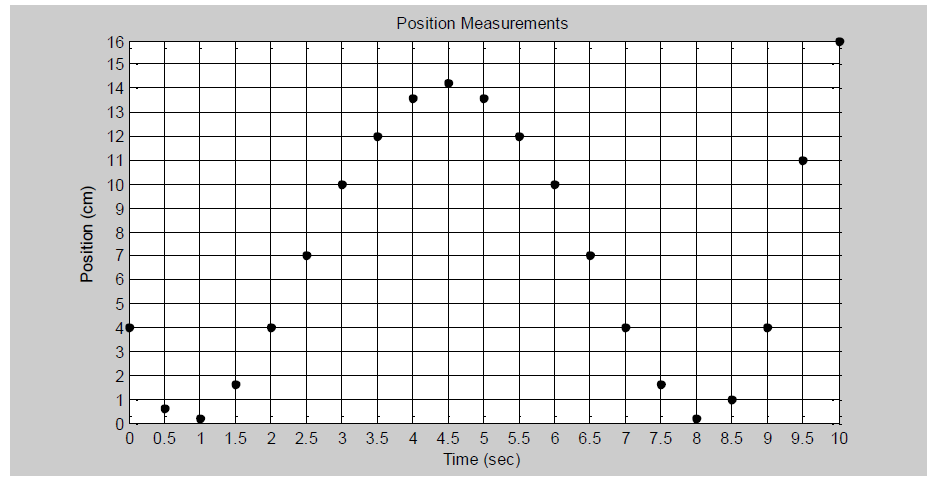
**ENED 1091 HW#3**

**Due Thursday, February 25, 2016 at noon**

**Problem 1:**  The graph below shows position measurements (in cm) collected every 0.5 seconds over a 10 second interval of time.



1. Using ***linear interpolation***, estimate the position of the object at 6.15 seconds. Do this by hand (no ***interp1***) and show your calculations.

**Calculation and Result (include units):**

**Formula for Linear Interpolation = d = d1 +((g-g1)/(g2-g1))(d2-d1)**

**Where g2 and g1 are the x values of the knowns that the desired value is between**

**D2 and d1 are the y values and coincide with the value that we are trying to find**

**G is the x value we want to find the y value for and d is the desired value**

**X = 10 - ((6.15 – 6)/(6.5 – 6))(10 – 7)**

**X = 9.1cm**

1. Now use ***interp1*** and linear interpolation to estimate the position of the object at 6.1, 6.2, 6.3, and 6.4 seconds.

**MATLAB Command and Results (include units):**

%Kyle O'Connor

%

%HW 3

%

home;

clear all;

clc;

x = [5.5 6 6.5 7];

y = [12 10 7 4];

data1 = interp1(x,y,6.1,'linear');

fprintf('6.1seconds: %0.2f cm\n',data1)

data2 = interp1(x,y,6.2,'linear');

fprintf('6.2seconds: %0.2f cm\n',data2)

data3 = interp1(x,y,6.3,'linear');

fprintf('6.3seconds: %0.2f cm\n',data3)

data4 = interp1(x,y,6.4,'linear');

fprintf('6.4seconds: %0.2f cm\n',data4)

**6.1seconds: 9.40 cm**

**6.2seconds: 8.80 cm**

**6.3seconds: 8.20 cm**

**6.4seconds: 7.60 cm**

**Problem 2:** For this problem, you need the data file, HW3P2.mat, posted on the Blackboard site with Homework #3. The file has a vector of times, t, which starts at 0 increments by 0.04 and ends at 0.8 seconds. It also has a vector of voltage measurements, V (volts), corresponding to the given times. Download the data file and save it in your current MATLAB folder. The command: load HW3P2 will load the two vectors into your workspace.

1. Plot the original data points (don’t connect the points with lines). Add title and axis labels (with units).

**Plot:**

****

1. Use ***interp1*** with a method of ***nearest*** to estimate the voltage every 0.005 seconds between 0 and 0.8 seconds. On a single plot, plot the original data points as red stars and the interpolated data points as black circles. Add title, axis labels (with units), and a legend.

**MATLAB Commands and Plot**

****

%Kyle O'Connor

%

%HW 3

%

home;

clear all;

clc;

load HW3P2

g = 0:0.005:.8;

nearest = interp1(t,V,g,'nearest');

linear = interp1(t,V,g,'linear');

spline = interp1(t,V,g,'spline');

plot(t,V,'r\*')

hold on;

plot(g,nearest,'bo')

title('Voltage Over Time')

xlabel('Time (sec)')

ylabel('Voltage (volts)')

legend('Actual Data','Nearest Estimate')

1. Repeat part (b) using ***linear*** interpolation.

**MATLAB Commands and Plot**

****

%Kyle O'Connor

%

%HW 3

%

home;

clear all;

clc;

load HW3P2

g = 0:0.005:.8;

nearest = interp1(t,V,g,'nearest');

linear = interp1(t,V,g,'linear');

spline = interp1(t,V,g,'spline');

plot(t,V,'r\*')

hold on;

plot(g,linear,'bo')

title('Voltage Over Time')

xlabel('Time (sec)')

ylabel('Voltage (volts)')

legend('Actual Data','Linear Estimate')

