Homework 1

Louis Lin

January 11, 2021

University of California, San Diego

Prof. José I. Restrepo

SE 211 Advance Structural Concrete

Contents

Figure

Introduction

In this report, the mechanical behavior of reinforcing steel bar used in reinforced concrete is investigated using three separate test data. The first portion of the report focuses on the material properties of the rebar: how is the strength of a rebar derived from test data. The second portion investigates the relationship between strain in a steel rebar when it is in compression and tension. The last portion of this report investigates the models used to simplify the stress-strain curve for steel rebar under cyclic loading.

Part I – Modeling Tensile Stress-Strain Response of Mild Steel Reinforcement

Question 1.

In this part of the assignment, the objective was to determine the properties of a no. 11 ASTM A706 mild steel reinforcement bar from test data. These properties being the Young's modulus E_s, the onset of strain hardening ϵ_{sh}, the tensile strength f_{su}, and unform strain ϵ_{su}. The test data had to be filtered with a moving-average algorithm in order to remove high-frequency forces or stress and/or strain oscillations.

The moving-average algorithm smooths the data set by considering the average values of points around each data point. For each data point in the set of data, its new value, after passing through the function, is the average of a range of data before and after the data point. This range is chosen such that the variation between points due to noise is smoothed out. For the general case, the larger the range, the smoother the set of data, unless the algorithm is applied where there are sharp changes in the data.

For the general steel rebar, there are three segments of its stress-strain curve. The first segment is referred to as the elastic region, where the stress-strain relationship is linear. This region is bounded by $(0,0)$ and (ϵ_y, f_y), where y signifies yielding. The second segment is referred to as the Lüder plateau or yield plateau, where the stress is assumed to be a constant yield stress value. This region is bounded by (ϵ_y, f_y) and (ϵ_{sh}, f_{sh}) where sh signifies strain hardening. The last segment of the curve is nonlinear, where the stress and strain relationship vary due to strain hardening of the steel. This region is bounded by (ϵ_{sh}, f_{sh}) and (ϵ_{su}, f_{su}) where su signifies the ultimate coordinates. Refer to Figure 1 by Dodd Restreppo (1995).

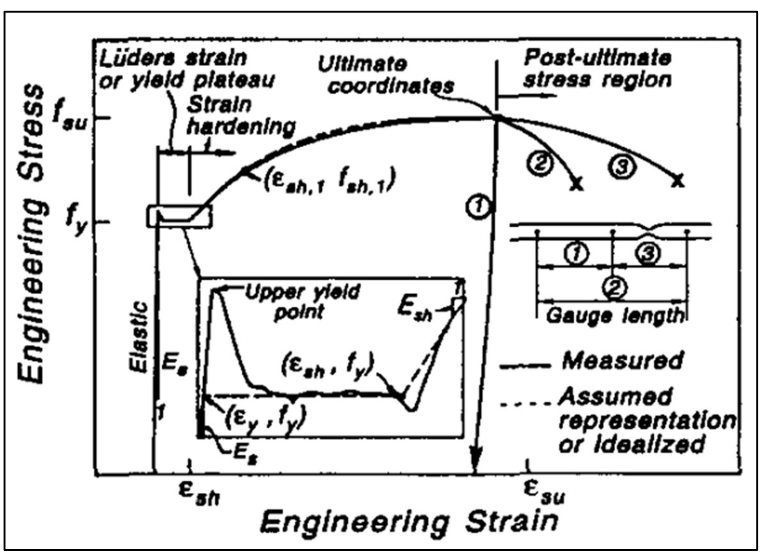
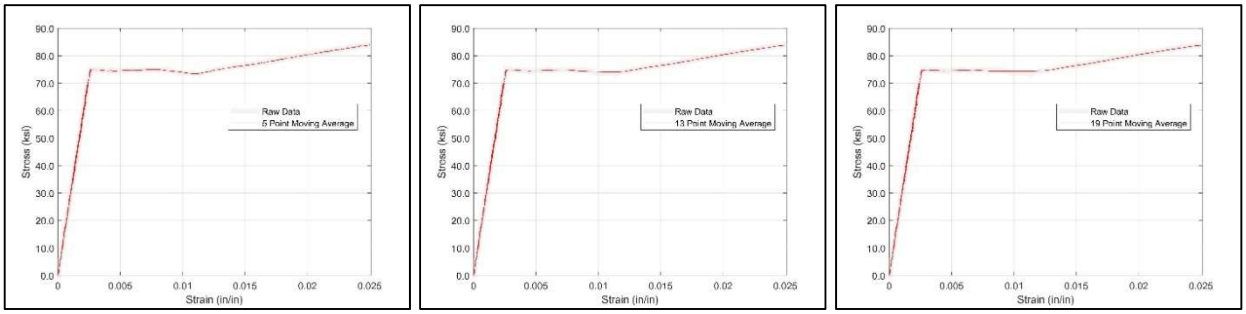


Figure 1 Monotonic Tensile Curve for Mild Steel (Dodd, Restrepo 1995)

Part (i) Determine f_y and ϵ_{sh}

The yielding stress is found to be the minimum of the stress values between ϵ_y and ϵ_{sh}. This stress represents the value that is assumed for the entire yield plateau. ϵ_{sh} is found where the onset of strain hardening is consistent. The data was smoothed using 5, 13, and 19 point moving average and showed in Figure 2 (a-c), larger figures can be found in appendix A. The 19-point moving average was used for analysis of the elastic and plateau region since it was adequate in removing most of the noise in the data.



(a) (b) (c)

Figure 2 Filtering the Elastic Region and Yield Plateau Portions of Steel Rebar Stress-Strain Curve

ϵ_y was chosen at the end of the linear region of the graph and ϵ_{sh} was chosen after the dip in the yield plateau and the slope was consistent. The strain at the end of the elastic region was $\epsilon_y = 0.00258$ and the strain hardening region began at $\epsilon_{sh} = 0.0145$. The yield stress was taken as $f_y = 74.3$ ksi, the minimum on the yield plateau of the 19-point moving average data. Considering that other filtering techniques or parameters may shift the identified boundary points, a range of acceptable answers should be provided for ϵ_y and ϵ_{sh}, but this would not change the f_y. Note f_{sh} does affect the parameters of the Mander's model described later, as the strain hardening region is fitted to this stress-strain pair.

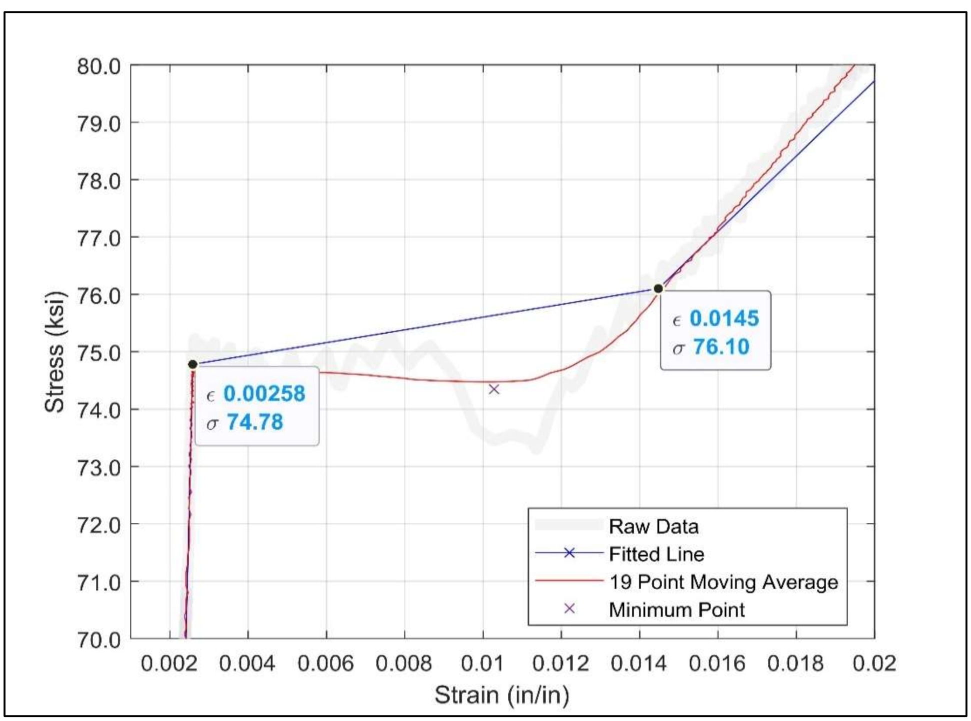


Figure 3 Yielding Stress and Onset of Strain Hardening

Part (ii) Determine E_s

The slope of the linear portion of the stress-strain curve defines the modulus of elasticity for steel rebar. The slope was fitted to the data between $0.05f_y$ and $0.6f_y$ without applying a moving average filter. The slope was found to be $E_s = 29045$ ksi, which is congruent with literature.

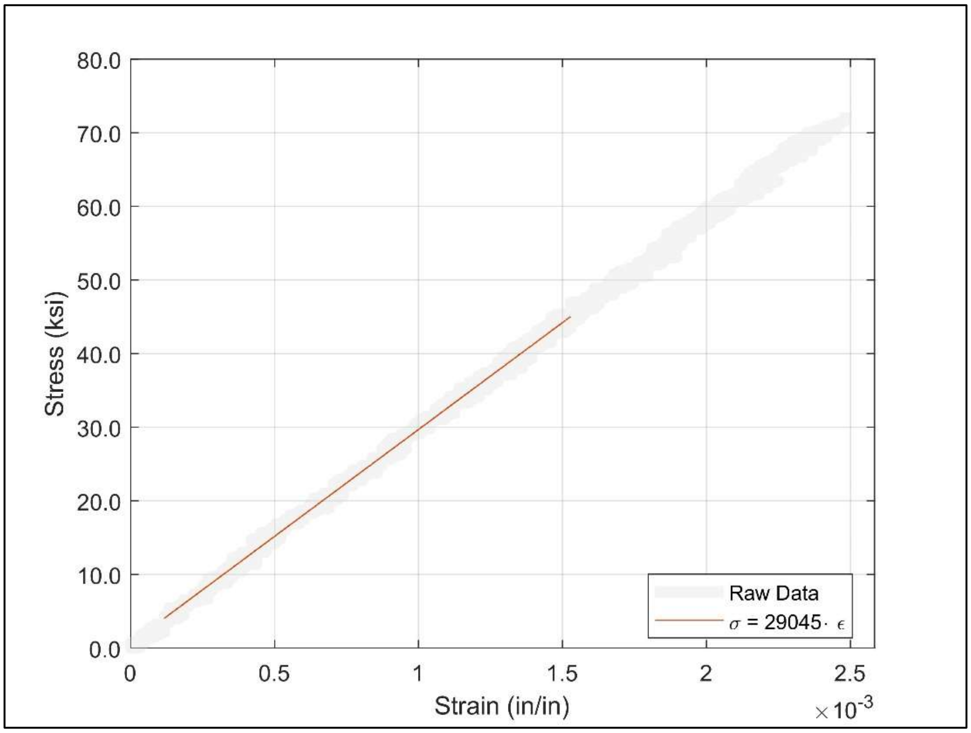
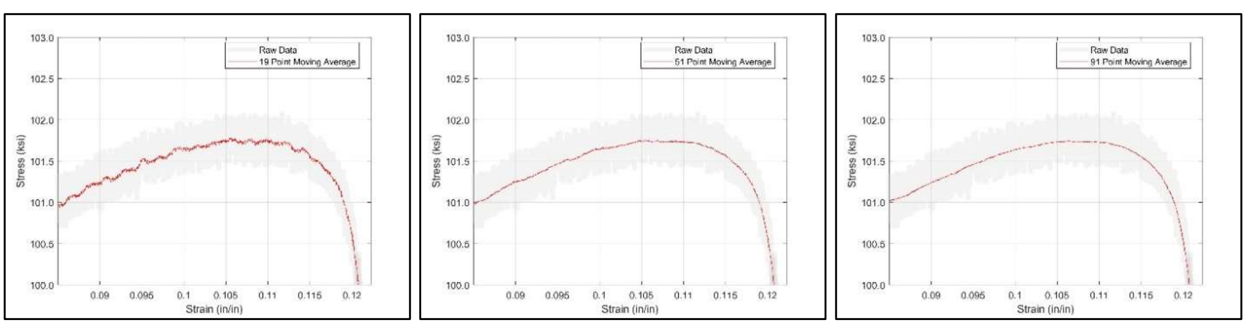


Figure 4 Fitting Linear Line to Elastic Region

Part (iii) Determining f_{su} and ϵ_{su}

The strain hardening portions of the stress-strain curve was smoothed with a higher range moving average function because there were larger variations in the data. In Figure 5(a), the 19-point moving average is shown to be an inadequate range to smooth the data. Upon recommendation of the instructor, 51 and 91 point moving averages were also calculated, which provided much smoother data. larger figures can be found in appendix A



(a) (b) (c)

Figure 5 Filtering Strain Hardening Portion of Stress-Strain Curve

Dodd and Restreppo (1995) states that f_{su} and ϵ_{su} is defined as the point on the strain-hardening region where the stress-strain curve has a zero slope. This is a rigorous definition and was used originally to produce Figure 6. Using a 90-point moving average stress dataset, all points where the slope was zero were plotted. Due to noise in the dataset, that could not be removed with the moving average, there are other points that show up. From this analysis the $\epsilon_{su} = 0.0154$ and $f_{su} = 101.7$ ksi.

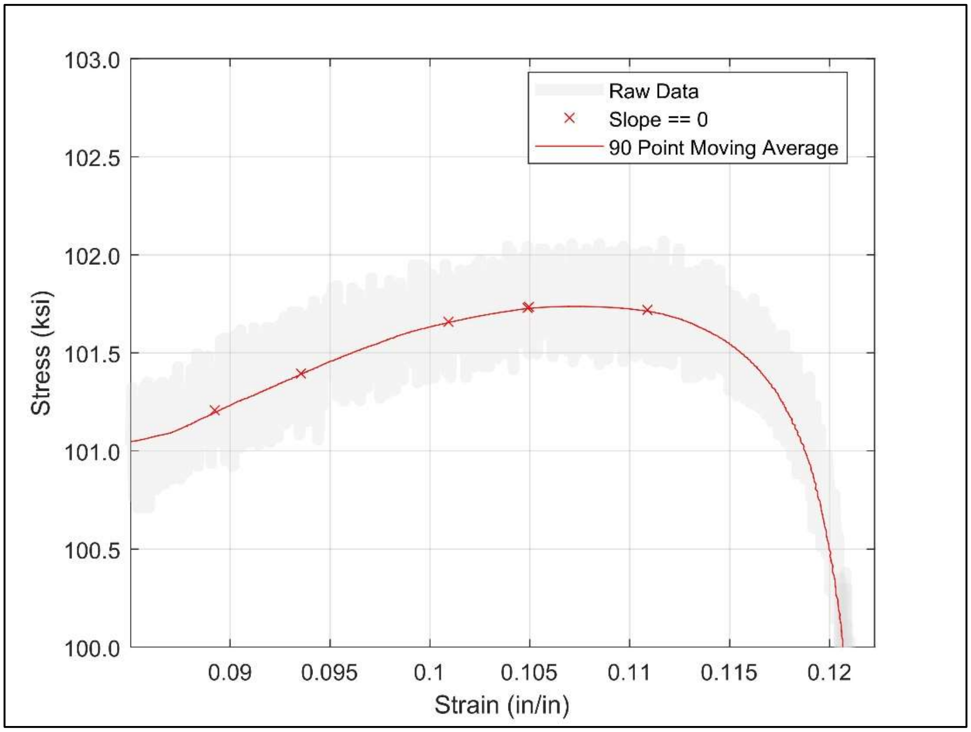


Figure 6 Locating Ultimate Strain and Ultimate Stress

After discussing with the instructor, it is best to actually locate a range of strain and stress values that could actually define (ϵ_{su}, f_{su}) . That is, depending on the range of the moving average function, the strain or stress that is associated with a slope of zero could change. By running various ranges for the moving average [50-100], a range was selected of $\epsilon_{su} = 0.104 - 0.106$ and $f_{su} = 101.65 - 101.7$ ksi.

Question 2.

A very simple model for the stress-strain curve is a trilinear model where the three regions are approximated with three straight lines. The lines are located by the points $(\epsilon_s, f_s) = [(0,0), (\epsilon_y, f_y), (\epsilon_{sh}, f_{su}), \text{and} (\epsilon_{su}, f_{su})]$, where strain hardening ϵ_{sh} is given by a factor of the ultimate strength, $\xi \cdot \epsilon_{su}$. In the report, $\xi = 0.55$. Using the previously found points, the trilinear model is show in Figure 7. This model ignores the yield plateau and has a strain-hardening region that extends from ϵ_y to $\xi \cdot \epsilon_{su}$. The slope of this strain-hardening region is often a related to the modulus of elasticity by a hardening ratio, r, which for the data provided, was calculated as 0.0181. But this could change depending on what the ϵ_{sh} was selected to be. The strain hardening formula is given below.

$$r = \frac{\frac{f_{su}}{f_y} - 1}{\frac{\xi \epsilon_{su}}{\epsilon_y} - 1}$$

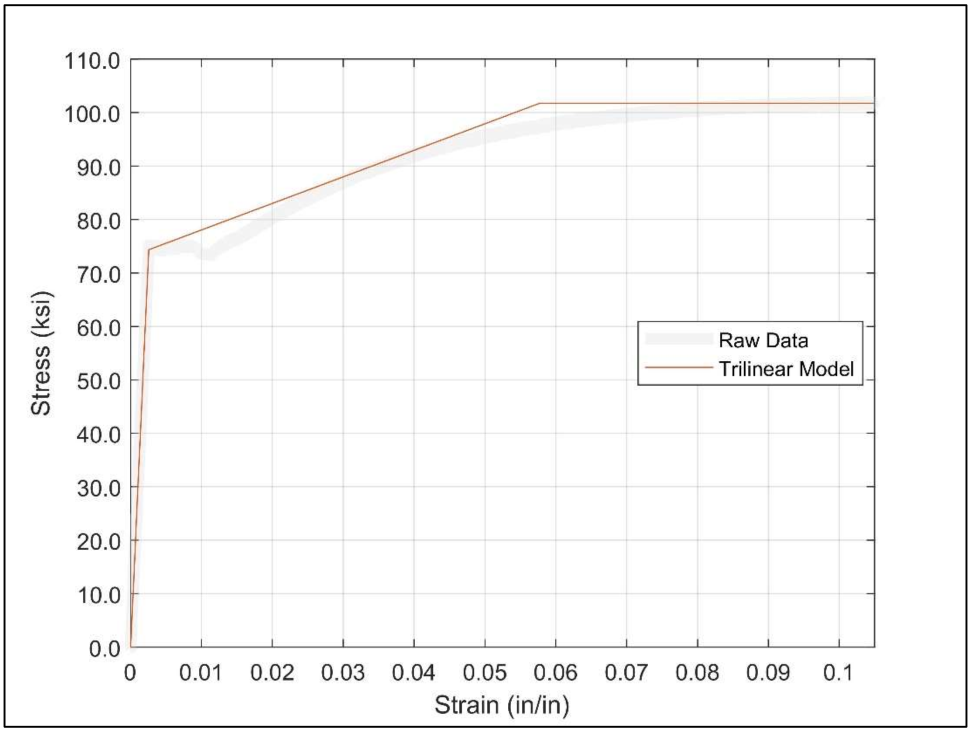


Figure 7 Tri-linear Model of Steel Rebar Stress-Strain Curve

Question 3.

The Mander's model is another model that is used to describe the shape of the stress-strain curve of a steel rebar. This model considers the elastic region and the yield plateau as a linear function given by

$$f_s(\epsilon_s) = \min(f_y, E_s \cdot \epsilon_s)$$

and the shape of the strain hardening portion as a P-power curve. This strain hardening portion of the curve, from $\epsilon_{su} < \epsilon < \epsilon_{sh}$, is given by

$$f_s(\epsilon_s) = f_{su} - (f_{su} - f_y) \cdot \left(\frac{\epsilon_{su} - \epsilon_s}{\epsilon_{su} - \epsilon_{sh}}\right)^P$$

The P-power is found through minimization of the sum of square of the errors between the test data and the predictions. A range of values were used for the P-power from 1 to 8, at increments of 0.2, and the sum of the squared error were calculated. The error was normalized and plotted against the various powers in Figure 8. It is shown that a power of 3 best fitted the provided test. P is typically between 3-6, so this value is reasonable.

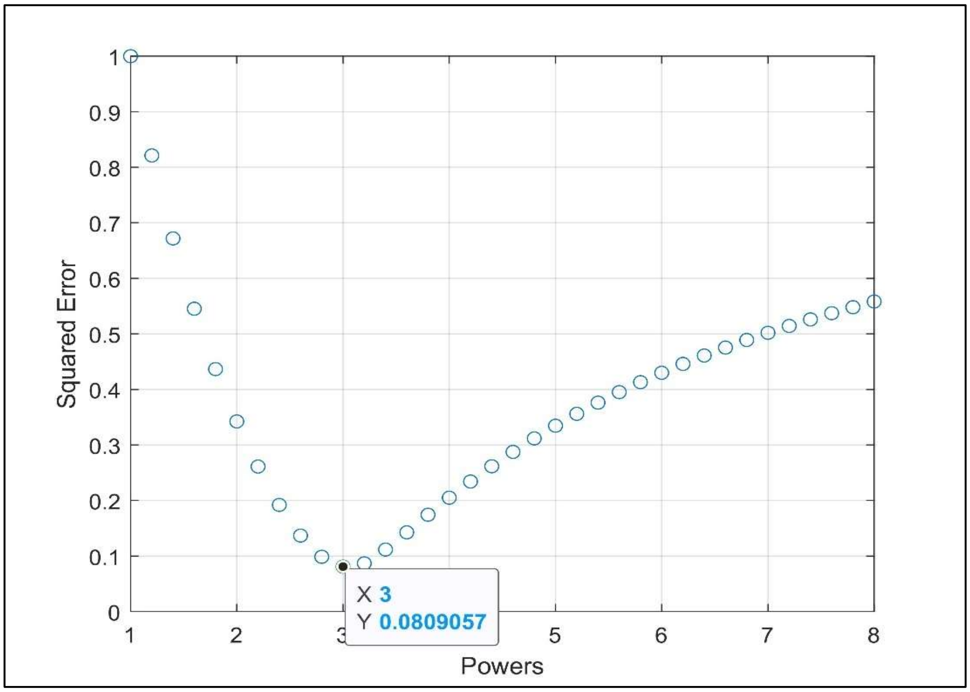


Figure 8 Optimization for Power in Mander's Model

Using the power of 3, the Mander's model was plotted against the test data and shown in Figure 9. By inspection, this model is a good fit for describing the behavior for the stress-strain curve up to ϵ_{su} and f_{su}. A power of three also seems reasonable by inspection. If a power less than 3 is used, the curve would underestimate the strength of the rebar at high strains, and at higher powers, the model may overestimate the strength of the rebar.

