Table of Contents

```
% Author(s): Vishnu Duriseti, Taiyo Takanashi Forbes
% Assignment title: Project 1, Case 1
% Purpose: Converting mV from Sensor data into Thrust Load Data
% Creation date: 10/16/2023
% Revisions: 10/28, 10/30, 11/1, 11/2, 11/3
```

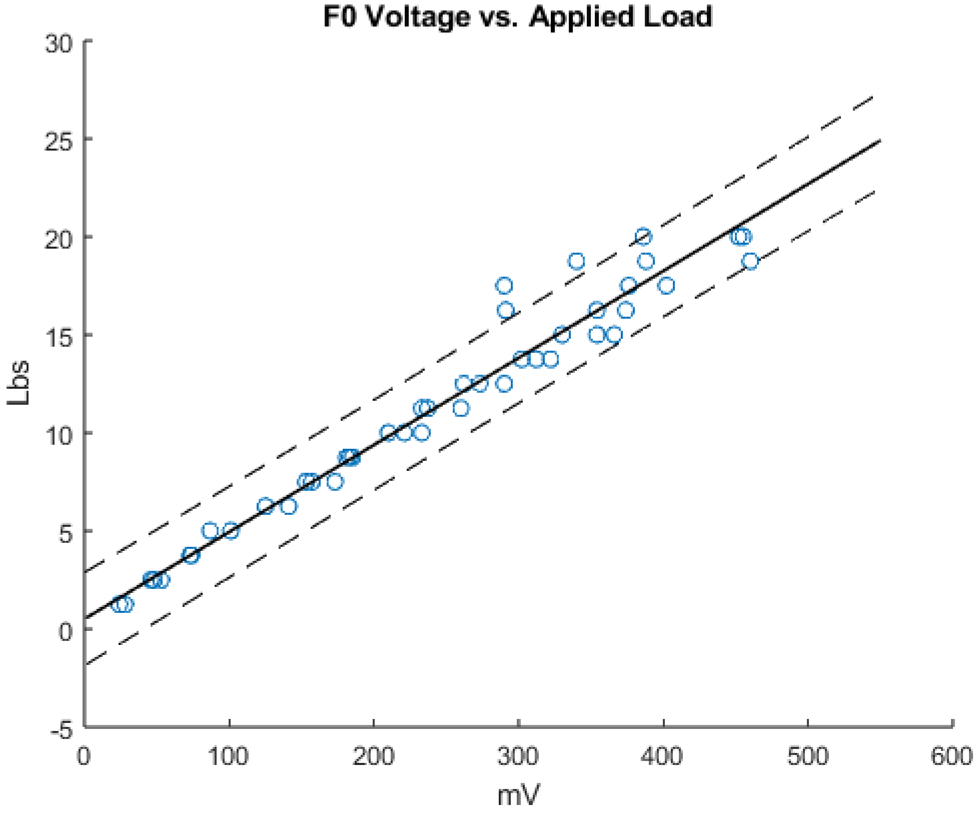
Housekeeping

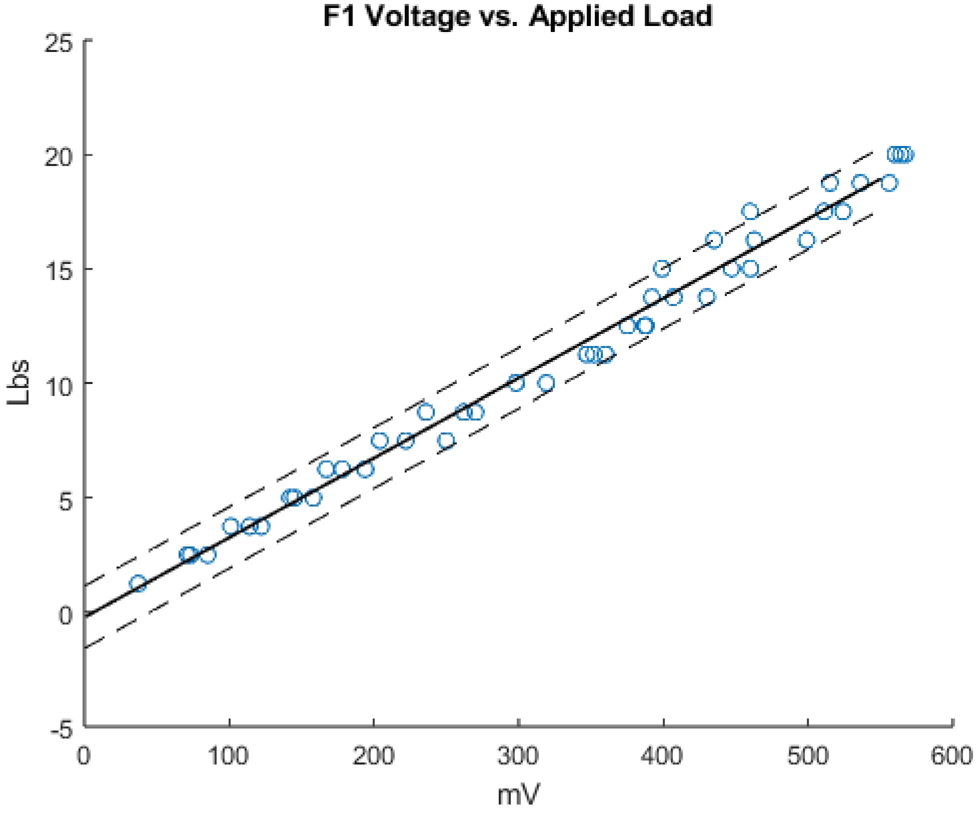
```
close all; clc; clear all;
```

Reading data in and creating a linear estimate of mV --> Load

```
data = readmatrix("Static Test Stand Calibration Case 1.xlsx");

applied_load = data(:,1);
F0_offset =    data(:,2);
F1_offset =    data(:,3);
F0 =           data(:,4);
F1 =           data(:,5);

mV = linspace(1, 550, 48);

% creating a polyfit
[coeff_0, S0] = polyfit(F0, applied_load, 1);
[coeff_1, S1] = polyfit(F1, applied_load, 1);

% polyval --> best-fit line
[peepee, del0] = polyval(coeff_0, mV, S0);
[poopoo, del1] = polyval(coeff_1, mV, S1);

% Calculating 95% confidence interval
upper_std_0 = peepee + (2 .* del0);
lower_std_0 = peepee - (2 .* del0);
upper_std_1 = poopoo + (2 .* del1);
lower_std_1 = poopoo - (2 .* del1);

% plotting Sensor Data (w/ subplots bc I'm chill like that)
```

```
figure(); hold on
title('F0 Voltage vs. Applied Load')
scatter(F0, applied_load)
plot(mV, peepee, 'k', 'Linewidth', 1.2)
plot(mV, upper_std_0, 'k--')
plot(mV, lower_std_0, 'k--')
xlabel('mV')
ylabel('Lbs')

figure(); hold on
title('F1 Voltage vs. Applied Load')
scatter(F1, applied_load)
plot(mV, poopoo, 'k', 'LineWidth', 1.2)
plot(mV, upper_std_1, 'k--')
plot(mV, lower_std_1, 'k--')
xlabel('mV')
ylabel('Lbs')

% calculating effective voltage
effective_voltage_0 = F0; %% not right, I don't have the zeroed out voltage
 data reading
effective_voltage_1 = F1;

% calculating Total Load (Force)

load_0 = applied_load .* (effective_voltage_0 ./ (effective_voltage_0 +
 effective_voltage_1));
load_1 = applied_load .* (effective_voltage_1 ./ (effective_voltage_0 +
 effective_voltage_1));
```


F0 Voltage vs. Applied Load
30
25
20
15
10
5
0
-5
Lbs
0
100
200
300
400
500
600
mV


F1 Voltage vs. Applied Load
25
20
15
10
5
0
-5
Lbs
0
100
200
300
400
500
600
mV

Loading in all the test data

```
test1 = load('testrun1.mat');
test2 = load('testrun2.mat');
test3 = load('testrun3.mat');
test4 = load('testrun4.mat');
test5 = load('testrun5.mat');
test6 = load('testrun6.mat');
test7 = load('testrun7.mat');
