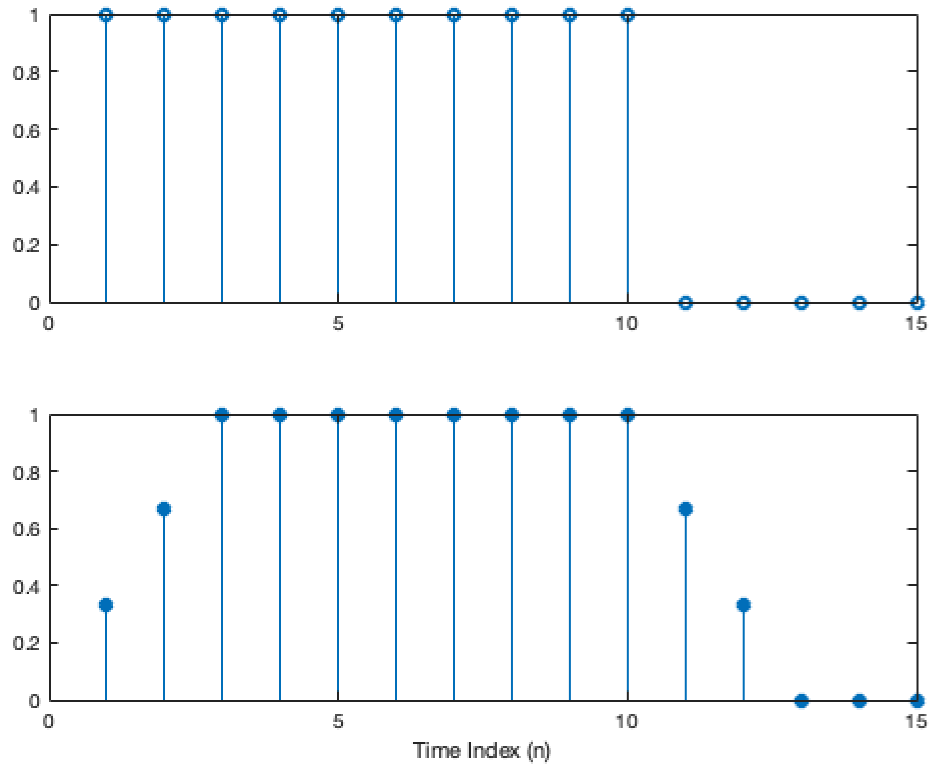
Lab 2 Part 1

Table of Contents

1.1 - Implement the 3-point Averager

1. There are 2 different functions to implement a FIR filter:

```
xx =[ones(1,10),zeros(1,5)]; %<--Input signal
nn = 1:length(xx); %<--Time indices
bk = [1/3 1/3 1/3]; %<--Filter coefficients
yy = firfilt(bk, xx); %<--Compute the output
yy = conv(bk, xx); %<--Equivalent method to compute output
figure;
clf
subplot(2,1,1);
stem(nn, xx(nn))
subplot(2,1,2);
stem(nn, yy(nn), 'filled')
xlabel('Time Index (n)')
```




```
Question 1: What characteristics of the input signal are most affected
by the averager? Least affected?

The characteristics of the input sigal that are the most affected by the
averager include the points near the beginning of the signal and the
points near the end of the signal. The characteristics that are the
least affected include the points near the middle of the signal.
Magitude of each of the points is relatively similar to the input
signal. The output signal is smoothed out around the edges in comparison
to the input signal.

Question 2: What is the relationship between the lengths of input
vector, output vector, and coefficient vector?

The relationship between the lengths of the input vector, output vector,
and coefficient vector can be explained by len(output vector) =
len(input vector) + len(coefficient vector) - 1.
```

1.2 - Echo Filter

```
load labdat.mat
```

1. Suppose you have an audio signal sampled at fs = 8000 and you want to simulate an echo.

Question 3: What values of r and P will give an echo with strength 85% of the original, with time delay 0.22s?

```
fs = 8000;
r = 0.85
p = .22 * 8000
```

```
r =

    0.8500


p =

        1760
```

