LAB 3

3.1 Deconvolution Experiment for 1-D Filters

```
clear
clc
% Definition of Signal and Filter
xx = 256*(rem(0:100, 50) < 10); % Input Signal
aa = 1; % Coefficient of output
bb = [1, -0.9]; % Coefficient of Input
ww = filter(bb, aa, xx); % Filtering
disp(ww); % Displaying

% Plotting
n = 0:75; % Range for Plotting
figure
subplot(2,1,1);
stem(n, xx(n+1), 'b', 'LineWidth', 1.5); % Input signal
title('Input Signal');
xlabel('n');
ylabel('x[n]');
xlim([0, 75]);
subplot(2,1,2);
stem(n, ww(n+1), 'r', 'LineWidth', 1.5); % Output signal
title('Output Signal');
xlabel('n');
ylabel('w[n]');
xlim([0, 75]);
```

In the code above, the length of the filtered signal $w[n]$ is stated by the length of the input signal $x[n]$ and the order of the FIR (Finite Impulse Response) filter coefficients. The length of the filtered signal w[n]w[n] can be determined using the formula:

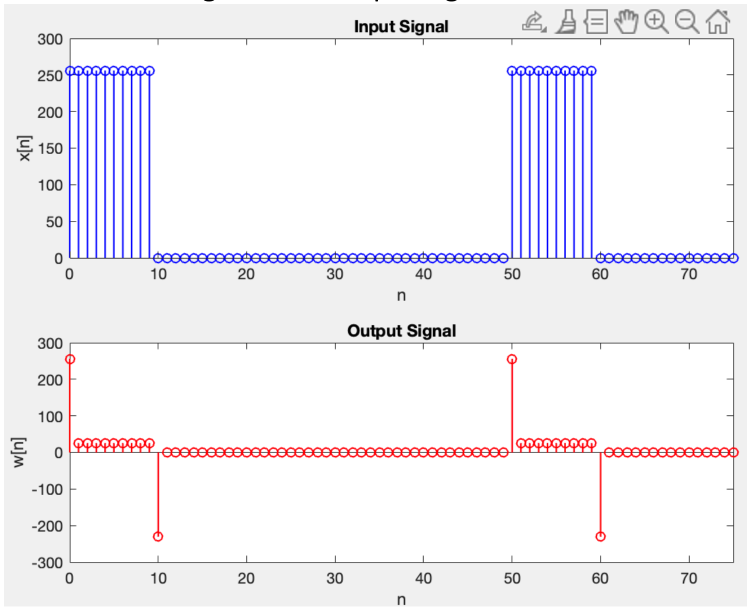
$$L_w = L_X + L_b - 1$$

Where:

- L_w is the length of the filtered signal $w[n]$.
- L_x is the length of the input signal $x[n]$.
- L_b is the length of the FIR filter coefficients.

In the MATLAB code, the length of the input signal $x[n]$ is 101 (from 0 to 100) due to it is generated by using the rem function which has a range of 0 to 100. The length of the FIR filter coefficients bb is 2 because covering two values: [1, -0.9]. Hereby, substituting the values into the formula:
$L_w = 101 + 2 - 1 = 102$
Thus, the length of the filtered signal $w[n]$ is 102.
The length of the filtered signal is interrelated to the length of the input signal and the length of the FIR filter coefficients since convolution, which is used in filtering, extends the length of the input signal. Particularly, the length of the last signal is the sum of the lengths of the input

signal and the FIR filter minus one. This is caused by the nature of linear convolution, where the output length is the sum of the lengths of the input signals minus one.



