TENSILE TESTING OVERVIEW

Ellen Klonowski
Department of Biomedical Engineering
Lerner Research Institute, Cleveland Clinic
Cleveland, Ohio, USA

Introduction

Mechanical testing is important to research to determine structural and material properties for representation in modeling and simulations. Testing in a Mach 1 Mechanical System (Biomomentum Inc, Laval, Québec, Canada) provides raw data as a result of a tensile test and the ability to design a test suitable for the material. A motion controller (Newport Model ESP301) controls movement of the load cell and position of the specimen held in clamps. Mach 1 Motion (Biomomentum Inc, Laval, Québec, Canada) software allows a user to control the machine and Mach 1 Analysis (Biomomentum Inc, Laval, Québec, Canada) software provides a user to look at the data by sequence. The specimen in the case is a rectangular piece of rubber tensile tested in the z direction. The test outputs time (seconds), the x, y, and z position (mm) of the load cell along with the gram load measurement on the specimen. The raw data is used to calculate force-displacement and stress-strain curves. Force-displacement curves are useful for determining stiffness and stress-strain curves are useful for evaluating the elastic modulus of a material. The particular test described in this report includes a ramp up of force at 15% max strain on the material and preconditioning followed by an identical force ramp up on the specimen to compare to the first ramp sequence. Data from the test and identical tests are compared for repeatability. Preparation of the sample and test parameters are noted for analyses and finding discrepancies in the data. An example case of using this procedure will be described below using a rubber sample tested in tension twice.

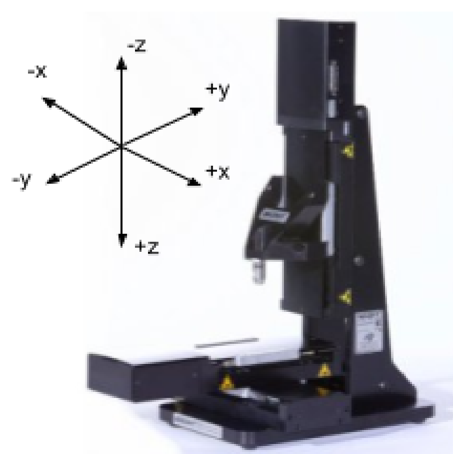


Figure 1. Mach 1 Micromechanical System [1] and the coordinate system used in the test.

Tensile Test Overview

Pycharm [https://www.jetbrains.com/pycharm/] with an Anaconda2 distribution [https://www.anaconda.com/distribution/] is a crucial part of the data analysis before the tensile test and after. Three scripts in total are used. Two scripts are needed for determining test parameters and one for the data extraction as mentioned. Displacement of the material and machine movement are recorded in millimeters, time is recorded in seconds, and load is converted from grams to Newtons. A 1 kg load cell (ATI Industrial Automation, Apex, NC) is used to measure the load on the material and calibrated with a 500g weight. A photo of the specimen is taken manually with a camera (FlyCapture, FMVV-13S2C-CS) for an optical measurement to find the thickness of the sample. A custom GUI script [2] is used to pick points along the boundaries of the sample lying flat on a surface. The number of pixels are initially calibrated with a ruler in the image for an accurate measurement. This method is useful for irregular samples such as biological materials. At least ten lines connecting each upper and lower boundary are sufficient to find the thickness as the script averages the optical length measurements. Before the second script is run, a sequence of movements on the machine are necessary to find the initial length to measures when the load cell registers a force under tension on the specimen up to a prescribed force. All data is collected at 2500 Hz in order to record the maximum signal output. The initial length recorded during the "Wait" period of the sequence is the input along with sample thickness into the second script to determine the max strain and velocity as well as other moves relative and absolute on the specimen. These moves are important input into the second script, 'tension_pc.py', to determine movement of the machine and ultimately tension on the material. During the "tension_pc" sequence, the Mach 1 software writes the data to a text file called "data.txt". The final script 'tissuecheck.py' [3] executed using PyCharm extracts data from the text file output from the Mach 1 Analysis software.