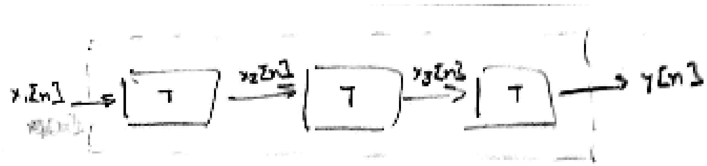
```
soundsc(x2, fs)

type echo_filter

yy = echo_filter(x2,r,p);
soundsc(yy,fs)
```

```
function [yy] = echo_filter(xx,r,p)
    % ECHO_FILTER - Applies an echo given the magnitude of an echo r
 and a
    % time delay p to an input signal xx
    yy = zeros(length(xx) + p, 1);
    yy(1:length(xx)) = xx;
    yy(p:length(yy)-1) = yy(p:length(yy)-1) + r .* xx;
end
```

2. Multiple echos can be accomplished by cascading several echo filters of the above form.

Derive (by hand) the impulse response of a reverb system produced by cascading three "single echo" systems.

```
pic = imread('impulse-calculation.png');
figure;
imshow(pic);
```



$$x_2[n] = x_1[n] + r x_1[n-p]$$
$$\begin{aligned} x_3[n] &= x_2[n] + r x_2[n-p] \\ &= (x_1[n] + r y_1[n-p]) + r(x_1[n-p] + r x_1[n-2p]) \\ &= x_1[n] + r x_1[n-p] + r x_1[n-p] + r^2 x_1[n-2p] \\ &= y_1[n] + 2r x_1[n-p] + r^2 x_1[n-2p] \end{aligned}$$

$$\begin{aligned} y[n] &= x_3[n] + r x_3[n-p] \\ &= (x_1[n] + 2r x_1[n-p] + r^2 x_1[n-2p]) + \\ &\quad r(x_1[n-p] + 2r x_1[n-2p] + r^2 x_1[n-3p]) \\ &= x_1[n] + 2r x_1[n-p] + r^2 x_1[n-2p] + \\ &\quad r x_1[n-p] + 2r^2 x_1[n-2p] + r^3 x_1[n-3p] \end{aligned}$$

$$y[n] = x_1[n] + 3r x_1[n-p] + 3r^2 x_1[n-2p] + r^3 x_1[n-3p]$$

$$h[n] = \delta[n] + 3r \delta[n-p] + 3r^2 \delta[n-2p] + r^3 \delta[n-3p]$$

```
yy3 = echo_filter(echo_filter(echo_filter(x2,r,p),r,p),r,p);
soundsc(yy3,fs);
audiowrite('xx_multi-echo_filter.wav',yy3/max(yy3),fs);
```

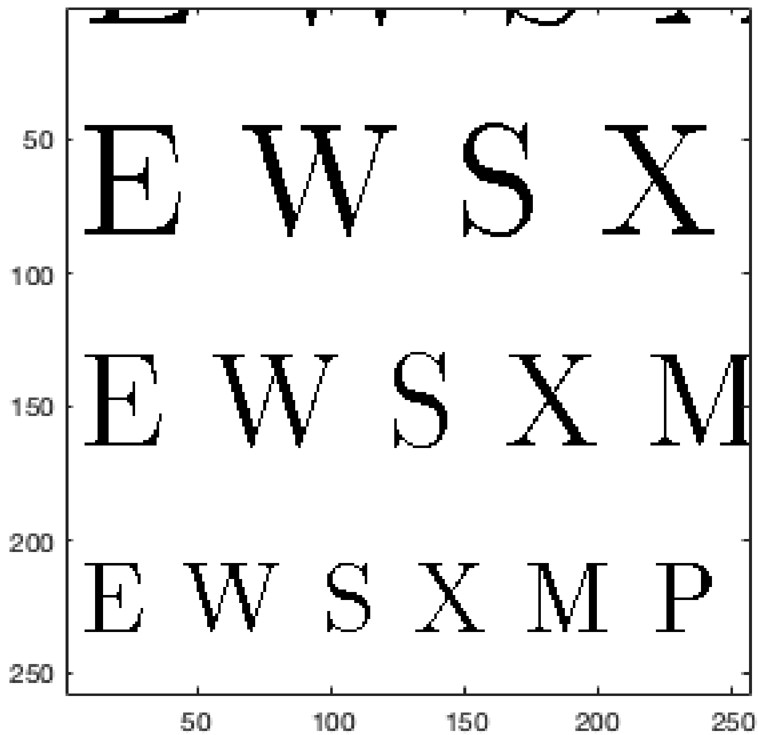
Warning: Data clipped when writing file.