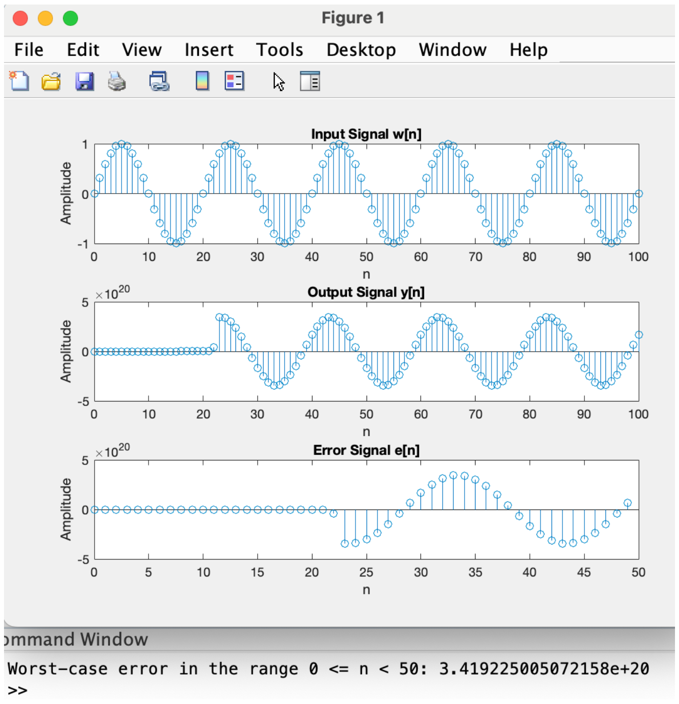
3.1.1 Deconvolution Experiment for 1-D Filters & 3.1.2 Worst Case Error

```
clear
clc
% Definition of Signal and Filter
xx = 256*(rem(0:100, 50) < 10); % Input Signal
aa = 1; % Coefficient of output
bb = [1, -0.9]; % Coefficient of Input
ww = filter(bb, aa, xx); % Filtering
disp(ww); % Displaying

% Plotting
n = 0:75; % Range for Plotting
figure
subplot(2,1,1);
stem(n, xx(n+1), 'b', 'LineWidth', 1.0); % Input signal
title('Input Signal');
xlabel('n');
ylabel('x[n]');
xlim([0, 75]);
subplot(2,1,2);
stem(n, ww(n+1), 'r', 'LineWidth', 1.0); % Output signal
title('Output Signal');
xlabel('n');
ylabel('w[n]');
xlim([0, 75]);
```

The error plot and worst-case error gives intelligibility into the quality of the restoration of $x[n]$. The error plot shows how much $y[n]$ deviates from the original signal $x[n]$ at each n index. The worst-case error shows the maximum absolute difference between $y[n]$ and $x[n]$

for a specified range, in this circumstance, $0 \leq n < 50$. A small worst-case error denotes that the restoration process is correct, as it shows that the output signal $y[n]$ is similar to the original signal $x[n]$ throughout the whole range. In converse, a worst-case error proposes that there are important deviations between the original and restored signals, determining a low quality of restoration.
The "visibility" of the error on a plot rely on different factors such as the scale of the plot, the nature of the signals, and the context of the implementation. Generally, if the worst-case error is very low, it may not be visibly distinguishible on a plot, especially if the scale of the plot is large or if the signals themselves have high amplitudes or important differences. If the worst-case error is lower than the amplitude variations or noise exist in the signals, it might not be particular on a plot.



```
Worst-case error in the range 0 <= n < 50: 3.419225005072158e+20
>>
```

3.1.3 An Echo Filter

```
clear
clc

fs = 8000; % Sampling frequency
duration = 2; % Signal time duration
t = 0:1/fs:duration-1/fs; % Time vector
f = 1000; % 1kHz frequency of sine wave
x1 = sin(2*pi*f*t); % Sine Wave
delay = 0.2; % Delay in seconds
P = round(delay * fs); % Time delay to samples
r = 0.9; % Echo strength
x1_delayed = [zeros(1,P), x1(1:end-P)]; % Delay signal generation with
shifting the signal by P samples
y = x1 + r * x1_delayed; % Final form of signal
```

```
% Plotting of Signals
figure
subplot(2,1,1);
plot(t, x1);
title('Original Signal');
xlabel('Time (s)');
ylabel('Amplitude');
xlim([0, 0.3]); % Limit x-axis to first 0.75 seconds
subplot(2,1,2);
plot(t, y);
title('Echoed Signal');
xlabel('Time (s)');
ylabel('Amplitude');
xlim([0, 0.3]); % Limit x-axis to first 0.75 seconds
% Play original and delayed signals
soundsc(x1,fs);
pause(2);
soundsc(x1_delayed,fs);
```

The system can be represented as an FIR filter with the following impulse response:

$$h[n] = \begin{cases} 1 \; if \; n = 0 \\ r \; if \; n = P \\ 0 \; otherwise \end{cases}$$

Where:

- r is the strength of the echo (0.9).
- P is the time delay in samples (1600).

Hence, the filter coefficients for this FIR filter are

$$h[n] = \begin{cases} 1 \; if \; n = 0 \\ 0.9 \; if \; n = 1600 \\ 0 \; otherwise \end{cases}$$

Filter has a length of 1601 samples, non-zero coefficients at $n = 0$ and $n = 1600$, and all other coefficients are zero.

