Buckling and Friction-Based Linear Motion Clutch with Application to Medical Devices

by

Erik K. Bassett

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

© Massachusetts Institute of Technology 2008. All rights reserved.

Author ....
Department of Mechanical Engineering
May 16, 2008

Certified by ....
Alexander H Slocum
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by ....
Lallit Anand
Chairman, Department Committee on Graduate Students

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

JUL 2 9 2008

ARCHIVES

Buckling and Friction-Based Linear Motion Clutch with Application to Medical Devices

by

Erik K. Bassett

Submitted to the Department of Mechanical Engineering
on May 16, 2008, in partial fulfillment of the
Requirements for the Degree of Master of Science in
Mechanical Engineering

Abstract

Improperly inserted and positioned needles and catheters often require repeated attempts at correct placement causing injury to adjacent structures or infusions into inappropriate spaces. Existing catheter insertion methods do not uniformly provide needle tip location feedback, nor prevent the needle from going beyond the target space. This research achieved the development of design methods and analysis tools that can be used to create a new catheter insertion device. This device can advance a needle through firm tissue but automatically stop advancing it upon entrance into a target space. Prototypes of the device were tested on raw chicken breast, the best of which had about a 50% success rate. Tests performed on deceased pigs showed the device advanced well through muscle but not the peritoneum. The system studied consisted of a flexible filament (OD ~0.9 mm) passing through a tube (ID ~1.2 mm) with both straight and curved sections. Initially it was believed that the capstan equation would provide a good model for the system in tension and compression. Though the capstan equation proved valid for the system under tension, models from drill strings used in the petroleum industry provided an accurate model for the system in compression. Based on the geometry of the tube, this model accurately predicts the compressive force in the filament and when the filament locks-up inside the tube (needle). An alternate method to measure the tube geometry using a flatbed scanner was developed and studied. This method was found to provide excellent accuracy and repeatability for measuring tubes, and has shown potential as a measurement method for many other applications.

Thesis Supervisor: Alexander H Slocum
Title: Pappalardo Professor of Mechanical Engineering

Acknowledgements

I would like to thank the following individuals for their roles in my research: Alex Slocum for his guidance and the opportunity to work with him; Omid Farokhzad for supplying the need for this research; Jeff Karp for his help connecting key people; the Deshpande Center for providing funding for the research; CIMIT for providing funds for completion of the research; Douglas Shook at Brigham and Women's Hospital for allowing me to view a procedure and give explanations of existing technology; Peter Masiakos at Massachusetts General Hospital (MGH) for his suggestions to improve device and expand potential market; Howard Pryor at MGH for performing and recording the prototype tests on pigs; my now wife, Britney, for supporting me in spite of my stress and staying up till all hours of the day and night helping me complete tasks; and last but not least my family for their encouragement, love, support, and help editing my documents.

Contents

List of Figures

List of Tables

List of Symbols

a_i- Inclination build rate (rad/m)
a_ϕ- Azimuth build rate (rad/m)
β- Buckling constant
$D_{filament}$- Filament diameter (m)
E- Modulus of elasticity (N/m^2)
f_b- Buoyancy factor
F_{1c}- Generalized contact force per unit length (N/m)
F_{1hbc}- Contact force per unit length of helically buckled filament (N/m)
F_{1nbc}- Contact force per unit length of non-buckled filament (N/m)
F_a- Input force (N)
$F_{contact}$- Predicted contact force (N)
F_{icr}- Critical buckling load (N)
F_{in}- Input force (N)
F_{out}- Output force (N)
$F_{out\_predicted}$- Predicted output force (N)
I- Moment of inertia of filament (m^4)
$L_{straight}$- Length of long straight section of design tube (m)
L_{curved}- Length of curved section of design tube (m)
mg- Weight per unit length of filament (N/m)
μ- Coefficient of friction between filament and tube wall
r- Radial gap spacing between filament and wall (m)
R- Average radius of curves in design tubes (m)
θ- Angular displacement of design tubes (rad)

Chapter 1

Introduction

Medical procedures often require placement of arterial, central venous, and epidural catheters. Misdirected insertion of needles and catheters can result in complications including discomfort from repeated catheterization attempts, bleeding, post-dural puncture headache, nerve injury, infusion of medication into inappropriate tissues, hemodynamic shock, and respiratory arrest [1]. A needle system that detects different tissues or tissue compartments could reduce these complications in placing central venous catheters, arterial lines, and epidural catheters. This work focused on creating a mechanical clutch system that advances a needle through more resistant tissues and then automatically releases it when the tip enters the desired soft tissue or cavity.

This clutch is based on the shape dependent sliding resistance of a flexible filament passing through a tube. At one extreme is a short, straight tube. A filament going through such a tube feels little resistance and is easily advanced. At the other extreme is a tube with many tight bends or loops. A filament going through this system only advances to a point before it begins to bind inside the tube and 'lock-up'. Between these extremes, both sliding and locking by adjusting the exit resistance for a single tube shape can be achieved. This thesis presents the results of testing and modeling the clutch system, the development and testing of a prototype device, and an improved method for quickly and accurately measuring the shapes of the tubes using a flatbed scanner.

1.1 Capstan Analysis

The Capstan equation has proved very valuable to predict the response of cable actuators and systems [2]. The first step of this research involved determining whether the capstan equation (Equation 1 and Figure 1-1) modeled the tensile and compressive forces in a thin filament passing through a tube with bends (Figure 1-2). If successful, it would indicate whether the proposed clutch action could be predicted and used in designing clutch systems. The primary tests for the study measured tensile and compressive loads to a filament passing through tubes of varying shapes.

$$F_{in} = F_{out} * e^{(\theta*\mu)} \tag{1}$$

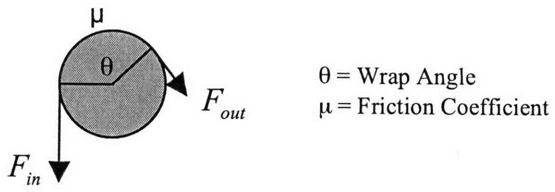


Figure 1-1: Capstan effect

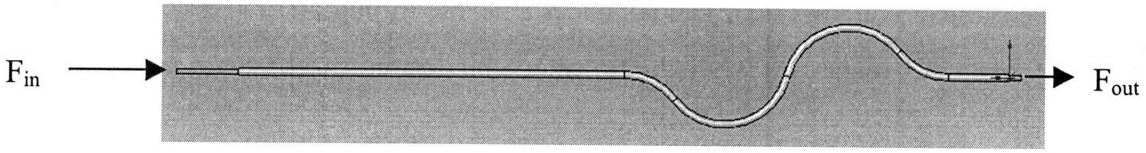


Figure 1-2: Flexible filament (green) passing through example design tube to exploit the capstan effect

The capstan equation proved to be a good model for the filament in tension but not compression. Sections 3.3.3 and 3.4.2 respectively show the analysis performed and results for each case. Other models were the researched to identify a better fit for the compression force findings.

1.2 Petroleum Industry Drill String Analysis

Drill strings, used in the oil industry, undergo similar loading and buckling as this system, but on a larger scale. For this reason drill string models were researched and analyzed. When studying a long slender object under compression, it is important to determine the point at which it will buckle, and how much force will be absorbed due to increased contact with the tube wall. The earliest model to comprehensively describe buckling forces in drill string systems was developed by Lubinski *et al.* [3] in the 1960's. In that study he derived the first model to predict helical buckling of a drill string in a vertical well. Mitchell *et al.* [4] further developed a simple helical buckling model that included friction. The effects of curvature and friction on buckling were studied by McCann *et al.* [5]. He *et al.* [6] studied helical buckling and lock-up conditions of coiled tubing in curved wells. Qiu *et al.* [7] developed models for sinusoidal and helical buckling in vertical and inclined straight as well as curved holes. The buckling behaviors of pipes and the resulting influence on axial force transfer was studied by Kuru *et al.* [8].

The models developed by He *et al.* and Qiu *et al.* most closely represented the system studied and were used to successfully predict compressive output forces as shown in section 3.4.3. These models include the geometry of the tube and properties of the drill string, making them comprehensive.

Equations 2-5 are from He *et al.* and were used in this study.

$$F_{icr} = \left[\frac{\beta F_{1nbc} EI}{r} \right]^{.5} \tag{2}$$

$$F_{1nbc} = \left[\left(f_b mg \sin\theta + F_a a_i \right)^{2} + \left(F_a a_\phi \sin\theta \right)^{2} \right] \tag{3}$$

$$F_{1hbc} = \frac{F_a^{\,2} r}{4EI} \tag{4}$$

$$F_{1c} = \left\langle \begin{array}{l} F_{1nbc} \; for \; F_a < F_{icr} \\ F_{1hbc} \; for \; F_a > F_{icr} \end{array} \right. \tag{5}$$

Where F_a is the axial input force to the top of the drill string, F_{icr} is the critical buckling force where $\beta = 4$ and 8 for sinusoidal and helical buckling respectively, F_{1nbc} is the contact force per unit length for a non-buckled configuration, F_{1hbc} is the contact force per unit length for a helically buckled string, and F_{1c} is the generalized contact force per unit length of the string. See section 3.4.3 for complete definition of equations.

Chapter 2

Design Parameters

For this study the design parameters chosen were based on the primary variables in the capstan equation. These three parameters included the number of bends (Figure 2-1), the angular displacement (Figure 2-2), and the average bend radius. Eight different tube designs were created (each 17.8 cm long) and used in both the tension and compression tests. Table 2-1 shows the values of each parameter for the designs tested. The tubing used for the designs was stainless steel with an OD 1.47 mm and ID 1.22 mm. Two different flexible filaments were used: nylon tubing (OD 0.90 mm, ID 0.47 mm) and PTFE monofilament (OD 0.97 mm). Figure 2-3 show the differences in size and geometry of the filament inside the tubing while Table 2-2 shows the values of E, I and E*I for each.

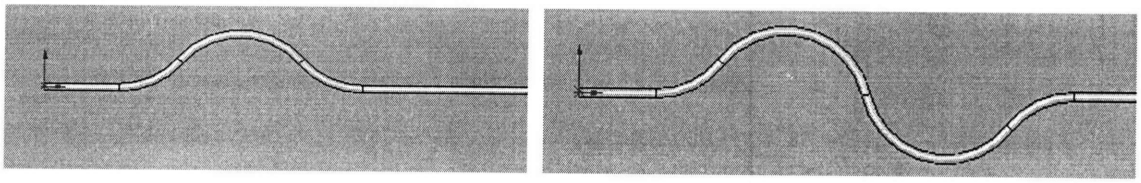


Figure 2-1: Example of tube designs with one and two bends

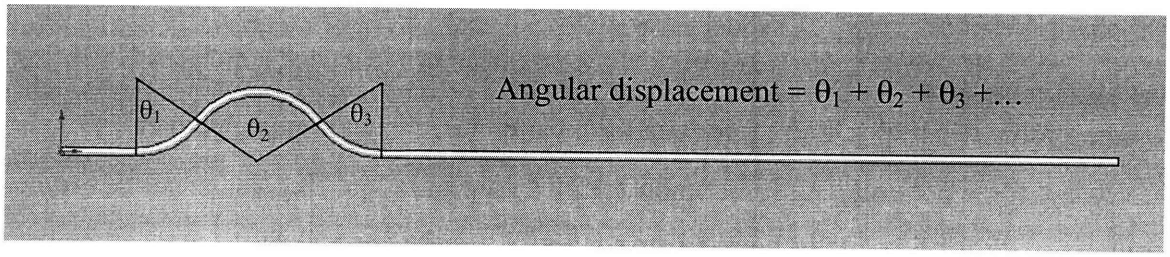
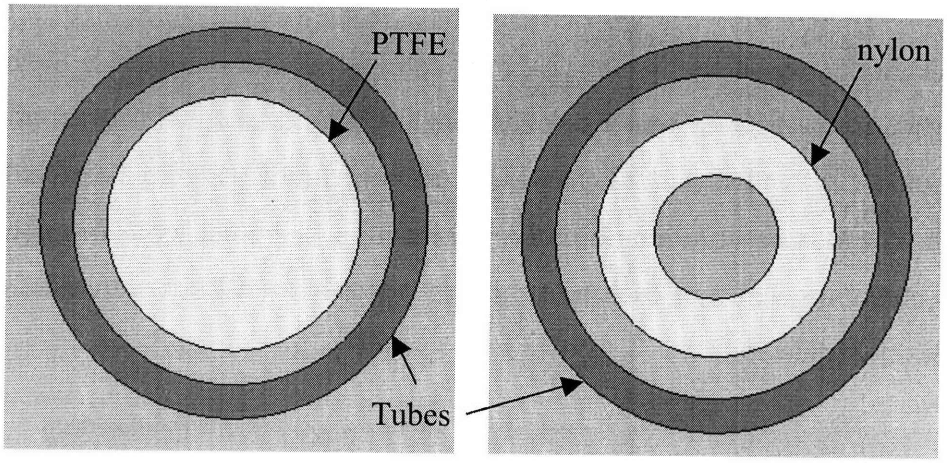

θ_1
θ_2
θ_3
Angular displacement = $\theta_1 + \theta_2 + \theta_3 + \ldots$

Figure 2-2: Angular displacement description and equation

Table 2-1: Design parameter values of tube designs tested

Design	# of Bends	Avg. Bend Radius (mm)	Angular displacement (rad)
1A	1	24.7	1.57
1B	1	29.7	1.73
1C	1	27.3	1.94
1D	1	31.6	1.99
2A	2	26.0	1.95
2B	2	34.7	1.83
2C	2	28.3	2.09
2D	2	32.0	2.22



(a) PTFE Monofilament

(b) Nylon Catheter Tube

Figure 2-3: Cross section views of flexible filaments inside tubes

Table 2-2: Values of E, I and E*I for nylon and PTFE

Material	I (mm^4)	E (Mpa)	E*I ($N\text{-}m^2$)
PTFE	4.26 E-02	73	3.11E-06
Nylon	3.01 E-02	172	5.17E-06

*Determined from tensile test performed in this research (see Modulus of Elasticity section)

2.1 Manufacturing of Test Parts

Plastic guides made of 3.2 mm thick acrylic sheet were constructed for bending the design tubes. The guides were cut using an OMAX CNC Waterjet Machining Center (www.omax.com). The shapes of the guides were chosen based on prior bench level experiments where the nylon was pushed through test guides. The guide shapes were defined by the angular displacement, bend radius, and number of bends. The values of each parameter were chosen to make a two level full factorial design. Table 2-3 shows the range of parameters used in the guides.

Table 2-3: Design parameter values of acrylic guides

	Guide Design			
Design	# of Bends	Bend Radius (mm)	Angle (rad)	Bend Amplitude (mm)
1A	1	14.0	2.79	6.6
1B	1	17.8	2.79	8.4
1C	1	14.0	3.49	9.9
1D	1	17.8	3.49	12.7
2A	2	14.0	2.79	3.6
2B	2	17.8	2.79	4.6
2C	2	14.0	3.49	4.6
2D	2	17.8	3.49	5.8

Each guide was made with a 12.7 mm straight segment on one side the bend. On the other side of the bend(s) was a second straight section that filled the rest of the 10.2 cm long guide. Each guide was 5.2 cm wide to provide rigidity (Figure 2-4). Before being bent, each tube was cut to a length of 17.8 cm using a dremmel tool. To minimize the effects of burs on the ends of each tube, honing stones and drill bits were used to round the outside and inside corners.

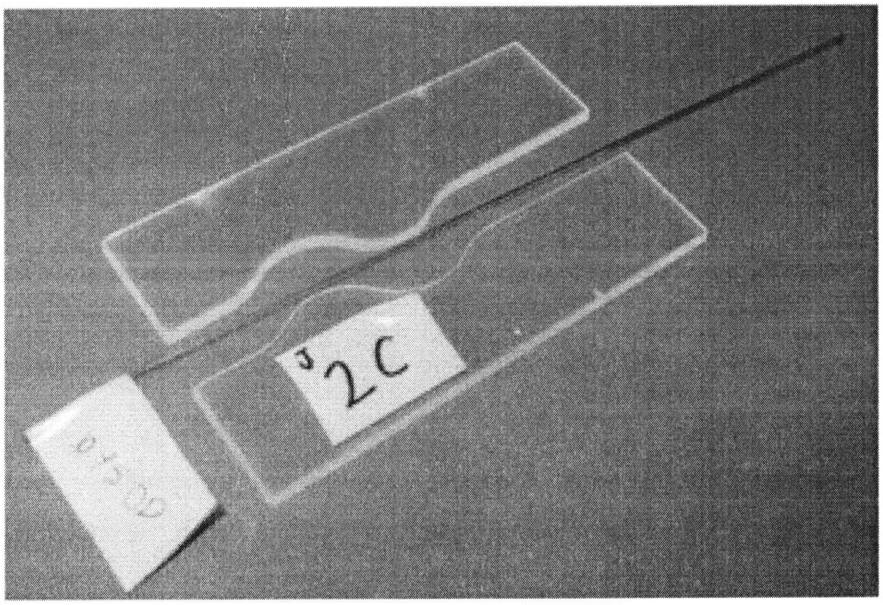


Figure 2-4: Acrylic guide for bending tube 2C

To prevent kinking or collapsing of the tube during the bending process, a PTFE filament (OD 1.14 mm) was placed inside. The tubes were bent by placing the two guides on a flat surface with the tube in between. A third piece of acrylic was placed on top (Figure 2-5 & Figure 2-6). All three guide pieces were secured to prevent the tube from displacing off the surface (Figure 2-7). The tube was positioned between the guides so that in the post-bent shape, one end would correspond with the edge of the guide (see Figure 2-4), resulting in a straight section ~13 mm long (see Figure 2-2). The two guides were then pressed together with a vise. Due to spring back of the metal, the resulting shapes were different from the guides and were not always coaxial. The tubes were then adjusted by hand so the two straight sections were as coaxial as possible. The true shape of the tubes then had to be measured to extract exact angles and lengths.

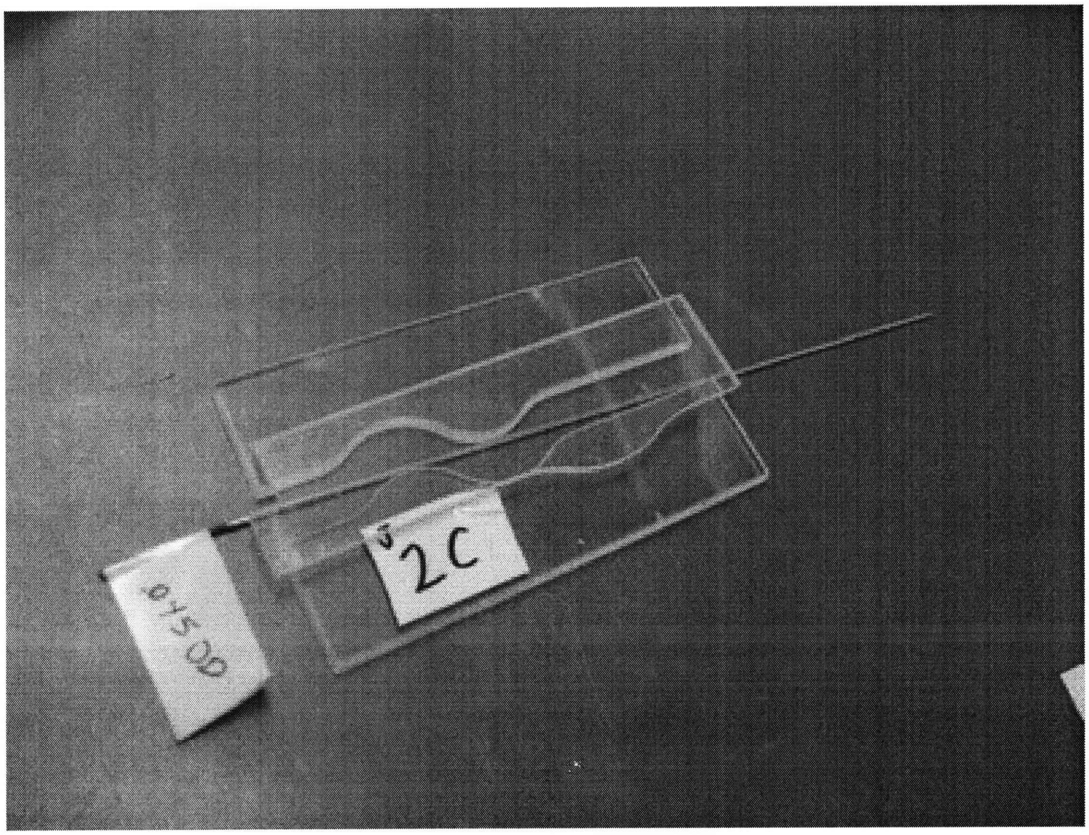


Figure 2-5: Tube guide and tube with additional acrylic piece on top

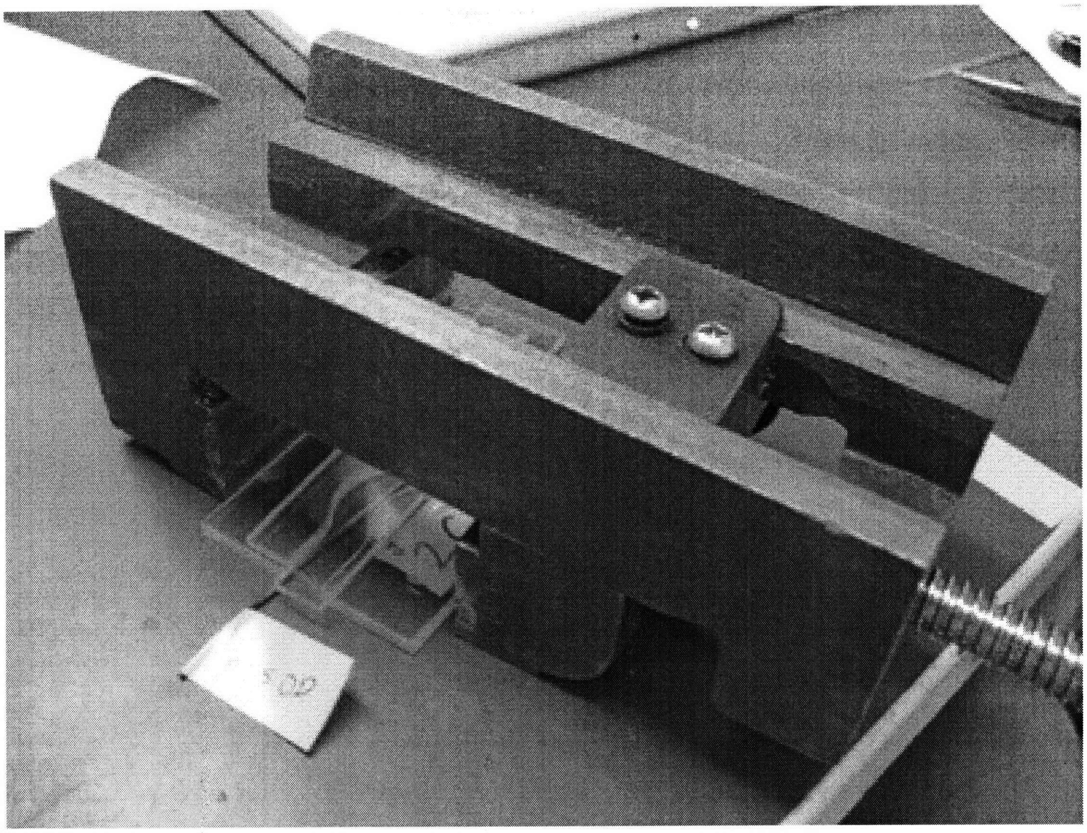


Figure 2-6: Tube guide and tube in vise prior to bending

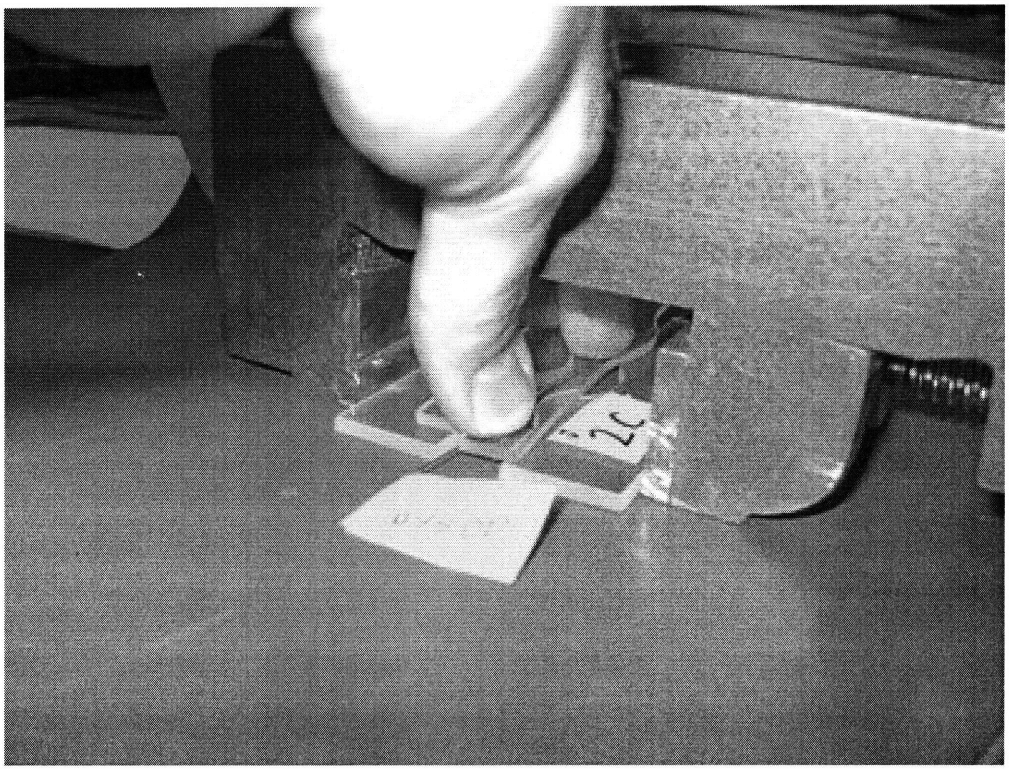


Figure 2-7: Tube and acrylic held down while being bent

Later, larger tubes were made with more bends, requiring larger holding forces. Instead of holding the tube in place vertically by hand, aluminum L-extrusions were bolted to a table and provided the necessary forces. The guides and tubes were placed between the aluminum and the tabletop as in Figure 2-8.

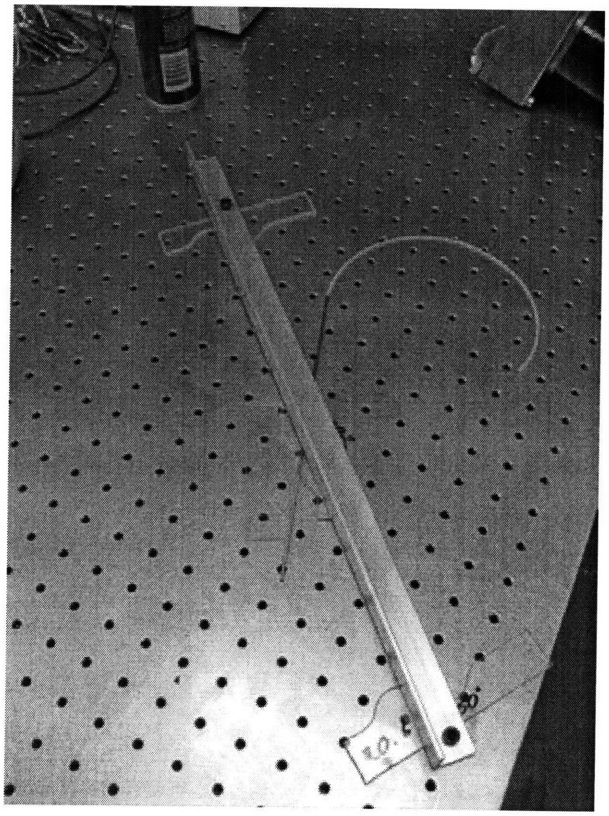


Figure 2-8: Improved method for containing the tubes vertically during bending

2.2 CMM Measurement of the As-Formed Tubes

The true final shape of each formed tube design was found using a coordinate measuring machine (CMM). Each tube was measured approximately every 1 mm along its entire length and the *x* and *y* locations recorded. These points were used to calculate the angular displacement and average radius of curvature of each tube design. Because the filaments had smaller diameters than the tube, the tensioned filaments transitioned from side to side thus passing through less of a angular displacement than the tube (Figure 2-9 & Figure 2-10). Thus, the tube provided an upper limit for the angular displacement while the lower limit could be calculated based on the diameter of each filament. This outer diameter difference between the nylon and PTFE caused there to be a difference in the minimum angular displacement of each filament inside the tube. Table 2-4 shows these max and min angular displacement values for each design. The average bend radius was calculated by dividing the total arc length of the bends by the angular displacement (Figure 2-11).

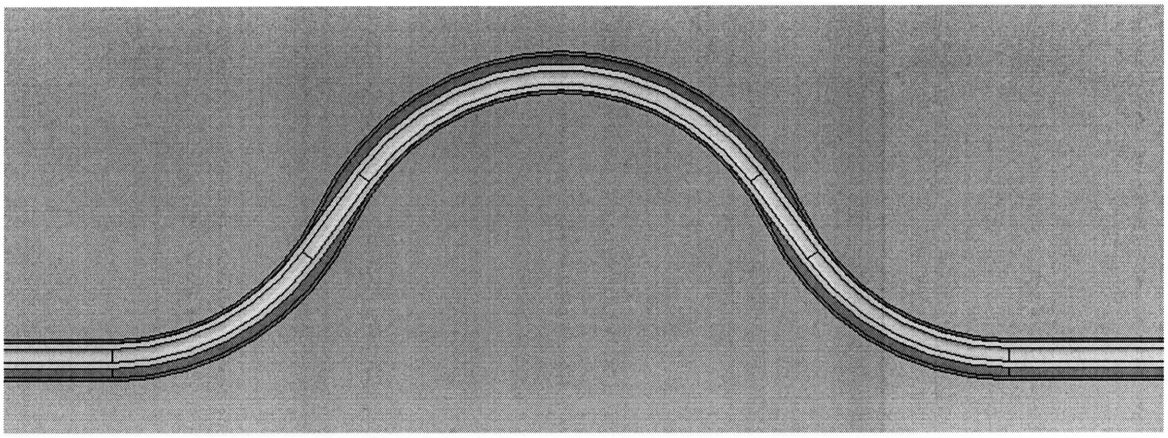


Figure 2-9: Nylon catheter tube path inside tube showing reduced angular displacement

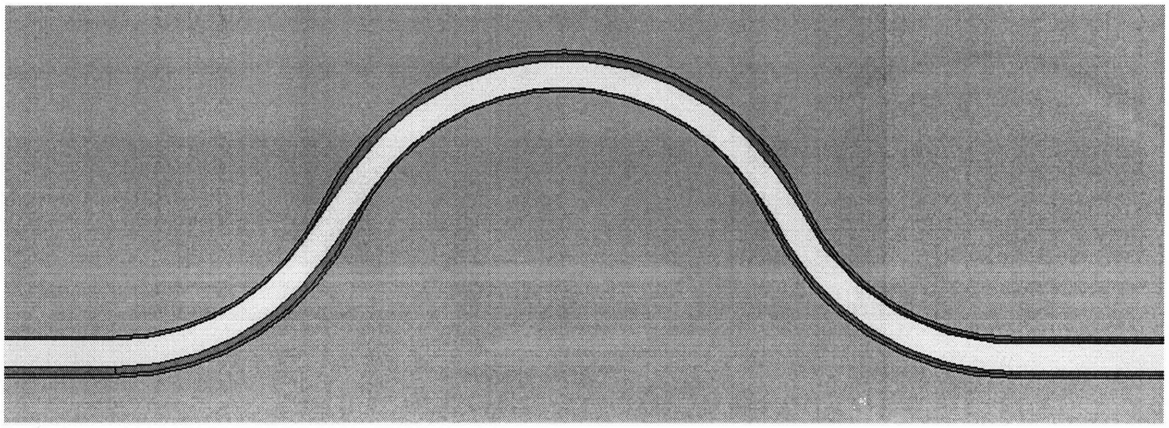


Figure 2-10: PTFE monofilament path inside tube showing reduced angular displacement

Table 2-4 Calculated max and min angular displacements for nylon and PTFE in each design

Design	Max Angle (Rad)	Min Nylon Angle (Rad)	Min PTFE Angle (Rad)
1A	1.567	1.265	1.309
1B	1.730	1.416	1.454
1C	1.934	1.710	1.741
1D	1.991	1.786	1.820
2A	1.947	1.497	1.574
2B	1.829	1.494	1.550
2C	2.092	1.728	1.787
2D	2.219	1.801	1.876

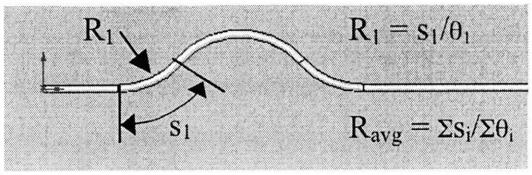


Figure 2-11: Average bend radius calculation method

Chapter 3

Experiments

All experiments were performed using an ADMET 5601 testing machine with one or two Interface SMT1-2.2 load cells (capacity 9.8 N). The ADMET machine included data acquisition software (Mtestw version 9.0.6) and CNC controls for the actuator head. Prior to testing, all critical components (i.e. tubes, filament, etc.) were cleaned with alcohol to remove any oil and moisture. Before the design tube was placed in the setup for testing the load cells were calibrated to ensure consistent force readings. During all tests, powder-free surgical gloves were worn to minimize surface contamination. Between each tension and compression test the design tube was removed and both the filament and tube were dried with compressed nitrogen gas. A similar procedure was conducted between friction tests.

3.1 Steel Tube-Filament Friction Coefficients

The friction coefficients between the filaments and tubing were obtained by performing a pull test. A test specimen was pulled horizontally along parallel filaments while measuring the pulling force (Figure 3-1). These tests were performed at rates of 0.13, 0.38, and 2.54 mm/sec and repeated three times for both nylon and PTFE filaments. The filaments were tensioned and then clamped between the main plate and two blocks with two parallel channels (0.8 mm wide, 12.7 mm apart, and 0.6 mm deep). A tubing holder was created by fixing sections of stainless steel tubing in two parallel channels (1.6 mm wide, 12.7 mm apart, 0.9 mm deep) cut in a steel block. The tubing was then ground in half exposing the inside surface (Figure 3-2). To minimize unwanted forces when pulling the block, it was pulled using a short section of nylon fishing line that was adhered to it. For the tests, the block rested on the filaments so the only contact occurred between the inside of the tubes and the filament. The clamps holding the filament ends inhibited pulling fishing line, so an L-bracket was used to reach around the clamps (visible in Figure 3-1). Figure 3-3 shows a typical curve for this test.

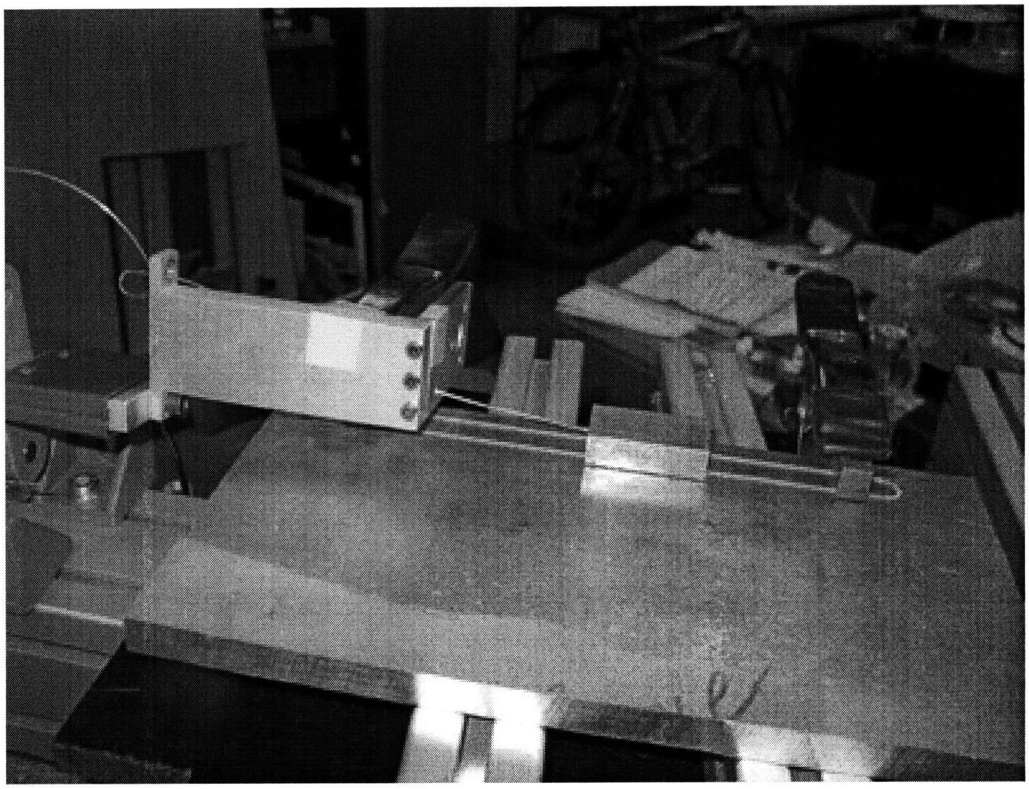


Figure 3-1: Friction coefficient measurement test setup

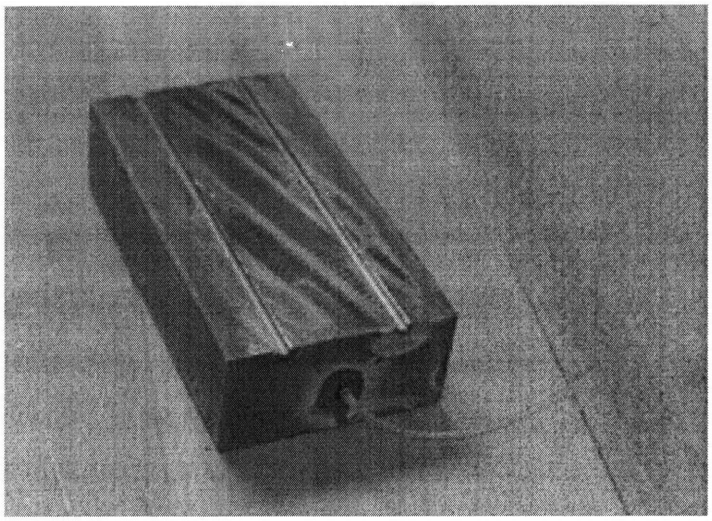


Figure 3-2: Friction test specimen with tubes glued in place

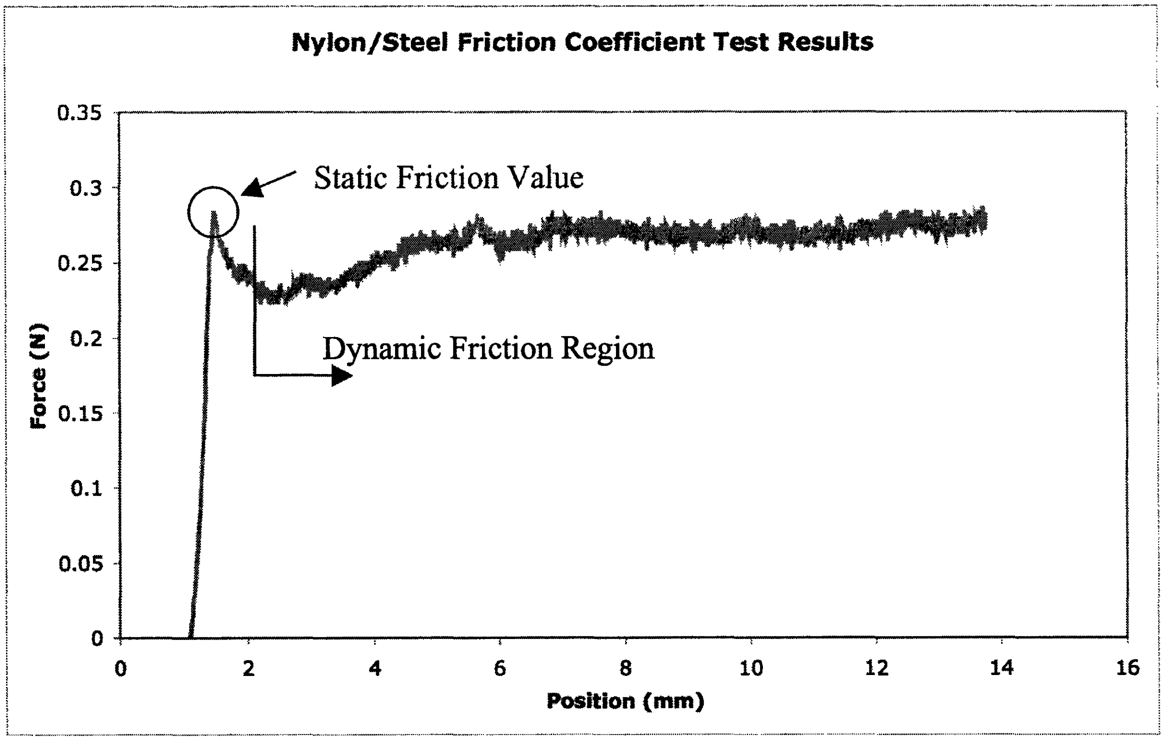


Figure 3-3: Typical friction coefficient test result for nylon and steel tubing

In these tests, the spike at the beginning corresponds to the static friction value (circled in Figure 3-3) and is followed by the dynamic region. The static friction coefficients measured are shown in Table 3-1.

Table 3-1 Friction coefficients for PTFE and nylon at different rates

	Measured Static Friction Coefficient	
Rate (mm/sec)	Nylon	PTFE
0.13	0.204	0.088
0.38	0.249	0.089
2.54	0.269	0.111

3.2 Modulus of Elasticity

Tension tests were performed on each filament to obtain the corresponding modulus of elasticity. They were achieved by bonding both ends of a ~23 cm filament section into hollow all-thread (Figure 3-4). The resulting test lengths were 17.5 cm. The filaments were then stretched at a rate of 0.25 mm/sec and a stress-strain curve generated (Figure 3-5 & Figure 3-6). From this, the modulus of elasticity was determined to be 172 and 73 MPa for the nylon and PTFE respectively. The epoxy bond failed while testing the

PTFE causing plateaus and inconsistent results in the latter part of the test. However, the failure occurred late enough that it was determined not to affect the results of the modulus calculation.

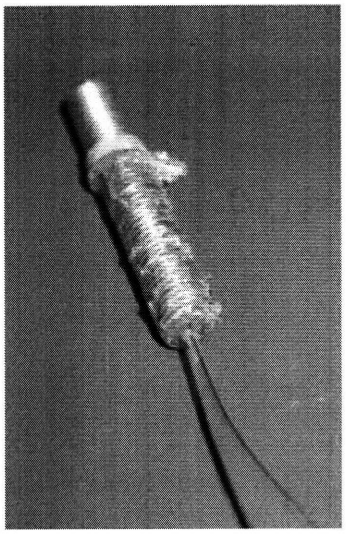


Figure 3-4: Nylon filament fixed with epoxy into threaded steel rod for modulus of elasticity test

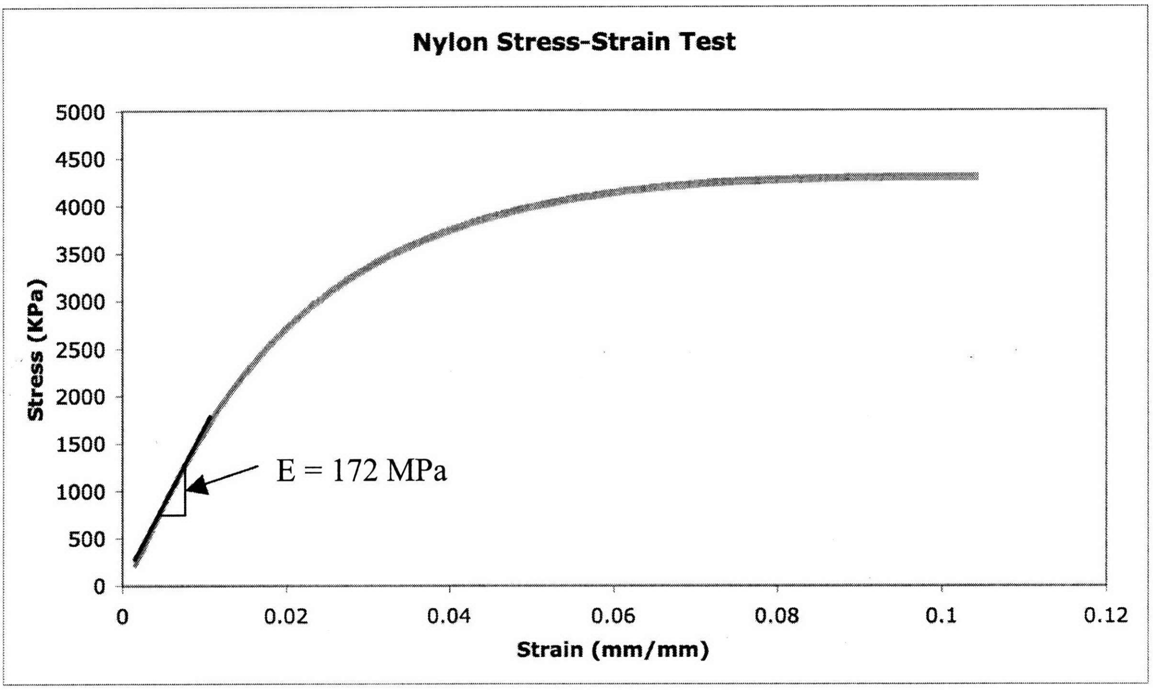


Figure 3-5: Stress-strain curve of nylon filament

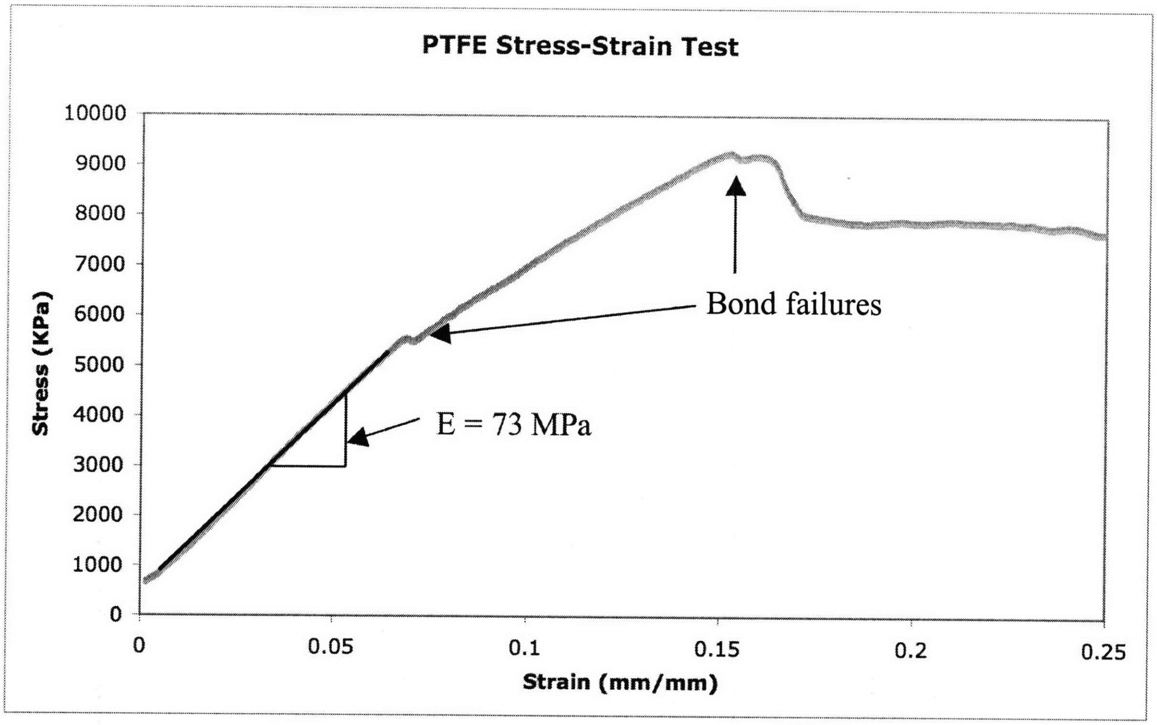


Figure 3-6: Stress-strain curve of PTFE filament

3.3 Tension Tests of Filament Inside Tubes with Curves

The tension test setup consisted of a computer controlled linear actuator and a bracket mounted on a frame (Figure 3-7). Attached to the bracket was a pin vise that held the design tube. A load cell was connected to the actuator above and to a second pin vise below. This vise held a different tube containing the filament (Figure 3-8). This tube (ID 1.60 mm) was oversized in order to slide easily over the design tube and was pinched closed at the top to hold the end of the filament. The load cell recorded the force in the upper end of the filament during the tests. A weight hanging from the bottom of the filament provided a constant tensile force (1.53 N). In this way the filament was always in tension but forces felt on the upper end depended on the direction of travel. For each test the actuator moved up and then down 10.2 mm at a rate of 0.38 mm/s. At the beginning of each test, the weight rested on a surface and then was raised so it hung freely before being returned at the same rate. Due to the method used to attach the weight to the filament, some tests began and ended with tensile forces greater than zero. Each design tube was tested four times. Figure 3-9 shows how the design tube passed through the pin vise and bracket in the setup. Figure 3-10 shows the different positions of the weight during the tension tests.

