# 5.3 - Appendix C – Matlab Code

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Lindsay Burton, Orion Miller, Ben Tory-Pratt, Austin Van Rossum

% OHLINS TTX25 DAMPER MODELER

% March 2019

% References:

%https://deepblue.lib.umich.edu/bitstream/handle/2027.42/49574/proj25\_report.pdf?sequence=2

%Known Issues:

%Legend labels on the graphs are shown duplicated. I don't know of a way these can

%be removed without making major changes to the way the data is

%plotted/organized.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clearvars

clc

tic

cd(uigetdir); % Select directory of data

OhlinsData0\_0\_0\_0 = load('0-0 0-0.csv'); %Import in individual data sets

OhlinsData0\_1\_0\_1 = load('0-1 0-1.csv');

OhlinsData0\_2\_0\_2 = load('0-2 0-2.csv');

OhlinsData0\_3\_0\_3 = load('0-3 0-3.csv');

OhlinsData0\_43\_0\_43 = load('0-4.3 0-4.3.csv'); %Used for both low and hi speed

OhlinsData2\_43\_2\_43 = load('2-4.3 2-4.3.csv');

OhlinsData4\_43\_4\_43 = load('4-4.3 4-4.3.csv');

OhlinsData6\_43\_6\_43 = load('6-4.3 6-4.3.csv');

OhlinsData10\_43\_10\_43 = load('10-4.3 10-4.3.csv');

OhlinsData15\_43\_15\_43 = load('15-4.3 15-4.3.csv');

OhlinsData25\_43\_25\_43 = load('25-4.3 25-4.3.csv');

index0\_0\_0\_0 = OhlinsData0\_0\_0\_0(:,2)>0; %Use logical indexing to split each into two files

%separate compression and rebound data

OhlinsData0\_0\_0\_0Comp = OhlinsData0\_0\_0\_0(index0\_0\_0\_0,1:2);

OhlinsData0\_0\_0\_0Reb = OhlinsData0\_0\_0\_0(~index0\_0\_0\_0,1:2);

%Order force vals from lowest to highest (fixes a quirk in the data)

OhlinsData0\_0\_0\_0Comp = sort(OhlinsData0\_0\_0\_0Comp);

OhlinsData0\_0\_0\_0Reb = sort(abs(OhlinsData0\_0\_0\_0Reb));

%Make the comp and rebound sets have equal lengths (some have a slight

%difference)

OhlinsData0\_0\_0\_0Reb(:,2) = -OhlinsData0\_0\_0\_0Reb(:,2);

%Repeat this procedure for all other data sets

index0\_1\_0\_1 = OhlinsData0\_1\_0\_1(:,2)>0;

OhlinsData0\_1\_0\_1Comp = OhlinsData0\_1\_0\_1(index0\_1\_0\_1,1:2);

OhlinsData0\_1\_0\_1Reb = OhlinsData0\_1\_0\_1(~index0\_1\_0\_1,1:2);

OhlinsData0\_1\_0\_1Comp = sort(OhlinsData0\_1\_0\_1Comp);

OhlinsData0\_1\_0\_1Reb = sort(abs(OhlinsData0\_1\_0\_1Reb));

OhlinsData0\_1\_0\_1Reb(:,2) = -OhlinsData0\_1\_0\_1Reb(:,2);

OhlinsData0\_1\_0\_1Comp = OhlinsData0\_1\_0\_1Comp(1:286,:);

index0\_2\_0\_2 = OhlinsData0\_2\_0\_2(:,2)>0;

OhlinsData0\_2\_0\_2Comp = OhlinsData0\_2\_0\_2(index0\_2\_0\_2,1:2);

OhlinsData0\_2\_0\_2Reb = OhlinsData0\_2\_0\_2(~index0\_2\_0\_2,1:2);

OhlinsData0\_2\_0\_2Comp = sort(OhlinsData0\_2\_0\_2Comp);

OhlinsData0\_2\_0\_2Reb = sort(abs(OhlinsData0\_2\_0\_2Reb));

OhlinsData0\_2\_0\_2Reb(:,2) = -OhlinsData0\_2\_0\_2Reb(:,2);

OhlinsData0\_2\_0\_2Reb = OhlinsData0\_2\_0\_2Reb(1:290,:);

index0\_3\_0\_3 = OhlinsData0\_3\_0\_3(:,2)>0;

OhlinsData0\_3\_0\_3Comp = OhlinsData0\_3\_0\_3(index0\_3\_0\_3,1:2);