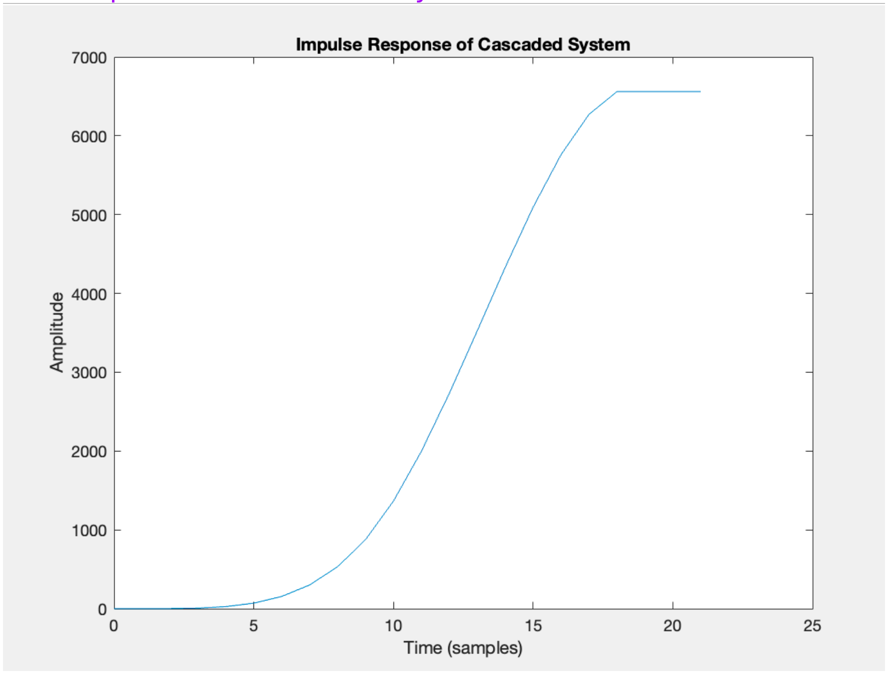
3.2.1 Overall Impulse Response

```
% Define the parameters of the system
q = 0:9;
r = 0:9;
M = 22;

% Define the filters
FIR1 = filter([1 1], [1], q);
FIR2 = filter([1 1], [1], r);
```

```
% Cascade the filters
Y = filter(FIR2, [1], filter(FIR1, [1], ones(1, M)));

% Plot the impulse response of the overall cascaded system
plot(0:M-1, Y)
xlabel('Time (samples)')
ylabel('Amplitude')
title('Impulse Response of Cascaded System')
```



3.2.2 Distorting and Restoring Images

```
clear
clc
%% Convolution of Image
echart = imread("meerkat.jpeg"); % Image Importing
echart = im2double(rgb2gray(echart)); % Converting Images to grayscale and
double

q = 0.9; % Define system parameter
filter_coeffs = [1 -q]; % Creating a FIR high-pass filter

% Apply FIR FILTER different orientations
filtered_image_horizontal = filter(filter_coeffs, 1, echart, [], 2);
filtered_image_vertical = filter(filter_coeffs, 1, filtered_image_horizontal,
[], 1);

% Other Operations
ech90 = filtered_image_vertical;
ech90 = ech90 - min(ech90(:));
ech90 = ech90 / max(ech90(:));

imshow(ech90); % Plotting the output
%% Deconv
```

```
q = 0.9; % Define system parameter
filter_coeffs = [1 -q]; % Creating a FIR high-pass filter


% Apply FIR FILTER different orientations
filtered_image_horizontal = filter(filter_coeffs, 1, echart, [], 2);
filtered_image_vertical = filter(filter_coeffs, 1, filtered_image_horizontal,
[], 1);

M = 22; % Length of the deconvolution filter
r = 0.9; % Decay factor of the deconvolution

% Generating and performing of deconvolution filter
deconv_filter = r.^(0:M);
recovered_image = conv2(filtered_image_vertical, deconv_filter, 'full');

% Other Operations
recovered_image = recovered_image(1:size(echart, 1), 1:size(echart, 2));
recovered_image = recovered_image - min(recovered_image(:));
recovered_image = recovered_image / max(recovered_image(:));

% Display the recovered image
figure
imshow(recovered_image);
```

The output of the deconvolution process might represent numerous visual characteristics, and comprehending these properties requires the mathematics of the cascade filtering process.