Initial Length Sequence

The initial length sequence is used to find the initial, unstretched length of the specimen where there is no force on the sample. The machine is zeroed on the z axis as the empty clamps that hold the specimen are moved to where there is virtually no distance between them. In this case, the zero length measurement is validated with only the naked eye as one must be cautious not to damage the load cell. Any length measurement after this point reflects the length of the sample. The material is then placed in the clamps securely at each end with some slack. The sequence should then be initiated with an aligned specimen and tared load. Steps in the sequence include Zero Load, Find Contact, Wait, and Move Relative. Zero load step indicates zero load on the specimen as there should be some slack and should be manually tared before running the test. Find contact applies tension to the sample as the machine reaches the prescribed contact load measurement and is then held in that position for twenty seconds before moving two millimeters in the positive direction. In this case, the find contact criteria is

10g at a rate of .005mm/s. The initial length is recorded from the "Wait" step and also corresponds to the point where the tensile force on the material reaches the find contact load specified by the user. Finally, the last movement compresses the material by two millimeters.

Tension_pc.py script

The tension_pc.py script accepts the initial length value, max strain, and thickness of the sample. It simplifies the process of calculating the velocity and uniaxial movements of the machine during the actual tensile test relative to the initial length. These values are inserted in the tension_pc sequence in the Mach 1 Motion software settings because they vary depending on the three previously mentioned parameters. The script prints values that follow sequential order of the tensile test for input into the Mach 1 Motion program. Movements include move absolute, move relative, move relative, move relative, sinusoid, move relative, move relative, and move relative. Move absolute is a 0.3 millimeter movement to compress the material before the initial ramp at 15% max strain on the material, or 15% strain beyond initial length. Velocity of the machine at all moves relative is calculated at 20% of the initial length per second. Preconditioning the sample consists of stretching the material to 10% beyond its initial length in the negative z direction, and then compressing the same distance in the positive z direction repeatedly. The amplitude of the sinusoidal preconditioning cycles occurs at 2.5% of the initial length. Preconditioning in this test is defined as 1000 cycles at 2Hz. The sample returns to its initial length after both ramps on the material and preconditioning.

Tissue Check Script

The tissue check script ultimately performs the calculations for the force-displacement and stress-strain curves. It has the ability to create results from multiple tests with varying initial lengths and overlay results on plots. The script requires initial lengths, thickness, and width of sample. Below is a list of equations that the code utilizes.

- Cross sectional area A= thickness*width
- Stress: ((c*.0098)/A) c comes from filtered_r_load converts to newtons
- Strain:(c/initial length) c comes from norm_f_disp
- Gram force conversion = mass(mm)*.0098

An important function of the code is to filter the data. During the initial length sequence a low pass filter in the Mach 1 Motion software is applied to filter noise, but turned off for the tensile test. Two Python functions are needed to filter the data, one determining the low pass butterworth filter's cutoff frequency and the other incorporates the data to pass through the function. A butterworth filter is supposed to provide a flat frequency response in the passband, a range of acceptable frequency [4] . 'Filtfilt', a built in python function that filters a signal without phase shift. The sinusoid load, ramp load, and ramp displacement data are all passed through this function.

The code maximizes the use of indexing to access data from the text file output by the Mach 1 Motion software. It loops through the data and creates a list called 'ramplist' to store the only names of the movements listed as 'Move Relative' or 'Sinusoid' in that order that they are executed. The code then stores the five columns of data under its respective ramplist in the following order: Time (seconds), Z, X, Y positions and load in the Z direction. Velocities calculated from the tension.py script are also stored in a list called 'vel'. The lists that store the five columns of data are further indexed into lists that include ramp load, ramp displacements, and ramp time for each of the eight move relative movements. Sinusoid load, displacement, and time are also indexed into their own list. All of these lists end up passing through the Butterworth filter functions to obtain the filtered data to be plotted and used for calculations.