OhlinsData0\_3\_0\_3Reb = OhlinsData0\_3\_0\_3(~index0\_3\_0\_3,1:2);

OhlinsData0\_3\_0\_3Comp = sort(OhlinsData0\_3\_0\_3Comp);

OhlinsData0\_3\_0\_3Reb = sort(abs(OhlinsData0\_3\_0\_3Reb));

OhlinsData0\_3\_0\_3Reb(:,2) = -OhlinsData0\_3\_0\_3Reb(:,2);

OhlinsData0\_3\_0\_3Comp = OhlinsData0\_3\_0\_3Comp(1:277,:);

index0\_43\_0\_43 = OhlinsData0\_43\_0\_43(:,2)>0;

OhlinsData0\_43\_0\_43Comp = OhlinsData0\_43\_0\_43(index0\_43\_0\_43,1:2);

OhlinsData0\_43\_0\_43Reb = OhlinsData0\_43\_0\_43(~index0\_43\_0\_43,1:2);

OhlinsData0\_43\_0\_43Comp = sort(OhlinsData0\_43\_0\_43Comp);

OhlinsData0\_43\_0\_43Reb = sort(abs(OhlinsData0\_43\_0\_43Reb));

OhlinsData0\_43\_0\_43Reb(:,2) = -OhlinsData0\_43\_0\_43Reb(:,2);

OhlinsData0\_43\_0\_43Comp = OhlinsData0\_43\_0\_43Comp(1:280,:);

index2\_43\_2\_43 = OhlinsData2\_43\_2\_43(:,2)>0;

OhlinsData2\_43\_2\_43Comp = OhlinsData2\_43\_2\_43(index2\_43\_2\_43,1:2);

OhlinsData2\_43\_2\_43Reb = OhlinsData2\_43\_2\_43(~index2\_43\_2\_43,1:2);

OhlinsData2\_43\_2\_43Comp = sort(OhlinsData2\_43\_2\_43Comp);

OhlinsData2\_43\_2\_43Reb = sort(abs(OhlinsData2\_43\_2\_43Reb));

OhlinsData2\_43\_2\_43Reb(:,2) = -OhlinsData2\_43\_2\_43Reb(:,2);

index4\_43\_4\_43 = OhlinsData4\_43\_4\_43(:,2)>0;

OhlinsData4\_43\_4\_43Comp = OhlinsData4\_43\_4\_43(index4\_43\_4\_43,1:2);

OhlinsData4\_43\_4\_43Reb = OhlinsData4\_43\_4\_43(~index4\_43\_4\_43,1:2);

OhlinsData4\_43\_4\_43Comp = sort(OhlinsData4\_43\_4\_43Comp);

OhlinsData4\_43\_4\_43Reb = sort(abs(OhlinsData4\_43\_4\_43Reb));

OhlinsData4\_43\_4\_43Reb(:,2) = -OhlinsData4\_43\_4\_43Reb(:,2);

OhlinsData4\_43\_4\_43Reb = OhlinsData4\_43\_4\_43Reb(1:266,:);

index6\_43\_6\_43 = OhlinsData6\_43\_6\_43(:,2)>0;

OhlinsData6\_43\_6\_43Comp = OhlinsData6\_43\_6\_43(index6\_43\_6\_43,1:2);

OhlinsData6\_43\_6\_43Reb = OhlinsData6\_43\_6\_43(~index6\_43\_6\_43,1:2);

OhlinsData6\_43\_6\_43Comp = sort(OhlinsData6\_43\_6\_43Comp);

OhlinsData6\_43\_6\_43Reb = sort(abs(OhlinsData6\_43\_6\_43Reb));

OhlinsData6\_43\_6\_43Reb(:,2) = -OhlinsData6\_43\_6\_43Reb(:,2);

index10\_43\_10\_43 = OhlinsData10\_43\_10\_43(:,2)>0;

OhlinsData10\_43\_10\_43Comp = OhlinsData10\_43\_10\_43(index10\_43\_10\_43,1:2);

OhlinsData10\_43\_10\_43Reb = OhlinsData10\_43\_10\_43(~index10\_43\_10\_43,1:2);

OhlinsData10\_43\_10\_43Comp = sort(OhlinsData10\_43\_10\_43Comp);

OhlinsData10\_43\_10\_43Reb = sort(abs(OhlinsData10\_43\_10\_43Reb));