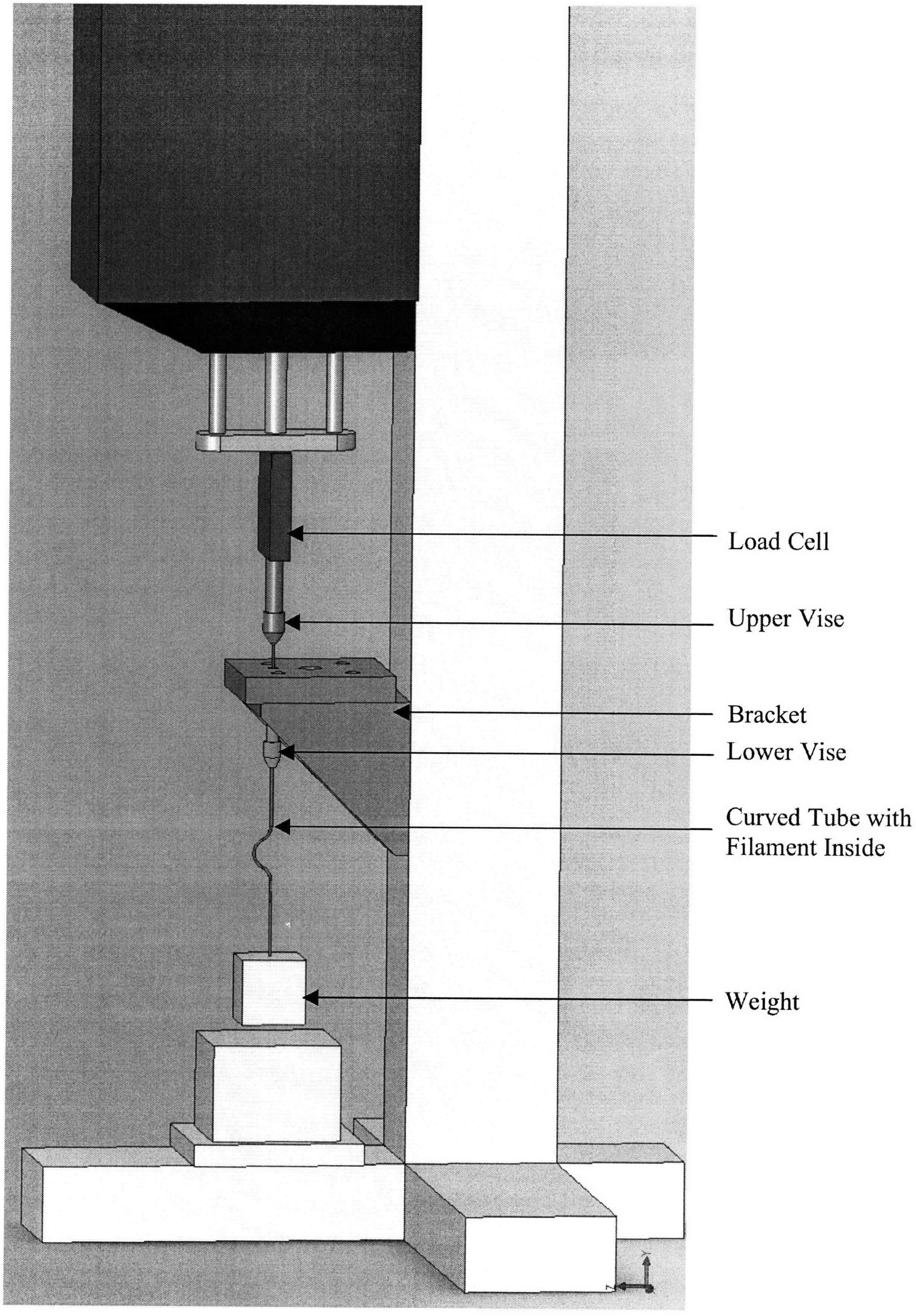


Figure 3-7: Tension test system

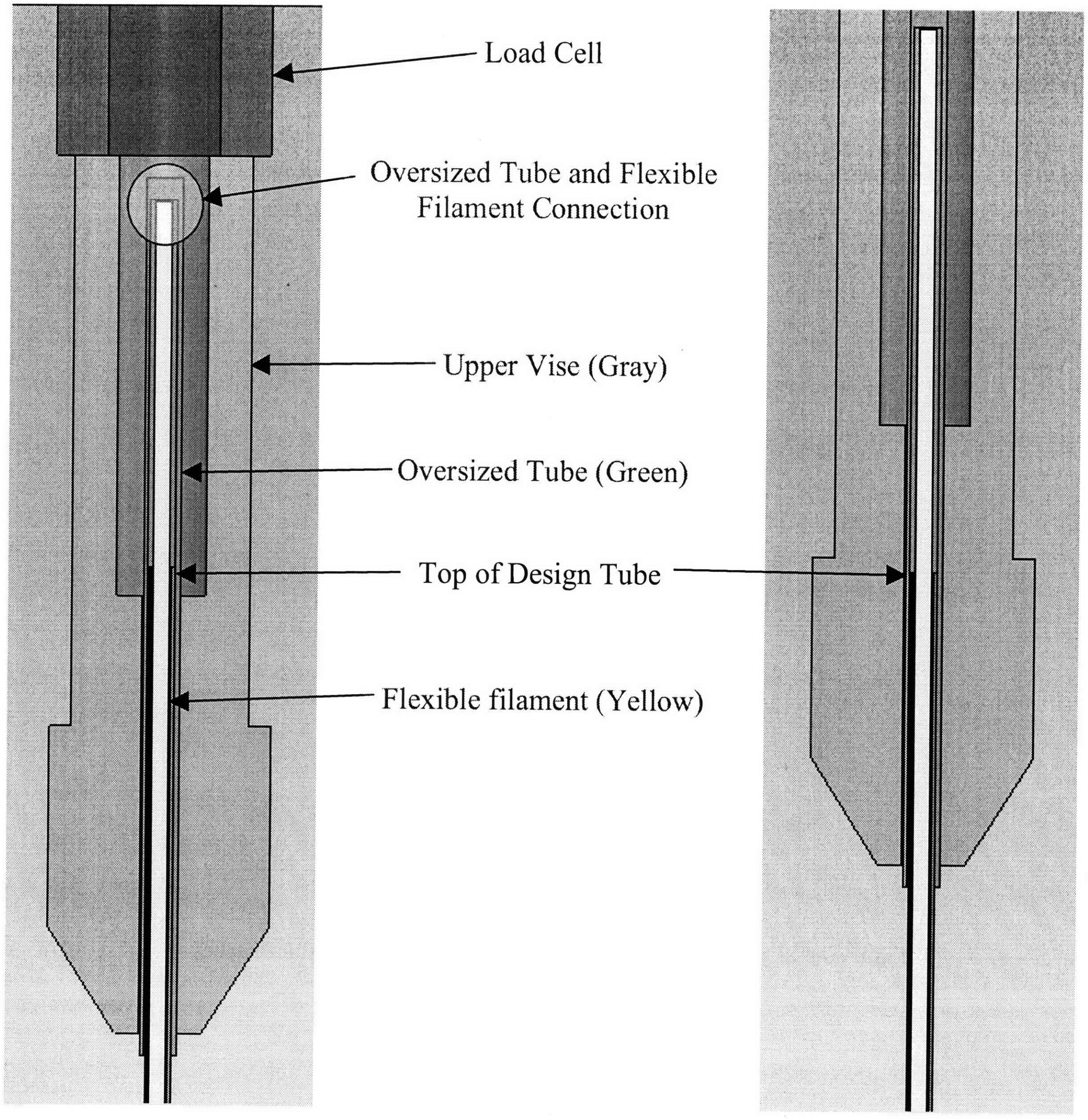

Load Cell
Oversized Tube and Flexible Filament Connection
Upper Vise (Gray)
Oversized Tube (Green)
Top of Design Tube
Flexible filament (Yellow)

Figure 3-8: Cross-section view of assembly in low (left) and high (right) positions

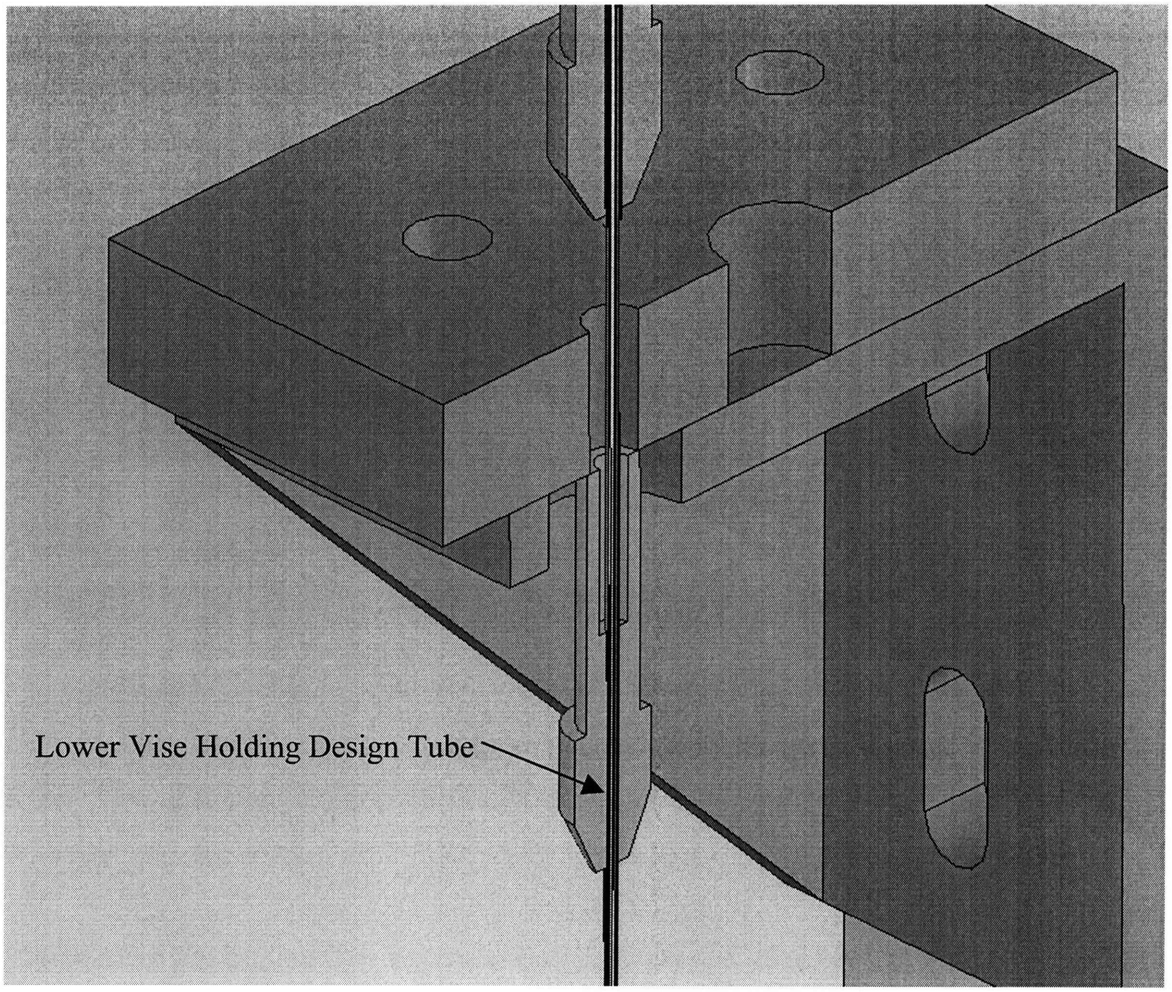
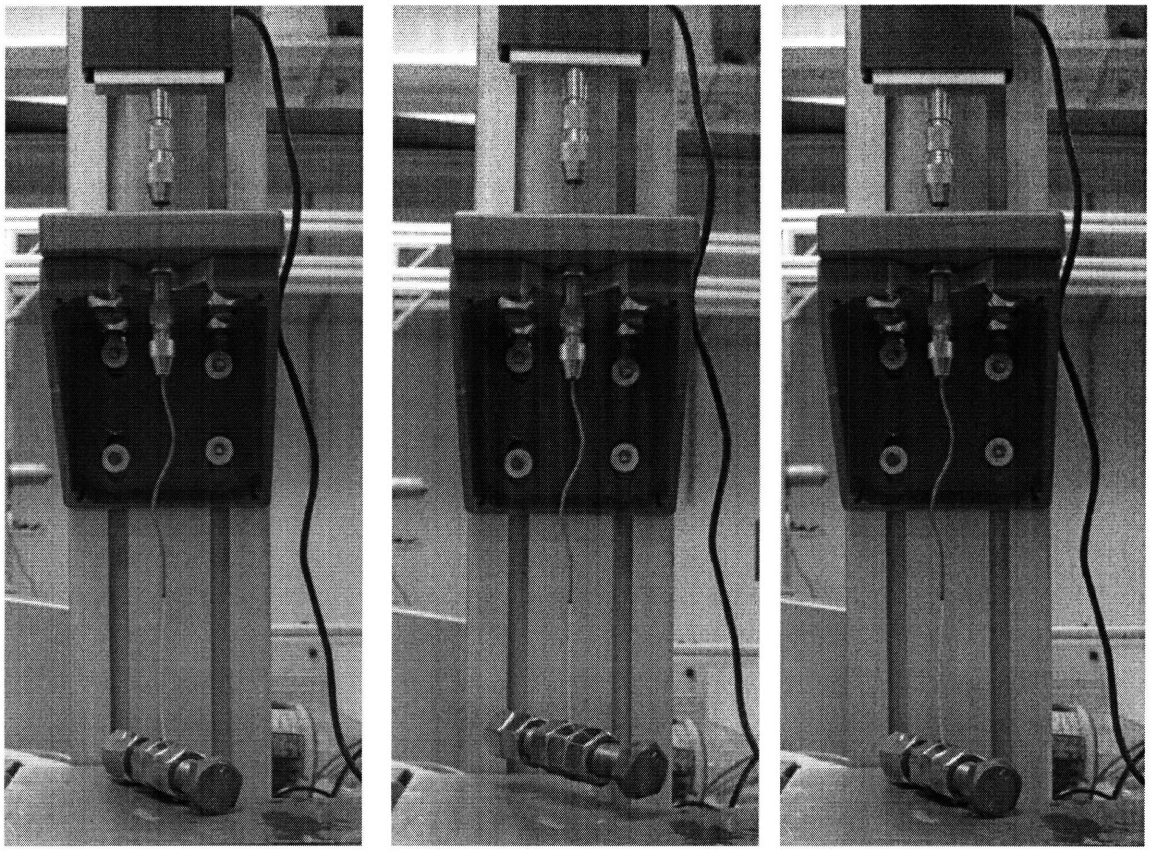


Figure 3-9: Cross-section of tube passing through vises and bracket



(a) Starting Position **(b) Highest Position** **(c) Ending Position**

Figure 3-10: Travel of load cell and weight during tension test

Results from a typical tension test can be seen in Figure 3-11. The two plateaus are the regions of interest for each test. While the weight is being raised, the upper part of the filament acts as the capstan pulling force (F_{in}) and the lower part acts as the capstan holding force (F_{out}) (See Figure 1-1). However, when the weight is lowered, the roles reverse. In this way, two unique capstan conditions can be analyzed from the same test. Mean values of the plateaus were found and averaged across the replicate tests for each design (Table 3-2).

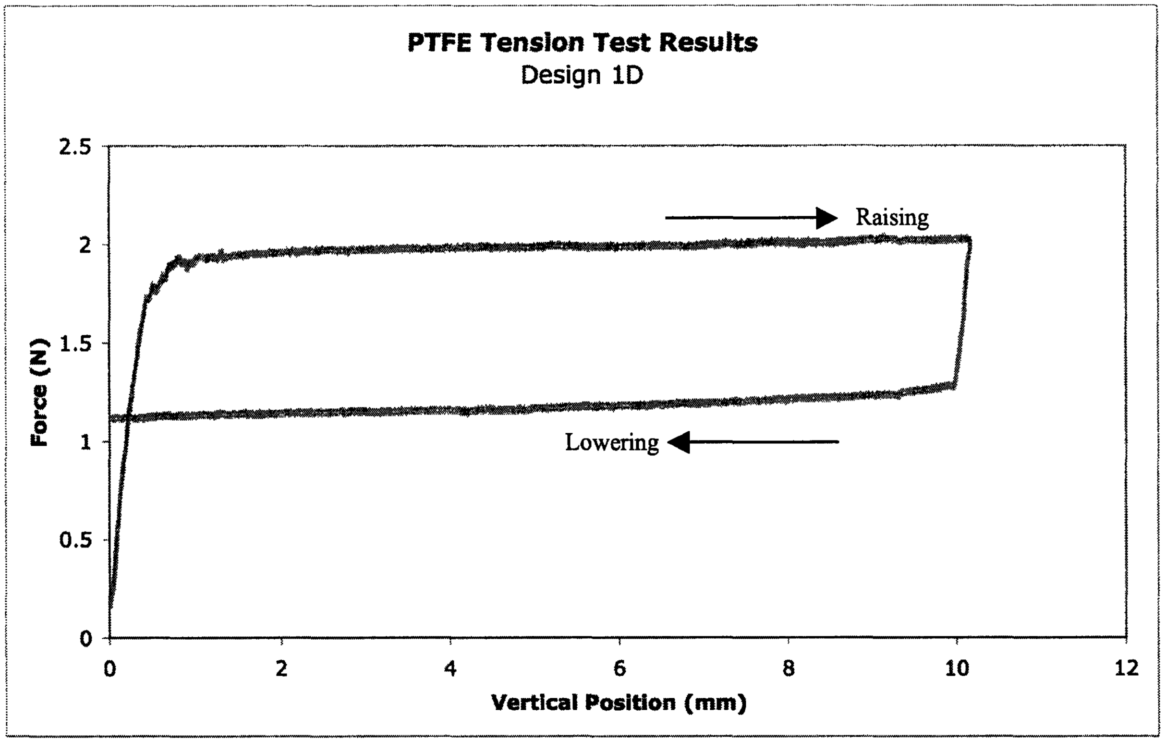


Figure 3-11: Typical PTFE tension test result

Table 3-2: Mean force values for tension tests

	Mean Tension Forces (N) - Nylon		Mean Tension Forces (N) - PTFE	
Design	Raising	Lowering	Raising	Lowering
1A	1.916	1.196	2.139	0.966
1B	1.913	1.204	2.056	0.809
1C	2.056	1.099	2.102	0.774
1D	2.034	1.139	2.043	0.685
2A	1.943	1.189	2.045	0.589
2B	1.993	1.149	2.001	0.623
2C	2.042	1.143	2.144	0.862
2D	1.885	1.228	1.923	0.531

3.3.1 Nested Steel Tube-on-Tube Friction

Analysis of tension tests showed that the capstan predictions did not match the measured tension values. This and other tests were performed to determine the cause of the difference. One potential source of error was the friction between the design tube and nested oversized tube used to hold the end of the filament (Figure 3-12) through which the force on the filament was applied. To measure this, the filament was removed from the setup and the oversized tube was moved up and down over the design tube. This was done at rates of 0.13, 0.38, and 2.54 mm/sec and repeated three times. These tests showed a relatively constant sliding force between the nested steel tubes. For the analysis, the last 2.5 mm of data points (circled regions in Figure 3-13) were averaged. Table 3-3 shows the averages calculated from these tests for each rate.

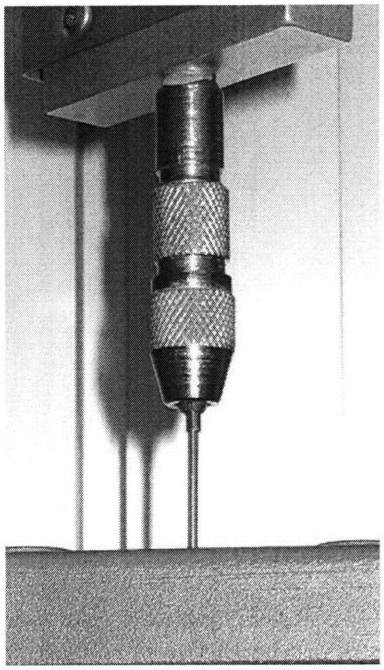


Figure 3-12: Nested oversized and design tubes

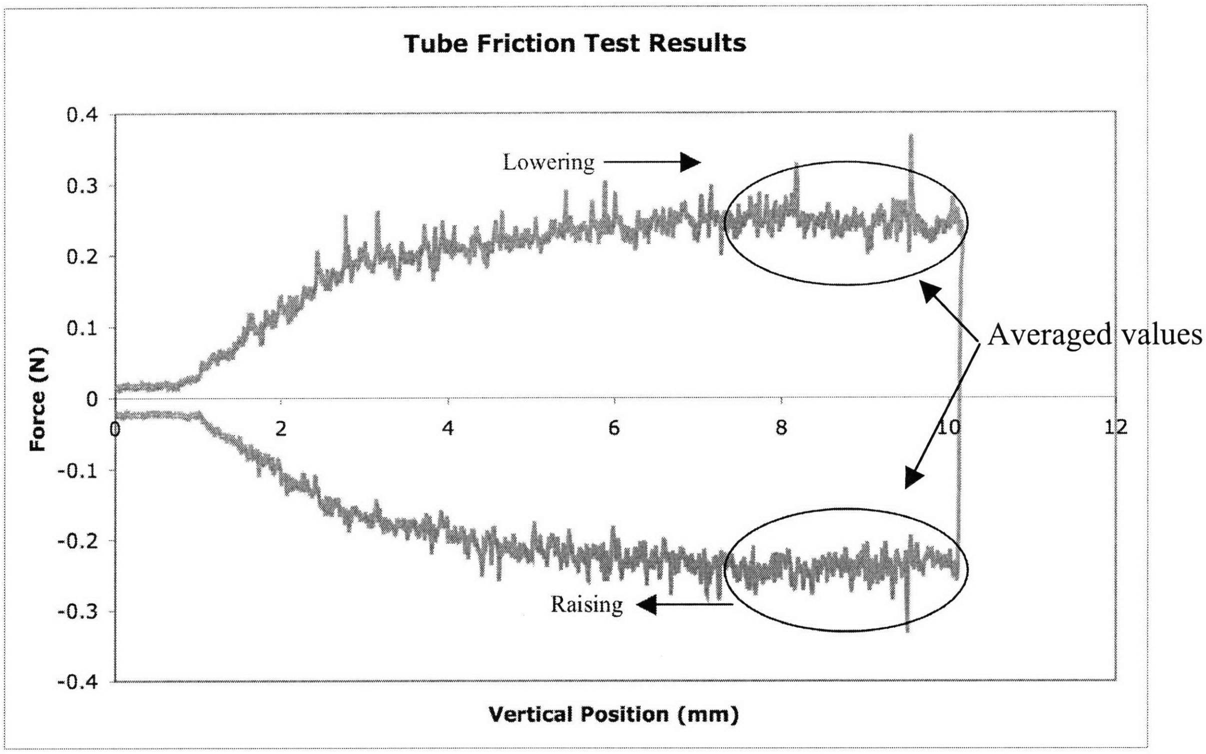


Figure 3-13: Example of tube friction force tests

Table 3-3: Averages of measured tube friction forces

Rate (mm/s)	Mean Force Up (N)	Mean force Down (lbf)
2.54	0.200	0.243
0.38	0.099	0.138
0.13	0.207	0.227

3.3.2 Horizontal Force Tests

Another possible source of error between the capstan prediction and the measured results was the difference of the sliding resistance of each filament. To quantify the effects of the sliding resistance on the tension results, tests were performed measuring the force required to slide the filament through the tube. These tests were completed horizontally to minimize gravitational effects (Figure 3-14). The same rate and travel were used for these tests as for the tension tests (0.38 mm/sec, 10.2 mm right then left). Figure 3-15 shows typical force results required to slide the filament through a tube. Table 3-4 shows the average forces measured which relate to raising and lowering forces in the tension tests. This test included the friction and sliding resistance, making it a more complete measure of error compared to the-tube-on tube friction tests.

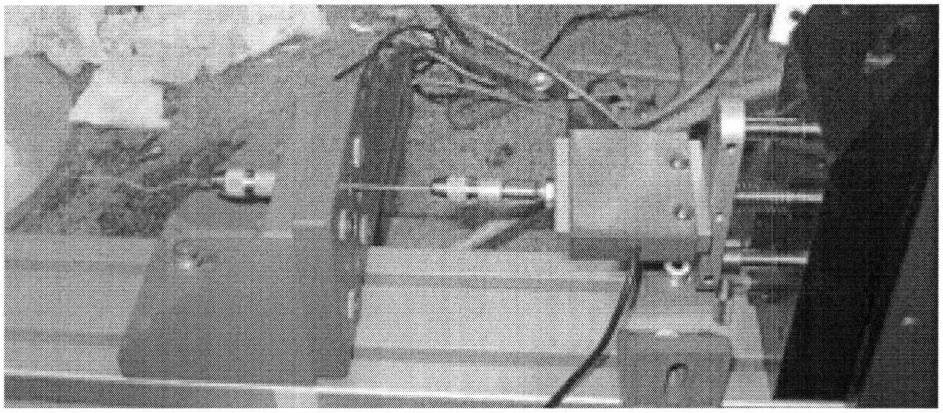


Figure 3-14: Horizontal force test setup

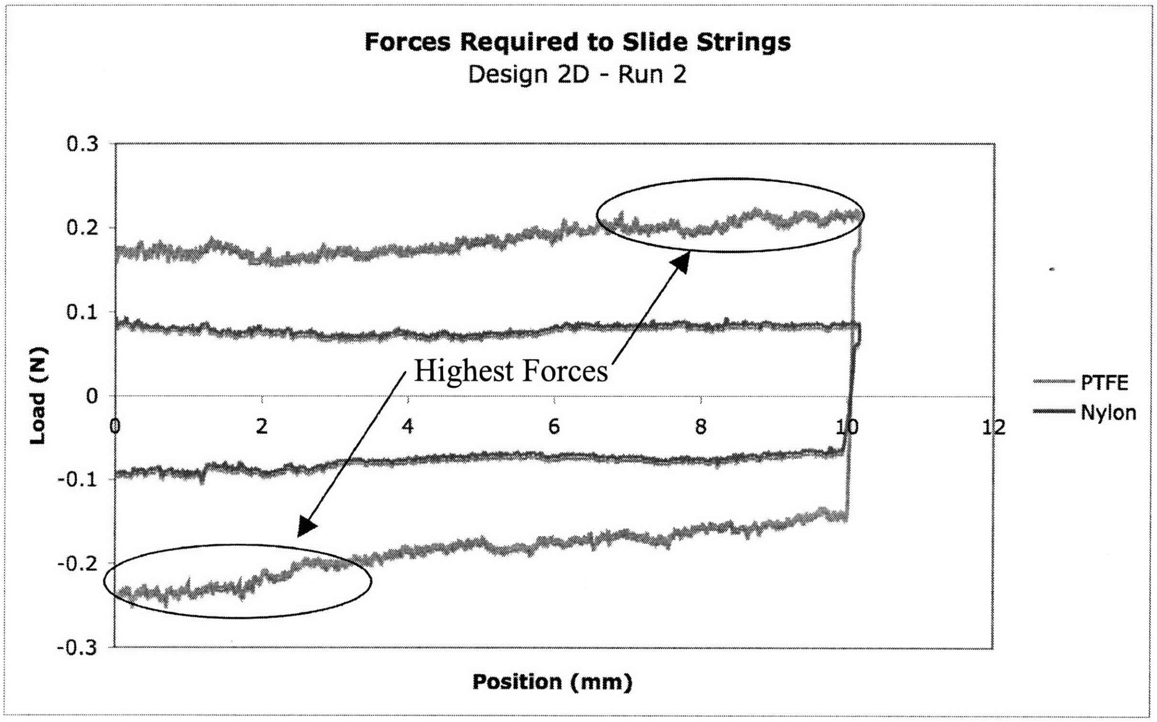


Figure 3-15: Force required to slide the filaments through tube design 2D

Table 3-4: Average force values of nylon and PTFE sliding through tubes

	Nylon Average Forces (N)		PTFE Average Forces (N)	
Design	Right (Raising)	Left (Lowering)	Right (Raising)	Left (Lowering)
1A	0.125	-0.136	0.335	-0.329
1B	0.097	-0.107	0.237	-0.224
1C	0.085	-0.099	0.259	-0.244
1D	0.093	-0.107	0.248	-0.187
2A	0.104	-0.102	0.312	-0.291
2B	0.101	-0.116	0.257	-0.230
2C	0.100	-0.118	0.331	-0.281
2D	0.082	-0.089	0.194	-0.165

3.3.3 Tension Analysis

The results from the previously described tests were analyzed to determine whether the capstan equation could provide an accurate model for the system. Maximum and minimum capstan predictions were established using the max and min values of the angular displacement and coefficients of friction. These numbers provided a range within which the actual values could be considered valid. The raw data was adjusted to account for the tube-on-tube friction and the sliding resistance (which included the tube-on-tube friction). The adjusted and original values were compared to capstan predictions. Figure 3-16 through Figure 3-19 show these predictions and adjusted forces for each filament type when raised and lowered. The measured forces for the nylon fell on or near the predictions made with the raw data, but the PTFE forces did not. When the tube-on-tube friction and sliding resistance forces were used to correct the PTFE measured forces, the results were within the prediction range. This suggests that the nested steel tube sliding resistance effects were significant in the case of the PTFE but not the nylon. Table 3-5 and Table 3-6 show the percent differences between the measured values, adjusted values, and predictions.

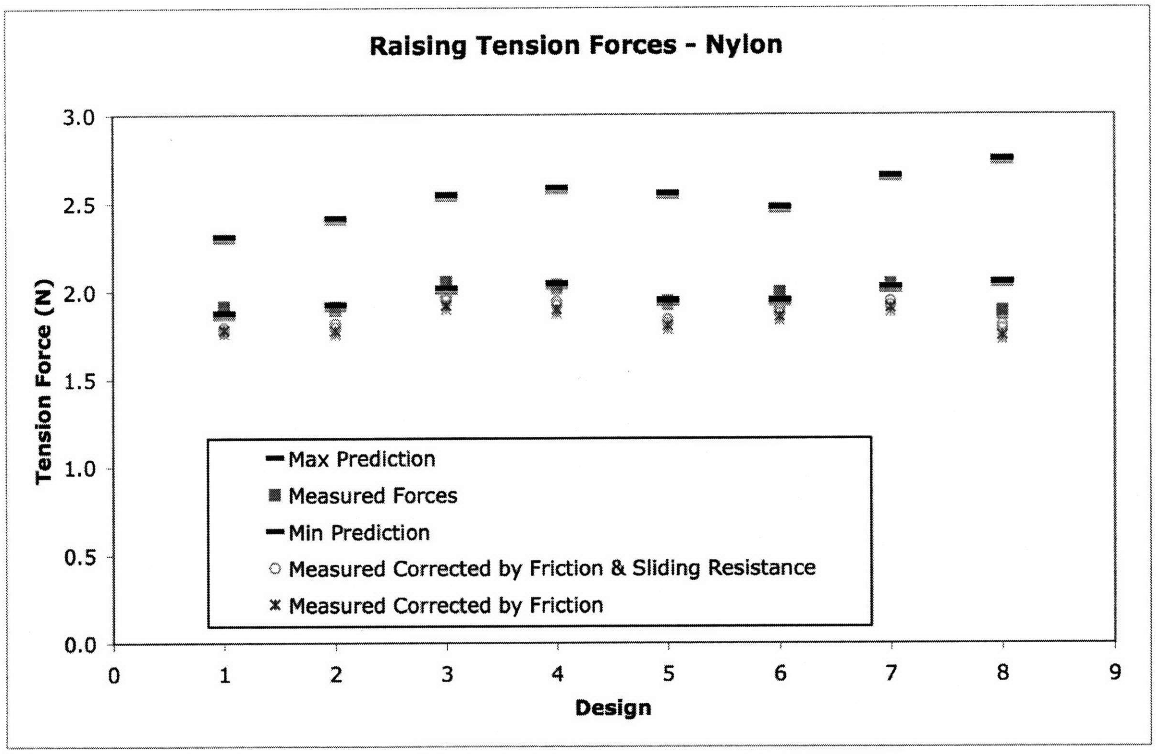


Figure 3-16: Predicted max, min, measured and adjusted raising nylon forces

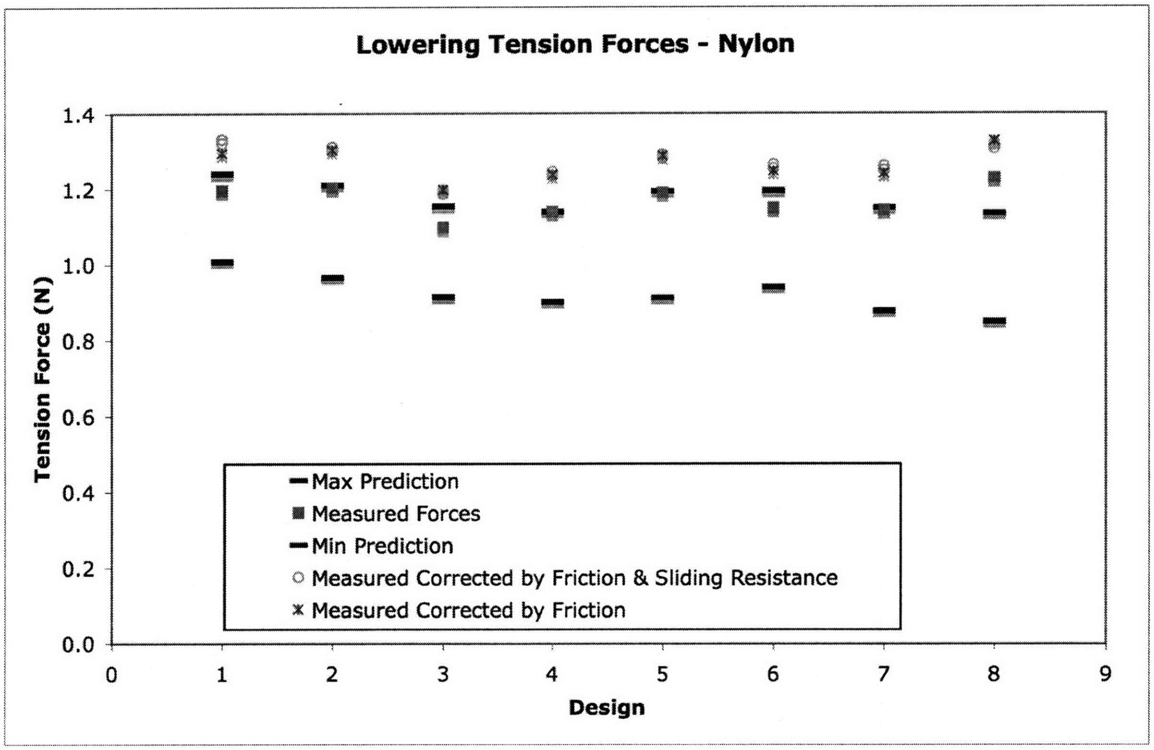


Figure 3-17: Predicted max, min, measured and adjusted lowering nylon forces

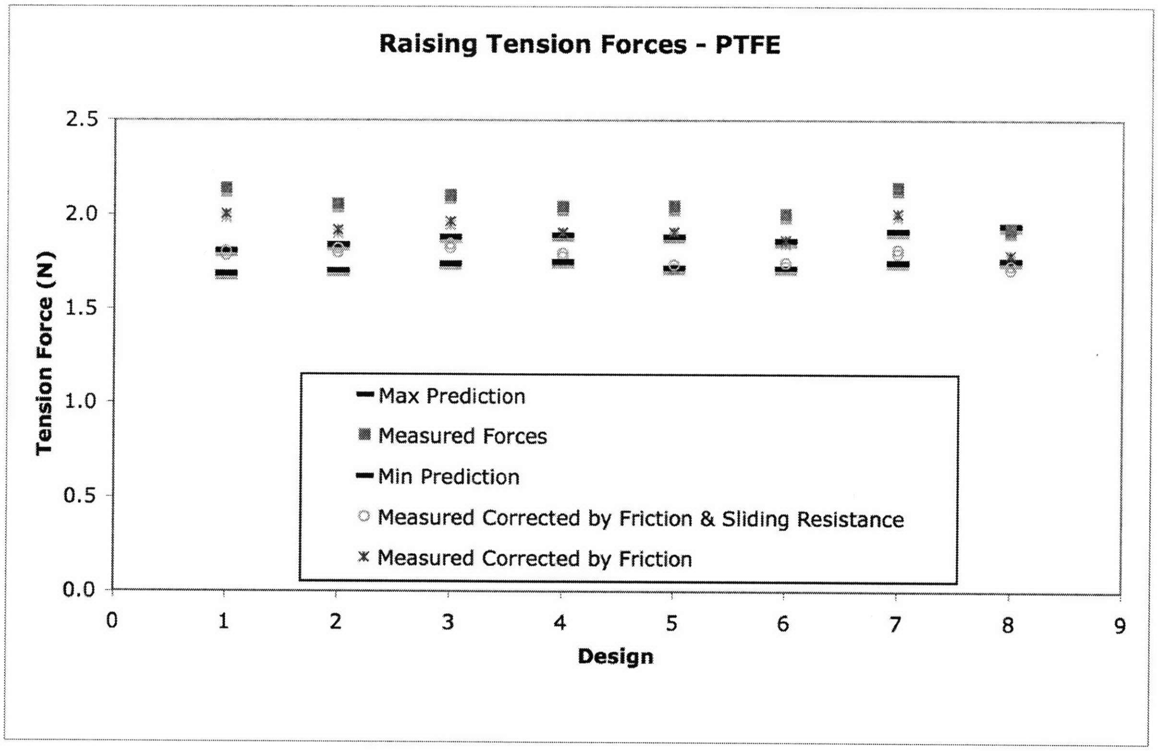


Figure 3-18: Predicted max, min, measured and adjusted raising PTFE forces

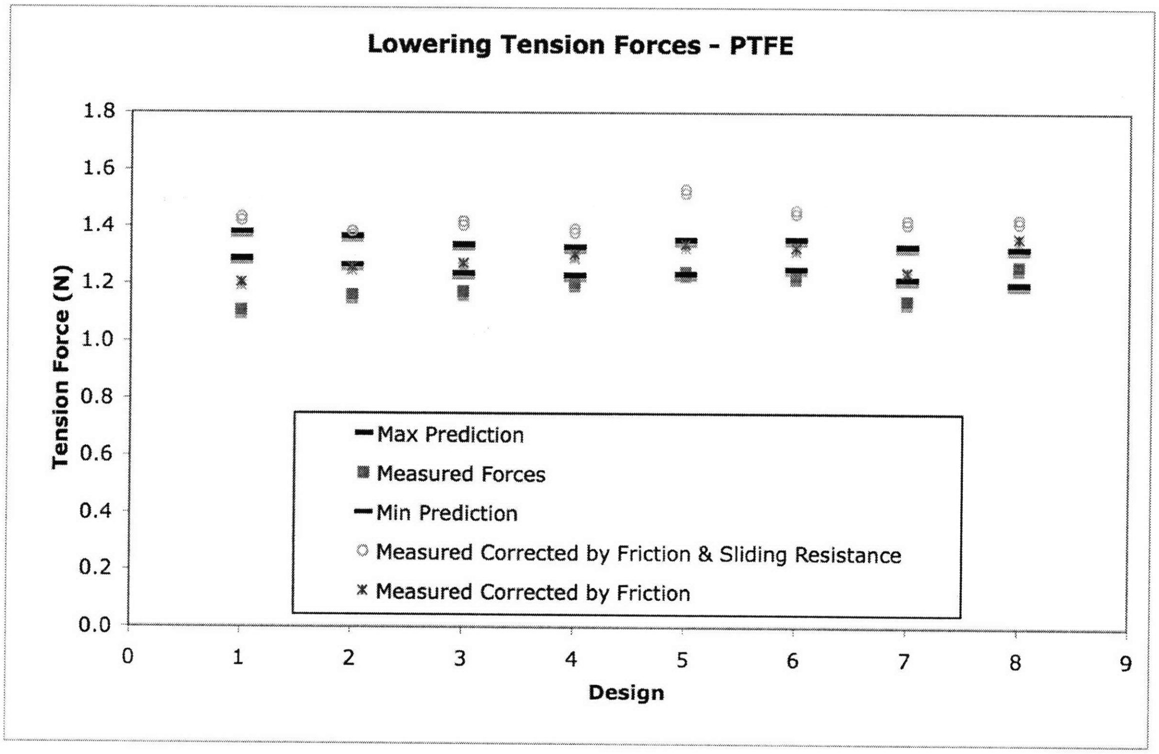


Figure 3-19: Predicted max, min, measured and adjusted lowering PTFE forces

Table 3-5: Comparison of measured, corrected and predicted forces using nylon filament

	Uncorrected Forces		Friction Corrected Forces		Sliding Resistance Corrected Forces	
Design	Raising	Lowering	Raising	Lowering	Raising	Lowering
1A	In Range	In Range	-5.3%	4.4%	-4.6%	7.4%
1B	-0.5%	In Range	-7.7%	7.7%	-5.6%	8.4%
1C	In Range	In Range	-5.0%	3.8%	-2.3%	3.9%
1D	-0.5%	In Range	-7.2%	8.7%	-5.0%	9.4%
2A	-0.4%	In Range	-7.4%	7.9%	-5.7%	8.2%
2B	In Range	In Range	-4.8%	4.4%	-2.9%	5.9%
2C	In Range	In Range	-6.0%	8.0%	-4.1%	9.7%
2D	-8.1%	8.2%	-14.8%	16.9%	-12.1%	16.0%

Table 3-6: Comparison of measured, corrected and predicted forces using PTFE filament

	Uncorrected Forces		Friction Corrected Forces		Sliding Resistance Corrected Forces	
Design	Raising	Lowering	Raising	Lowering	Raising	Lowering
1A	18.4%	-14.1%	10.7%	-6.5%	In Range	3.8%
1B	11.8%	-8.3%	8.8%	-0.5%	In Range	1.3%
1C	11.8%	-5.3%	2.0%	In Range	In Range	5.9%
1D	8.0%	-2.0%	3.8%	In Range	In Range	4.7%
2A	8.6%	In Range	1.2%	In Range	In Range	13.1%
2B	7.7%	-1.9%	2.6%	In Range	In Range	7.4%
2C	12.1%	-6.2%	-2.6%	In Range	In Range	6.7%
2D	In Range	In Range	In Range	-6.3%	-1.7%	7.9%

Factors that influence the sliding resistance include friction, resistance to bending (E*I), hysteresis, and radial clearance. The friction tests showed that the PTFE has a much lower friction coefficient than the nylon so that influence was ruled out. The modulus of elasticity test showed that the resistance to bending of each filament was comparable, eliminating the potential influence of that factor. The remaining potential factors include the radial clearance and hysteresis of the materials. The nylon had a larger radial clearance and little to no hysteresis properties making it easier to slide through the tube. The PTFE on the other hand had noticeable hysteresis. During these tests, the PTFE did not produce flat line results as did the nylon. Instead, particularly on the second half of the test as the PTFE was returning to the zero position, the force decreased (visible in Figure 3-15). This was observed in most of the PTFE tests, particularly when the same filament had been used repeatedly. This trend may be attributed to the hysteresis of the PTFE and can be explained by thinking of the PTFE taking on an initial shape inside the tube. As it is pulled away from the starting position it is forced to take an increasingly different shape. Due to hysteresis, the PTFE resists changing shape. The contact force of the filament against the inside of the tube increases as it advances through the tube. Therefore, the effects of hysteresis result in lower contact forces at initial filament movement in each direction of travel and increasing contact forces as the test progresses (circled regions in Figure 3-15). This trend was also noticed in the earlier tension tests but not understood until the horizontal sliding tests were completed.

It is important to remember that when the tension test direction changes from raising to lowering the weight, F_{in} and F_{out} trade places (from the upper to lower parts of the filament and vise versa). This switch causes a reversal in the equation and the results. When this occurs, points that were near the upper predictions when being raised then become near the lower predictions when being lowered (Figure 3-16 through Figure 3-19).

3.3.4 Discussion of Tension Results

The results from the tension tests show that the capstan equation can be used to predict forces in filaments passing through tubes if error factors are taken into account. The uncorrected nylon and corrected PTFE results closely approximated the predictions. Correcting the measured forces caused the nylon results to move outside the predicted range while subsequently moving the PTFE results into the predicted range. This may be due to the hysteresis of the PTFE. In the case of the nylon, the tensile forces seem to dominate the mechanics, simplifying the system to the standard capstan case. With the PTFE, hysteresis added a significant effect to the tension test results. With these corrections the PTFE values were all within 13% of the prediction, and most were within 7%. The nylon results were all within the predicted range. Based on the close correlation of predicted and experimentally derived results, the capstan equation can be considered a valid model.

3.4 Compression Tests of Filaments Inside Tubes with Curves

The compression test setup was similar to the tension setup and used nearly all the same components. The primary difference was that an additional tube and load cell replaced the weight (Figure 3-20). This load cell was attached to the foot of the setup frame below and to a third pin vise above. This vise held the additional tube. These additional pieces enabled a compressive load to be applied to the filament while being measured by the two load cells. The forces were applied as the actuator moved. F_{in} (the applied compressive force) and F_{out} (the measured exiting compressive force) were measured by the upper and lower load cells respectively. The added tube had a collar into which the design tube was placed. This collar constrained the design tube laterally but did not transmit axial forces to the lower load cell. For these tests the actuator was advanced at a constant rate of 0.38 mm/sec until the input force reached 8.9 N.

During the compression experiments the nylon filaments did not last more than about three tests before kinking. Kinking occurred where the filament passed from the oversized tube into the design tube. In the oversized tube, the filament had less lateral support and could catch the corner of the design tube. Once the nylon kinked it could not be reused. Due to the limited quantity of the nylon filament, some design tubes were only tested four times (instead of five). Kinking was not a problem with the PTFE because of the higher moment of inertia, so five tests of each tube were completed.