test8 = load('testrun8.mat');
test9 = load('testrun9.mat');
test10 = load('testrun10.mat');

time = test1.time; % Same for all (I assume Hz is the same)

% processing data
[CH0_1, CH1_1, avg_peak_1, erm0_1, erm1_1, time0_1, time1_1, errTime0_1,
 errTime1_1, scoopdiwoop0_1, scoopdiwoop1_1] = magicDataConversion(test1,
 coeff_0, coeff_1, S0, S1);
[CH0_2, CH1_2, avg_peak_2, erm0_2, erm1_2, time0_2, time1_2, errTime0_2,
 errTime1_2, scoopdiwoop0_2, scoopdiwoop1_2] = magicDataConversion(test2,
 coeff_0, coeff_1, S0, S1);
[CH0_3, CH1_3, avg_peak_3, erm0_3, erm1_3, time0_3, time1_3, errTime0_3,
 errTime1_3, scoopdiwoop0_3, scoopdiwoop1_3] = magicDataConversion(test3,
 coeff_0, coeff_1, S0, S1);
[CH0_4, CH1_4, avg_peak_4, erm0_4, erm1_4, time0_4, time1_4, errTime0_4,
 errTime1_4, scoopdiwoop0_4, scoopdiwoop1_4] = magicDataConversion(test4,
 coeff_0, coeff_1, S0, S1);
[CH0_5, CH1_5, avg_peak_5, erm0_5, erm1_5, time0_5, time1_5, errTime0_5,
 errTime1_5, scoopdiwoop0_5, scoopdiwoop1_5] = magicDataConversion(test5,
 coeff_0, coeff_1, S0, S1);
[CH0_6, CH1_6, avg_peak_6, erm0_6, erm1_6, time0_6, time1_6, errTime0_6,
 errTime1_6, scoopdiwoop0_6, scoopdiwoop1_6] = magicDataConversion(test6,
 coeff_0, coeff_1, S0, S1);
[CH0_7, CH1_7, avg_peak_7, erm0_7, erm1_7, time0_7, time1_7, errTime0_7,
 errTime1_7, scoopdiwoop0_7, scoopdiwoop1_7] = magicDataConversion(test7,
 coeff_0, coeff_1, S0, S1);
[CH0_8, CH1_8, avg_peak_8, erm0_8, erm1_8, time0_8, time1_8, errTime0_8,
 errTime1_8, scoopdiwoop0_8, scoopdiwoop1_8] = magicDataConversion(test8,
 coeff_0, coeff_1, S0, S1);
[CH0_9, CH1_9, avg_peak_9, erm0_9, erm1_9, time0_9, time1_9, errTime0_9,
 errTime1_9, scoopdiwoop0_9, scoopdiwoop1_9] = magicDataConversion(test9,
 coeff_0, coeff_1, S0, S1);
[CH0_10, CH1_10, avg_peak_10, erm0_10, erm1_10, time0_10, time1_10,
 errTime0_10, errTime1_10, scoopdiwoop0_10, scoopdiwoop1_10] =
 magicDataConversion(test10, coeff_0, coeff_1, S0, S1);

avg_peak_matrix = [avg_peak_10, avg_peak_9, avg_peak_8, avg_peak_7,
 avg_peak_6, avg_peak_5, avg_peak_4, avg_peak_3, avg_peak_2, avg_peak_1];
pinnochio_avg_peak = (avg_peak_10 + avg_peak_9 + avg_peak_8 + avg_peak_7
 + avg_peak_6 + avg_peak_5 + avg_peak_4 + avg_peak_3 + avg_peak_2 +
 avg_peak_1) / 10 % no semicolon to display in terminal
```

```
% pinnochios_stds = std(avg_peak_matrix) % I'm bout to.... I'm bout to... I'm
 bout to.... - Drake
```