OhlinsData10\_43\_10\_43Reb(:,2) = -OhlinsData10\_43\_10\_43Reb(:,2);

index15\_43\_15\_43 = OhlinsData15\_43\_15\_43(:,2)>0;

OhlinsData15\_43\_15\_43Comp = OhlinsData15\_43\_15\_43(index15\_43\_15\_43,1:2);

OhlinsData15\_43\_15\_43Reb = OhlinsData15\_43\_15\_43(~index15\_43\_15\_43,1:2);

OhlinsData15\_43\_15\_43Comp = sort(OhlinsData15\_43\_15\_43Comp);

OhlinsData15\_43\_15\_43Reb = sort(abs(OhlinsData15\_43\_15\_43Reb));

OhlinsData15\_43\_15\_43Reb(:,2) = -OhlinsData15\_43\_15\_43Reb(:,2);

OhlinsData15\_43\_15\_43Comp = OhlinsData15\_43\_15\_43Comp(1:277,:);

index25\_43\_25\_43 = OhlinsData25\_43\_25\_43(:,2)>0;

OhlinsData25\_43\_25\_43Comp = OhlinsData25\_43\_25\_43(index25\_43\_25\_43,1:2);

OhlinsData25\_43\_25\_43Reb = OhlinsData25\_43\_25\_43(~index25\_43\_25\_43,1:2);

OhlinsData25\_43\_25\_43Comp = sort(OhlinsData25\_43\_25\_43Comp);

OhlinsData25\_43\_25\_43Reb = sort(abs(OhlinsData25\_43\_25\_43Reb));

OhlinsData25\_43\_25\_43Reb(:,2) = -OhlinsData25\_43\_25\_43Reb(:,2);

OhlinsData25\_43\_25\_43Reb = OhlinsData25\_43\_25\_43Reb(1:271,:);

%Concatenate all these files into a single master data set so that analysis

%of the different conditions can be done with a loop

OhlinsDataMaster = vertcat(OhlinsData0\_0\_0\_0Comp, OhlinsData0\_0\_0\_0Reb, OhlinsData0\_1\_0\_1Comp,...

OhlinsData0\_1\_0\_1Reb, OhlinsData0\_2\_0\_2Comp,OhlinsData0\_2\_0\_2Reb, OhlinsData0\_3\_0\_3Comp,...

OhlinsData0\_3\_0\_3Reb, OhlinsData0\_43\_0\_43Comp, OhlinsData0\_43\_0\_43Reb, OhlinsData2\_43\_2\_43Comp,...

OhlinsData2\_43\_2\_43Reb, OhlinsData4\_43\_4\_43Comp, OhlinsData4\_43\_4\_43Reb, OhlinsData6\_43\_6\_43Comp,...

OhlinsData6\_43\_6\_43Reb, OhlinsData10\_43\_10\_43Comp, OhlinsData10\_43\_10\_43Reb, OhlinsData15\_43\_15\_43Comp,...

OhlinsData15\_43\_15\_43Reb, OhlinsData25\_43\_25\_43Comp, OhlinsData25\_43\_25\_43Reb);

%Define the start and end points of each sweep so that these can be indexed

%through

SweepStarts = [1 282 563 849 1135 1425 1715 1992 2269 2549 2829 3102 ...

3375 3641 3907 4176 4445 4713 4981 5258 5535 5806];

SweepEnds = [281 562 848 1134 1424 1714 1991 2268 2548 2828 3101 3374 ...

3640 3906 4175 4444 4712 4980 5257 5534 5805 6076];

%Define number of different data sets within the master data set

SweepsNum = numel(SweepStarts);

%Define the adjustment conditions of the dampers through each of the sweeps

OhlinsDataLSC = [0 0 0 0 0 0 0 0 0 0 2 2 4 4 6 6 10 10 15 15 25 25];

OhlinsDataHSC = [0 0 1 1 2 2 3 3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 4.3 ...

