGNITUDE AND PHASE OF IMAGES

% Step - 1: Combining Magnitude and phase

% Combining magnitude of Manish and phase of Dhoni

manish_dhoni = magnitude_manish .* exp(1i*phase_ms_dhoni);


% Combining magnitude of MS Dhoni and phase of Manish

ms_prajapati = magnitude_ms_dhoni .* exp(1i*phase_manish);


% Step - 2: Computing Inverse Fourier Transform

manish_dhoni = real(ifft2(manish_dhoni));

ms_prajapati = real(ifft2(ms_prajapati));
```

```
% Step - 3: Normalizing and displaying the new images

% Normalizing images

manish_dhoni = mat2gray(manish_dhoni);
ms_prajapati = mat2gray(ms_prajapati);

% Displaying new images

figure;
subplot(1, 2, 1);
imshow(manish_dhoni);
title("Manish Dhoni");

subplot(1, 2, 2);
imshow(ms_prajapati);
title("MS Prajapati");
```



4.) Observations.

Answer:

- ➪ First image is combination of magnitude of Manish and phase of MS Dhoni, and the overall image created dominates to the face of MS Dhoni.
- ➪ Second image is combination of magnitude of MS Dhoni and phase of Manish, and, overall image that was obtained, dominated to the face of Manish.
- ➪ Hence, regardless of the Magnitude, the reconstruction of image signal heavily depends on the phase.

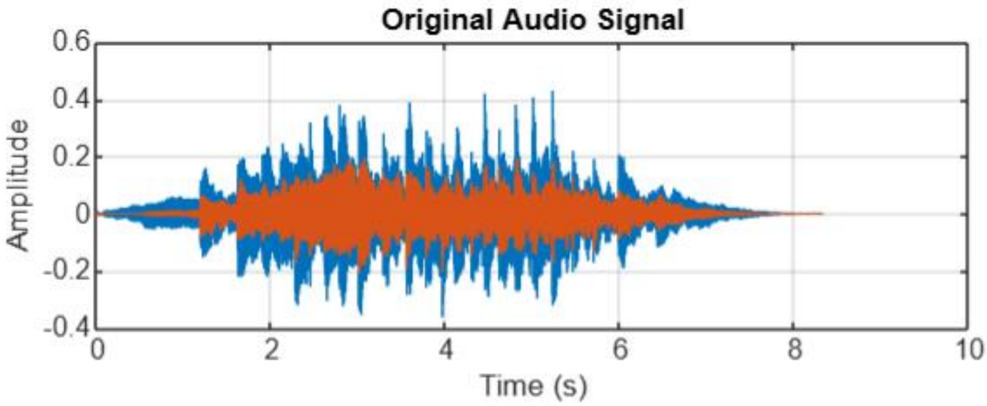
Reason:

- ⇨ Phase contains critical information about the spatial structure and edges of the image, while the magnitude primarily represents the strength of the frequency components.
- ⇨ **Phase** of the Fourier transform encodes the relative timing (or alignment) of the sinusoidal components that make up the image.
- ⇨ **Even small changes** in the phase can significantly **alter the appearance of the reconstructed image.**
- ⇨ **Phase determines the structure:** The phase information is responsible for aligning the frequency components in such a way that they reconstruct original spatial features (e.g., edges, shapes, and textures)
- ⇨ **Magnitude determines the contrast:** The magnitude affects the contrast and intensity of the reconstructed image, but does not significantly alter the spatial arrangement of features.

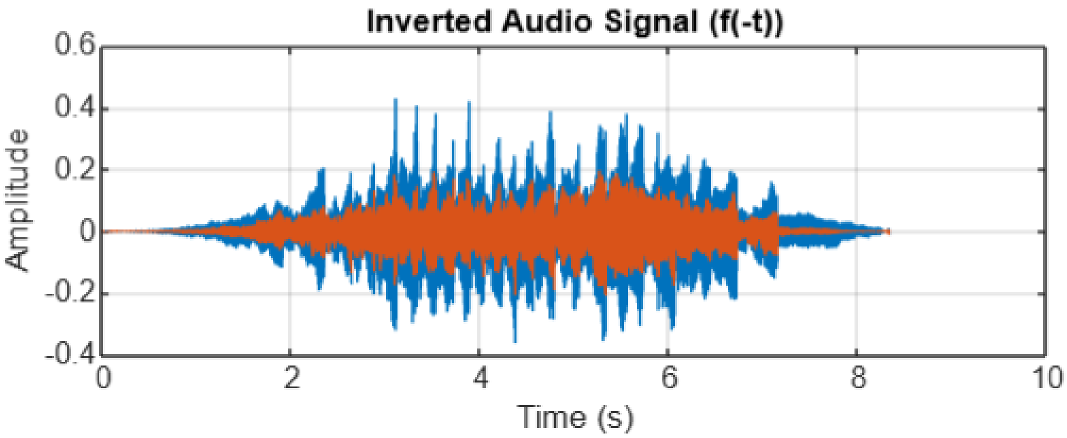
5.) Inverting an audio signal and listening it, i.e., convert x(t) to x(-t).

Code:

```
% 202411012
% Inverting the audio signals and playing them

% Step - 1: Loading the audio signal/file
[audioSignal, Fs] = audioread('BAK.wav');


% Step - 2: Plotting the original audio signal
t = (0: length(audioSignal)-1)/Fs;

figure;
subplot(2, 1, 1);
plot(t, audioSignal);
xlabel('Time (s)');
ylabel('Amplitude');
title('Original Audio Signal');
grid on;
```




```
% Step - 3: Inverting the audio signal (time reversal)
% We use flip function to invert the audio file
invertedAudioSignal = flip(audioSignal);


% Step - 4: Plot the inverted audio signal
subplot(2, 1, 2);
plot(t, invertedAudioSignal);
xlabel('Time (s)');
ylabel('Amplitude');
title('Inverted Audio Signal (f(-t))');
grid on;
```




```
% Step - 5: Playing the inverted audio signal
sound(invertedAudioSignal, Fs);

audiowrite('inverted_BAK.wav', invertedAudioSignal, Fs);
```

<><><><><><><><><><><><><><><><><><><><><><><><><><><><>

<><><><><><><><><><><><><><><><><><><><><><><><><><><><>

<><><><><><><><><><><><><><><><><><><><><><><><><><><><>