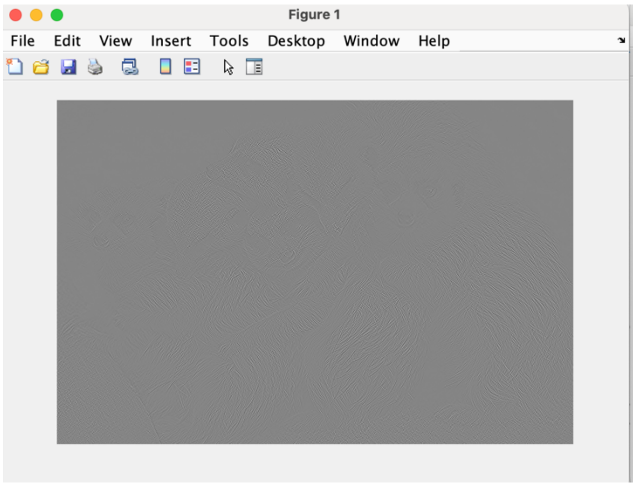
1.Visual Appearance: The output image may reveal blurred or distorted outputs compared to the original image. This blurring impact is the deconvolution process to get the original image from the filtered signal. However, due to noise, errors in the filtering process, and the finite length of the deconvolution filter, the recovered image might not fully meet the original.

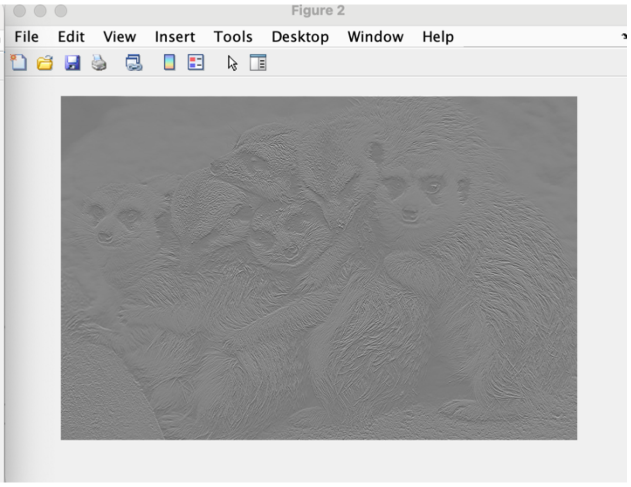
2.Ghosts or Echoes: The output image can display "ghosts" or echoes of high-contrast edges on the original image. These ghosts occur due to the finite length of the deconvolution filter and the features of the convolution operation. When a high-contrast edge is convolved with the filter, echoes can be presented that are relevant to the impulse response of filter.

3.Size and Location of Ghosts: The size and location of the ghosts can be expressed by analyzing the impulse response of the deconvolution filter. In this circumstance, the deconvolution filter is a geometric series with a decay factor $r = 0.9$ and a length $M = 22$. The ghosts will be located around high-contrast edges in the image and will extend up to M pixels away from the edge.

4.Worst-Case Error: The worst-case error can be evaluated by looking for the maximum intensity alteration through all edges in the original image. In a "black-white" transition, the intensity changes from 0 to 255. The ghosts will show additional intensity alteration in the output image, which can be compared to the original intensity change to indicate their relative size. The error can be calculated as the difference between the intensity changes caused by the ghosts and the original intensity differences.

By analyzing these elements, we can get virtue into the visual appearance of the output image, comprehending the presence of ghosts, and evaluate the extent of errors caused by the deconvolution process relative to the original image.





3.2.3 A Second Restoration Experiment

```
clear
clc
echart = imread("meerkat.jpeg"); % Image Importing
echart = im2double(rgb2gray(echart)); % Converting Images to grayscale and
double

q = 0.9; % Define system parameter
filter_coeffs = [1 -q]; % Creating a FIR high-pass filter

% Apply FIR FILTER different orientations
filtered_image_horizontal = filter(filter_coeffs, 1, echart, [], 2);
filtered_image_vertical = filter(filter_coeffs, 1, filtered_image_horizontal,
[], 1);


M_values = [1, 5, 8, 11, 22, 33]; % % Defining deconvolution parameters with
length of the filter
r = 0.9; % Decay factor of the deconvolution

for M = M_values
    deconv_filter = r.^(0:M); % Deconvolution filter
    recovered_image = conv2(filtered_image_vertical, deconv_filter, 'full');
% Performing deconvolution
    recovered_image = recovered_image(1:size(echart, 1), 1:size(echart, 2));
% Trimming

    % Other Operations
    recovered_image = recovered_image - min(recovered_image(:));
    recovered_image = recovered_image / max(recovered_image(:));

    % Plotting
    figure
    imshow(recovered_image);
    title(['Recovered Image (M = ', num2str(M), ')']);
end
```

To understand why $M = 33$ is the best way for the deconvolution filter, for more detailed inofrmation about content is stated below.

