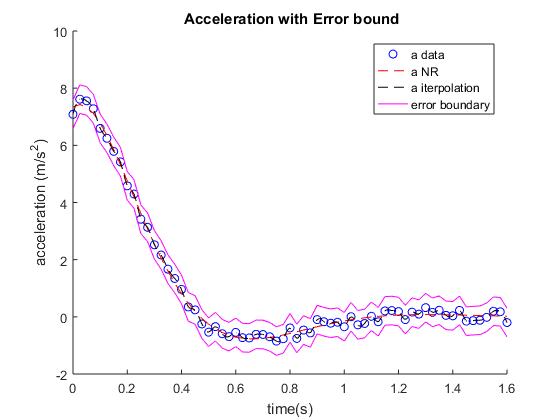
Hung Le (John) & Anh Nguyen (Ryan)

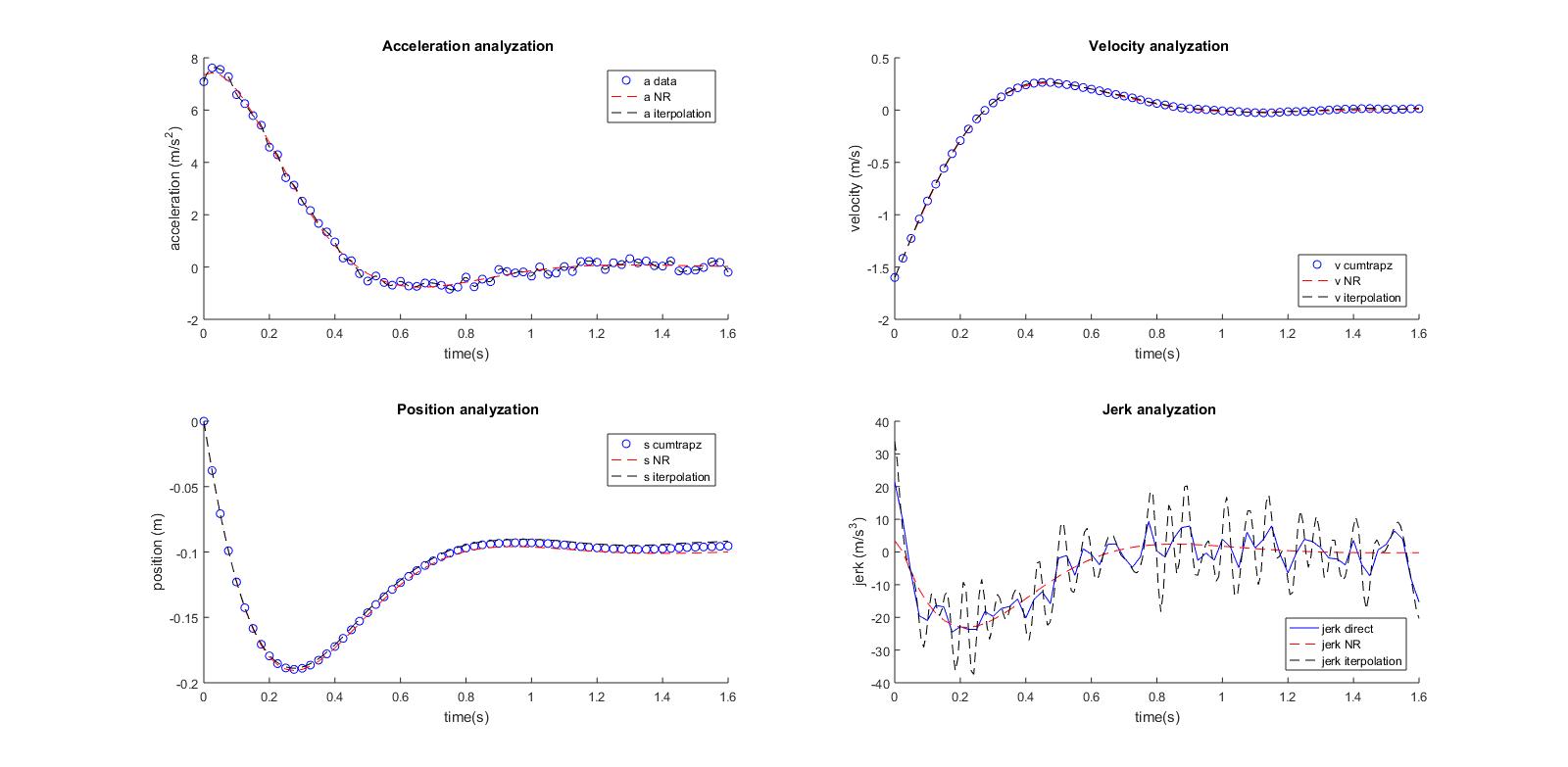
Professor Stephanie Bostwick

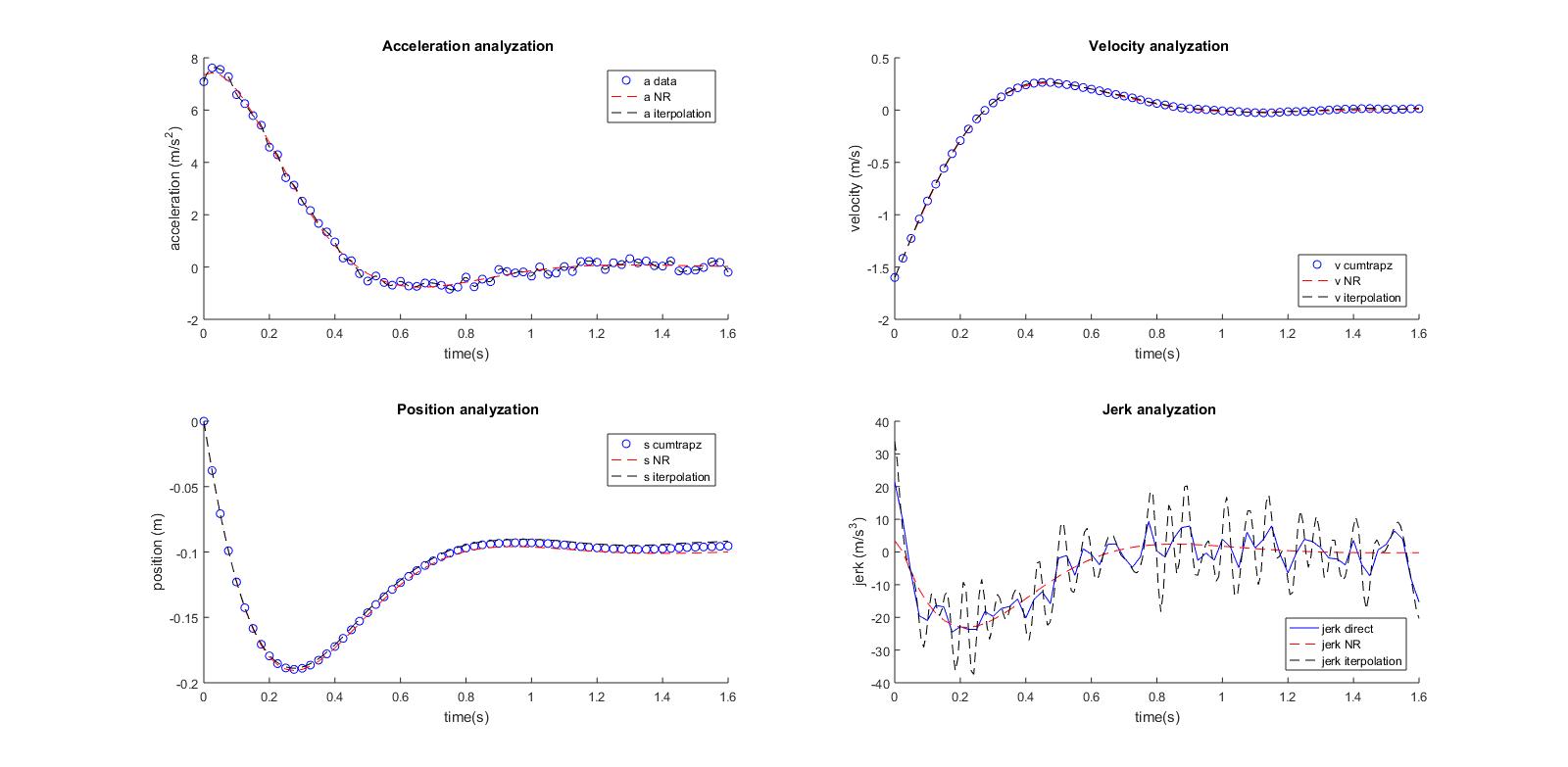
26 Feb, 2018

**Project 3**

**Resulting graphs**







**Discussion**

We were asked to integrate an acceleration data set in order to find velocity and position. We first followed the instruction given and used composite trapezoid rule to integrate the data directly. Finding that direct trapezoid might not be the most efficient way, we used cubic spline to interpolate the data onto a better interval and then integrate afterwards. Then, we also tried to use nonlinear curve fit and integrate the results. We did the similar differentiations approaches with the jerk.

After finishing the code, we decided to plot the graphs to all methods and observe the results. We found out that when we directly integrated and used cubic spline, acceleration data and its differentiation (jerk) really jiggles due to error. The usage of non-linear curve fit however gave a better graphical result. However, when it comes to velocity and position, the result of the two methods does not seem to be different from each other. Because the error in acceleration does not affect the quality of integration too much, so by interpolating (connecting data points), they also show reasonable results.