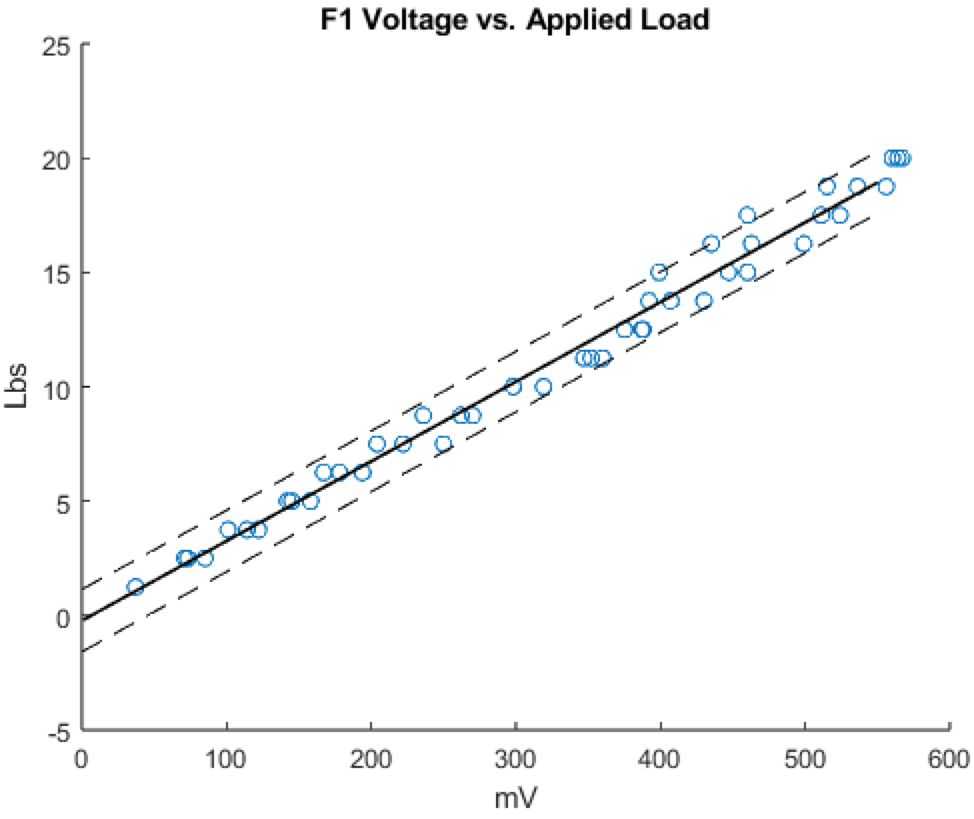
Plotting!!!!! -- with cool colors oooooooo

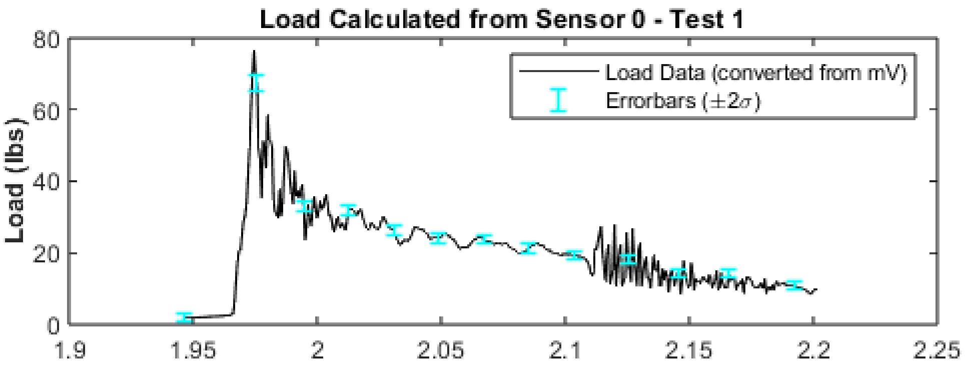
```
figure();
subplot(2, 1, 1)
plot(time0_1, CH0_1, 'k'); hold on
errorbar(errTime0_1, scoopdiwoop0_1,
 erm0_1, 'c', 'LineStyle','none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 0 - Test 1')
hold off

subplot(2, 1, 2)
plot(time1_1, CH1_1, 'k'); hold on
errorbar(errTime1_1, scoopdiwoop1_1,
 erm1_1, 'c', 'LineStyle', 'none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
xlabel('Time (s)', 'FontWeight', 'bold')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 1 - Test 1')
hold off

figure();
subplot(2, 1, 1)
plot(time0_2, CH0_2, 'k'); hold on
errorbar(errTime0_2, scoopdiwoop0_2,
 erm0_2, 'c', 'LineStyle','none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 0 - Test 2')
hold off

subplot(2, 1, 2)
plot(time1_2, CH1_2, 'k'); hold on
errorbar(errTime1_2, scoopdiwoop1_2,
 erm1_2, 'c', 'LineStyle', 'none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
xlabel('Time (s)', 'FontWeight', 'bold')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 1 - Test 2')
hold off

figure();
subplot(2, 1, 1)
plot(time0_3, CH0_3, 'k'); hold on
errorbar(errTime0_3, scoopdiwoop0_3,
 erm0_3, 'c', 'LineStyle','none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 0 - Test 3')
```

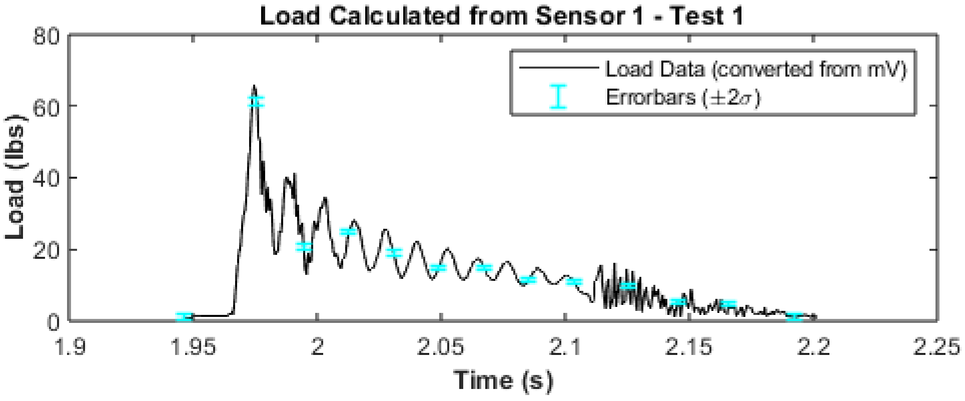
```
hold off

subplot(2, 1, 2)
plot(time1_3, CH1_3, 'k'); hold on
errorbar(errTime1_3, scoopdiwoop1_3,
 erm1_3, 'c', 'LineStyle', 'none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
xlabel('Time (s)', 'FontWeight', 'bold')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 1 - Test 3')
hold off

figure();
subplot(2, 1, 1)
plot(time0_4, CH0_4, 'k'); hold on
errorbar(errTime0_4, scoopdiwoop0_4,
 erm0_4, 'c', 'LineStyle','none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 0 - Test 4')
hold off

subplot(2, 1, 2)
plot(time1_4, CH1_4, 'k'); hold on
errorbar(errTime1_4, scoopdiwoop1_4,
 erm1_4, 'c', 'LineStyle', 'none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
xlabel('Time (s)', 'FontWeight', 'bold')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 1 - Test 4')
hold off

figure();
subplot(2, 1, 1)
plot(time0_5, CH0_5, 'k'); hold on
errorbar(errTime0_5, scoopdiwoop0_5,
 erm0_5, 'c', 'LineStyle','none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 0 - Test 5')
hold off

subplot(2, 1, 2)
plot(time1_5, CH1_5, 'k'); hold on
errorbar(errTime1_5, scoopdiwoop1_5,
 erm1_5, 'c', 'LineStyle', 'none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
xlabel('Time (s)', 'FontWeight', 'bold')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 1 - Test 5')
hold off

figure();
subplot(2, 1, 1)
```