4.3 4.3 4.3 4.3 4.3 4.3];

OhlinsDataLSR = OhlinsDataLSC;

OhlinsDataHSR = OhlinsDataHSC;

disp('Valve adjustment naming convention: LSC-HSC LSR-HSR (L=low, H=high, S=speed, C=compression, R=rebound).');

disp('LS adjustments (3mm hex) are counted in clicks from fully closed (fully clockwise), HS adjustments (12mm hex)');

disp('are counted in turns from fully open (fully counter-clockwise)');

TheoreticalFit = zeros(length(OhlinsDataMaster),1); %Initialize exponential fit

for i=1:SweepsNum

%Create and exponential fit for each sweep:

TheoreticalFit = fit(OhlinsDataMaster(SweepStarts(i):SweepEnds(i),1),OhlinsDataMaster(SweepStarts(i):SweepEnds(i),2),'exp1');

%Store the equation coefficiencts as Cd (Initial Slope) and Lambda

%(Non-linearity coefficient):

Cd(i) = TheoreticalFit.a;

Lambda(i) = TheoreticalFit.b;

end

Rce = zeros(SweepsNum); %Initialize ratio of compression over extension

for i=1:SweepsNum-1

AvgCompForce(i) = mean(OhlinsDataMaster(SweepStarts(i):SweepEnds(i),2));

AvgRebForce(i) = mean(abs(OhlinsDataMaster(SweepStarts(i+1):SweepEnds(i+1),2)));

% Find the average ratio of compression over extension force for each

% sweep:

Rce(i) = AvgCompForce(i)./AvgRebForce(i);

end

Rce = Rce(1:2:SweepsNum); %Eliminates half the entries from the previous form of the matrix, because only every other value is used

% FZ\_CLASS(i) = interp1(FZ\_LIST,FZ\_LIST,FZ\_AVG(i),'nearest');

%Load Test Data

TestData = load(uigetfile);

%Index Test by Variables

TestDataTime = TestData(:,1);

TestDataLinDisp = TestData(:,4);

TestDataLinVel = zeros(numel(TestDataTime),1);

for i=1:numel(TestDataTime)-1

%Find linear velocity by differentiating displacement vs. time:

TestDataLinVel(i) = abs(TestDataLinDisp(i+1)-TestDataLinDisp(i))./(TestDataTime(i+1)-TestDataTime(i));

end

TestDataLinVel(numel(TestDataTime)) = TestDataLinVel(numel(TestDataTime)-1);

TestDataForce = TestData(:,2);

TestDataForce = TestDataForce\*-0.009806; %Unit Conversion: Grams to Newtons

% TestDataTemp = TestData(:,4);

disp('Importing Test Data...')

while(1)

AdjustmentChoice = input('What kind of adjustment is being tested? Enter 1 for Low-Speed, enter 2 for High-Speed: ');

if AdjustmentChoice == 1

TestDataLSC = input('Enter Low Speed Compression: ');

TestDataLSR = input('Enter Low Speed Rebound: ');

TestDataHSC = 4.3;

TestDataHSR = 4.3;

break

else if AdjustmentChoice == 2

TestDataLSC = 0;

TestDataLSR = 0;

TestDataHSC = input('Enter High Speed Compression: ');

TestDataHSR = input('Enter High Speed Rebound: ');

break

end

end

end

%Create a display of all measured variables vs. time

PlotTitle = sprintf('Test Data vs. Time, LSC: %s, LSR: %s, HSC: %s, HSR: %s', ...

num2str(TestDataLSC), num2str(TestDataLSR), ...

num2str(TestDataHSC), num2str(TestDataHSR));

f = figure('Name',PlotTitle,'NumberTitle','off','Position',[0 0 1300 670]);

figure(f)

subplot(3,1,1)

plot(TestDataTime, TestDataLinDisp)

title('Damper Displacement vs . Time');

xlabel('Time (s)');

ylabel('Displacement (mm)');

grid on

% set(gca,'color',[0 0 0])

subplot(3,1,2)

plot(TestDataTime, TestDataLinVel)

title('Damper Velocity vs. Time');

xlabel('Time (s)');

ylabel('Velocity (mm/s)');

grid on

% set(gca,'color',[0 0 0])

subplot(3,1,3)

plot(TestDataTime, TestDataForce)

title('Force vs. Time');

xlabel('Time (s)');

ylabel('Force (N)');

grid on

% set(gca,'color',[0 0 0])

%This code below is for if a temperature sensor were to be added

% to the apparatus in the future. Shows damper temperature during test.

% subplot(3,1,4

% plot(TestDataTime, TestDataTemp)

% title('Damper Temperature vs. Time');

% xlabel('Time (s)');

% ylabel('Temperature (C)');

% grid on

% % set(gca,'color',[0 0 0])

%Define model parameters for the given adjustment vals:

TestRce =(Rce(2)+Rce(3))./2; %Redundant, as the two curves are being defined by their different initial Cd's

TestCdComp = (Cd(3)+Cd(5))./2;

TestCdReb = (Cd(4)+Cd(6))./2;

TestLambdaComp = (Lambda(3)+Lambda(5))./2;

TestLambdaReb = (Lambda(4)+Lambda(6))./2;

%I just defined the model parameters manually by checking what sweeps a high

%speed adjustment of 1.5 would match up with. It would be better to make a

%curve fit of Cd and Lambda (lsqnonlin?) which can be interpolated through for whatever

%adjustment is specified.