Before data is plotted to view results, a function called 'Concatenate_time' is used to adjust the time elapsed. The Mach 1 software records the time continuously over the course of the tension_pc sequence, but in order to plot the results of each move relative the time needs to be adjusted for the second figure.

After the data is collected into lists it is ready to properly create plots to display the results of the test as well as force, stress and strain calculations. Load is measured in grams from the load cell and then converted to Newtons (*0.0098) for the stress calculation. As mentioned above, the filtered force values are divided by the cross sectional area to obtain stress values on the material. Strain is calculated by taking the change in length from the initial length and dividing by the initial length of the specimen. The code is run using Pycharm 2.7 to create several plots. The first figure shown below includes displacement as a function of time and load over time.

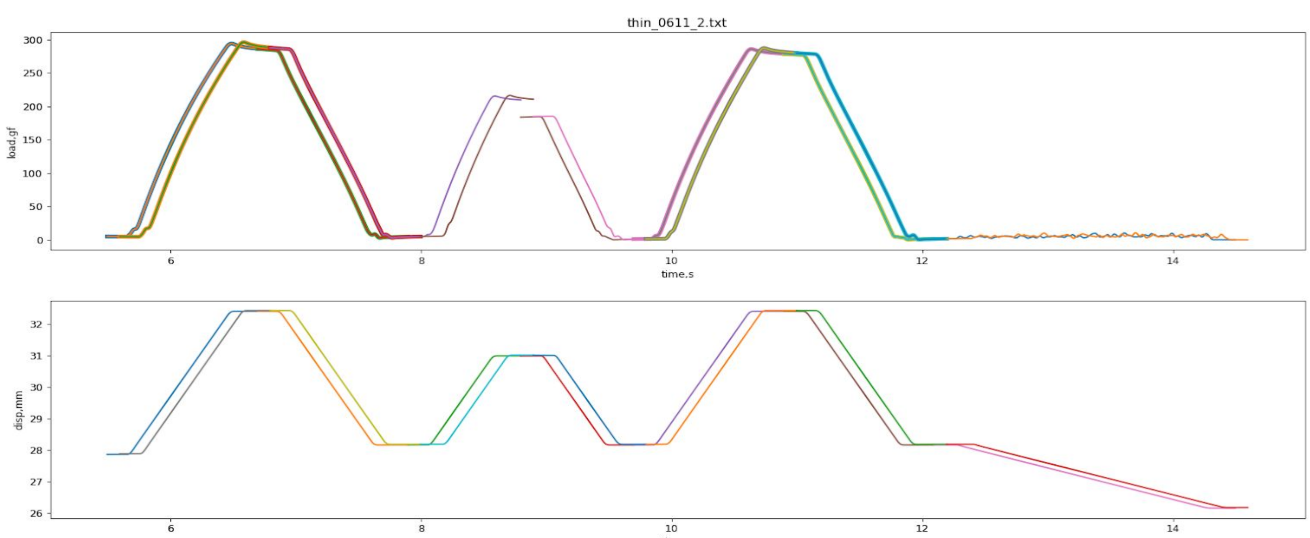


Figure 2. Displacement and load as a function of time for both tests.

Figure 3 can accommodate several tests to compare the results of each ramp of force for both specimens. The figure next to it is the stress-strain curve for each specimen that represents the second ramp (15%) on the material. It is important to note the similarity of these curves for an indication of repeatability among tests. The data may need to be adjusted if the force at zero load does not equal the displacement at zero load. One can fix this by manually reading the text file to search for the z displacement value that corresponds to the value that the load begins to register beyond the resolution of the load cell. The force at this point can be subtracted from all sequential force values to adjust the data accordingly.

