ENGR 160 Algorithms and Applications Sec 02 Spring 2018

Homework Assignment #8 Due 9:00am Mar 02 (F) 2018

Read Chapter 6 of the MATLAB book.

Problem 8.1

The resistance R in ohms of a conductor is given by R = V/I, where V is the voltage potential in volts and I is the current in amperes. The power consumed by the resistor in watts is given by P = VI. Write a MATLAB program with the following files,

1. A function named **readVI** that prompts the user to input the potential and the current
2. A function named **calcRP** that takes in two input arguments as potential and current, and returns two output arguments as resistance and power.
3. A function named **printRP** that takes in two input arguments as resistance and power, and prints out the values of the resistance and power to the screen.
4. A script file that will call the above three functions.

Run a test of your program using input values of V=5volts and I=0.5ampere. Attach the results.

Problem 8.2

Write a program to calculate the position of a projectile at a given time t. For an initial velocity v_0 and angle of departure θ_0, the position is given by x and y coordinates as follows (note: the gravity constant g is 9.81m/s^2):

$x = v_0 \cos(\theta_0) t$;

$$y = v_0 \sin(\theta_0) t - \frac{1}{2} g t^2$$

Your program should consist of the following files:

1. A function to get the values for the initial velocity and angle of departure from user's input.
2. A function that takes the above information as inputs, calculates and returns how long it takes for the projectile to fall on ground. The result of time (in seconds) should be accurate to the second decimal place (i.e., 0.01second).
3. A function that takes the outputs from the above functions as input, and plots the trajectory of the projectile when it is above ground.
4. A script file that will call the above three functions.

Use initial velocity of 60mph and angle of departure of 45^o to test your program. Attach your results.

Problem 8.3

A M-th order median filter runs through the signal x element by element, replacing each element x(n) (n=1:length(x) with the median value of its M neighboring elements from x(n-$\frac{M-1}{2}$) to x(n+$\frac{M-1}{2}$). For simplicity, we only use odd numbers for M. The boundary pixels are going to be repeated for $\frac{M-1}{2}$ times to obtain enough elements. For example, a median filter with M=3 will be applied to the following simple signal stored in vector x:

x = [2 80 6 3]

As the results, the median filtered output signal y will be:
y[1] = Median[2 2 80] = 2
y[2] = Median[2 80 6] = Median[2 6 80] = 6
y[3] = Median[80 6 3] = Median[3 6 80] = 6
y[4] = Median[6 3 3] = Median[3 3 6] = 3

Write a MATLAB function to implement the median filter. This function should take two input arguments, one is the vector to be filtered, and the other one is an odd integer M as the order of the filter. It should output the resulting median-filtered vector. Run and test your function with the above x and M=3, attach your results.

Problem 8.4
Other than random noises, there is another type of noises that is common in real-world digital signals, called spike noises, i.e., isolated data samples whose values are very different from the normal data samples around. For example, dark pixels in bright area or bright pixels in dark area in a digital image, which could be a result of dead pixel of imaging device. In this problem, we will write a program to simulate the spike noise reduction by median filtering.

1. Generate a one-period sinusoidal signal s(n) with 100 linearly spaced samples from 0 to 2π.
2. Add a spike noise with value 10 at a random position in s(t) to generate a noisy signal x(t)
3. Call the median filtering function you wrote in Problem 8.3 to perform median filtering to x(t) with M=3, and M=5, respectively.
4. Plot the clean signal s(t), the noisy signal x(t), and the two filtered signals in the same plot.

Run your program and attach your results.