TheoreticalCompEqn = TestCdComp\*exp(TestLambdaComp\*TestDataLinVel); %Define modeled curve for compression portion

TheoreticalRebEqn = TestCdReb\*exp(TestLambdaReb\*TestDataLinVel); % Define modeled curve for rebound portion

f = figure('Name',PlotTitle, 'NumberTitle', 'off');

set(f,'units','normalized','outerposition',[0,0,1,1]);

plot(TestDataLinVel, TestDataForce,'.k', 'DisplayName', 'Test Data', 'MarkerSize',20);

hold on

% plot(TestDataTime(SweepStarts(i):SweepEnds(i)),TestDataForce\_Fit(SweepStarts(i):SweepEnds(i)),'k');

% hold on

plot(TestDataLinVel,TheoreticalCompEqn,'.r', 'DisplayName', 'Theoretical Fit', 'MarkerSize',20);

plot(TestDataLinVel,TheoreticalRebEqn,'.r', 'DisplayName', 'Theoretical Fit', 'MarkerSize',20);

grid on

title('Force vs. Velocity')

xlabel('Velocity (mm/s)')

ylabel('Force (N)')

legend

% %This code is for if a temperature sensor were to be added to the apparatus in the future. Produces a heat plot to show how the ...

% %damper temperature changes throughout a test.

%

% f = figure('Name',sprintf('Test Data Heat Map: LSC = %s, LSR = %s, HSC = %s, HSR = %s', TestDataLSC, TestDataLSR, TestDataHSC, TestDataHSR),'NumberTitle','off');

% set(f,'units','normalized','outerposition',[0,0,1,1]);

% dotsz = 25;

% scatter(TestDataTime(SweepStarts(i):SweepEnds(i)),TestDataForce(SweepStarts(i):SweepEnds(i)),dotsz,TestDataTemp(SweepStarts(i):SweepEnds(i)));

% title('Force vs. Velocity, Temperature')

%

%The manufacturer data will now be graphed for each type of adjustment variation. These plots are being added

%with the intention that as more tests are done with the apparatus in the

%future, their data can be plotted against each of these curves to

%see if they can be validated/reproduced.

f = figure('Name',sprintf('Manufacturer vs. Tested Damping Plots, Ohlins TTX25 Low-Speed Adjustment Variation'),'NumberTitle','off');

set(f,'units','normalized','outerposition',[0,0,1,1]);

% plot(,'.g'); %These commented out lines are for adding test data

hold on

plot(OhlinsData0\_43\_0\_43Comp(:,1),OhlinsData0\_43\_0\_43Comp(:,2),'r', 'DisplayName','0-4.3 0-4.3');

plot(OhlinsData0\_43\_0\_43Reb(:,1),OhlinsData0\_43\_0\_43Reb(:,2),'r', 'DisplayName','0-4.3 0-4.3');

% plot(,'.r');

hold on

plot(OhlinsData2\_43\_2\_43Comp(:,1),OhlinsData2\_43\_2\_43Comp(:,2),'b', 'DisplayName','2-4.3 2-4.3');

plot(OhlinsData2\_43\_2\_43Reb(:,1),OhlinsData2\_43\_2\_43Reb(:,2),'b', 'DisplayName','2-4.3 2-4.3');

% plot(,'.c');

hold on

plot(OhlinsData4\_43\_4\_43Comp(:,1),OhlinsData4\_43\_4\_43Comp(:,2), 'Color', [0.6350 0.0780 0.1840], 'DisplayName','4-4.3 4-4.3');

plot(OhlinsData4\_43\_4\_43Reb(:,1),OhlinsData4\_43\_4\_43Reb(:,2), 'Color', [0.6350 0.0780 0.1840], 'DisplayName','4-4.3 4-4.3');

% plot(,'.m');

hold on

plot(OhlinsData6\_43\_6\_43Comp(:,1),OhlinsData6\_43\_6\_43Comp(:,2), 'Color', [1 0.5 0], 'DisplayName','6-4.3 6-4.3');

plot(OhlinsData6\_43\_6\_43Reb(:,1),OhlinsData6\_43\_6\_43Reb(:,2), 'Color', [1 0.5 0], 'DisplayName','6-4.3 6-4.3');