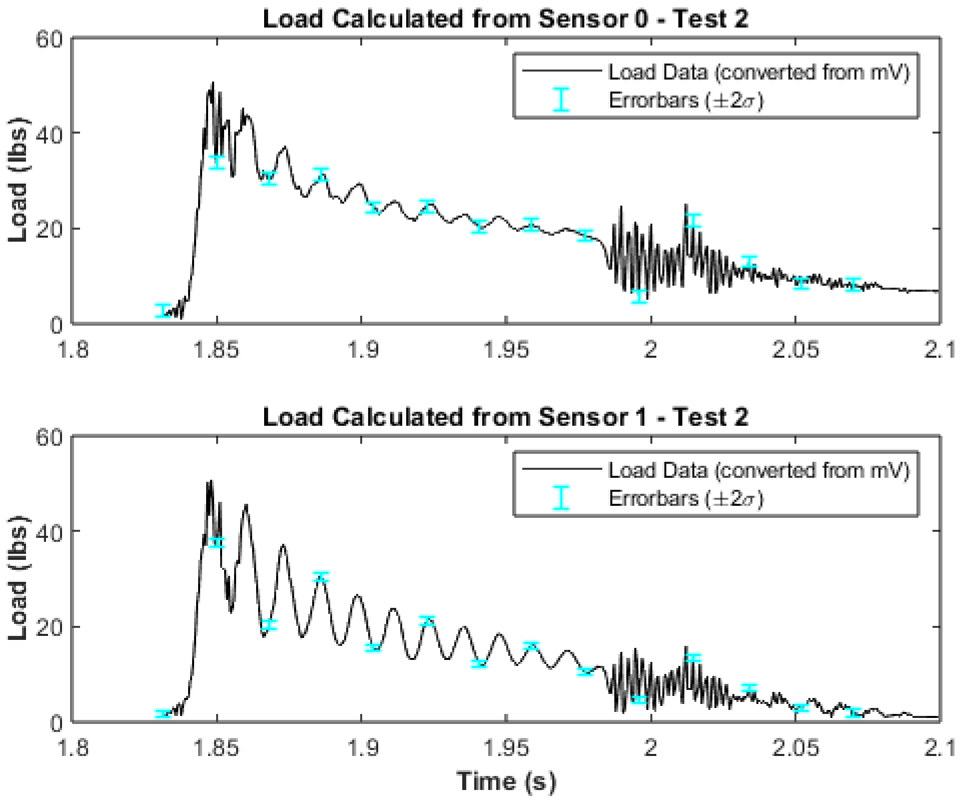
```
plot(time0_6, CH0_6, 'k'); hold on
errorbar(errTime0_6, scoopdiwoop0_6,
 erm0_6, 'c', 'LineStyle','none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 0 - Test 6')
hold off

subplot(2, 1, 2)
plot(time1_6, CH1_6, 'k'); hold on
errorbar(errTime1_6, scoopdiwoop1_6,
 erm1_6, 'c', 'LineStyle', 'none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
xlabel('Time (s)', 'FontWeight', 'bold')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 1 - Test 6')
hold off

figure();
subplot(2, 1, 1)
plot(time0_7, CH0_7, 'k'); hold on
errorbar(errTime0_7, scoopdiwoop0_7,
 erm0_7, 'c', 'LineStyle','none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 0 - Test 7')
hold off

subplot(2, 1, 2)
plot(time1_7, CH1_7, 'k'); hold on
errorbar(errTime1_7, scoopdiwoop1_7,
 erm1_7, 'c', 'LineStyle', 'none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
xlabel('Time (s)', 'FontWeight', 'bold')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 1 - Test 7')
hold off

figure();
subplot(2, 1, 1)
plot(time0_8, CH0_8, 'k'); hold on
errorbar(errTime0_8, scoopdiwoop0_8,
 erm0_8, 'c', 'LineStyle','none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 0 - Test 8')
hold off

subplot(2, 1, 2)
plot(time1_8, CH1_8, 'k'); hold on
errorbar(errTime1_8, scoopdiwoop1_8,
 erm1_8, 'c', 'LineStyle', 'none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
xlabel('Time (s)', 'FontWeight', 'bold')
```