1.3 - Image Processing

1. Show a Test Image

```
load echart.mat;
show_img(echart);
```

Image being scaled so that min value is 0 and max value is 255

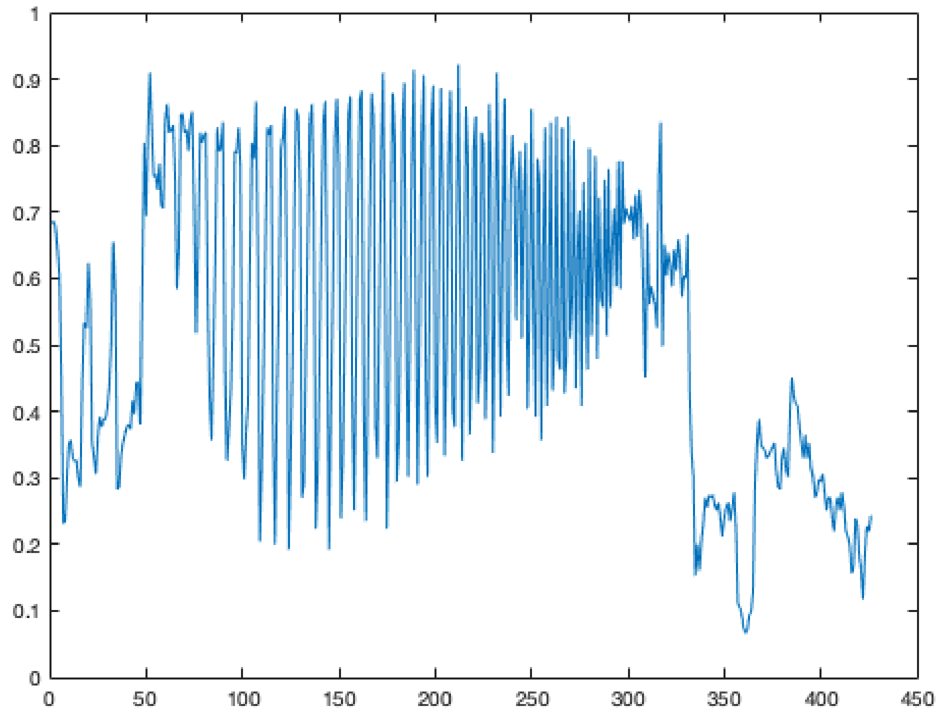


2. The Lighthouse Image

```
load lighthouse.mat;
show_img(xx);
```

Image being scaled so that min value is 0 and max value is 255


```
xx225 = xx(225,:);
plot(xx225);
```



Question 4: Which values represent white? Black?

The value 1 corresponds to white and the value 0 corresponds to black. This is supported by my annotation above, where one can observe that there is heavy oscillation in the plot of the 225th row between ~75 and ~300, correspoding to the white picket fence offset by the black backgroud. Additionally, one can observe that the signal drops in value, taking on a value closer to 0 from ~325 until the end. This matches closely with the darker colors seen in the grass and viewing post near the right side of the image.

Question 5: Where does the 225th row cross the fence?

The 225th row crosses the fence at ~325, as that is when the value sharply drops off in the plot of the 225th row.