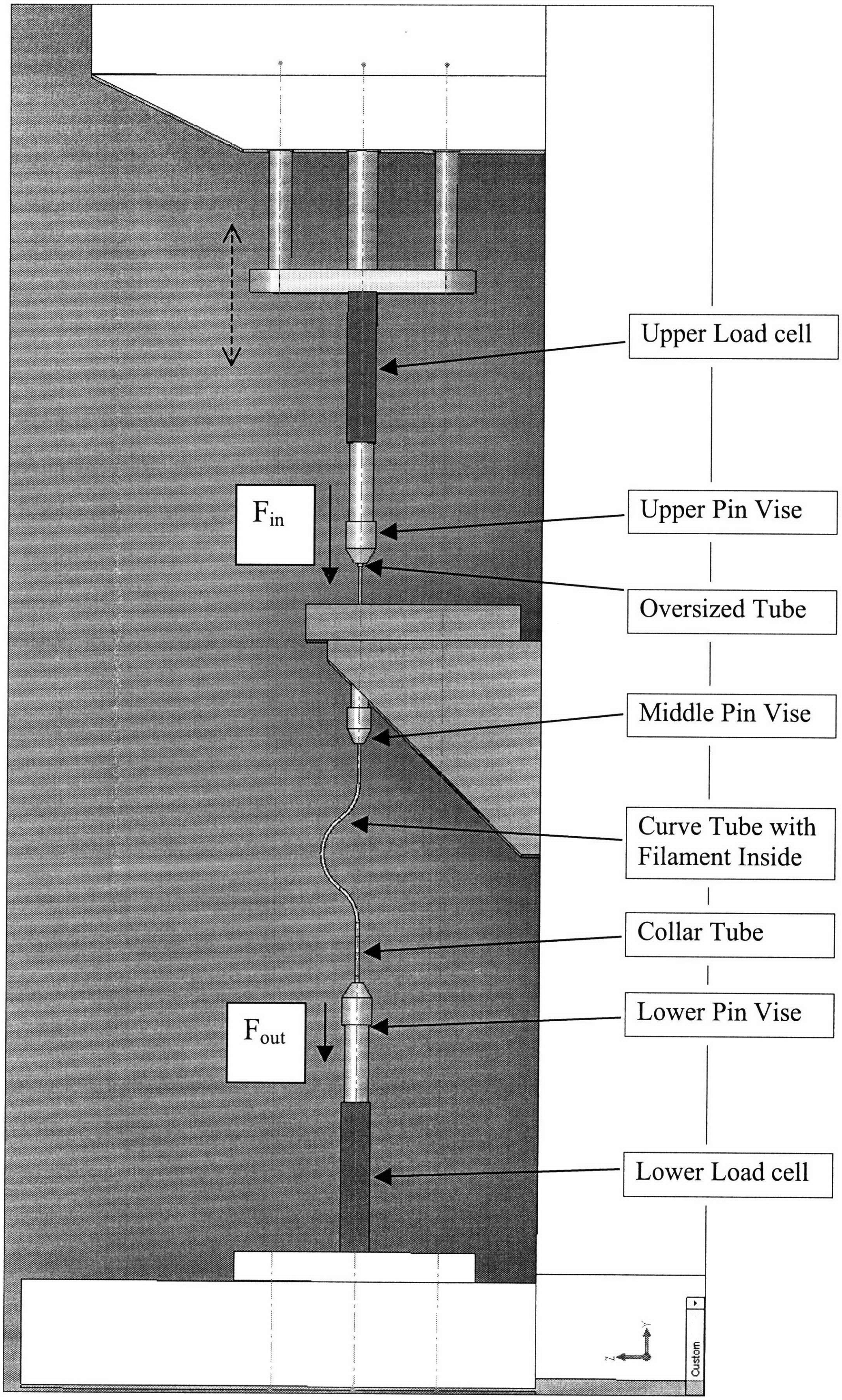


Figure 3-20: CAD model of test setup

3.4.1 Compression Test Results

The same tubes tested in tension were also tested in compression. These compression tests provided very consistent results for each tube tested. Each test started with a flat force region while the filament advanced prior to contacting the lower load cell. This can be seen in the first 0.5 cm of the test data in Figure 3-21. For the analysis these data points were removed (Figure 3-22). Once contact (and compression) was achieved, the input and output forces increased. The output forces tended to follow a near parabolic path and generally leveled off near the end of each test, while the measured input forces had no common shape. Due to the differences in the two materials and geometries of the filaments, the actuator had to advance about twice as far with the nylon as with the PTFE to reach the 8.9 N limit of each test. For the analysis both the original and re-zeroed data (zeroing of the data where compression begins) were studied. Zeroing was achieved by averaging the flat sections at the beginning of each test and subtracting that value from the rest of the data points. Figure 3-23 shows a force vs. position plot of zeroed values.

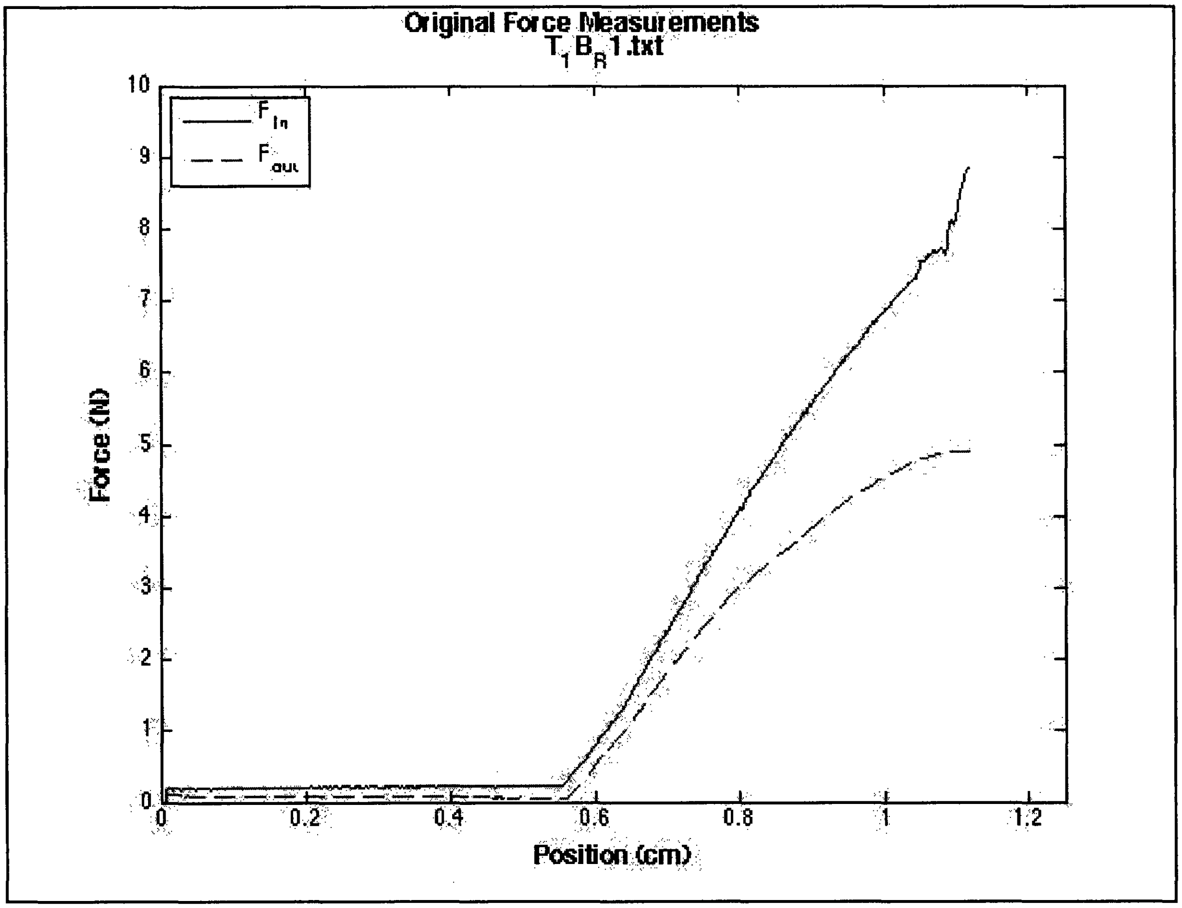


Figure 3-21: Compression data for design 1B using PTFE Filament

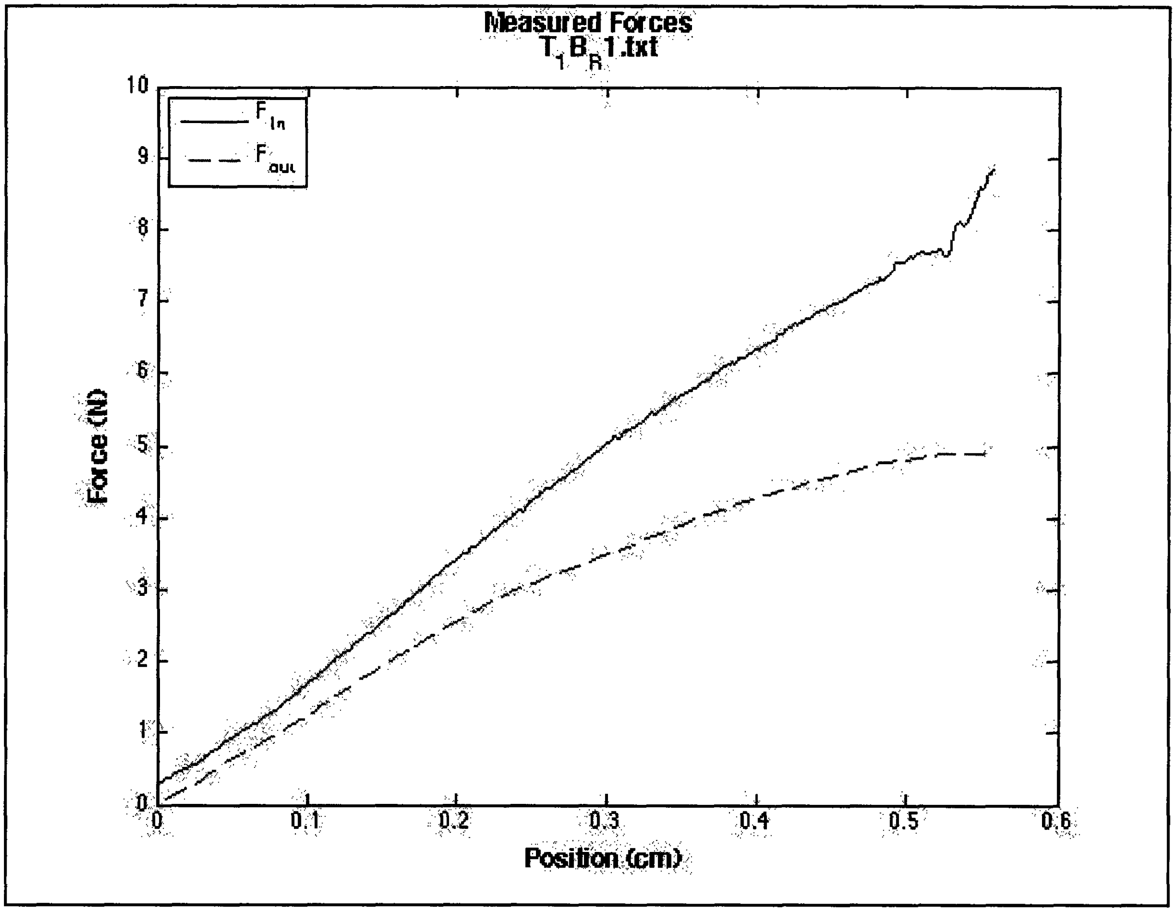


Figure 3-22: Truncated compression data for design 1B using PTFE filament

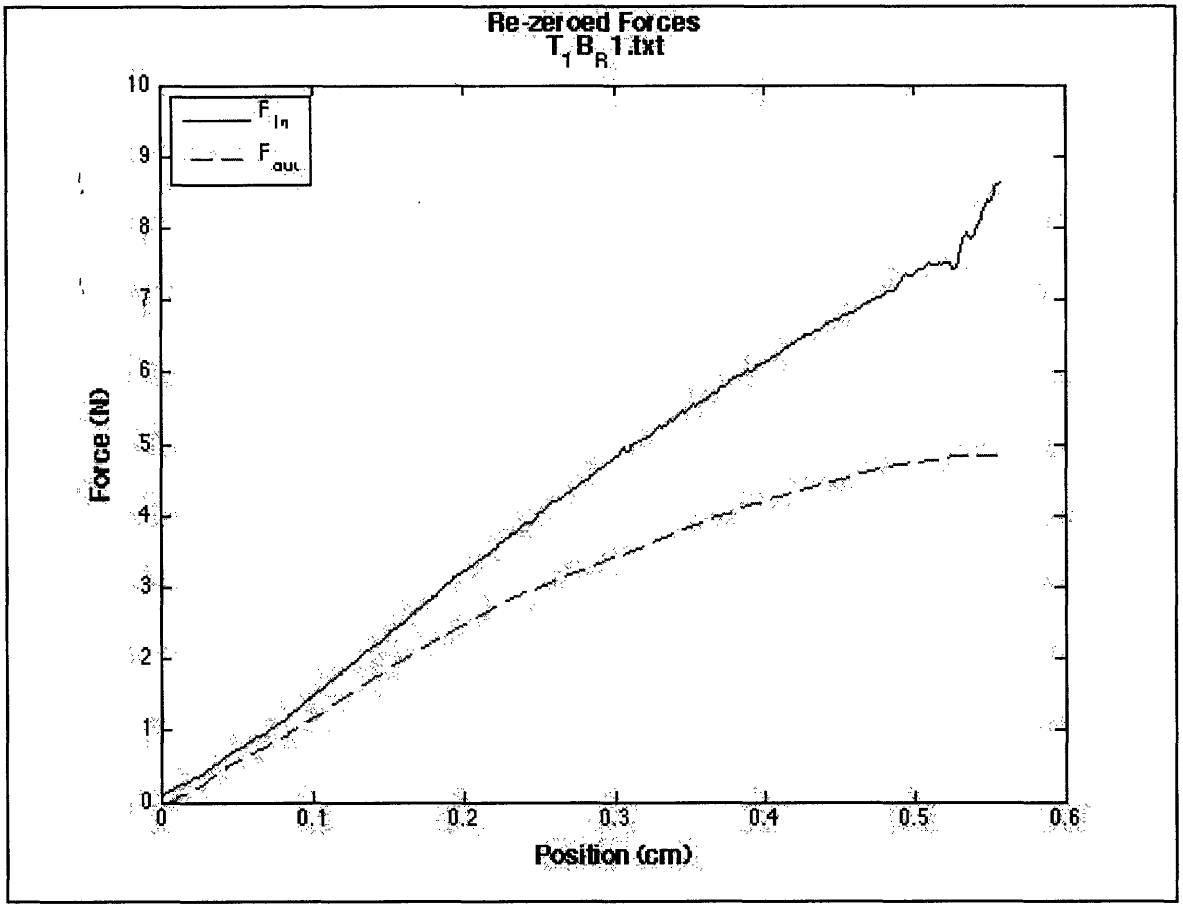


Figure 3-23: 'Zeroed' F_{in} and F_{out} curves at beginning of compression loading for PTFE filament

To study the effects of the length of the long straight section of the design tube, two design tubes were shortened and tested. Designs 1C and 2C were selected for the tests. Table 3-7 shows the original and shortened lengths of the straight section of the design tubes. Each was tested 5 times at three shortened lengths with the PTFE filament. The final tests were also repeated with the nylon filament. Typical results from these tests can be seen in Figure 3-24 and Figure 3-25.

Table 3-7: Length of long straight sections of design tubes to test effect on output

	Lengths of Long Straight Sections (cm)	
	Design 1C	Design 2C
Original	11.7	11.1
Test 1	10.5	9.8
Test 2	9.0	8.4
Test 3	7.3	6.7

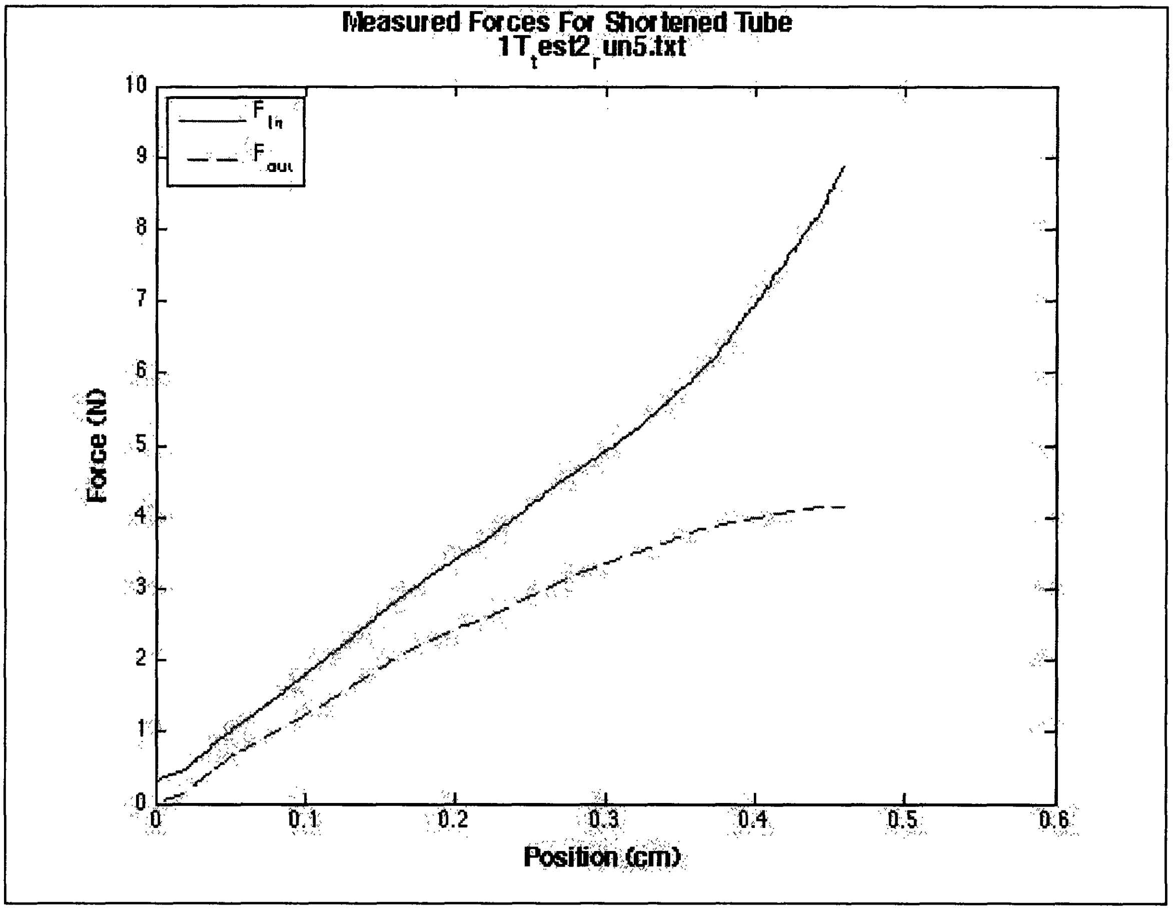


Figure 3-24: Typical results from shortened test 2 using tube design 1C and PTFE filament

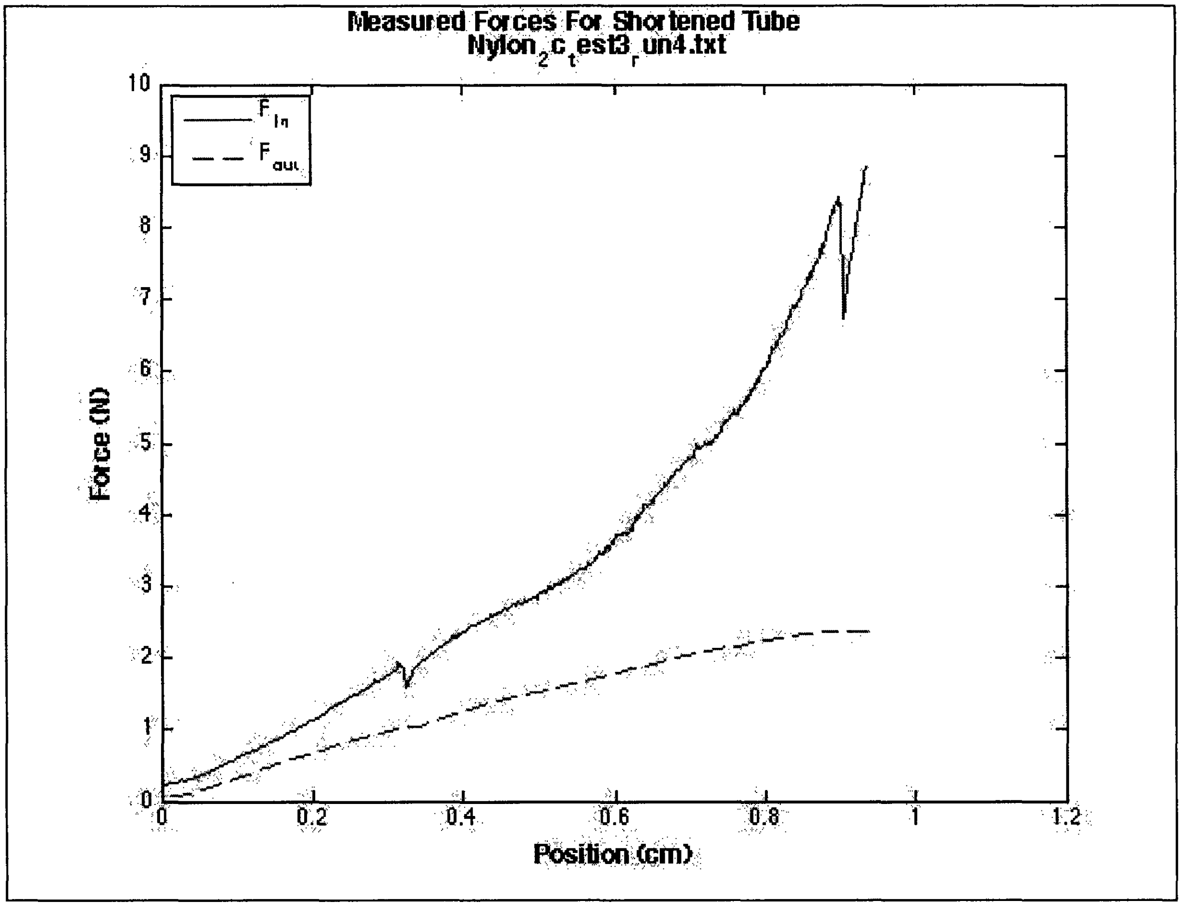


Figure 3-25: Typical results from shortened test 3 using tube design 2C and nylon filament

In order to develop a functional clutch system for testing on raw chicken breast, nine additional tubes were made. The shapes of these tubes were selected to produce lower lock-up forces than the original eight. These tubes were tested in compression to ensure that the model remained valid. Each included two bends and were the same diameter (OD 1.47 mm and ID 1.22 mm) and length (17.8 cm) as the original design tubes. These tests were performed with the PTFE filament and repeated five times for each tube. The design parameters of the new tubes are presented in Table 3-8. These tests produced similarly shaped curves as the originals but with lower values of F_{out}.

Table 3-8: Design parameters of new nine tubes tested

Design	# of Bends	Avg. Bend Radius (mm)	Angular displacement (rad)
R0.5_A250	2	22.6	2.7123
R0.5_A300	2	20.7	3.4574
R0.5_A350	2	19.6	4.3547
R0.6_A250	2	26.2	2.6975
R0.6_A300	2	22.4	3.8198
R0.6_A350	2	23.7	4.1792
R0.7_A250	2	27.6	2.9556
R0.7_A300	2	29.4	3.3545
R0.7_A350	2	29.1	3.7416

3.4.2 Capstan Model Compression Analysis

The output forces on all of the tests produced similarly shaped curves. It was determined that a second-order quadratic function closely approximated these curves. Each output data set was fit with a quadratic function (Figure 3-26). The average and standard deviations of the maximum values of each fit were found and recorded in Table 3-9. Table 3-10 and Table 3-11 contain the coefficients of the curve fit (Fout = $A*x^2+B*x+C$) for each design and filament type.

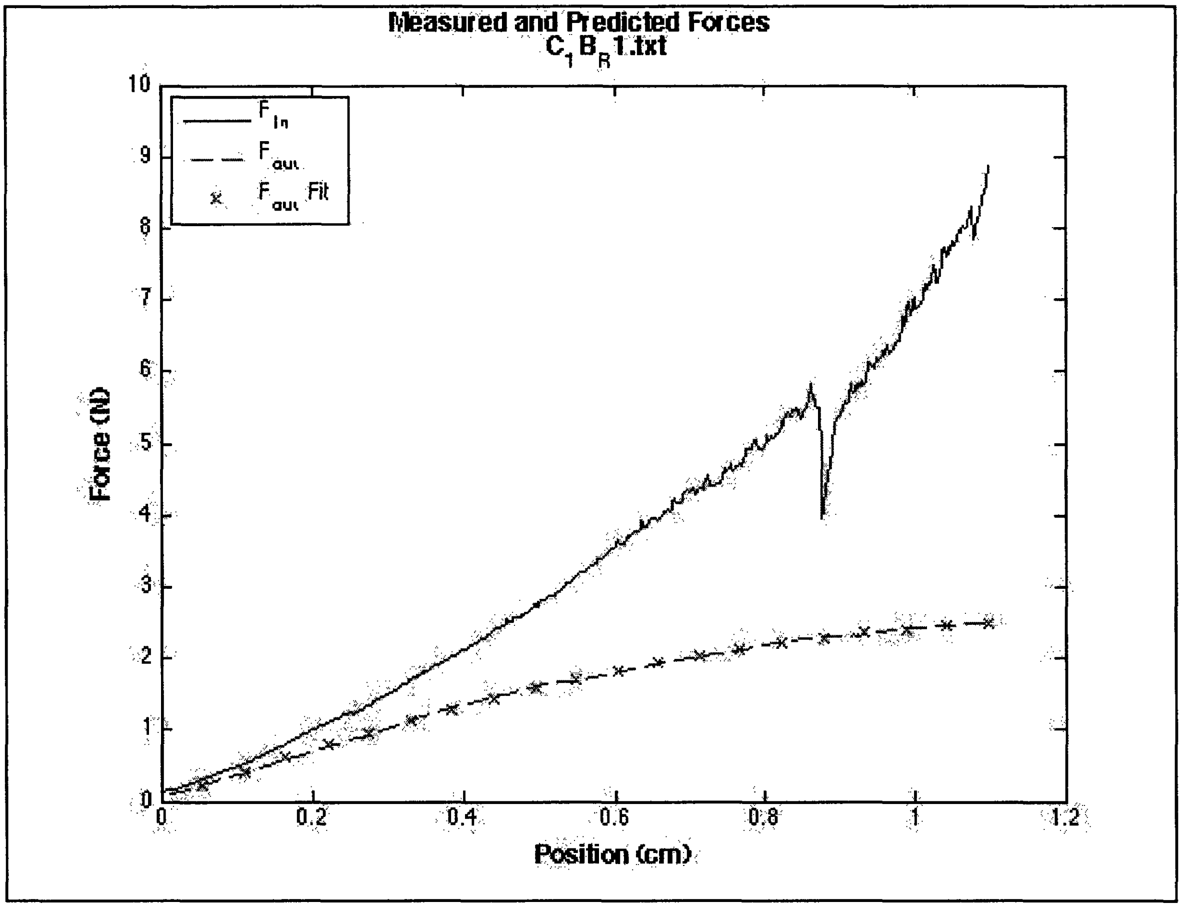


Figure 3-26: F_{in} vs. F_{out} with parabolic curve fit on F_{out} for nylon filament

Table 3-9: Maximum F_{out} values calculated from curve fits of data

Design	Nylon Avg. (N)	Nylon Stdev.	PTFE Avg. (N)	PTFE Stdev.
1A	2.69	0.16	5.03	0.24
1B	2.63	0.23	4.95	0.19
1C	2.43	0.31	4.73	0.33
1D	2.43	0.52	4.87	0.13
2A	2.58	0.42	5.05	0.23
2B	3.67	1.40	5.61	0.93
2C	2.73	0.38	5.20	0.34
2D	2.26	0.31	4.29	0.26

Table 3-10: F_{out} curve-fit coefficients for nylon ($F_{out} = A*x^2+B*x+C$)

Design	A	B	C
1A	-1.53	4.07	-0.04
1B	-1.54	4.00	0.00
1C	-1.43	3.70	0.00
1D	-1.52	3.82	-0.06
2A	-1.41	3.79	-0.01
2B	-1.06	3.74	-0.05
2C	-1.28	3.71	0.00
2D	-1.52	3.71	-0.02

Table 3-11: F_{out} curve-fit coefficients for PTFE ($F_{out} = A*x^2+B*x+C$)

Design	A	B	C
1A	-11.14	15.06	-0.07
1B	-11.84	15.50	-0.13
1C	-10.33	14.09	-0.09
1D	-10.07	14.21	-0.14
2A	-11.98	15.53	-0.04
2B	-6.96	12.38	-0.07
2C	-8.86	13.55	-0.08
2D	-12.23	14.69	-0.14

Another interesting method of comparison plotted the input forces against the output forces. This created an efficient way to compare the capstan prediction to the actual data. The capstan equation in these coordinates produces a straight line of slope $e^{(\theta*\mu)}$. Using the maximum and minimum values of the angular displacement and friction coefficient max and min prediction curves were expressed. Figure 3-27 shows a typical F_{in} vs. F_{out} result and how it is similar to the predictions over a range but then diverges from it.

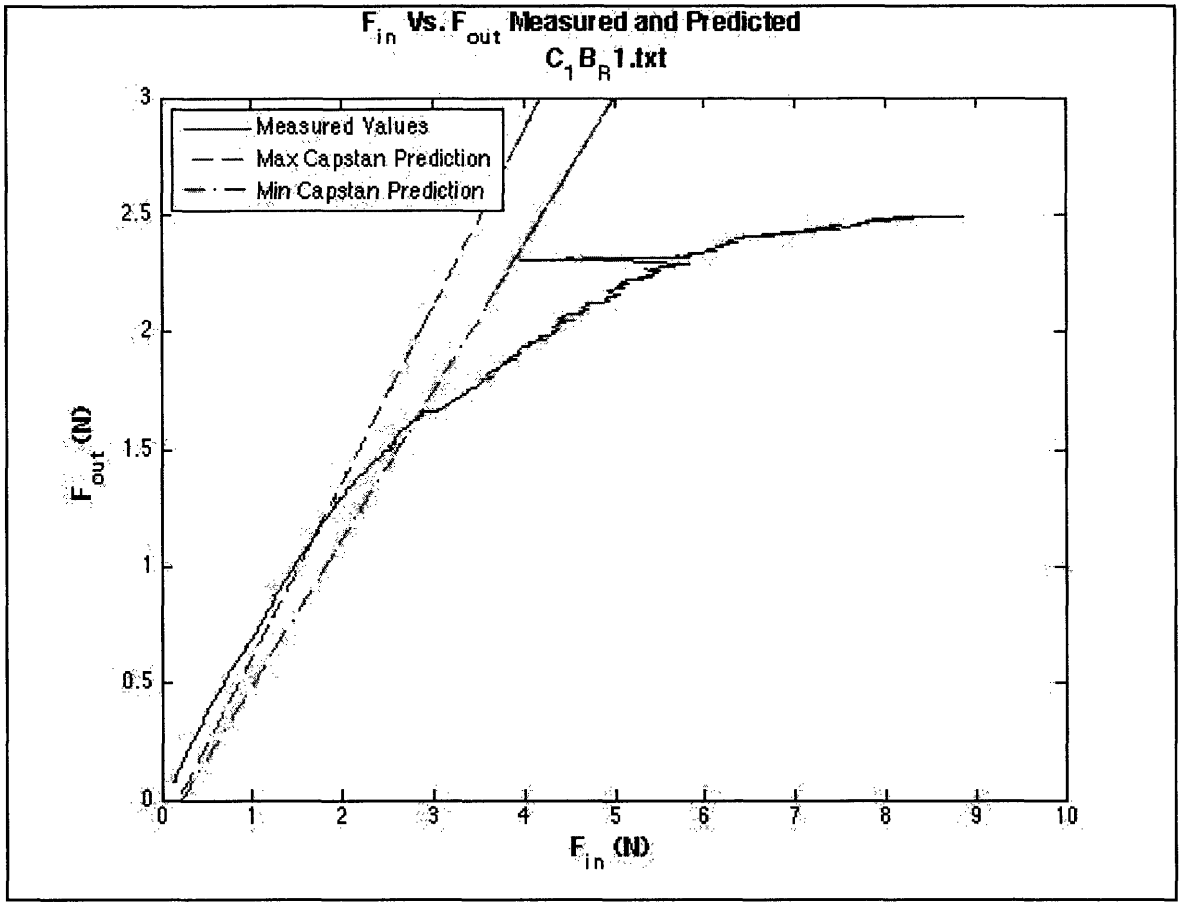


Figure 3-27: Typical F_{in} vs F_{out} curve with capstan predictions for nylon filament

Initially the percent differences between the measured and predicted values were studied for trends. Unfortunately there proved to be too much variation between tests and designs to find anything conclusive.

Another point of interest was the slope of the measured curve. Theoretically, if the measured curve was be modeled with the capstan prediction, similar slopes would be found. To study this, the first and second derivatives of F_{in} vs. F_{out} were calculated to determine where the measured curve diverged from the prediction. The break point was determined to be where the first derivative significantly diverged from the prediction. The cutoff value was chosen to be where the second derivate reached -0.2 (1/N^2). Figure 3-28 through Figure 3-30 show that point on the F_{in} vs. F_{out}, first and second derivatives respectively.

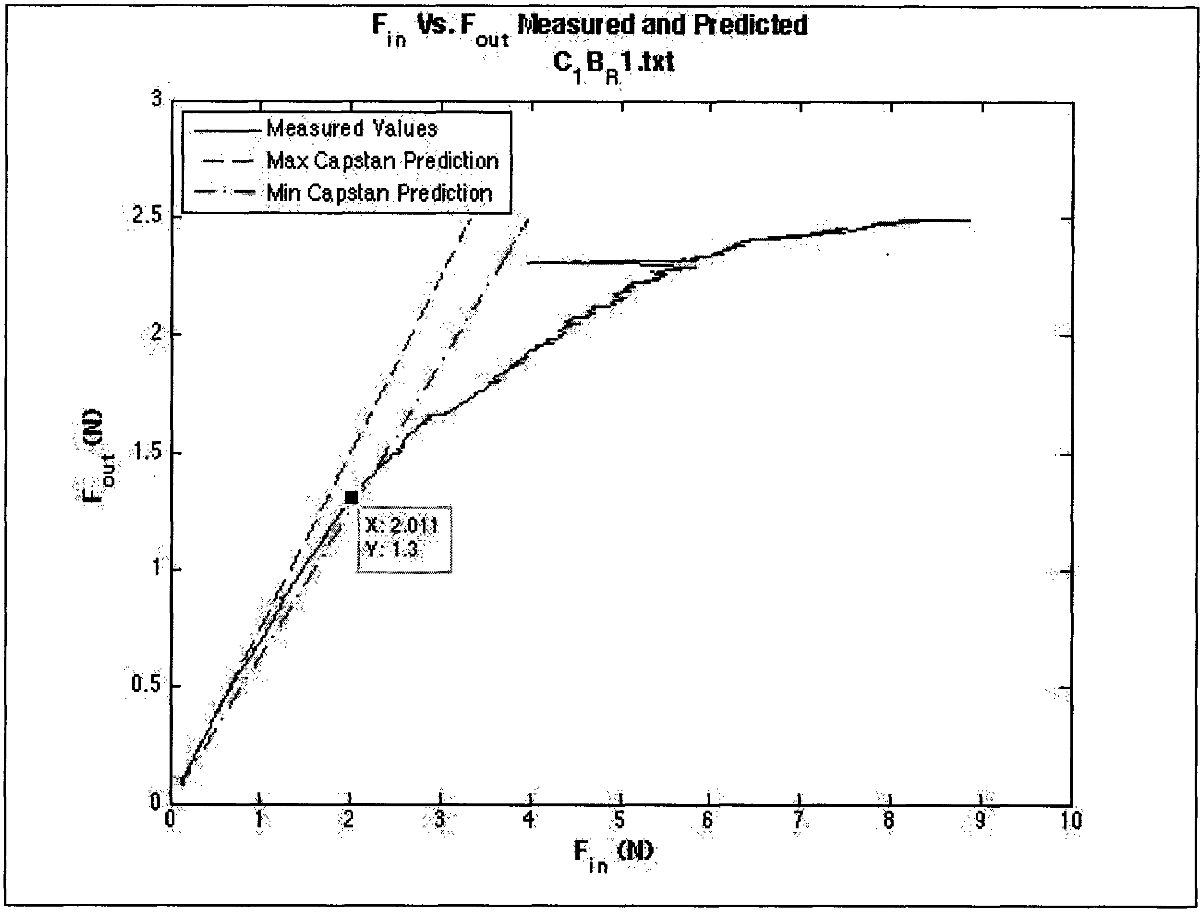


Figure 3-28: Plot of F_{in} vs F_{out} with cutoff point marked for nylon filament

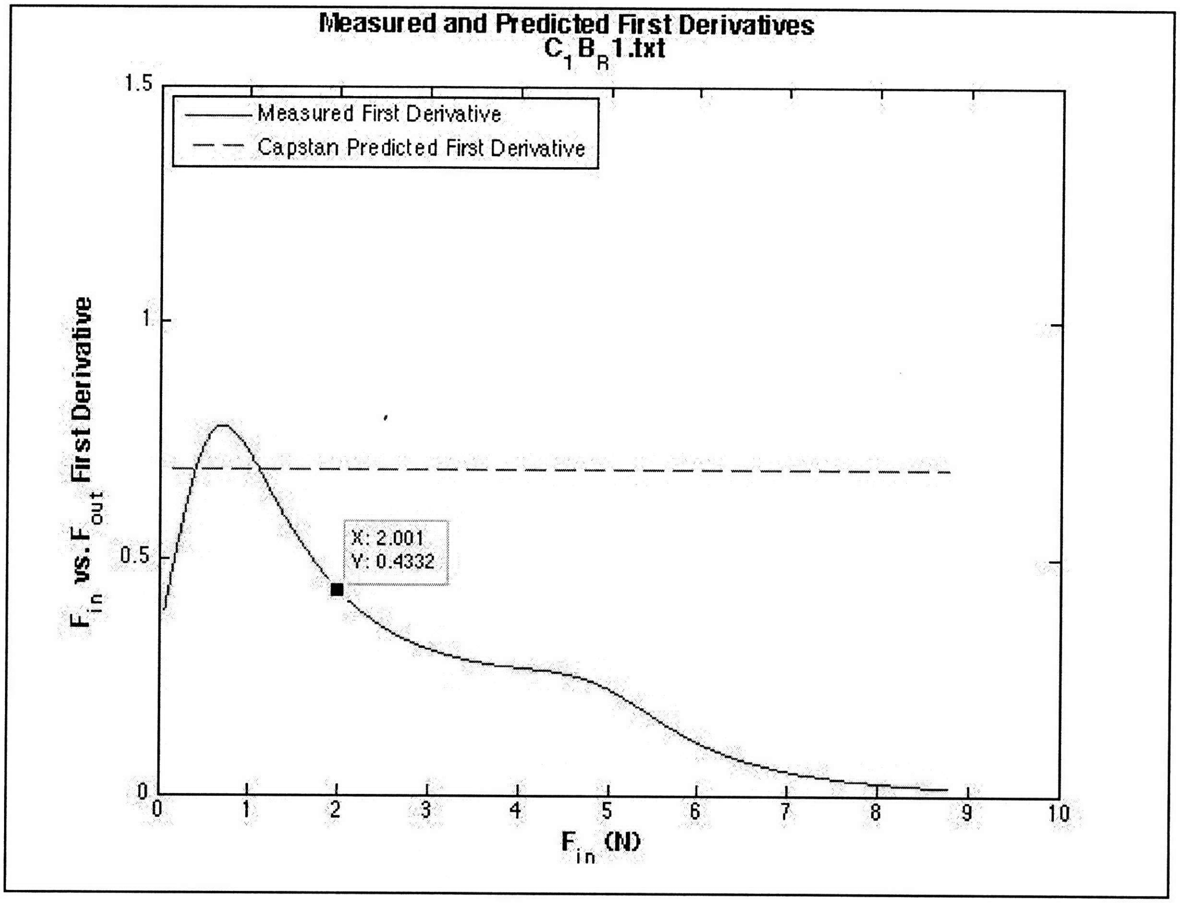


Figure 3-29: Plot of F_{in} vs F_{out} first derivative with cutoff point marked for nylon filament

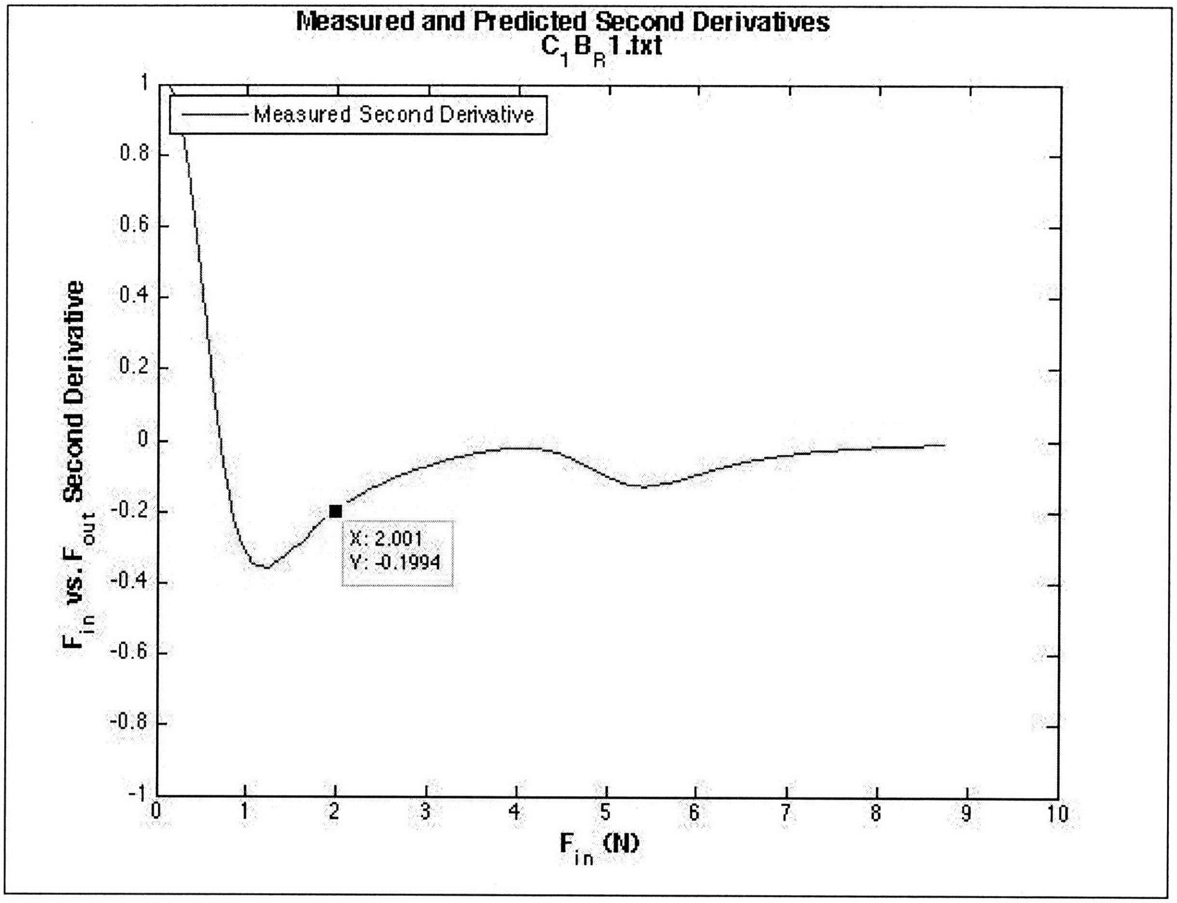


Figure 3-30: Plot F_{in} vs F_{out} second derivative with cutoff point marked for nylon filament

At this point of interest, F_{in}, F_{out}, F_{in}/F_{out}, and position values were recorded. The following tables (Table 3-12 through Table 3-15) contain the averages of these values for the zeroed and non-zeroed results for each filament. Each average was ranked highest (being 1) to lowest (being 8) to show the trends in the data. For comparison to the measured values, the tube design parameters angular displacement and average bend radius with their associated ranks are shown in Table 3-16.

Table 3-12: Break point values for zeroed forces in nylon filament

	F_{in} (N)		F_{out} (N)		Position (mm)		F_{in} /F_{out}	
Design	Average	Rank	Average	Rank	Average	Rank	Average	Rank
1A	0.62	2	0.48	2	0.14	2	0.42	7
1B	0.55	4	0.46	3	0.12	4	1.19	6
1C	0.44	6	0.34	6	0.10	6	1.48	3
1D	0.43	7	0.25	7	0.09	7	-4.24	8
2A	0.60	3	0.35	5	0.13	3	2.63	2
2B	0.35	8	0.19	8	0.09	8	4.11	1
2C	0.64	1	0.50	1	0.16	1	1.25	5
2D	0.48	5	0.37	4	0.12	5	1.33	4

Table 3-13: Break point values for zeroed forces in PTFE filament

	F_{in} (N)		F_{out} (N)		Position (mm)		F_{in} /F_{out}	
Design	Average	Rank	Average	Rank	Average	Rank	Average	Rank
1A	2.77	1	2.00	1	0.18	2	1.23	6
1B	1.56	5	1.09	5	0.10	5	1.30	3
1C	0.89	7	0.78	6	0.06	7	1.15	7
1D	2.71	2	1.93	2	0.19	1	1.24	5
2A	0.95	6	0.76	7	0.07	6	1.44	1
2B	1.69	4	1.20	4	0.12	4	1.11	8
2C	1.79	3	1.23	3	0.13	3	1.28	4
2D	0.47	8	0.44	8	0.04	8	1.38	2

Table 3-14: Break point values for measured forces in nylon filament

	F_{in} (N)		F_{out} (N)		Position (mm)		F_{in} /F_{out}	
Design	Average	Rank	Average	Rank	Average	Rank	Average	Rank
1A	1.01	2	0.70	3	2.01	2	1.43	6
1B	0.97	3	0.70	2	1.97	3	1.43	7
1C	0.86	5	0.56	5	1.68	4	1.74	2
1D	0.61	8	0.37	8	1.17	8	1.44	5
2A	0.89	4	0.58	4	1.66	5	1.61	3
2B	0.73	7	0.44	7	1.45	7	1.93	1
2C	1.07	1	0.77	1	2.30	1	1.34	8
2D	0.80	6	0.53	6	1.64	6	1.52	4

Table 3-15: Break point values for measured forces using PTFE filament

	F_{in} (N)		F_{out} (N)		Position (mm)		F_{in} /F_{out}	
Design	Average	Rank	Average	Rank	Average	Rank	Average	Rank
1A	3.45	1	2.23	1	2.08	1	1.40	7
1B	2.28	3	1.44	3	1.33	3	1.54	3
1C	1.28	7	0.88	7	0.76	7	1.52	4
1D	1.57	6	1.11	6	1.13	5	1.11	8
2A	1.76	5	1.33	4	1.12	6	1.58	2
2B	2.47	2	1.50	2	1.63	2	1.73	1
2C	2.01	4	1.31	5	1.27	4	1.47	5
2D	0.65	8	0.48	8	0.41	8	1.44	6

Table 3-16: Values and ranks of design parameters bend radius and angular displacement

	Average Bend Radius (mm)		Angular displacement (rad)	
Design	Value	Rank	Value	Rank
1A	24.7	8	1.57	8
1B	29.7	4	1.73	7
1C	27.3	6	1.94	5
1D	31.6	3	1.99	3
2A	26.0	7	1.95	4
2B	34.7	1	1.83	6
2C	28.3	5	2.09	2
2D	32.0	2	2.22	1

Further analysis revealed that the F_{in} vs. F_{out} curve was comprised of a mostly linear section and an exponential section. This could be observed by changing the scale of the x-axis from linear to logarithmic. Making that change the linear portion becomes logarithmic while the exponential region becomes linear. Figure 3-31 and Figure 3-32 show the difference between the normal-normal and log-normal scales of the same data for F_{in} vs. F_{out}.