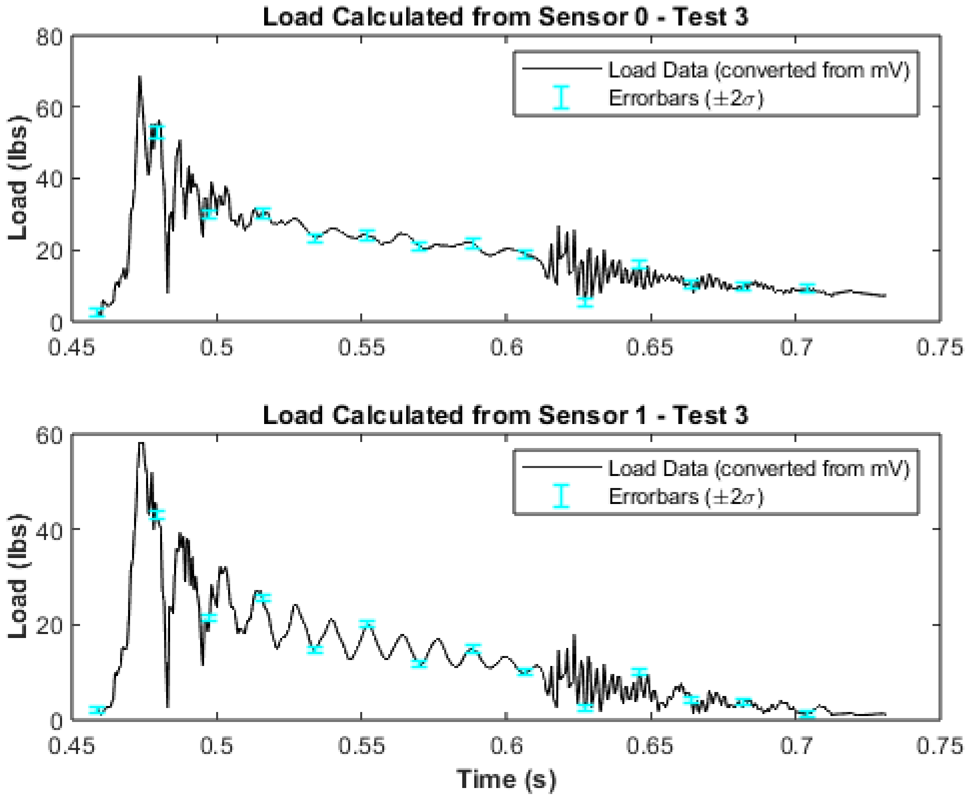
```
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 1 - Test 8')
hold off

figure();
subplot(2, 1, 1)
plot(time0_9, CH0_9, 'k'); hold on
errorbar(errTime0_9, scoopdiwoop0_9,
 erm0_9, 'c', 'LineStyle','none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 0 - Test 9')
hold off

subplot(2, 1, 2)
plot(time1_9, CH1_9, 'k'); hold on
errorbar(errTime1_9, scoopdiwoop1_9,
 erm1_9, 'c', 'LineStyle', 'none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
xlabel('Time (s)', 'FontWeight', 'bold')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 1 - Test 9')
hold off

figure();
subplot(2, 1, 1)
plot(time0_10, CH0_10, 'k'); hold on
errorbar(errTime0_10, scoopdiwoop0_10,
 erm0_10, 'c', 'LineStyle','none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 0 - Test 10')
hold off

subplot(2, 1, 2)
plot(time1_10, CH1_10, 'k'); hold on
errorbar(errTime1_10, scoopdiwoop1_10,
 erm1_10, 'c', 'LineStyle', 'none', 'LineWidth', 1.2)
legend('Load Data (converted from mV)', 'Errorbars (\pm2\sigma)')
xlabel('Time (s)', 'FontWeight', 'bold')
ylabel('Load (lbs)', 'FontWeight', 'bold')
title('Load Calculated from Sensor 1 - Test 10')
hold off
```

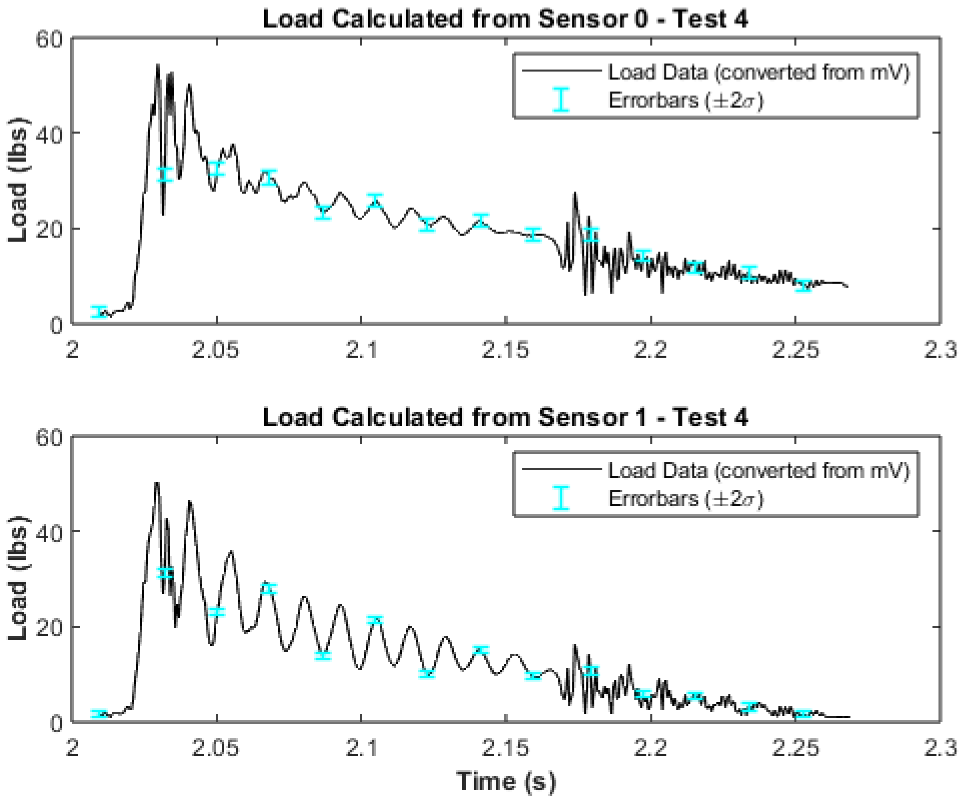

F1 Voltage vs. Applied Load
Lbs
mV
25
20
15
10
5
0
-5
0
100
200
300
400
500
600


Load Calculated from Sensor 0 - Test 1
Load (lbs)
Load Data (converted from mV)
Errorbars (±2σ)
80
60
40
20
0
1.9
1.95
2
2.05
2.1
2.15
2.2
2.25


Load Calculated from Sensor 1 - Test 1
Load (lbs)
Time (s)
Load Data (converted from mV)
Errorbars (±2σ)
80
60
40
20
0
1.9
1.95
2
2.05
2.1
2.15
2.2
2.25

