EEE 391

MATLAB Assignment 2

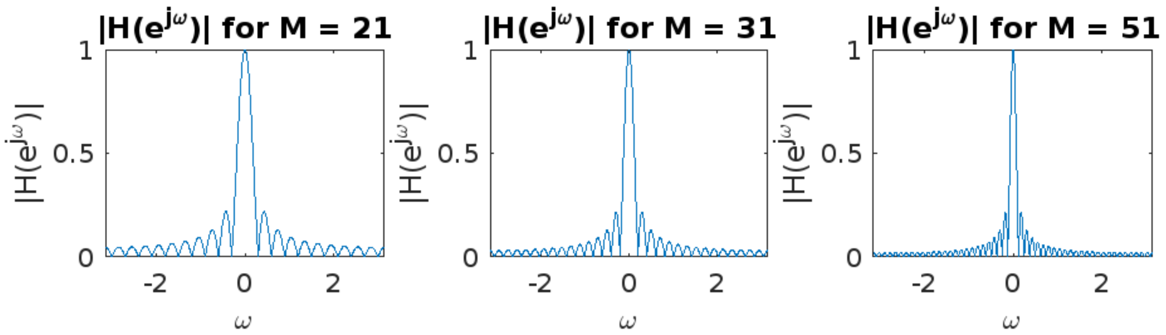
Full Name: Arda İynem
ID: 22002717
Section: 2

Part 1

1.1.1) MATLAB Code 1 (Averaging without Noise)

```
image = imread('Car.bmp');
imageMatrix = double(image);
grayscaleMatrix = mat2gray(imageMatrix, [0 255]);

filterSizeM = [21, 31, 51];
frequencyOmega = linspace(-pi, pi, 1000);

figure;

for i = 1:length(filterSizeM)
   M = filterSizeM(i);

   OneDimAveragingFilter = ones(1, M) / M;
   frequencyResponseFunc = freqz(OneDimAveragingFilter, 1,
frequencyOmega);

   subplot(3, 3, i);
   plot(frequencyOmega, abs(frequencyResponseFunc));
   title(['|H(e^{j\omega})| for M = ' num2str(M)]);
   xlabel('\omega');
   ylabel('|H(e^{j\omega})|');

   filtered_image = imfilter(grayscaleMatrix,
OneDimAveragingFilter);

   subplot(2, length(filterSizeM), i + length(filterSizeM));
   imshow(filtered_image);
   title(['Averaged Image (M = ' num2str(M) ')']);
end
```

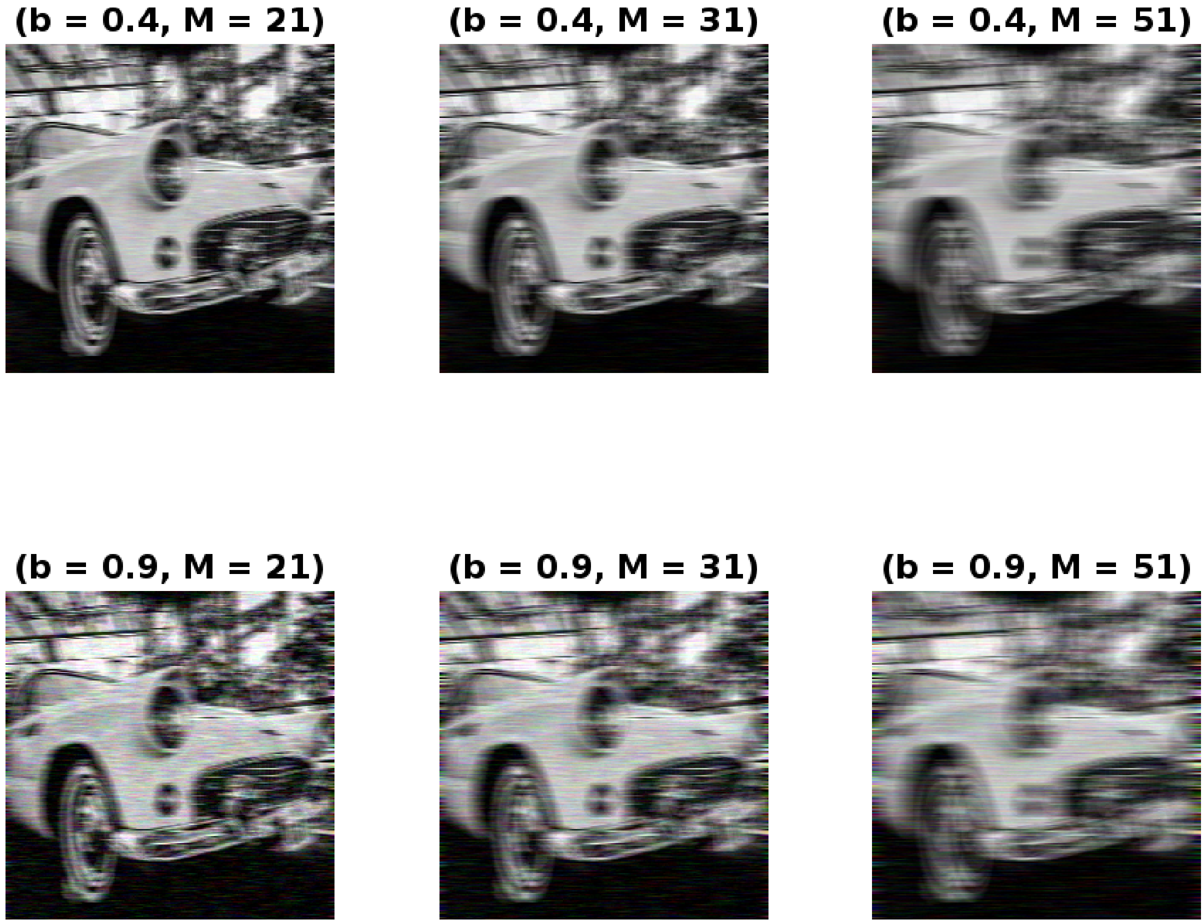
1.1.2) Output 1: Filtered Images (Averaging without Noise)