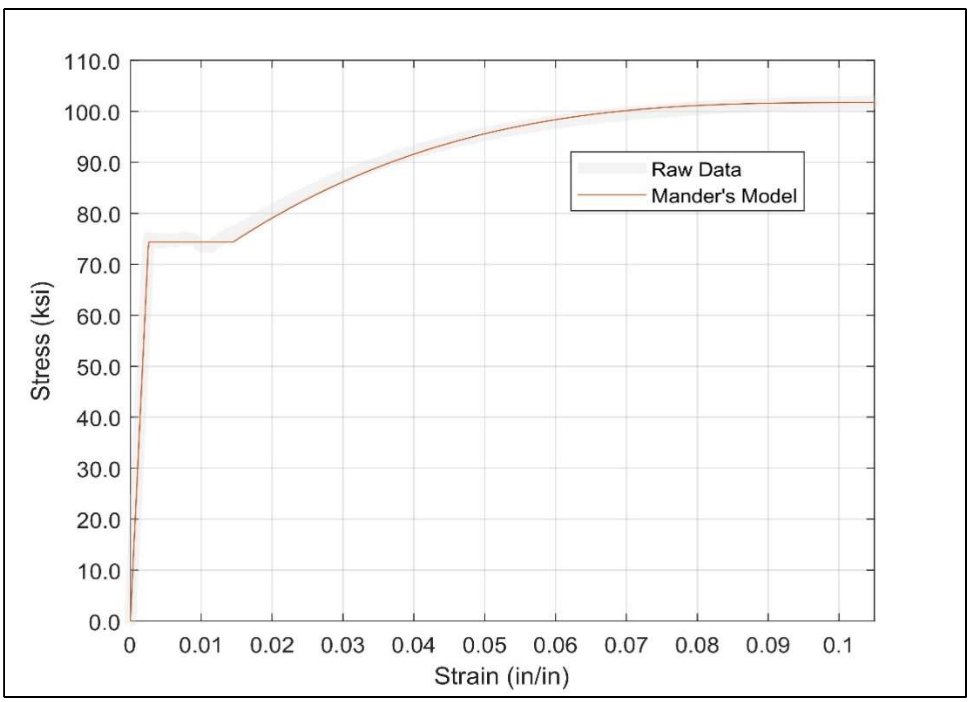


Figure 9 Mander's Model

Part II Comparing Tensile and Compressive Stress-Strain Relationship in True Coordinate System

Question 1

In this portion of the report, the behavior of the steel rebar in compression and tension are explored. There are two systems that can be used to describe the stress-strain relationship of a steel rebar: a true coordinate system and the engineering coordinate system. The engineering coordinate system uses the original area of the rebar and the original length of the rebar to calculate the stress and strain, respectively. In the true coordinate system, the stress and strain are calculated with the shrinkage or expansion of the bar area and the overall change in rebar length. From Dodd Restreppo (1995), it is shown that if the measured engineering stress-strain data is transformed into the true coordinate stress-strain system, the compression and tension curves are very similar, even past the yield plateau. The transformation between the two systems is based upon the assumption that the volume remains constant at low strains with $\nu = 0.5$. The transformation from engineering stress and strain σ and ϵ to the true stress and strain σ' and ϵ' is given by

$$\epsilon' = \ln(1+\epsilon) \qquad \epsilon = e^{\epsilon'} - 1$$
$$\sigma' = \sigma(1+\epsilon) \qquad \sigma = \frac{\sigma'}{e^{\epsilon'}} = \frac{\sigma'}{1+\epsilon}$$

In figure 10, a plot of the raw data for the engineering strain range 0 to 0.025 is given. It shows that the for the region below strain hardening the stress-strain curve for a rebar in the engineering coordinate system, the compression and tension curves are very comparable, almost on top of one another.

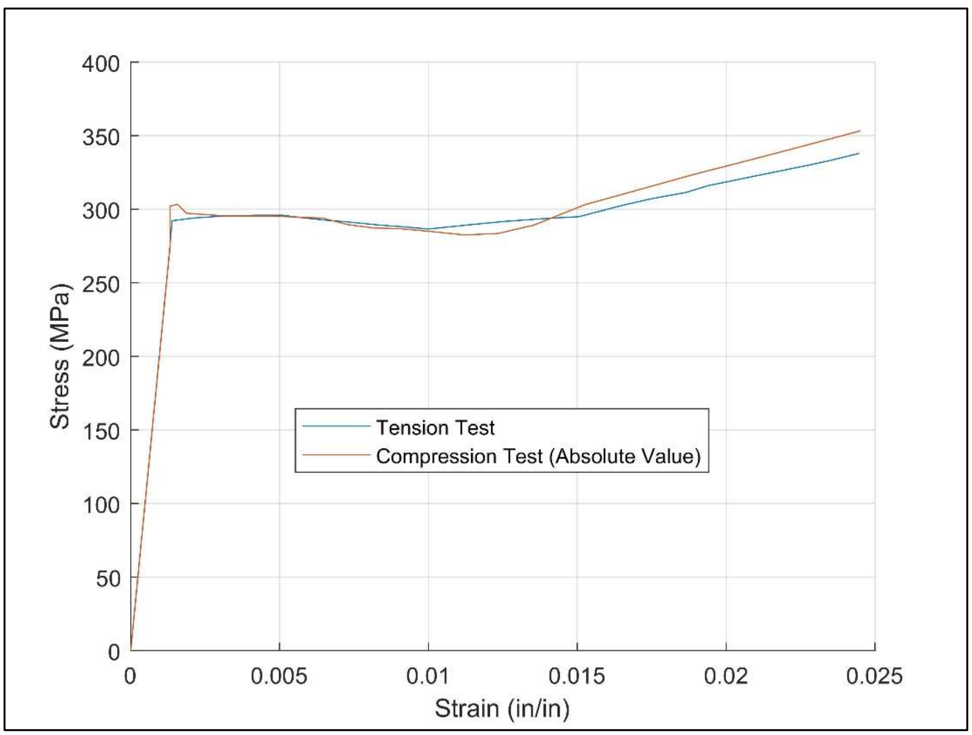


Figure 10 Engineering Stress-Strain Curve up to 0.025

However, examining the two curves after the yield plateau, Figure 11, shows that they diverge greatly in the strain hardening region. The rebar takes a lot more force to deform in compression than in tension.

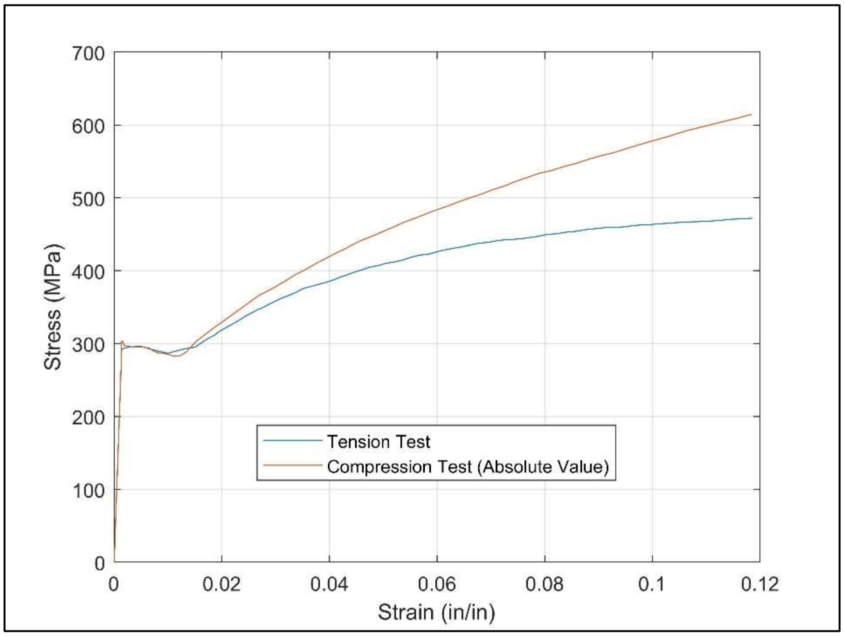


Figure 11 Engineering Stress-Strain Curve

In the true coordinate system, the tension and compression rebar data are on top of each other as shown in Figure 12. The transformation between the engineering and true coordinate system assumes that the Poisson ratio $\nu = 0.5$. This ν value gives that the volume remains constant at small strains. In Stang et al., the Poisson ratio for structural low-carbon steel ranged from 0.41- 0.46 for high strains, greater than 2%. This means that steel would not retain a constant volume when loaded at high strains, leading to variance in its behavior at large strains, when in compression or tension. The transformation between the two systems assuming constant volume is a valid and good assumption for all strains.

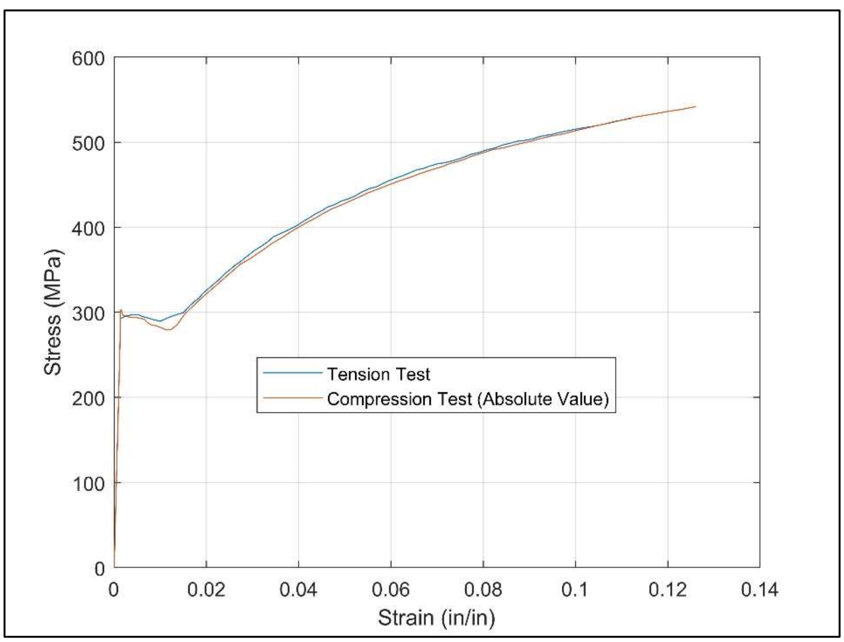


Figure 12 True Coordinate Stress-Strain Curve

Part III Modeling of the Cyclic Response of Reinforcing Steel

In this report two bilinear models for cyclic behavior of rebar are implemented. These two models are the kinematic hardening and isotropic hardening models as shown in Figure 12. The kinematic hardening model assumes that the rebar will always yield along a yield line which simulates the Bauschinger's effect which are parallel in compression and tension. In the isotropic model, the yield point increases during hardening and is equal to the largest stress previously experienced by the rebar.

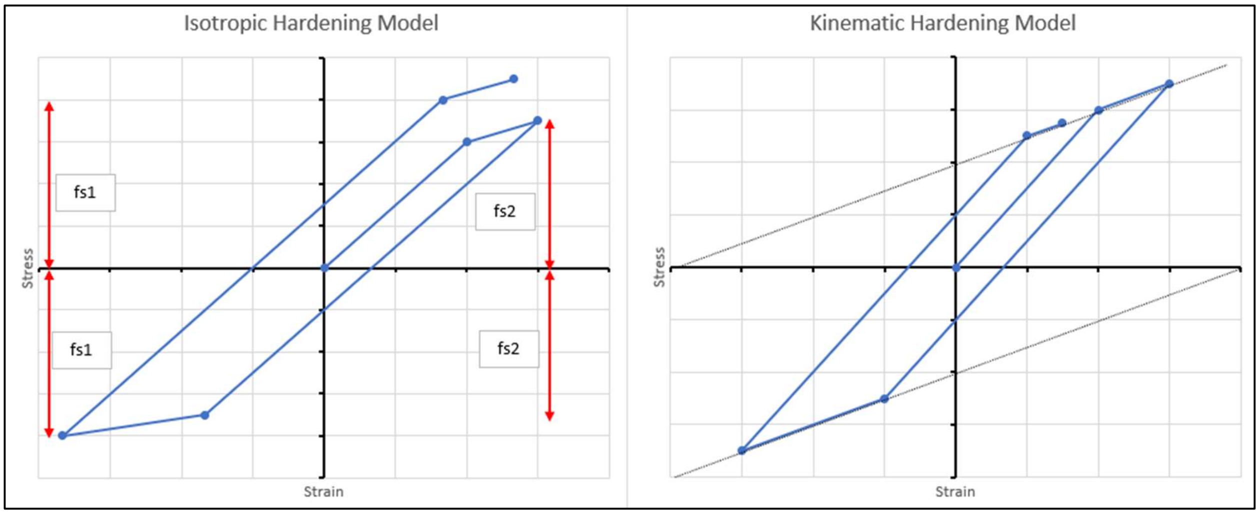


Figure 13 Isotropic and Kinematic Hardening Models Example

For modeling, a method of branches was used to calculate the stress-strain relationship. These branches are for linear tension, yielding in tension, linear compression, and yielding in compression as shown in Figure 14. For a set of strain data from an experiment, the corresponding model stress could be calculated by determining which branch it was currently on and calculate stress using the correct yield points and slope.

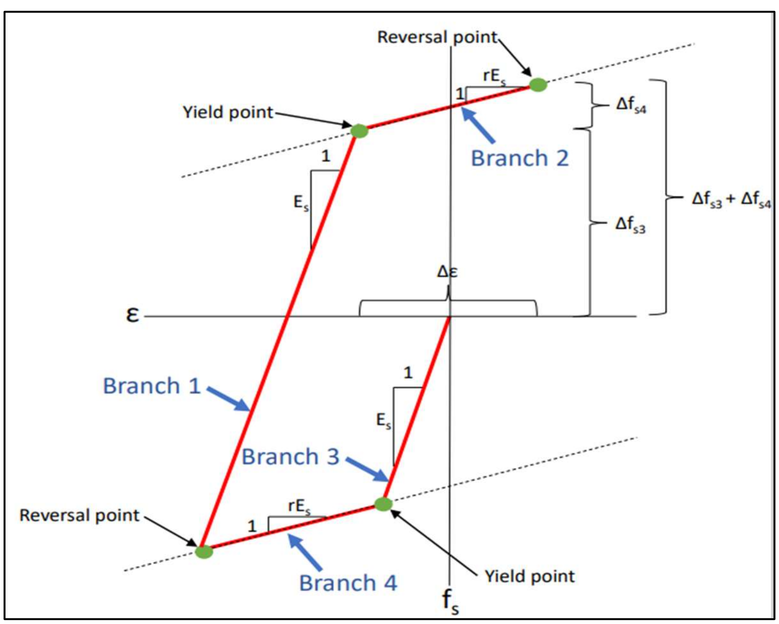


Figure 14 Description of Branches used in Bilinear Model

For example, if the model started on branch 3, the first yield point is defined as the ϵ_y and f_s and the equation of this line has a slope of E_s; using a point slope form, any stress on this line can be found for a corresponding test strain. Once the strain data exceeds these yield points, either for branch 1 or 3, the model switches branches from 3 to 4, and from 1 to 2. Branch 2 and branch 4 are calculated using the yield points from the prior branch and a slope of $r \cdot E_s$ where r is the strain hardening ratio and calculated as previously described. The reversal points are set when there is a change of signs between the difference of two consecutive strain data points. The yield point is then calculated based upon the assumed model. For the kinematic hardening model, the yield point is the reversal point plus $2\epsilon_y$ and $2f_y$. For the isotropic hardening model, the yield point is set to the opposite of the previous stress value $-f_s$ and the "yield" strain is the previous strain plus the strain associated with two times the previous stress $-2f_s/E_s$.

Question 1

The kinematic hardening cyclic model shows reasonable similarity with the experimental data as shown in Figure 15. The major drawback of this model is not properly accounting for softening of the steel as it yields and underestimating the final strain hardening of the curve.

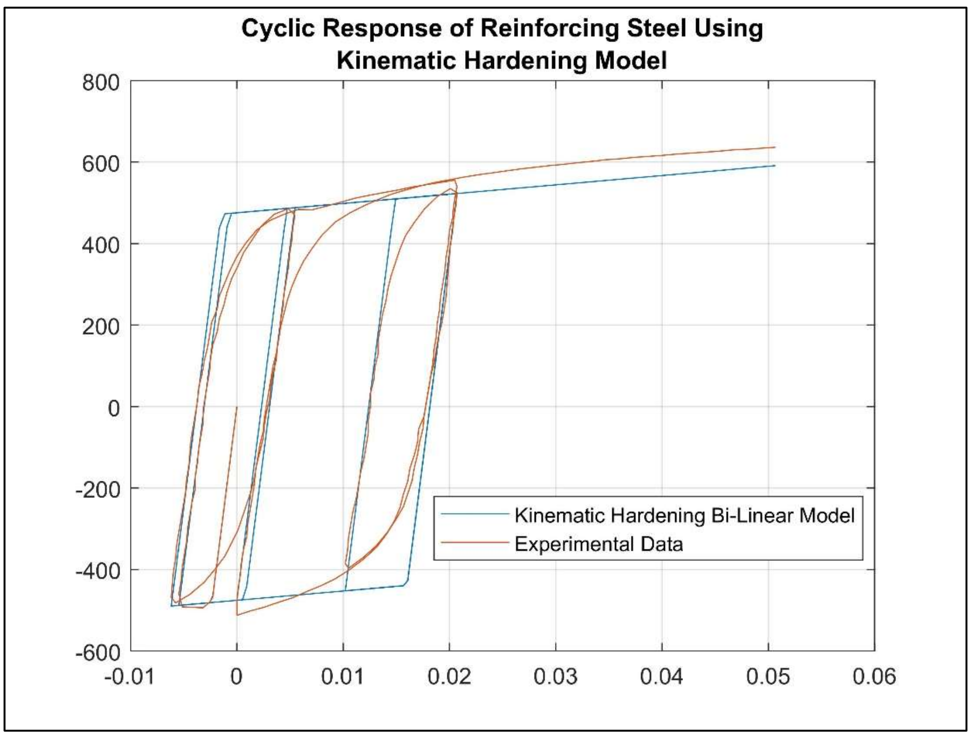


Figure 15 Kinematic Hardening Model of Steel Rebar Under Cyclic Loading

Question 2.

The isotropic model is supposed to capture the behavior of the strain hardening but is quite off from the experimental data as shown in Figure 16. This stems from the asymmetric loading seen in the experimental data. Since the shape of the model assumes that the upper and lower regions of the data are symmetric, the model overshoots some regions, and this error propagates into future cycles of the model.

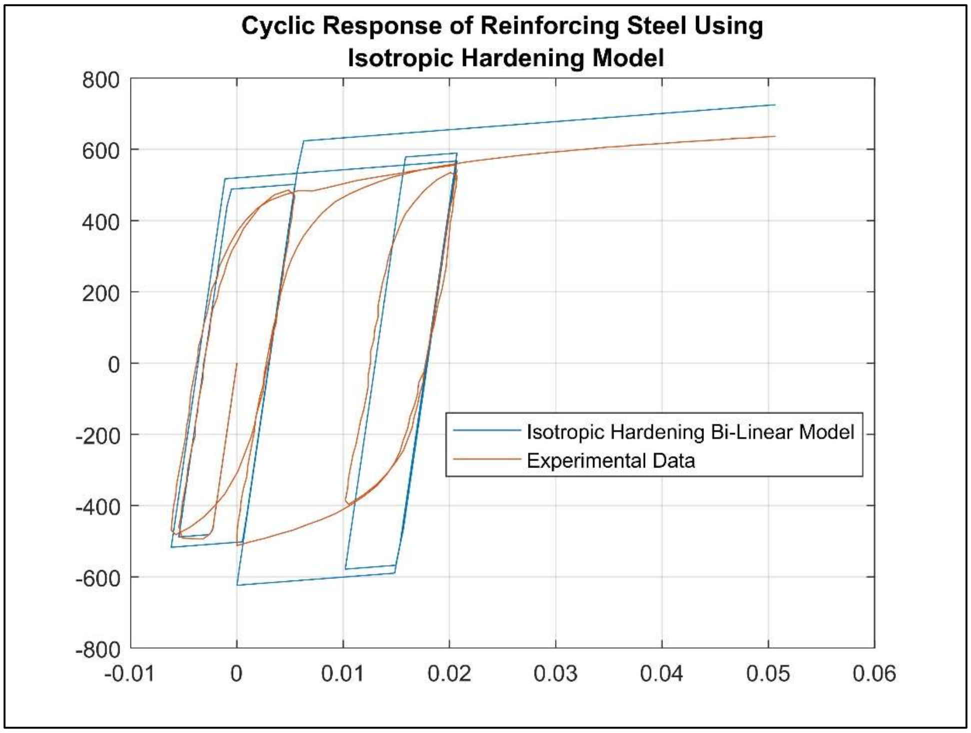


Figure 16 Isotropic Hardening Model of Steel Rebar Under Cyclic Loading

Question 3.

The cyclic stress-strain relationship, under a bilinear model, can be improved by using a linear combination of the isotropic hardening and kinematic hardening models. The formula used to combine the two stress models is given by

$$f_s = \alpha f_{s,kinematic} + (1-\alpha) f_{s,isotropic}$$

In Figure 17, a range of ratios from 0.15, 0.25, 0.3, 0.45, 0.7, 0.75 were used to generate the final model stress values. By inspection, the α values of 0.25 and 0.3 seem to fit the data set well. The softening of the steel is not captured well by any one of the models and continues to be a source of error in the linear combination. Using as similar sum of the squared error between the experimental and model stress values, an optimal ratio of $\alpha = 0.24$ was found, which was close to the value deduced from inspection, Figure 18.