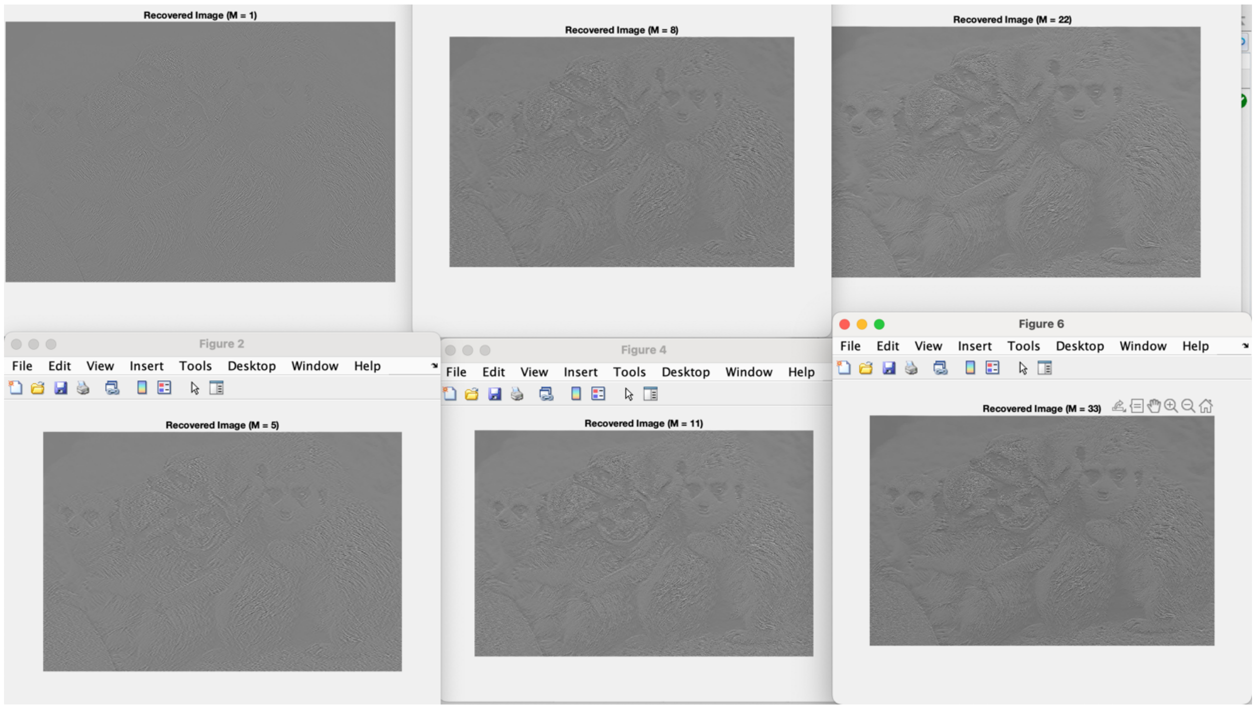
1. Cascaded Filtering Process:
- The image is firstly filtered with a high-pass FIR filter in both horizontal and vertical directions.
- Then, deconvolution is implemented to recover the original image.
- Deconvolution is changing the filtering effects caused by the high-pass filter.

2.Impulse Response of Cascaded System:
- The impulse response of the cascaded system can be showed as the convolution of the impulse response of the FIR filter with the impulse response of the deconvolution filter.
- The impulse response of a high-pass filter normally occurs with a central impulse by smaller impulses of opposite mark.
- The deconvolution filter is designed to amplify the central impulse while suppressing the other impulses.

3.Visual Appearance of Output Image:
- With a proper decision of M, the deconvolution filter can amplify the central features of the image while getting rid of artifacts caused by the filtering process.
- The recovered image should present better details and edges, with decreasing blurring and artifacts.
- As M increases, the deconvolution filter has a longer length, providing better avoiding of unwished details and better improvement of the original image features.

4.Explanation of Why $M = 33$ is the Best:
- With $M = 33$, the deconvolution filter has long length to suppress unwished details while covering image properties.
- Shorter lengths might not properly suppress artifacts, causing to a less correct recovery of the original image.
- Longer lengths can introduce over-smoothing or ringing distortions, reducing the image quality of the recovered image.

5.Conclusion:
- By choosing $M = 33$, the deconvolution filter balances preservation rate of image details, resulting in the better output image quality.
- The appearance of the output image should enhance details, sharp edges, and reduced blurring or artifacts respect to images with shorter or longer deconvolution filters.


Recovered Image (M = 1)
Recovered Image (M = 8)
Recovered Image (M = 22)
Figure 2
File Edit View Insert Tools Desktop Window Help
Recovered Image (M = 5)
Figure 4
File Edit View Insert Tools Desktop Window Help
Recovered Image (M = 11)
Figure 6
File Edit View Insert Tools Desktop Window Help
Recovered Image (M = 33)