

Chapter 2

Sensors & Actuators

2.1 Problems

Problem 2.1 (Music icon address) What screen-row-column address would the controller assign to the music icon shown in Figure 2.10 if the icon is located on the third screen of 16 possible screens?

(ans: Sixteen screens have 4-bit addresses from 0000 (first screen) to 1111 (sixteenth screen). The third screen has address 0010, giving the music icon the screen-row-column address 0010-10-01.

)

Problem 2.2 (Calculator switch array) A scientific calculator has 50 keys for digits and logarithmic and trigonometric functions arranged in five rows and ten columns. Specify a binary address code to indicate what key was pressed.

(ans: Five rows require a 3-bit code and ten columns a 4-bit code. Hence, each key has a 7 bit address.

)

Problem 2.3 (Forming a touch screen switch array) A touch screen array has a count of rows and columns that sums to 10. What is the structure of the array that accommodates the maximum number of keys?

(ans: Five rows and five columns accommodate 25 switch locations.

)

Problem 2.4 (Finger swipe along a switch array) Extending Example 2.7, a linear switch array is 10 cm long and has a resolution $R_s = 2$ switches/mm. A swipe motion is detected if the mid-point changes by more than 8 switches. If the sampling period $T_s = 0.1$ s, what is the minimum finger swipe speed along the linear array that indicates a swipe motion?

(ans: $\delta x_m = 8$ switches, $R_s = 2$ switches/mm, and $T_s = 0.1$ s, gives

$$v_{min} = \frac{\delta x_m}{R_s T_s} = \frac{8 \text{ switches}}{2 \text{ switches/mm} \times 0.1 \text{ s}} = 40 \text{ mm/s}$$

)

Problem 2.5 (Multiple finger gesture) Extending Example 2.8, a linear switch array is 10 cm long and has a resolution $R_s = 2$ switches/mm, and sampling period $T_s = 0.1$ s. If $P1 = 20$ and $P2 = 40$ and $P1' = 22$ and $P2' = 36$ is sensed as a gesture, what is the finger swipe speed? Is it widening or spreading?

(ans:

$$S = P2 - P1 = 40 - 20 = 20$$

$$\frac{S}{R_s} = \frac{20 \text{ switches}}{2 \text{ switches/mm}} = 10 \text{ mm}$$

$$S' = P2' - P1' = 36 - 22 = 14$$

$$\frac{S'}{R_s} = \frac{14 \text{ switches}}{2 \text{ switches/mm}} = 7 \text{ mm}$$

Hence, the finger separation narrows. The widening separation speed is

$$v_F = \frac{S'/R_s - S/R_s}{T_s} = \frac{7 \text{ mm} - 10 \text{ mm}}{0.1 \text{ s}} = -30 \text{ mm/s}$$

Equivalently, the finger narrowing separation speed is 30 mm/s.

)

Problem 2.6 (Number of bits in a large color LED display) A large color billboard is a two-dimensional array of $2^{10} \times 2^{10}$ pixels, with each pixel containing red, green and blue LEDs. (Single LED packages contain separate R, G, and B LEDs inside.) Assuming that each LED is controlled to shine at one of 256 levels, how many bits are needed to specify a color image on the billboard? How many different colors can each 3-LED pixel display?

(ans: Number of LEDs is

$$2^{10} \times 2^{10} \times 3 = 3 \times 2^{20} (\approx 3 \text{ million})$$

256 levels are set by 8 bits ($= 2^3$), so the total number of bits per image equals 3×2^{23} , or 24 million bits. The number of colors that each 3-LED pixel can display equals

$$2^{24} = 2^4 \times 2^{20} = 16 \times 10^6 (16 \text{ million colors})$$

)

Problem 2.7 (Number of possible images in a large color LED display) A large color billboard is a two-dimensional array of $2^{10} \times 2^{10}$ pixels, with each pixel containing red, green and blue LEDs. Assuming that each LED is controlled to shine at one of 256 levels, what is the number of different images that can be displayed? Express answer as a power of 10.