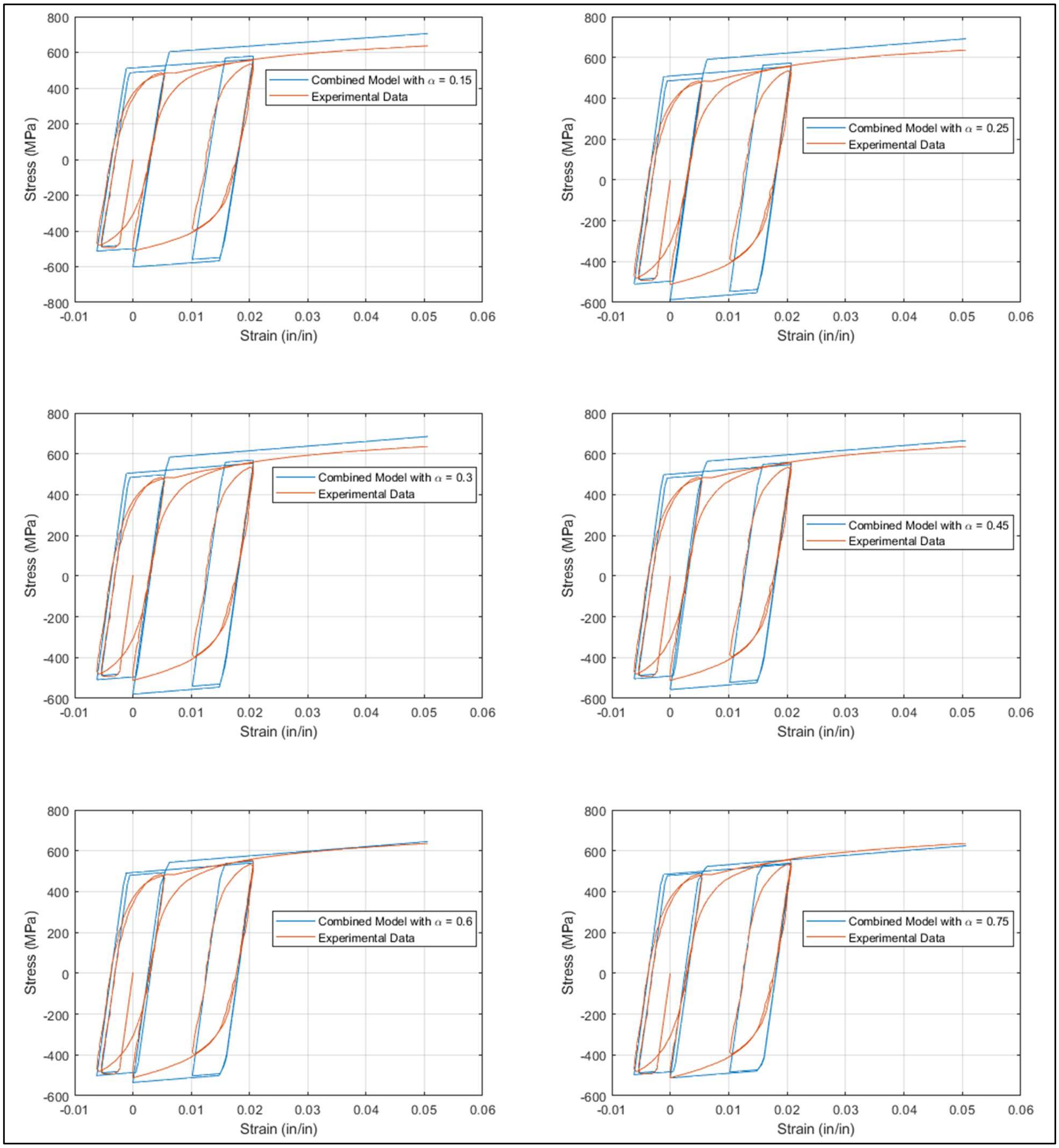


Figure 17 Combinations of Isotropic and Kinematic Hardening Model for Steel Rebar

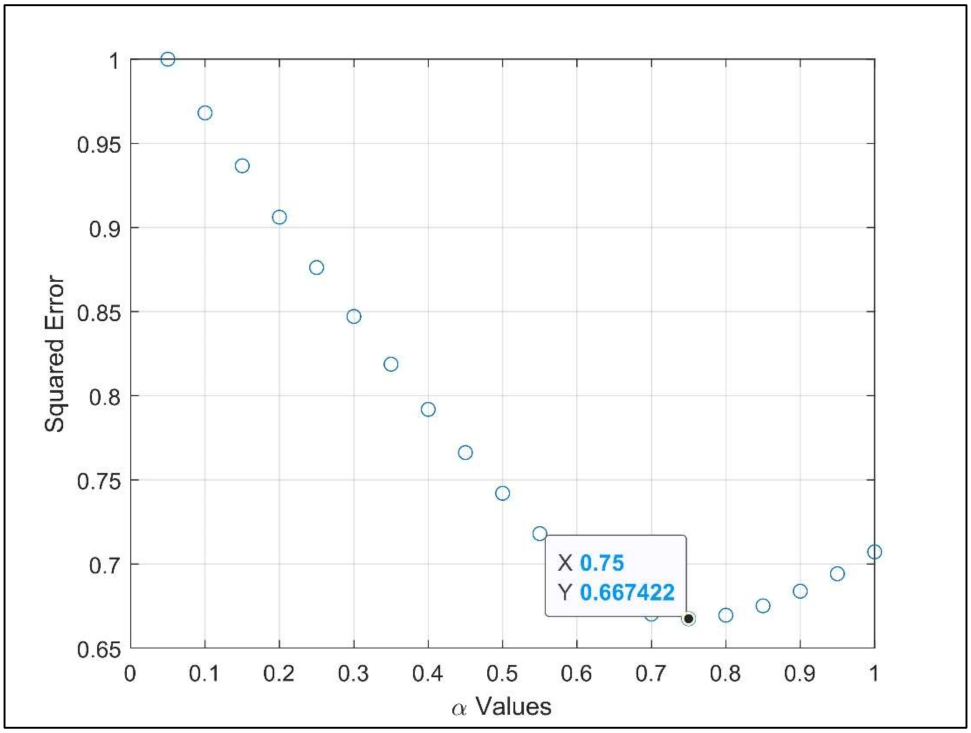


Figure 18 Normalized Error for Various Ratios of Bilinear Model

Conclusion

In this report, the mechanical behavior of reinforcing steel bar used in reinforced concrete is investigated using three separate test data.

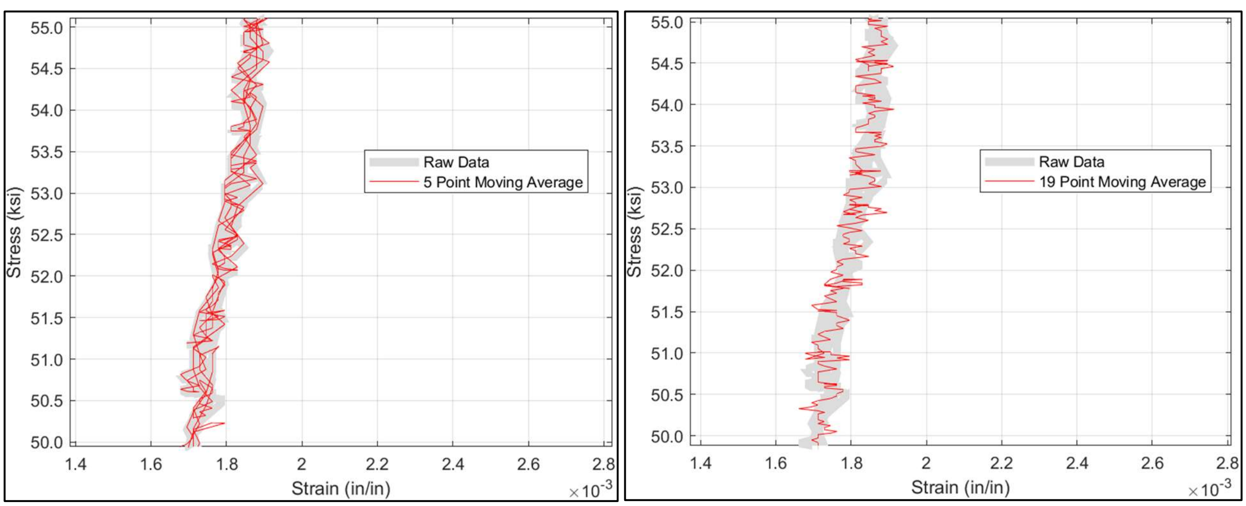


Figure 19 Inspection of the Effects of Filtering Data

In the first region, the material properties of a no. 11 ASTM A706 rebar was found using filtered test data. The filtering of data is important if Figure 19 is examined. The data has many points where it jumps erratically and loops back onto itself which could be caused measurement errors, nonlinear material behavior, or non-consistent load application. Even with the 19-point moving average, the stress-strain history of a monotonically loaded rebar still shows variation in a supposed linear region.

In general, the stress-strain curve of a steel reinforcing bar is broken into three regions that consist of an elastic region, a yield plateau, and a strain hardening region. These regions are bounded by a stress-strain pairs that characterizes the shape of the region; these points being $(0,0)$ and (ϵ_y, f_y) for the elastic region, (ϵ_y, f_y) and (ϵ_{sh}, f_{sh}) for the yield plateau, and (ϵ_{sh}, f_{sh}) and (ϵ_{su}, f_{su}) for the strain hardening region. These boundary points are not specific for all steel rebar as there will be variations in the experimental data, due to measurement error or nonlinear behavior of steel in some regions. Thus, to characterize a rebar's stress-strain curve, a bound of acceptable stress-strain pairs can actually be given. Low-alloy, steel bars used for concrete reinforcement should have a yield strength in the range of 60-78 ksi shown in Figure 20 (Table 2, ASTM A706). In the test, the yield stress was found to be 74 ksi which is in the acceptable range but shows that rebars tend to have strengths greater than the minimum for manufacturing reasons: to avoid providing understrength rebar.

TABLE 2 Tensile Requirements

	Grade 60 [420]	Grade 80 [550]
Tensile strength, min, psi [MPa]	80 000 [550][A]	100 000 [690][A]
Yield strength, min, psi [MPa]	60 000 [420]	80 000 [550]
Yield strength, max, psi [MPa]	78 000 [540]	98 000 [675]
Elongation in 8 in. [200 mm], min, %		
Bar Designation Nos.		
3, 4, 5, 6 [10, 13, 16, 19]	14	12
7, 8, 9, 10, 11 [22, 25, 29, 32, 36]	12	12
14, 18 [43, 57]	10	10

[A] Tensile strength shall not be less than 1.25 times the actual yield strength.

Figure 20 ASTM A706 Table 2

There are different models that can be used to describe the monotonic loading of a steel rebar. The two explored in this report was the trilinear model and the Mander's model. A quantitative comparison between the two models is unnecessary as the trilinear model is too simple compared with the Mander's model. It is obvious that there are more errors in the trilinear model since it ignores the yield plateau and approximates the strain hardening region as a linear line. The Mander model accounts for the elastic region and the yield plateau as well as fit a polynomial function to the strain hardening region. From the report, a cubic function represents this region the best, having the lowest sum of difference between model and experimental data. This power for the strain-hardening region can vary for other experimental data.

The second portion investigates the relationship between strain in a steel rebar when it is in compression and tension. For small strain <2%, the stress-strain curve for a compression and tension loaded rebar are similar. One way to measure the similarity between two curves would be to calculate the Fréchet distance between the two data sets. Using a discrete Fréchet distance algorithm, provided online by Zachary Danziger, the Fréchet distance for the engineering stress-strain values for the strain range [0 0.025] was 17.4, which shows a close correlation between the stress-strain values. Over the entire range of the engineering stress-strain curve, the Fréchet distance was 143.0 while for the true coordinate system stress-strain values, the Fréchet distance was 16.8. The stress-strain relationship in the true coordinate system for compression and tension for a rebar are more similar than if they were in the engineering coordinate system using a Fréchet distance measurement.

The last portion of this report investigates models used to describe the stress-strain curve for steel rebar under cyclic loading. There were two cyclic model used, a kinematic hardening model and an isotropic hardening model. For both models, the strain hardening is not well captured due to the linear form of the functions describing the model. A linear combination of the 75% of the kinematic hardening model and 25% isotropic hardening model yielded the optimal result based on the sum of the squared difference between the experimental data and the model.

References

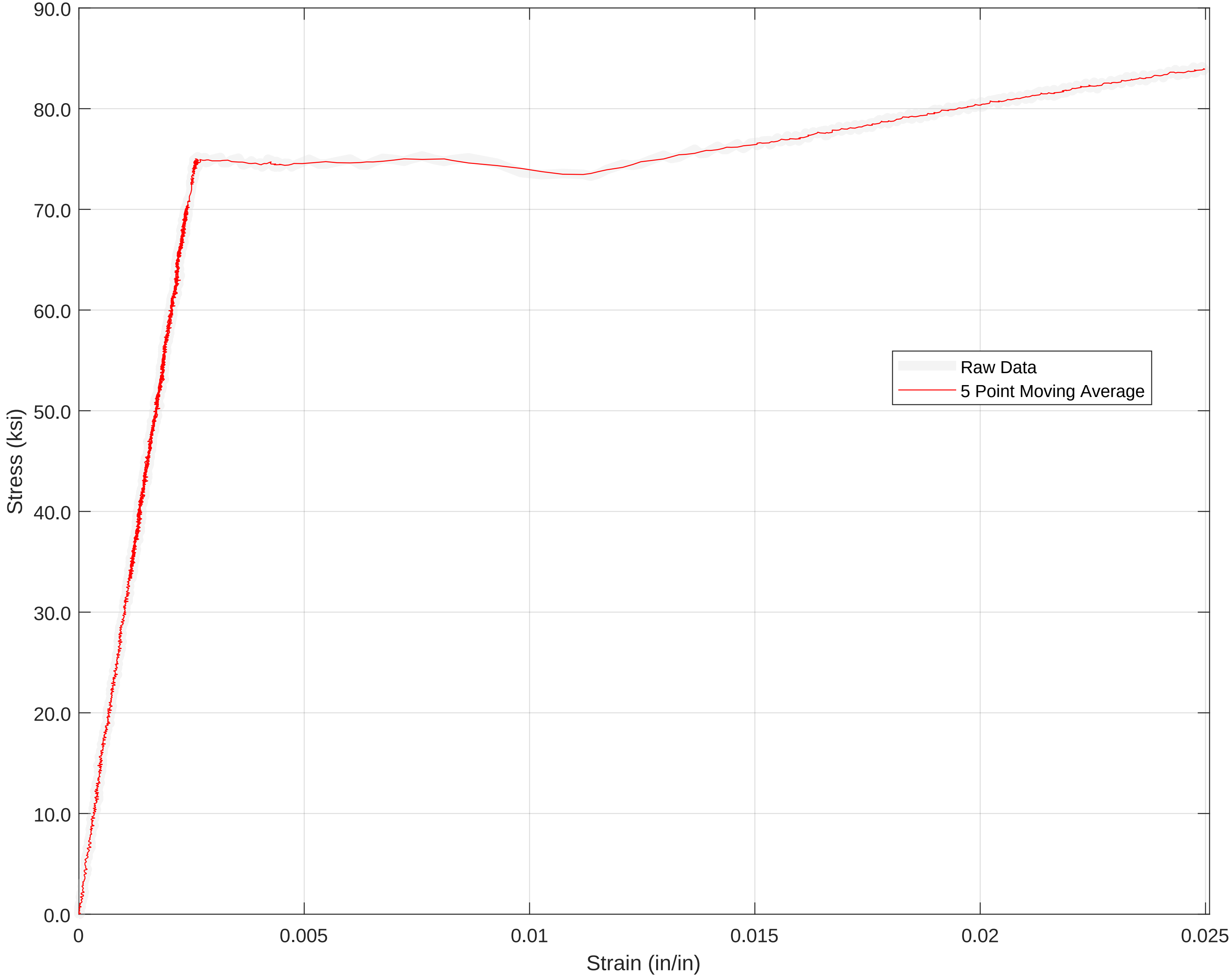
ASTM International. *A706/A706M-16 Standard Specification for Deformed and Plain Low-Alloy Steel Bars for Concrete Reinforcement*. West Conshohocken, PA; ASTM International, 2016. doi: https://doi.org/10.1520/A0706_A0706M-16

Dodd, L. L., and J. I. Restrepo-Posada. "Model For Predicting Cyclic Behavior Of Reinforcing Steel". *Journal Of Structural Engineering*, vol 121, no. 3, 1995, pp. 433-445. *American Society Of Civil Engineers (ASCE)*, doi:10.1061/(asce)0733-9445(1995)121:3(433).

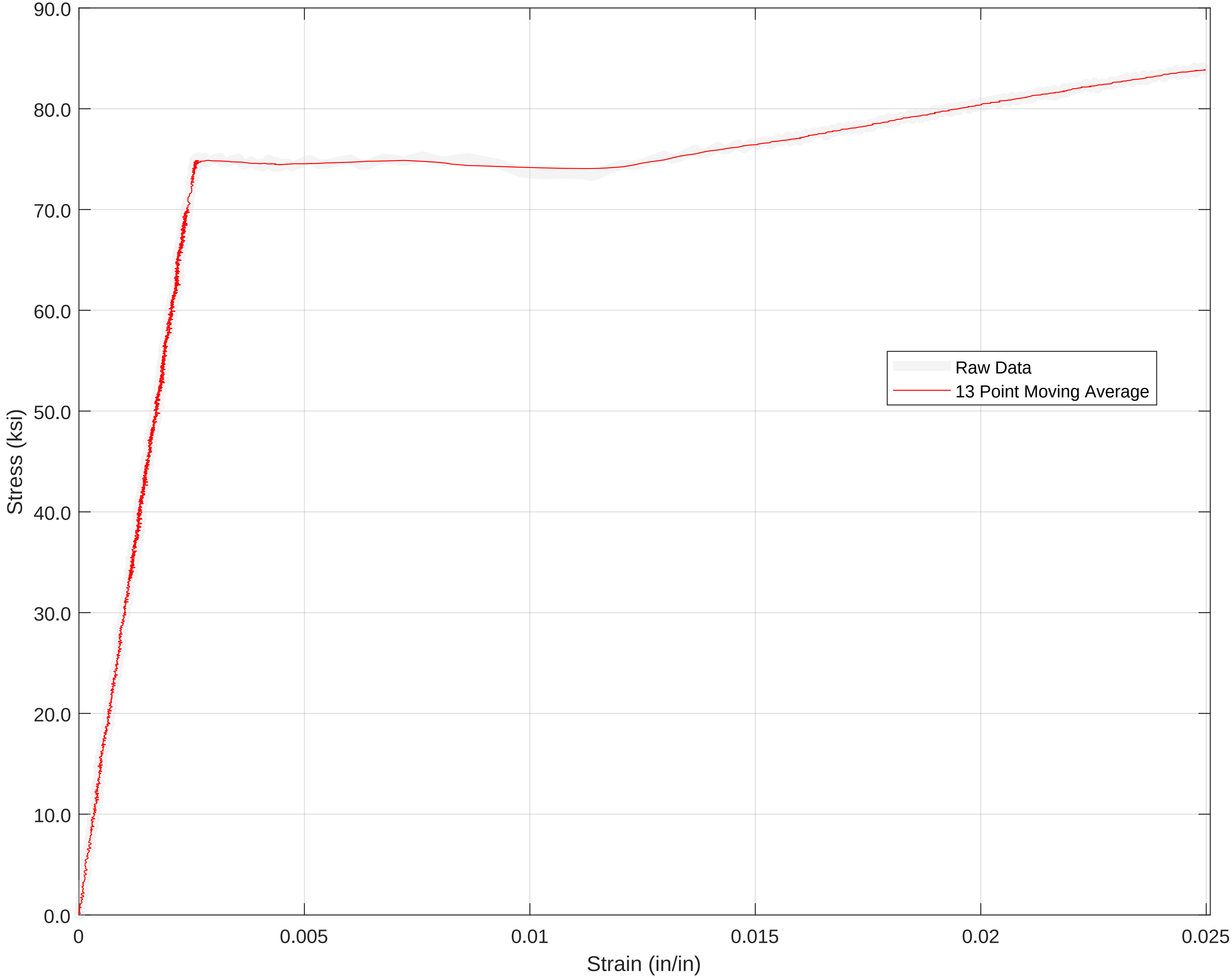
Stang, A.H. et al. "Poisson's Ratio Of Some Structural Alloys For Large Strains". *Journal Of Research Of The National Bureau Of Standards*, vol 37, no. 4, 1946, p. 211. *National Institute Of Standards And Technology (NIST)*, doi:10.6028/jres.037.012.

Zachary Danziger (2021). Discrete Frechet Distance (https://www.mathworks.com/matlabcentral/fileexchange/31922-discrete-frechet-distance), MATLAB Central File Exchange. Retrieved January 11, 2021.