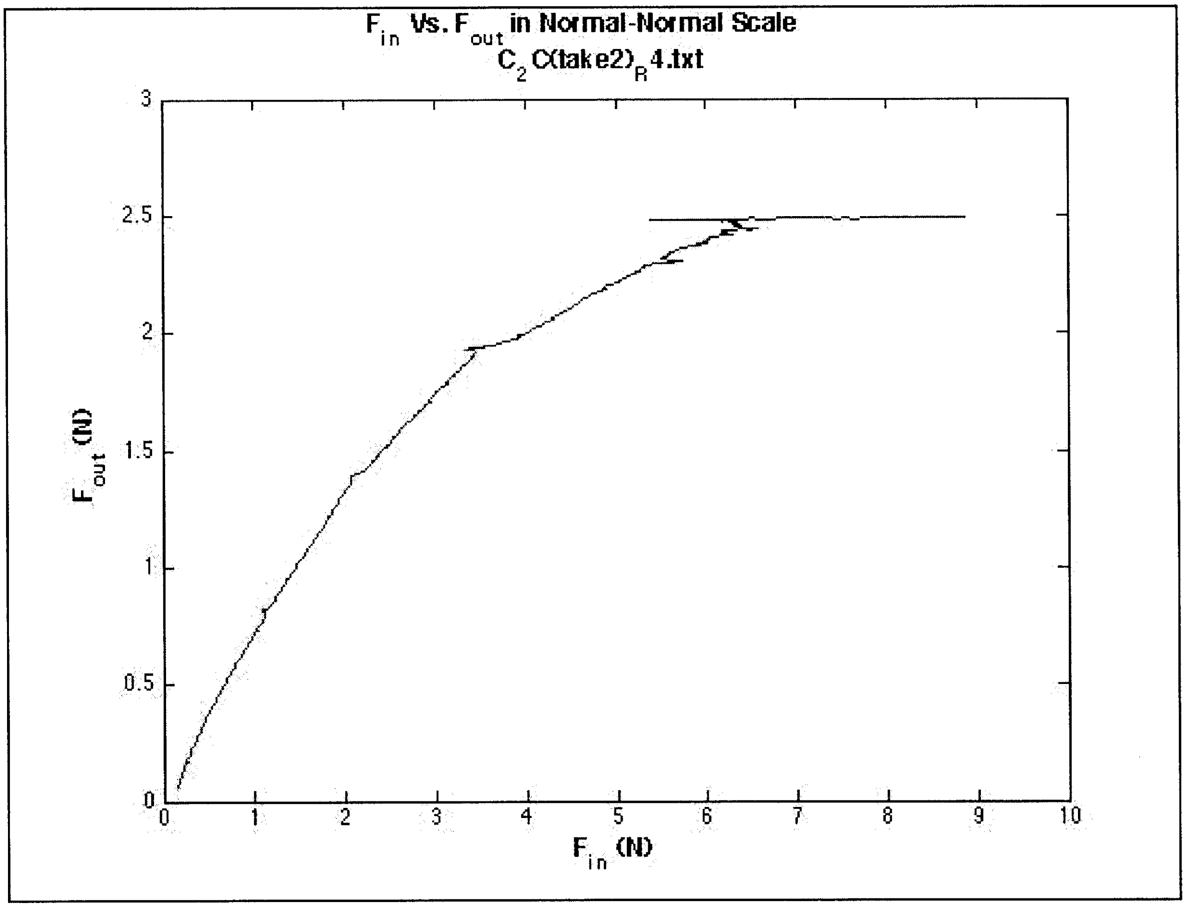


Figure 3-31: F_{in} vs. F_{out} with Capstan predictions for nylon filament

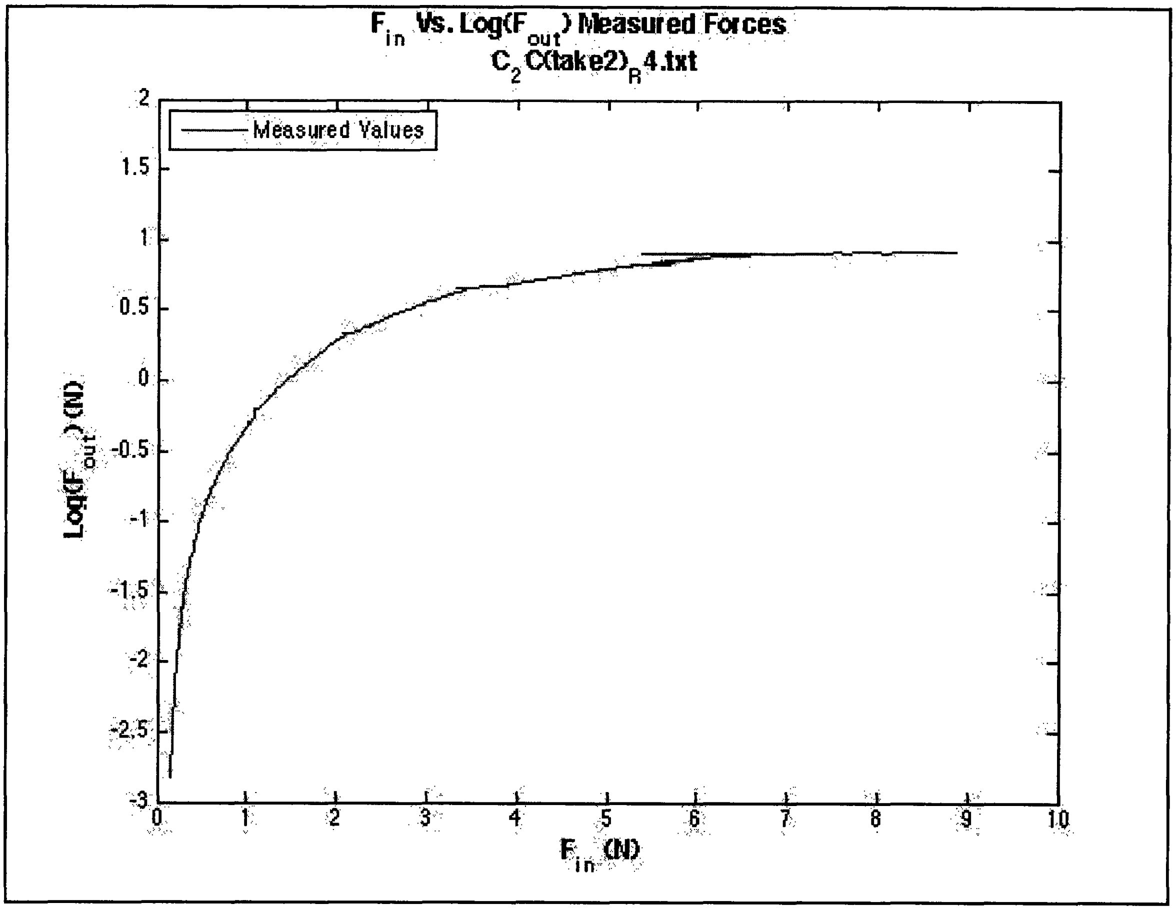


Figure 3-32: F$_{in}$ vs. F$_{out}$ with log-normal scale showing exponential region for nylon filament

The first and second derivatives of the F$_{in}$ vs. log(F$_{out}$) proved that the curve truly becomes linear. It is unclear what causes this transition but it may be the point where the filament forms a full helically buckled shape. The transition point was approximated to be where the second derivative first reached -1.0. Figure 3-33 shows the second derivative of the log of F$_{in}$ with that point marked. Figure 3-34 shows the same point marked on a in normal-normal plot. Based on this criterion, this buckling point was found for each data set. Table 3-17 through Table 3-20 show the values and ranks of F$_{in}$, F$_{out}$, Position and F$_{in}$ /F$_{out}$ for this log based cutoff point.

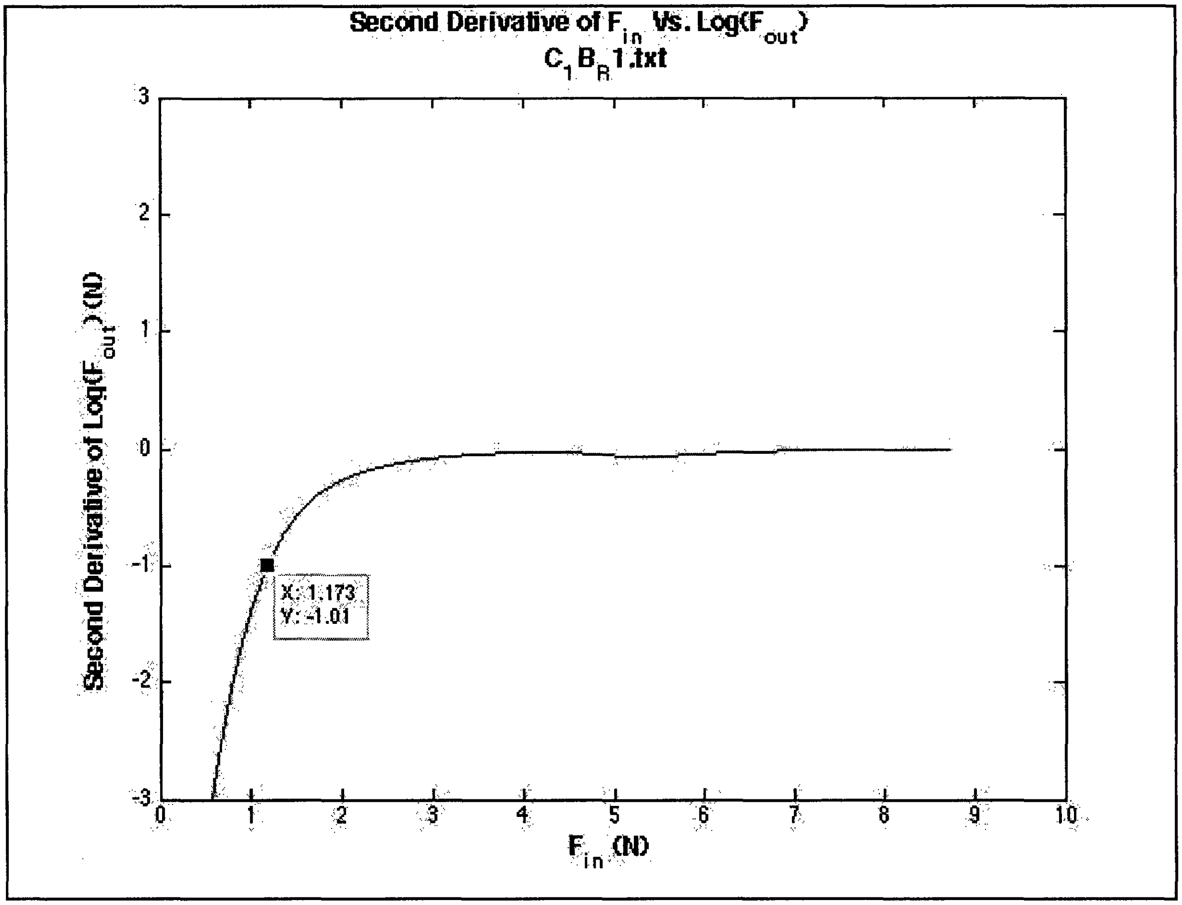


Figure 3-33: Second derivative of Log(F_{in}) vs. F_{out} showing buckling transition point for nylon filament

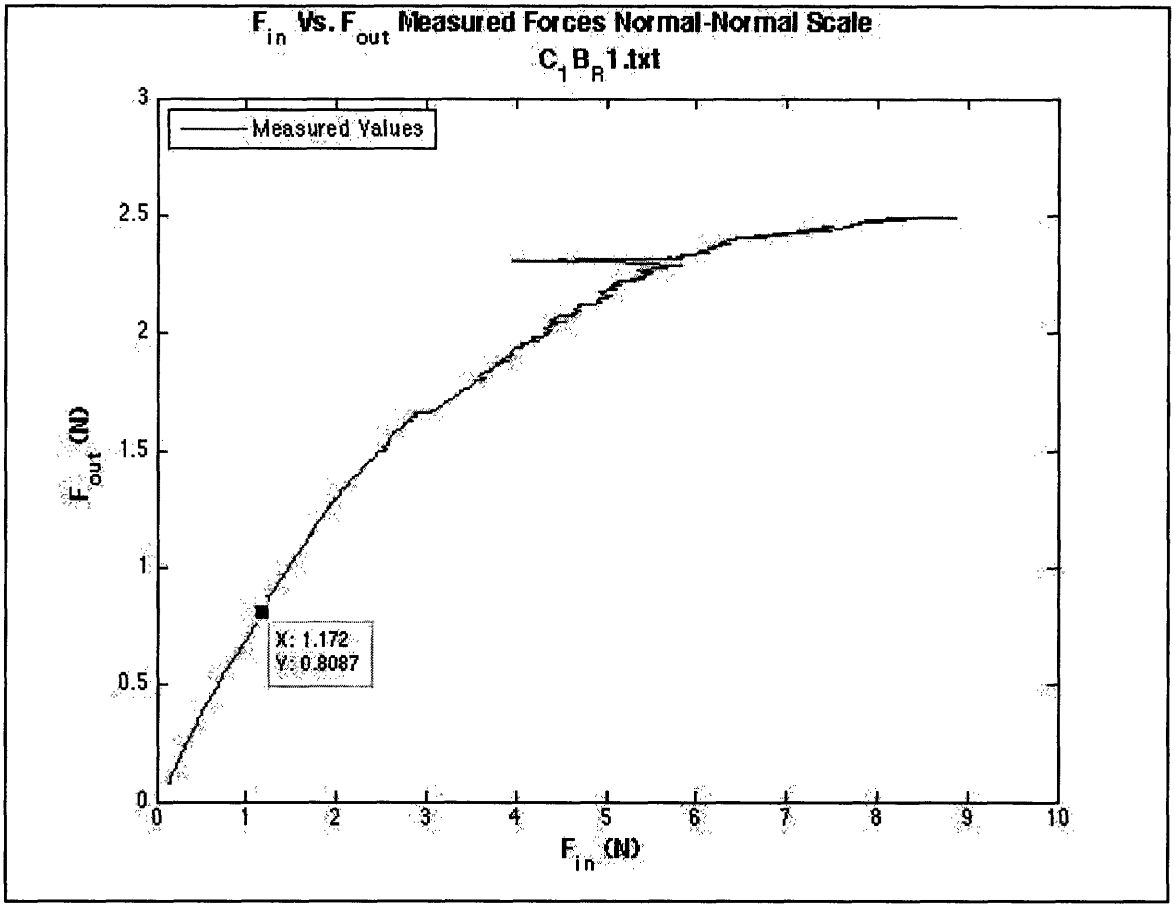


Figure 3-34: F_{in} vs. F_{out} with buckling transition point marked for nylon filament

Table 3-17: Log-based buckling point values for zeroed forces in nylon filament

	F_{in} (N)		F_{out} (N)		Position (mm)		F_{in} /F_{out}	
Design	Average	Rank	Average	Rank	Average	Rank	Average	Rank
1A	0.964	4	0.728	2	2.087	4	1.331	6
1B	0.923	5	0.738	1	2.076	5	1.248	7
1C	0.848	7	0.634	7	1.958	7	1.332	5
1D	1.062	3	0.654	5	2.284	3	1.592	2
2A	1.100	2	0.708	3	2.475	1	1.577	3
2B	1.111	1	0.688	4	2.385	2	1.639	1
2C	0.747	8	0.599	8	1.835	8	0.779	8
2D	0.919	6	0.651	6	2.073	6	1.420	4

Table 3-18: Log-based buckling point values for zeroed forces in PTFE filament

	F_{in} (N)		F_{out} (N)		Position (mm)		F_{in} /F_{out}	
Design	Average	Rank	Average	Rank	Average	Rank	Average	Rank
1A	1.057	3	0.891	3	0.685	8	1.207	3
1B	1.157	2	0.897	2	0.792	3	1.317	1
1C	1.012	5	0.882	5	0.749	6	1.150	6
1D	0.936	8	0.818	7	0.709	7	1.145	7
2A	1.263	1	1.067	1	0.910	1	1.206	4
2B	1.004	6	0.807	8	0.821	2	1.232	2
2C	1.028	4	0.870	6	0.758	5	1.191	5
2D	0.977	7	0.891	4	0.770	4	1.103	8

Table 3-19: Log-based buckling point values for measured forces in nylon filament

	F_{in} (N)		F_{out} (N)		Position (mm)		F_{in} /F_{out}	
Design	Average	Rank	Average	Rank	Average	Rank	Average	Rank
1A	1.036	2	0.726	2	2.087	2	0.704	3
1B	0.984	5	0.740	1	2.072	3	0:753	2
1C	0.925	7	0.638	6	1.932	6	0.695	4
1D	1.004	3	0.609	7	2.014	5	0.606	8
2A	0.989	4	0.659	4	1.916	7	0.627	6
2B	1.073	1	0.662	3	2.093	1	0.615	7
2C	0.589	8	0.458	8	1.325	8	2.613	1
2D	0.983	6	0.655	5	2.044	4	0.669	5

Table 3-20: Log-based buckling point values for measured forces in PTFE filament

	F_{in} (N)		F_{out} (N)		Position (mm)		F_{in} /F_{out}	
Design	Average	Rank	Average	Rank	Average	Rank	Average	Rank
1A	1.240	4	0.889	2	0.694	8	0.725	4
1B	1.278	2	0.885	3	0.735	6	0.691	7
1C	1.277	3	0.879	5	0.757	3	0.689	8
1D	1.084	8	0.813	7	0.726	7	0.751	2
2A	1.419	1	1.067	1	0.904	1	0.749	3
2B	1.168	6	0.810	8	0.768	2	0.696	6
2C	1.228	5	0.859	6	0.750	4	0.703	5
2D	1.104	7	0.885	4	0.735	5	0.802	1

A linear curve fit was used to study the straight portion of the F_{in} vs. log(F_{out}) curves. The slope and Y-intercept of each curve-fit line were recorded (Table 3-21 and Table 3-22). It was believed that the slope of this section may be related to the important variables in the capstan equation (angular displacement and friction coefficient). To study this, the slope of the curve fit was divided by each variable and their product (Table 3-23 and Table 3-24).

Table 3-21: Parameters for linear curve fit of log(F_{in}) vs. F_{out} in nylon filament

Design	Slope	Y-int
1A	0.15	-0.03
1B	0.14	-0.01
1C	0.15	-0.11
1D	0.15	-0.13
2A	0.16	-0.12
2B	0.17	-0.12
2C	0.19	-0.28
2D	0.14	-0.14

Table 3-22: Parameters for linear curve fit of log(F_{in}) vs. F_{out} in PTFE filament

Design	Slope	Y-int
1A	0.19	0.16
1B	0.19	0.14
1C	0.19	0.08
1D	0.20	0.10
2A	0.18	0.19
2B	0.19	0.07
2C	0.20	0.07
2D	0.20	0.20

Table 3-23: Curve fit slope divided by capstan variables for nylon filament

Design	Slope/(Average Angular displacement)	Slope/(Average Mu)	Slope/(Average Angular displacement*Average Mu)
1A	0.11	0.64	0.45
1B	0.09	0.60	0.38
1C	0.08	0.62	0.34
1D	0.08	0.63	0.33
2A	0.10	0.69	0.40
2B	0.10	0.72	0.43
2C	0.10	0.82	0.43
2D	0.07	0.60	0.30

Table 3-24: Curve fit slope divided by capstan variables for PTFE filament

Design	Slope/(Average Angular displacement)	Slope/(Average Mu)	Slope/(Average Angular displacement*Average Mu)
1A	0.13	2.03	1.41
1B	0.12	2.08	1.31
1C	0.10	2.07	1.13
1D	0.10	2.15	1.13
2A	0.10	1.99	1.13
2B	0.11	2.12	1.25
2C	0.10	2.14	1.10
2D	0.10	2.16	1.05

3.4.3 Drill String Model Compression Analysis

Drill strings used in the petroleum industry undergo similar loading conditions and buckling states as the system being studied. The primary difference is the scale. While drill strings have and L/D in the thousands and the system studied here has and L/D on the order of 200 at most, the oil drill string model appeared to be applicable.

After reviewing prior work as previously discussed, it was determined that the models by He *et al.* [6] and Qiu *et al.* [7] were the most relevant to this study. Upon further comparison between these models, the He model was selected for its simplicity and the inclusion of buckling to explain the behavior of the drill string in the drill hole. Buckling causes increased contact force resulting in a decrease in the transmitted axial force. When the contact force becomes great enough, it will absorb the axial force so the output force becomes constant. This condition is called 'lock-up' in the drilling industry. Equations 2-5 are from He *et al.* and were used in this study (repeated here from Section 1.2 above).

$$F_{icr} = \left[\frac{\beta F_{1nbc} EI}{r}\right]^{.5} \tag{2}$$

$$F_{1nbc} = \left[\left(f_b mg \sin\theta + F_a a_i\right)^2 + \left(F_a a_\phi \sin\theta\right)^2\right] \tag{3}$$

$$F_{1hbc} = \frac{F_a^{\,2} r}{4EI} \tag{4}$$

$$F_{1c} = \left\langle \begin{array}{l} F_{1nbc} \ for \ F_a < F_{icr} \\ F_{1hbc} \ for \ F_a > F_{icr} \end{array} \right. \tag{5}$$

Where F_a is the axial input force (F_{in}) applied to the filament, F_{icr} is the critical buckling force where β = 4 and 8 for sinusoidal and helical buckling respectively, F_{1nbc} is the radial contact force per unit length for a non-buckled filament, F_{1hbc} is the radial contact force per unit length for a helically buckled filament, and F_{1c} is the generalized radial contact force per unit length of the string. For the complete list of variable definitions, see the List of Symbols.

The effect of the change in azimuth (direction of wellbore with respect to true north) is not relevant to this study, so a_ϕ here is zero. The inclination build rate (a_i) was approximated as 1/R where R is the average radius of all the bends in each individual tube. Also, $f_b mg$ sinθ was found to be four orders of magnitude smaller than F_a a_i, so Equation 3 simplified to:

$$F_{1nbc} = \frac{F_a}{R} \tag{6}$$

MODEL 1

This equation and F_{1hbc} represent the contact forces in curved and straight sections of the tube respectively. A first order approximation of the contact forces was calculated by assuming:

$$F_{1c} = F_{1hbc} \quad \text{for all } F_a \tag{7}$$

The total contact force ($F_{contact}$) was obtained by multiplying F_{1c} by $\mu L_{straight}$ where $L_{straight}$ is the length of the long straight section of the design tube and μ is the coefficient of friction. Substituting Equation 4 into 7 produces:

$$F_{contact1} = \frac{\mu L_{straight} F_a^{\ 2} r}{4EI} \tag{8}$$

$F_{out\_predicted}$ can then be found by subtracting $F_{contact}$ from F_{in} resulting in:

$$F_{out\_predicted1} = F_{in} - \frac{\mu L_{straight} F_a^{\ 2} r}{4EI} \tag{9}$$

This model produced predictions that had similar shapes as the measured values but were of the wrong scale. To adjust for this, $F_{contact1}$ was multiplied by a scaling factor.

$$F_{1c} = F_{contact1} * CF \qquad \text{Where: } CF_{PTFE} = \frac{1}{2} \tag{10}$$

$$CF_{nylon} = \frac{2}{\pi}$$

With these scaling factors, the model proved to be better than the capstan model but did not capture all contributing factors. Figure 3-35 and Figure 3-36 show examples of this fit.

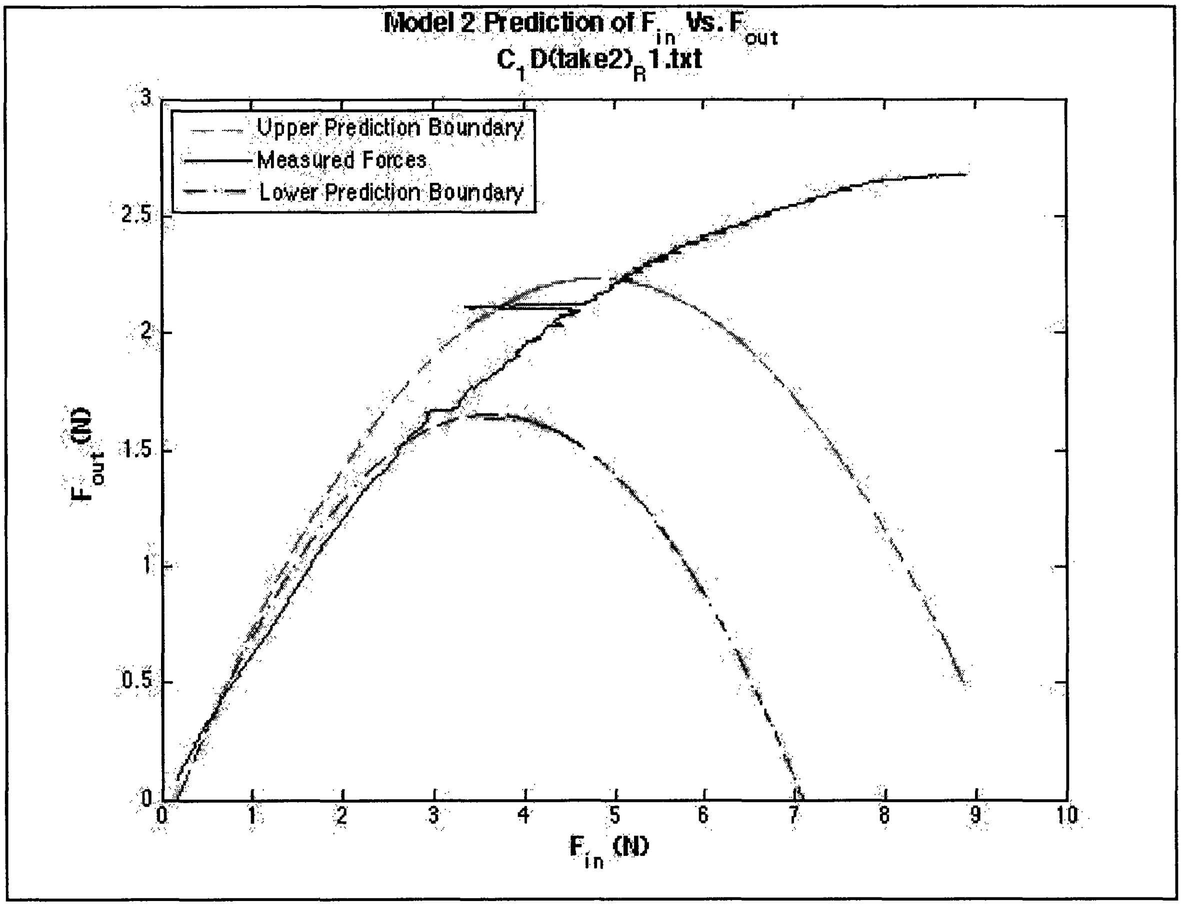


Figure 3-35: Drill string model prediction curves with measured data from nylon filament

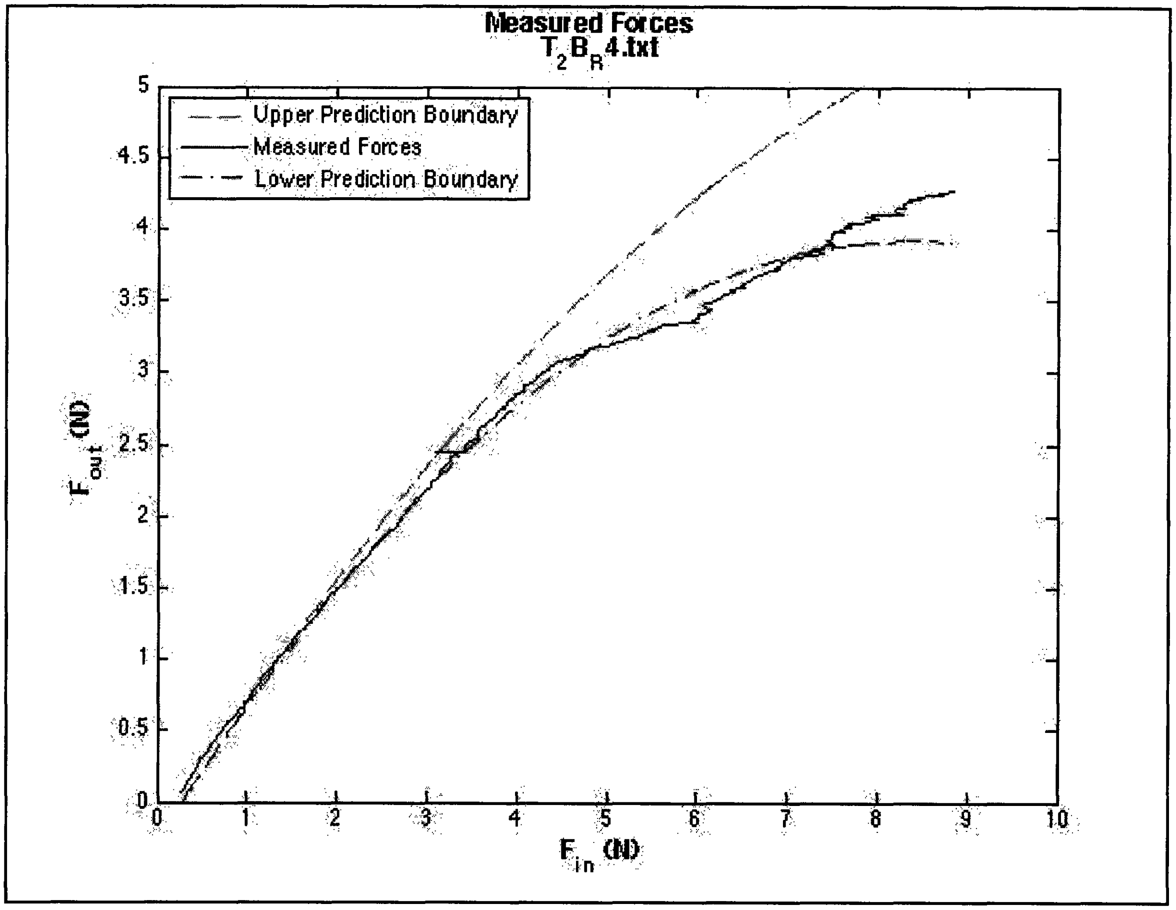


Figure 3-36: Drill string model prediction curves with measured data from PTFE filament

MODEL 2

The model was expanded to include the conditional statement on F_{1c} as stated in Equation 5. In order to include this, it was assumed initially that when the filament buckled, it immediately took on a helical shape (β = 8). Using the scaling factor from above and this value of β, the transition point occurred at an unreasonable load, so β was changed to 4 to assume sinusoidal buckling. The inclusion of this condition resulted in reasonable predictions for the PTFE filament only. With the nylon filament, the buckling point (where the prediction switched from linear to parabolic) was too high and the unbuckled section was far from the actual. This can be observed graphically in Figure 3-37 and Figure 3-38. For comparison purposes, the capstan equation instead of F_{1c} was used to model the pre-buckled state (Figure 3-39 and Figure 3-40).

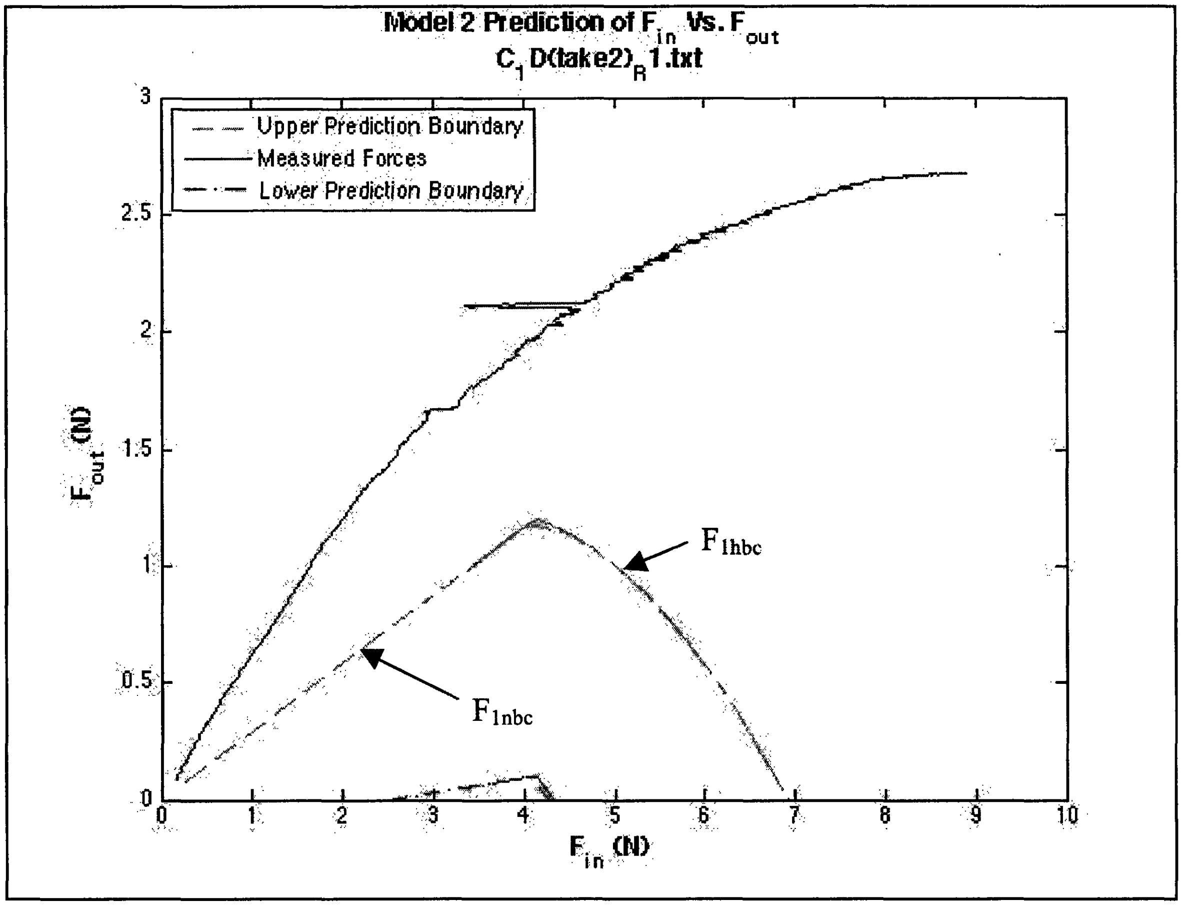


Figure 3-37: Drill string model prediction curves including buckling condition for nylon filament

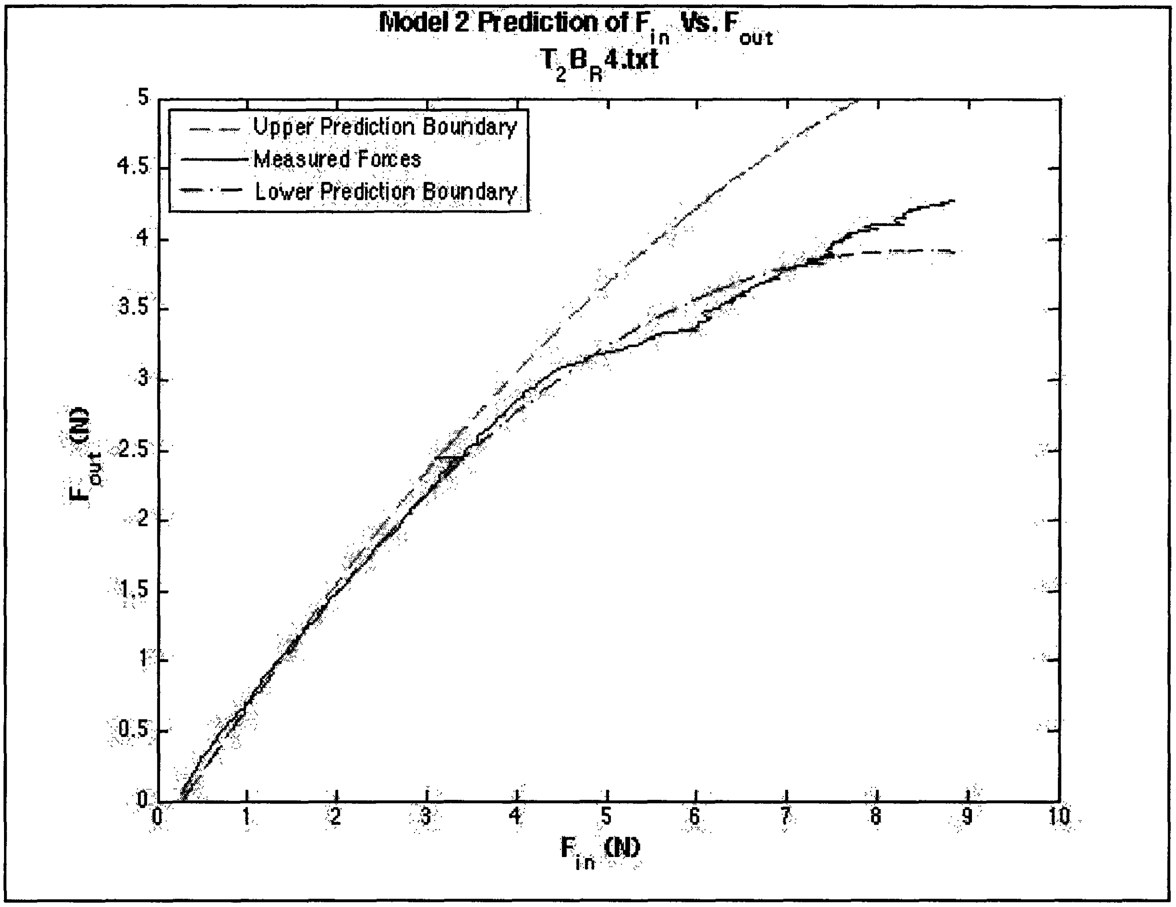


Figure 3-38: Drill string model prediction curves including buckling condition for PTFE filament

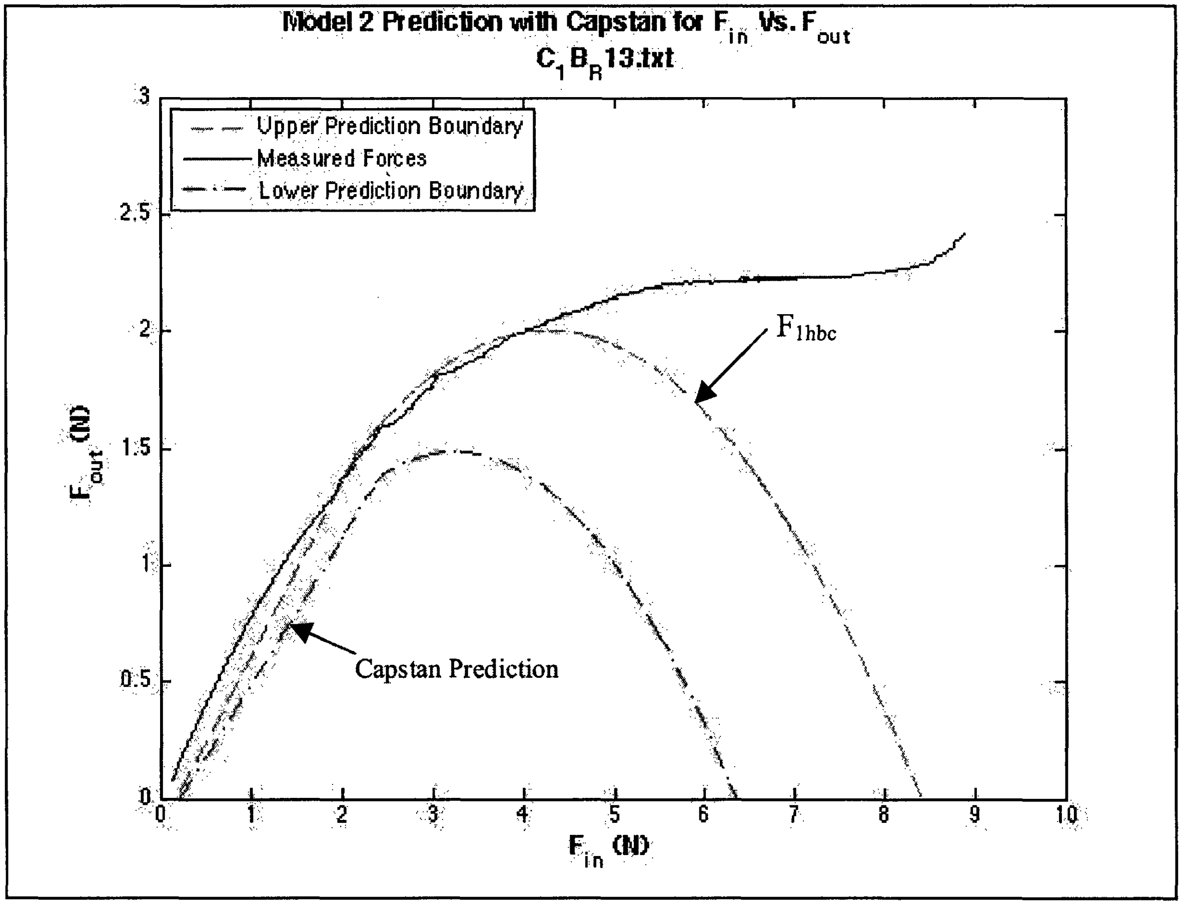


Figure 3-39: Combined capstan and drill string model prediction curves for nylon filament

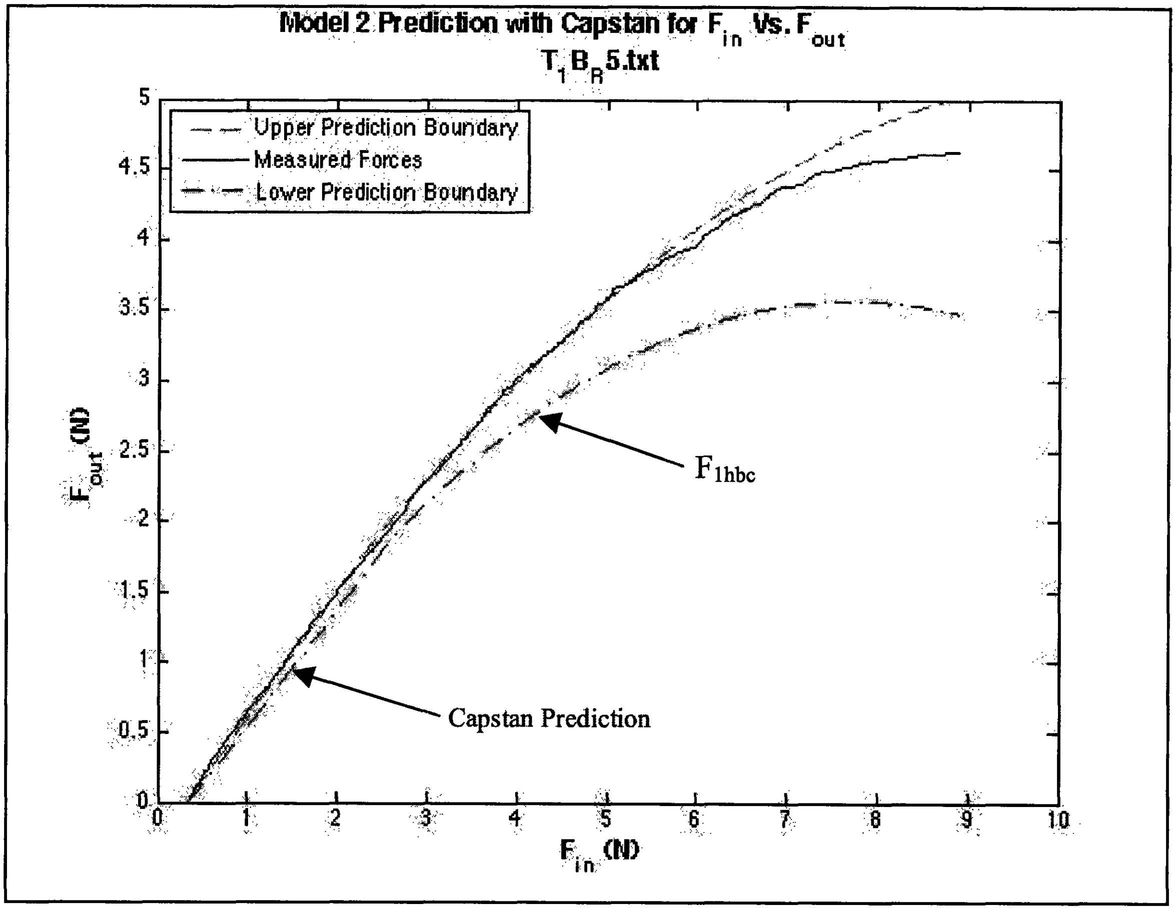


Figure 3-40: Combined capstan and drill string model prediction curves for PTFE filament

Combining F_{1nbc} and F_{1hbc} did not provide an improved approximation. Replacing F_{1nbc} with the capstan equation improved the approximation, although it did not always capture the true shape of the curve at low forces. This model still included two different scaling factors, so further improvements on potential system models were sought.

A sinusoidal buckling contact force model developed by Qiu *et al.* (Eqn B.20 of reference) was also checked for potential use. However, when applied to this data, the equation produced a unit contact force of approximately 1E-7 N/m. This value was many orders of magnitude too small, so it was not investigated further.

MODEL 3

The model was further improved by dividing the contact forces from the two different tube sections, straight and curved.

$$F_{contact\_curved} = \mu L_{curved} F_{1nbc} \tag{11}$$

$$\cdot F_{contact\_straight} = \mu L_{straight} F_{1hbc} \tag{12}$$

This model assumes continuous helical buckling and no buckling in the straight and curved sections of tubing respectively. This assumption eliminates the linear section of the prediction and the need of a transition or buckling point.

Another model improvement reduced the contact length (L_{curved}) of the filament accounting for where it switches from one side of the tube to the other. This transition length ($L_{transition}$) was subtracted from L_{curved} to get the true contact length of the filament. The actual transition length depended not only on the filament diameter and radial clearance in the hole but also the applied compressive force, moment of inertia and modulus of elasticity of the filament. To simplify this, $L_{transition}$ was approximated as the transition length if the filament were in tension, a value that depends only on the filament diameter, radial clearance, and curvature of the tube.

Both of these assumptions were tested and found to be valid (See Glass Tube Tests to Verify Buckling Mode Assumption). Substituting Equations 6 and 4 respectively into 11 and 12 produces:

$$F_{contact\_curved} = \mu(L_{curved} - L_{transition})F_{1nbc} \tag{13}$$

$$F_{contact\_straight} = \frac{\mu L_{straight} F_{in}^{2} r}{4EI} \tag{14}$$

These forces correspond to the force absorbed by the different sections of tubing. When subtracted from the input force to predict the output force the model was completed:

$$F_{out\_predicted3} = F_{in} - (F_{contact\_curved} + F_{contact\_straight}) \tag{15}$$

Substituting Equations 13 and 14 into 15 produced:

$$F_{out\_predicted3} = F_{in} - \left[\frac{\mu \left(L_{curved} - L_{transition} \right) F_{in}}{R} + \frac{\mu L_{straight} F_{in}^{2} r}{4EI} \right] \tag{16}$$

The compression tests where the long straight section was shortened showed that the scaling factor was dependent upon the straight length. One possible value for this factor is $40D_{filament}/L_{straight}$. Due to the near equal values of the PTFE and nylon filaments, it is unclear whether this is correct or not. The true factor may be a constant divided by the length or a factor of a different variable. It could be clarified by performing similar compression tests using filaments with a wider range in the outer diameter. Assuming that the scaling factor is $40D_{filament}/L_{straight}$, the resulting predicted output force equation is:

$$F_{out\_predicted3} = F_{in} - \left[\frac{\mu \left(L_{curved} - L_{transition} \right) F_{in}}{R} + \frac{10 \mu D_{filament} F_{in}^{2} r}{EI} \right] \tag{17}$$

To make this model more graphically accurate, the maximum predicted value of F_{out} was extended to show how it would remain a constant once ‘locked up’. This improved F_{out} prediction model can be seen in Figure 3-41 and Figure 3-42 and can be compared to the same data in Figure 3-39 and Figure 3-40 respectively.