% plot(,'.y');

hold on

plot(OhlinsData10\_43\_10\_43Comp(:,1),OhlinsData10\_43\_10\_43Comp(:,2), 'Color', [0 0.5 0], 'DisplayName','10-4.3 10-4.3');

plot(OhlinsData10\_43\_10\_43Reb(:,1),OhlinsData10\_43\_10\_43Reb(:,2), 'Color', [0 0.5 0], 'DisplayName','10-4.3 10-4.3');

% plot(,'.b');

hold on

plot(OhlinsData15\_43\_15\_43Comp(:,1),OhlinsData15\_43\_15\_43Comp(:,2),'c', 'DisplayName','15-4.3 15-4.3');

plot(OhlinsData15\_43\_15\_43Reb(:,1),OhlinsData15\_43\_15\_43Reb(:,2),'c', 'DisplayName','15-4.3 15-4.3');

% plot(,'.k');

hold on

plot(OhlinsData25\_43\_25\_43Comp(:,1),OhlinsData25\_43\_25\_43Comp(:,2),'m', 'DisplayName','25-4.3 25-4.3');

plot(OhlinsData25\_43\_25\_43Reb(:,1),OhlinsData25\_43\_25\_43Reb(:,2),'m', 'DisplayName','25-4.3 25-4.3');

grid on

title('Force vs. Velocity')

xlabel('Velocity (mm/s)');

ylabel('Force (N)');

legend

% legend('0-4.3 0-4.3', '2-4.3 2-4.3', '4-4.3 4-4.3', '6-4.3 6-4.3', '10-4.3 10-4.3', '15-4.3 15-4.3', '25-4.3 25-4.3')

f = figure('Name',sprintf('Manufacturer vs. Tested Damping Plots, Ohlins TTX25 High-Speed Adjustment Variation'),'NumberTitle','off');

set(f,'units','normalized','outerposition',[0,0,1,1]);

% plot(,'.g');

hold on

plot(OhlinsData0\_43\_0\_43Comp(:,1),OhlinsData0\_43\_0\_43Comp(:,2),'r', 'DisplayName', '0-4.3 0-4.3');

plot(OhlinsData0\_43\_0\_43Reb(:,1),OhlinsData0\_43\_0\_43Reb(:,2),'r', 'DisplayName', '0-4.3 0-4.3');

% plot(,'.r');

hold on

plot(OhlinsData0\_3\_0\_3Comp(:,1),OhlinsData0\_3\_0\_3Comp(:,2), 'Color', [0.75 0.75 0], 'DisplayName', '0-3 0-3');

plot(OhlinsData0\_3\_0\_3Reb(:,1),OhlinsData0\_3\_0\_3Reb(:,2), 'Color', [0.75 0.75 0], 'DisplayName', '0-3 0-3');

% plot(,'.c');

hold on

plot(OhlinsData0\_2\_0\_2Comp(:,1),OhlinsData0\_2\_0\_2Comp(:,2), 'Color', [0 0.75 0.75], 'DisplayName', '0-2 0-2');

plot(OhlinsData0\_2\_0\_2Reb(:,1),OhlinsData0\_2\_0\_2Reb(:,2), 'Color', [0 0.75 0.75], 'DisplayName', '0-2 0-2');

% plot(,'.m');

hold on

plot(OhlinsData0\_1\_0\_1Comp(:,1),OhlinsData0\_1\_0\_1Comp(:,2), 'Color', [0.3010 0.7450 0.9330], 'DisplayName', '0-1 0-1');

plot(OhlinsData0\_1\_0\_1Reb(:,1),OhlinsData0\_1\_0\_1Reb(:,2), 'Color', [0.3010 0.7450 0.9330], 'DisplayName', '0-1 0-1');

% plot(,'.y');

hold on

plot(OhlinsData0\_0\_0\_0Comp(:,1),OhlinsData0\_0\_0\_0Comp(:,2), 'Color', [0.2 0 0], 'DisplayName', '0-0 0-0');

plot(OhlinsData0\_0\_0\_0Reb(:,1),OhlinsData0\_0\_0\_0Reb(:,2), 'Color', [0.2 0 0], 'DisplayName', '0-0 0-0');

grid on

title('Force vs. Velocity');

xlabel('Velocity (mm/s)');

ylabel('Force (N)');

legend

toc