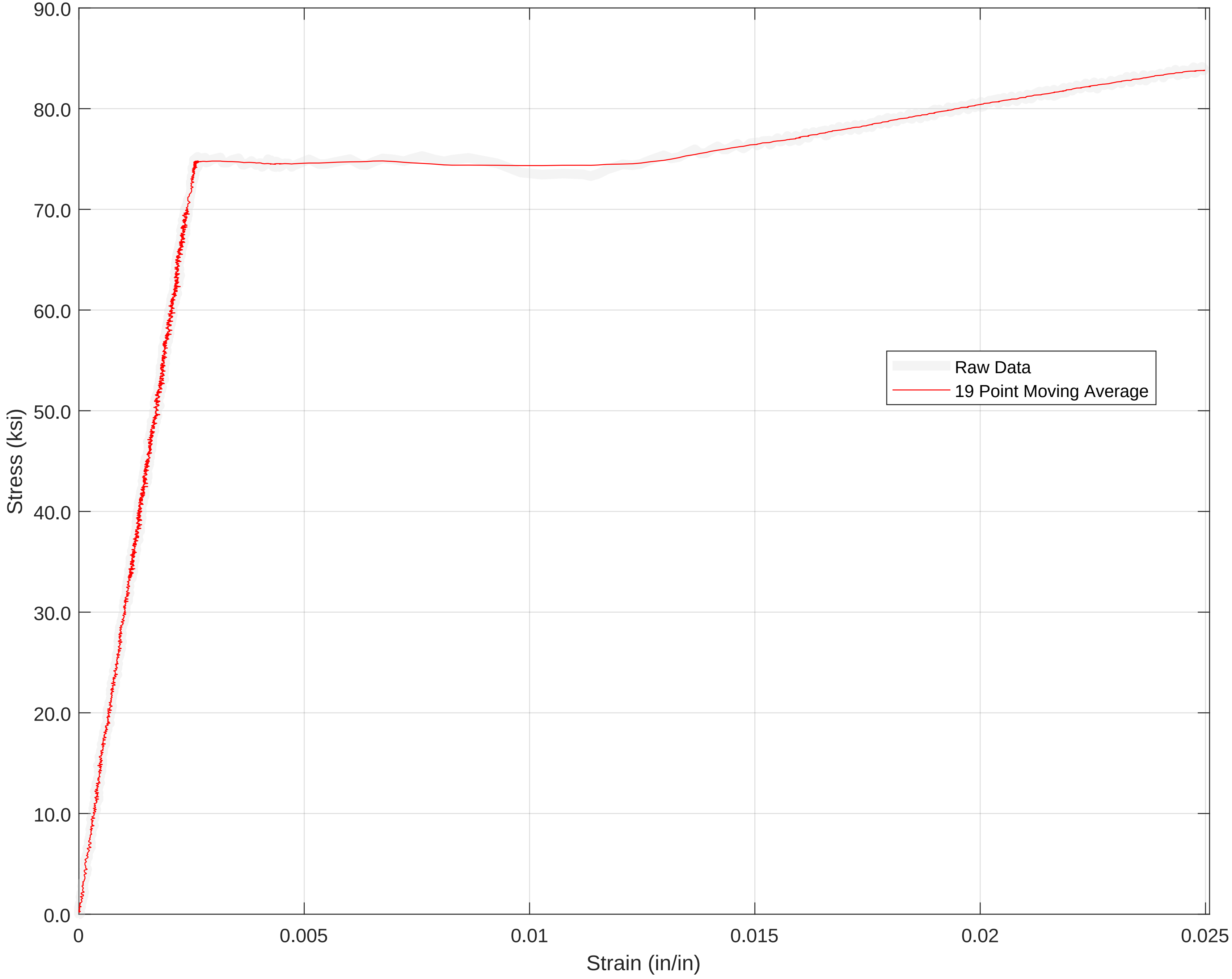
Appendix A. Figures



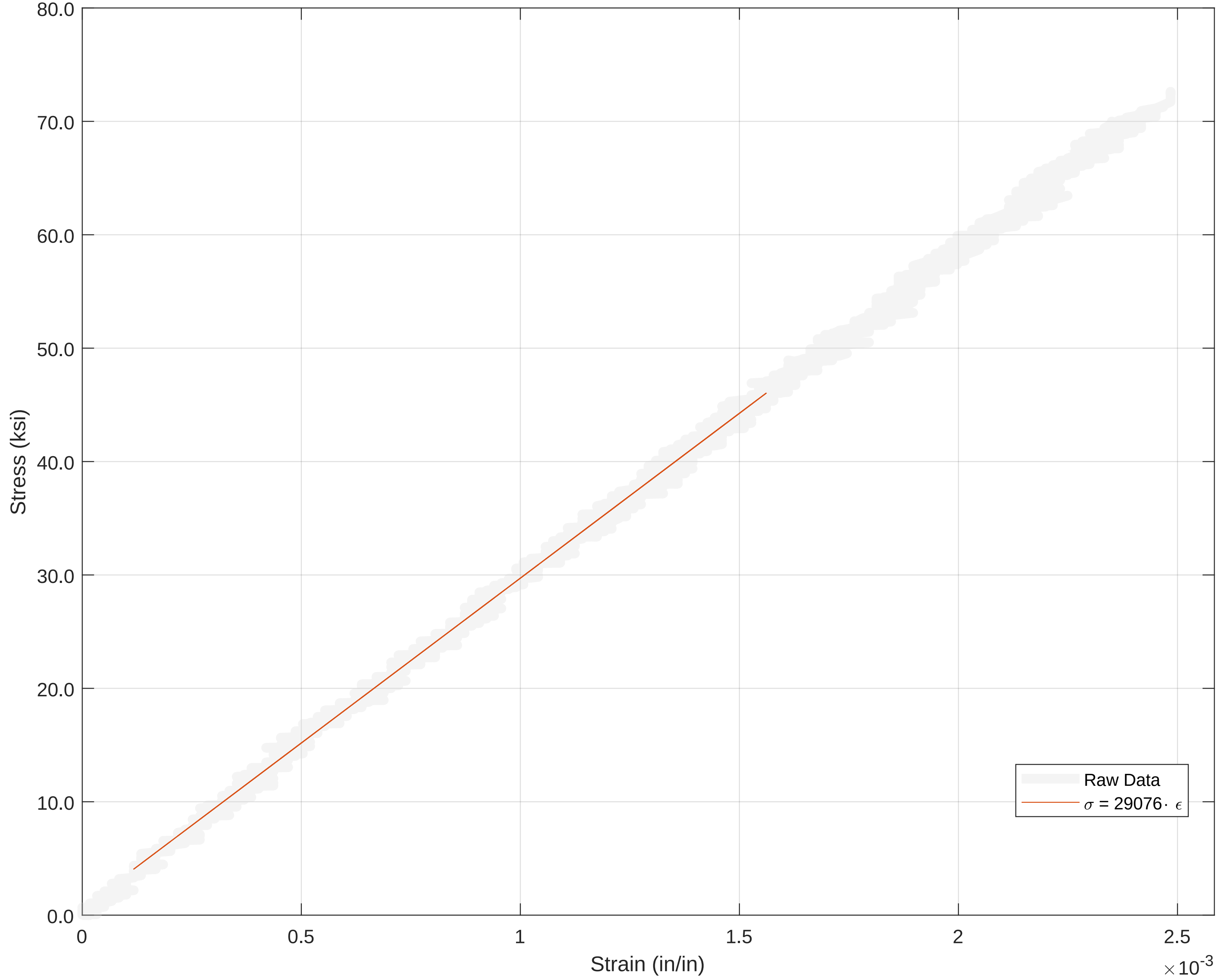
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
Stress (ksi)
0
0.005
0.01
0.015
0.02
0.025
Strain (in/in)
Raw Data
5 Point Moving Average



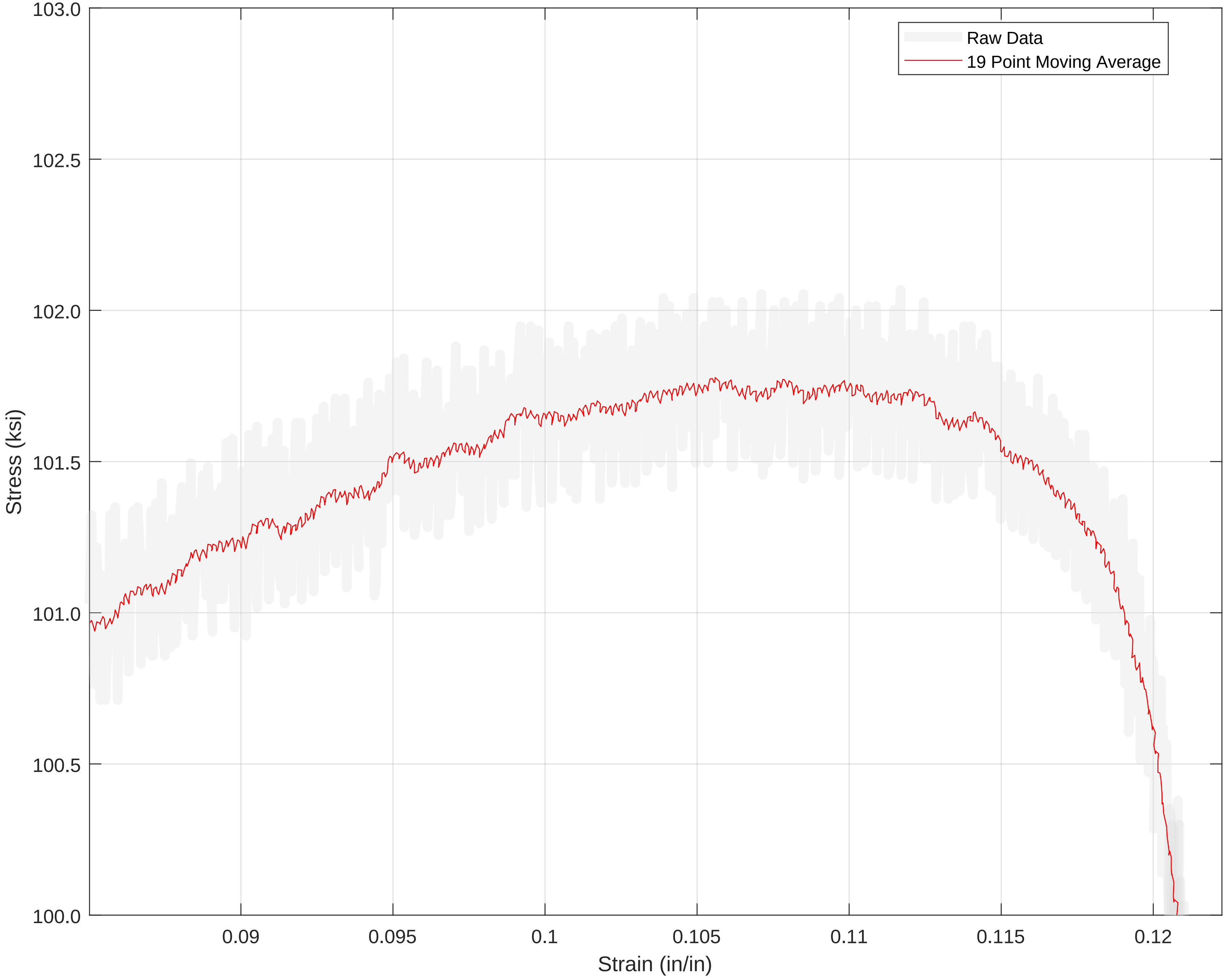
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
Stress (ksi)
0
0.005
0.01
0.015
0.02
0.025
Strain (in/in)
Raw Data
13 Point Moving Average



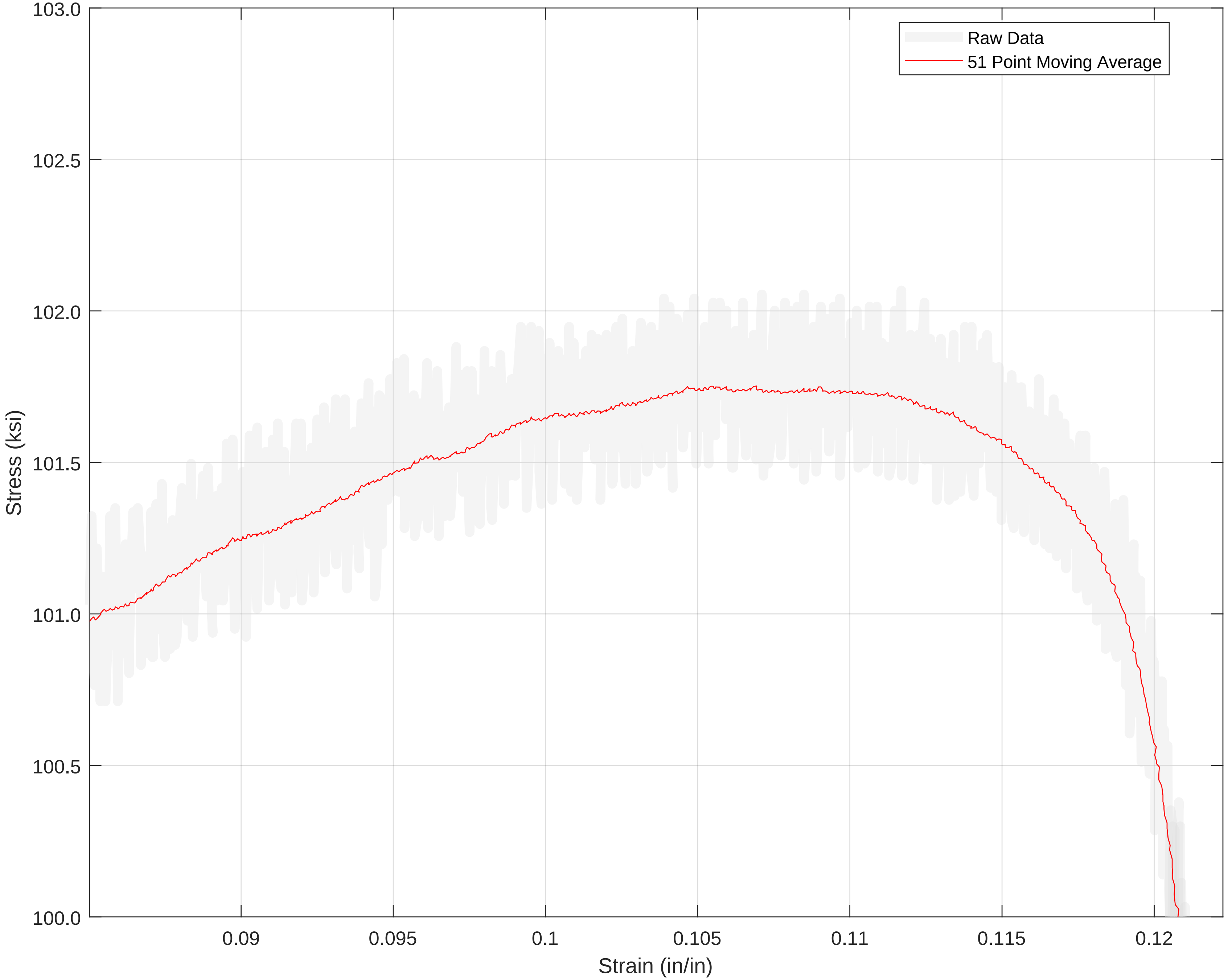
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
Stress (ksi)
0
0.005
0.01
0.015
0.02
0.025
Strain (in/in)
Raw Data
19 Point Moving Average

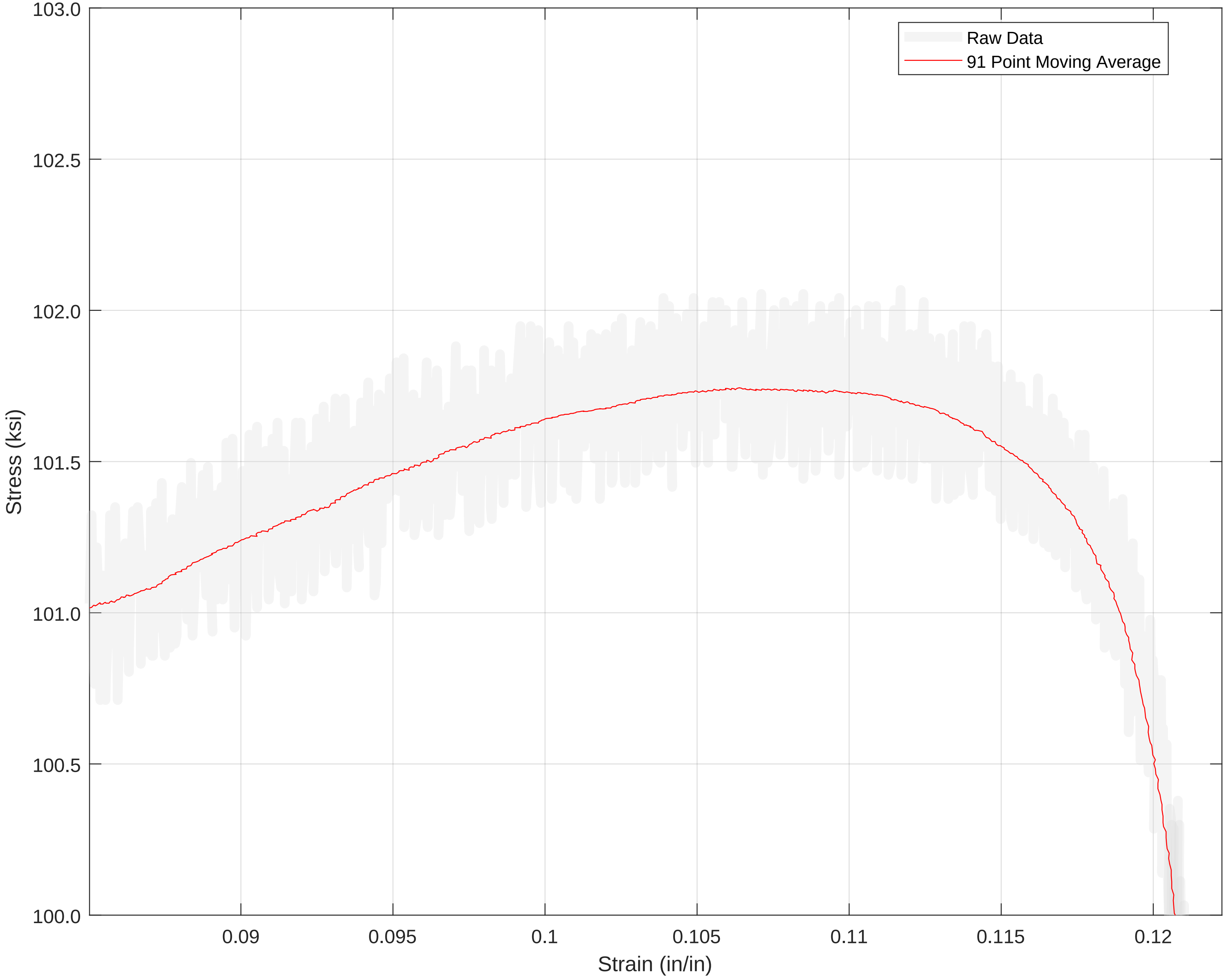


80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
Stress (ksi)
0
0.5
1
1.5
2
2.5
Strain (in/in)
×10^-3
Raw Data
σ = 29076· ϵ

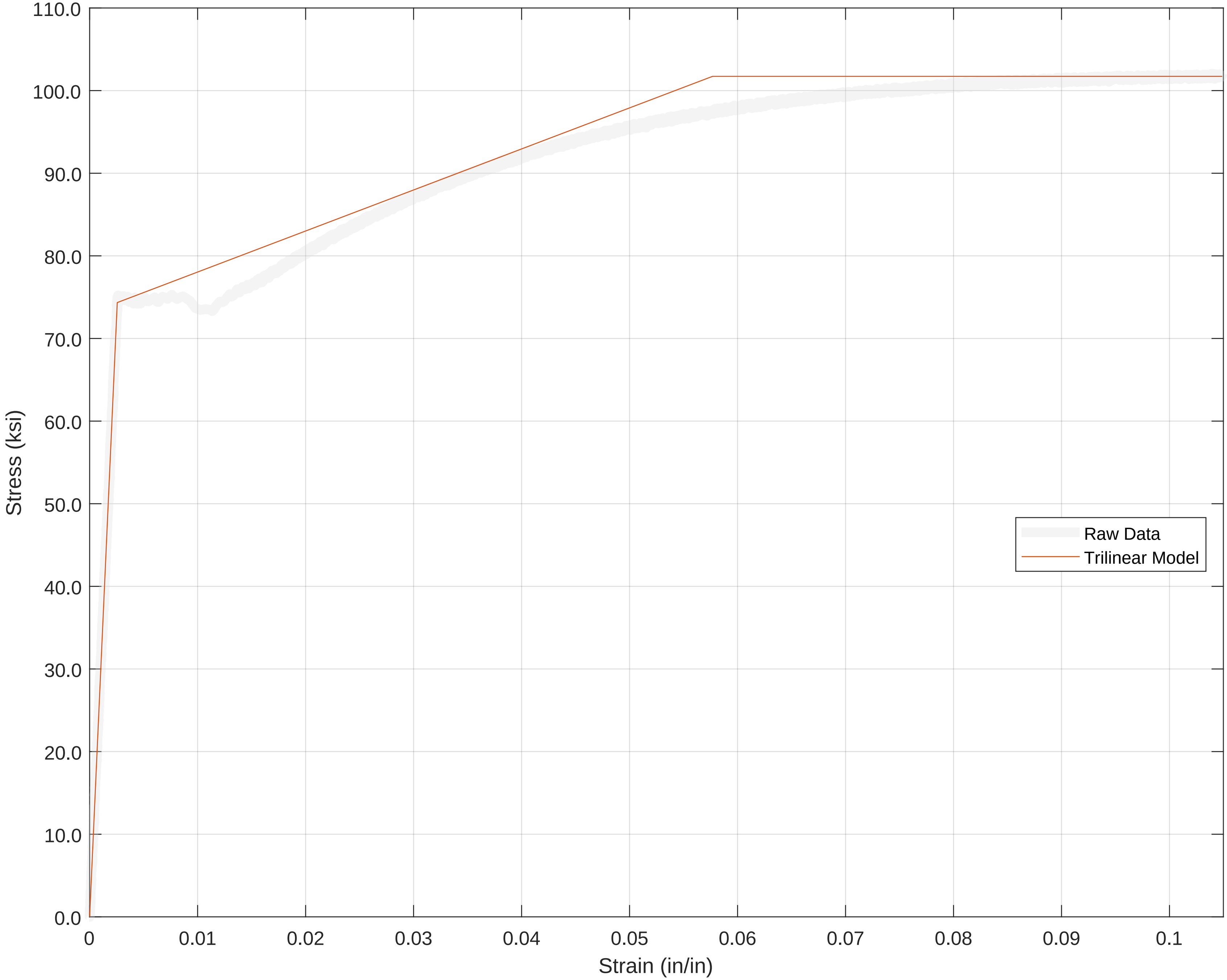


103.0
102.5
102.0
101.5
101.0
100.5
100.0
Stress (ksi)
0.09
0.095
0.1
0.105
0.11
0.115
0.12
Strain (in/in)
Raw Data
19 Point Moving Average