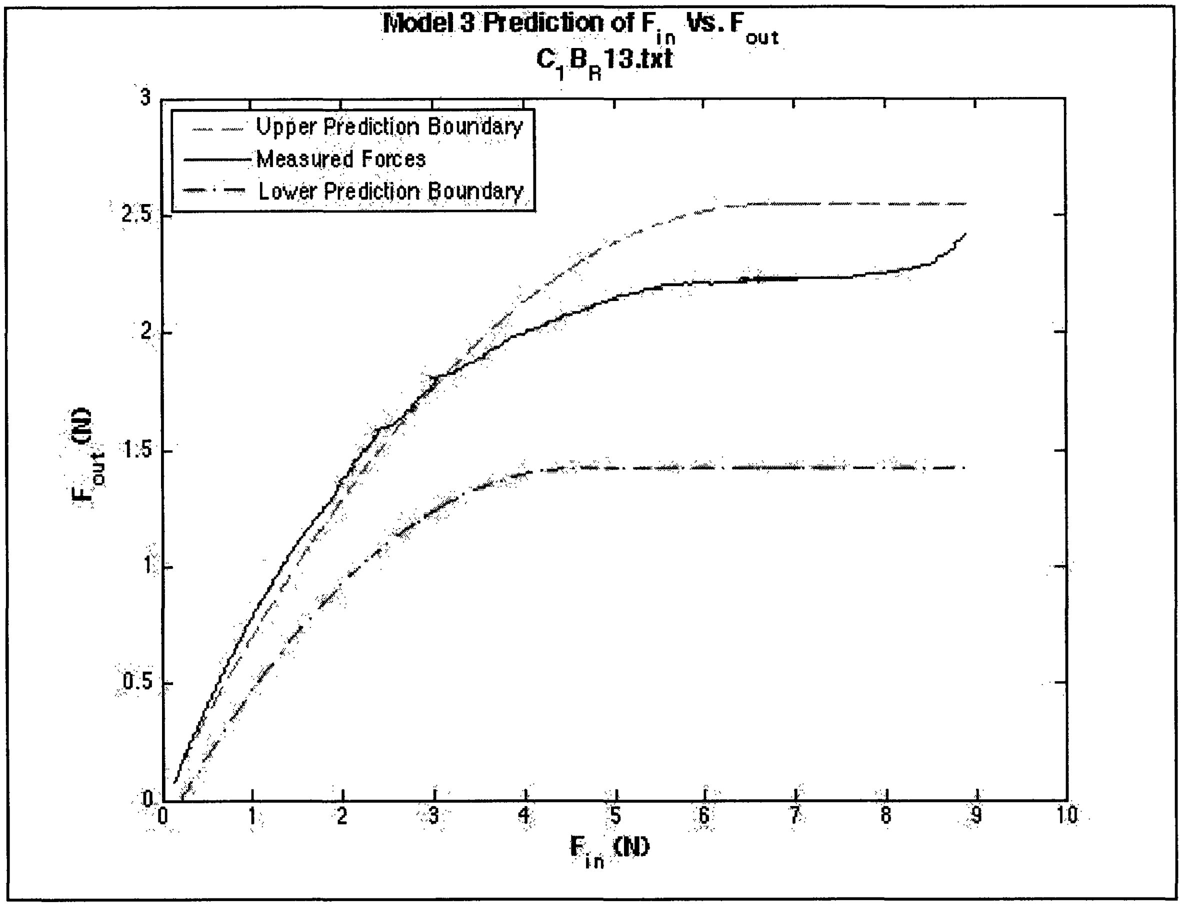


Figure 3-41: Best drill string based model prediction for nylon filament

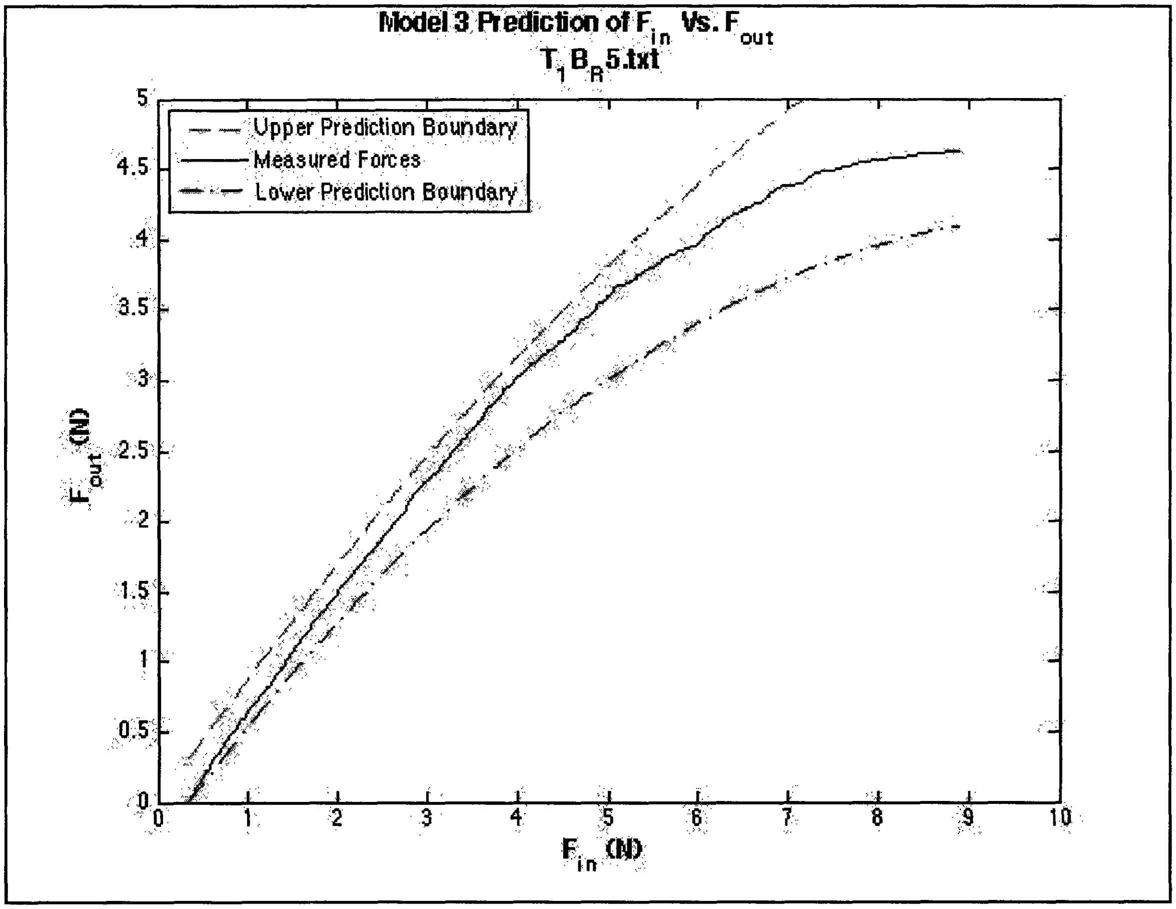


Figure 3-42: Best drill string based model prediction for PTFE filament

This model proved to be far superior to the others investigated. It not only provided a better prediction but it also included a single scaling factor that was valid for both PTFE and nylon. Originally there was an additional scaling factor of 1/2 on the F_{curved} term. Subtracting $L_{transition}$ from $L_{straight}$ eliminated this factor while accounting for greater detail, thus improving the model.

3.4.4 Discussion of Compression Results

The capstan model alone did not accurately predict the output force for this system. It did provide a reasonable first order model when forces are low but broke down quickly. No good transition point was found that could be used to predict the region where the capstan equation remained valid. However, this information may prove useful in future research.

The drill string-based model (Equation 17) proved to be far superior than the capstan model. It included all of the possible parameters of the system, and by including a single scaling factor, matched all of the measured data. It was shown that this factor is a function of $1/L_{straight}$ and is likely $40D_{filament}/L_{straight}$. The filament diameters were so similar that it was not possible to determine with certitude their significance on the scaling factor. This factor was found by trial and error but predicts all of the tests with the steel tubes. Several assumptions used in the derivation of the original drill model may not apply to this system and may be the source for the need of the scaling factor. The first assumption was that the system was sufficiently long that end effects could be neglected. The second assumption was that the buckling pitch was constant. In this system, the pitch appeared to be relatively constant, but because it is so much shorter than a drill string, a variation in pitch could influence the system more. The final possible reason for this scaling factor could be the influence of the corner effects where the filament entered the design tube. Future work should include a more complete study of the exact value of the scaling factor and the cause of it.

The two main assumptions used in the drill string-based model (continuous helical buckling and transition lengths being equal in tension and compression) were studied to test their validity. From a high level perspective, the first appeared valid because the filaments used had a curved memory from being stored on a spool. They were unable to support an axial load without lateral support. These assumptions were studied in further detail and validate in section 3.5.

One interesting question that the capstan work generated was whether an improved exponential-based model exists. It was quite evident that the latter part of the compression curves F_{in} vs F_{out} formed an exponential but the drill model approximated it as a quadratic. Through additional work, a more accurate exponential model based on the measured data may be found.

3.5 Glass Tube Tests to Verify Buckling Mode Assumption

The drill model assumptions were studied to confirm their validity. The primary assumptions were that continuous helical buckling occurred in the straight section of the tube and that transition lengths were equal in both tension and compression. To determine if the helical buckling assumption was correct, it was important to visualize exactly what was occurring inside the tubes during the compression tests. For this purpose clear glass tubes with one and two bends were obtained (Figure 3-43). These tubes had centerline lengths of 127 mm, OD of 7.5 mm, and ID of 1.5 mm. The bend radius and angular displacements can be found in Table 3-25. For better visualization during the tests, a black PTFE monofilament (OD 0.99 mm) was used instead of the nylon or opaque PTFE filaments.

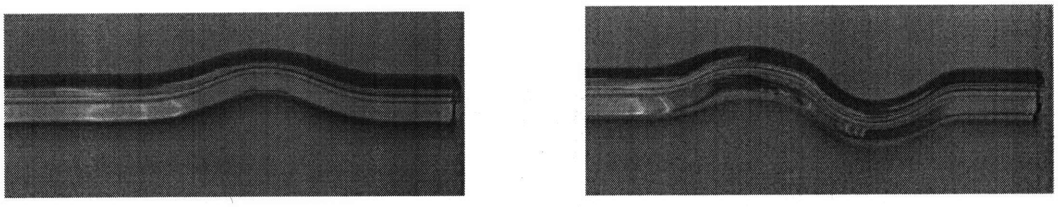


Figure 3-43: Glass tubes with one and two bends

Table 3-25: Bend radius and angular displacements of glass tubes

Glass Design	Bend Radius (mm)	Max Angle (Rad)	Min PTFE Angle (Rad)
1 Bend	12.7	2.618	1.816
2 Bends	12.7	4.056	2.852

The procedures for the compression tests with the glass tubes were the same as with the steel tubes. The filament was advanced at a rate of 0.38 mm/sec until the input force reached 8.9 N. The main differences between the original compression test system and this were that aluminum collars were made to fit over each end of the glass tubes (Figure 3-44 and Figure 3-45). The lower collar screwed directly into the lower load cell, and the filament made direct contact with it during these tests. The filament was held in the same manner as before in an oversized steel tube of OD 2.41 mm and ID 1.96 mm. Another steel tube (OD 1.83 mm, ID 1.52 mm) was fixed in the center of the upper collar. This tube contained the filament between the oversized tube and the glass tube. The glass tube was held in place by a plate with a clamp section. The plate had a tab that went around the glass tube and could be tightened with screws. A typical compression test result can be seen in Figure 3-46. One unusual result seen in these tests, which was not observed in previous tests, was that the input force began increasing before the output force. This was due to slight collapsing of the tube during the bending process. Small ball bearings were dropped through the tubes and it was confirmed that the insidę diameter did not maintain a constant 1.5 mm. This added additional sliding resistance, causing the input force to increase prior to the output force.

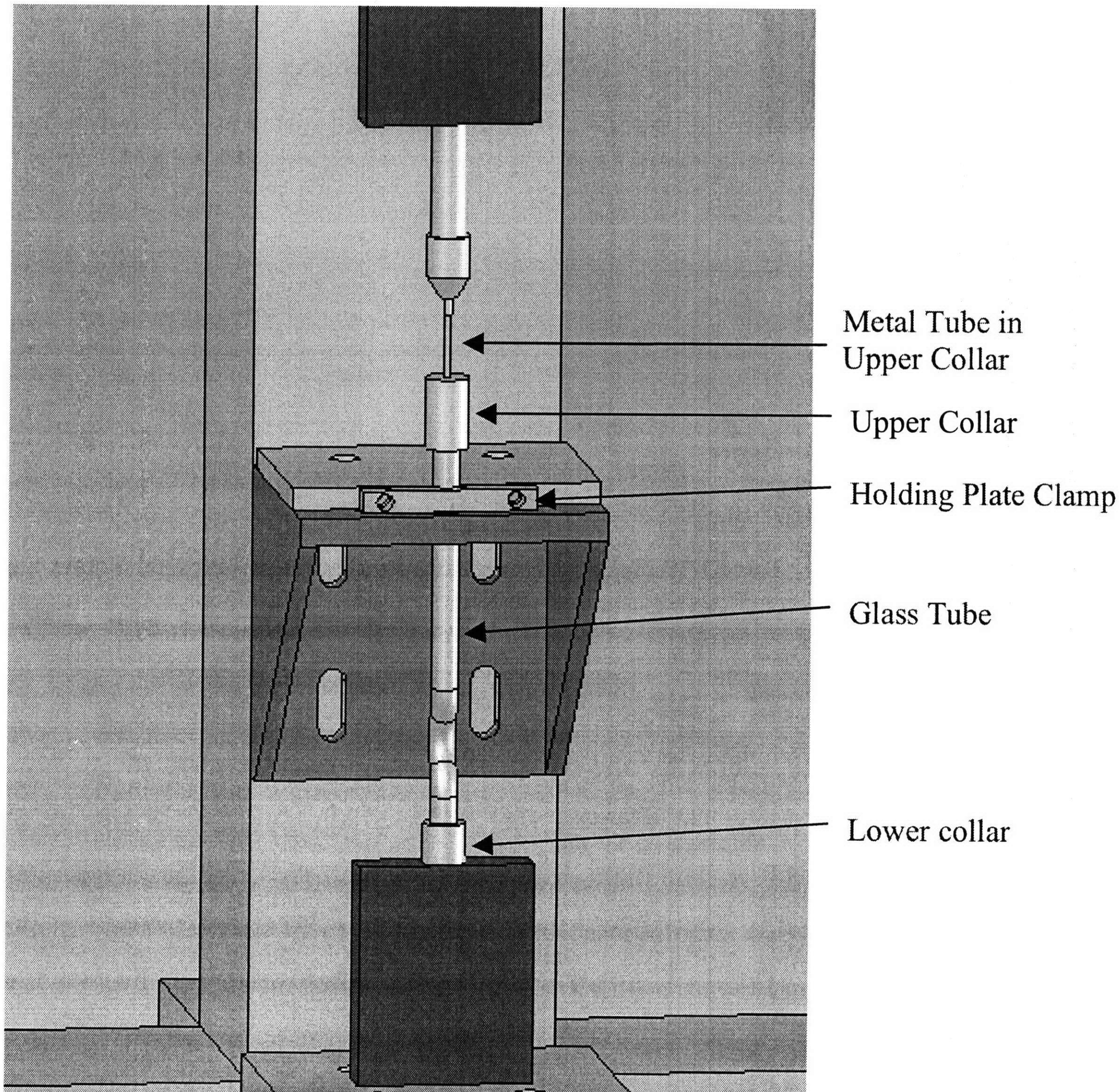


Figure 3-44: Glass tube compression test system

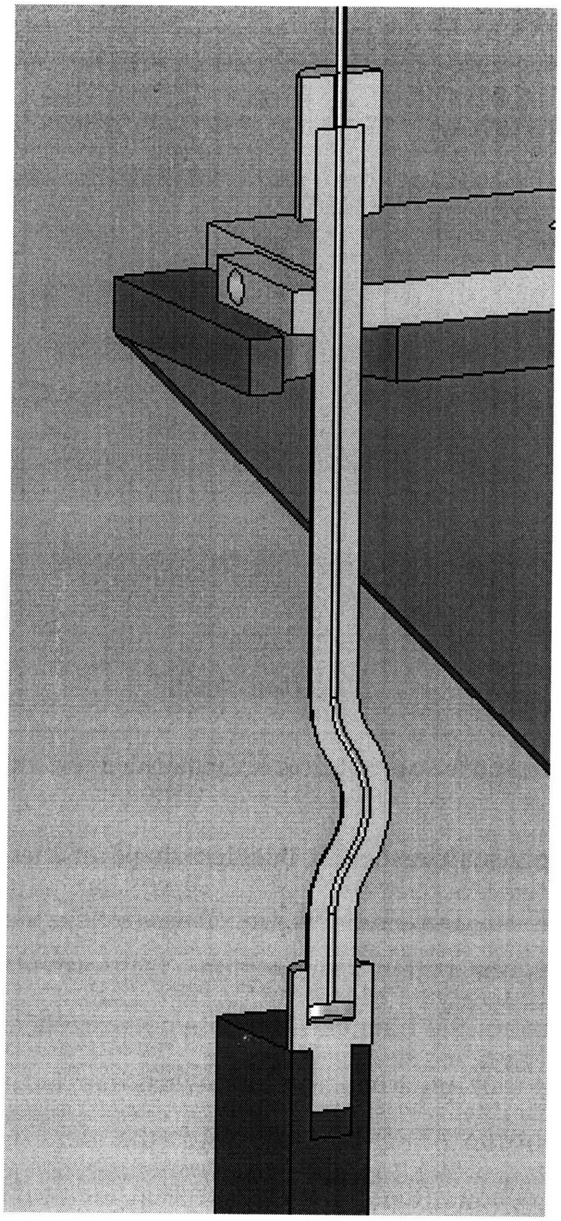


Figure 3-45: Section view of glass tube compression test system

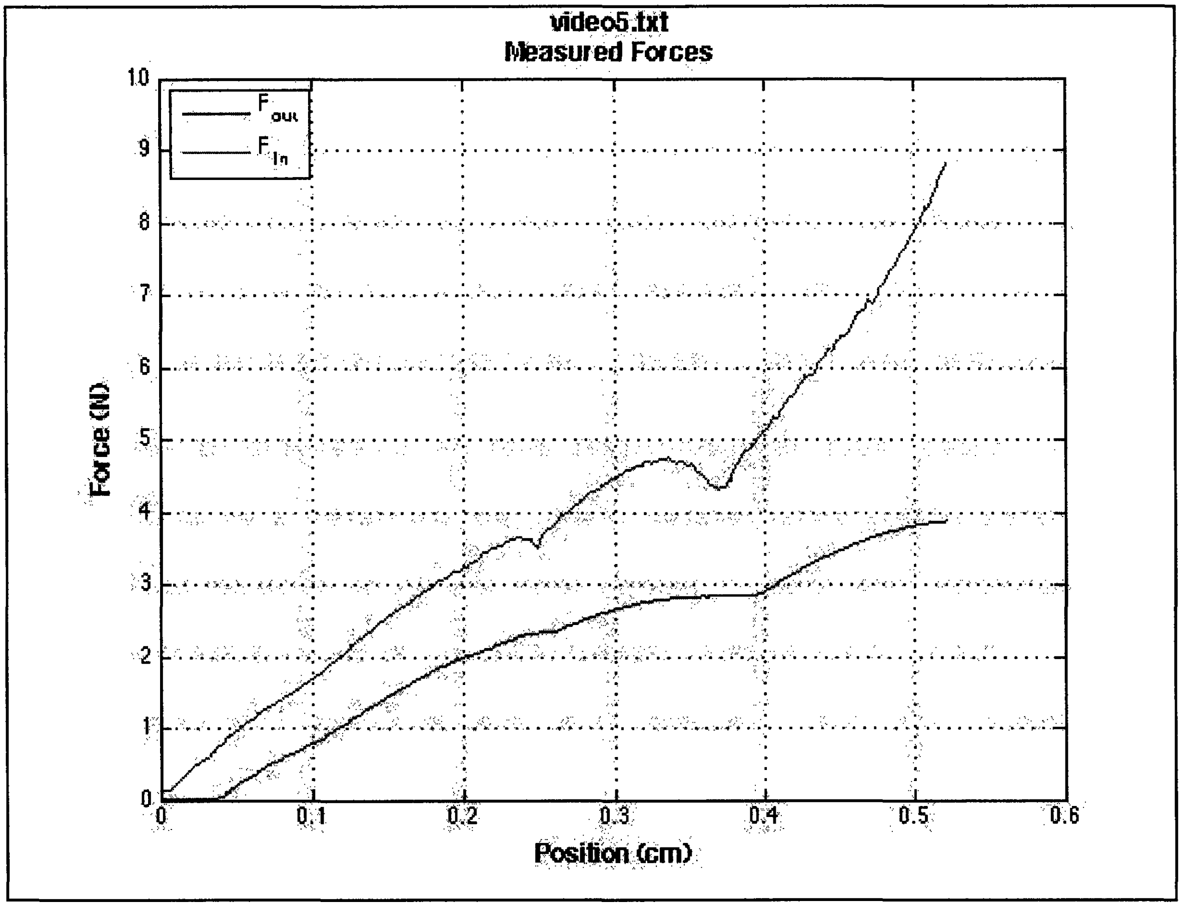


Figure 3-46: Typical measured forces in compression test with glass tubes

In order to better visualize the 3D buckled shape of the filament, blue dye was inserted into the tube and the compression tests were rerun. An opaque PTFE filament was used for these tests for better visualization. This provided an easier means to visualize the buckled shape of the filament. It was hypothesized, and later proven through this test that the filament took on a helical shape when buckled. Figure 3-47 shows this helically buckled shape of the PTFE filament inside the glass tube. The whiter regions correspond to where the opaque filament touched the inside of the tube. These tests also showed the filament path through the curves and how it transitioned from one side of the tube to the other.

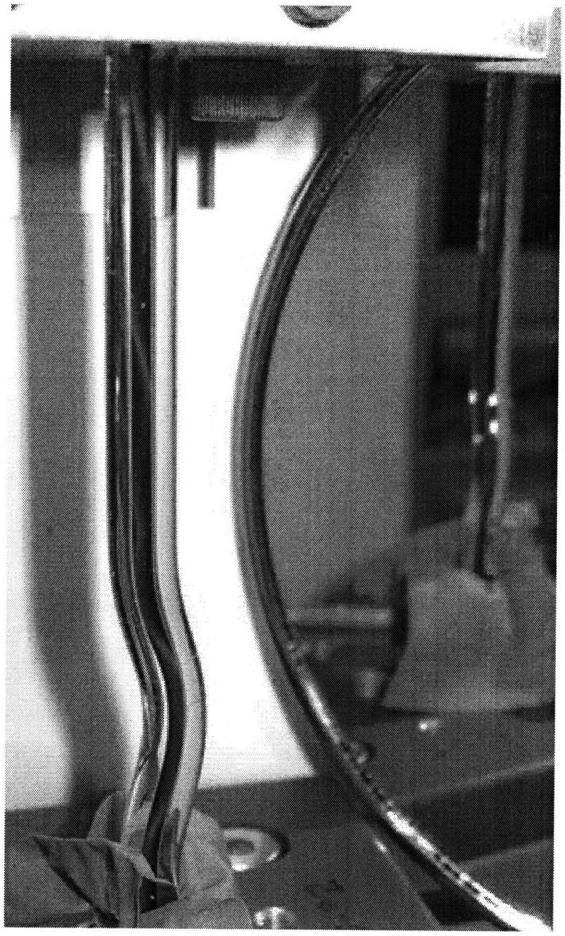


Figure 3-47: Helically buckled PTFE filament inside ink filled glass tube

SolidWorks CAD models and pictures from the tests with the glass tubes were used to check the assumption that the transition lengths in tension and compression were approximately equal. In the glass tube case (where $D_{tube}/D_{filament}$ = 1.51), the actual compression transition length was determined to be ~75% of the tension length (Figure 3-48). In the limiting case when $D_{tube}/D_{filament}$ approaches 1.0, these transition lengths approach equality. For the case of the nylon and PTFE filaments, the diameter ratios were 1.26 and 1.35 respectively. These $D_{tube}/D_{filament}$ values were closer than with the glass tube, making this transition length approximation better than 75%, an acceptable approximation for the purposes of this study.

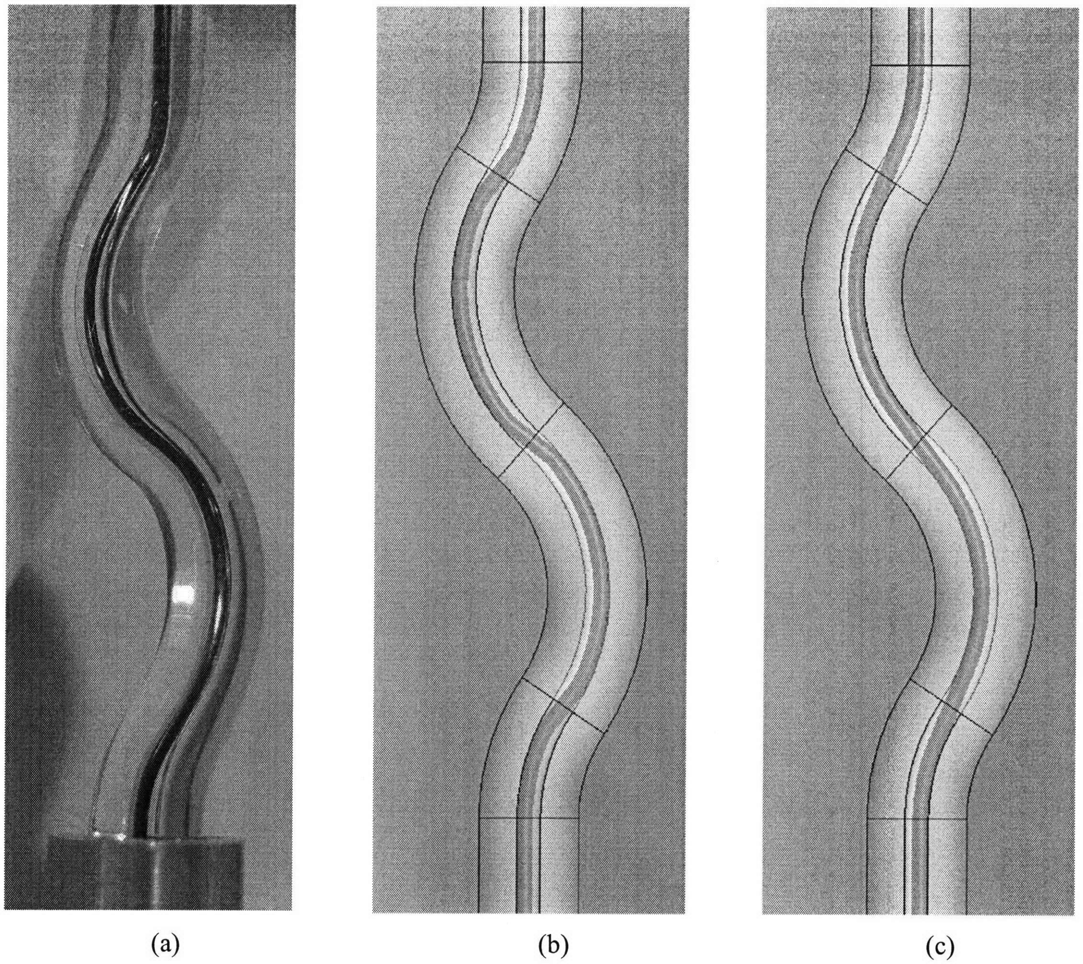


Figure 3-48: (a) Picture of PTFE filament in compression inside glass tube (b) SolidWorks model of filament in compression in glass tube (c) SolidWorks model of filament in tension in glass tube

By including this approximation, one scaling factor was eliminated and it also accounted for the number of bends in a tube. Thus the model included all relevant parameters of the system: friction, bend radius, moment of inertia, modulus of elasticity, radial clearance, number of bends, and angular displacement.

3.5.1 Glass Tube-Filament Friction Coefficient

To study the validity of the drill model on the glass tube system, the coefficient of friction between the glass and PTFE was measured. These tests were performed the same as the previous friction tests. Instead of pulling a steel block, two glass tubes were connected together then weighted down with steel bars to amplify the results (Figure 3-49). The total weight of the glass and bars was 1.74 N.

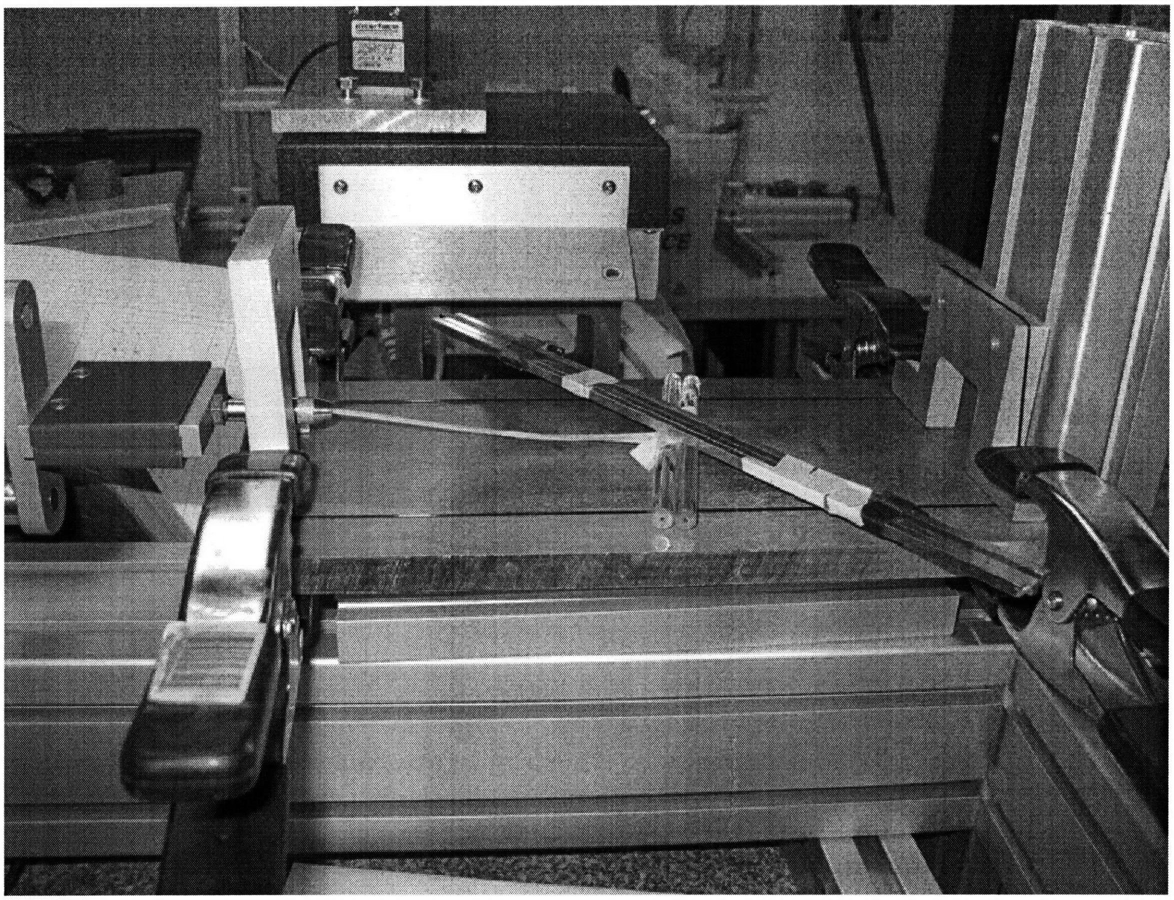


Figure 3-49: Glass-PTFE friction coefficient test with steel rods (diagonal pieces) to add weight

The friction tests were performed at the same rates as the tests performed previously, 0.13, 0.38, and 2.54 mm/sec and repeated three times. The results from one of the tests performed at a rate of 0.13 mm/sec can be seen in Figure 3-50. Table 3-26 shows the average friction coefficients for each rate.

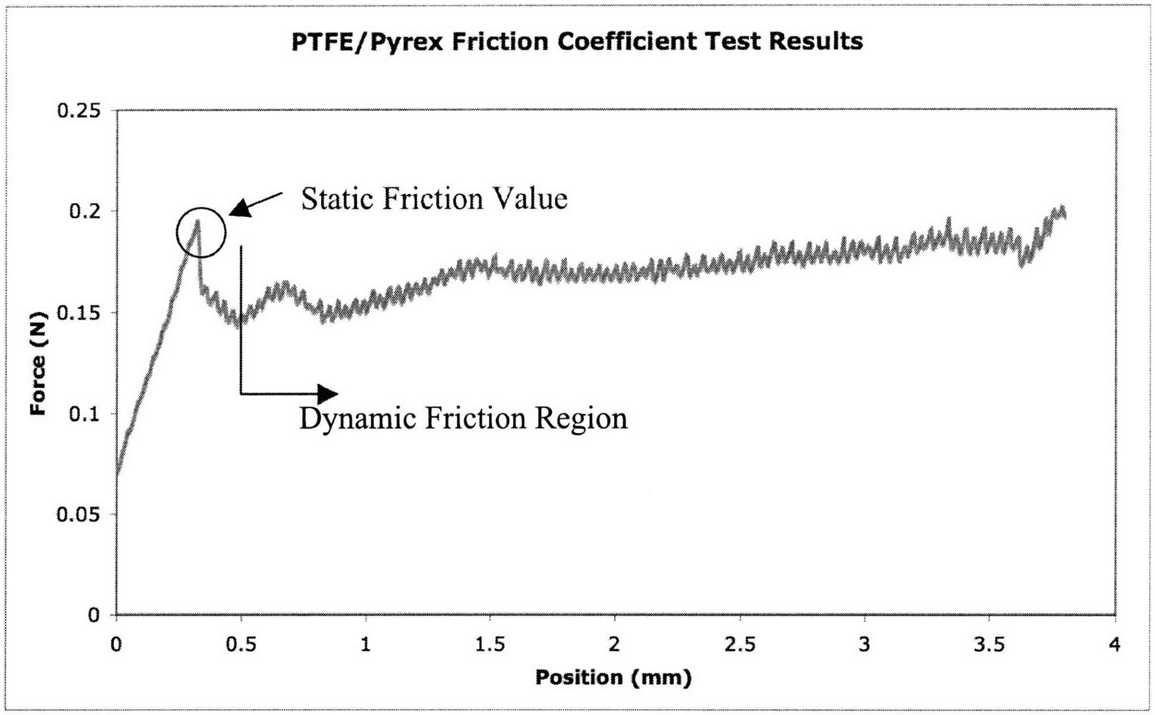


Figure 3-50: Typical friction test result between glass and black PTFE

Table 3-26: Friction coefficients between PTFE and glass at different rates

Rate (mm/sec)	Average Measured Static Coefficient of Friction
0.13	0.0935
0.38	0.0852
2.54	0.1379

3.5.2 Glass Tube Compression Test Analysis

To test the dependence of the diameter and radial clearance on the output, the glass tube compression tests were predicted with the model. Figure 3-51 and Figure 3-52 show typical results with their predictions. The steel tube in the upper collar (visible in Figure 3-44 and Figure 3-45) was included in the original prediction. Interestingly, when this metal segment was ignored, the prediction became much better (Figure 3-53 and Figure 3-54). This suggests that the friction and contact forces of the system were dominated by those in the glass tubes.

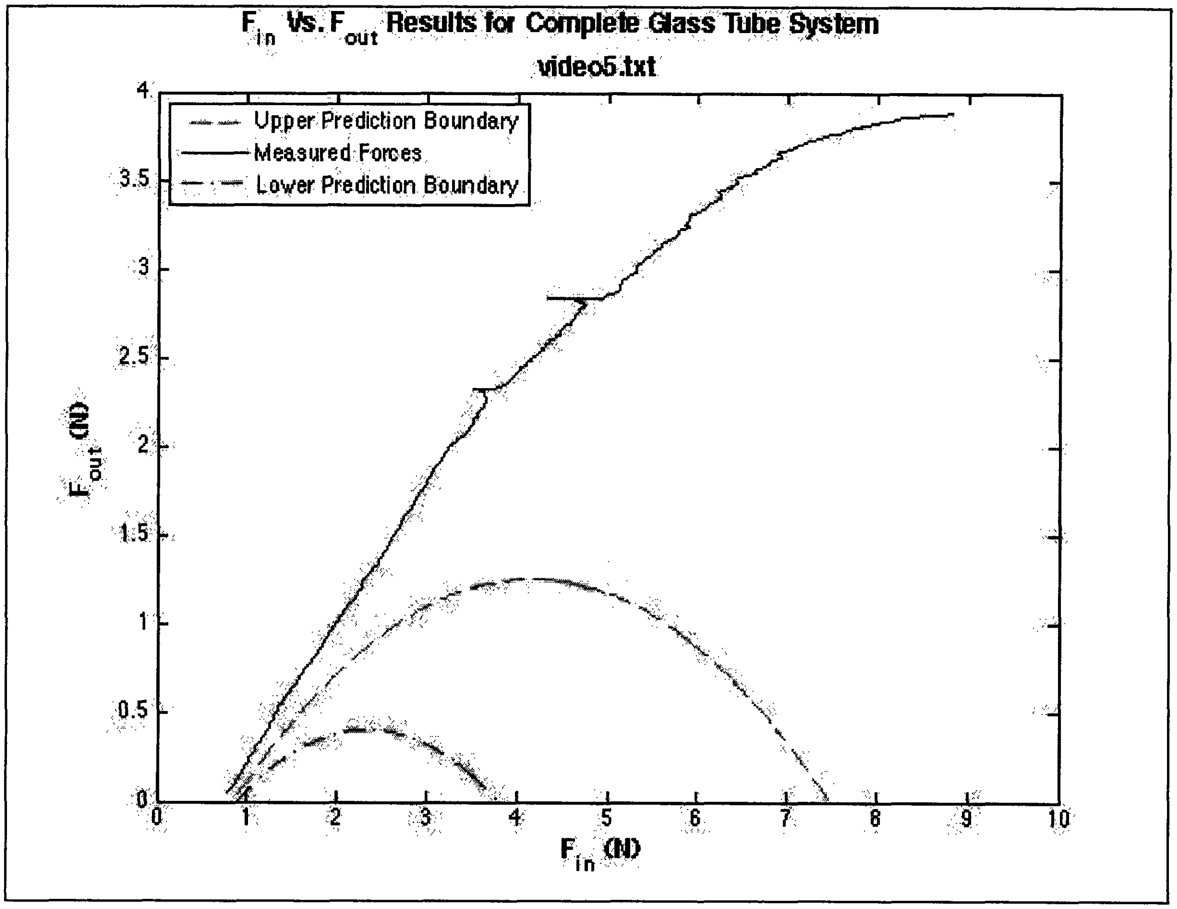


Figure 3-51: Measured and predicted values for forces using PTFE in glass tube with one bend

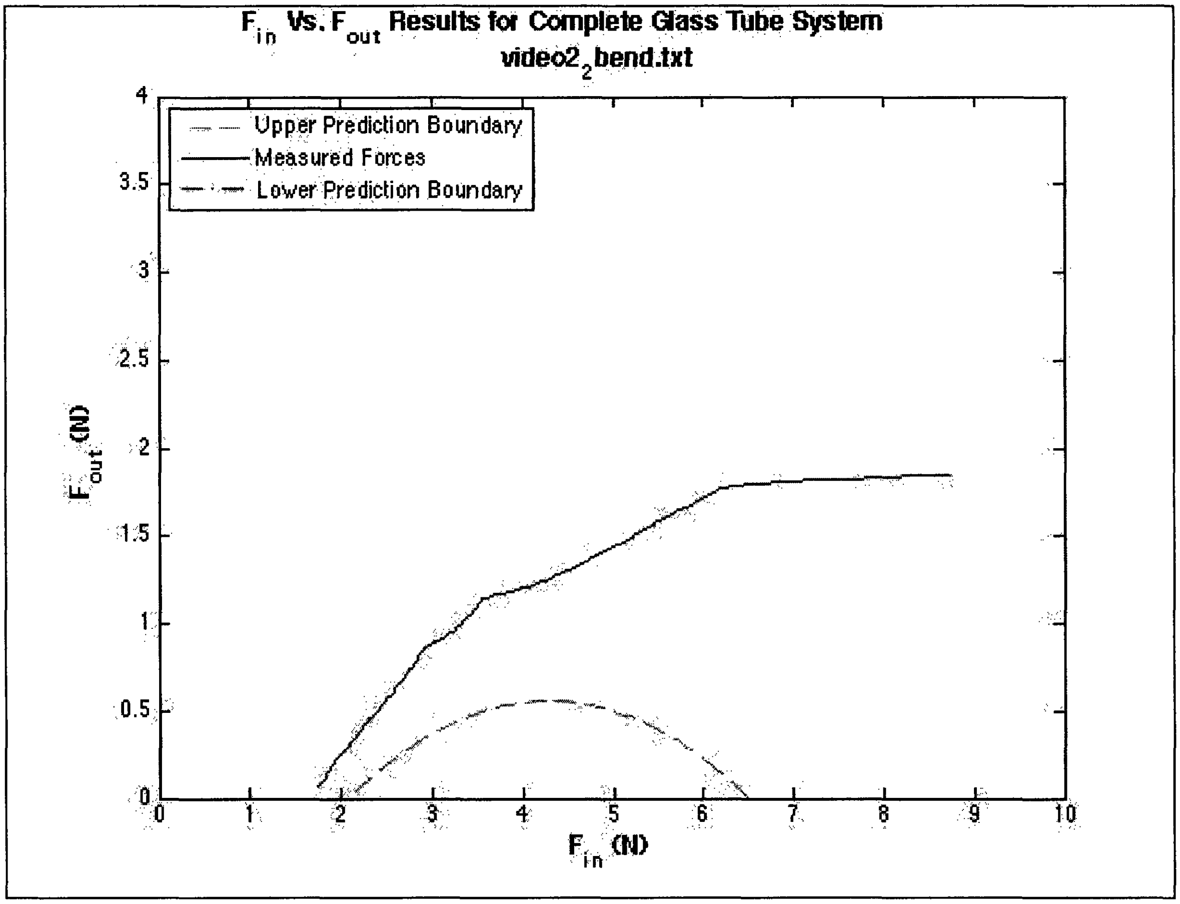


Figure 3-52: Measured and predicted values for forces using PTFE in glass tube with two bend

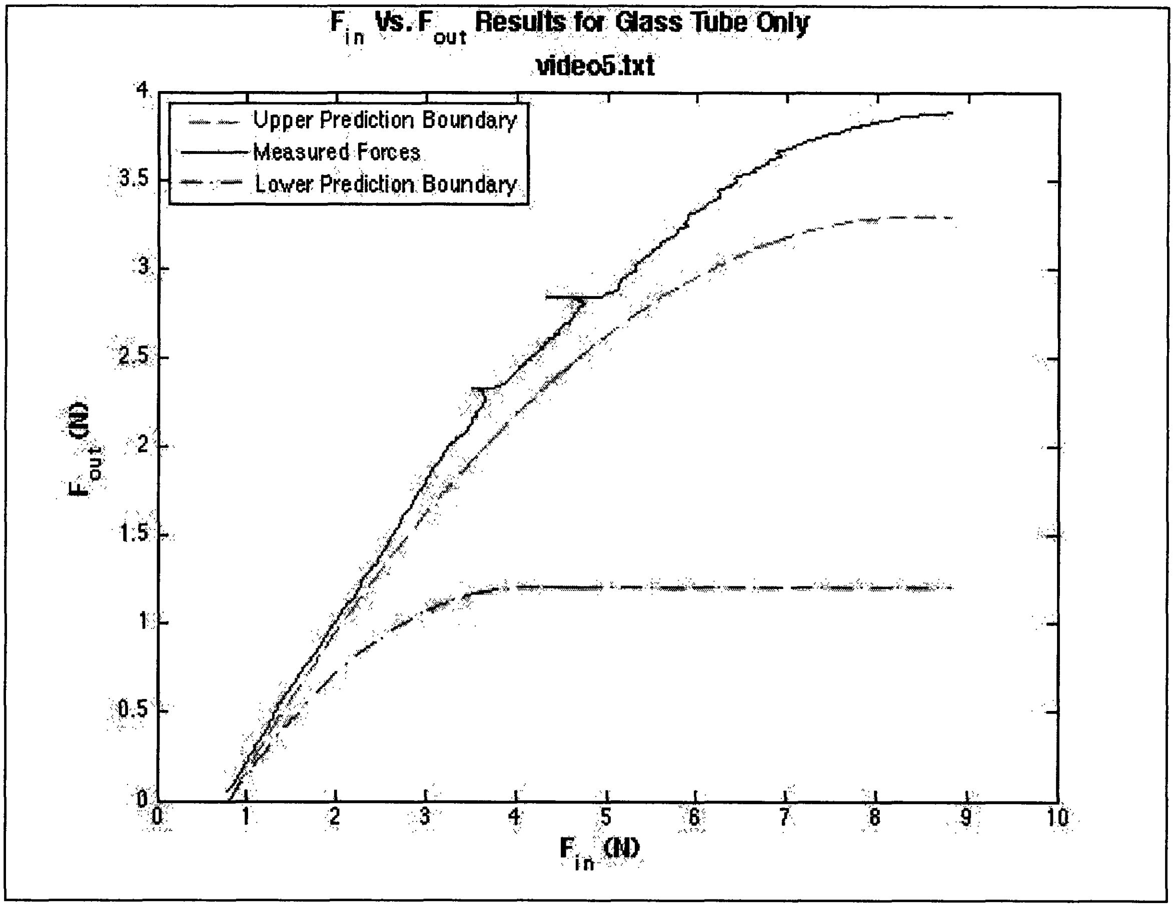


Figure 3-53: Measured and predicted (neglecting metal tubing) values for forces using PTFE in glass tube with one bend

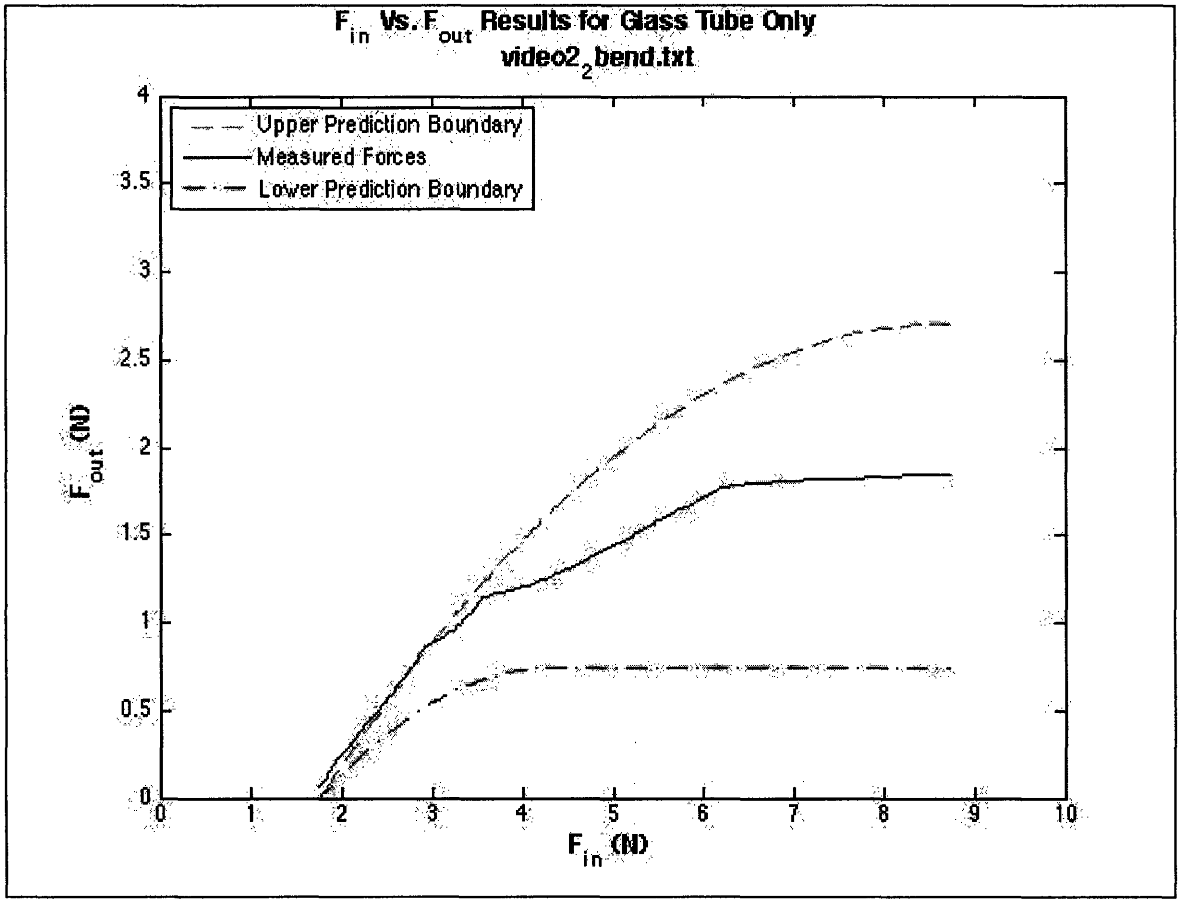


Figure 3-54: Measured and predicted (neglecting metal tubing) values for forces using PTFE in glass tube with two bend

3.5.3 Discussion of Glass Tube Tests

The glass tube tests clearly showed that helical buckling initiated at very low forces in the straight tubing. Results also showed that there was no helical buckling in the curved sections of the tubes and that the length approximation of $L_{transition}$ was roughly the same in both tension and compression. These finding and the good fit of these test support the validity of model and assumptions used to develop it (see section 3.4.3). It was unclear why the inclusion of the metal tubing in the prediction produced a poor fit while its exclusion produced a good fit.

3.6 Measuring and Modeling Conclusions

This work achieved both of the primary goals of the research: to determine the efficacy of the capstan equation in predicting the tensile and compressive forces in a filament, and to determine whether the clutch action discussed above could be created and predicted. Test results showed that the capstan equation provided a reasonable model when the filament was in tension but not when in compression. The capstan equation predicted the tensile forces well for the nylon filament, but not for the PTFE. Additional influences due to hysteresis were suspected to prevent the capstan prediction from matching the measured data. On the other hand, it was found that a petroleum industry based model for drill strings in compression in wellbores developed by He *et al.* [6] accurately predicted the compressive forces in the roughly 150 tests completed. The inclusion of a scaling factor in this model enabled the complete prediction in all the steel tube tests. The model accounted for all of the possible design parameters including geometry of the tube. Using this model, the output forces can be accurately predicted from and used by designers to create a new two-part mechanical linear motion clutch that can be used for catheter/needle insertion systems.

Chapter 4

Prototype Testing

4.1 Motivation

The ultimate goal of this research was to produce a functional prototype that could demonstrate the concept of the clutch-based catheter insertion system. With the original tension and compression tests complete and models generated, the focus quickly transitioned to testing on tissue. Raw chicken breast was chosen for the initial prototype tests. Through the course of these tests, the device was refined and finally tested successfully on deceased pigs.

Prior to the development of the mathematical model, tests were performed on chicken breast sitting on knox gelatin (see Figure 4-1). These tests were performed in the ADMET testing machine and provided insights into challenges that would have to be overcome to produce a fully functional device. The first challenge was that the system included more variables than could easily be tested and analyzed together with that setup. Also, it was observed that needle tip geometry had a significant effect on how the filament interacted with the tissue. For these and other reasons, the system was simplified and tested with just the tubes in the compression setup discussed in Section 3.4.