(ans: The number of colors each RGB pixel can display equals 16×10^6 . The number of different possible images that 10^6 pixels can display is

$$16 \times 10^6 \times 10^6 = 16 \times 10^{12}$$

or 16 trillion images.

)

Problem 2.8 (Bit rate to generate a full-screen movie) A video game displays images on your laptop monitor having a resolution of 1680×1050 pixels. Each pixel contains a red, green, and blue LEDs, and each LED is controlled to shine at one of 256 levels. The game produces a new image on the screen 60 times per second. How many bits per second are being sent to your monitor while you are playing your game? Give answer in scientific notation ($x.xx \times 10^y$).

(ans: Number of LEDs equals

$$1,680 \times 1,050 \times 3 = 5.20 \times 10^6 \text{ LEDs/frame}$$

256 levels per LED are set by 8 bits, so the total number of bits per frame equals

$$8 \text{ bits/LED} \times 5.20 \times 10^6 \text{ LEDs/frame} = 4.16 \times 10^7 \text{ bits/frame}$$

At 60 frames per second, the bit rate equals

$$60 \text{ frames/s} \times 4.16 \times 10^7 \text{ bits/frame} = 2.50 \times 10^9 \text{ bits/s}$$

)

Problem 2.9 (Smartphone location from two range measurements) This problem considers the location information using the range values measured by two antennas. Let antennas A1 and A2 be located 5 km apart. Determine the two possible locations for the smartphone relative to antenna A1 when the smartphone range from A1 is 3 km and from A2 is 3.5 km.

(ans: Let A1 be at (0,0), and A2 at (5,0) km. Smartphone location is (x_S, y_S) . Then, A1 range value $R_1 = 3$ gives

$$x_S^2 + y_S^2 = 9$$

A2 range value $R_2 = 3.5$ gives

$$(x_S - 5)^2 + y_S^2 = 12.25$$

Equating both to y_S^2 gives

$$y_S^2 = 9 - x_S^2 = 12.25 - (x_S - 5)^2 \quad [= 12.25 - (x_S^2 - 10x_S + 25)]$$

Canceling x_S^2 and solving for x_S yields

$$x_S = \frac{9 - 12.25 + 25}{10} = 2.175 \text{ km}$$

The two solutions for y_S come from the A1 equation

$$y_S = \pm \sqrt{9 - x_S^2} = \pm \sqrt{9 - 4.730625} = \pm \sqrt{4.269375} = \pm 2.066 \text{ km}$$

)

Problem 2.10 (Smartphone location region caused by range errors) Sketch and determine the four points defining the region that contains your smartphone when the range measured from antenna A1 is (3 ± 0.1) km and that from A2 is (3.5 ± 0.1) km.

(ans: Let A1 be at $(0,0)$, and A2 at $(5,0)$ km. Smartphone location is (x_S, y_S) . Consider solutions due to positive (+) and negative (-) errors with the four cases $(++)$, $(+-)$, $(-+)$, $(--)$.

First $(++)$, A1 range value $R_1 = 3.1$ and A2 range value $R_2 = 3.6$ give

$$x_{S++} = \frac{9.61 - 12.96 + 25}{10} = 2.17 \text{ km}$$

The two solutions for y_S come from the A1 equation

$$y_{S++} = \pm \sqrt{9.61 - 2.17^2} = \pm 2.22 \text{ km}$$

For $(+-)$, A1 range value $R_1 = 3.1$ and A2 range value $R_2 = 3.4$ give

$$x_{S+-} = \frac{9.61 - 11.56 + 25}{10} = 2.31 \text{ km}$$

The two solutions for y_S come from the A1 equation

$$y_{S+-} = \pm \sqrt{9.61 - 2.31^2} = \pm 2.07 \text{ km}$$

For $(-+)$, A1 range value $R_1 = 2.9$ and A2 range value $R_2 = 3.6$ give

$$x_{S-+} = \frac{8.41 - 12.96 + 25}{10} = 2.05 \text{ km}$$

The two solutions for y_S come from the A1 equation

$$y_{S-+} = \pm \sqrt{8.41 - 2.05^2} = \pm 2.06 \text{ km}$$

For $(--)$, A1 range value $R_1 = 2.9$ and A2 range value $R_2 = 3.4$ give

$$x_{S--} = \frac{8.41 - 11.56 + 25}{10} = 2.19 \text{ km}$$

The two solutions for y_S come from the A1 equation

$$y_{S--} = \pm \sqrt{8.41 - 2.19^2} = \pm 1.91 \text{ km}$$

)