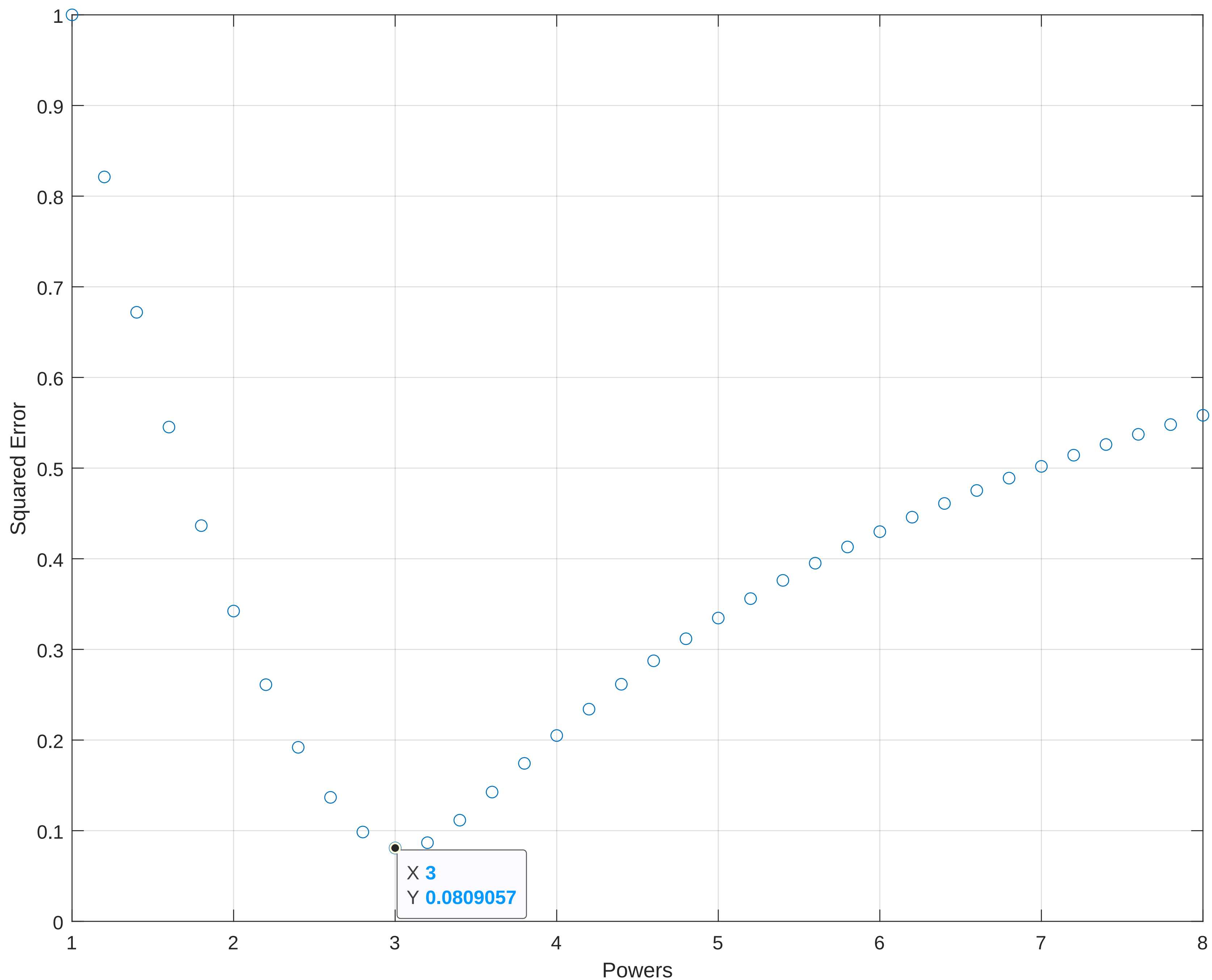

Raw Data
51 Point Moving Average
103.0
102.5
102.0
101.5
101.0
100.5
100.0
Stress (ksi)
0.09
0.095
0.1
0.105
0.11
0.115
0.12
Strain (in/in)



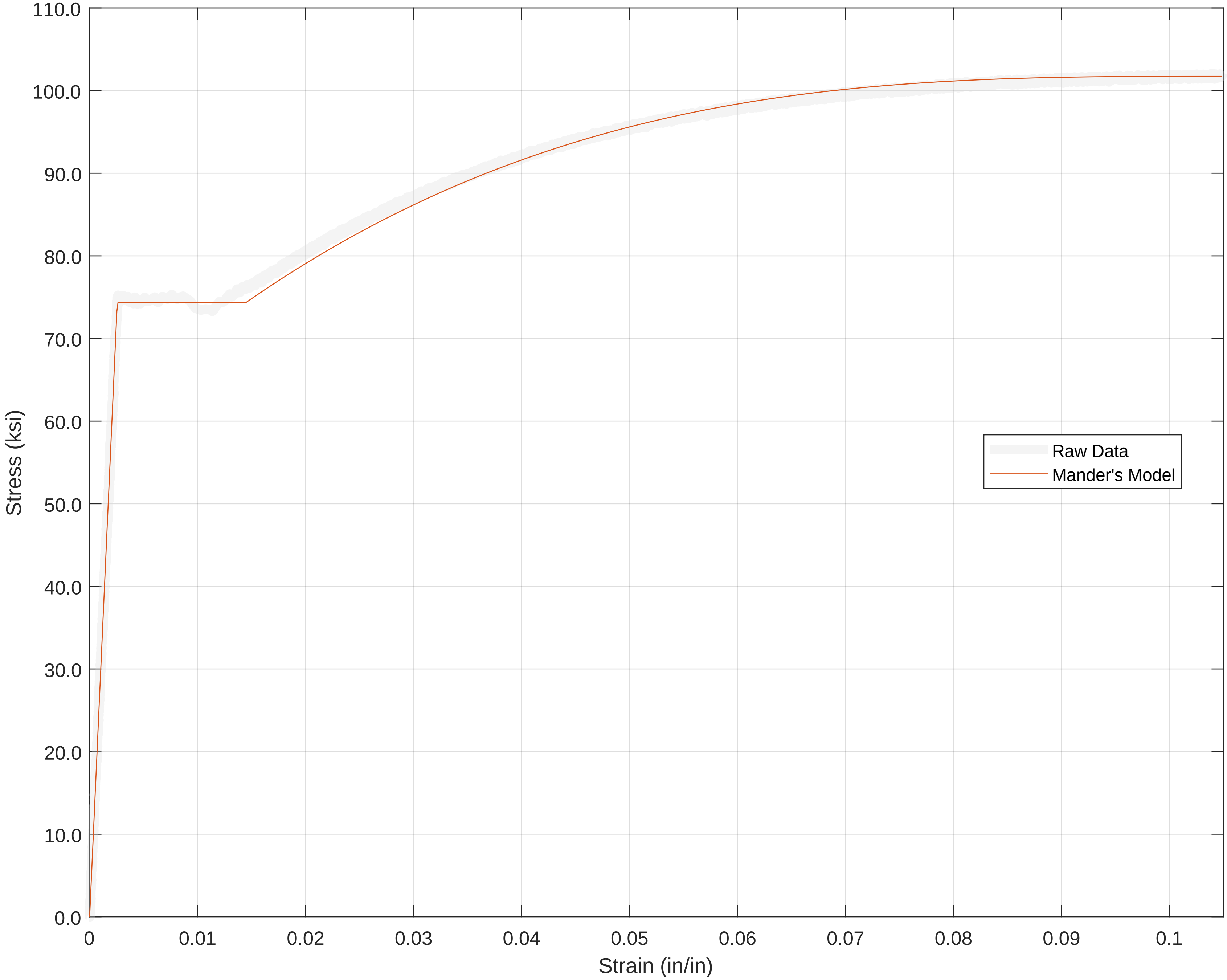
103.0
102.5
102.0
101.5
101.0
100.5
100.0
Stress (ksi)
0.09
0.095
0.1
0.105
0.11
0.115
0.12
Strain (in/in)
Raw Data
91 Point Moving Average



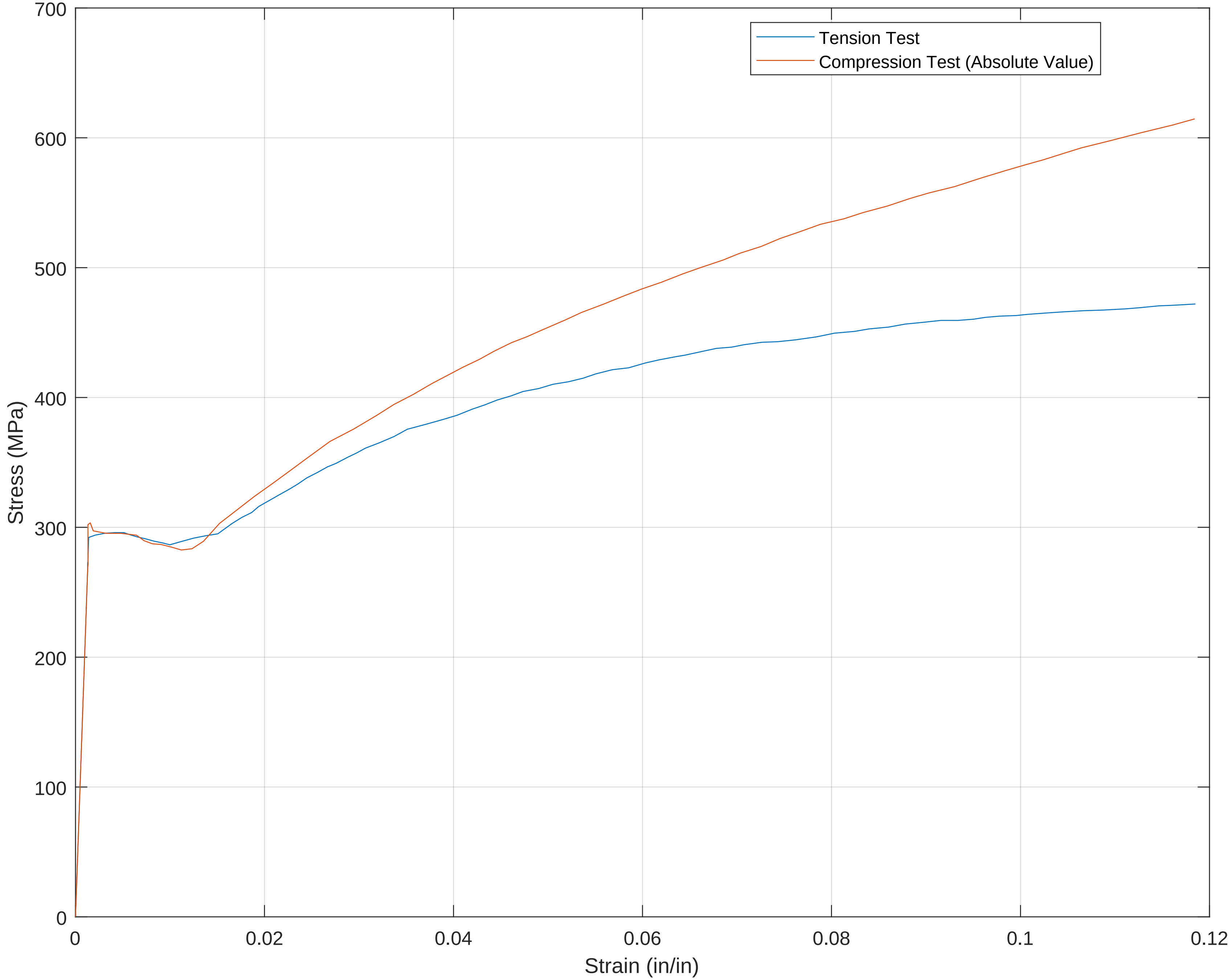
110.0
100.0
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
Stress (ksi)
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
Strain (in/in)
Raw Data
Trilinear Model



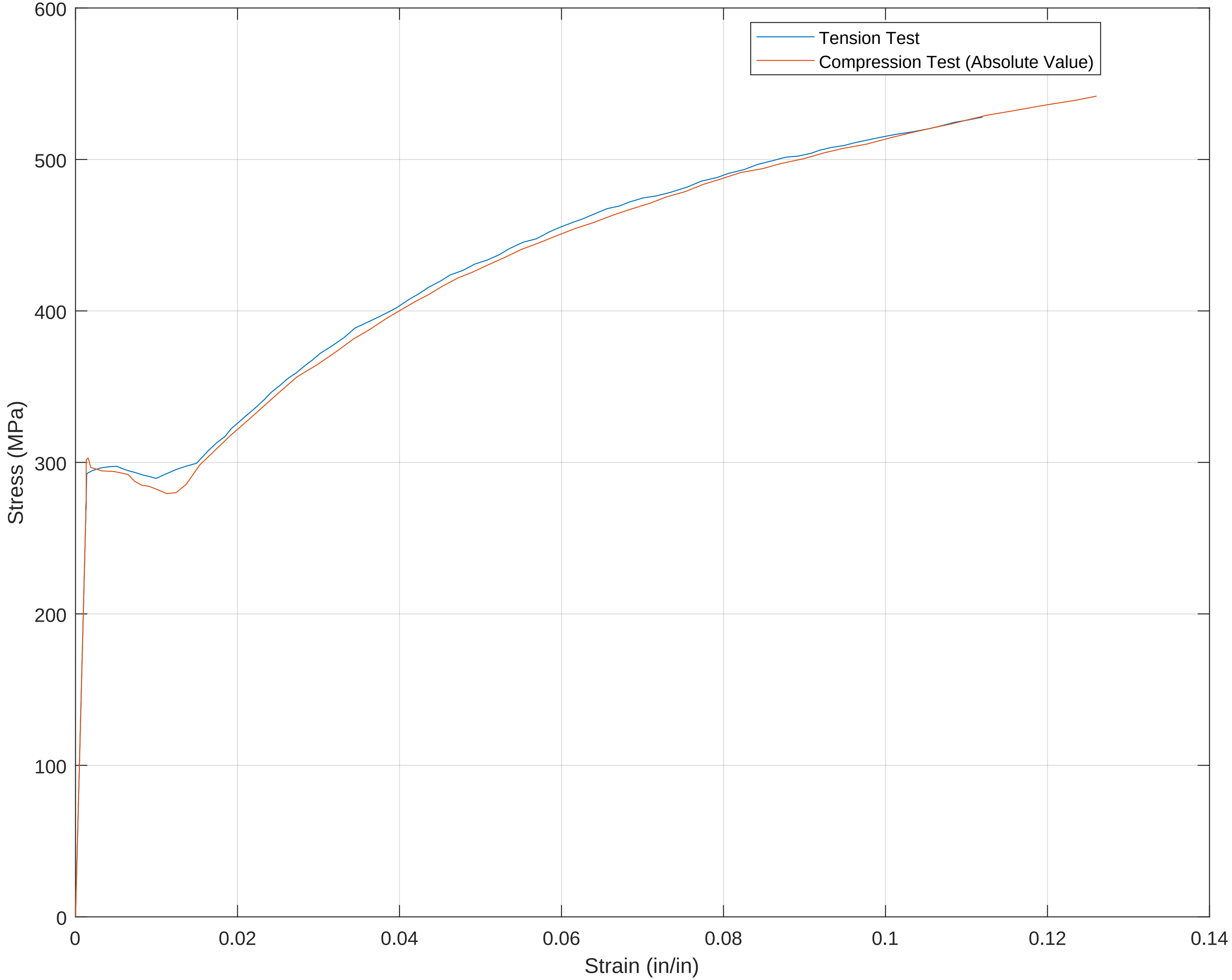
Squared Error
Powers
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1
2
3
4
5
6
7
8
X 3
Y 0.0809057



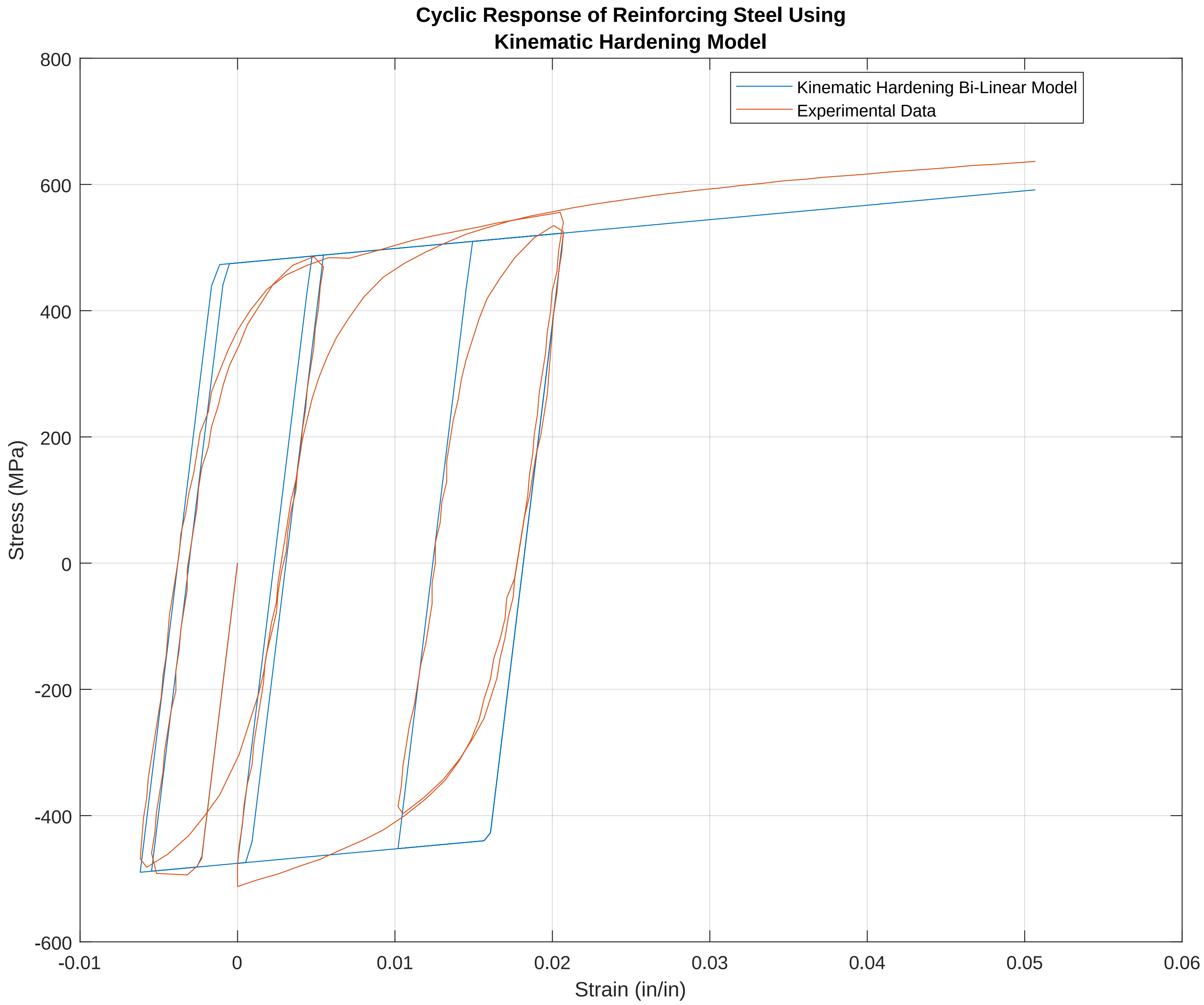
110.0
100.0
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
Stress (ksi)
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
Strain (in/in)
Raw Data
Mander's Model

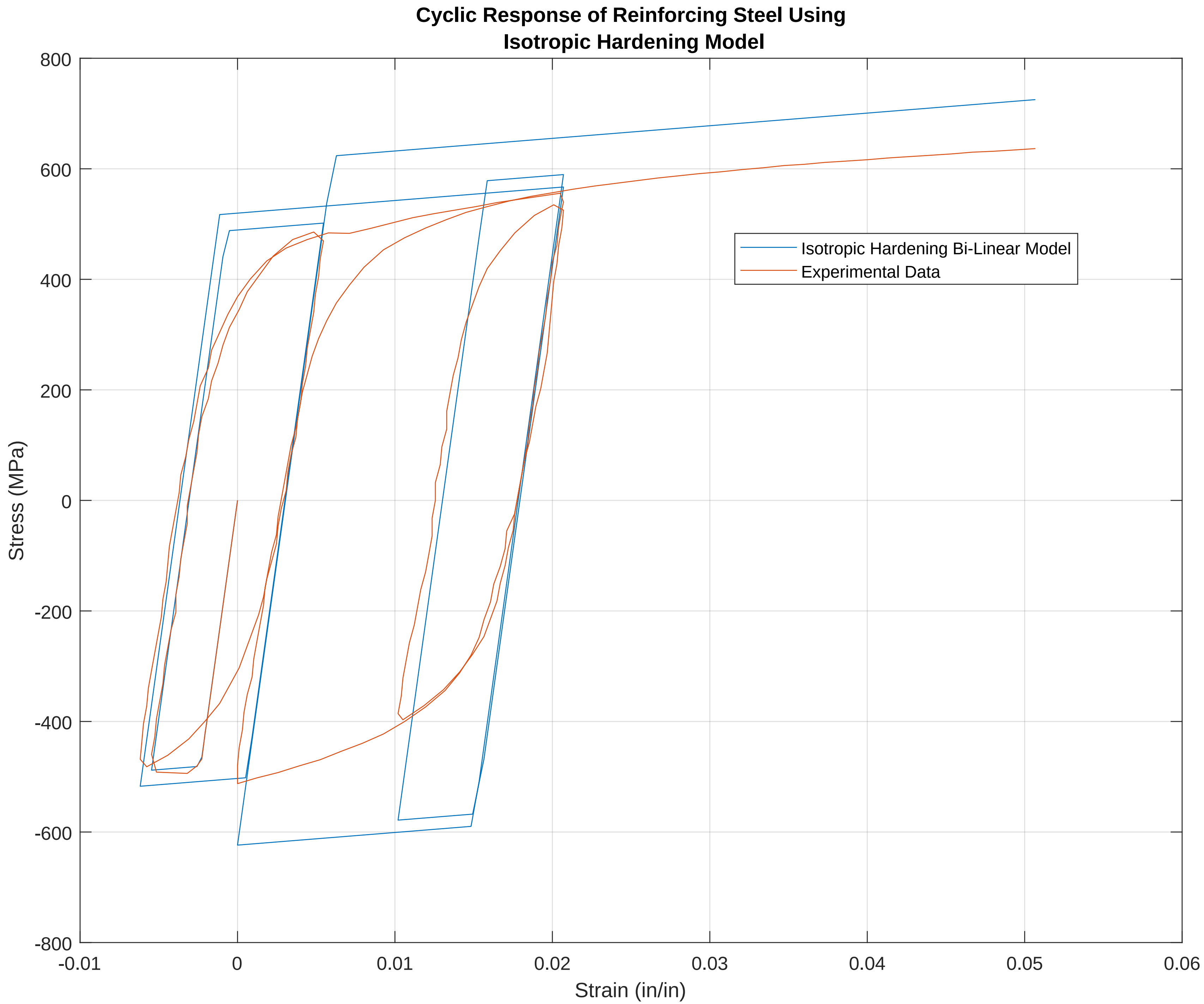


Tension Test
Compression Test (Absolute Value)
Stress (MPa)
Strain (in/in)
0
100
200
300
400
500
600
700
0
0.02
0.04
0.06
0.08
0.1
0.12