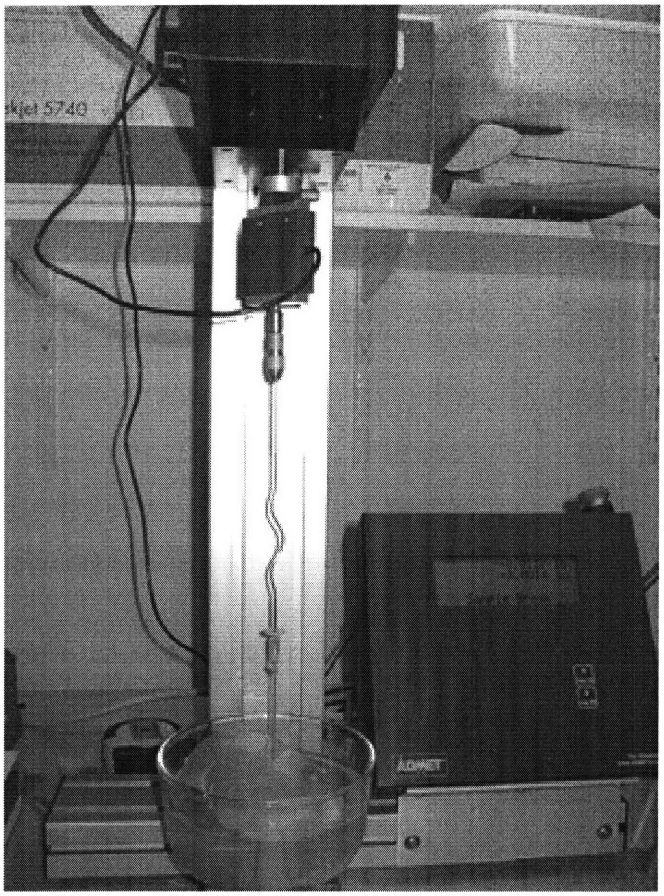


Figure 4-1: Early setup for testing chicken breast on knox gelatin

4.2 Prototype Development

Following the development of the model, tests were again performed on raw chicken. Initially, the original eight tubes tested in tension and compression were used. These tests were performed manually with the chicken on a sheet of acrylic with a small hole drilled in it where the needle could exit. The PTFE filament was used, but proved to produce lock-up forces that were too high for the chicken, resulting in the filament penetrating the tissue without advancing the needle tip.

Initial experiments with the chicken showed that a traditional beveled needle tip caused the filament to exit to the side, resulting in unsuccessful attempts. This was due to the asymmetric filament-tissue contact caused by the needle bevel. To improve this, the traditional needle tip was replaced with a hand-ground needle with two points (Figure 4-2). This brought the filament-tissue contact in line with the needle axis and improved the loading condition of the filament.

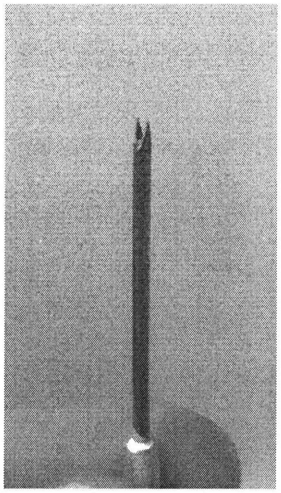


Figure 4-2: Two-point doubly symmetric needlepoint used in prototype tests

With the addition of this needle tip, improvements in performance were observed. Instead of the filament always piercing the chicken without advancing the needle, the clutch would occasionally engage and advance the system. On a few occasions, the system worked as intended and advanced the needle until it penetrated the chicken. Unfortunately, the majority of the tests were not successful. The most common failure remained the same; the filament penetrated the chicken without advancing the needle because the clutch was not engaging.

These prototypes were shown and demonstrated to a general surgeon who provided suggestions to improve the device. The first included adding a feature(s) to increase both the sliding resistance of the needle through the tissue and the tactile feedback to the physician. This added feature would also stop the needle after the clutch disengaged. He also suggested testing the system on pig ribs, which have similar muscle and peritoneum as humans. Lastly, he acknowledged the systems' potential use in accessing the abdominal cavity for laparoscopic procedures. Over-advancement and incorrect placement injuries are common in these procedures because existing veress needles provide insufficient feedback to the physician.

After these suggestions were offered, pig ribs were tested using the original tube designs. Unfortunately, most of these tests ended unsuccessfully, with similar results as the chicken tests. One significant problem observed during these tests was that the needles did not penetrate the peritoneum. The needle would advance through much of the muscle tissue only to stop before the peritoneum. The peritoneum had become very tough and dry, and required high forces to puncture. The needle caused the peritoneum to tent away from the muscle tissue without being pierced. This problem may be reduced or eliminated by using fresher ribs, but because of it, no further tests were performed with ribs. Another problem discovered was that there were several layers of fat intermixed with the muscle tissue. This caused the filament to deploy early and pierce the remaining tissue without the needle. After these tests were completed, it became evident that significant changes needed to be made to achieve a functional system.

With the information gathered from previous tests and the prediction model, nine new tube shapes were designed and made. These shapes had much lower lock-up thresholds than the original eight. The shape parameters of those tubes can be found in Table 3-8. In addition to the new tubes, many other variables were changed and iterated. The following is a list of the different parameters tested in chronological order.

- Needle tip bevel angle- Two different bevel angles (~45 and ~90 degrees)
- Filament tip- Steel ball bearings (OD 1.0 mm) were added to the end of the filaments to increase the surface contact area thus reducing the contact pressure on the tissue
- Filament- Original PTFE filament was replaced with a PTFE-coated spring wire guide (OD 0.76 mm) that had lower bending stiffness once central nitinol wire was removed
- Filament tip- A tube (OD 1.4 mm, ID 0.9 mm) was nested and adhered to over the filament tip to increase its bending stiffness and maintain axial loading
- Needle surface- Needles were sandblasted to increase the sliding resistance in the tissue and reduce the likelihood of needle over-advancement
- Needle tip- A three point needle tip was created to increase the stable contact zone of the filament tip on the tissue (Figure 4-3)

- Tube and needle diameter- Larger tubing (OD 2.1 mm, ID 1.8 mm, 10 mm long) with same shape as the three best of the nine later designs
- Filament tip- Larger steel ball bearings (OD 1.6 mm) to further increase the contact area and fit better in the larger tubing
- Number of bends- The number of bends was increased from 2 to 3 and 4 for the designs (of the nine) with the lowest lock-up thresholds (Table 4-1)

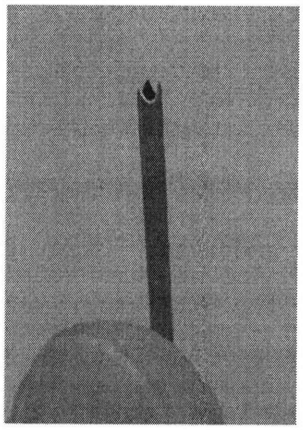


Figure 4-3: Three point needle tip used in prototype tests

Table 4-1: Improved tube shapes

Design	# of Bends	Avg. Bend Radius (mm)	Angular displacement (rad)
R0.53_A525	3	13.462	9.16
R0.53_A700	4	13.462	12.22
R0.6_A525	3	15.24	9.16

4.3 Prototype Test Results with Chicken

The most successful design tested on raw chicken breast included the PTFE spring wire guide (OD 0.8 mm) with the steel ball tip (OD 1.6 mm) and plastic tubing (OD 2.1 mm, ID 1.8 mm, 10 mm long) over the end, larger steel tubing (OD 2.1 mm, ID 1.8 mm) bent from the acrylic form R0.6_A525 (15.2 mm bend radius, 3 bends, 525 degree angular displacement), and a three-point needle tip. One successful test with this system can be seen in Figure 4-4. In this figure, the needle tip is just visible where the filament exits the tissue.

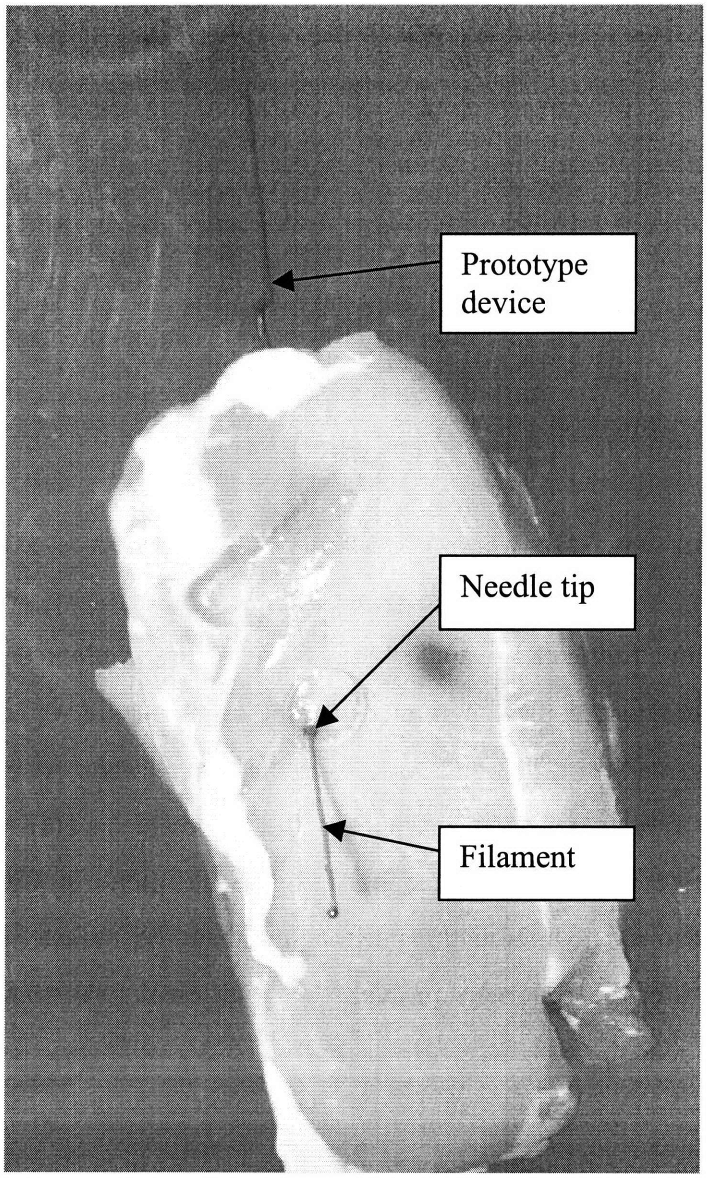


Figure 4-4: Demonstration of prototype device that deployed after passing through raw chicken breast

Through all of these iterations, slight improvements were achieved. By the end, the best designs worked with about a 50% success rate. The most common mode of failure in this design consisted of the filament tip jamming inside the needle tip, delaying or preventing deployment of the filament after the needle tip exited the tissue. Upon further inspection, it was found that a small piece of tissue cored by the needle had wedged between the filament tip and inside of the needle, inhibiting the motion of the filament. Because this design showed consistent locking of the clutch, it was deemed superior than previous designs.

4.4 Prototype Test Results with Deceased Pig

To obtain more data, prototypes were tested on deceased pigs. Physicians at Massachusetts General Hospital (MGH) inserted the needles into the insufflated abdominal cavity of the pigs. An endoscope was used to view each attempt from the inside and record the results. The three most successful tube shapes (see Table 4-1) and filament used for the tests with one minor modification. The inside diameter of the tubing was reduced from 1.8 to 1.7 mm. This change decreased the radial clearance between the steel ball bearing tip and the inside of the needle.

Results from these tests were very encouraging. The decrease in radial clearance improved release of the filament upon penetration by reducing the problem of tissue wedging between the filament tip and needle. The major problem experienced during these tests was piercing the peritoneum. The three-point needle tip used did not have large cutting edges, making it difficult for the needle to penetrate. Also, because the tips were hand made, they weren't as sharp as standard needles. The endoscope videos showed the needles consistently reached but struggled to pierce the peritoneum. Much more force was required to pierce the peritoneum than the muscle which caused the needle to advance into the abdomen on average ~1.75 cm before stopping. This can be seen in Figure 4-5.

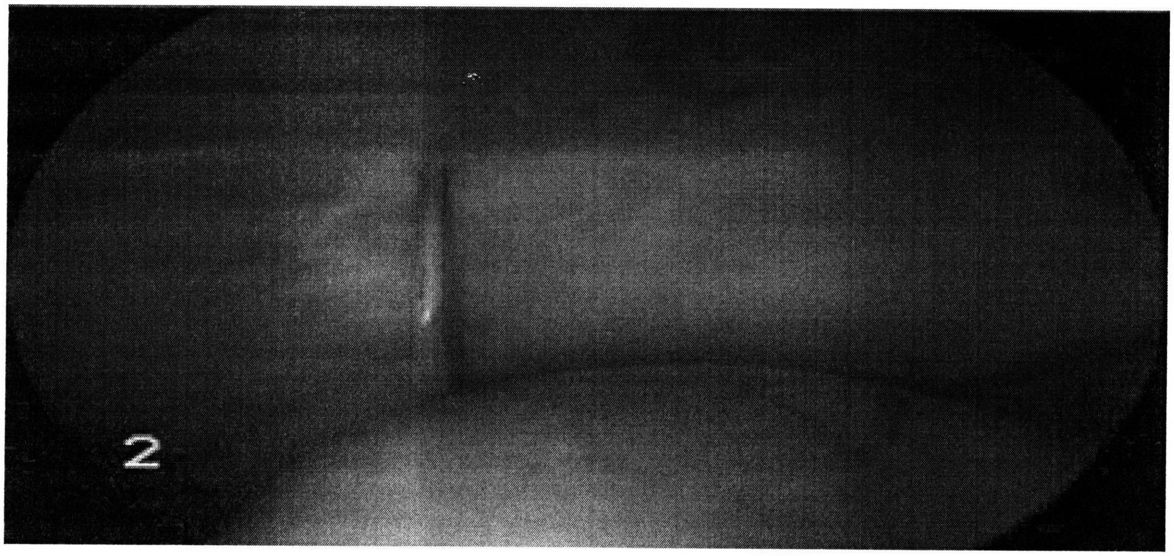


Figure 4-5: Endoscopic image of needle over penetrating abdominal wall of pig

4.5 Prototype Test Conclusions

In all, over 160 tests using 33 different designs were tested on chicken and pigs. Complete test results for the chicken tests were documented in Appendix I. By the end of the testing, the best design performed with ~50% success rate in chicken. In the pig model, it is believed that without the peritoneum, the system would work about 90% of the time. However, if the system can be improved to work even with the peritoneum, it should work in all other applications. The addition of bumps or other features to the shaft of the needle may reduce this overshoot problem by increasing the drag of the needle as it advances through tissue. Fundamentally this is an undesirable solution because it could increase pain and damage to the surrounding tissue while causing other problems. The addition of a point or spike (~1 mm long, ~0.25 mm OD) to the filament tip may enhance piercing of the peritoneum without increasing the drag of the system. Manufacturing the needle tips with larger and sharper cutting edges could achieve a similar improvement. Additionally, replacing the PTFE-coated spring wire guide with authentic nylon epidural catheters may enhance locking and may release differently than the wire. These catheters were ordered, but due to a two-month delay in delivery, no tests could be performed. Future work will include the incorporation and testing of these additional design features.

Chapter 5

Image Processing

5.1 Motivation

Coordinate measuring machines (CMM) were originally used to measure the actual shape of the original tubes studied in the work. Unfortunately, the CMM used was slow and difficult to use, and did not allow easy transfer of information to other analysis programs. As a result, alternate methods for measuring 2D shapes were sought. A flatbed scanner and Matlab were respectively chosen to acquire and process images of the tubes. Experiments showed this method to be viable for many situations, but demonstrated some limitations. The method, analysis, results and conclusions from this study are presented here.

5.2 Introduction

This work studied a new method to measure the shape and size of 2D features in a tube (OD 1.47 mm) with bends in it. Coordinate measuring machines (CMM) are often used but can be slow, tedious, and include room for error. Also, older CMM's do not include software that enables easy storage and transfer of information to other analysis programs. Another problem with CMM's for measuring something of this scale was the tube needed to be stabilized with a holding fixture. If dimensional errors occurred while machining the fixture, they could transfer to measurement of the tube. Also, holding parts in a fixture does not easily allow for part variation. Depending on the design of the holding fixture, deformation of the tube may be necessary for it to fit in the fixture, adding additional error to the measurement of the part. For these reasons, a quick, robust, repeatable, and non-intrusive method was sought to acquire the measurements.

The method selected to obtain this information was to use a common desktop scanner to acquire an image of the shape and then processing it using Matlab (analysis software). For this study, an HP Scanjet G4050 Photo Scanner using the software HP Scan Pro V. 7.2.5 acquired the images. Using the program Matlab, these images were processed producing output values of length, angle, and relative position of tube features. This paper describes in detail the tools used in this method.

5.3 How Flatbed Scanners Work

Flatbed desktop scanners acquire images by moving a scanning head along the length of the object being scanned. The head includes a light source and set of mirrors. The light illuminates the item being scanned while the mirrors (typically 2 or 3) reflect the image to a lens while reducing its size. After passing through the lens, the image is acquired by a charge-coupled device (CCD) array. The CCD, along with the other electronics and software, converts the image into a series of colored squares, or pixels, of the appropriate color. The scanning head slides on a rail and is advanced by a stepper motor. The precision of the mirrors, lens, CCD, and stepper motor define the maximum optical resolution of the scanner. The hardware resolution for this particular scanner is 4800 x 9600 dpi.

5.4 Theory

Scanners theoretically produce images composed of pixels of equal and specifiable size. This critical assumption led to the decision to use a flatbed scanner as a measurement tool. Based on this assumption, each pixel location in an image can be mapped to a position location. For this work, images were in Portable Network Graphics (png) format. In this format, the image is stored as a 2D matrix, where each value corresponds to a pixel. The location in the matrix corresponds to the pixel location in the image and thus also its position. Each value stored within the matrix defines the color of the pixel. Using the differences in pixel color, edges and other features can be extracted from an image. With this information, relative positions of part features can be accurately measured and analyzed.

5.5 Image Processing Steps

The tube shapes studied were all of similar shape, a sine-wave bent into a straight tube, similar to the one found in Figure 5-1. They differed only in the amplitude and length of the sine-wave. Each tube was scanned at 1200 dpi using 8-bit grayscale color and no sharpening (an option in the scanning software). Once scanned, several pre-processing steps were performed on the images. These included cropping and flipping (when necessary) so the bends were on the same side, and touching up the boundaries to ensure the program could distinguish between the different regions of the image. Detailed lists of the pre-processing and processing steps can be found in Appendix III. Also included there is a list of steps for taking the image processing output values and preparing them to be used in the compression test analysis. With the pre-processing of the images complete, they were read into Matlab and processed.

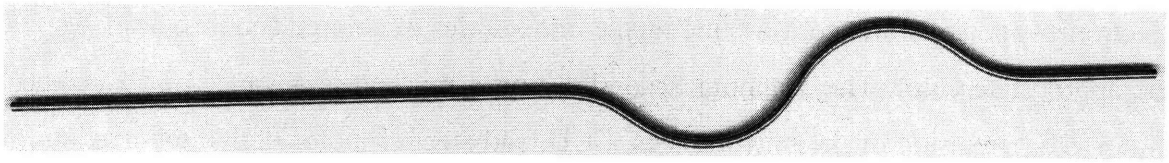


Figure 5-1: Example image of scanned tube

5.6 Image Scans Performed

To validate that the image pixels could be used to determine actual distances, a ruler was scanned both horizontally and vertically at 1200 dpi. These images were then imported into Matlab and measured using the 'imtool' distance function. The vertical (long axis of the scanning bed) and horizontal scan directions were found to be within 1.2% and 0.8% of the 1200 pixels per inch respectively, an acceptable approximation for the purposes of this study.

To determine whether the scanning optics (and other scanner design features) affected the images acquired and resulting post-processed numbers, and analysis of variance experiment was performed. Four scanning parameters were tested: side-to-side placement of the tube on the scan window (left/right), bend orientation (bends go up/down first), angle with respect to horizontal (positive/flat/negative), and the horizontal direction the tube (bends to the left/right). A single tube was scanned once for each parameter variation and the resulting outputs compared. In total, 24 different arrangements were scanned and processed (2^3*3=24). Table 5-1 shows the position of each variable for each scan and Table 5-2 shows the output values for each scan.

Table 5-1: Scan variable values for each scan

Scan Number	Side-to-side placement	Bend orientation	Placement Angle	Tube direction
1	Right	Up	Positive	Right
2	Right	Down	Positive	Right
3	Left	Up	Positive	Right
4	Left	Down	Positive	Right
5	Left	Down	Negative	Right
6	Left	Up	Negative	Right
7	Right	Up	Negative	Right
8	Right	Down	Negative	Left
9	Left	Up	Positive	Left
10	Left	Down	Positive	Left
11	Left	Up	Negative	Left
12	Left	Down	Negative	Left
13	Right	Up	Positive	Left
14	Right	Down	Positive	Left
15	Right	Down	Negative	Left
16	Right	Up	Negative	Left
17	Right	Up	Flat	Left
18	Right	Down	Flat	Left
19	Left	Down	Flat	Left
20	Left	Up	Flat	Left
21	Left	Up	Flat	Right
22	Left	Down	Flat	Right
23	Right	Down	Flat	Right
24	Right	Up	Flat	Right

Table 5-2: Output values for each scan

Scan Number	Total Length (m)	Curved Length (m)	Long Length (m)	Short Length (m)	Transition Length 1 (m)	Transition Length 2 (m)	Bend Radius (m)
1	0.1771	0.1372	0.0363	0.0036	0.0198	0.0183	0.0471
2	0.1771	0.1369	0.0364	0.0038	0.0184	0.0171	0.0462
3	0.1768	0.137	0.0361	0.0037	0.0199	0.0181	0.0464
4	0.1768	0.1369	0.0362	0.0037	0.0185	0.0171	0.0461
5	0.1769	0.1353	0.0379	0.0037	0.0187	0.0173	0.0456
6	0.1769	0.137	0.0363	0.0036	0.02	0.0181	0.0467
7	0.1772	0.1372	0.0363	0.0037	0.0201	0.0189	0.0473
8	0.1772	0.1364	0.0371	0.0036	0.0187	0.0174	0.0461
9	0.1768	0.1357	0.0374	0.0037	0.0188	0.0176	0.0455
10	0.1768	0.136	0.0371	0.0038	0.0185	0.0169	0.0463
11	0.1768	0.1369	0.0362	0.0037	0.0192	0.0179	0.0465
12	0.1769	0.1366	0.0366	0.0037	0.0189	0.0175	0.0468
13	0.177	0.1363	0.0371	0.0035	0.019	0.0178	0.0459
14	0.177	0.1365	0.0369	0.0037	0.0187	0.017	0.0463
15	0.1771	0.1368	0.0367	0.0036	0.0194	0.0171	0.0466
16	0.1771	0.138	0.0354	0.0036	0.0192	0.0176	0.0469
17	0.1769	0.1358	0.0373	0.0038	0.019	0.0177	0.0461
18	0.177	0.1352	0.0381	0.0038	0.0189	0.0162	0.046
19	0.1767	0.1353	0.0376	0.0038	0.0195	0.0174	0.0462
20	0.1767	0.136	0.0369	0.0037	0.0187	0.017	0.046
21	0.1768	0.1362	0.037	0.0037	0.0197	0.0181	0.0463
22	0.1767	0.1354	0.0376	0.0037	0.0194	0.0179	0.0454
23	0.1771	0.1344	0.0391	0.0036	0.0193	0.0178	0.0456
24	0.1769	0.1359	0.0371	0.0039	0.02	0.0184	0.0467

5.7 Statistical Analysis

A statistical regression analysis was performed on the results to study the influence of the scanning parameters on the processed output values. Regression slope and y-intercept coefficients and their 95% confidence bounds were calculated for each output with respect to each variable studied. Table 5-3 and Table 5-4 contain the slope and y-intercept values and their confidence bounds. The confidence bounds in Table 5-4 that do not include zero (bolded) were of most importance.

Table 5-3: Regression slope and y-intercept values for outputs and variables tested

Output	Slope	Side Placement y-intercept	Bend Orientation y-intercept	Placement Angle y-intercept	Tube Direction y-intercept
Total Length (m)	1.768E-01	2.637E-04	-3.442E-05	-4.568E-05	4.747E-05
Curved Length (m)	1.358E-01	2.934E-04	5.751E-04	-4.540E-05	1.236E-04
Long Length (m)	3.724E-02	-1.578E-05	-6.107E-04	2.089E-06	-6.136E-05
Short Length (m)	3.724E-03	-3.832E-05	-2.238E-05	9.889E-06	-9.013E-06
Transition Length 1 (m)	1.863E-02	8.556E-05	5.881E-04	-1.989E-04	4.544E-04
Transition Length 2 (m)	1.691E-02	8.921E-05	7.476E-04	-1.494E-04	5.808E-04
Bend Radius (m)	4.589E-02	3.258E-04	3.731E-04	-1.306E-04	1.500E-04

Table 5-4: Confidence intervals (95%) for the regression slope and y-intercept

Output	Slope	Side Placement y-intercept	Bend Orientation y-intercept	Placement Angle y-intercept	Tube Direction y-intercept
Total Length (m)	1.767e-01 1.767e-01	**3.271e-04** **2.002e-04**	2.878e-05 -9.762e-05	**-6.834e-06** **-8.453e-05**	1.118e-04 -1.685e-05
Curved Length (m)	1.365e-01 1.352e-01	9.581e-04 -3.714e-04	1.237e-03 -8.716e-05	3.617e-04 -4.525e-04	7.977e-04 -5.505e-04
Long Length (m)	3.787e-02 3.662e-02	6.126e-04 -6.441e-04	1.535e-05 -1.237e-03	3.869e-04 -3.827e-04	5.758e-04 -6.985e-04
Short Length (m)	3.803e-03 3.645e-03	4.029e-05 -1.169e-04	5.593e-05 -1.007e-04	5.803e-05 -3.825e-05	7.070e-05 -8.872e-05
Transition Length 1 (m)	1.896e-02 1.830e-02	4.190e-04 -2.479e-04	**9.204e-04** **2.559e-04**	5.345e-06 -4.031e-04	**7.926e-04** **1.163e-04**
Transition Length 2 (m)	1.721e-02 1.661e-02	3.917e-04 -2.133e-04	**1.049e-03** **4.463e-04**	3.580e-05 -3.347e-04	**8.875e-04** **2.741e-04**
Bend Radius (m)	4.627e-02 4.551e-02	7.057e-04 -5.409e-05	7.516e-04 -5.340e-06	1.020e-04 -3.633e-04	5.352e-04 -2.353e-04

To more accurately calculate the bolded confidence bounds, results were recalculated using only the relevant input and output variables. This procedure removed the insignificant parameters and reduced the degrees of freedom, thus refining the confidence bounds. Table 5-5 and Table 5-6 show the new confidence interval ranges.

Table 5-5: Revised confidence intervals (95%) for Total Length regression coefficient

Output	Slope	Side Placement y-intercept	Placement Angle y-intercept
Total Length (m)	1.769e-01 1.768e-01	3.279e-04 1.979e-04	-1.232e-06 -7.941e-05

Table 5-6: Revised confidence intervals (95%) for Transition 1 & 2 regression coefficients

Output	Slope	Bend Orientation y-intercept	Tube Direction y-intercept
Transition Length 1 (m)	1.898e-02 1.836e-02	9.734e-04 2.727e-04	7.181e-04 1.527e-05
Transition Length 2 (m)	1.718e-02 1.665e-02	1.088e-03 4.965e-04	8.598e-04 2.664e-04

5.8 Study of 3D Effects

To determine the 3D effects and optics of the scanner, a small length (22.7 mm) of tubing (OD 1.47 mm) was scanned at different horizontal locations on the scanning surface. These scans showed that there is a 3D effect due to the mirror optics. This can be observed in Figure 5-2, which contains the image at the 5 different scanned locations. From these images it is evident that the 3D effects of the object are affected by the optics. This could result in minor errors in measuring lengths, especially if it spans the majority of the scanning window. Extracting boundaries and relevant edges in the image may also be complicated by this effect. Future work will include a more thorough study of this effect.

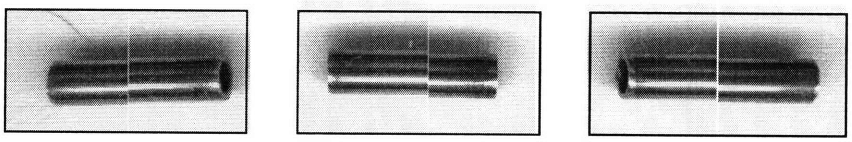


Figure 5-2: Tube section scanned at left middle and right locations on the scanner

5.9 Discussion of Image Processing Results

The purpose of this analysis was to see if a flatbed scanner could be a viable tool for measuring 2D features on a part. Of the variables studied in this analysis, four were found to influence the output values. The 'Total Length' was only significantly influenced by the side-to-side placement and angle of the tube. The side-to-side placement affected the result by as much as 0.4 mm. One explanation for this variation can be attributed to the method used to select where the edge began and ended and was known to be a potential source of minor error. The associated error was less than 0.3% and considered negligible. The other variable affecting the 'Total Length' was the placement angle. This too could have been influenced by the analysis method used to choose the edge points. The error from this variable was less than the error from the side-to-side placement, so it too was negligible.

The other output value that was influenced by the input variables was the 'Transition Length'. This was an approximate measure of the length string that did not touch the tube wall as it took the shortest path through the tube. This value was influenced by the variables 'bend orientation' and 'tube direction' by as much as 1.1 mm (~7% of output). This was a difficult value to measure and the analysis method used to calculate it could have caused a significant portion of this error. Unlike the other values analyzed, the 'Transition Length' was used only as an approximation, so a 7% variation was considered acceptable. Another potential cause of this error is the 3D effects of scanner optics that could slightly distort the curves slightly, thus affecting the shortest path of the string.

This method proved efficacious for the steel tubes analyzed, but has limitations. It is also not a comprehensive system and would have to be modified for it to work with other systems. For example, this analysis utilized the reflection from the light source on the tube, so if an object with a different cross-section shape was scanned, the method for choosing the edge boundary might need to be modified. Additionally, if the tube were a different color(s) or had a different surface finish, a similar modification would be required. This method relies heavily on obtaining clean edge boundaries of objects in an image, like the lower edge of tubes in Figure 5-2. The light source in the scanner head is offset, causing shadows on the upper edge of objects which would likely cause problems if the 2D shape became too complex or contained loops. An improved way to differentiate between shadow and object would reduce this problem, but that was not studied here. This method is size limited by the scanner itself which is generally not much larger than a standard sheet of paper (22 X 28 cm). Another potential source of complication is from the scanning optics which produce a slight 3D distortion that depends on the thickness of the object (visible in Figure 5-2). Future work will study the effect of this 3D distortion on measurements.

5.10 Image Processing Conclusions

This study found that using a flatbed scanner with an analysis program is a reliable method for measuring 2D planar part features. Scanning variables such as horizontal and angular placement of a part on the scanning surface were shown to affect the images slightly, but in most cases can be considered negligible. For the tubing studied (OD 1.47 mm) 3D effects appear to be negligible but future work will include a more thorough analysis of 3D and optical effects on image results. Although the analysis method may not be suitable for all applications and may require tailoring for others, this study supports its continued use and development.

Chapter 6

Challenges Overcome

6.1 Friction Test Block

Obtaining accurate friction coefficient values were the first major challenge experienced. To accomplish this, two short lengths of the steel tubing were adhered in respective parallel machined grooves in a metal block. Initially, the grooves were milled with a square end-mill bit while the block sat at roughly a 45 degree angle, producing V-channel grooves. The first block was aluminum but subsequent blocks were steel so they could be magnetically secured to the grinding table. This technique produced several problems. The first was that it was very difficult to get the block surface exactly parallel to the mill axis, resulting in channels that where depth changed along the length of the cut. The other difficulty was to achieve two cuts of equal depth. Because the exact surface angle of the block was unknown the digital readouts on the mills were rendered near useless. To overcome these challenges, a horizontal slot saw blade (1.59 mm thick) was used instead of a square end-mill bit. Using this tool, square grooves of constant and equal depths (0.76 mm) were achievable. This was done by clamping the block to a large steel plate that was aligned with the mill axes. Thus the digital readouts on the mill could be utilized to achieve precise machining. With those challenges overcome, the next was to improve the bonding between the tubes and block.

In an attempt to produce a strong bond, both the block and tubes were cleaned with alcohol and scored with sand paper prior to applying the epoxy. The epoxy was then applied, the tubes placed in the grooves and left to sit. Once the epoxy had fully cured, the tubes were ground in half using a surface grinder. This proved more difficult than expected. The bond on the tubes consistently failed as in Figure 6-1. This would typically happen when more than half of the tube had been ground away and occurred with 4 different types of epoxies, including one designed to be good up to temperatures of 700 degrees F. Initially it was thought that temperature was the problem so epoxies of increasing working temperatures were used. In an attempt to minimize thermal and mechanical loads on the bonds, complete flood coolant, feed rates of only .005 mm, and grinding direction 45 degrees off the tube axis were implemented. In one case, annealing the tubes was even tried. None of these changes resulted in successful grinding. These unsuccessful attempts showed that the root cause was most likely the poor bonding of the epoxy to the tubes. To improve this, the tubes were sandblasted to roughen the surface. This proved to solve the problem and a block was successfully ground for the friction tests.

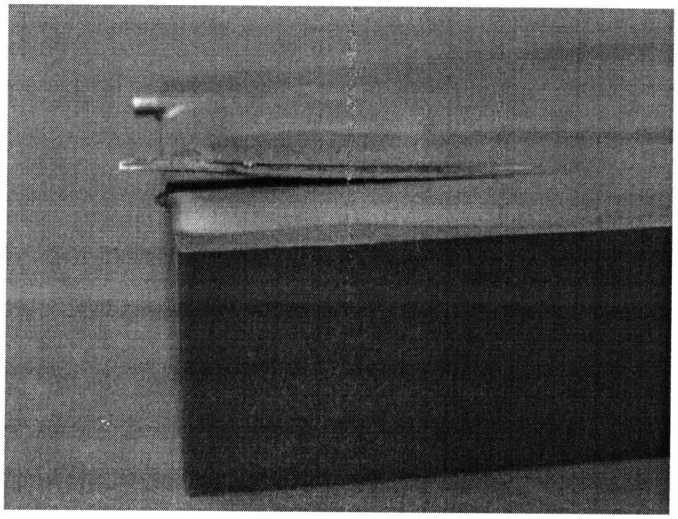


Figure 6-1: Example of failed grinding attempt

6.2 Friction Test Method

With the blocks ground, the last challenge was to acquire the friction coefficient values. Initially, a tilt test was used as seen in Figure 6-2. For this test, the block was placed on parallel filaments on a plate, and one end of a plate was raised. When the block began slipping down the plate, the plate angle was measured and used to calculate the friction coefficient. Unfortunately, this method contained sources of error. First, raising the plate without a ball joint meant that small vibration occurred as the plate was being lifted. This vibration could be transmitted to the block, causing it to slip early. A similar problem occurred at the downhill end of the plate. There, it was supported by a plastic round rod, but it too had the potential of adding vibration as it rolled. The second major source of error was from manually stopping the machine once the block began to slip. This not only added variation, but this problem was compounded because the block would start to slip then stop, and it was difficult to maintain consistency of the true slipping point. The final source of error was originally unknown to the author; polymer transfer. Through the course of the tests, it was observed that the slip angle consistently decreased. The cause of this was that they polymer began coating the inside of the metal tubes, thus reducing the friction. For these reasons, an improved method was investigated and found (see Section 3.1 Steel Tube-Filament Friction Coefficients).

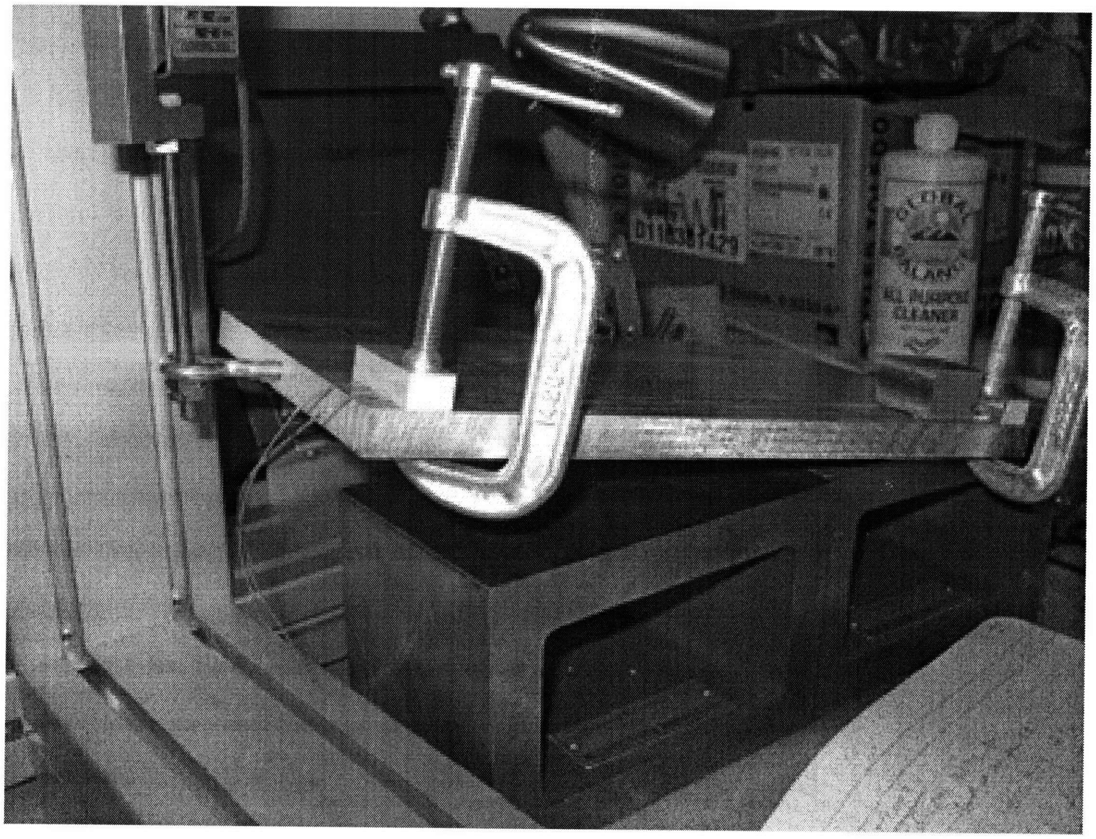


Figure 6-2: Picture of Friction test setup

6.3 Acquiring Supplies

Acquiring actual epidural catheters proved to be far more difficult than expected. It was important to acquire authentic catheters or a very similar material to determine whether they could be used in the device. Nylon fishing line and other similar products were compared to the catheters but found to be much stiffer. It then became necessary to acquire authentic medical grade catheters. The doctors on the research team were unable to acquire any from the hospital, so venders and suppliers that would sell to an individual were sought out. It was found that large medical companies like BD do not sell to or give samples to individuals. Finally a distributor was found that would sell to individuals, Henry Schein Inc. With the help of the doctor at Brigham and Women's Hospital, an order was able to be placed. The catheters were expensive and took ~3 weeks to arrive, but were invaluable in the research and well worth the effort and cost to acquire.

Chapter 7

Conclusions and Future Work

This works studied the tensile and compressive forces of a filament passing through a tube for the development of a new two-part linear clutch mechanism. In tension, the capstan equation was found to provide a good approximation of the forces. The clutch action occurred when the filament was in compression, and these fundamental principles were well described by the mathematical model presented here. This model was based on drill string models from the oil industry and required the addition of a scaling factor (40D/L). The inclusion of the scaling factor produced accurate predictions of the measured test results for all tube shapes studied. A more thorough study of the system needs to be performed to determine why the scaling factor is necessary.

Tests with prototype devices were performed on raw chicken breast showed that many more variables influence the function of the system than could be fully studied here. Additional variables including needle tip geometry, filament type, and filament tip geometry require further study for further improvement of the prototypes. After many iterations of the prototypes, a success rate of about 50% was achieved in raw chicken breast. Tests in deceased pigs showed the system advanced easily through muscle but struggled to pierce the peritoneum. Once it finally penetrated the peritoneum it advanced farther than was desirable before stopping. Future work should include a more thorough study of the ideal filament material, the filament tip shapes and sizes, and the needle tip shape with additional tests on pigs.

A fully functional device has great potential in catheterization and laparoscopic procedures. It is expected to significantly decrease procedure times and complication rates. This reduces procedure costs while greatly improving both physician and patient experiences.

Additional work went into the study and development of a different method for measuring the 2D shape of the tubes in the device. This method consisted of scanning the tubes with a flatbed scanner and then processing the image using Matlab to obtain relevant measurements. This method proved to have many benefits over using a coordinate measuring machine. It was non-intrusive, precise and robust. This method may not be ideal for all applications, but works very well for measuring the edge of the constant diameter tubing. Future work in this area will include a study of the 3D effects of the tube and optical distortion of the scanner mirrors on the results.

Appendix I- Prototype Test Results

4/2/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.5_A350	Point Angle: Acute	Solid Pebax	Didn't really work
ID: 1.22 mm	No. of Points: 2	OD: 0.89 mm	The moment of inertia (I) seems too high, causing too much drag through tube
			Mostly didn't work
			Saw same poor results as before…filament would pierce chicken but not stop upon penetration
			Did work once but may have been luck (see pics)

4/2/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.5_A350	Point Angle: Acute	PTFE coated spring-wire guide	J-tip worked some of time (see pics)
ID: 1.22 mm	No. of Points: 2	OD: 0.76 mm	Of last 4 or 5 tries worked 3 times
		Tip: J-tip	Push on it with a rod not a tube
		Included tiny nitinol wire inside coil	Has very low sliding resistance in tube which is very good
			J-tip still has really thin piece of wire but not primary one

4/2/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.5_A350	Point Angle: Acute	PTFE coated spring-wire guide	Penetrated tissue often
ID: 1.22 mm	No. of Points: 2	OD: 0.76 mm	Never really locked in place

4/2/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.5_A350	Point Angle: Acute	PTFE coated spring-wire guide	Penetrated tissue
ID: 1.22 mm	No. of Points: 2	OD: 0.76 mm	Often came out of needle tip prematurely
		Steel ball on tip (OD 0.79 mm)	

4/2/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.5_A350	Point Angle: Acute	PTFE coated spring-wire guide	Worked a few times but not many
ID: 1.22 mm	No. of Points: 2	OD: 0.76 mm	Sometimes deployed too early
		Steel ball on tip (OD 1.00 mm)	Once or twice went all the way through without deploying upon penetration

4/3/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.5_A350	Point Angle: Acute	PTFE coated spring-wire guide	Deployed early
ID: 1.22 mm	No. of Points: 2	OD: 0.76 mm	Didn't stop or deploy upon penetration
		Tip: J-tip	Didn't stop or deploy upon penetration
		Included tiny nitinol wire inside coil	Needle stopped a bit too late
			Deployed early

4/3/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.5_A350	Point Angle: Acute	PTFE coated spring-wire guide	Deployed early
ID: 1.22 mm	No. of Points: 2	OD: 0.76 mm	worked
		Steel ball on tip (OD 1.00 mm)	worked
		Included tiny nitinol wire inside coil	Deployed early
			Didn't stop or deploy upon penetration

4/3/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.5_A350	Point Angle: Acute	PTFE coated spring-wire guide	Deployed early
ID: 1.22 mm	No. of Points: 2	OD: 0.76 mm	Deployed early
		Steel ball on tip (OD 0.79 mm)	Deployed early
		Included tiny nitinol wire inside coil	Deployed early
			Deployed early

4/3/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.5_A350	Point Angle: Acute	PTFE coated spring-wire guide	worked but may have deployed a bit early
ID: 1.22 mm	No. of Points: 2	OD: 0.76 mm	worked but may have deployed a bit early
		Steel ball on tip (OD 1.00 mm)	Went through but deployed ~6 mm before penetration
			Deployed early
			Deployed early

4/3/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.5_A350	Point Angle: Acute	PTFE coated spring-wire guide	Deployed early
ID: 1.22 mm	No. of Points: 2	OD: 0.76 mm	Deployed early
		Steel ball on tip (OD 0.79 mm)	Deployed early
			Deployed early
			Deployed early

4/3/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.5_A350	Point Angle: Obtuse	PTFE coated spring-wire guide	worked
ID: 1.22 mm	No. of Points: 2	OD: 0.76 mm	worked
		Steel ball on tip (OD 1.00 mm)	Deployed ~3 mm early but mostly worked
			Needle didn't quite penetrate all of the tissue (~3 mm short)
			deployed early

4/3/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.7_A250	Point Angle: Obtuse	PTFE coated spring-wire guide	worked
ID: 1.22 mm	No. of Points: 2	OD: 0.76 mm	ball jammed in needle tip
		Steel ball on tip (OD 1.00 mm)	ball jammed in needle tip
			worked
			deployed early (I think)

4/3/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.7_A250	Point Angle: Obtuse	PTFE coated spring-wire guide	worked
ID: 1.22 mm	No. of Points: 2	OD: 0.76 mm	deployed early
		Steel ball on tip (OD 1.00 mm)	deployed early
			deployed early
			deployed early

4/8/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.5_A350	Point Angle: Obtuse	PTFE coated spring-wire guide	deployed early
ID: 1.83 mm	No. of Points: 2	OD: 0.76 mm	worked
	Sandblasted shaft	Steel ball on tip (OD 1.59 mm)	worked
			deployed early
			deployed early

4/8/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.6_A350	Point Angle: Obtuse	PTFE coated spring-wire guide	didn't quickly disengage upon penetration and needle advanced too far
ID: 1.83 mm	No. of Points: 2	OD: 0.76 mm	didn't quickly disengage upon penetration and needle advanced too far
	Sandblasted shaft	Steel ball on tip (OD 1.59 mm)	deployed early

			deployed early
			deployed early

4/8/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.5_A350	Point Angle: Obtuse	PTFE coated spring-wire guide	didn't deploy
ID: 1.83 mm	No. of Points: 2	OD: 0.76 mm	deployed just a bit early
	Sandblasted shaft	Steel ball on tip (OD 1.59 mm)	deployed early
		Included tiny nitinol wire inside coil	deployed early
			deployed just a bit early, the ball got stuck, needle kept going so there was a loop of filament sticking out

4/8/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.5_A350	Point Angle: Acute	PTFE coated spring-wire guide	Deployed early
ID: 1.83 mm	No. of Points: 2	OD: 0.76 mm	advanced a bit then deployed early
	Sandblasted shaft	Steel ball on tip (OD 1.59 mm)	didn't deploy
		Included tiny nitinol wire inside coil	didn't deploy
			didn't deploy
			(the ball tended to get stuck in the tube, probably on the burs)

4/11/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.53_A700	Point Angle: Obtuse	PTFE coated spring-wire guide	Didn't deploy
ID: 1.83 mm	No. of Points: 2	OD: 0.76 mm	Didn't deploy
	Sandblasted shaft	Steel ball on tip (OD 1.59 mm)	deployed late
			worked
			Didn't deploy
			Didn't deploy
			(It looks like sometimes the ball gets jammed in the tip by the chicken tissue)

4/11/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.53_A525	Point Angle: Obtuse	PTFE coated spring-wire guide	didn't deploy early enough
ID: 1.83 mm	No. of Points: 2	OD: 0.76 mm	deployed early
	Sandblasted shaft	Steel ball on tip (OD 1.59 mm)	deployed early
			didn't deploy enough
			didn't deploy

4/11/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.6_A525	Point Angle: Obtuse	PTFE coated spring-wire guide	worked
ID: 1.83 mm	No. of Points: 2	OD: 0.76 mm	didn't deploy
	Sandblasted shaft	Steel ball on tip (OD 1.59 mm)	deployed late
			worked
			worked
			worked
			worked
			worked
			deployed late
			worked

4/11/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.53_A700	Point Angle: Obtuse	PTFE coated spring-wire guide	deployed early
ID: 1.83 mm	No. of Points: 2	OD: 0.76 mm	didn't deploy
	Sandblasted shaft	Steel ball on tip (OD 1.59 mm)	didn't deploy
		Included tiny nitinol wire inside coil	deployed late
			didn't deploy

4/11/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.6_A525		deployed early	didn't deploy
ID: 1.83 mm		worked	deployed late
		deployed just a bit early	deployed late
		deployed early	worked
		didn't deploy	deployed late
		worked	didn't deploy
		deployed early	deployed early

4/14/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.6_A525	Point Angle: Obtuse	PTFE coated spring-wire guide	worked
ID: 1.83 mm	No. of Points: 3	OD: 0.76 mm	deployed ~10 mm late
	Sandblasted shaft	Steel ball on tip (OD 1.59 mm)	deployed ~5 mm late
		Included tube over last ~12 mm	Worked
			didn't deploy
			Worked
			deployed late

			deployed ~3 mm early
			Worked
			deployed ~5 mm late

4/14/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.53_A700	Point Angle: Obtuse	PTFE coated spring-wire guide	Deployed late
ID: 1.83 mm	No. of Points: 3	OD: 0.76 mm	Deployed late
	Sandblasted shaft	Steel ball on tip (OD 1.59 mm)	Deployed late
		Included tube over last ~12 mm	Deployed late
			worked
			Didn't deploy
			Didn't deploy

4/14/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.6_A525	Point Angle: Obtuse	PTFE coated spring-wire guide	Deployed early
ID: 1.83 mm	No. of Points: 3	OD: 0.76 mm	Deployed early
	Sandblasted shaft	Steel ball on tip (OD 1.59 mm)	Deployed early
		Included tube over last ~12 mm	Deployed early
		Included tiny nitinol wire inside coil	Deployed early