Problem 2.11 (Pulse time for a bar code scan) In Example 2.21, if a laser spot moves across the bar code at 10 m/s, and the width of the thinnest bar is 1 mm, what is the duration of the shortest pulse produced by the scanner? Give answer in μs (10^{-6} s).

(ans:

$$t = \frac{d}{v} = \frac{10^{-3} \text{ m}}{10 \text{ m/s}} = 10^{-4} \text{ s} = 100 \mu\text{s}$$

)

Problem 2.12 (IR range sensor) In an IR autofocus camera, the emitter and detector are separated by 1 cm and positioned 1 cm behind the lenses, which are modeled as pinholes. The light reflected from an object produces a spot 1 mm from the centerline of the detector pinhole. What is the range of the object from the camera in meters (m)?

(ans: With $s = 10^{-2} m$, $f = 10^{-2} m$, $x = 10^{-3} m$ gives

$$r = \frac{sf}{x} = \frac{10^{-2} m \times 10^{-2} m}{10^{-3} m} = 0.10 m$$

)

Problem 2.13 (Digital IR range sensor) In a digital IR autofocus camera, the emitter and detector are 1 cm apart and the detector array is 1 cm behind the lens. An IR detector element has near and far limits $x_F = 0.01 mm$ and $x_N = 0.02 mm$ that senses light reflected from an object located from r_N to r_F in range. Determine the values of r_N and r_F in m.

(ans: $s = 10^{-2} m$, $f = 10^{-2} m$, $x_F = 10^{-5} m$ gives

$$r_F = \frac{sf}{x_F} = \frac{10^{-2} m \times 10^{-2} m}{10^{-5} m} = 10 m$$

$s = 10^{-2} m$, $f = 10^{-2} m$, $x_N = 2 \times 10^{-5} m$ gives

$$r_N = \frac{sf}{x_N} = \frac{10^{-2} m \times 10^{-2} m}{2 \times 10^{-5} m} = 5 m$$

)

Problem 2.14 (Digital IR range sensor dimensions) In a digital IR autofocus camera, the emitter and detector are 1 cm apart and the detector array is 1 cm behind the lens. What are the detector element's near and far limits (x_F and x_N) that senses light reflected from an object located 1 m to 4 m away? Give answer in millimeters (mm).

(ans: $s = 10^{-2} m$, $f = 10^{-2} m$, $r_F = 4 m$ gives

$$x_F = \frac{sf}{r_F} = \frac{10^{-2} m \times 10^{-2} m}{4 m} = 0.25 \times 10^{-4} m = 0.025 \times 10^{-3} m = 0.025 mm$$

$s = 10^{-2} m$, $f = 10^{-2} m$, $r_N = 1 m$ gives

$$x_N = \frac{sf}{r_N} = \frac{10^{-2} m \times 10^{-2} m}{1 m} = 10^{-4} m = 0.1 \times 10^{-3} m = 0.1 mm$$

)

Problem 2.15 (Sonar ranging - range to TOF) A sonar system operates in air up to a maximum range of 4 m. What is the maximum TOF? Give answer in ms (10^{-3} s)?

(ans: $c = 343$ m/s gives

$$TOF_{max} = \frac{2r_{max}}{c} = \frac{8 \text{ m}}{343 \text{ m/s}} = 0.023 \text{ s} = 23 \text{ ms}$$

)

Problem 2.16 (Sonar ranging - TOF to range) A sonar system observes a $TOF = 10$ ms. What is the object range in meters (m)?