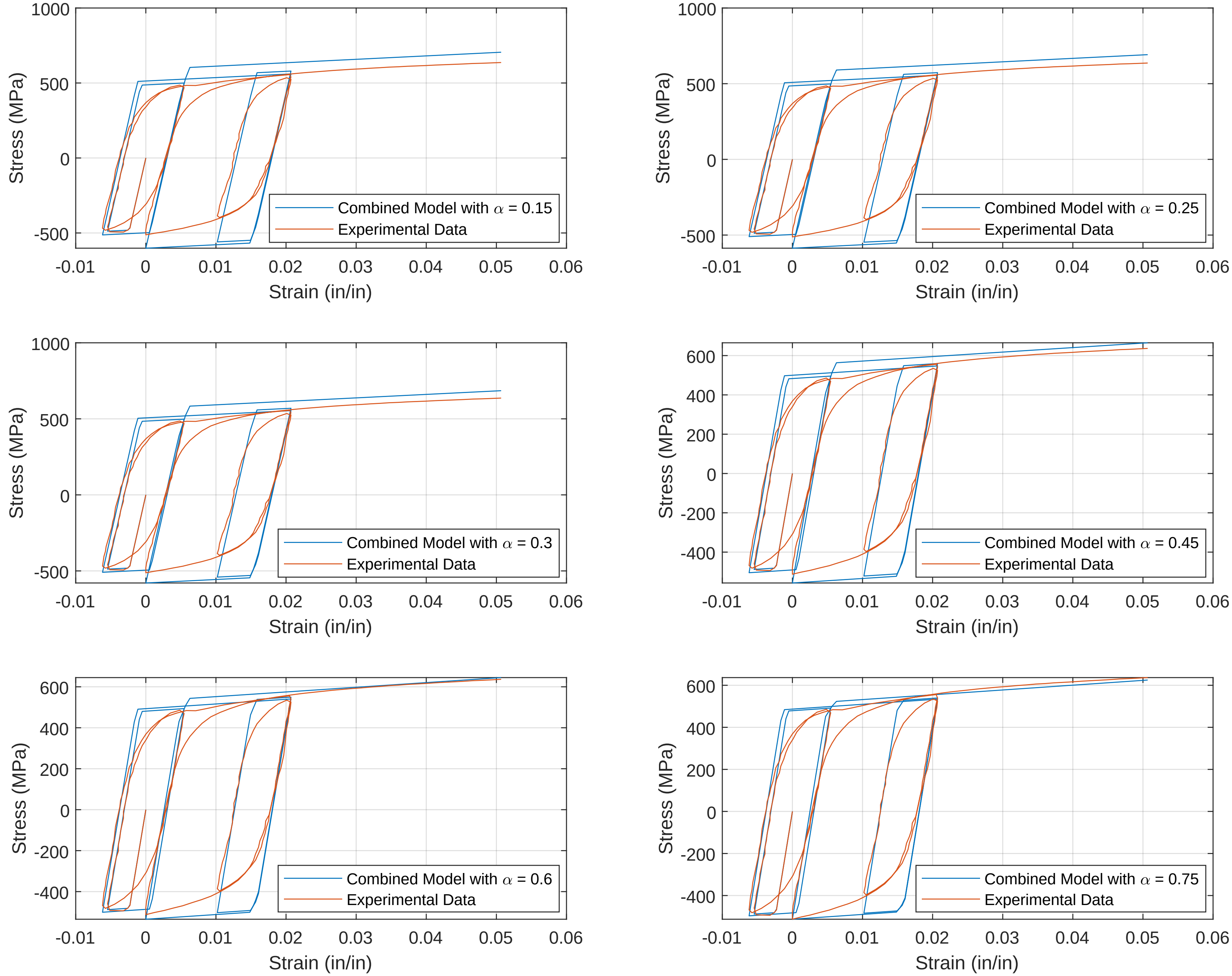


Tension Test
Compression Test (Absolute Value)
Stress (MPa)
Strain (in/in)
600
500
400
300
200
100
0
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14


Cyclic Response of Reinforcing Steel Using
Kinematic Hardening Model
Kinematic Hardening Bi-Linear Model
Experimental Data
800
600
400
200
0
-200
-400
-600
Stress (MPa)
-0.01
0
0.01
0.02
0.03
0.04
0.05
0.06
Strain (in/in)



Cyclic Response of Reinforcing Steel Using
Isotropic Hardening Model
800
600
400
200
0
-200
-400
-600
-800
Stress (MPa)
-0.01
0
0.01
0.02
0.03
0.04
0.05
0.06
Strain (in/in)
Isotropic Hardening Bi-Linear Model
Experimental Data


Stress (MPa)
Strain (in/in)
Combined Model with α = 0.15
Experimental Data
Stress (MPa)
Strain (in/in)
Combined Model with α = 0.25
Experimental Data
Stress (MPa)
Strain (in/in)
Combined Model with α = 0.3
Experimental Data
Stress (MPa)
Strain (in/in)
Combined Model with α = 0.45
Experimental Data
Stress (MPa)
Strain (in/in)
Combined Model with α = 0.6
Experimental Data
Stress (MPa)
Strain (in/in)
Combined Model with α = 0.75
Experimental Data

Appendix B. MATLAB Code

Table of Contents

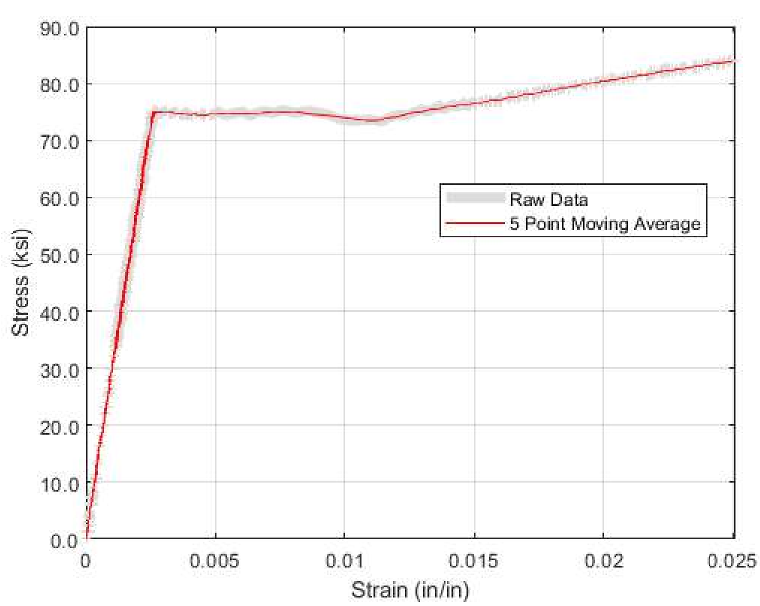
Part(I) Modeling Tensile Stress-Strain Response of Mild Steel Reinforcement

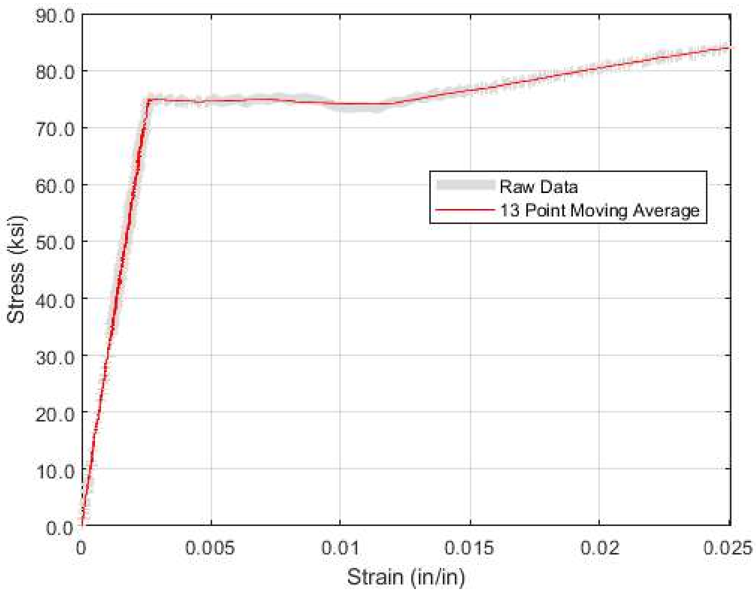
Question 1. Use the data and determine the Young’s modulus, the yield strength, the onset of strain hardening, the tensile strength, and the uniform strain

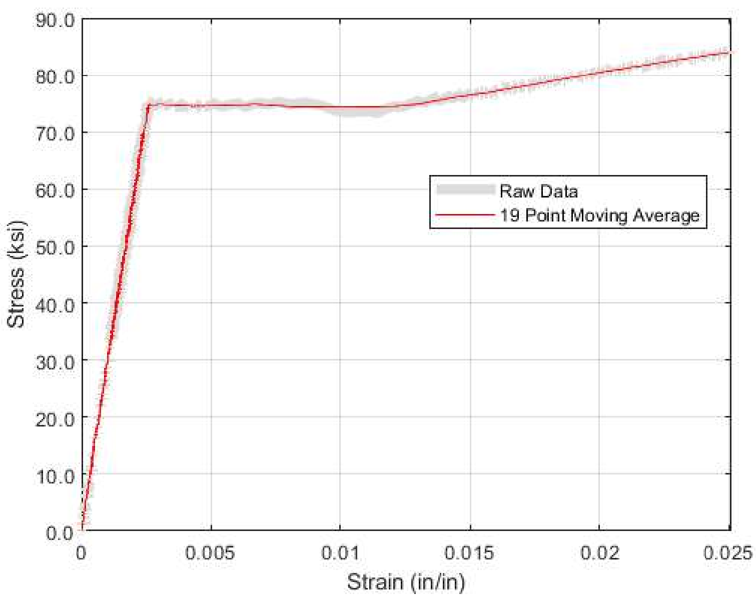
Using the data from the excel sheet the young's Young’s modulus E_s, the onset of strain hardening ϵ_{sh}, the tensile strength f_y, and unform strain ϵ_u.

Part (i) Determining Yeidling Stress and Onset of Strain Hardening

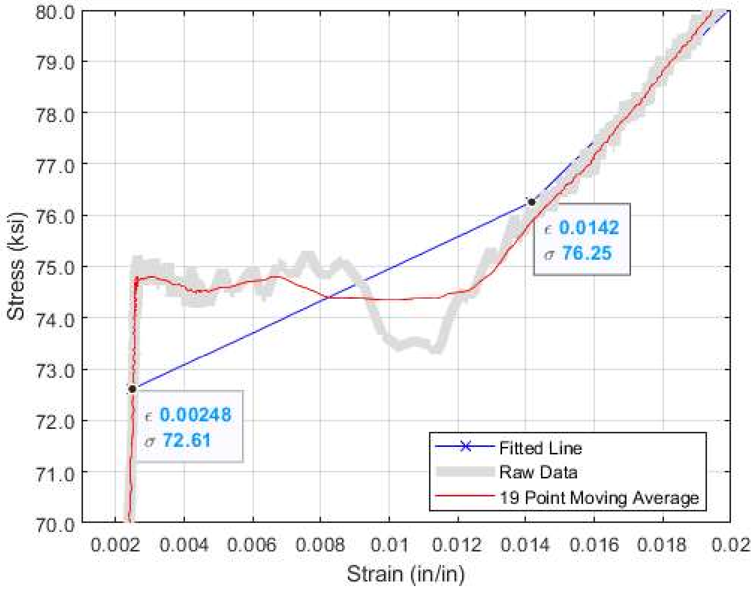
```
close all;
strain_range = [0, 0.025]; % As specified
[stress, strain] = get_data_for_range(strain_range); % Gets the data from the range

avg_values = [5, 13, 19]; % The number of samples that need to be taken for the moving average
for i = 1:length(avg_values) % Loops through the avg values
    figure(i); % Set up Figure
    mov_pt_avg = avg_values(i); % Sets the sampling size
    plot_raw_data(strain, stress) % Plots raw data
    int2str(mov_pt_avg) + " Point Moving Average"; % Title of plot, just set as a temp variable
    plot(strain, movmean(stress,mov_pt_avg),'r','DisplayName',ans); % Filtered Data
    print_figure(i) % Saves the figure as a pdf
end
```








```
strain_range = [0, 0.13]; % As specified
[stress, strain] = get_data_for_range(strain_range); % Gets the data from the range

close all;
idx_pts = findchangepts(stress,'Statistic','linear','MaxNumChanges',7); % Signal processing add-on
pts = plot(strain(idx_pts),stress(idx_pts),'x-b',"DisplayName",'Fitted Line'); % Fitted Line
plot_datatips(pts, strain, stress, idx_pts, (3:4)) % Using these two points
plot_raw_data(strain, stress); % Raw Data
plot(strain, movmean(stress,19),'r','DisplayName','19 Point Moving Average'); % Filtered Data
xlim([0.001 0.02]); ylim([70 80]);
print_figure('3a') % Saves figure as PDF
```




```
fy = min(stress(idx_pts(3:4))) % Finds the fy as the minimum stress in the yield plateau
```

```
fy = 72.6104
```

```
e_sh = strain(idx_pts(4)) % Strain where strain hardening is consistent
```

```
e_sh = 0.0142
```

```
f_sh = stress(idx_pts(4)) % Stress where strain hardening is consistent
```

```
f_sh = 76.2543
```

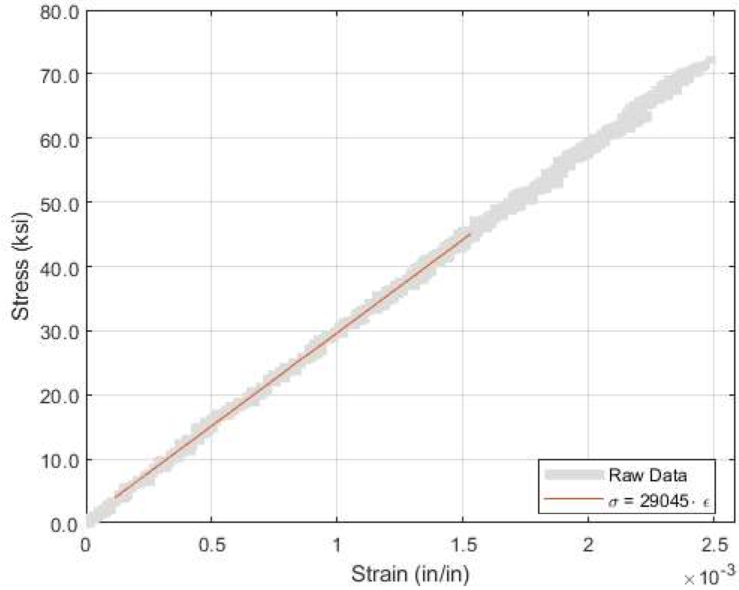
Using the signal processing toolkit, I was able to segment the data in seven linear portions. From there, using the indices as located by the algorithm, the strains at the boundaries of the Lüders yeild plateau were found. The yeild stress was taken as the minimum stress between the two corresponding stresses.

Part (ii) Determining Young's Modulus, E_s

```
strain_range = [0, 0.0025]; % As specified
[stress, strain] = get_data_for_range(strain_range);% Gets the data from the range

close all;
[~, idx1] = min(abs(0.05*fy- stress)); % Finds the index where the stress is equal to 0.05*fy
[~, idx2] = min(abs(0.6*fy- stress)); % Finds the index where the stress is equal to 0.6*fy
range = (idx1:idx2); % The ids where the two strains are met
coefs = polyfit(strain(range),stress(range),1); % Use a linear fit to find best fit the data
Es = coefs(1) % Modulus is just the slope of this line
```

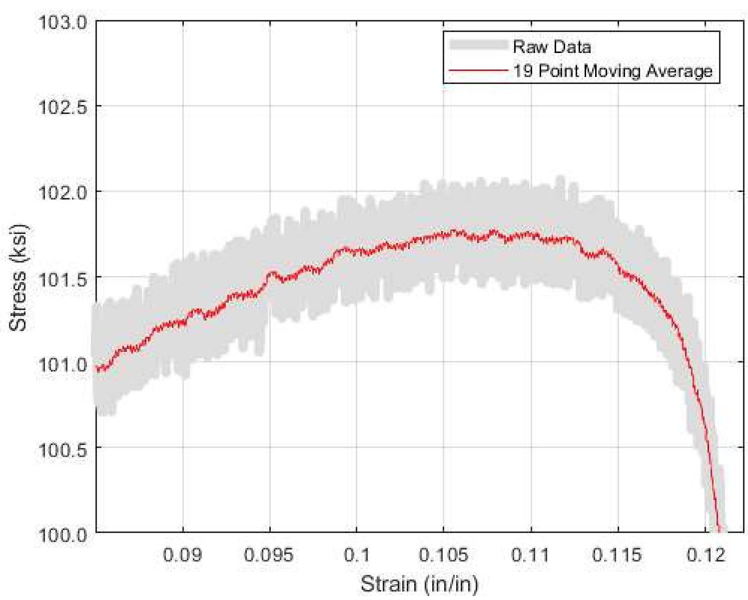
```
Es = 2.9045e+04
```

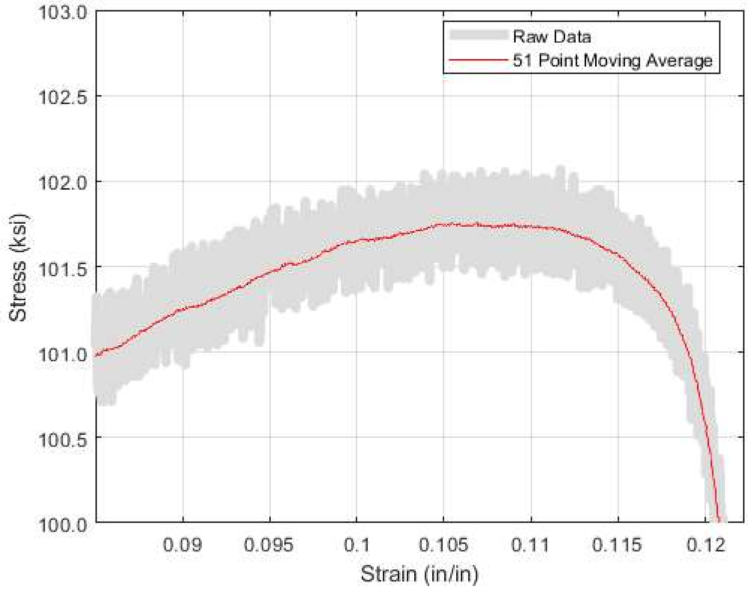
```
figure % Plot
plot_raw_data(strain, stress); % Plots raw data from start to 0.025 strain
"\sigma = " + sprintf("%5.0f", coefs(1)) + "\cdot \epsilon"; % Text for the plot name- stored as temporary variable
plot(strain(range), polyval(coefs,strain(range)),'-',"DisplayName", ans); % Plots the fitted line between 0.05fy and 0.6fy
print_figure(4)
```