4/14/2008			
Tube Details	**Needle Details**	**Filament Details**	**Results**
Shape: R0.53_A525	Point Angle: Obtuse	PTFE coated spring-wire guide	deployed ~3 mm early
ID: 1.83 mm	No. of Points: 3	OD: 0.76 mm	deployed ~3 mm early
	Sandblasted shaft	Steel ball on tip (OD 1.59 mm)	deployed ~3 mm early
		Included tube over last ~12 mm	deployed ~5 mm late
			worked

Appendix II- Matlab Code

Example of file to read in compression test results for analysis

File name: C_1B.m

```
clear all;
close all;
clc;


%Define constants
Conversion_Factor = double(-2.46);

C = 1;
T = 2;

material = C;
Tube_number = 2;
correction_factor = 2/pi; %this accounts for the difference between the measured data and the model by He
for the first model
% correction_factor2 = 1/5; %this accounts for the difference between the measured data and the model by
He for the best model

Sweep_Angle_Max = double(1.7300);
Sweep_Angle_Min = double(1.4155);
Sweep_Angle_Avg = mean([Sweep_Angle_Max; Sweep_Angle_Min]);

Mu_Max = double(0.2692);
Mu_Min = double(0.2040);
Mu_Avg = mean([Mu_Min; Mu_Max]);

% Percent difference interval
percent_max = 0.055;
percent_min = 0.049;

%Define Files
Files ={'C_1B_R1.txt'; 'C_1B_R2.txt'; 'C_1B_R3.txt'; 'C_1B_R4.txt'; 'C_1B_R5.txt'; 'C_1B_R6.txt';
'C_1B_R7.txt'; 'C_1B_R8.txt'; 'C_1B_R9.txt'; 'C_1B_R10.txt'; 'C_1B_R13.txt'; 'C_1B_R14.txt';
'C_1B_R15.txt'}; %Files not used because kinking messed up results 'C_1B_R11.txt'; 'C_1B_R12.txt';
'C_1B_R16.txt'

Log_cutoff_Vals_zeroed = zeros(size(Files)+3,7);
Log_cutoff_Vals_non_zeroed = zeros(size(Files)+3,7);
exponential_fit_info = zeros(size(Files)+3,6);
Fout_P_fin_info = zeros(size(Files)+3,5);

importfile('Tube_summary_condensed.txt')
importfile('String_specs2.txt')
transition_lengths = load('transition_lengths.txt');

correction_factor2 = 40*D(material)/Long_Straight_Length(Tube_number); %this accounts for the
difference between the measured data and the model by He for the best model
```

```
%Import Files and create Fin, Fout, and Position matrices
for i = 1:size(Files)

   Raw_Data = double(load(Files{i}));
   File_name = Files(i);

   %calculate values with zeroed forces
%    [Cutoff_Vals_zeroed(i,:),Log_cutoff_Vals_zeroed(i,:)] =
Compression_analysis_zeroed(Raw_Data,Conversion_Factor,Sweep_Angle_Max,Mu_Max,Sweep_Angle_
Avg,Mu_Avg,Sweep_Angle_Min,Mu_Min,percent_max,percent_min,File_name,material);

   %calculate values with actual forces

[Cutoff_Vals_non_zeroed(i,:),Log_cutoff_Vals_non_zeroed(i,:),exponential_fit_info(i,:),Fout_P_fin_info(i
,:)] =
Compression_analysis_non_zeroed(Raw_Data,Conversion_Factor,Sweep_Angle_Max,Mu_Max,Sweep_A
ngle_Avg,Mu_Avg,Sweep_Angle_Min,Mu_Min,percent_max,percent_min,File_name,material,E(material)
,correction_factor,correction_factor2,I(material),R(material),D(material),r(material),d(material),Long_Strai
ght_Length(Tube_number),w(material),Curved_length(Tube_number),Bend_Radius(Tube_number),Avg_I
nclination(Tube_number),transition_lengths(Tube_number,material));
   clear Raw_Data;

end
Column_names =
{'Fin_P_fit','Fout_P_fit','Position','Percent_Diff','Fin_fit_ratio','Fin/Fout_fit_ratio','Fin_P_fit','Fout_P_fit','P
osition','slope_of_fit_P','Fin_P_fit','Fout_P_fit','Position','Fin/Fout_fit_ratio','slope_of_fit_P','slope_slope_v
al'};

sizer = size(Log_cutoff_Vals_zeroed);
for l = 1:sizer(2)
   Log_cutoff_Vals_zeroed(i+2,l) = mean(Log_cutoff_Vals_zeroed(1:length(Log_cutoff_Vals_zeroed)-
3,l));
   Log_cutoff_Vals_zeroed(i+3,l) = std(Log_cutoff_Vals_zeroed(1:length(Log_cutoff_Vals_zeroed)-3,l));
   Log_cutoff_Vals_non_zeroed(i+2,l) =
mean(Log_cutoff_Vals_non_zeroed(1:length(Log_cutoff_Vals_non_zeroed)-3,l));
   Log_cutoff_Vals_non_zeroed(i+3,l) =
std(Log_cutoff_Vals_non_zeroed(1:length(Log_cutoff_Vals_non_zeroed)-3,l));
end

sizer2 = size(exponential_fit_info);
for l = 1:sizer2(2)
   exponential_fit_info(i+2,l) = mean(exponential_fit_info(1:length(exponential_fit_info)-3,l));
   exponential_fit_info(i+3,l) = std(exponential_fit_info(1:length(exponential_fit_info)-3,l));
end

sizer3 = size(Fout_P_fin_info);
for l = 1:sizer3(2)
   Fout_P_fin_info(i+2,l) = mean(Fout_P_fin_info(1:length(Fout_P_fin_info)-3,l));
   Fout_P_fin_info(i+3,l) = std(Fout_P_fin_info(1:length(Fout_P_fin_info)-3,l));
end

% Cutoff_Vals_zeroed
% Column_names
% averages = mean(Cutoff_Vals_zeroed)
% standard_devs = std(Cutoff_Vals_zeroed)
% Log_cutoff_Vals_zeroed;
```

```
Log_cutoff_Vals_non_zeroed;
exponential_fit_info;
Fout_P_fin_info;
%
%
%
% Output_zeroed = fopen('C_1B_Cutoff_Vals_zeroed.txt','w+');
% save('C_1B_Cutoff_Vals_zeroed.txt','Cutoff_Vals_zeroed','-ascii', '-tabs')
% fclose(Output_zeroed);
%
% Output_non_zeroed = fopen('C_1B_Cutoff_Vals_non_zeroed.txt','w+');
% save('C_1B_Cutoff_Vals_non_zeroed.txt','Cutoff_Vals_non_zeroed','-ascii', '-tabs')
% fclose(Output_non_zeroed);
%
% Output_Log_cutoff_Vals_zeroed = fopen('C_1B_Log_cutoff_Vals_zeroed.txt','w+');
% save('C_1B_Log_cutoff_Vals_zeroed.txt','Log_cutoff_Vals_zeroed','-ascii', '-tabs')
% fclose(Output_Log_cutoff_Vals_zeroed);
%
% Output_Log_cutoff_Vals_non_zeroed = fopen('C_1B_Log_cutoff_Vals_non_zeroed.txt','w+');
% save('C_1B_Log_cutoff_Vals_non_zeroed.txt','Log_cutoff_Vals_non_zeroed','-ascii', '-tabs')
% fclose(Output_Log_cutoff_Vals_non_zeroed);
%
% Output_exponential_fit_info = fopen('C_1B_exponential_fit_info.txt','w+');
% save('C_1B_exponential_fit_info.txt','exponential_fit_info','-ascii', '-tabs')
% fclose(Output_exponential_fit_info);
%
% Output_Fout_P_fin_info = fopen('C_1B_Fout_P_fin_info.txt','w+');
% save('C_1B_Fout_P_fin_info.txt','Fout_P_fin_info','-ascii', '-tabs')
% fclose(Output_Fout_P_fin_info);
```

Analysis file for processing compression test data

File Name: Compression_analysis_non_zeroed.m

```
%Compression_analysis File
%This file receives data (Forces, Position) from the other file and
%returns relevant numbers

function [Cutoff_Vals_non_zeroed,Log_cutoff_Vals_non_zeroed,exponential_fit_info,Fout_P_fin_info] =
Compression_analysis_non_zeroed(Raw_Data,Conversion_Factor,Sweep_Angle_Max,Mu_Max,Sweep_A
ngle_Avg,Mu_Avg,Sweep_Angle_Min,Mu_Min,percent_max,percent_min,File_name,material,E,correctio
n_factor,correction_factor2,I,R,D,r,d,Long_Straight_Length,w,Curved_length,Bend_Radius,Avg_Inclinati
on,transition_lengths)

C = 1;
T = 2;

inclination = 0; %inclination angle often called theta
a_i = 1/Bend_Radius; %Inclination build rate (rad/m)
a_phi = 0; %azimuth build rate
alpha = Avg_Inclination;

Conversion_to_N = 4.44822162; %conversion from lbf to Newtons
Conversion_to_cm = 2.54; %conversion from inches to centimeters

Load_cell_deflection = 0.012*Conversion_to_cm/10; %This is the deflection of the loadcell as a function
of load (cm/N)

%split raw data into Fin, Fout, Position...
Fin_all(:,1) = Raw_Data(:,1)*Conversion_to_N;
Fout_all(:,1) = Raw_Data(:,2)*Conversion_Factor*Conversion_to_N;
Position_all(:,1) = Raw_Data(:,5)*Conversion_to_cm;
offset = 1;

% importfile('Tube_summary_condensed.txt')
% importfile('String_specs.txt')
% syms s;
l = 1; %This is a variable that counts the number of times that Fin_all(m,1) < F_icr_max(m-offset+1) is
true


%Remove extra data points at beginning and zero position
for m = 1:size(Fin_all)
  if Fin_all(m) <= 0.0667+mean(Fin_all(1:m)) || Position_all(m) <= 0 %Fin_all(m) <= 0.02 %This cuts off
the data points where Fin is less than 0.02 for Nylon, and 0.05 for PTFE
    offset = m;
  end

  Fin(m-offset+1) = Fin_all(m,1);
  Fout(m-offset+1) = Fout_all(m,1);
  Position(m-offset+1) = Position_all(m,1) - Position_all(offset) - Load_cell_deflection*Fin_all(m,1);
%MAY WANT TO CHANGE THIS TO ZERO ALL POSITIONS
%   F_1nbc_max(m-offset+1) = Fin_all(m,1)*a_i; %((w*sin(Sweep_Angle_Max) + Fin_all(m,1)*a_i)^2 +
(Fin_all(m,1)*a_phi*sin(Sweep_Angle_Max))^2)^.5
%   F_1nbc_min(m-offset+1) = Fin_all(m,1)*a_i; %((w*sin(Sweep_Angle_Min) + Fin_all(m,1)*a_i)^2 +
(Fin_all(m,1)*a_phi*sin(Sweep_Angle_Min))^2)^.5
```

```
%    F_icr_max(m-offset+1) = (4*F_1nbc_max(m-offset+1)*E*I/r)^.5;
%    F_icr_min(m-offset+1) = (4*F_1nbc_min(m-offset+1)*E*I/r)^.5;
  F_1nbc(m-offset+1) = Fin_all(m,1)/Bend_Radius; %= Fin_all(m,1)*a_i; %
  F_icr(m-offset+1) = (4*F_1nbc(m-offset+1)*E*I/r)^.5;


% difference = (Fin_all(transition_point-
1,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor - (Fin_all(transition_point-
1,1)/exp(Sweep_Angle_Max*Mu_Max) - Fin_all(offset,1)));

  if Fin_all(m,1) < F_icr(m-offset+1)
%    if Fin_all(m,1) < F_icr_max(m-offset+1)
%        F_1c_max(m-offset+1) = F_1nbc_max(m-
offset+1)*Mu_Max*Long_Straight_Length*correction_factor;
%        Fout_combined_model_max(m-offset+1) = Fin_all(m,1)/exp(Sweep_Angle_Max*Mu_Max) -
Fin_all(offset,1);
%        transition_point = l-offset+1;
%        l = l + 1;
      F_1c_max(m-offset+1) = F_1nbc(m-offset+1)*Mu_Max*Long_Straight_Length*correction_factor;
      F_1c_min(m-offset+1) = F_1nbc(m-offset+1)*Mu_Min*Long_Straight_Length*correction_factor;
%        Fout_combined_model_max(m-offset+1) = Fin_all(m,1)/exp(Sweep_Angle_Max*Mu_Max) -
Fin_all(offset,1);
%        Fout_combined_model_min(m-offset+1) = Fin_all(m,1)/exp(Sweep_Angle_Min*Mu_Min) -
Fin_all(offset,1);
%        Fout_combined_model_max(m-offset+1) = Fin_all(m,1) -
Fin_all(m,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor - Fin_all(offset,1);% +
(Fin_all(transition_point+offset-1,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor -
(Fin_all(transition_point+offset-1,1)/exp(Sweep_Angle_Min*Mu_Min) - Fin_all(offset,1)))/2;% +
Fout_combined_model_max(transition_point) ;%-
Fin_all(transition_point,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor;
%        Fout_combined_model_min(m-offset+1) = Fin_all(m,1) -
Fin_all(m,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor - Fin_all(offset,1);% +
(Fin_all(transition_point+offset-1,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor -
(Fin_all(transition_point+offset-1,1)/exp(Sweep_Angle_Max*Mu_Max) - Fin_all(offset,1)))/2;% +
Fout_combined_model_min(transition_point) ;%-
Fin_all(transition_point,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor;

      transition_point = l-offset+1;
      l = l + 1;
      Straight_and_Curve_model_Max1(m-offset+1) =
Fin_all(m,1)/Bend_Radius*Mu_Max*Long_Straight_Length*correction_factor +
Fin_all(m,1)/Bend_Radius*Mu_Max*(Curved_length-transition_lengths) + Fin_all(offset,1);%F_1nbc(m-
offset+1)*Mu_Max*Curved_length;
      Straight_and_Curve_model_Min1(m-offset+1) =
Fin_all(m,1)/Bend_Radius*Mu_Min*Long_Straight_Length*correction_factor +
Fin_all(m,1)/Bend_Radius*Mu_Min*(Curved_length-transition_lengths) ;% +
Fin_all(offset,1);%F_1nbc(m-offset+1)*Mu_Min*Curved_length;
      Straight_and_Curve_model_Max2(m-offset+1) =
Fin_all(m,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor2 +
Fin_all(m,1)/Bend_Radius*Mu_Max*(Curved_length-transition_lengths) + Fin_all(offset,1) ;%F_1nbc(m-
offset+1)*Mu_Max*Curved_length;
      Straight_and_Curve_model_Min2(m-offset+1) =
Fin_all(m,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor2 +
Fin_all(m,1)/Bend_Radius*Mu_Min*(Curved_length-transition_lengths) ;% +
Fin_all(offset,1);%F_1nbc(m-offset+1)*Mu_Min*Curved_length;
  else
```

```
%       F_1c_max(m-offset+1) =
Fin_all(m,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor;
%       Fout_combined_model_max(m-offset+1) = Fin_all(m,1) -
%       Fin_all(m,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor - Fin_all(offset,1);%
+ (Fin_all(transition_point+offset-1,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor -
(Fin_all(transition_point+offset-1,1)/exp(Sweep_Angle_Min*Mu_Min) - Fin_all(offset,1)))/2;% +
Fout_combined_model_max(transition_point) ;%-
Fin_all(transition_point,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor;
    F_1c_max(m-offset+1) =
Fin_all(m,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor;
    F_1c_min(m-offset+1) =
Fin_all(m,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor;
%       Fout_combined_model_max(m-offset+1) = Fin_all(m,1) -
Fin_all(m,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor - Fin_all(offset,1);% +
(Fin_all(transition_point+offset-1,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor -
(Fin_all(transition_point+offset-1,1)/exp(Sweep_Angle_Min*Mu_Min) - Fin_all(offset,1)))/2;% +
Fout_combined_model_max(transition_point) ;%-
Fin_all(transition_point,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor;
%       Fout_combined_model_min(m-offset+1) = Fin_all(m,1) -
Fin_all(m,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor - Fin_all(offset,1);% +
(Fin_all(transition_point+offset-1,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor -
(Fin_all(transition_point+offset-1,1)/exp(Sweep_Angle_Max*Mu_Max) - Fin_all(offset,1)))/2;% +
Fout_combined_model_min(transition_point) ;%-
Fin_all(transition_point,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor;
    Straight_and_Curve_model_Max1(m-offset+1) =
Fin_all(m,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor +
Fin_all(m,1)/Bend_Radius*Mu_Max*(Curved_length-transition_lengths) + Fin_all(offset,1) ;
    Straight_and_Curve_model_Max2(m-offset+1) =
Fin_all(m,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor2 +
Fin_all(m,1)/Bend_Radius*Mu_Max*(Curved_length-transition_lengths) + Fin_all(offset,1) ;
    Straight_and_Curve_model_Min1(m-offset+1) =
Fin_all(m,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor +
Fin_all(m,1)/Bend_Radius*Mu_Min*(Curved_length-transition_lengths) ;% + Fin_all(offset,1);
    Straight_and_Curve_model_Min2(m-offset+1) =
Fin_all(m,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor2 +
Fin_all(m,1)/Bend_Radius*Mu_Min*(Curved_length-transition_lengths) ;% + Fin_all(offset,1);
  end

%   if Fin_all(m,1) < F_icr_min(m-offset+1)
%       F_1c_min(m-offset+1) = F_1nbc_min(m-
offset+1)*Mu_Min*Long_Straight_Length*correction_factor;
%       Fout_combined_model_min(m-offset+1) = Fin_all(m,1)/exp(Sweep_Angle_Min*Mu_Min) -
Fin_all(offset,1);
%   else
%       F_1c_min(m-offset+1) =
Fin_all(m,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor;
%       Fout_combined_model_min(m-offset+1) = Fin_all(m,1) -
Fin_all(m,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor - Fin_all(offset,1);% +
(Fin_all(transition_point+offset-1,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor -
(Fin_all(transition_point+offset-1,1)/exp(Sweep_Angle_Max*Mu_Max) - Fin_all(offset,1)))/2;% +
Fout_combined_model_min(transition_point) ;%-
Fin_all(transition_point,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor;
%   end

%   F_new_prediction_max(m-offset+1) = Fin_all(m,1) - F_1c_max(m-offset+1) - Fin_all(offset,1);
%   F_new_prediction_min(m-offset+1) = Fin_all(m,1) - F_1c_min(m-offset+1) - Fin_all(offset,1);
```

```
  if (Fin_all(m,1)/exp(Sweep_Angle_Max*Mu_Max)) < (Fin_all(m,1) -
Fin_all(m,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor)
    Fout_combined_model_max(m-offset+1) = Fin_all(m,1)/exp(Sweep_Angle_Max*Mu_Max) -
Fin_all(offset,1);
  else
    Fout_combined_model_max(m-offset+1) = Fin_all(m,1) -
Fin_all(m,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor - Fin_all(offset,1);% +
(Fin_all(transition_point+offset-1,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor -
(Fin_all(transition_point+offset-1,1)/exp(Sweep_Angle_Min*Mu_Min) - Fin_all(offset,1)))/2;% +
Fout_combined_model_max(transition_point) ;%-
Fin_all(transition_point,1)^2*r/(4*E*I)*Mu_Max*Long_Straight_Length*correction_factor;
  end

  if (Fin_all(m,1)/exp(Sweep_Angle_Min*Mu_Min)) < (Fin_all(m,1) -
Fin_all(m,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor)
    Fout_combined_model_min(m-offset+1) = Fin_all(m,1)/exp(Sweep_Angle_Min*Mu_Min) -
Fin_all(offset,1);
  else
    Fout_combined_model_min(m-offset+1) = Fin_all(m,1) -
Fin_all(m,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor - Fin_all(offset,1);% +
(Fin_all(transition_point+offset-1,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor -
(Fin_all(transition_point+offset-1,1)/exp(Sweep_Angle_Min*Mu_Min) - Fin_all(offset,1)))/2;% +
Fout_combined_model_Min(transition_point) ;%-
Fin_all(transition_point,1)^2*r/(4*E*I)*Mu_Min*Long_Straight_Length*correction_factor;
  end

  Straight_and_Curve_model_Fout_Max1(m-offset+1) = Fin_all(m,1) -
Straight_and_Curve_model_Max1(m-offset+1);
  Straight_and_Curve_model_Fout_Min1(m-offset+1) = Fin_all(m,1) -
Straight_and_Curve_model_Min1(m-offset+1);

  if Fin_all(m,1) < E*I*(Bend_Radius-Mu_Max*(Curved_length-
transition_lengths))/(20*Mu_Max*D*r*Bend_Radius)
%       Straight_and_Curve_model_Fout_Max1(m-offset+1) = Fin_all(m,1) -
Straight_and_Curve_model_Max1(m-offset+1);
    Straight_and_Curve_model_Fout_Max2(m-offset+1) = Fin_all(m,1) -
Straight_and_Curve_model_Max2(m-offset+1);
  else
%       Straight_and_Curve_model_Fout_Max1(m-offset+1) = E*I*(Bend_Radius-
Mu_Max*(Curved_length-transition_lengths))^2/(40*Mu_Max*D*r*Bend_Radius^2) - Fin_all(offset,1);
    Straight_and_Curve_model_Fout_Max2(m-offset+1) = E*I*(Bend_Radius-
Mu_Max*(Curved_length-transition_lengths))^2/(40*Mu_Max*D*r*Bend_Radius^2) - Fin_all(offset,1);
  end

  if Fin_all(m,1) < E*I*(Bend_Radius-Mu_Min*(Curved_length-
transition_lengths))/(20*Mu_Min*D*r*Bend_Radius)
%       Straight_and_Curve_model_Fout_Min1(m-offset+1) = Fin_all(m,1) -
Straight_and_Curve_model_Min1(m-offset+1);
    Straight_and_Curve_model_Fout_Min2(m-offset+1) = Fin_all(m,1) -
Straight_and_Curve_model_Min2(m-offset+1);
  else
%       Straight_and_Curve_model_Fout_Min1(m-offset+1) = E*I*(Bend_Radius-
Mu_Min*(Curved_length-transition_lengths))^2/(40*Mu_Min*D*r*Bend_Radius^2);
    Straight_and_Curve_model_Fout_Min2(m-offset+1) = E*I*(Bend_Radius-Mu_Min*(Curved_length-
transition_lengths))^2/(40*Mu_Min*D*r*Bend_Radius^2);
```

```
   end
end

position_interval = max(Position)/20;
for i = 1:20
   Position_reduced(i) = i*position_interval;
end

% sinusoidal_contact_F
% Fout_sine
% Fout_combined_model_min;
%Find curve fit coefficients for Fin and Fout vs. Position
Fin_P_fit_coeffs = polyfit(Position,Fin,6);
Fout_P_fit_coeffs = polyfit(Position,Fout,6);
Fout_P_fit_coeffs_reduced = polyfit(Position,Fout,2);

%Find curve fit values for Fin vs. Fout vs. Position
Fin_P_fit = polyval(Fin_P_fit_coeffs,Position);
Fout_P_fit = polyval(Fout_P_fit_coeffs,Position);
Fout_P_fit_reduced = polyval(Fout_P_fit_coeffs_reduced,Position_reduced);

% %Find curve fit coefficients for Fin vs. Fout
% Fin_fit_coeffs = polyfit(Fout,Fin,6);
%
% %Find curve fit values for Fin vs Fout
% Fin_fit = polyval(Fin_fit_coeffs,Fout,6);

%Find curve fit coefficients for Fout vs. Fin
Fout_fit_coeffs = polyfit(Fin,Fout,6); %%USE THIS ONE

%Find curve fit values for Fout vs Fin
Fout_fit = polyval(Fout_fit_coeffs,Fin,6);

% %Find simple curve fits of Fout vs position and Fin vs Fout
% options.StartPoint = [4 1.1 2 1];
% [Fin_fit_new,Fin_new_stats] = fit(Fout',Fin','a*exp(c*Sweep_Angle_Min*Mu_Min*(x-
b))+Sweep_Angle_Min*Mu_Min*x')%c*x^3+d*x^2+exp(e*Sweep_Angle_Min*Mu_Min*x)*dirac(x-b)
%
% options.StartPoint = [1 5 .2];
[Fout_P_fit_new,Fout_P_new_stats] = fit(Position',Fout','g*x^2+h*x+k'); %a*x^3+

Fout_max = Fout_P_fit_new.k-(Fout_P_fit_new.h)^2/(4*Fout_P_fit_new.g);
Fout_P_fin_info = [Fout_P_fit_new.g Fout_P_fit_new.h Fout_P_fit_new.k Fout_max
Fout_P_new_stats.rsquare];


% Pos_Val_of_max_Fout = fminbnd(@(x) (-Fout_P_fit_new.g*x^2-Fout_P_fit_new.h*x-
Fout_P_fit_new.k), .1, 3);
% Force_Val_of_max_Fout =
Fout_P_fit_new.g*Pos_Val_of_max_Fout^2+Fout_P_fit_new.h*Pos_Val_of_max_Fout+Fout_P_fit_new.
k;

% Fin_P_fit_new.c
% Fin_P_new_stats.rsquare

%Calculate the log of Fin
```

```
for i = 1:size(Fin')
   Fin_subtracted(i) = Fin(i) - Fout(i)*exp(Sweep_Angle_Min*Mu_Min);
%    Fin_log(i) = log(Fin(i));
   Fout_log(i) = log(Fout(i));
%    Fin_P_fit_log(i) = log(Fin_P_fit(i));
   Fout_P_fit_log(i) = log(Fout_P_fit(i));
end


%Calculate Slope of Fout_P_fit_log vs. Fout_P_fit
for m = 1:size(Fout_P_fit_log')-1
%    slope_of_Fin_P_fit_log(m,1) = (Fin_P_fit_log(m+1)-Fin_P_fit_log(m))/(Fout_P_fit(m+1)-
Fout_P_fit(m));
%    Fout9(m,1) = Fout_P_fit (m);
   slope_of_Fout_P_fit_log(m,1) = (Fout_P_fit_log(m+1)-Fout_P_fit_log(m))/(Fin_P_fit(m+1)-
Fin_P_fit(m)); %Use this one
   Fin9(m,1) = Fin_P_fit (m);
end

% slope_of_Fin_P_fit_log_smoothed = smooth(Fout9,slope_of_Fin_P_fit_log,0.05,'loess');
slope_of_Fout_P_fit_log_smoothed = smooth(Fin9,slope_of_Fout_P_fit_log,0.05,'loess'); %Use this one

%Calculate Slope of Fin_P_fit vs. Fout_P_fit
for m = 1:size(Fout_P_fit_log')-2
%    slope_slope_of_Fin_P_fit_log(m,1) = (slope_of_Fin_P_fit_log_smoothed(m+1)-
slope_of_Fin_P_fit_log_smoothed(m))/(Fout_P_fit(m+1)-Fout_P_fit(m));
%    Fout_for_log(m,1) = Fout_P_fit (m);
   slope_slope_of_Fout_P_fit_log(m,1) = (slope_of_Fout_P_fit_log_smoothed(m+1)-
slope_of_Fout_P_fit_log_smoothed(m))/(Fin_P_fit(m+1)-Fin_P_fit(m)); %Use this one
   Fin_for_log(m,1) = Fin_P_fit (m);
end

% slope_slope_of_Fin_P_fit_log;
slope_slope_of_Fout_P_fit_log;



%initialize Log_Slope_Slope_Val2 to zeros
Log_Slope_Slope_Val2 = zeros(1,7);

%Find the value of log of smoothed second derivative of Fin closest to -1
%%%%%Corrected
for m = 1:size(slope_slope_of_Fout_P_fit_log)-1
   if (slope_slope_of_Fout_P_fit_log(m) > -1.01 && slope_slope_of_Fout_P_fit_log(m) < -0.90 &&
(slope_slope_of_Fout_P_fit_log(m+1)-slope_slope_of_Fout_P_fit_log(m))/(Fin_for_log(m+1)-
Fin_for_log(m)) > 0)
      Log_Slope_Slope_Val =
[Fin_P_fit(m),Fout_P_fit(m),Position(m),Fout_P_fit(m)/Fin_P_fit(m),Fout_P_fit_log(m),slope_of_Fout_P
_fit_log_smoothed(m),slope_slope_of_Fout_P_fit_log(m)];
      break
   else
      Log_Slope_Slope_Val = Log_Slope_Slope_Val2;
   end
   Log_Slope_Slope_Val2 = Log_Slope_Slope_Val;
end
```

```
Log_cutoff_F_diffs = Fin_P_fit(m) - Fout_P_fit(m);


%fit to exponential portion of Fin vs Fout curves
[exponential_fit,exponential_fit_stats] = fit(Fin(m:size(Fin'))',Fout_P_fit_log(m:size(Fin'))','Slope*x+yInt');
%relevant curve fit information for exponential portion of Fin vs Fout
exponential_fit_info = [exponential_fit.Slope exponential_fit.Slope/Sweep_Angle_Avg
exponential_fit.Slope/Mu_Avg exponential_fit.Slope/(Sweep_Angle_Avg*Mu_Avg) exponential_fit.yInt
exponential_fit_stats.rsquare];


Log_cutoff_Vals_non_zeroed = Log_Slope_Slope_Val;


% %Calculate theta over test
% for m = 1:size(Fin')
%     theta(m) = log(Fin(m)/Fout(m))/Mu_Avg;
%     theta_actual_Avg(m) = Sweep_Angle_Avg;
%     theta_actual_Max(m) = Sweep_Angle_Max;
%     theta_actual_Min(m) = Sweep_Angle_Min;
% end


% %Create Fout predictions
% Fin_Predicted_Max = Fout*exp(Sweep_Angle_Max*Mu_Max);
% Fin_Predicted_Avg = Fout*exp(Sweep_Angle_Avg*Mu_Avg);
% Fin_Predicted_Min = Fout*exp(Sweep_Angle_Min*Mu_Min);

%Create Fout predictions
Fout_Predicted_Max = Fin/exp(Sweep_Angle_Max*Mu_Max) - Fin_all(offset,1);
Fout_Predicted_Avg = Fin/exp(Sweep_Angle_Avg*Mu_Avg) - Fin_all(offset,1);
Fout_Predicted_Min = Fin/exp(Sweep_Angle_Min*Mu_Min) - Fin_all(offset,1);

%Define important ratios
% Fin_fit_prediction_ratio = Fin_P_fit./Fin_Predicted_Avg;
Fin_fit_Fout_fit_ratio = Fin_P_fit./Fout_P_fit;
Fout_fit_prediction_ratio = Fout_P_fit./Fout_Predicted_Avg;


%Find where Fout is 5% off the predicted values (Fin vs. Fout)
Percent_Diff = Fout_P_fit./Fout_Predicted_Avg - ones(size(Fout_P_fit),1);


for m = 1:size(Percent_Diff')
   if Percent_Diff(m) <= 0.0
      Percent_Diff(m) = 10;
   end
end

min(Percent_Diff);
Percent_Diff;

%Define the number of points to average
N = 11;
n = (N+1)/2;
```

```
% %Calculate Slope of Fin_P_fit vs. Fout_P_fit
% for m = 1:size(Fin_P_fit')-1
%     slope_of_fit_P(m,1) = (Fin_P_fit(m+1)-Fin_P_fit(m))/(Fout_P_fit(m+1)-Fout_P_fit(m));
%     Fout4(m,1) = Fout_P_fit (m);
% end
%
%
% %Eliminate NaN's and INF's from Fin_fit vs. Fout slope
% for m = 2:size(slope_of_fit_P)-1
%     if isnan(slope_of_fit_P(m,1)) == 1 || isinf(slope_of_fit_P(m,1)) == 1
%         slope_of_fit_P(m,1) = (slope_of_fit_P(m-1,1) + slope_of_fit_P(m+1,1))/2;
%     end
% end
%
% %Calculate Second derivative of Fin_P_fit vs. Fout_P_fit
% for m = 1:size(slope_of_fit_P)-1
%     slope_of_slope_P(m,1) = (slope_of_fit_P(m+1)-slope_of_fit_P(m))/(Fout_P_fit(m+1)-
Fout_P_fit(m));
%     Fout6(m,1) = Fout_P_fit (m);
% end
%
% %Smooth second derivative of Fin_P_fit vs. Fout_P_fit
% slope_slope_smoothed = smooth(Fout6,slope_of_slope_P,0.05,'loess');



%Calculate Slope of Fin_P_fit vs. Fout_P_fit %%%USE THIS ONE
for m = 1:size(Fout_P_fit')-1
   slope_of_fit_P2(m,1) = (Fout_P_fit(m+1)-Fout_P_fit(m))/(Fin_P_fit(m+1)-Fin_P_fit(m));
   Fin4(m,1) = Fin_P_fit (m);
end


%Eliminate NaN's and INF's from Fin_fit vs. Fout slope
for m = 2:size(slope_of_fit_P2)-1
   if isnan(slope_of_fit_P2(m,1)) == 1 || isinf(slope_of_fit_P2(m,1)) == 1
      slope_of_fit_P2(m,1) = (slope_of_fit_P2(m-1,1) + slope_of_fit_P2(m+1,1))/2;
   end
end

%Calculate Second derivative of Fin_P_fit vs. Fout_P_fit
for m = 1:size(slope_of_fit_P2)-1
   slope_of_slope_P2(m,1) = (slope_of_fit_P2(m+1)-slope_of_fit_P2(m))/(Fin_P_fit(m+1)-Fin_P_fit(m));
   Fin6(m,1) = Fin_P_fit (m);
end

%Smooth second derivative of Fin_P_fit vs. Fout_P_fit
slope_slope_smoothed2 = smooth(Fin6,slope_of_slope_P2,0.05,'loess');

% %Calculate Slope of Fin_fit vs. Fout
% for m = 1:size(Fin_fit')-1
%     slope_of_fit(m,1) = (Fin_fit(m+1)-Fin_fit(m))/(Fout(m+1)-Fout(m));
%     Fout3(m,1) = Fout (m);
% end
% %Eliminate NaN's and INF's from Fin vs. Position slope
% for m = 2:size(slopeIn_P)-1
%    if isnan(slopeIn_P(m,1)) == 1 | isinf(slopeIn_P(m,1)) == 1
%        slopeIn_P(m,1) = (slopeIn_P(m-1,1) + slopeIn_P(m+1,1))/2;
%    end
% end



% %Smooth slope
% slope_smooth = smooth(Fout2,slope,0.25,'loess',0);

% slope_of_fit;


Cutoff_Val = ones(1,6);
Cutoff_Val2 = zeros(1,6);



%Find the value closest to 5% difference from prediction
for m = 1:size(Fin_P_fit')
   if (Percent_Diff(m) > percent_min && Percent_Diff(m) <= percent_max && Cutoff_Val2(4) <=
Cutoff_Val(4)) && Fout_P_fit(m) > .1
       Cutoff_Val =
[Fin_P_fit(m),Fout_P_fit(m),Position(m),Percent_Diff(m),Fout_fit_prediction_ratio(m),Fin_fit_Fout_fit_ra
tio(m)];
%       break
   else
       Cutoff_Val = Cutoff_Val2;

   end
   Cutoff_Val2 = Cutoff_Val;
end


Slope_Val = ones(1,4);
Slope_Val2 = zeros(1,4);

%Find the value closest to 0.5
for m = 1:size(slope_of_fit_P2)-1
   if (slope_of_fit_P2(m) > 0.45 && slope_of_fit_P2(m) < 0.55 && (slope_of_fit_P2(m+1)-
slope_of_fit_P2(m))/(Position(m+1)-Position(m)) < 0)
       Slope_Val = [Fin_P_fit(m),Fout_P_fit(m),Position(m),slope_of_fit_P2(m)];
%       break
   else
       Slope_Val = Slope_Val2;

   end
   Slope_Val2 = Slope_Val;
end
Slope_Vals = Slope_Val;
```

```
% Slope_Slope_Val2 = zeros(1,6);

% %Find the value of smoothed second derivative closest to 2
% for m = 1:size(slope_slope_smoothed)-1
%    if (slope_slope_smoothed(m) > 1.95 && slope_slope_smoothed(m) < 2.2 &&
(slope_slope_smoothed(m+1)-slope_slope_smoothed(m))/(Fout6(m+1)-Fout6(m)) > 0)
%       Slope_Slope_Val =
[Fin_P_fit(m),Fout_P_fit(m),Position(m),Fin_P_fit(m)/Fout_P_fit(m),slope_of_fit_P(m),slope_slope_smoo
thed(m)];
% %       break
%    else
%       Slope_Slope_Val = Slope_Slope_Val2;
%    end
%    Slope_Slope_Val2 = Slope_Slope_Val;
% end
% Slope_Slope_Vals = Slope_Slope_Val2;



Slope_Slope_Val3 = zeros(1,6);
Slope_Slope_Val4 = zeros(1,6);

%Find the value of smoothed second derivative closest to -.2 THESE ARE THE VALUES I WANT TO
RECORD AND KEEP (JAN 23, 2008)
for m = 1:size(slope_slope_smoothed2)-1
   if (slope_slope_smoothed2(m) > -.2 && slope_slope_smoothed2(m) < -.16 &&
(slope_slope_smoothed2(m+1)-slope_slope_smoothed2(m))/(Fin6(m+1)-Fin6(m)) > 0)
      Slope_Slope_Val4 =
[Fin_P_fit(m),Fout_P_fit(m),Position(m),Fin_P_fit(m)/Fout_P_fit(m),slope_of_fit_P2(m),slope_slope_smo
othed2(m)];
      break
   else
      Slope_Slope_Val3 = Slope_Slope_Val4;
   end
   Slope_Slope_Val4 = Slope_Slope_Val3;
end
Slope_Slope_Vals2 = Slope_Slope_Val4;


% Second_deriv_cutoffs = horzcat(Slope_Slope_Val,Slope_Slope_Val2);

% Predicted_Avg_slope = exp(Sweep_Angle_Avg*Mu_Avg)*ones(size(slope_of_fit_P));
Predicted_Avg_slope2 = ones(size(slope_of_fit_P2))/exp(Sweep_Angle_Avg*Mu_Avg);


if material == C

   % %Find values at 5% slope difference
%    figure
%    plot(Fin4,slope_of_fit_P2,'-',Fin4,Predicted_Avg_slope2,'--')%Fout3,slope_of_fit,
%    axis([0 10 0 1.5])
%    title(['Measured and Predicted First Derivatives',File_name],'fontsize',12,'fontweight','b')
%    xlabel('F_i_n (N)','fontsize',12,'fontweight','b')
%    ylabel('F_i_n vs. F_o_u_t First Derivative','fontsize',12,'fontweight','b')
%    legend('Measured First Derivative','Capstan Predicted First Derivative', 2)
```

```
%
%   figure
%   plot(Fout6,slope_slope_smoothed2)%Fout3,slope_of_fit,Fout6,slope_of_slope_P,
%   axis([0 2.75 0 5])
%   title([File_name,'Measured Forces'],'fontsize',12,'fontweight','b')
%   xlabel('F_o_u_t (N)','fontsize',12,'fontweight','b')
%   ylabel('Second Derivative of F_i_n vs. F_o_u_t','fontsize',12,'fontweight','b')
%   grid
%   legend('Second Derivative',2)%,'Smoothed Second derivative', 2)
%
%   figure
%   plot(Fin6,slope_slope_smoothed2,'-')%Fout3,slope_of_fit,
%   axis([0 10 -1 1])
%   title(['Measured and Predicted Second Derivatives',File_name],'fontsize',12,'fontweight','b')
%   xlabel('F_i_n (N)','fontsize',12,'fontweight','b')
%   ylabel('F_i_n vs. F_o_u_t Second Derivative','fontsize',12,'fontweight','b')
%   legend('Measured Second Derivative','Capstan Predicted Second Derivative', 2)
%
%   figure
%   plot(Fin6,slope_of_slope_P2,Fin6,slope_slope_smoothed2)%Fout3,slope_of_fit,
%   axis([0 10 -2 2])
%   title([File_name,'Measured Forces'],'fontsize',12,'fontweight','b')
%   xlabel('F_i_n (N)','fontsize',12,'fontweight','b')
%   ylabel('Second Derivative of F_i_n vs. F_o_u_t','fontsize',12,'fontweight','b')
%   grid
%   legend('Second Derivative','Smoothed Second derivative', 2)
% % %
%   figure
% %   hold on
% % %   figure
%   plot(Fin,Straight_and_Curve_model_Fout_Min2,'--g',Fin,Fout,'-b',Fin,S
%   traight_and_Curve_model_Fout_Max2,'-.r')%,Straight_and_Curve_model_Fo
%   ut_Max2,Fin,Straight_and_Curve_model_Fout_Min2,Fin)
%   axis([0 10 0 2.75])
% %   plot(Fin_fit_new)
% %   title('Compression Test F_i_n Vs. F_o_u_t','fontsize',12,'fontweight','b')
%   title(['Measured Forces' File_name],'fontsize',12,'fontweight','b')
%   ylabel('F_o_u_t (N)','fontsize',12,'fontweight','b')
%   xlabel('F_i_n (N)','fontsize',12,'fontweight','b')
%   legend('Upper Prediction Boundary','Measured Forces','Lower Prediction Boundary', 2)

%   figure
%   plot(Model_ratio,Fin)

%   figure
%   plot(F_new_prediction_max,Fin)
%   figure
%   plot(F_new_prediction_min,Fin)
%   semilogy(Fin)
%
%   grid
%   legend('F_i_n','Max Capstan Prediction','Min Capstan Prediction',2)%,'Curve Fit', 2)
% %   hold off
% %
%   figure
%    plot(Position,Fin,'-',Position,Fout,'--
')%,Position_reduced,Fout_P_fit_reduced,'x')%,Position,Fin_P_fit)
%    axis([0 1.2 0 10])
% %    title('Compression Test Forces','fontsize',12,'fontweight','b')
%    title(['Measured Forces For Shortened Tube',File_name],'fontsize',12,'fontweight','b')
%    xlabel('Position (cm)','fontsize',12,'fontweight','b')
%    ylabel('Force (N)','fontsize',12,'fontweight','b')
%    legend('F_i_n','F_o_u_t',2)%,'F_o_u_t Fit',2)%,'F_i_n Fit', 2)
%
%    hold on
%    plot(Fout_P_fit_new)
%    hold off

%    figure
%    plot(Position_all,Fout_all,Position_all,Fin_all)
%    axis([0 1.6 0 10])
%    title(['Original Force Measurements',File_name],'fontsize',12,'fontweight','b')
%    xlabel('Position (cm)','fontsize',12,'fontweight','b')
%    ylabel('Force (N)','fontsize',12,'fontweight','b')
%    grid
%    legend('F_o_u_t','F_i_n', 2)

%
   figure
%    plot(Fin,Fout_combined_model_min,'--g',Fin,Fout,'-b',Fin,Fout_combined_model_max,'-.r')
   plot(Fin,Straight_and_Curve_model_Fout_Min2,'--g',Fin,Fout,'-
b',Fin,Straight_and_Curve_model_Fout_Max2,'-.r')%,Fin_fit2,Fout_fit2)
   axis([0 10 0 3])
   title(['Model 3 Prediction of F_i_n Vs. F_o_u_t',File_name],'fontsize',12,'fontweight','b')
   xlabel('F_i_n (N)','fontsize',12,'fontweight','b')
   ylabel('F_o_u_t (N)','fontsize',12,'fontweight','b')
   legend('Upper Prediction Boundary','Measured Forces','Lower Prediction Boundary', 2);%,'Max Capstan
Prediction','Min Capstan Prediction', 2)
%    grid
%
%    figure
%    plot(Position,theta,Position,theta_actual_Avg,Position,theta_actual_Max,Position,theta_actual_Min)
%    axis([0 1.2 0 10])
%    title([File_name,'Measured Forces'],'fontsize',12,'fontweight','b')
%    xlabel('Position (cm)','fontsize',12,'fontweight','b')
%    ylabel('Sweep Angle (rad)','fontsize',12,'fontweight','b')
%    grid

%    figure
%    plot(Fin, Fout_log)
%    axis([0 2.5 -2.5 2.5])
%    grid

%
%    figure
%    plot(Fin9,slope_of_Fout_P_fit_log)
%    axis([0 10 -3 3])
%    xlabel('F_i_n (N)','fontsize',12,'fontweight','b')
%    ylabel('First Derivative of Log(F_o_u_t) Vs. F_i_n','fontsize',12,'fontweight','b')
%    title(['First Derivative of F_i_n Vs. Log(F_o_u_t)',File_name],'fontsize',12,'fontweight','b')
%
```

```
%    figure
%    plot(Fin_for_log,slope_slope_of_Fout_P_fit_log)
%    axis([0 10 -3 3])
%    xlabel('F_i_n (N)','fontsize',12,'fontweight','b')
%    ylabel('Second Derivative of Log(F_o_u_t) (N)','fontsize',12,'fontweight','b')
%    title(['Second Derivative of F_i_n Vs. Log(F_o_u_t)',File_name],'fontsize',12,'fontweight','b')

else
     % %Find values at 5% slope difference
%    figure
%    plot(Fout4,Predicted_Avg_slope,Fout4,slope_of_fit_P)%Fout3,slope_of_fit,
%    axis([0 5 0 5])
%    title([File_name,'Measured Forces'],'fontsize',12,'fontweight','b')
%    xlabel('F_o_u_t (N)','fontsize',12,'fontweight','b')
%    ylabel('F_i_n vs. F_o_u_t Slope','fontsize',12,'fontweight','b')
%    grid
%    legend('Predicted Capstan Slope','Measured Slope', 2)
%
%    figure
%    plot(Fout6,slope_of_slope_P,Fout6,slope_slope_smoothed)%Fout3,slope_of_fit,
%    axis([0 5 0 5])
%    title([File_name,'Measured Forces'],'fontsize',12,'fontweight','b')
%    xlabel('F_o_u_t (N)','fontsize',12,'fontweight','b')
%    ylabel('Second Derivative of F_i_n vs. F_o_u_t','fontsize',12,'fontweight','b')
%    grid
%    legend('Second Derivative','Smoothed Second derivative', 2)
%
%    figure
%    plot(Fin4,Predicted_Avg_slope2,Fin4,slope_of_fit_P2)%Fout3,slope_of_fit,
%    axis([0 10 0 1.5])
%    title([File_name,'Measured Forces'],'fontsize',12,'fontweight','b')
%    xlabel('F_i_n (N)','fontsize',12,'fontweight','b')
%    ylabel('Slope','fontsize',12,'fontweight','b')
%    grid
%    legend('Predicted Capstan Slope','Measured Slope', 2)
%
%    figure
%    plot(Fin6,slope_of_slope_P2,Fin6,slope_slope_smoothed2)%Fout3,slope_of_fit,
%    axis([0 10 -3 1])
%    title([File_name,'Measured Forces'],'fontsize',12,'fontweight','b')
%    xlabel('F_i_n (N)','fontsize',12,'fontweight','b')
%    ylabel('Second Derivative of F_i_n vs. F_o_u_t','fontsize',12,'fontweight','b')
%    grid
%    legend('Second Derivative','Smoothed Second derivative', 2)
%
   figure
%    plot(Fin,Fout_combined_model_min,'--g',Fin,Fout,'-b',Fin,Fout_combined_model_max,'-.r')
   plot(Fin,Straight_and_Curve_model_Fout_Min2,'--g',Fin,Fout,'-
b',Fin,Straight_and_Curve_model_Fout_Max2,'-
.r')%,Straight_and_Curve_model_Fout_Max2,Fin,Straight_and_Curve_model_Fout_Min2,Fin)
   %plot(Fout,Fin,Fout,Fin_Predicted_Max,Fout,Fin_Predicted_Min)
   axis([0 10 0 5])
   title(['Model 3 Prediction of F_i_n Vs. F_o_u_t' File_name],'fontsize',12,'fontweight','b')
   ylabel('F_o_u_t (N)','fontsize',12,'fontweight','b')
   xlabel('F_i_n (N)','fontsize',12,'fontweight','b')
   legend('Upper Prediction Boundary','Measured Forces','Lower Prediction Boundary', 2)
```

```
%    grid

%    figure
%    plot(Model_ratio,Fin)
%
%    figure
%    plot(F_new_prediction_max,Fin)
%    figure
%    plot(F_new_prediction_min,Fin)
%    legend('F_i_n','Max F_i_n Prediction','Min F_i_n Prediction','Curve Fit', 2)
%
%    figure
%    hold on
%
plot(Fout,Fin,Fout,Fin_Predicted_Max,Fout,Fin_Predicted_Min)%,Fout_P_fit,Fin_P_fit)%,Fout,Fin_Predi
cted_Avg
%    axis([0 5 0 10])
% %        plot(Fin_fit_new)
%    title([File_name,'Measured Forces'],'fontsize',12,'fontweight','b')
%    xlabel('F_o_u_t (N)','fontsize',12,'fontweight','b')
%    ylabel('F_i_n (N)','fontsize',12,'fontweight','b')
%    grid
%    legend('F_i_n','Max F_i_n Prediction','Min F_i_n Prediction',2)%'Curve Fit', 2)
%    hold off

%    figure
%    plot(Position,Fout,Position,Fin)%,Position,Fout_P_fit,Position,Fin_P_fit)
%    axis([0 .6 0 10])
%    title([File_name,'Measured Forces'],'fontsize',12,'fontweight','b');
%    xlabel('Position (cm)','fontsize',12,'fontweight','b')
%    ylabel('Force (N)','fontsize',12,'fontweight','b')
%    grid
%    legend('F_o_u_t','F_i_n',2)%,'F_o_u_t Fit','F_i_n Fit', 2)

%    hold on
%    plot(Fout_P_fit_new)
%    hold off
%
%    figure
%    plot(Position,Fin,'-',Position,Fout,'--')%,Position,Fout_P_fit,Position,Fin_P_fit)
%    axis([0 .6 0 10])
%    title(['Measured Forces',File_name,],'fontsize',12,'fontweight','b');
%    xlabel('Position (cm)','fontsize',12,'fontweight','b')
%    ylabel('Force (N)','fontsize',12,'fontweight','b')
%    legend('F_i_n','F_o_u_t',2)%'F_o_u_t Fit','F_i_n Fit', 2)
%
%    figure
%    plot(Position_all,Fin_all,'-',Position_all,Fout_all,'--')
%    axis([0 1.25 0 10])
%    title(['Original Force Measurements',File_name],'fontsize',12,'fontweight','b');
%    xlabel('Position (cm)','fontsize',12,'fontweight','b')
%    ylabel('Force (N)','fontsize',12,'fontweight','b')
%    legend('F_i_n','F_o_u_t', 2)

%
%    figure
```

```
%
plot(Fin,Fout,Fin_Predicted_Max,Fout,Fin_Predicted_Avg,Fout,Fin_Predicted_Min,Fout,Fin_P_fit,Fout_P
_fit)%,Fin_fit2,Fout_fit2)
%    axis([0 10 0 5])
%    title([File_name,'Measured Forces'],'fontsize',12,'fontweight','b')
%    xlabel('F_i_n (N)','fontsize',12,'fontweight','b')
%    ylabel('F_o_u_t (N)','fontsize',12,'fontweight','b')
%    grid
%
%    figure
%    plot(Position,theta,Position,theta_actual_Avg,Position,theta_actual_Max,Position,theta_actual_Min)
%    axis([0 1.25 0 10])
%    title([File_name,'Measured Forces'],'fontsize',12,'fontweight','b')
%    xlabel('Position (cm)','fontsize',12,'fontweight','b')
%    ylabel('Sweep Angle (rad)','fontsize',12,'fontweight','b')
%    grid

%    figure
%    plot(Fin, Fout_log)
%    axis([0 6 -2.5 2.5])
%    grid
%
%    figure
%    plot(Fin_for_log,slope_of_Fout_P_fit_log)
%    axis([0 6 -10 10])
%    grid
%
%    figure
%    plot(Fout_for_log,slope_slope_of_Fin_P_fit_log)
%    axis([0 6 -10 10])
%    grid

end

E*I*(Bend_Radius-Mu_Min*(Curved_length-transition_lengths))/(20*Mu_Min*D*r*Bend_Radius)

Cutoff_Vals_non_zeroed = horzcat(Cutoff_Val,Slope_Val,Slope_Slope_Vals2);
%%,Slope_Slope_Val,Slope_Slope_Val2);
```