1.2.1) MATLAB Code 2 (Averaging with Noise)

```
image = imread('Car.bmp');
imageMatrix = double(image);
grayscaleMatrix = mat2gray(imageMatrix, [0 255]);

noiseLevels = [0.4, 0.9];
filterSizeM = [21, 31, 51];

figure;

for j = 1:length(noiseLevels)
   b = noiseLevels(j);

   noiseImage = grayscaleMatrix + b *
(rand(size(grayscaleMatrix)) - 0.5);

   for i = 1:length(filterSizeM)
       M = filterSizeM(i);

       OneDimAveragingFilter = ones(1, M) / M;
       filteredImage = imfilter(noiseImage,
OneDimAveragingFilter);

       subplot(length(noiseLevels), length(filterSizeM), (j -
1) * length(filterSizeM) + i);
       imshow(filteredImage);
       title(['(b = ' num2str(b) ', M = ' num2str(M) ')']);
   end
end
```

1.2.2) Output 2: Filtered Images (Averaging with Noise)



1.3) Comments on the Outputs

1) The filter applied to an image results in an effect akin to smudging, blurring, or smoothing its visual details.

2) Smaller values of M, for instance, M=21, cause a less pronounced impact by the filter, thus preserving finer details within the image. Conversely, larger values of M, like M=51, intensify the filter's effect, leading to a loss of smaller details and a blurring of the image. This blurring is particularly noticeable in the absence of sharp transitions.

3) As the filter exclusively operates on the q variable, representing the horizontal dimension, its blurring effect manifests primarily along the horizontal axis, not affecting the vertical axis.

4) The magnitude of the frequency response, denoted as $|H(e^{\wedge}j\omega)|$, delineates the filter's responsiveness to diverse frequency components. Analysis of graphical representations reveals that the filter accentuates lower frequencies while attenuating higher frequencies, consequently contributing to image blurring. Moreover, an increase in M from 21 to 51 amplifies this accentuation and attenuation, further contributing to the blurring effect.

5) Given the filter's nature in smoothing sharp edges, corresponding to abrupt transitions from exterior (zero values) to interior within the image, it tends to smooth these edges as well. The extent of this blurring is less evident with smaller M values and becomes more apparent with larger M values.

6) Averaging operations aid in noise reduction. By employing an averaging filter, random noise introduced into each pixel undergoes smoothing as the filter computes the average of adjacent pixel values. The efficacy of noise reduction depends significantly on the filter size, M. Larger M values exhibit superior noise reduction capabilities but concurrently lead to increased image blurring.

7) With larger M values, the blurring effect becomes notably pronounced. In conjunction with added noise, the image may appear distorted and markedly divergent from its original form.

8) Striking a balance between noise reduction and image blurring is crucial. Lower noise levels, such as b = 0.4, warrant a less aggressive filter, like M = 21, to mitigate noise without inducing substantial blurring. Conversely, higher noise levels, e.g., b = 0.9, necessitate a more potent filter. However, an excessively strong filter could induce excessive blurring, making a moderate choice, such as M = 31, more favorable.