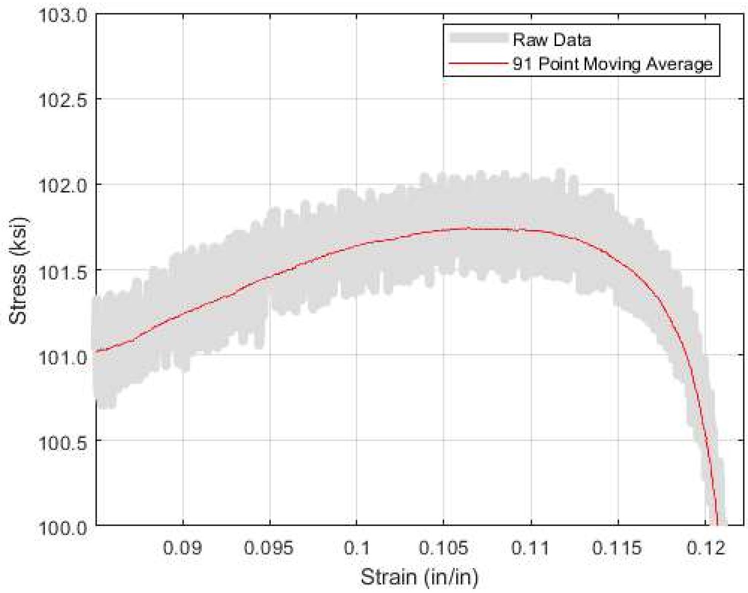


Part (iii) Determine Ultimate Stress f_{su} and Ultimate Strain ϵ_{su}

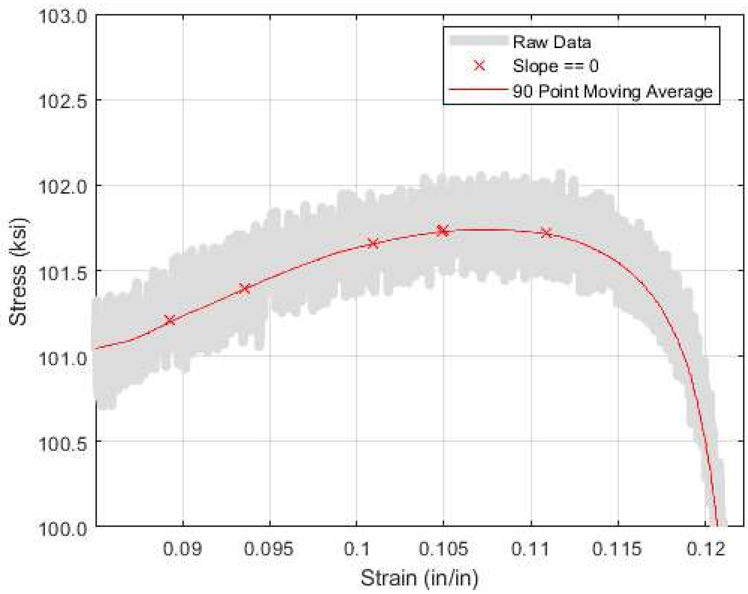
```
strain_range = [0.085, 0.125]; % As specified
[stress, strain] = get_data_for_range(strain_range); % Gets the data from the range

close all
avg_values = [19, 51, 91]; % The number of samples that need to be taken for the moving average
for i = 1:length(avg_values) % Loops through the avg values
    figure; % Set up figure, adjusted for figure numbers
    mov_pt_avg = avg_values(i); % Sets the sampling size
    plot_raw_data(strain, stress) % Plots raw data
    int2str(mov_pt_avg) + " Point Moving Average"; % Title for plot- temporary variable
    plot(strain, movmean(stress,mov_pt_avg),'r','DisplayName',ans); % Filtered Data
    ylim([100,103]); % Per office hours
    print_figure(i+4) % Save figure as PDF
end
```








```
strain_range = [0.085, 0.125]; % As specified
[stress, strain] = get_data_for_range(strain_range); % Gets data

close all
plot_raw_data(strain, stress) % Plots raw data
mov_pt_avg = 90; % Arbitrary
stress = movmean(stress,mov_pt_avg); % Filtered by moving average
slopes = diff(stress) ./ diff(strain); % Finds the slope of at various points
slopes(isinf(slopes)) = 10^-10; % Remove the inf from data
pt = find(slopes == 0); % Finds where the plot has 0 slope
plt = scatter(strain(pt),stress(pt),"Marker",'x',"MarkerEdgeColor","Red","DisplayName","Slope == 0"); % Plot thse points
string(mov_pt_avg) + " Point Moving Average"; % Title for plot- temporary variable
plot(strain, movmean(stress,mov_pt_avg),'r','DisplayName',ans); % Filtered Data
ylim([100,103]);
print_figure('7a')
```




```
e_su = mean(strain(pt(3:6))) % Start of the admissible range for strain
```

```
e_su = 0.1054
```

```
f_su = mean(stress(pt(3:6))) % Admissible Fsu
```

```
f_su = 101.7103
```

Question 2. Fit a Tri-Linear Model to the Processed Data

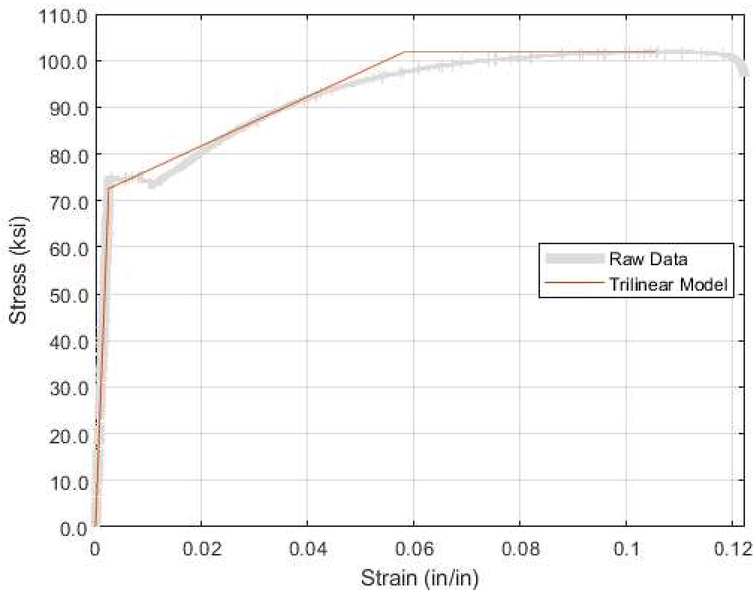
Plot the complete stress strain relationship to point (ϵ_{su}, f_{su}).

Compare the model and the processed data and show the value computed for the hardening ratio r.

```
strain_range = [0, 0.13]; % As specified
[stress, strain] = get_data_for_range(strain_range); % Get data for range

close all
plot_raw_data(strain, stress) % Plot raw data
zeta = 0.55; % As specifiec
e_y = fy/Es;
strain_pt = [0, e_y, zeta*e_su, e_su]; % As specifiec
stress_pt = [0, fy, f_su, f_su]; % As specifiec
plot(strain_pt,stress_pt,"DisplayName","Trilinear Model") % Plot
r = (f_su/fy - 1) / (zeta*e_su/e_y - 1) % Strain hardening example
```

```
r = 0.0181
```

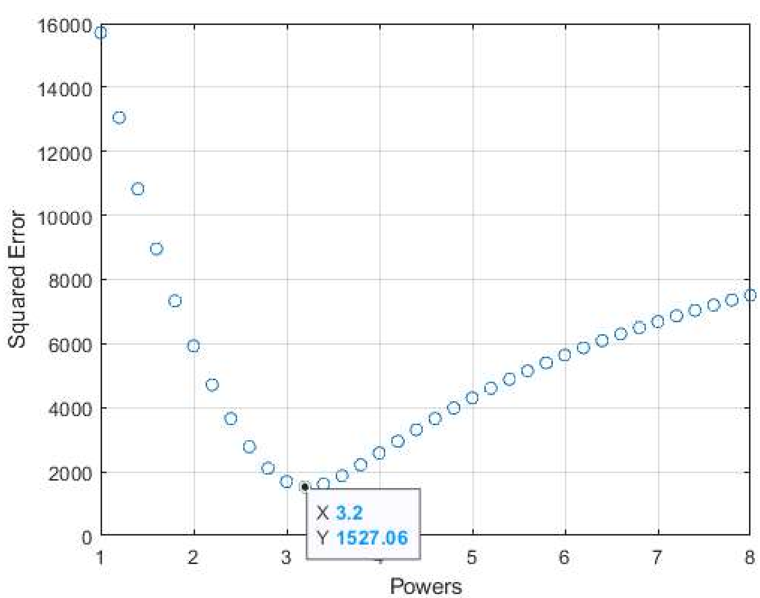
```
print_figure(8)
```



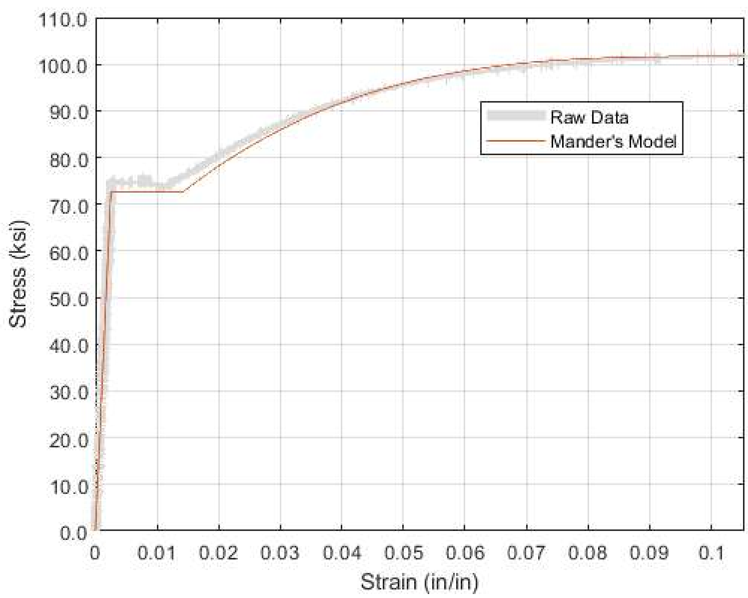
Question 3. Fit the Data to the Modified Mander's Model

Determine the power factor, P>1, through a least square error minimization procedure.

```
strain_range = [e_sh, e_su]; % Data range is from 0 to esu
[stress, strain] = get_data_for_range(strain_range);  % Get data

stress = movmean(stress, 91); % Assumed stress values; Varying this did not affect the P

P_trials = 1:0.2:8; % Selected powers for trial
sq_error = zeros(1,length(P_trials)); % Initialized sq erorr for results
i = 1; % Double loop counter- limitation of MATLAB (I'm joking)
for P = P_trials
    stress_guess = f_su - (f_su -fy) * ((e_su - strain)/(e_su- e_sh)).^P; % Generates the stress
    sq_error(i) = sum(sqrt((stress - stress_guess).^2)); % Calculates the sq. error
    i = i+1; % counter ++
end
P = P_trials(sq_error == min(sq_error));
close all;
plt = scatter(P_trials,sq_error); % As needed
datatip(plt,'DataIndex',find(sq_error == min(sq_error)),"Location","southeast"); % Min Point
xlabel("Powers"); ylabel("Squared Error");box on, grid on;
print_figure(9)
```



```
[model_stress, model_strain] = deal(linspace(0,e_su,1000)); % Initialize model strain range
i = 1;  % Double loop counter- limitation of MATLAB (I'm joking)
for epsilon = model_strain
    if epsilon <= e_sh % If the strain is before the start of the strain hardending
      model_stress(i) = min(fy, Es*epsilon); % Linear Function
      i = i+1;  % counter ++
    elseif epsilon >= e_sh && epsilon <= e_su % Else use the power function
      model_stress(i) = f_su - (f_su -fy) * ((e_su - epsilon)/(e_su- e_sh)).^P; % Mander's Model
      i = i+1; % counter ++
    end
end

close all
strain_range = [0, e_su]; % Data range
[stress, strain] = get_data_for_range(strain_range); % Get Data
plot_raw_data(strain, stress) % Plot
plot(model_strain, model_stress,"DisplayName","Mander's Model"); % Model data
print_figure(10)
```



Functions

```
function plot_datatips(pts, strain,stress,idx,range)
    % For that one figure ... spent too much time on this one
    pts.DataTipTemplate.Interpreter = 'tex';
    pts.DataTipTemplate.DataTipRows(1).Format = '%0.3g';
    pts.DataTipTemplate.DataTipRows(1).Label = '\epsilon';
    pts.DataTipTemplate.DataTipRows(2).Format = '%.2f';
    pts.DataTipTemplate.DataTipRows(2).Label = '\sigma';
    for pt = idx(range)'
        datatip(pts,strain(pt),stress(pt),"location","southeast");
    end
end

function [stress, strain] = get_data_for_range(range_)
    % Import the data but also cuts it into the range that I need
    opts = spreadsheetImportOptions("NumVariables", 4);
    opts.Sheet = "Data 2021";
    opts.DataRange = "A7:D5116";
    opts.VariableNames = ["Var1", "Stress", "Var3", "Strain"];
    opts.SelectedVariableNames = ["Stress", "Strain"];
    opts.VariableTypes = ["char", "double", "char", "double"];
    opts = setvaropts(opts, ["Var1", "Var3"], "WhitespaceRule", "preserve");
    opts = setvaropts(opts, ["Var1", "Var3"], "EmptyFieldRule", "auto");
    file = "C:\Users\Louis Lin\Workspace\Academic\UCSD\SE 211\Homework\HW 1\data\No 11 Bar Tensile Test 2021.xlsx";
    No11BarTensileTest2021 = readtable(file, opts, "UseExcel", false);

    % For the range I need
    stress = No11BarTensileTest2021.Stress; % Saves the stress data
```

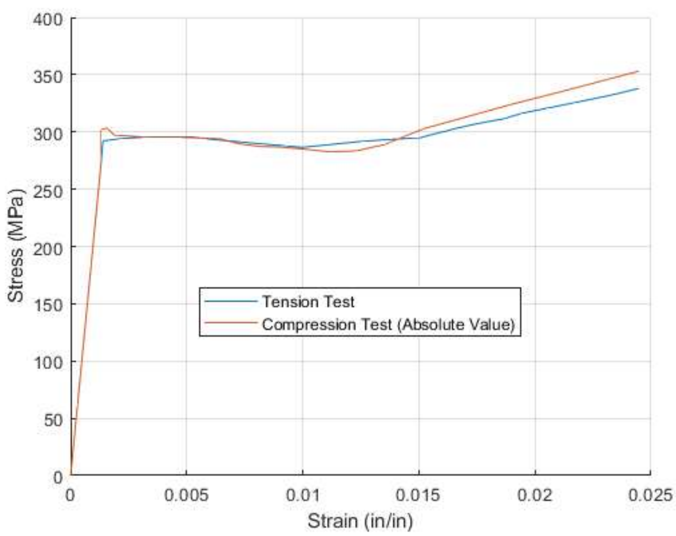
```
    strain = No11BarTensileTest2021.Strain; % Saves the strain data
    strain_range = and(strain >= range_(1), strain<=range_(2)); % Selects a data range as specified
    stress = stress(strain_range);
    strain = strain(strain_range);
end

function plot_raw_data(strain, stress)
    % Formats the raw data plot consistently
    hold on; grid on; box on; % Per choice
    ylabel("Stress (ksi)"); xlabel("Strain (in/in)"); ytickformat('%.1f');
    axis([max(0,min(strain)),max(strain)+0.0001,max(0,round(min(stress),-1)-10),round(max(stress),-1)+10]) % Per choice
    raw_data = plot(strain,stress,"Color",[220,220,220]/255,'LineWidth', 5,'DisplayName',"Raw Data");
    raw_data.Color(4) = 0.3;% Raw Data alpha
    legend('Location',"best","Interpreter","tex");
end

function print_figure(no)
    % Saves the figures in a consistent manner
    orient(gcf,'landscape');
    folder = '..\figures\';
    name = 'Figure' +string(no);
    print(folder+name,'-dpdf','-PMicrosoft Print to PDF','-fillpage','-r600','-painters')
end
```

Part(II) Modeling Tensile Stress-Strain Response of Mild Steel Reinforcement

Question 1. Plot the stress-strain data in both engineering and true coordinate systems

Part (i)

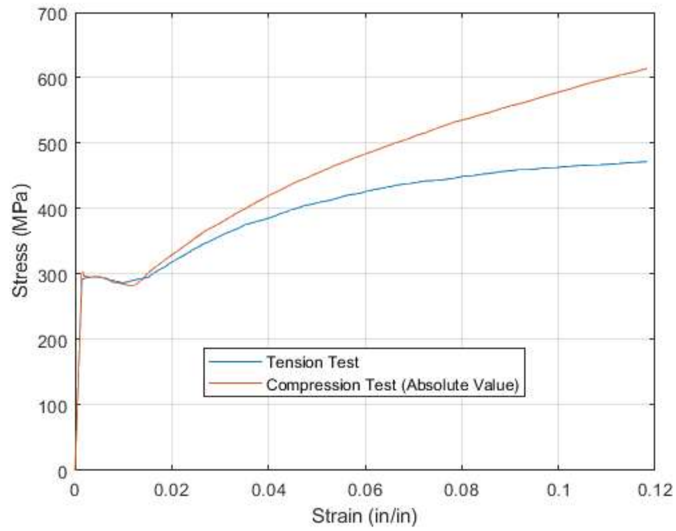
```
% This portion is to examine only the data before strain hardening
clc; clear; close all;
range_ = [0, 0.025]; % data range
[compressive_strain, compressive_stress, tensile_strain, tensile_stress] = get_data_for_range(range_); % gets data; see functions

figure(1); hold on; grid on;  % Plot
plot(tensile_strain,tensile_stress,"DisplayName","Tension Test");
plot(-compressive_strain,-compressive_stress,"DisplayName","Compression Test (Absolute Value)"); % Note: include a negative sign
legend("Location","best"); ylabel("Stress (MPa)"); xlabel("Strain (in/in)"); %
print_figure('11b') % Save figure as pdf
```




```
tensile = [tensile_strain, tensile_stress];
compressive =- [compressive_strain,compressive_stress];
DiscreteFrechetDist(compressive,tensile)
```

```
ans = 17.3740
```

```
% This portion is asked for the homework.
close all; % New plot
figure(2); hold on; grid on; box on; % Choice
range_ = [0, 0.12]; % Strain range
[compressive_strain, compressive_stress, tensile_strain, tensile_stress] = get_data_for_range(range_); % Get data
plot(tensile_strain,tensile_stress,"DisplayName","Tension Test"); %Plot
plot(-compressive_strain,-compressive_stress,"DisplayName","Compression Test (Absolute Value)"); % Note negative values
legend("Location","best"); ylabel("Stress (MPa)"); xlabel("Strain (in/in)"); % Plot things
print_figure(11); % Save fig as pdf
```

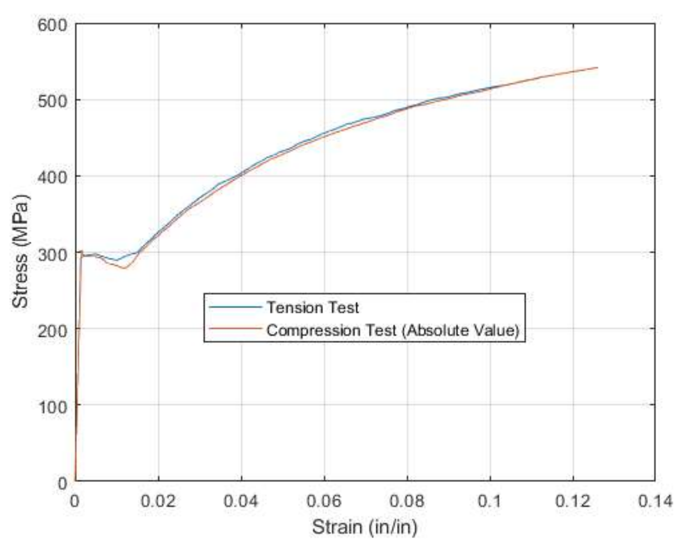



```
tensile = [tensile_strain, tensile_stress];
compressive =- [compressive_strain,compressive_stress];
DiscreteFrechetDist(compressive,tensile)
```

```
ans = 142.5810
```

Transform the stress-strain relationship from engineering to true coordinate system and plot the true stress-strain tensile and compressive relationships in absolute values.

```
% This section is the plot for true stress-strain
close all; % New plot
figure(3); hold on; grid on; box on;
range_ = [0, 0.12]; % Defines the range of strain data
[compressive_strain, compressive_stress, tensile_strain, tensile_stress] = get_data_for_range(range_); % Grabs the data from excel

natural_tensile_stress = tensile_stress.*(1+tensile_strain); % stress' = stress * (1 + strain)
natural_tensile_strain = log(1 + tensile_strain); % strain' = ln(1 + strain)

natural_compressive_stress = - compressive_stress.*(1 + compressive_strain); % stress' = stress * (1 + strain)
natural_compressive_strain = -log(1 + compressive_strain); % strain = ln(1 + strain)

plot(natural_tensile_strain,natural_tensile_stress,"DisplayName","Tension Test"); % Plot
plot(natural_compressive_strain,natural_compressive_stress,"DisplayName","Compression Test (Absolute Value)"); % Plot
legend("Location","best"); ylabel("Stress (MPa)"); xlabel("Strain (in/in)"); % Plot
print_figure(12); % Save fig as pdf
```




```
tensile = [natural_tensile_strain, natural_tensile_stress];
compressive = [natural_compressive_strain,natural_compressive_stress];
DiscreteFrechetDist(compressive,tensile)
```

```
ans = 16.7994
```

```
function [compressive_strain, compressive_stress, tensile_strain, tensile_stress] = get_data_for_range(range_)
```

```
    opts = spreadsheetImportOptions("NumVariables", 2);
    % Specify sheet and range
    opts.Sheet = "Compressive";
    opts.DataRange = "B3:C99";
    % Specify column names and types
    opts.VariableNames = ["Strain", "Stress"];
    opts.VariableTypes = ["double", "double"];
    % Import the data
    tbl = readtable("C:\Users\Louis Lin\Workspace\Academic\UCSD\SE 211\Homework\HW 1\data\Monotonic tensile compressive on machined coupor
    strain_range = and(tbl.Strain <= -range_(1), tbl.Strain >= -range_(2)); % Selects a data range as specified
    compressive_strain = tbl.Strain(strain_range);
    compressive_stress = tbl.Stress(strain_range);

    opts = spreadsheetImportOptions("NumVariables", 2);
    % Specify sheet and range
    opts.Sheet = "Tensile";
    opts.DataRange = "B4:C128";
    % Specify column names and types
    opts.VariableNames = ["Strain", "Stress"];
    opts.VariableTypes = ["double", "double"];
    % Import the data
    tbl = readtable("C:\Users\Louis Lin\Workspace\Academic\UCSD\SE 211\Homework\HW 1\data\Monotonic tensile compressive on machined coupor
    strain_range = and(tbl.Strain >= range_(1), tbl.Strain<=range_(2)); % Selects a data range as specified
    tensile_strain = tbl.Strain(strain_range);
    tensile_stress = tbl.Stress(strain_range);
end

function print_figure(no)
    orient(gcf,'landscape');
    folder = '..\figures\';
    name = 'Figure' +string(no);
    print(folder+name,'-djpeg','-PMicrosoft Print to PDF','-r600','-painters')
    print(folder+name,'-dpdf','-fillpage','-PMicrosoft Print to PDF','-r600','-painters')
end

function [cm, cSq] = DiscreteFrechetDist(P,Q,dfcn)
% Calculates the discrete Frechet distance between curves P and Q
%
% [cm, cSq] = DiscreteFrechetDist(P,Q)
% [cm, cSq] = DiscreteFrechetDist(...,dfcn)
%
% P and Q are two sets of points that define polygonal curves with rows of
% vertices (data points) and columns of dimensionality. The points along
% the curves are taken to be in the order as they appear in P and Q.
%
% Returned in cm is the discrete Frechet distance, aka the coupling
% measure, which is zero when P equals Q and grows positively as the curves
% become more dissimilar.
%
% The optional dfcn argument allows the user to specify a function with
% which to calculate distance between points in P and Q. If not provided,
% the L2 norm is used.
%
% The secondary output, cSq, is the coupling sequence, that is, the
% sequence of steps along each curve that must be followed to achieve the
% minimum coupling distance, cm. The output is returned in the form of a
% matrix with column 1 being the index of each point in P and column 2
% being the index of each point in Q. (NOTE: the coupling sequence is not
% unique in general)
%
% Explanation:
% The Frechet distance is a measure of similarity between to curves, P and
% Q. It is defined as the minimum cord-length sufficient to join a point
% traveling forward along P and one traveling forward along Q, although the
% rate of travel for either point may not necessarily be uniform.
%
% The Frechet distance, FD, is not in general computable for any given
% continuous P and Q. However, the discrete Frechet Distance, also called
% the coupling measure, cm, is a metric that acts on the endpoints of
% curves represented as polygonal chains. The magnitude of the coupling
% measure is bounded by FD plus the length of the longest segment in either
% P or Q,  and approaches FD in the limit of sampling P and Q.
%
% This function implements the algorithm to calculate discrete Frechet
% distance outlined in:
% T. Eiter and H. Mannila. Computing discrete Frechet distance. Technical
% Report 94/64, Christian Doppler Laboratory, Vienna University of
% Technology, 1994.
%
%
%
```