1. Repeat part (b) using ***spline*** interpolation.

**MATLAB Commands and Plot**

****

%Kyle O'Connor

%

%HW 3

%

home;

clear all;

clc;

load HW3P2

g = 0:0.005:.8;

nearest = interp1(t,V,g,'nearest');

linear = interp1(t,V,g,'linear');

spline = interp1(t,V,g,'spline');

plot(t,V,'r\*')

hold on;

plot(g,spline,'bo')

title('Voltage Over Time')

xlabel('Time (sec)')

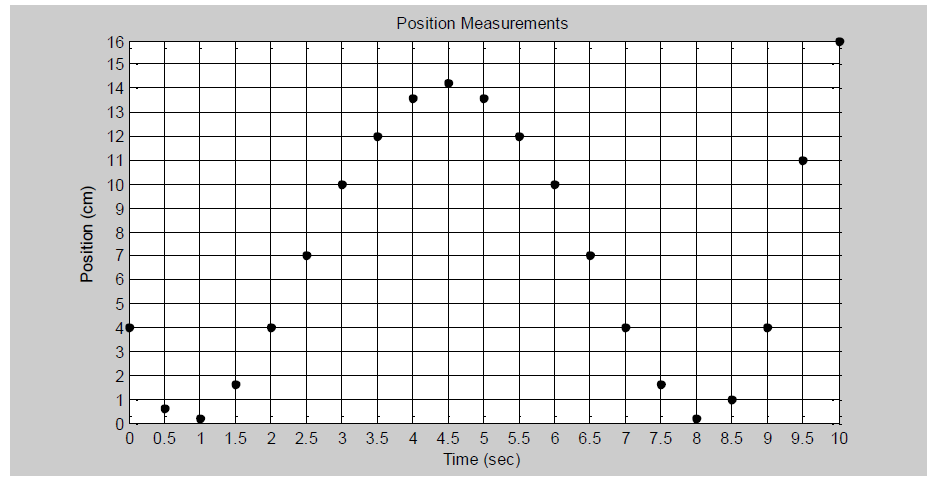
ylabel('Voltage (volts)')

legend('Actual Data','Spline Estimate')

1. What kind of waveform does your plot in part (d) look like? Could you possibly have picked this up from looking at the original data points?

The wave form in part (d) looks very similar to a cosine wave form, even though it is not exact. There was definitely a pattern that could be seen from the original data but I don’t think I could’ve spotted the cosine wave form that was made from the spline interpolation.

**Problem 3:** The graph below shows position measurements (in cm) collected every 0.5 seconds over a 10 second interval of time.

****

1. Using a Δt = 0.5 sec, estimate the velocity at t =  sec using the 2-point estimate and the 3-point estimate for the derivative. **Be sure to show your work and include units!**

f `(6) = (f `(6 + 0.5) – f(6))/( Δt)

f `(6) = (7 – 10)/0.5

f `(6) = (-3) / 0.5 = -6 cm/sec

2-point Estimate of Velocity at t = 6 sec: -6 cm/second

f `(6) = (f(6 + 0.5) – f(6 – 0.5))/ (2Δt)

f `(6) = (7 – 12) / (2Δt)

f `(6) = -5 / 1 = -5 cm /second

3-point Estimate of Velocity at t = 6 sec: -5 cm/second

1. Using the estimate for 2nd derivative and a Δt = 0.5 sec, estimate the acceleration at t = 6 sec. **Again, show work and include units!**

**f ``(6) = (f(6 + 0.5) – 2\*f(6) + f(6 – 0.5)) / (**0.5**^2)**

**f ``(6) = (7 – 20 + 12) / 0.25**

**f ``(6) = (-1) / 0.25 = -4cm / second^2**

Estimate of Acceleration at t = 6 sec: -4 cm/second^2

1. What could be changed to improve the accuracy of the derivative estimates?

**Problem 4:** For this problem, you need the data file, HW3P4.mat, posted on the Blackboard site with Homework #3. The MATLAB command: load HW3P4 will load the data into the MATLAB workspace. The data file has 4 vectors:

* t is a vector of times (seconds) starting at 0, incrementing by 0.1, and ending at 4
* pos is a vector of position measurements (mm) taken at the times specified in vector t
* t\_act is a vector of times (seconds) starting at 0, incrementing by 0.005, and ending at 4
* v\_act is a vector of velocity measurements (mm/s) corresponding to the times in t\_act

1. Plot the position measurements over time (don’t connect the individual data points with lines). Add a title and axis labels (with units) to the plot.

**Plot:**

****

1. Use the position measurements and the 2-pt reverse estimate for derivative to estimate the velocity at each time in vector t. Assume the initial velocity is 0.

**MATLAB Code for 2-PT Estimate of Velocity:**

%Kyle O'Connor

%

%HW 3

%

home;

clear all;

clc;

load HW3P4

plot(t,pos,'blo')

title('Position over Time')

xlabel('Time (sec)')

ylabel('Position (mm)')

delt = 0.1;

for i = 2:length(t)

f(i) = (pos(i) - pos(i-1))/(delt)

end

1. On a single graph, plot the velocity estimates over time, t, as individual data points and the actual velocity (v\_act) over t\_act. Add a title, axis labels (with units), and a legend.

**Plot and MATLAB Code for Plotting:**

****

1. How could the accuracy of the velocity estimates be improved?

The accuracy of the velocity estimate could be improved by instead of taking the derivative and actually finding the slope equation and estimating, using a spline estimate that might ‘hug’ the estimated data better. Using a higher order polynomial fit to the data will ‘hug’ the data better.