We decided to settle on the regression curve fit method. By doing that , we can neglect the errors in the given data while also collect reasonable results. Interpolating only works when the data is correct or close enough but in this case the data really varies (+-0.5m/s^2). It works for every cases and we can check our fit quality using r^2. As we have r^2=0.9962 from our fit, we are really confident about our number. Also, I sketch the graph of acceleration with its error bound. I notice that the extra area adding up or taking away from the error would be neglectable. However, the slope of acceleration graph (jerk) will be really off due to error . Therefore, we choose linear fit which is a solid and general model for our graph and have a better prediction for the jerk. Also, I learned from physics that we often do experiment to collect data and try different fit to determine which is the best fit so that we can general our problem and apply our equation to similar model. In addition, the jerk value from model equation have nice behavior and close to the one generate from data. Finally, I believe that the oscillation from our data is the result of sound wave from the noise from the vehicle. I believe that if we want to look in to the mechanic behavior like speed and have fast it can speed up, we can ignore those noise, and linear fit graph does that.

Also, observing the graphs, we can see the familiar trend of velocity, position and acceleration. The jerk is also reasonably appeared to be correct. While in the other methods, the resulting data points oscillates vastly. That is why we are confident that our method and results are correct.

The challenging aspect of this project is when we had to apply the mathematical methods into MATLAB and we had to go through all the constraints that each method has. By doing that, we has the ability to provide each method everything it needs to perform well. The reasoning to choose the most efficient and accurate method is also needed to be well-considered.

The interesting aspect of this project is that we can see the pros and cons of each integrating and differentiating methods. We also tried different approaches which give different answers to each cases. The problem seems to be just an acceleration physics problem but more complicated when it comes to data analyzation.

**Code**

%Project 3 - main m-file

clc

clear

close

%% Getting data and given constants

load data

time = Dat(:,1);

accel = Dat(:,2);

v0 = -1.6; %m/s

s0 = 0; %m

h = (time(end)-time(1))/length(time);

%% Direct\_data using cumtrapz and gradient

v\_cumtrapz = v0+cumtrapz(time,accel);

s\_cumtrapz = s0+cumtrapz(time,v\_cumtrapz);

jerk\_direct= gradient(accel,h);

%% Interpolation using spline

tt = linspace(min(time),max(time),200);

a\_spline = spline(time,accel,tt);

v\_spline = v0+cumtrapz(tt,a\_spline);

s\_spline = s0+cumtrapz(tt,v\_spline);

jerk\_iterpolation=gradient(a\_spline,(max(tt)-min(tt))/200);

%% Nonlinear Regression curve fit using fminsearch

a\_model = @(t,A,B,C,D) A.\*exp(B.\*t).\*cos(C.\*t+D);

Sr\_fun = @(b,xi,yi) sum((yi-a\_model(xi,b(1),b(2),b(3),b(4))).^2);

b = fminsearch(Sr\_fun,[1,1,1,1],[],time,accel);

A = b(1);

B = b(2);

C = b(3);

D = b(4);

a\_NR = a\_model(time,A,B,C,D);

v\_NR = v0+cumtrapz(time,a\_NR);

s\_NR = s0+cumtrapz(time,v\_NR);

jerk\_NR = gradient(a\_NR,h);

% fit quality

Sr = sum((accel-a\_model(time,A,B,C,D)).^2);

St = sum((accel-mean(accel)).^2);

r\_squared = 1-Sr/St;

%% Plot data

% graph acceleration with error

figure(1)

hold on

plot(time,accel,'ob');

plot(time,a\_NR,'r--');

plot(tt,a\_spline,'k--');

plot(time,accel+0.5,'m');

plot(time,accel-0.5,'m');

ylabel('acceleration (m/s^2)');

xlabel('time(s)');

title(' Acceleration with Error bound');

hold off

legend('a data','a NR','a iterpolation','error boundary');

% acceleration vs time

figure(2)

subplot(221)

hold on

plot(time,accel,'ob');

plot(time,a\_NR,'r--');

plot(tt,a\_spline,'k--');

ylabel('acceleration (m/s^2)');

xlabel('time(s)');

title(' Acceleration analyzation');

legend('a data','a NR','a iterpolation');

hold off

% velocity vs time

subplot(222)

hold on

plot(time,v\_cumtrapz,'ob');

plot(time,v\_NR,'r--');

plot(tt,v\_spline,'k--');

ylabel('velocity (m/s)');

xlabel('time(s)');

title(' Velocity analyzation');

legend('v cumtrapz','v NR','v iterpolation','location','southeast');

hold off

% position vs time

subplot(223)

hold on

plot(time,s\_cumtrapz,'ob');

plot(time,s\_NR,'r--');

plot(tt,s\_spline,'k--');

ylabel('position (m)');

xlabel('time(s)');

title(' Position analyzation');

legend('s cumtrapz','s NR','s iterpolation');

hold off

% jerk vs time

subplot(224)

hold on

plot(time,jerk\_direct,'b-');

plot(time,jerk\_NR,'r--');

plot(tt,jerk\_iterpolation,'k--');

ylabel('jerk (m/s^3)');

xlabel('time(s)');

title(' Jerk analyzation');

legend('jerk direct','jerk NR','jerk iterpolation','location','southeast');

hold off