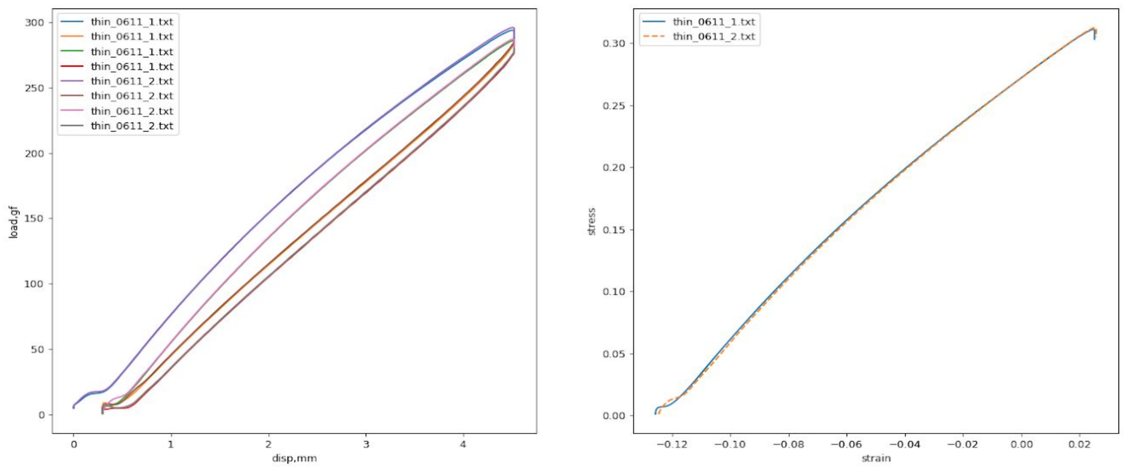


Figure 3. The figure on the left displays the ramps 15% its initial length and unloading. The figure on the right is the stress strain-curve.

The stress-strain curves next to it describe the relationship of stress as a function of strain on the material, particularly after preconditioning. The slope of curve's linear region defines the elastic modulus of the material, a very important parameter of modeling and simulation that describes its resistance to deformation. Samples represented on the curves below have already been tested and exhibit repeatable results. These plots required an adjustment of the data because they do not exhibit zero force at zero displacement and zero stress at zero strain. Negative strain indicates the sample is being compressed. The third figure produces displacement as a function of time and load as a function of time over the whole duration of the tensile test.