(ans: $c = 343$ m/s gives

$$d = \frac{c \times TOF}{2} = \frac{343 \text{ m/s} \times 10^{-2} \text{ s}}{2} = 1.72 \text{ m}$$

)

Problem 2.17 (Sonar ranging resolution) A sonar system experiences a jitter in the echo arrival time because of dynamic temperature variations in air, which limits the TOF resolution to $\Delta TOF = \pm 50 \mu\text{s}$. What is the corresponding sonar range resolution Δr in mm?

(ans: $c = 343$ m/s gives

$$\Delta d = \frac{c \times \Delta TOF}{2} = \frac{343 \text{ m/s} \times \pm 5 \times 10^{-5} \text{ s}}{2} = \pm 858 \times 10^{-5} \text{ m} = 8.58 \text{ mm}$$

)

Problem 2.18 (Radar ranging - range to TOF) A radar system operates up to a maximum range of 100 m. What is the maximum TOF?

(ans: $c = 3 \times 10^8$ m/s gives

$$TOF_{max} = \frac{2r_{max}}{c} = \frac{200 \text{ m}}{3 \times 10^8 \text{ m/s}} = 66.7 \times 10^{-8} \text{ s} = 0.667 \mu\text{s}$$

)

Problem 2.19 (Radar ranging - TOF to range) A radar system observes a $TOF = 0.1 \mu\text{s}$. What is the object range in meters (m)?

(ans: $c = 3 \times 10^8$ m/s gives

$$r = \frac{c \times TOF}{2} = \frac{3 \times 10^8 \text{ m/s} \times 10^{-7} \text{ s}}{2} = 15 \text{ m}$$

)

Problem 2.20 (Radar ranging resolution) A radar system is specified to have a range resolution of $\pm 0.1\text{m}$. What is the corresponding resolution in the radar TOF?

(ans: $c = 3 \times 10^8 \text{ m/s}$ gives

$$\Delta TOF = \frac{2\Delta r}{c} = \frac{\pm 2 \times 10^{-1} \text{ m}}{3 \times 10^8 \text{ m/s}} = \pm 0.67 \times 10^{-9} \text{ s} = \pm 6.7 \times 10^{-10} \text{ s} (\pm 0.67 \text{ ns})$$

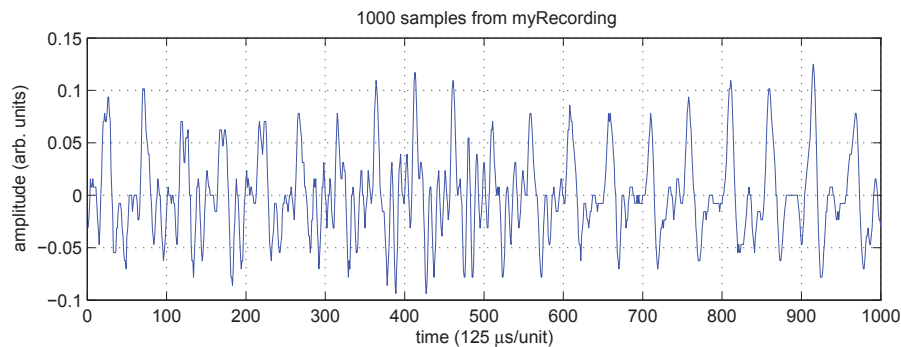
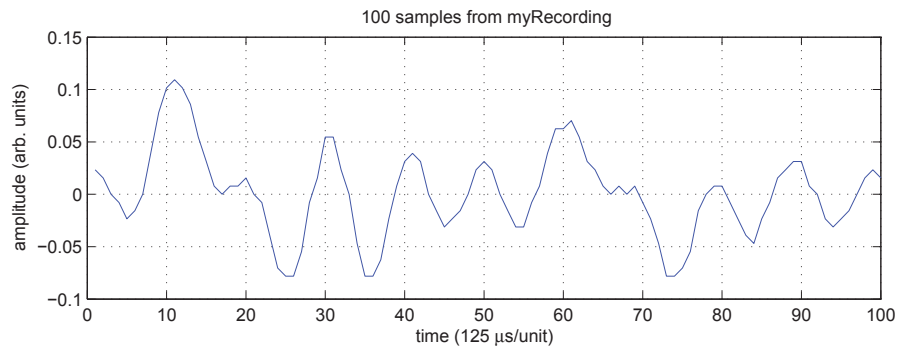
)