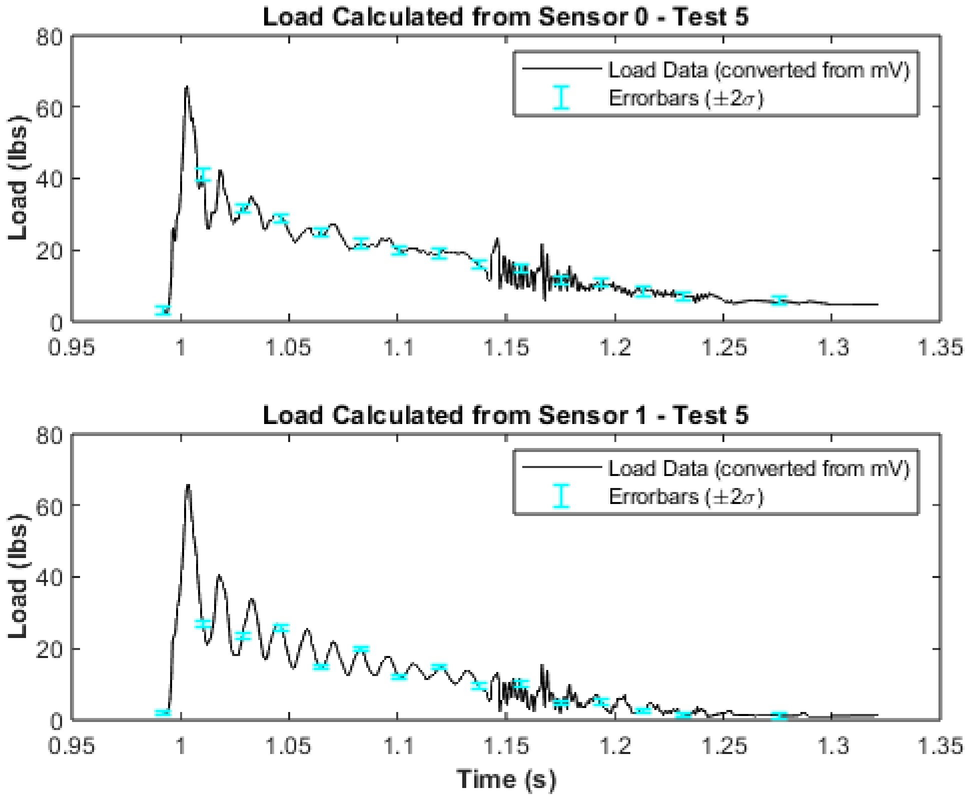

Load Calculated from Sensor 0 - Test 2
Load (lbs)
0
20
40
60
1.8
1.85
1.9
1.95
2
2.05
2.1
Load Data (converted from mV)
Errorbars (±2σ)
Load Calculated from Sensor 1 - Test 2
Load (lbs)
0
20
40
60
1.8
1.85
1.9
1.95
2
2.05
2.1
Time (s)
Load Data (converted from mV)
Errorbars (±2σ)


Load Calculated from Sensor 0 - Test 3
Load (lbs)
0
20
40
60
80
0.45
0.5
0.55
0.6
0.65
0.7
0.75
Load Data (converted from mV)
Errorbars (±2σ)
Load Calculated from Sensor 1 - Test 3
Load (lbs)
0
20
40
60
0.45
0.5
0.55
0.6
0.65
0.7
0.75
Time (s)
Load Data (converted from mV)
Errorbars (±2σ)