Part 2

2.1.1) MATLAB Code 1 (Averaging without Noise)

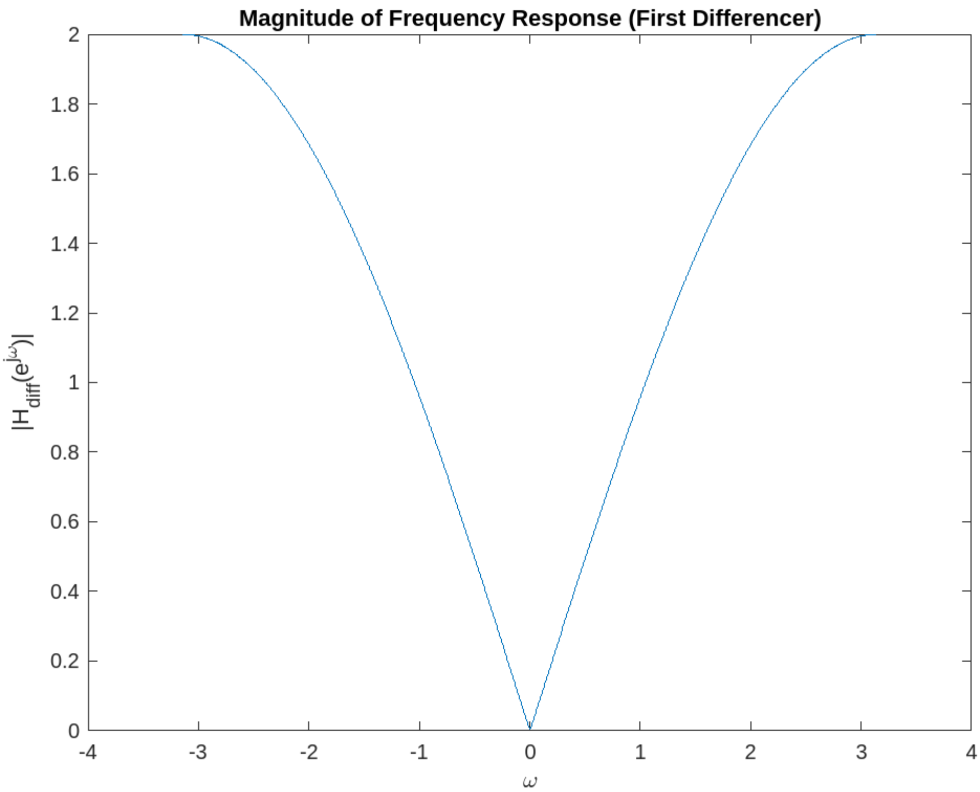
```
image = imread('Car.bmp');
imageMatrix = double(image);
grayscaleMatrix = mat2gray(imageMatrix, [0 255]);

firstDifFilter = [-1, 1, 0];
outImage = imfilter(grayscaleMatrix, firstDifFilter);

figure;
subplot(1, 2, 2);
imshow(grayscaleMatrix);
title('Original Image');

subplot(1, 2, 1);
imshow(outImage);
title('Output Image (First Differencer)');

omega = linspace(-pi, pi, 1000);
H_diff = freqz(firstDifFilter, 1, omega);

figure;
outImageNormalized = mat2gray(outImage);
imshow(outImageNormalized);
title('Normalized Output Image (First Differencer)');

figure;
plot(omega, abs(H_diff));
title('Magnitude of Frequency Response (First Differencer)');
xlabel('\omega');
ylabel('|H_{diff}(e^{j\omega})|');
```

2.1.2) Output 1: Filtered Images (First Differencer)

Output Image (First Differencer)