Matlab file for processing scanned images

File Name: scanning_tests.m

```
clear all;
close all;
clc;

dpi = 1200; %pixel/inch

tube_wall = 0.005; % Wall thickness in inches
tube_ID = 0.048; %Tube ID in inches
tube_OD = 0.058; %Tube ID in inches

tube_centerline_offset = tube_wall + tube_ID/2;

Nylon_thickness = 0.0355; %Nylon tubing OD in inches
PTFE_thickness = 0.038; %PTFE monofilament in inches

Nylon_upper_offset = tube_wall + tube_ID - Nylon_thickness/2;
Nylon_lower_offset = tube_wall + Nylon_thickness/2;

PTFE_upper_offset = tube_wall + tube_ID - PTFE_thickness/2;
PTFE_lower_offset = tube_wall + PTFE_thickness/2;

% Tube_number = 9;

Files ={'Angled1_cropped.png'; 'Angled2b_cropped.png'; 'Angled3_cropped.png'; 'Angled4_cropped.png';
'Angled11d_cropped_rev.png'; 'Angled12_cropped_rev.png'; 'Angled13_cropped_rev.png';
'Angled14_cropped_rev.png'};

for Tube_number = 1:2;%size(Files,1)
%%
   clear pic_original bw_matrix filtered_image B L
   clear vert_size_vals horiz_size_vals sorted_vert_size_vals Array_number
   clear boundary3 rev_boundary3 half_size_boundary3 start_of_half_rev_boundary3 counter
end_of_half_rev_boundary3
   clear half_rev_boundary3 half_size smoothed_y_half_rev_boundary3 smoothed_x_half_rev_boundary3
dy dx
   clear slope smoothed_slope slope2 smoothed_slope2
   clear maxtab mintab
   clear avg_final_slope avg_initial_point avg_final_point all_peaks %avg_initial_slope
   clear maxtab_old mintab_old
   clear Long_S_Length Short_S_Length Total_L
   clear x_test_points y_test_points half_slope b b2
   clear y_tube_Centerline x_tube_Centerline
   clear x_Nylon_upper_Centerline y_Nylon_upper_Centerline x_PTFE_upper_Centerline
y_PTFE_upper_Centerline
   clear x_Nylon_lower_Centerline y_Nylon_lower_Centerline x_PTFE_lower_Centerline
y_PTFE_lower_Centerline
   clear centerline_slope smoothed_centerline_slope centerline_slope2 smoothed_centerline_slope2
   clear x_Nylon_intersect y_Nylon_intersect x_Nylon_intersect_old y_Nylon_intersect_old
x_Nylon_intersect_old2 y_Nylon_intersect_old2
   clear x_Nylon_curve y_Nylon_curve x_Nylon_line y_Nylon_line New_max_Nylon_slope
   clear x_PTFE_intersect y_PTFE_intersect x_PTFE_intersect_old y_PTFE_intersect_old
x_PTFE_intersect_old2 y_PTFE_intersect_old2
```

```
  clear x_PTFE_curve y_PTFE_curve x_PTFE_line y_PTFE_line New_max_PTFE_slope
  clear x_Nylon_intersect2 y_Nylon_intersect2 x_Nylon_intersect_old3 y_Nylon_intersect_old3
 x_Nylon_intersect_old4 y_Nylon_intersect_old4
  clear x_Nylon_curve2 y_Nylon_curve2 x_Nylon_line2 y_Nylon_line2
  clear x_PTFE_intersect2 y_PTFE_intersect2 x_PTFE_intersect_old3 y_PTFE_intersect_old3
 x_PTFE_intersect_old4 y_PTFE_intersect_old4
  clear x_PTFE_curve2 y_PTFE_curve2 x_PTFE_line2 y_PTFE_line2
  clear x_Nylon_intersect3 y_Nylon_intersect3 x_Nylon_intersect_old5 y_Nylon_intersect_old5
 x_Nylon_intersect_old6 y_Nylon_intersect_old6
  clear x_Nylon_curve3 y_Nylon_curve3 x_Nylon_line3 y_Nylon_line3
  clear x_PTFE_intersect3 y_PTFE_intersect3 x_PTFE_intersect_old5 y_PTFE_intersect_old5
 x_PTFE_intersect_old6 y_PTFE_intersect_old6
  clear x_PTFE_curve3 y_PTFE_curve3 x_PTFE_line3 y_PTFE_line3
  clear Min_PTFE_Sweep_slopes Min_Nylon_Sweep_slopes

  if Tube_number == 25
    pic_original = imrotate(imread(Files{Tube_number}),1);
  else
    pic_original = imread(Files{Tube_number});
  end

  bw_matrix = 255*(pic_original > 200);
  filtered_image = medfilt2(bw_matrix,[3 10]);
  [B,L] = bwboundaries(bw_matrix,'holes');
  %%%%%%filter the image: L = medfilt2(bw_matrix,[3 10]);

%%
  % bw_image = im2bw(pic_original);

%    %%%% Fined array number in cell B that corresponds to the respective lower loop
%    for i = 2:size(B,1)
%      [vert_size_vals(i-1,1) horiz_size_vals(i-1,1)] = size(B{i});
%    end
%
%    sorted_vert_size_vals = sort(vert_size_vals,'descend');
%
%    for i = 1:size(sorted_vert_size_vals,1)
%      if vert_size_vals(i) == sorted_vert_size_vals(2)
%        Array_number = i + 1
%      end
%    end

  %%%% Fined array number in cell B that corresponds to the respective lower loop
  for i = 2:size(B,1)
    [vert_size_vals(i-1,1) horiz_size_vals(i-1,1)] = size(B{i});
  end

  sorted_vert_size_vals = sort(vert_size_vals,'descend');

  for i = 1:size(sorted_vert_size_vals,1)
    if vert_size_vals(i) >= 10*dpi
      Array_number = i + 1
    end
  end

%    figure
%    imshow(filtered_image);
%
%    hold on
%
%    for k = 1:length(B)
%       boundary = B{k};
%    %    B_sizes(k) = size(B{k});
%       plot(boundary(:,2), boundary(:,1), 'b', 'LineWidth', 2)
%    end
%%
   boundary3 = B{Array_number}; %Extract lower loop
   rev_boundary3 = [wrev(boundary3(:,2)) wrev(boundary3(:,1))]; %reverse order of lower loop
   half_size_boundary3 = round(size(boundary3)/2); %Find half the size of lower loop

   start_of_half_rev_boundary3 = round(0.004*half_size_boundary3(1)); %define the number of points to
cut off at the beginning of the string


   counter = half_size_boundary3(1) - round(0.075*half_size_boundary3(1));

   while rev_boundary3(counter,1) <= rev_boundary3(counter+5,1)
      end_of_half_rev_boundary3 = counter;
      counter = counter + 1;
   end

%    %define the number of points to cut off at the end of the string
%    if Tube_number == 9
%       end_of_half_rev_boundary3 = half_size(1) + round(0.015*half_size(1));
%    else
%       end_of_half_rev_boundary3 = half_size(1);
%    end

   half_rev_boundary3 = rev_boundary3(start_of_half_rev_boundary3:end_of_half_rev_boundary3,1:2);
%Take half the reversed lower loop, matrix is now in x,y...

   half_size = size(half_rev_boundary3);

   smoothed_y_half_rev_boundary3 = smooth(half_rev_boundary3(:,2),round(0.02*half_size(1)),'loess');
   smoothed_x_half_rev_boundary3 = smooth(half_rev_boundary3(:,1),round(0.02*half_size(1)),'loess');
%This is the new Y-value matrix

   % plot(half_rev_boundary3(:,1), smoothed_half_rev_boundary3(:,1), 'b', 'LineWidth', 2)
   %
   % hold off

%%

   %%%Calculate the slope of the boundary
   for m = 1:size(smoothed_x_half_rev_boundary3) - round(.003*half_size(1))
      dy = (smoothed_y_half_rev_boundary3(m+round(.003*half_size(1)))-
smoothed_y_half_rev_boundary3(m));
      dx = (smoothed_x_half_rev_boundary3(m+round(.003*half_size(1)))-
smoothed_x_half_rev_boundary3(m,1));

      if dy == 0 && dx == 0
         slope(m) = 0;
```

```
    elseif dx == 0
      slope(m) = 0;
    else
      slope(m) = dy/dx;
    end
  end

  smoothed_slope = smooth(slope,round(.015*half_size(1)),'moving'); %This is the smoothed Y-value of
the slope

  %%%Calculate the second derivative of the boundary
  for m = 1:size(smoothed_x_half_rev_boundary3) - 2*round(.003*half_size(1))
    dy = (smoothed_slope(m+round(.003*half_size(1)))-smoothed_slope(m));
    dx = (smoothed_x_half_rev_boundary3(m+round(.003*half_size(1)),1)-
smoothed_x_half_rev_boundary3(m,1));

    if dy == 0 && dx == 0
      slope2(m) = 0;
    elseif dx == 0
      slope2(m) = 0;
    else
      slope2(m) = dy/dx;
    end
  end

  smoothed_slope2 = smooth(slope2,round(.015*half_size(1)),'moving');

%%
  %Calculate the offset curves
  y_test_points = smoothed_y_half_rev_boundary3(1:size(smoothed_slope))./dpi;
  x_test_points = smoothed_x_half_rev_boundary3(1:size(smoothed_slope))./dpi;
  half_slope = slope(1:size(smoothed_slope))';

  b = y_test_points - half_slope.*x_test_points;
  b2 = y_test_points + x_test_points./half_slope;

  for i = 1:size(half_slope)
    if half_slope(i) >= 0
      x_tube_Centerline(i,1) = x_test_points(i) - tube_centerline_offset/(1+(1/half_slope(i))^2)^.5;
      y_tube_Centerline(i,1) = b2(i) - x_tube_Centerline(i)/half_slope(i);

      x_Nylon_upper_Centerline(i,1) = x_test_points(i) - Nylon_upper_offset/(1+(1/half_slope(i))^2)^.5;
      y_Nylon_upper_Centerline(i,1) = b2(i) - x_Nylon_upper_Centerline(i)/half_slope(i);
      x_Nylon_lower_Centerline(i,1) = x_test_points(i) - Nylon_lower_offset/(1+(1/half_slope(i))^2)^.5;
      y_Nylon_lower_Centerline(i,1) = b2(i) - x_Nylon_lower_Centerline(i)/half_slope(i);

      x_PTFE_upper_Centerline(i,1) = x_test_points(i) - PTFE_upper_offset/(1+(1/half_slope(i))^2)^.5;
      y_PTFE_upper_Centerline(i,1) = b2(i) - x_PTFE_upper_Centerline(i)/half_slope(i);
      x_PTFE_lower_Centerline(i,1) = x_test_points(i) - PTFE_lower_offset/(1+(1/half_slope(i))^2)^.5;
      y_PTFE_lower_Centerline(i,1) = b2(i) - x_PTFE_lower_Centerline(i)/half_slope(i);
    else
      x_tube_Centerline(i,1) = x_test_points(i) + tube_centerline_offset/(1+(1/half_slope(i))^2)^.5;
      y_tube_Centerline(i,1) = b2(i) - x_tube_Centerline(i)/half_slope(i);

      x_Nylon_upper_Centerline(i,1) = x_test_points(i) + Nylon_upper_offset/(1+(1/half_slope(i))^2)^.5;
      y_Nylon_upper_Centerline(i,1) = b2(i) - x_Nylon_upper_Centerline(i)/half_slope(i);
      x_Nylon_lower_Centerline(i,1) = x_test_points(i) + Nylon_lower_offset/(1+(1/half_slope(i))^2)^.5;
      y_Nylon_lower_Centerline(i,1) = b2(i) - x_Nylon_lower_Centerline(i)/half_slope(i);

      x_PTFE_upper_Centerline(i,1) = x_test_points(i) + PTFE_upper_offset/(1+(1/half_slope(i))^2)^.5;
      y_PTFE_upper_Centerline(i,1) = b2(i) - x_PTFE_upper_Centerline(i)/half_slope(i);
      x_PTFE_lower_Centerline(i,1) = x_test_points(i) + PTFE_lower_offset/(1+(1/half_slope(i))^2)^.5;
      y_PTFE_lower_Centerline(i,1) = b2(i) - x_PTFE_lower_Centerline(i)/half_slope(i);
    end
  end


%%
  %Calculate Centerline slopes
  for m = 1:size(x_tube_Centerline) - round(.003*half_size(1))
    dy = (y_tube_Centerline(m+round(.003*half_size(1)))-y_tube_Centerline(m));
    dx = (x_tube_Centerline(m+round(.003*half_size(1)))-x_tube_Centerline(m,1));

    if dy == 0 && dx == 0
      centerline_slope(m) = 0;
    elseif dx == 0
      centerline_slope(m) = 0;
    else
      centerline_slope(m) = dy/dx;
    end
  end

  smoothed_centerline_slope = smooth(centerline_slope,round(.015*half_size(1)),'moving'); %This is the
smoothed Y-value of the slope

  %%%Calculate the second derivative of the boundary
  for m = 1:size(x_tube_Centerline) - 2*round(.003*half_size(1))
    dy = (smoothed_centerline_slope(m+round(.003*half_size(1)))-smoothed_centerline_slope(m));
    dx = (x_tube_Centerline(m+round(.003*half_size(1)),1)-x_tube_Centerline(m,1));

    if dy == 0 && dx == 0
      centerline_slope2(m) = 0;
    elseif dx == 0
      centerline_slope2(m) = 0;
    else
      centerline_slope2(m) = dy/dx;
    end
  end

  smoothed_centerline_slope2 = smooth(centerline_slope2,round(.015*half_size(1)),'moving');
%%
  %Calculate the peak slope values
  [maxtab, mintab] = peakdet(smoothed_centerline_slope, 0.1);



%%

  %Calculate the total angular displacement of boundary

  avg_initial_slope(Tube_number) =
mean(smoothed_centerline_slope(round(.1*half_size(1)):round(.3*half_size(1))));
```

```
 avg_final_slope =
mean(smoothed_centerline_slope(round(.965*half_size(1)):round(.975*half_size(1))));

 avg_initial_point = round((.1*half_size(1)+.3*half_size(1))/2);
 avg_final_point = round((.965*half_size(1)+.975*half_size(1))/2);


%%

 %Eliminate any extra peaks from mintab and maxtab
 if size(maxtab,1) + size(mintab,1) > 3
  maxtab_old = maxtab;
  mintab_old = mintab;
  maxtab2 = [];
  mintab2 = [];

  new_row_val = 1;
  old_row_val = 1;
  void_range = (max(maxtab(:,2)) - avg_initial_slope(Tube_number))/5;

  for n = 1:size(maxtab,1)
   if maxtab(n,2) <= (avg_initial_slope(Tube_number) + void_range) && maxtab(n,2) >=
(avg_initial_slope(Tube_number) - void_range)
    old_row_val = old_row_val + 1;
   else
    maxtab2(new_row_val,:) = maxtab_old(old_row_val,:);
    new_row_val = new_row_val + 1;
    old_row_val = old_row_val + 1;
   end
  end

  new_row_val = 1;
  old_row_val = 1;
  for n = 1:size(mintab,1)
   if mintab(n,2) <= (avg_initial_slope(Tube_number) + void_range) && mintab(n,2) >=
(avg_initial_slope(Tube_number) - void_range)
    old_row_val = old_row_val + 1;
   else
    mintab2(new_row_val,:) = mintab_old(old_row_val,:);
    new_row_val = new_row_val + 1;
    old_row_val = old_row_val + 1;
   end
  end

  maxtab = maxtab2;
  mintab = mintab2;
 end

%%

%   figure
%   hold on
%
%   plot(smoothed_x_half_rev_boundary3(1:size(smoothed_centerline_slope),1),
smoothed_centerline_slope, 'b', 'LineWidth', 2)
%   plot(smoothed_x_half_rev_boundary3(mintab(:,1)), mintab(:,2), 'g*');
%    plot(smoothed_x_half_rev_boundary3(maxtab(:,1)), maxtab(:,2), 'r*');
%    hold off

%%

  % sort rows after combining maxtab and mintab to get in order
  all_peaks = sortrows(cat(1,maxtab,mintab,[avg_initia-l_point
avg_initial_slope(Tube_number)],[avg_final_point avg_final_slope]));

  total_angle(Tube_number) = 0; %-avg_initial_slope(Tube_number);

  %Calculate the angle in radians between the extreme slopes  %% angle = ATAN((B20-
B19)/(1+B19*B20))
  for n = 1:size(all_peaks,1)-1
    if 0.1*half_size(1) < abs(all_peaks(n,1)) < 0.995*half_size(1)
      total_angle(Tube_number) = total_angle(Tube_number) + abs(atan((all_peaks(n+1,2)-
all_peaks(n,2))/(1+all_peaks(n+1,2)*all_peaks(n,2))));
    end
  end



%%

  Long_S_Length = 0;
  Short_S_Length = 0;
  Total_L = 0;

  %Integrate along the lengths
  % I can probably get rid of one block of these calculations since they should give the same results

  n = 1;

  while abs(smoothed_centerline_slope2(n)) < 0.0001 || n <
round(0.2*size(smoothed_y_half_rev_boundary3,1))
    Long_S_Length = Long_S_Length + (((smoothed_x_half_rev_boundary3(n+1,1)-
smoothed_x_half_rev_boundary3(n,1))^2+(smoothed_y_half_rev_boundary3(n+1)-
smoothed_y_half_rev_boundary3(n))^2)^.5);
    n = n+1;
  end


  n = size(smoothed_centerline_slope2,1)-1;
  while abs(smoothed_centerline_slope2(n)) < 0.0002 || n >
round(0.98*size(smoothed_centerline_slope2,1))
    Short_S_Length = Short_S_Length + (((smoothed_x_half_rev_boundary3(n+1,1)-
smoothed_x_half_rev_boundary3(n,1))^2+(smoothed_y_half_rev_boundary3(n+1)-
smoothed_y_half_rev_boundary3(n))^2)^.5);
    n = n-1;
  end

  for n = 1:size(smoothed_y_half_rev_boundary3)-1
    Total_L = Total_L + (((smoothed_x_half_rev_boundary3(n+1,1)-
smoothed_x_half_rev_boundary3(n,1))^2+(smoothed_y_half_rev_boundary3(n+1)-
smoothed_y_half_rev_boundary3(n))^2)^.5);
  end
  n = 1;
  while abs(smoothed_centerline_slope2(n)) < 0.0001 || n < round(0.2*size(y_tube_Centerline,1))
    New_Long_S_Length = Long_S_Length + (((x_tube_Centerline(n+1,1)-
x_tube_Centerline(n,1))^2+(y_tube_Centerline(n+1)-y_tube_Centerline(n))^2)^.5);
    n = n+1;
  end


  n = size(smoothed_centerline_slope2,1)-1;
  while abs(smoothed_centerline_slope2(n)) < 0.0002 || n >
round(0.98*size(smoothed_centerline_slope2,1))
    New_Short_S_Length = Short_S_Length + (((x_tube_Centerline(n+1,1)-
x_tube_Centerline(n,1))^2+(y_tube_Centerline(n+1)-y_tube_Centerline(n))^2)^.5);
    n = n-1;
  end

  for n = 1:size(y_tube_Centerline)-1
    New_Total_L = Total_L + (((x_tube_Centerline(n+1,1)-
x_tube_Centerline(n,1))^2+(y_tube_Centerline(n+1)-y_tube_Centerline(n))^2)^.5);
  end

%%
% Nylon loop #1

  x_Nylon_intersect = [0;0];
  y_Nylon_intersect = [0;0];
  x_Nylon_intersect_old = [0;0];
  y_Nylon_intersect_old = [0;0];

  i = all_peaks(size(all_peaks,1)-1,1) + round(.05*size(x_Nylon_upper_Centerline,1));%
size(x_Nylon_upper_Centerline,1);%
  while isempty(x_Nylon_intersect) == 0 && isempty(x_Nylon_intersect_old) == 0

    clear x_Nylon_intersect_old2 y_Nylon_intersect_old2
    x_Nylon_intersect_old2 = x_Nylon_intersect_old;
    y_Nylon_intersect_old2 = y_Nylon_intersect_old;

    clear x_Nylon_intersect_old y_Nylon_intersect_old
    x_Nylon_intersect_old = x_Nylon_intersect;
    y_Nylon_intersect_old = y_Nylon_intersect;

    if all_peaks(size(all_peaks,1)-1,2) - avg_initial_slope(Tube_number) > 0
%smoothed_centerline_slope2(all_peaks(size(all_peaks,1)-1,1)) < 0 %
      x_Nylon_curve = x_Nylon_upper_Centerline;
      y_Nylon_curve = y_Nylon_upper_Centerline;
      x_Nylon_line = [x_Nylon_lower_Centerline(i,1); 0];
      y_Nylon_line = [y_Nylon_lower_Centerline(i,1); b(i,1)];
    else
      x_Nylon_curve = x_Nylon_lower_Centerline;
      y_Nylon_curve = y_Nylon_lower_Centerline;
      x_Nylon_line = [x_Nylon_upper_Centerline(i,1); 0];
      y_Nylon_line = [y_Nylon_upper_Centerline(i,1); b(i,1)];
    end
```

```
        [x_Nylon_intersect, y_Nylon_intersect] =
polyxpoly(x_Nylon_curve,y_Nylon_curve,x_Nylon_line,y_Nylon_line);
        i = i-1;
    end

    i
    Nylon_transition_length(Tube_number) = ((mean(x_Nylon_intersect_old2)-x_Nylon_line(1))^2 +
(mean(y_Nylon_intersect_old2)-y_Nylon_line(1))^2)^.5
    New_max_Nylon_slope(3,:) = [i,slope(i)];

%%
    % PTFE loop #1

    x_PTFE_intersect = [0;0];
    y_PTFE_intersect = [0;0];
    x_PTFE_intersect_old = [0;0];
    y_PTFE_intersect_old = [0;0];

    %i = all_peaks(size(all_peaks,1)-1,1) + round(.1*size(x_PTFE_upper_Centerline,1));%
size(x_PTFE_upper_Centerline,1);%
    while isempty(x_PTFE_intersect) == 0 && isempty(x_PTFE_intersect_old) == 0

        clear x_PTFE_intersect_old2 y_PTFE_intersect_old2
        x_PTFE_intersect_old2 = x_PTFE_intersect_old;
        y_PTFE_intersect_old2 = y_PTFE_intersect_old;

        clear x_PTFE_intersect_old y_PTFE_intersect_old
        x_PTFE_intersect_old = x_PTFE_intersect;
        y_PTFE_intersect_old = y_PTFE_intersect;

        if all_peaks(size(all_peaks,1)-1,2) - avg_initial_slope(Tube_number) > 0
%smoothed_centerline_slope2(all_peaks(size(all_peaks,1)-1,1)) < 0 %
            x_PTFE_curve = x_PTFE_upper_Centerline;
            y_PTFE_curve = y_PTFE_upper_Centerline;
            x_PTFE_line = [x_PTFE_lower_Centerline(i,1); 0];
            y_PTFE_line = [y_PTFE_lower_Centerline(i,1); b(i,1)];
        else
            x_PTFE_curve = x_PTFE_lower_Centerline;
            y_PTFE_curve = y_PTFE_lower_Centerline;
            x_PTFE_line = [x_PTFE_upper_Centerline(i,1); 0];
            y_PTFE_line = [y_PTFE_upper_Centerline(i,1); b(i,1)];
        end
        [x_PTFE_intersect, y_PTFE_intersect] =
polyxpoly(x_PTFE_curve,y_PTFE_curve,x_PTFE_line,y_PTFE_line);
        i = i-1;
    end

    i
    PTFE_transition_length(Tube_number) = ((mean(x_PTFE_intersect_old2)-x_PTFE_line(1))^2 +
(mean(y_PTFE_intersect_old2)-y_PTFE_line(1))^2)^.5
    New_max_PTFE_slope(3,:) = [i,slope(i)];

%%
    % Nylon loop #2

    x_Nylon_intersect2 = [0;0];
```

```
  y_Nylon_intersect2 = [0;0];
  x_Nylon_intersect_old3 = [0;0];
  y_Nylon_intersect_old3 = [0;0];

  i = all_peaks(size(all_peaks,1)-2,1) + round(.05*size(x_Nylon_upper_Centerline,1));
%size(x_Nylon_upper_Centerline,1); size(x_Nylon_upper_Centerline,1);%
  while isempty(x_Nylon_intersect2) == 0 && isempty(x_Nylon_intersect_old3) == 0

    clear x_Nylon_intersect_old4 y_Nylon_intersect_old4
    x_Nylon_intersect_old4 = x_Nylon_intersect_old3;
    y_Nylon_intersect_old4 = y_Nylon_intersect_old3;

    clear x_Nylon_intersect_old3 y_Nylon_intersect_old3
    x_Nylon_intersect_old3 = x_Nylon_intersect2;
    y_Nylon_intersect_old3 = y_Nylon_intersect2;

    if all_peaks(size(all_peaks,1)-1,2) - avg_initial_slope(Tube_number) < 0
%smoothed_centerline_slope2(all_peaks(size(all_peaks,1)-1,1)) > 0 %
      x_Nylon_curve2 = x_Nylon_upper_Centerline;
      y_Nylon_curve2 = y_Nylon_upper_Centerline;
      x_Nylon_line2 = [x_Nylon_lower_Centerline(i,1); 0];
      y_Nylon_line2 = [y_Nylon_lower_Centerline(i,1); b(i,1)];
    else
      x_Nylon_curve2 = x_Nylon_lower_Centerline;
      y_Nylon_curve2 = y_Nylon_lower_Centerline;
      x_Nylon_line2 = [x_Nylon_upper_Centerline(i,1); 0];
      y_Nylon_line2 = [y_Nylon_upper_Centerline(i,1); b(i,1)];
    end
    [x_Nylon_intersect2, y_Nylon_intersect2] =
polyxpoly(x_Nylon_curve2,y_Nylon_curve2,x_Nylon_line2,y_Nylon_line2);
    i = i-1;
  end

  i
  Nylon_transition_length(Tube_number) = Nylon_transition_length(Tube_number) +
((mean(x_Nylon_intersect_old4)-x_Nylon_line2(1))^2 + (mean(y_Nylon_intersect_old4)-
y_Nylon_line2(1))^2)^.5
  New_max_Nylon_slope(2,:) = [i,slope(i)];

%%
  % PTFE loop #2

  x_PTFE_intersect2 = [0;0];
  y_PTFE_intersect2 = [0;0];
  x_PTFE_intersect_old3 = [0;0];
  y_PTFE_intersect_old3 = [0;0];

  % i = all_peaks(size(all_peaks,1)-2,1) + round(.1*size(x_PTFE_upper_Centerline,1));
%size(x_PTFE_upper_Centerline,1); size(x_PTFE_upper_Centerline,1);%
  while isempty(x_PTFE_intersect2) == 0 && isempty(x_PTFE_intersect_old3) == 0

    clear x_PTFE_intersect_old4 y_PTFE_intersect_old4
    x_PTFE_intersect_old4 = x_PTFE_intersect_old3;
    y_PTFE_intersect_old4 = y_PTFE_intersect_old3;

    clear x_PTFE_intersect_old3 y_PTFE_intersect_old3
    x_PTFE_intersect_old3 = x_PTFE_intersect2;
    y_PTFE_intersect_old3 = y_PTFE_intersect2;

    if all_peaks(size(all_peaks,1)-1,2) - avg_initial_slope(Tube_number) < 0
%smoothed_centerline_slope2(all_peaks(size(all_peaks,1)-1,1)) > 0 %
      x_PTFE_curve2 = x_PTFE_upper_Centerline;
      y_PTFE_curve2 = y_PTFE_upper_Centerline;
      x_PTFE_line2 = [x_PTFE_lower_Centerline(i,1); 0];
      y_PTFE_line2 = [y_PTFE_lower_Centerline(i,1); b(i,1)];
    else
      x_PTFE_curve2 = x_PTFE_lower_Centerline;
      y_PTFE_curve2 = y_PTFE_lower_Centerline;
      x_PTFE_line2 = [x_PTFE_upper_Centerline(i,1); 0];
      y_PTFE_line2 = [y_PTFE_upper_Centerline(i,1); b(i,1)];
    end
    [x_PTFE_intersect2, y_PTFE_intersect2] =
polyxpoly(x_PTFE_curve2,y_PTFE_curve2,x_PTFE_line2,y_PTFE_line2);
    i = i-1;
  end

  i
  PTFE_transition_length(Tube_number) = PTFE_transition_length(Tube_number) +
((mean(x_PTFE_intersect_old4)-x_PTFE_line2(1))^2 + (mean(y_PTFE_intersect_old4)-
y_PTFE_line2(1))^2)^.5
  New_max_PTFE_slope(2,:) = [i,slope(i)];

%%
  % Nylon loop #3

  x_Nylon_intersect3 = [0;0];
  y_Nylon_intersect3 = [0;0];
  x_Nylon_intersect_old5 = [0;0];
  y_Nylon_intersect_old5 = [0;0];

  i = all_peaks(size(all_peaks,1)-3,1) + round(.05*size(x_Nylon_upper_Centerline,1));
%size(x_Nylon_upper_Centerline,1);
  while isempty(x_Nylon_intersect3) == 0 && isempty(x_Nylon_intersect_old) == 0

    clear x_Nylon_intersect_old6 y_Nylon_intersect_old6
    x_Nylon_intersect_old6 = x_Nylon_intersect_old5;
    y_Nylon_intersect_old6 = y_Nylon_intersect_old5;

    clear x_Nylon_intersect_old5 y_Nylon_intersect_old5
    x_Nylon_intersect_old5 = x_Nylon_intersect3;
    y_Nylon_intersect_old5 = y_Nylon_intersect3;

    if all_peaks(size(all_peaks,1)-1,2) - avg_initial_slope(Tube_number) > 0
%smoothed_centerline_slope2(all_peaks(size(all_peaks,1)-1,1)) < 0 %
      x_Nylon_curve3 = x_Nylon_upper_Centerline;
      y_Nylon_curve3 = y_Nylon_upper_Centerline;
      x_Nylon_line3 = [x_Nylon_lower_Centerline(i,1); 0];
      y_Nylon_line3 = [y_Nylon_lower_Centerline(i,1); b(i,1)];
    else
      x_Nylon_curve3 = x_Nylon_lower_Centerline;
      y_Nylon_curve3 = y_Nylon_lower_Centerline;
      x_Nylon_line3 = [x_Nylon_upper_Centerline(i,1); 0];
```

```
         y_Nylon_line3 = [y_Nylon_upper_Centerline(i,1); b(i,1)];
      end
      [x_Nylon_intersect3, y_Nylon_intersect3] =
polyxpoly(x_Nylon_curve3,y_Nylon_curve3,x_Nylon_line3,y_Nylon_line3);
      i = i-1;
   end

   i
   Nylon_transition_length(Tube_number) = Nylon_transition_length(Tube_number) +
((mean(x_Nylon_intersect_old6)-x_Nylon_line3(1))^2 + (mean(y_Nylon_intersect_old6)-
y_Nylon_line3(1))^2)^.5
   New_max_Nylon_slope(1,:) = [i,slope(i)];

%%
   % PTFE loop #3

   x_PTFE_intersect3 = [0;0];
   y_PTFE_intersect3 = [0;0];
   x_PTFE_intersect_old5 = [0;0];
   y_PTFE_intersect_old5 = [0;0];

   % i =all_peaks(size(all_peaks,1)-3,1) + round(.1*size(x_PTFE_upper_Centerline,1));
%size(x_PTFE_upper_Centerline,1);
   while isempty(x_PTFE_intersect3) == 0 && isempty(x_PTFE_intersect_old) == 0

      clear x_PTFE_intersect_old6 y_PTFE_intersect_old6
      x_PTFE_intersect_old6 = x_PTFE_intersect_old5;
      y_PTFE_intersect_old6 = y_PTFE_intersect_old5;

      clear x_PTFE_intersect_old5 y_PTFE_intersect_old5
      x_PTFE_intersect_old5 = x_PTFE_intersect3;
      y_PTFE_intersect_old5 = y_PTFE_intersect3;

      if all_peaks(size(all_peaks,1)-1,2) - avg_initial_slope(Tube_number) > 0
%smoothed_centerline_slope2(all_peaks(size(all_peaks,1)-1,1)) < 0 %
         x_PTFE_curve3 = x_PTFE_upper_Centerline;
         y_PTFE_curve3 = y_PTFE_upper_Centerline;
         x_PTFE_line3 = [x_PTFE_lower_Centerline(i,1); 0];
         y_PTFE_line3 = [y_PTFE_lower_Centerline(i,1); b(i,1)];
      else
         x_PTFE_curve3 = x_PTFE_lower_Centerline;
         y_PTFE_curve3 = y_PTFE_lower_Centerline;
         x_PTFE_line3 = [x_PTFE_upper_Centerline(i,1); 0];
         y_PTFE_line3 = [y_PTFE_upper_Centerline(i,1); b(i,1)];
      end
      [x_PTFE_intersect3, y_PTFE_intersect3] =
polyxpoly(x_PTFE_curve3,y_PTFE_curve3,x_PTFE_line3,y_PTFE_line3);
      i = i-1;
   end

   i
   PTFE_transition_length(Tube_number) = PTFE_transition_length(Tube_number) +
((mean(x_PTFE_intersect_old6)-x_PTFE_line3(1))^2 + (mean(y_PTFE_intersect_old6)-
y_PTFE_line3(1))^2)^.5
   New_max_PTFE_slope(1,:) = [i,slope(i)];
%%
```

```
  Min_PTFE_Sweep_slopes = sortrows(cat(1,New_max_PTFE_slope,[avg_initial_point
avg_initial_slope(Tube_number)],[avg_final_point avg_final_slope]));
  Min_Nylon_Sweep_slopes = sortrows(cat(1,New_max_PTFE_slope,[avg_initial_point
avg_initial_slope(Tube_number)],[avg_final_point avg_final_slope]));


  Sweep_Angle_Min_PTFE(Tube_number) = 0;
  Sweep_Angle_Min_Nylon(Tube_number) = 0;

  %Calculate the angle in radians between the extreme slopes  %% angle = ATAN((B20-
B19)/(1+B19*B20))
  for n = 1:size(Min_PTFE_Sweep_slopes,1)-1
     if 0.1*half_size(1) < abs(all_peaks(n,1)) < 0.995*half_size(1)
        Sweep_Angle_Min_PTFE(Tube_number) = Sweep_Angle_Min_PTFE(Tube_number) +
abs(atan((Min_PTFE_Sweep_slopes(n+1,2)-
Min_PTFE_Sweep_slopes(n,2))/(1+Min_PTFE_Sweep_slopes(n+1,2)*Min_PTFE_Sweep_slopes(n,2))));
        Sweep_Angle_Min_Nylon(Tube_number) = Sweep_Angle_Min_Nylon(Tube_number) +
abs(atan((Min_Nylon_Sweep_slopes(n+1,2)-
Min_Nylon_Sweep_slopes(n,2))/(1+Min_Nylon_Sweep_slopes(n+1,2)*Min_Nylon_Sweep_slopes(n,2))));
     end
  end

%%

  figure
  hold on
  subplot(3,1,1);
  plot(smoothed_x_half_rev_boundary3, smoothed_y_half_rev_boundary3, 'b', 'LineWidth', 2)
  subplot(3,1,2);
%   figure
  plot(x_tube_Centerline(1:size(smoothed_centerline_slope),1), smoothed_centerline_slope, 'b',
'LineWidth', 2)
%   plot(smoothed_x_half_rev_boundary3(1:size(smoothed_centerline_slope),1),
smoothed_centerline_slope, 'b', 'LineWidth', 2)
  subplot(3,1,3);
%   figure
  plot(x_tube_Centerline(1:size(smoothed_centerline_slope2),1), smoothed_centerline_slope2, 'b',
'LineWidth', 2)
%   plot(smoothed_x_half_rev_boundary3(1:size(smoothed_centerline_slope2),1),
smoothed_centerline_slope2, 'b', 'LineWidth', 2)
  % axis([0 2500 -.001 .001])
  hold off


%%
%   figure
%   hold on
%   plot(x_PTFE_upper_Centerline, y_PTFE_upper_Centerline, 'r', x_PTFE_lower_Centerline,
y_PTFE_lower_Centerline, 'r')
%   plot(x_PTFE_intersect_old6, y_PTFE_intersect_old6, 'b', x_PTFE_line3, y_PTFE_line3, 'g')
%   plot(x_PTFE_intersect_old4, y_PTFE_intersect_old4, 'b', x_PTFE_line2, y_PTFE_line2, 'g')
%   plot(x_PTFE_intersect_old2, y_PTFE_intersect_old2, 'b', x_PTFE_line, y_PTFE_line, 'g')
%   % plot(x_Nylon_upper_Centerline, y_Nylon_upper_Centerline, 'g', x_Nylon_lower_Centerline,
y_Nylon_lower_Centerline, 'g')
%
```

```
%   figure
%       hold on
%   plot(x_Nylon_upper_Centerline, y_Nylon_upper_Centerline, 'r', x_Nylon_lower_Centerline,
y_Nylon_lower_Centerline, 'r')
%   plot(x_Nylon_intersect_old6, y_Nylon_intersect_old6, 'b', x_Nylon_line3, y_Nylon_line3, 'g')
%   plot(x_Nylon_intersect_old4, y_Nylon_intersect_old4, 'b', x_Nylon_line2, y_Nylon_line2, 'g')
%   plot(x_Nylon_intersect_old2, y_Nylon_intersect_old2, 'b', x_Nylon_line, y_Nylon_line, 'g')

%%
  %Convert values to inches from dpi
  Long_Straight_Length(Tube_number) = Long_S_Length/dpi;
  Total_Length(Tube_number) = Total_L/dpi;
  Short_Straight_Length(Tube_number) = Short_S_Length/dpi;
  Curved_length(Tube_number) = (Total_L - Long_S_Length - Short_S_Length)/dpi;
  Bend_Radius(Tube_number) = Curved_length(Tube_number)/total_angle(Tube_number);

  New_Long_Straight_Length(Tube_number) = New_Long_S_Length/dpi;
  New_Total_Length(Tube_number) = New_Total_L/dpi;
  New_Short_Straight_Length(Tube_number) = New_Short_S_Length/dpi;
  New_Curved_length(Tube_number) = (New_Total_L - New_Long_S_Length -
New_Short_S_Length)/dpi;
  New_Bend_Radius(Tube_number) = New_Curved_length(Tube_number)/total_angle(Tube_number);

end
%%
%Convert to metric units (meters)
Long_Straight_Length_metric = Long_Straight_Length.*0.0254; %converts inches to meters
Short_Straight_Length_metric = Short_Straight_Length.*0.0254; %converts inches to meters
Curved_length_metric = Curved_length.*0.0254; %converts inches to meters
Total_Length_metric = Total_Length.*0.0254; %converts inches to meters
Bend_Radius_metric = Bend_Radius.*0.0254; %converts inches to meters
Sweep_Angle_Max2 = total_angle;
Avg_Inclination = zeros(size(Sweep_Angle_Max2,2),1)';
PTFE_transition_length_metric = PTFE_transition_length.*0.0254; %converts inches to meters
Nylon_transition_length_metric = Nylon_transition_length.*0.0254; %converts inches to meters

New_Long_Straight_Length_metric = New_Long_Straight_Length.*0.0254; %converts inches to meters
New_Short_Straight_Length_metric = New_Short_Straight_Length.*0.0254; %converts inches to meters
New_Curved_length_metric = New_Curved_length.*0.0254; %converts inches to meters
New_Total_Length_metric = New_Total_Length.*0.0254; %converts inches to meters
New_Bend_Radius_metric = New_Bend_Radius.*0.0254; %converts inches to meters
Sweep_Angle_Max2 = total_angle;
% Avg_Inclination = zeros(size(Sweep_Angle_Max2,2),1)';
% PTFE_transition_length_metric = PTFE_transition_length.*0.0254; %converts inches to meters
% Nylon_transition_length_metric = Nylon_transition_length.*0.0254; %converts inches to meters


All_transition_lengths_metric = cat(2,PTFE_transition_length_metric',Nylon_transition_length_metric');


All_numbers =
cat(1,Bend_Radius_metric,Sweep_Angle_Max2,Sweep_Angle_Min_Nylon,Sweep_Angle_Min_PTFE,Lon
g_Straight_Length_metric,Short_Straight_Length_metric,Curved_length_metric,Avg_Inclination)

Tube_specs = fopen('Scanning_test_tube_specs.txt','w+');
```

```
fprintf(Tube_specs,'Bend_Radius\tSweep_Angle_Max2\tSweep_Angle_Min_Cath\tSweep_Angle_Min_PTFE\tLong_Straight_Length\tShort_Straight_Length\tCurved_length\tAvg_Inclination\n');
fprintf(Tube_specs,'%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\n',All_numbers);
fclose(Tube_specs);

All_numbers2 =
cat(1,Bend_Radius_metric,Sweep_Angle_Max2,Sweep_Angle_Min_Nylon,Sweep_Angle_Min_PTFE,Long_Straight_Length_metric,Short_Straight_Length_metric,Curved_length_metric,Total_Length_metric,Avg_Inclination,PTFE_transition_length_metric,Nylon_transition_length_metric,avg_initial_slope)

All_Tube_specs = fopen('All_Scanning_test_tube_specs.txt','w+');
fprintf(All_Tube_specs,'Bend_Radius\tSweep_Angle_Max2\tSweep_Angle_Min_Cath\tSweep_Angle_Min_PTFE\tLong_Straight_Length\tShort_Straight_Length\tCurved_length\tTotal_Length\tAvg_Inclination\tPTFE_Transition_Length\tNylon_Transition_Length\tAvg_Initial_slope\n');
fprintf(All_Tube_specs,'%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\n',All_numbers2);
fclose(All_Tube_specs);

New_All_numbers2 =
cat(1,Bend_Radius_metric,Sweep_Angle_Max2,Sweep_Angle_Min_Nylon,Sweep_Angle_Min_PTFE,Long_Straight_Length_metric,Short_Straight_Length_metric,Curved_length_metric,Total_Length_metric,Avg_Inclination,PTFE_transition_length_metric,Nylon_transition_length_metric,avg_initial_slope)

New_All_Tube_specs = fopen('New_All_Scanning_test_tube_specs.txt','w+');
fprintf(New_All_Tube_specs,'New_Bend_Radius\tSweep_Angle_Max2\tSweep_Angle_Min_Cath\tSweep_Angle_Min_PTFE\tNew_Long_Straight_Length\tNew_Short_Straight_Length\tNew_Curved_length\tNew_Total_Length\tAvg_Inclination\tPTFE_Transition_Length\tNylon_Transition_Length\tAvg_Initial_slope\n');
fprintf(New_All_Tube_specs,'%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\t%6.4f\n',All_numbers2);
fclose(New_All_Tube_specs);

Transition_lengths = fopen('Scanning_test_transition_lengths.txt','w+');
save('Scanning_test_transition_lengths.txt','All_transition_lengths_metric','-ascii', '-tabs');
fclose(Transition_lengths);
```

Appendix III- Image Processing Steps

Image preprocessing user steps

1. Place tube on scanner with long straight section horizontal
2. Scan tube at desired resolution without sharpening
3. Using Photoshop or other program, remove any dust or hairs from image
4. Clean up the reflection on the metal so it is white at the ends so the boundary edges can be easily detected (if it isn't, it will cause problems with the processing)
5. Crop image to desired size
6. Flip image if necessary so the long straight part is to the left (like the tube shown in Figure 5-1)
7. Save file as file type .png
8. Re-title cleaned file for use in Matlab

Image processing user steps

1. Change image file names to be processed in Matlab .m file
2. Change the name of the output file (at the end of the analysis file) that you want the results saved to
3. Make sure that the same tube and filament sizes are being used and change them as necessary
4. Run analysis file

Image processing analysis steps

1. Extract the edge of the tube image
2. Smooth edge points
3. Create parallel offset lines from the edge of the tube to represent the centerline path of the filaments
4. Calculate the lengths of the straight and curved sections of the tube
5. Find the points of tangency to get the transition lengths
6. Convert from dpi to inches
7. Convert from inches to meters
8. Save results to file

Steps to take image processing results to analyze and predict compression test data

1. Make sure that the tube shape parameters are all in the respective files
2. Change the importfile file names to make sure the correct tube shapes are being used for the predictions
3. Change the measured data file names
4. Check that they correct string type is being used for analysis
5. Change the file names of the information that you are saving (at end of .m file)
6. Rename the .m file to represent the tube being tested
7. Run file

References

[1] Aders, A., and Aders, H., 2005, "Anaesthetic adverse incident reports: An Australian study of 1,231 outcomes," Anaesth. Intensive Care, **33**(3), pp. 336-44.

[2] Werkmeister, J., and Slocum, A., 2007, "Theoretical and experimental determination of capstan drive stiffness," Precis. Eng., **31**(1), pp. 55-67.

[3] Lubinski, A., and Althouse, W., and Logan, J., 1962, "Helical buckling of tubing sealed in packers," JPT, **14**(6), pp. 655-67.

[4] Mitchell, RF., and SPE, and Enertech Engineering & Research, 1986, "Simple frictional analysis of helical buckling of tubing," SPE Drilling Engineering, **1**(6), pp. 457-465.

[5] McMann, R.C., and Suryanarayana, P.V.R., and Mobil E&P Technical Center, 1994, "Experimental study of curvature and frictional effects on buckling," OTC7568.

[6] He, XJ., and Kyllingstad, A., 1995, "Helical Buckling and Lock-up Conditions for Coiled Tubing in Curved Wells," Spe Drilling & Completion, **10**(1), pp. 10-15.

[7] Qiu,W.Y., and SPE, and Baker Oil Tools, 1998, "Prediction of Contact Force for Drill Pipe/Coiled Tubing," SPE eastern regional conference and exhibition, **14**, pp. 329-342.

[8] Kuru, E., and Martinez, A., Miska, S., and Qiu, WY., 2000, "The buckling behavior of pipes and its influence on the axial force transfer in directional wells," Journal of Energy Resources Technology-Transactions of the ASME, **122**(3), pp. 129-135.