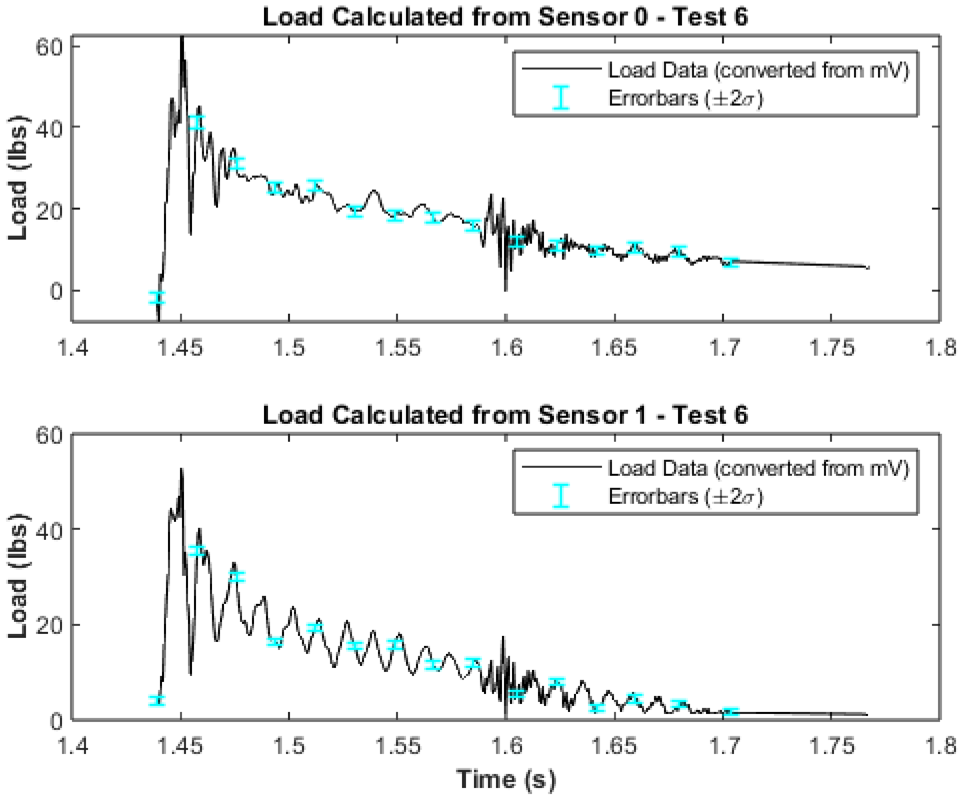
Load Calculated from Sensor 0 - Test 4

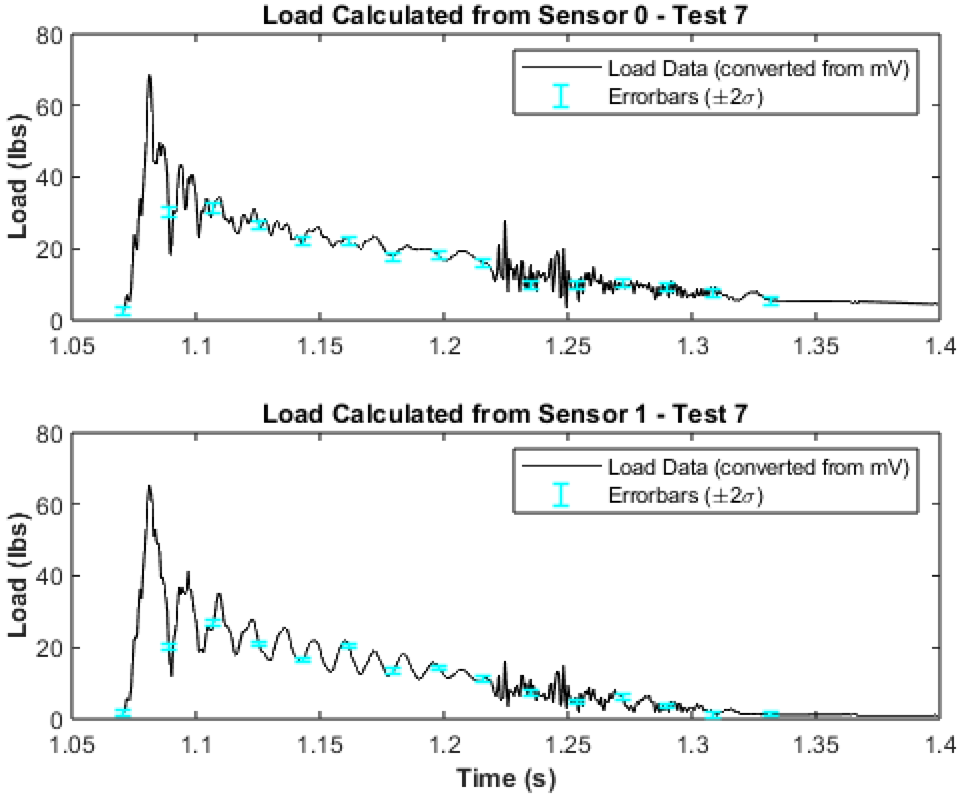
Load Calculated from Sensor 1 - Test 4

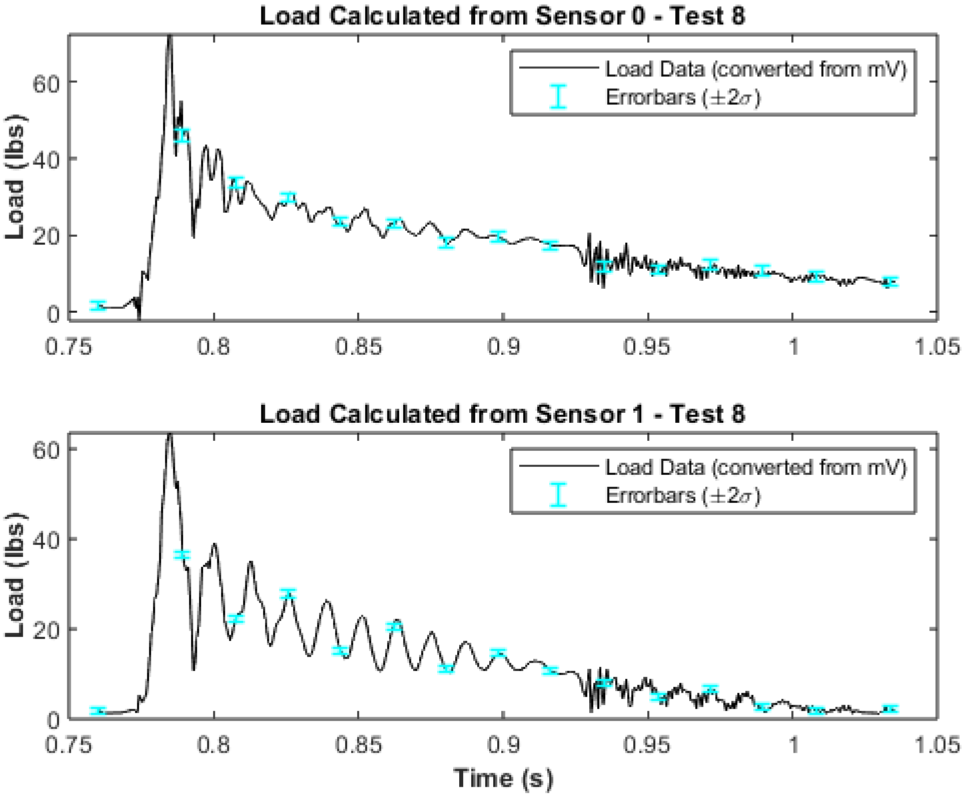


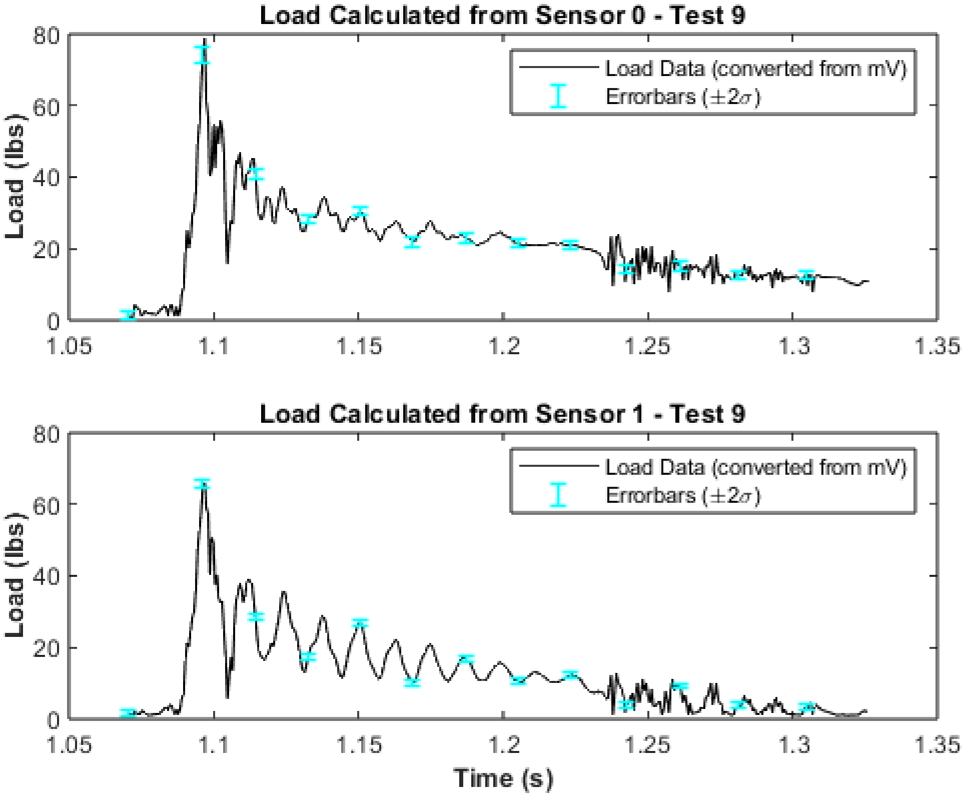
Load Calculated from Sensor 0 - Test 5

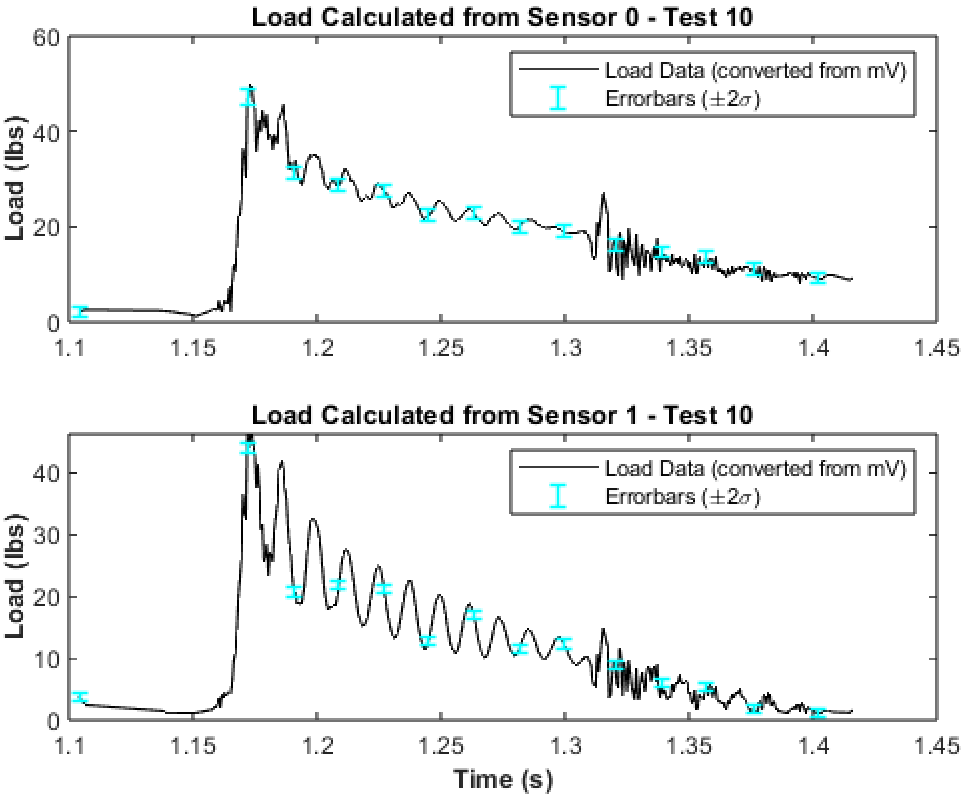
Load Calculated from Sensor 1 - Test 5


Load Calculated from Sensor 0 - Test 6
Load (lbs)
60
40
20
0
1.4
1.45
1.5
1.55
1.6
1.65
1.7
1.75
1.8
Load Data (converted from mV)
Errorbars (±2σ)
Load Calculated from Sensor 1 - Test 6
Load (lbs)
60
40
20
0
1.4
1.45
1.5
1.55
1.6
1.65
1.7
1.75
1.8
Time (s)
Load Data (converted from mV)
Errorbars (±2σ)


Load Calculated from Sensor 0 - Test 7
Load (lbs)
80
60
40
20
0
1.05
1.1
1.15
1.2
1.25
1.3
1.35
1.4
Load Data (converted from mV)
Errorbars (±2σ)
Load Calculated from Sensor 1 - Test 7
Load (lbs)
80
60
40
20
0
1.05
1.1
1.15
1.2
1.25
1.3
1.35
1.4
Time (s)
Load Data (converted from mV)
Errorbars (±2σ)


Load Calculated from Sensor 0 - Test 8
Load (lbs)
0
20
40
60
0.75
0.8
0.85
0.9
0.95
1
1.05
Load Data (converted from mV)
Errorbars (±2σ)
Load Calculated from Sensor 1 - Test 8
Load (lbs)
0
20
40
60
0.75
0.8
0.85
0.9
0.95
1
1.05
Time (s)
Load Data (converted from mV)
Errorbars (±2σ)


Load Calculated from Sensor 0 - Test 9
Load (lbs)
0
20
40
60
80
1.05
1.1
1.15
1.2
1.25
1.3
1.35
Load Data (converted from mV)
Errorbars (±2σ)
Load Calculated from Sensor 1 - Test 9
Load (lbs)
0
20
40
60
80
1.05
1.1
1.15
1.2
1.25
1.3
1.35
Time (s)
Load Data (converted from mV)
Errorbars (±2σ)



Creating a function to streamline the data pulling process from each test

```
% lalalalalalalalalalalalalalalalalalal
function [CH0_ooglyboogly, CH1_ooglyboogly, avg_peak, errBound_0, errBound_1,
 time_0, time_1, time_err0, time_err1, sensor_err0, sensor_err1] =