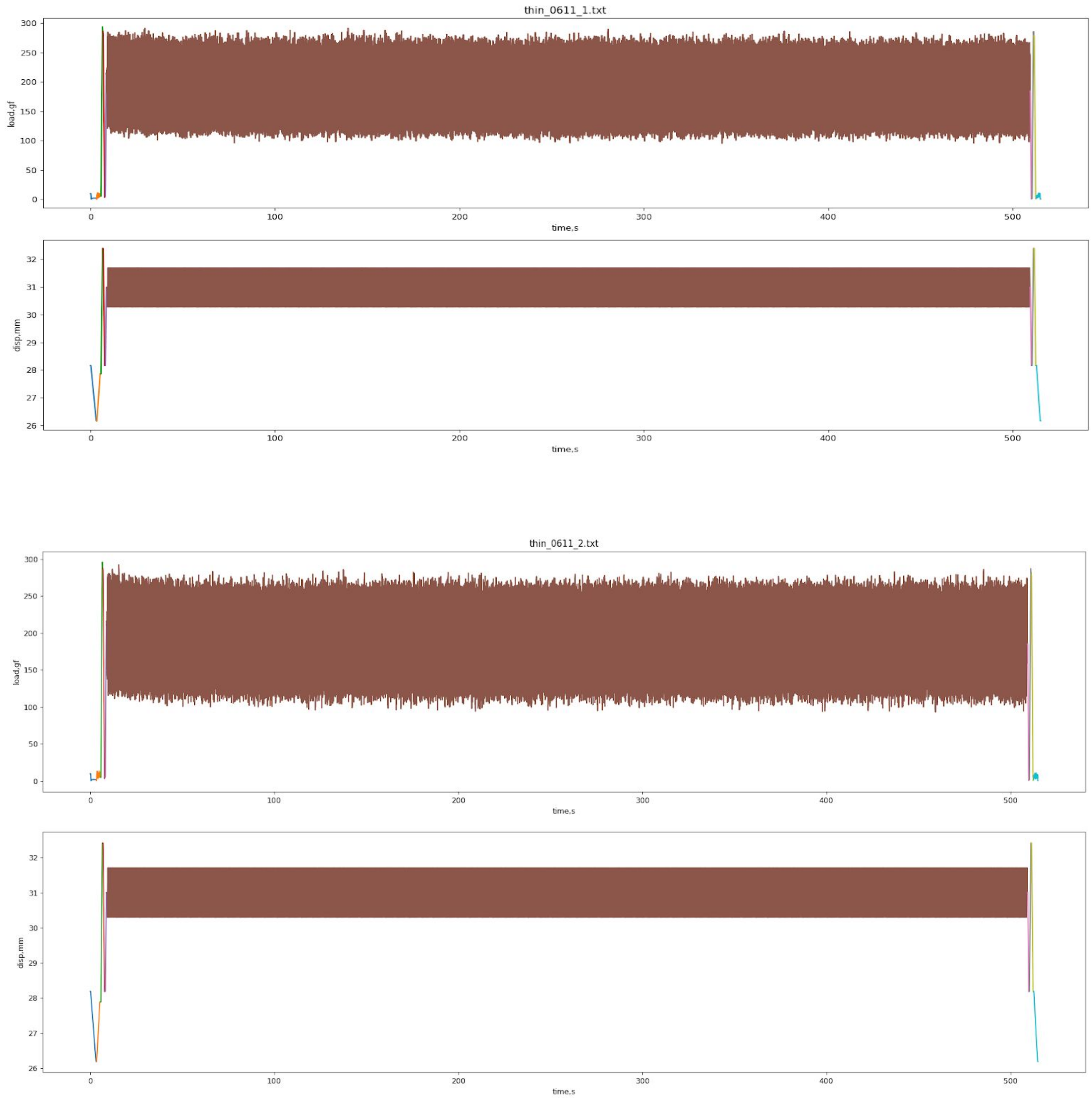
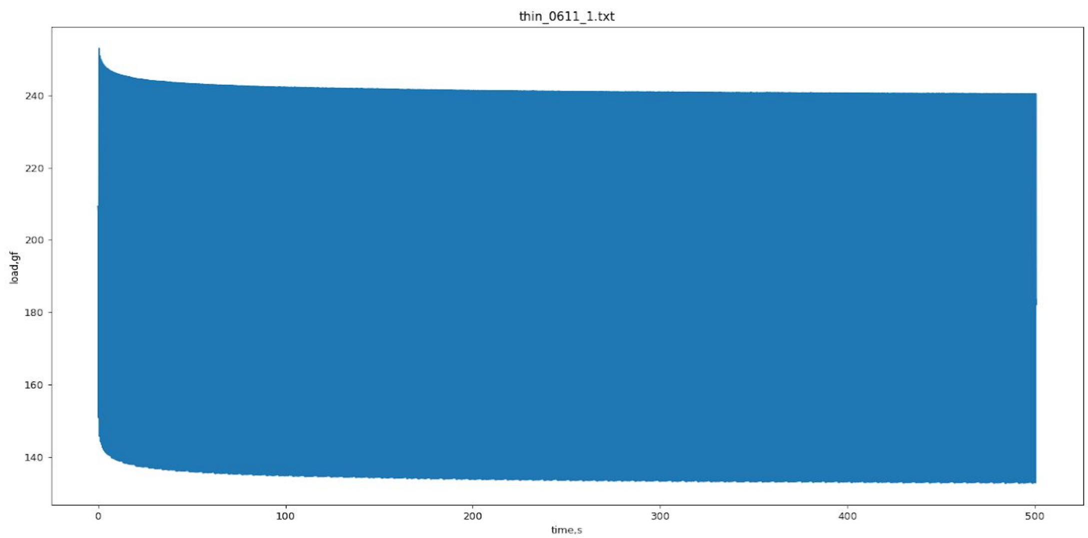


Figure 4. Displacement and load over time from test 1 (above) and test 2 (below).