2.2 Matlab Projects

Project 2.1 (Acquire microphone speech signal) *Using the Matlab script in Example 16.9 as a guide, acquire speech data from the microphone on your laptop and display 100-sample and 1,000-sample waveforms.*

(ans:

```
% Microphone_input & speaker output
clear % clears workspace
clf % clears figures
recObj = audiorecorder(8000, 8, 1); % define ADC specs
disp('Start speaking now') % prompt speaker
recordblocking(recObj, 2); % record for 2 sec
disp('End of recording'); % indicate end
play(recObj); % playback recording.
myRecording = getaudiodata(recObj); % form data array
nr = length(myRecording)/2; % middle of array
subplot(2,1,1),plot(myRecording(nr-50:nr+49)); % Plot 100 samples from middle.
grid on
title(' 100 samples from myRecording')
xlabel('time (125 \mus/unit)')
ylabel('amplitude')
subplot(2,1,2),plot(myRecording(nr-500:nr+499)); % Plot 1000 samples from middle.
grid on
title(' 1000 samples from myRecording')
xlabel('time (125 \mus/unit)')
ylabel('amplitude')
```



)

Project 2.2 (Having fun with speech) Write a Matlab program that plays acquired microphone speech normally and after a one second pause backwards, that is, in time-reversed order.

(ans:

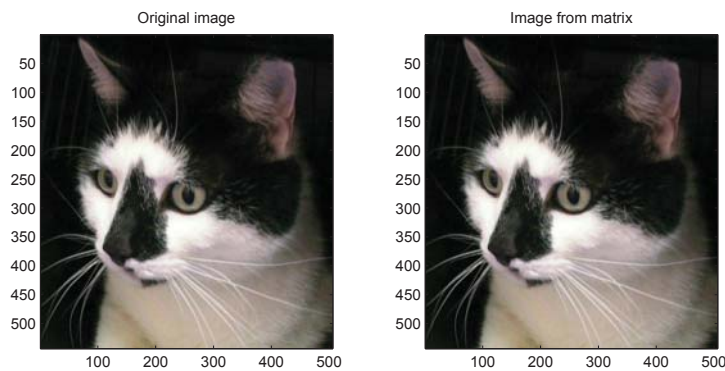
```
clear % clears workspace
recObj = audiorecorder(8000, 8, 1); % define ADC specs
disp('Start speaking now') % prompt speaker
recordblocking(recObj, 2); % record for 2 sec
disp('End of recording'); % indicate end
play(recObj); % playback recording.
myRecording = getaudiodata(recObj); % store data in array
sound(myRecording,8000) % play the speech on the speaker
revRecording = myRecording;
for i=1:length(myRecording)
    revRecording(length(myRecording)+1-i) = myRecording(i);
end
sound(revRecording,8000) % play the speech on the speaker

)
```

Project 2.3 (Transform a jpeg image file into 3D matrix) Using the Matlab script in Example 16.12 as a guide, acquire a jpeg image file on your laptop, transform it into 3D matrix and display in image format.

(ans: The Matlab function `image()` takes either a double-precision variable in range $[0,1]$ or a `uint8` (unsigned 8-bit) variable in the range $[0,255]$. The `image()` function figures out the variable type.