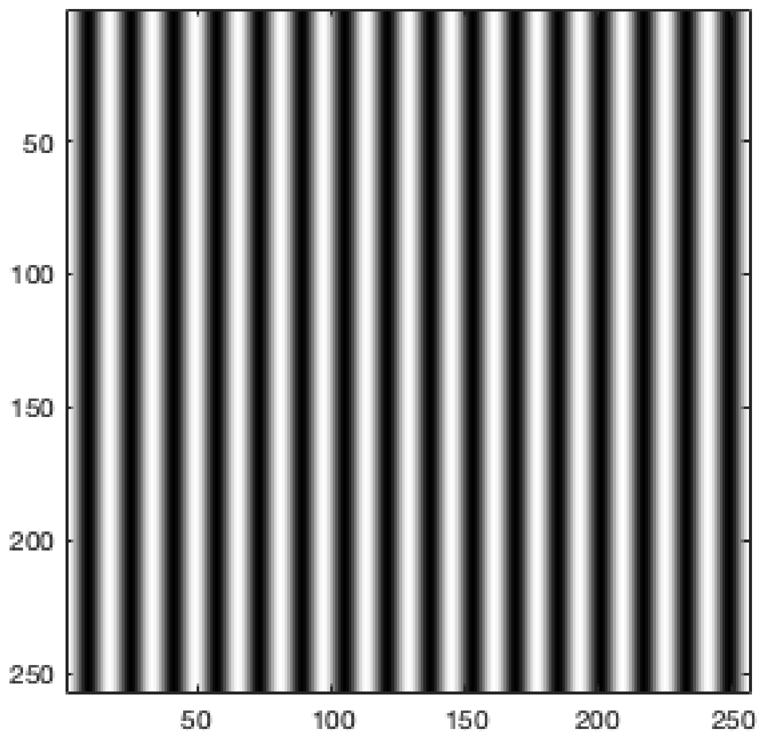
Question 6: What features of the image correlate with the periodic-like portion of the plot?

The feature of the image that correlates with the periodic-like portion of the plot is the white picket fence, where one can observe that the color oscillates between the white fence and the dark background.

3. Synthesize a Test Image

```
xpix = ones(256,1)*cos(2*pi*(0:255)/16);
show_img(xpix);
```

Image being scaled so that min value is 0 and max value is 255



Question 7: How wide are the bands in number of pixels? How is this width related to the formula for xpix?

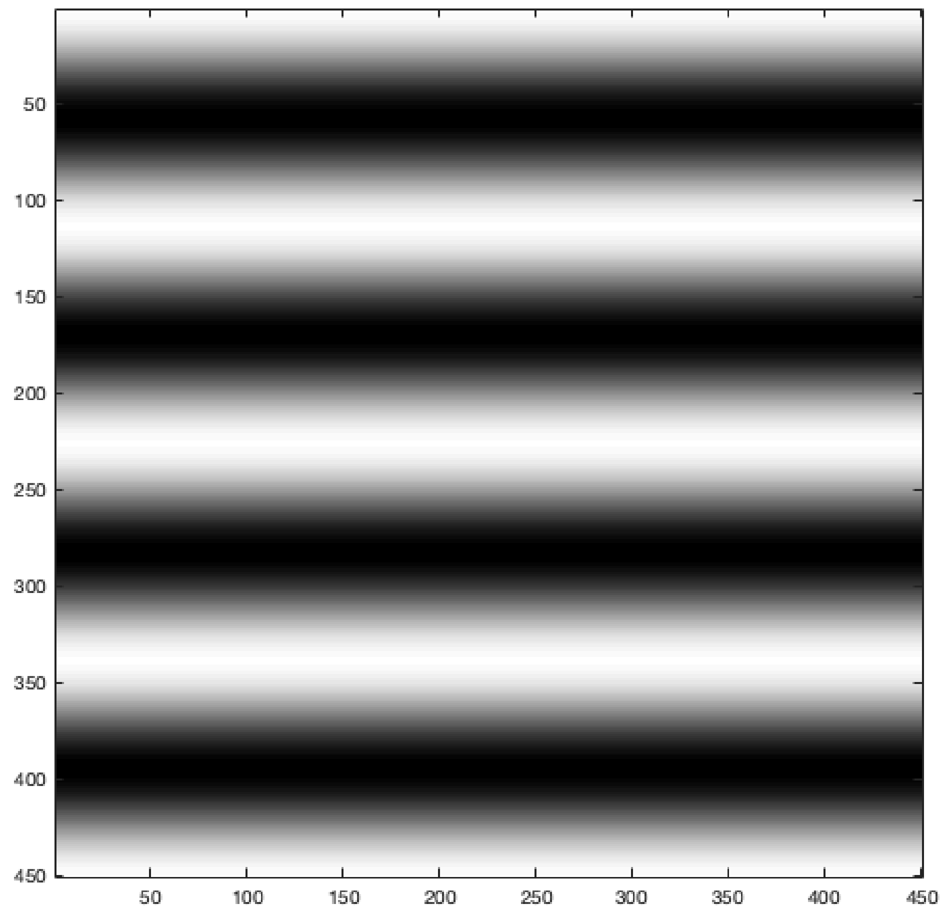
The width of these bands in number of pixels is 16. This is because the period of the sinusoid that is multiplied with the vector of ones to generate the output sigal is divided by 16.