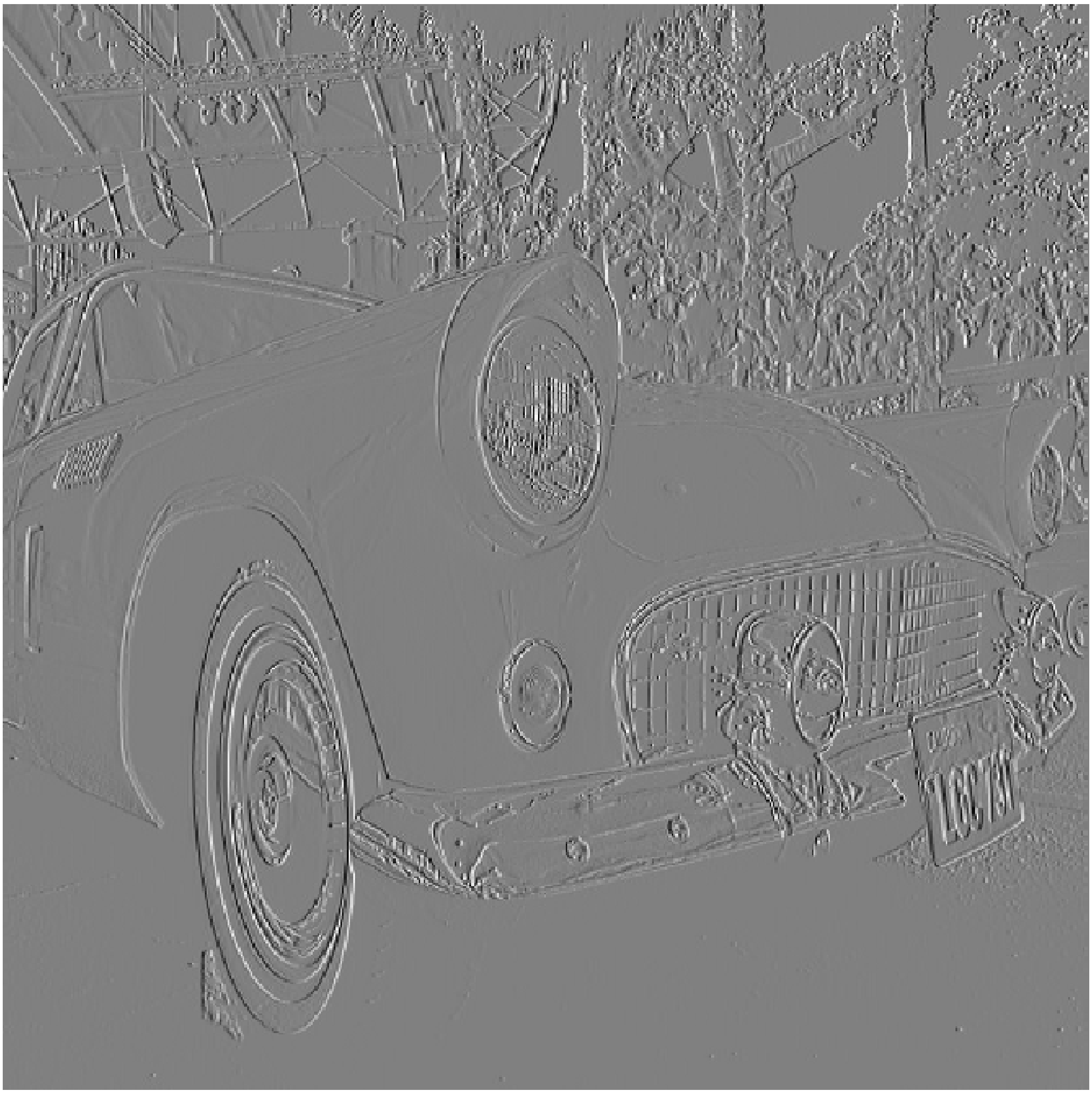
Original Image

2.1.3) Output 2: Magnitude of Frequency Response Plot



2.1.4) Output 3:Normalized Filtered Image(First Differencer)

Normalized Output Image (First Differencer)



2.2) Comments on the Outputs

1) The image predominantly appears black, revealing solely the faint outlines of the cat. This can be interpreted as a process that sifts out smoother regions of the picture, exposing the more distinct contrasts.

2) The initial difference filter, when applied horizontally, prioritizes accentuating sudden and sharp alterations within the image. It calculates the disparity between adjoining pixel values along each row, accentuating shifts in intensity. The outcome aligns with this principle: the original image lacks pronounced color or pattern variances, causing much of the image to be filtered out. Only the cat's outlines, where these marked differences manifest in distinguishing the cat from the background, are marginally discernible. However, this distinction might not be immediately evident in the output due to its darkened appearance. To facilitate clearer observation of the filter's impact, normalizing the output image to grayscale aids in discerning the filter's outcomes, exemplified in Output 3 at 2.1.3.

3) Analogous to the preceding illustration, the filter exclusively manipulates the q variable, representing the horizontal axis. Consequently, the filter processes sharp variances along the horizontal axis, neglecting the vertical axis. Consequently, in this instance, the vertical edges are highlighted. To emphasize horizontal edges, employing the filter on the p variable would be more suitable.

4) The graphical representation illustrates that the first difference filter accentuates higher frequencies while diminishing lower frequencies. In the image context, the visible outlines correlate with higher-frequency components, representing rapid changes, while the filtered-out regions correspond to lower-frequency components devoid of pronounced differences.