 magicDataConversion(test, coeff_0, coeff_1, S0, S1)
    % x = 1:5:
    time = test.time;
    CH0 = test.mV(:, 1);
    CH1 = test.mV(:, 2);

    % assuming 1 mV offset in CH1, 0 mV in CH0
    CH1 = CH1 - 1;

    % convert mV into lbs using fits first calculated
    [CH0, errBound_giggly] = polyval(coeff_0, CH0, S0);
    [CH1, errBound_diddly] = polyval(coeff_1, CH1, S1);

    % erase erranious values of test where mV < 1
    eepymeepyuno = CH0 >= 1;
    eepymeepydos = CH1 >= 1;

    CH0_ooglyboogly = CH0(eepymeepydos);
    time_0 = time(eepymeepydos);
    errBound_g_0 = errBound_giggly(eepymeepydos);

    CH1_ooglyboogly = CH1(eepymeepydos);
    time_1 = time(eepymeepydos);
    errBound_g_1 = errBound_diddly(eepymeepydos);

    % calculating peaks
    [peak0, index0] = max(CH0, [], 'all');
    [peak1, index1] = max(CH1, [], 'all');
    avg_peak = (peak0 + peak1) / 2;
    avg_peak_error = 2 * (errBound_giggly(index0) + errBound_diddly(index1)) /
 2;

    % cutting error values to only x number of times
    x = 30;
    errBound_0 = errBound_g_0(1:x:end);
    time_err0 = time_0(1:x:end);
    sensor_err0 = CH0_ooglyboogly(1:x:end);

    errBound_1 = errBound_g_1(1:x:end);
    time_err1 = time_1(1:x:end);
    sensor_err1 = CH1_ooglyboogly(1:x:end);
    disp(['The error in the Average Peak: ', num2str(avg_peak_error)])
end
```

```
The error in the Average Peak: 3.3474
The error in the Average Peak: 2.5666
The error in the Average Peak: 3.0685
The error in the Average Peak: 2.6435
The error in the Average Peak: 3.0921
The error in the Average Peak: 2.853
The error in the Average Peak: 3.1536
The error in the Average Peak: 3.2228
The error in the Average Peak: 3.4141
The error in the Average Peak: 2.498

pinnochio_avg_peak =

   61.6375
```

respectfully, we are in tears - V, T

Published with MATLAB® R2022a