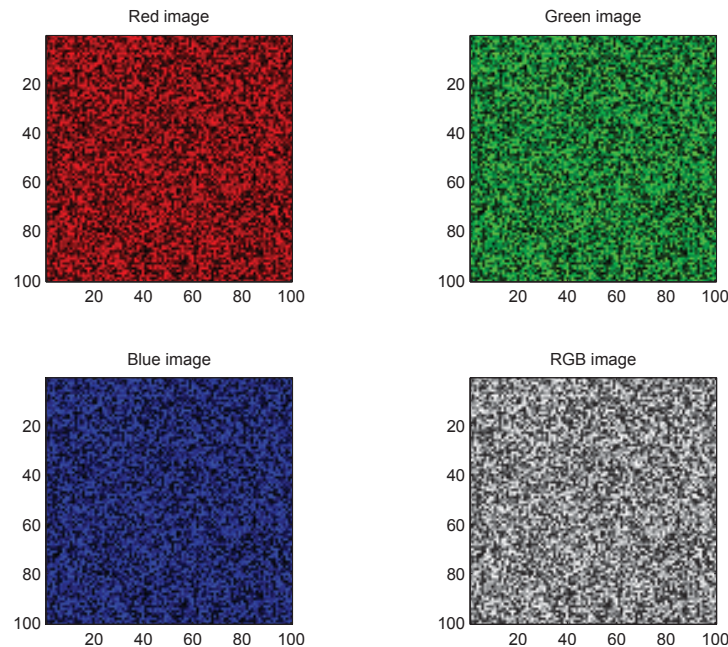
```
clear
filename = input('enter filename ', 's');
filename = [filename '.jpg'];
Im = imread(filename); %Im is uint8 [0,255]
subplot(1,2,1), image(Im)
title('Original image')
axis image
R = zeros(size(Im)); % R is double
R = double(Im)/255; % convert R to [0,1]
subplot(1,2,2), image(R) % R image
title('Image from matrix')
axis image
```



)

Interesting addition: Matlab script that forms a random image. The Matlab function `image()` takes either a double-precision variable in range `[0,1]` or a `uint8` (unsigned 8-bit) variable in the range `[0,255]`. The `image()` function figures out the variable type.

```
clear
matrix= rand(100,100); % color intensities [0,1]
[m n]=size(matrix);
my_imageR = zeros(m,n,3); %initialize the R image
my_imageG = zeros(m,n,3); %initialize the G image
my_imageB = zeros(m,n,3); %initialize the B image
my_imageRGB = zeros(m,n,3); %initialize the RGB image
my_imageR(:,:,1) = matrix; % R image
subplot(2,2,1),image(my_imageR) % plot R image
axis image % plots square pixels
title('Red image')
my_imageG(:,:,2) = matrix; % G image
subplot(2,2,2),image(my_imageG) % plot G image
axis image % plots square pixels
title('Green image')
my_imageB(:,:,3) = matrix; % B image
subplot(2,2,3),image(my_imageB) % plot B image
axis image % plots square pixels
title('Blue image')
my_imageRGB(:,:,1) = matrix; % R component
my_imageRGB(:,:,2) = matrix; % G component
my_imageRGB(:,:,3) = matrix; % B component
subplot(2,2,4),image(my_imageRGB) % plot RGB image
axis image % plots square pixels
title('RGB image')
imwrite(my_imageRGB,'rand_image.jpg') % save 10,000-pixel jpg
```



Project 2.4 (Transform Matlab color image) *The 3D matrix produced by a jpeg displays the x, y spatial location in the first two dimensions and the third dimension defining the red, blue, and green (RGB) values at each spatial location. Modify an acquired jpeg image to display its R, G, and B components as separate images.*