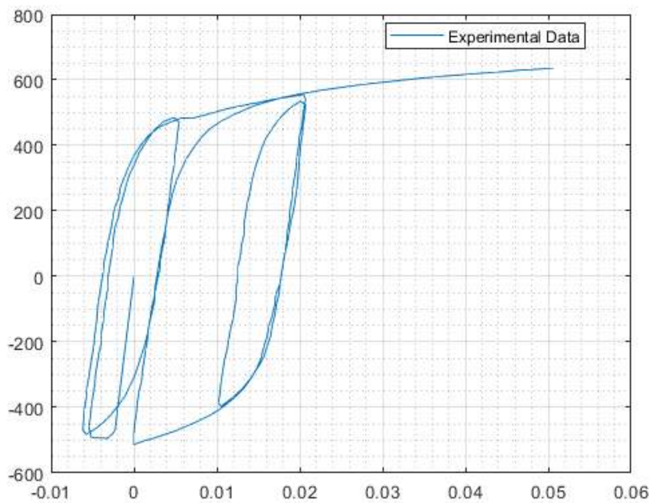
```
% EXAMPLE:
% % create data
% t = 0:pi/8:2*pi;
% y = linspace(1,3,6);
% P = [(2:7)' y']+0.3.*randn(6,2);
% Q = [t' sin(t')]+2+0.3.*randn(length(t),2);
% [cm, cSq] = DiscreteFrechetDist(P,Q);
% % plot result
% figure
% plot(Q(:,1),Q(:,2),'o-r','linewidth',3,'markerfacecolor','r')
% hold on
% plot(P(:,1),P(:,2),'o-b','linewidth',3,'markerfacecolor','b')
% title(['Discrete Frechet Distance of curves P and Q: ' num2str(cm)])
% legend('Q','P','location','best')
% line([2 cm+2],[0.5 0.5],'color','m','linewidth',2)
% text(2,0.4,'dFD length')
% for i=1:length(cSq)
%   line([P(cSq(i,1),1) Q(cSq(i,2),1)],...
%        [P(cSq(i,1),2) Q(cSq(i,2),2)],...
%        'color',[0 0 0]+(i/length(cSq)/1.35));
% end
% axis equal
% % display the coupling sequence along with each distance between points
% disp([cSq sqrt(sum((P(cSq(:,1),:) - Q(cSq(:,2),:)).^2,2))])
%
%
%
% %%% ZCD June 2011 %%%
% %%% edits ZCD May 2013: 1) remove excess arguments to internal functions
% and persistence for speed, 2) added example, 3) allowed for user defined
% distance function, 4) added aditional output option for coupling sequence
%
% size of the data curves
sP = size(P);
sQ = size(Q);
% check validity of inputs
if sP(2)~=sQ(2)
    error('Curves P and Q must be of the same dimension')
elseif sP(1)==0
    cm = 0;
    return;
end
% initialize CA to a matrix of -1s
CA = ones(sP(1),sQ(1)).*-1;
% distance function
if nargin==2
    dfcn = @(u,v) sqrt(sum( (u-v).^2 ));
end
% final coupling measure value
cm = c(sP(1),sQ(1));
% obtain coupling measure via backtracking procedure
if nargout==2
    cSq = zeros(sQ(1)+sP(1)+1,2);    % coupling sequence
    CApad = [ones(1,sQ(1)+1)*inf; [ones(sP(1),1)*inf CA]];  % pad CA
    Pi=sP(1)+1; Qi=sQ(1)+1; count=1;  % counting variables
    while Pi~=2 || Qi~=2
        % step down CA gradient
        [v,ix] = min([CApad(Pi-1,Qi) CApad(Pi-1,Qi-1) CApad(Pi,Qi-1)]);
        if ix==1
            cSq(count,:) = [Pi-1 Qi];
            Pi=Pi-1;
        elseif ix==2
            cSq(count,:) = [Pi-1 Qi-1];
            Pi=Pi-1; Qi=Qi-1;
        elseif ix==3
            cSq(count,:) = [Pi Qi-1];
            Qi=Qi-1;
        end
        count=count+1;
    end
    % format output: remove extra zeroes, reverse order, subtract off
    % padding value, and add in the last point
    cSq = [flipud(cSq(1:find(cSq(:,1)==0,1,'first')-1,:))-1; sP(1) sQ(1)];
end
% debug
% assignin('base','CAw',CA)
function CAij = c(i,j)
    % coupling search function
    if CA(i,j)>-1
        % don't update CA in this case
        CAij = CA(i,j);
```

```
    elseif i==1 && j==1
        CA(i,j) = dfcn(P(1,:),Q(1,:));     % update the CA permanent
        CAij = CA(i,j);                    % set the current relevant value
    elseif i>1 && j==1
        CA(i,j) = max( c(i-1,1), dfcn(P(i,:),Q(1,:)) );
        CAij = CA(i,j);
    elseif i==1 && j>1
        CA(i,j) = max( c(1,j-1), dfcn(P(1,:),Q(j,:)) );
        CAij = CA(i,j);
    elseif i>1 && j>1
        CA(i,j) = max( min([c(i-1,j), c(i-1,j-1), c(i,j-1)]),...
            dfcn(P(i,:),Q(j,:)) );
        CAij = CA(i,j);
    else
        CA(i,j) = inf;
    end
end     % end function, c
end     % end main function
```

Part(III) Modeling Tensile Stress-Strain Response of Mild Steel Reinforcement

This portion of the homework looks to model the behavior of steel rebar using different stiffness hardening models using results from a cyclic loading on a coupon test. The aspect ratio of the test coupon is sufficiently small to ensure that nonlinear geometrical effects are negligible.

```
clear; clc; close all; hold on; grid on; box on; % Housekeeping
[test_stress, test_strain] = get_data_for_range(); % Get data from excel
plot_model(test_strain, test_stress,(1:273),"Experimental Data"); % Plot the test data
```



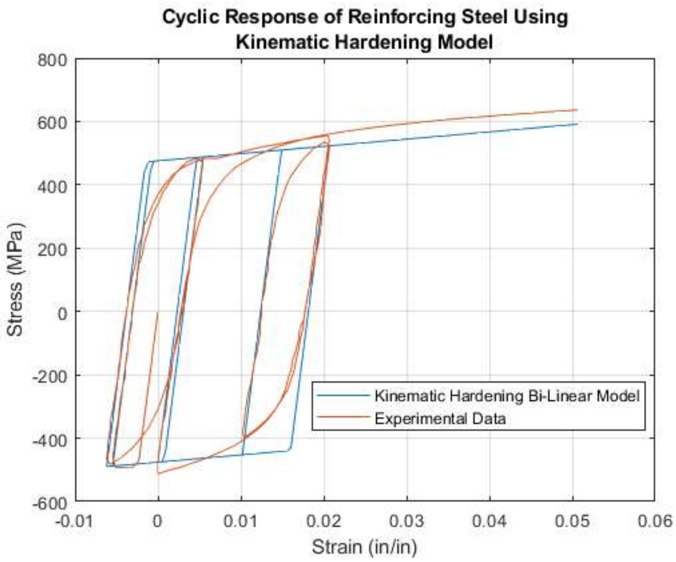

```
[fy, fsu, Es, esh, esu] = get_material_prop(); Es = Es*10^3; % Grabs the material property
ey = fy/Es; zeta = 0.6; r = (fsu/fy - 1) / (zeta*esu/ey - 1); % Calculates some properties
```

Question 1. Kinematic Hardening Cyclic Model

Use a kinematic hardneing cyclic model for $\zeta = 0.6$ to predic the bar stress corresponding to each strain. The code was optimized a little in comparison with the Isotropic Hardening model as this model was developed second- also why it's lavel model2.

```
[model2_strain, model2_stress] = deal(zeros(numel(test_strain),1)); % Initialize model
strain_max = 0; stress_max = 0; % Initial max bounds
strain_min = -ey; stress_min = -fy; % Initial min bounds
branch = 3; % Initial branch
end_no = 273; % Last index of model (used for developing branch by branch)

for index = 2:end_no % For all of the strains in the test strains
    delta_strain = test_strain(index) - test_strain(index -1); % Find the change in strain
    model2_strain(index) = test_strain(index); % Take the current test strain (unnecessary actually)

    if branch == 1 || branch == 3 % If you're on branch 1 or 3
        if strain_min <= model2_strain(index) && model2_strain(index) <= strain_max % As long as you're within the bounds
            model2_stress(index) = min(max(Es * (model2_strain(index)-strain_min) + stress_min,stress_min),stress_max); % Calculate the st
        else % You're moving to a different branch
            if model2_strain(index) > strain_max % If moving to the right, Moving from branch 1 -> 2
                branch = 2; % Set new branch
                strain_min = strain_max; % Set new min
                stress_min = stress_max; % Set new min
            elseif model2_strain(index) < strain_min % If moving to the left,  Moving from Branch 3 -> 4
                branch = 4; % Set new branch
            end
        end
    end

    if (branch == 2 && delta_strain > 0) || (branch == 4 && delta_strain < 0) % On Branch 2 || 4, and the directions are correct
        model2_stress(index) = r*Es*(model2_strain(index) - strain_min) + stress_min; % Calculate the stress
    else % The direction has reversed and moving to a new branch
        if delta_strain >= 0 && branch == 4 % If moving to the right, Moving from Branch 4 -> 1
            branch = 1; % New Branch
            strain_min = model2_strain(index -1); stress_min = model2_stress(index-1); % Set the min as the previous dot
            stress_max = stress_min + (2*fy); strain_max = strain_min + 2*ey; % The yield point is 2*fy and 2*ey away
            model2_stress(index) = Es * (model2_strain(index)-strain_min) + stress_min; % Should be capped at stress min
        elseif delta_strain < 0 && branch == 2 % If moving to the left, Moving from Branch 2 -> 3
            branch = 3; % New Branch
            strain_max = model2_strain(index -1 ); stress_max = model2_stress(index-1); % Set the min as the previous dot
            stress_min = model2_stress(index-1) - (2*fy); strain_min = strain_max - 2*ey; % The yield point is 2*fy and 2*ey away
            model2_stress(index) = Es * (model2_strain(index)-strain_min) + stress_min; % Should be capped at stress max
        end
```

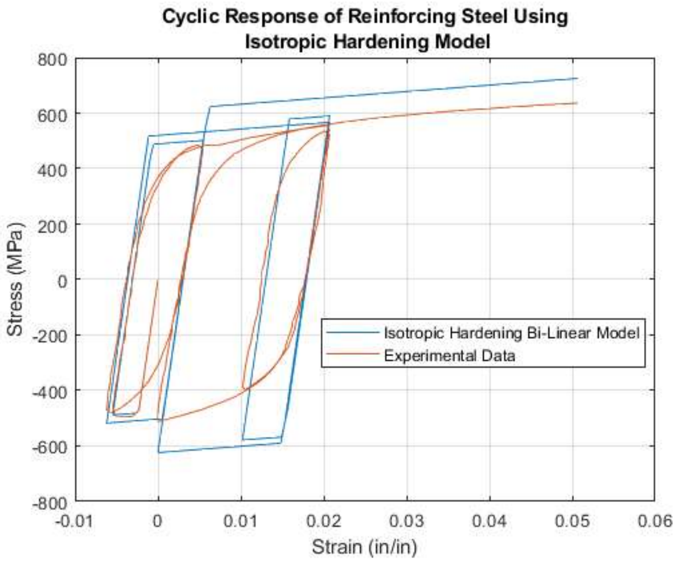
```
    end
end

close all; range = 1:end_no;
plot_model(model2_strain,model2_stress,range,"Kinematic Hardening Bi-Linear Model");
plot_model(test_strain, test_stress,range,"Experimental Data");
title(["Cyclic Response of Reinforcing Steel Using","Kinematic Hardening Model"])
legend("Location","best"); ylabel("Stress (MPa)"); xlabel("Strain (in/in)"); %
print_figure(13)
```



Question 2. Isotropic Hardening Cyclic Model

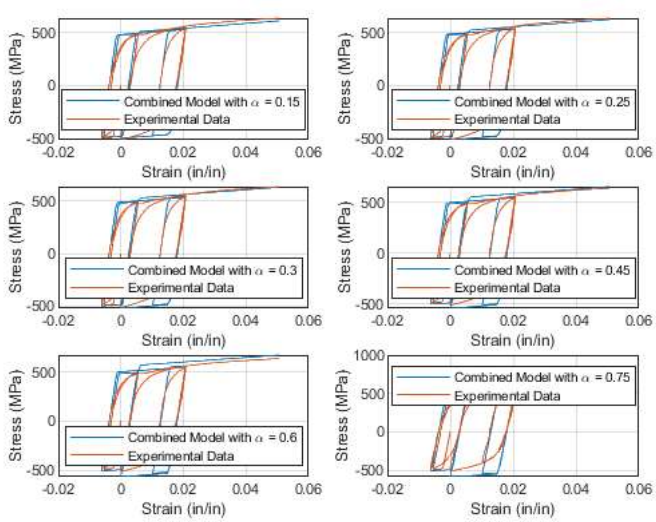
```
[model1_strain, model1_stress] = deal(zeros(numel(test_strain),1)); % Initialize model
strain_max = 0; stress_max = 0; % Initial max bounds
strain_min = -ey; stress_min = -fy; % Initial min bounds
branch = 3; % Initial branch
end_no = 273; % Last index of model (used for developing branch by branch)

for index = 2:end_no % For all of the strains in the test strains
    delta_strain = test_strain(index) - test_strain(index -1); % Find the change in strain
    model1_strain(index) = test_strain(index); % Take the current model strain

    if branch == 1 || branch == 3 % If you're on branch 1 or 3
        if strain_min <= model1_strain(index) && model1_strain(index) <= strain_max % As long as you're within the bounds
            model1_stress(index) = Es * (model1_strain(index)-strain_max) + stress_max; % Calculate the stress
        else % You're moving to a different branch
            if model1_strain(index) > strain_max % If moving to the right, Moving from branch 1 -> 2
                branch = 2; % Set new branch
                model1_stress(index) = min(r*Es * (model1_strain(index)-strain_max) + stress_max, stress_max); % Use branch 2's eq
                [strain_min, stress_min] = deal(model1_strain(index) , model1_stress(index)); % Set new min
            elseif model1_strain(index) < strain_min % If moving to the left,  Moving from Branch 3 -> 4
                branch = 4; % Set new branch
                model1_stress(index) = r*Es * (model1_strain(index)-strain_min) + stress_min; % Use branch 4's eq
                [strain_max, stress_max] = deal(model1_strain(index) , model1_stress(index)); % Set new max
            end
        end
    end

    if (branch == 2 && delta_strain > 0) || (branch == 4 && delta_strain < 0) % On Branch 2 || 4, and the directions are correct
        model1_stress(index) = r*Es*(model1_strain(index) - strain_min) + stress_min; % Calculate the stress
    else % The direction has reversed and moving to a new branch
        if delta_strain >= 0 && branch == 4 % If moving to the right, Moving from Branch 4 -> 1
            branch = 1; % New Branch
            strain_min = model1_strain(index -1 ); stress_min = model1_stress(index-1); % Set the min as the previous dot
            stress_max = -model1_stress(index-1); strain_max = strain_min + abs(2*model1_stress(index-1)/Es); % yield point is equal dista
            model1_stress(index) = max(Es * (model1_strain(index) - strain_min) + stress_min, stress_min); % Use branch 1's eq
        elseif delta_strain < 0 && branch == 2 % If moving to the left, Moving from Branch 2 -> 3
            branch = 3; % New Branch
            strain_max = model1_strain(index -1 ); stress_max = model1_stress(index-1);% Set the min as the previous dot
            stress_min = -model1_stress(index-1); strain_min = strain_max - abs(2*model1_stress(index-1)/Es); % yield point is equal dista
            model1_stress(index) = min(Es * (model1_strain(index) - strain_max) + stress_max, stress_max); % Use branch 3's eq
        end
    end
end

close all; range = 1:end_no;
plot_model(model1_strain,model1_stress,range,"Isotropic Hardening Bi-Linear Model"); plot_model(test_strain, test_stress,range,"Experiment
```

```
title(["Cyclic Response of Reinforcing Steel Using"," Isotropic Hardening Model"])
legend("Location","best"); ylabel("Stress (MPa)"); xlabel("Strain (in/in)"); %
print_figure(14)
```



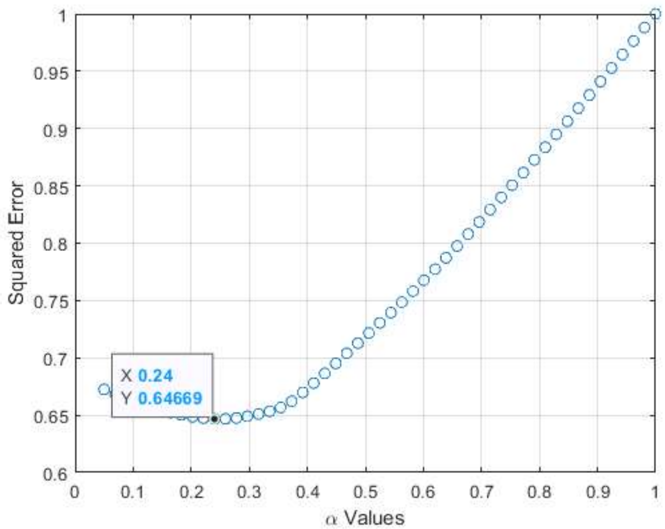
Question 3. Combined Kinematic-Isotropic Hardening Cyclic Model

```
alpha_values = [0.15, 0.25, 0.3, 0.45, 0.6, 0.75]; % As specified
figure % One figure with 6 subplots
i = 1; % Counter for plot number
for a = alpha_values
    subplot(3,2,i); i = i+1; % Subplot i-th
    stress = a.*model1_stress + (1-a).*model2_stress; % Calculate the combined stress values
    plot_model(model2_strain,stress,range,"Combined Model with \alpha = " + string(a)); % Plots combined stress
    plot_model(test_strain, test_stress,range,"Experimental Data"); % Adds the experimental data
    legend("Location","best"); ylabel("Stress (MPa)"); xlabel("Strain (in/in)"); %
end
print_figure('14a')
```



Find the best alpha value.

```
figure() % new figure
alpha_values = linspace(0.05,1,51); % Test values for alpha
sq_error = zeros(length(alpha_values),1); % Sq errors
i = 1; % Double loop counter
for a = alpha_values % For various alpha values
    stress = a.*model1_stress + (1-a).*model2_stress; % Calculate the all of the stresses
    sq_error(i) = sum(sqrt((stress - test_stress).^2)); i = i+1; % Calculate the sq error
end
sq_error = sq_error/ max(sq_error);
plt = scatter(alpha_values,sq_error); % Plot
datatip(plt,'DataIndex',find(sq_error==min(sq_error)),"Location","northwest"); %Label Minimum
xlabel("\alpha Values",'Interpreter','tex'); ylabel("Squared Error"); box on, grid on; % options
print_figure('14b') % Save as pdf
```




```
function plt = plot_model(strain,stress,range,name)
    plt = plot(strain(range),stress(range),"DisplayName",name);
    hold on; legend("location","best"); grid minor; grid on;
end


function [stress, strain] = get_data_for_range()
    opts = spreadsheetImportOptions("NumVariables", 3);

    % Specify sheet and range
    opts.Sheet = "Data 2021";
    opts.DataRange = "A14:C286";

    % Specify column names and types
    opts.VariableNames = ["Strain","Stress","Blank"];
    opts.VariableTypes = ["double", "double", "double"];

    % Import the data
    tbl = readtable("C:\Users\Louis Lin\Workspace\Academic\UCSD\SE 211\Homework\HW 1\data\Machined coupon cyclic test 2021.xlsx", opts, "U

    stress = tbl.Stress;
    strain = tbl.Strain;
end

function [fy, fsu, Es, esh, esu] = get_material_prop()
    opts = spreadsheetImportOptions("NumVariables", 1);
    opts.Sheet = "Data 2021";
    opts.DataRange = "B4:B8";
    opts.VariableNames = "info";
    opts.VariableTypes = "double";
    tbl = readtable("C:\Users\Louis Lin\Workspace\Academic\UCSD\SE 211\Homework\HW 1\data\Machined coupon cyclic test 2021.xlsx", opts, "U
    info = num2cell(tbl.info);
    [fy, fsu, Es, esh, esu] = deal(info{:});
end

function print_figure(no)
    orient(gcf,'landscape');
    folder = '..\figures\';
    name = 'Figure' +string(no);
    print(folder+name,'-dpdf','-PMicrosoft Print to PDF','-fillpage','-r600','-painters')
end
```