Tissue check.py will create an individual figure for each sample tested, not overlapped as the previous figures. The last figure produces is filtered load over time during the preconditioning period, lasting 500 seconds. It indicates that the force on the specimen decreases over time and then stabilizes.



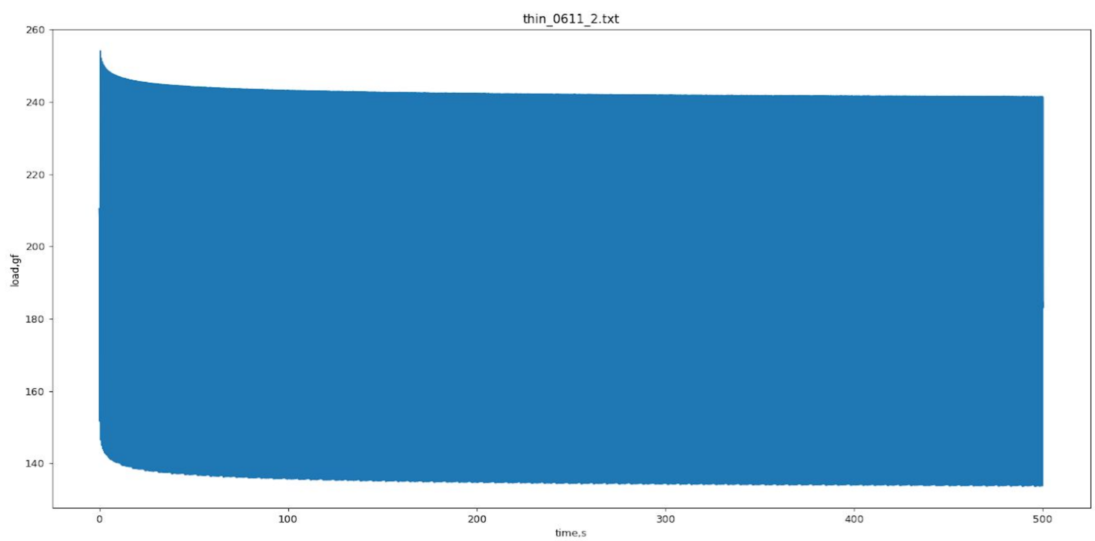


Figure 5. Preconditioning over time for test 1 (top) and test 2 (bottom).

Analysis

Mechanical testing is very important to understand how materials behave under certain loading conditions. Performing these tests to acquire accurate and repeatable results can be difficult. Temperature, humidity, time between tests, errors in measuring initial length and storage of the material can all have an effect on the results of the test. One must pay careful attention to these parameters in order to troubleshoot data discrepancies. Data on the plots can also be adjusted by manipulating the threshold parameter at the beginning of the code or by looking at the raw data to determine the adjustment manually. The ‘tissuecheck.py’ script can be improved by adjusting the curves automatically. Also, it should be able to produce results that compare the stress-strain curves numerically.

References

[1] “Catalog.” *Biomomentum*, biomomentum.com/catalog/.

[2] Chokhandre, S. *SimTK*.
https://simtk.org/svn/openknee/utl/TissueThickness/manual-thickness-measurement.py

[3] Chokhandre, S. *SimTK*. https://simtk.org/svn/multis/app/TissueTesting/Python/tissuecheck.py

[4] “Butterworth Filter.” *Wikipedia*, Wikimedia Foundation, 10 June 2019.