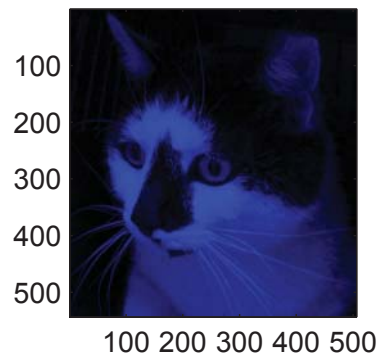
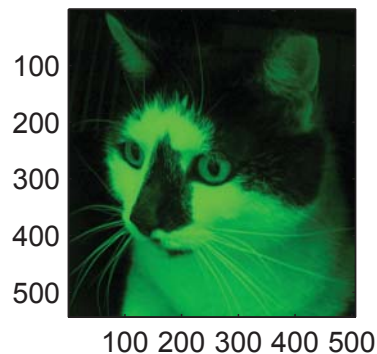
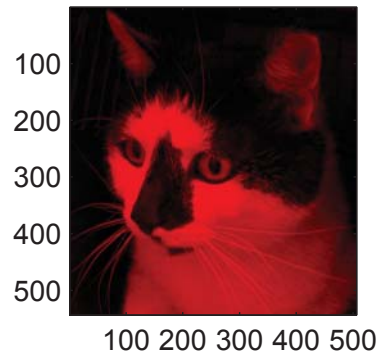
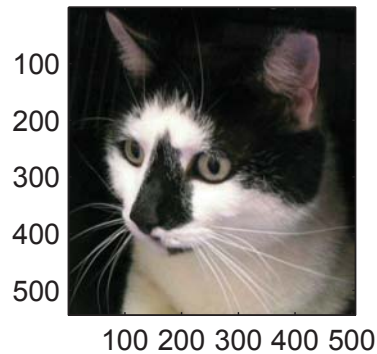
(ans:

```
clear
filename = input('enter filename ', 's');
filename = [filename '.jpg']
Im = imread(filename);           %Im is uint8 [0,255]
subplot(2,2,1), image(Im)
axis image

R = zeros(size(Im));             % R is double
R(:, :, 1) = double(Im(:, :, 1))/255; % convert R to [0,1]
subplot(2,2,2), image(R)        % R image
axis image

G = zeros(size(Im));
G(:, :, 2) = double(Im(:, :, 2))/255;
subplot(2,2,3), image(G)       % G image
axis image

B = zeros(size(Im));
B(:, :, 3) = double(Im(:, :, 3))/255;
subplot(2,2,4), image(B)       % B image
axis image
```

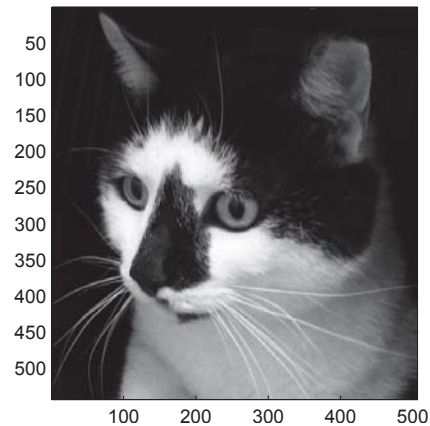
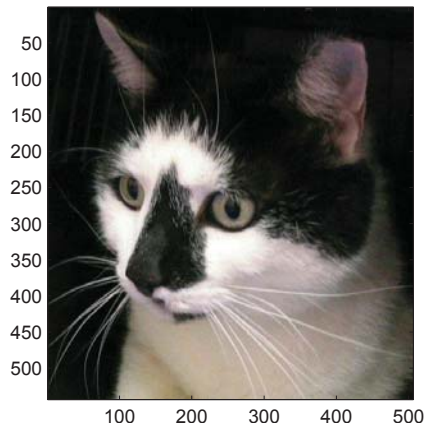


)

Project 2.5 (Convert a color jpeg image into a gray-scale image) Using the Matlab script in Example 16.12, generate a gray-scale image that is a 2D matrix of numbers that vary from 0 to 255.

(ans:

```
clear
filename = input('enter filename ', 's');
filename = [filename '.jpg']
A8 = imread(filename); % uint8 values [0,255]
subplot(1,2,1),image(A8); axis image % produces square pixels
[R C D] = size(A8); % row, column and depth
Gray = zeros(R,C); % form gray-scale image matrix
for i=1:R
    for j = 1:C
        D = cast(A8(i,j,:), 'double'); % convert from uint8 to double for calcs
        Gray(i,j) = sqrt(sum(D.^2)/3); % sqrt (sum of squares/3) = gray-level
    end
end
end
Gray = cast(floor(Gray),'uint8'); % convert to 8-bit integer
A(:, :,1) = Gray; % gray-scale image has equal RGB values
A(:, :,2) = Gray;
A(:, :,3) = Gray;
subplot(1,2,2),image(A); axis image % plot gray-scale image
imwrite(A,'gray_image.jpg') % save gray-scale jpg
```



)