Question 8: How would you produce an image with horizontal bands?

You could produce a image with horizontal bands by transposing the generated matrix.

```
xpix2 = transpose(ones(450,1)*cos(2*pi*(0:449)/450*4));
show_img(xpix2);
```

Image being scaled so that min value is 0 and max value is 255



4. Sampling of Images

```
xx_downsampled = xx(1:2:end,1:2:end);
show_img(xx_downsampled);
```

Image being scaled so that min value is 0 and max value is 255

Question 9: What is the size of the down-sampled image?

The new down-sampled image is 1/4 of the size of the original image, as both the x and the y dimensions are half the size of the original image and area is proportional to the product of these two dimensions.

```
size(xx_downsampled)
```

```
ans =

   163   213
```

Question 10: Describe how the aliasing appears visually.

The aliasing appears blocky and less detailed in some parts of the image. Some parts of the image show the aliasing more obviously than others.

Question 11: Which parts of the image most dramatically show the effects of aliasing? Why does the aliasing manifest itself in these places?

The parts of the image that previously had lots of detail more dramatically show the effects of aliasing brought on by the down-sampling. This is because when you lose samples due to down sampling, you tend to lose the information that provides detail to the image.

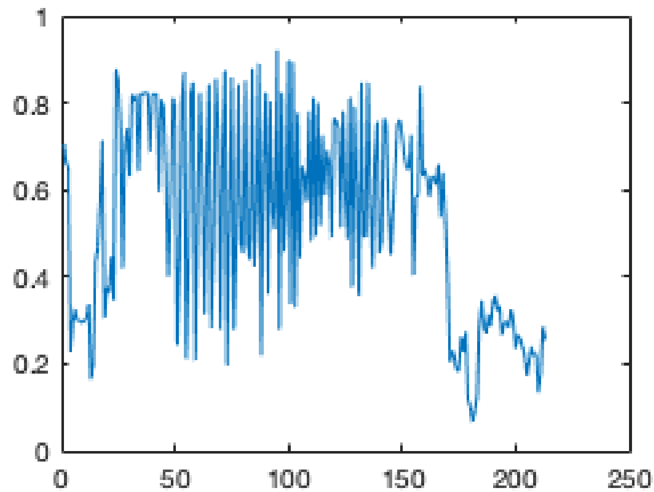
Question 12: From your row plot and from zooming in on the image, estimate the frequency of the aliased features in cycles per pixel. When estimating spatial frequency, consider recurring features of the images as ?peaks?. From this you can obtain a period (how many pixels until the image ?peaks? again), and from there a frequency.

16 cycles / 50 pixels. I counted 16 cycles in the time period from pixel 50 to pixel 100.

```
freq = 16/50
```

```
freq =

    0.3200
```

```
xx_downsampled225 = xx_downsampled(floor(225/2),:);
plot(xx_downsampled225);
```



Question 13: How does your estimation of aliased features fit into the Sampling Theorem?

My estimation of the aliased features fit quite well with the Sampling Theorem, as one can observe that the areas that experience the highest amount of aliasig is the fence, which corresponds to the area that has the highest frequecy in the row plot.

Published with MATLAB® R2019a