AWARD NUMBER: M1406600

TITLE:

ARL-HRED-STTC Tissue Characterization Research Initiative

PRINCIPAL INVESTIGATOR: Jack Norfleet

CONTRACTING ORGANIZATION: REGENTS OF THE UNIVERSITY OF MINNESOTA
OFFICE SPONSORED PROJECTS ADMINISTRATION
200 OAK ST SE # 224
MINNEAPOLIS MN 55455-2009

REPORT DATE: JULY 2020

TYPE OF REPORT: FINAL
Fort Detrick, Maryland 21702-5012

PREPARED FOR: U.S. Army Medical Research and Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JULY 2020	FINAL	1JAN2015 - 17MAR2020

4. TITLE AND SUBTITLE

ARL-HRED-STTC Tissue Characterization Research Initiative

5a. CONTRACT NUMBER
M1406600

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Jack Norfleet

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

REGENTS OF THE UNIVERSITY OF MINNESOTA
OFFICE SPONSORED PROJECTS ADMINISTRATION
200 OAK ST SE # 224
MINNEAPOLIS MN 55455-2009

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

There is a great deal of interest in simulation as an approach for training providers to develop and maintain skills and to theoretically model interactions of human tissue for military applications. Accurate theoretical models require accurate mechanical properties of the simulated tissue. The goal of this collaboration is to develop measurement capabilities to characterize human, animal and synthetic tissues for physical parameters. These parameters are the key element in developing both virtual and synthetic models to simulate interactions of tissue for both military injury prediction and medical procedure training. Initial efforts will result in a collaborative approach to characterization and sharing results. Measurements initially focus on bio-mechanical characterization and are expanding to electrical, optical and thermal effects. The Cooperative Agreement with ARL is developing complimentary measurement capabilities to characterize human, animal and synthetic models to so that we can develop realistic modeling and training systems.

15. SUBJECT TERMS
NONE LISTED

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			USAMRMC
Unclassified	Unclassified	Unclassified	Unclassified	95	19b. TELEPHONE NUMBER *(include area code)*

111 Church ST SE, 1100ME
Minneapolis, MN 55455

612-626-6751
timk@umn.edu

University of Minnesota
Mechanical Engineering

CREST
Center for Research in Education and Simulation Technologies
UNIVERSITY OF MINNESOTA
Driven to Discover℠

ARL-HRED-STTC Tissue Characterization Research Initiative

PROGRESS REPORT
JULY 2020

Cooperative Agreement for Tissue Characterization

M1406600
Contract Effective Date: 05/01/2014

Report Date: 7/28/2019

New Principal Investigator: Timothy Kowalewski, PhD
University of Minnesota, Mechanical Engineering

Former Principal Investigator: Robert Sweet, MD
University of Minnesota Medical School

Cooperative Agreement Manager: Jack Norfleet (ARL)

in vivo-ex vivo Tissue Property Decay (in vivo)

Summary

Since the previous update, all tissue testing has been completed, the device has been shipped to the CREST lab at the University of Washington for potential future testing, and data analysis is near complete with one publication complete and at least one other publication currently being worked on.

Porcine tissue testing was conducted across five sperate animals at five different tissue states including: in-vivo, in-situ immediately post-mortem, same day ex-situ, post-refrigeration, and post-freeze-thaw. For each animal, the liver, spleen, peritoneum, and lung were tested, while inferior vena cava, descending aorta, and esophagus were tested on select animals when available. All final porcine tissue testing was completed between June 15th, 2019 (7/15/219) and October 28th, 2019 (10/28/2019).

The months following were primarily spent reviewing data from the porcine tissue testing in the form of data handling/processing and data analysis. Data handling and processing included segmentation of the raw database into usable grasps for analysis as well as filtering out any erroneous data. This was followed by validation of the grasper via analysis of puck calibration grasps which required additional processing in order to properly present the data. From this a data processing procedure was determined and this procedure was applied to the entire database. From here the data is now being analyzed across all pigs and states for liver, spleen, and peritoneum tissues, which will be followed by statistical analysis for publication purposes.

As of the release of this final update, one publication has been published through the Design of Medical Devices conference and an additional publication is currently in progress. The publication through the Design of Medical Devices conference focused on the design of the grasper device itself and has a video presentation associated with the publication posted via the Design of Medical Devices as part of the rapid-fire presentation sessions. The publication currently in progress focuses on tissue mechanical response of porcine liver, spleen, and peritoneum tissue across all five pigs at all five tissue states tested. Pending results from statistical analysis, additional publication(s) are planned.

Risk/Concern – N/A

Procedure Testing

Preparation

Primary preparation for in-vivo porcine tissue testing consisted of ensuring the grasper remained in a working state, capable of moving between testing locations and recording the necessary applied force and encoder jaw angle, and training of new staff in their designated roles to ensure proper testing protocol was followed. To do this, dry runs of tissue data collection was carried out multiple times from porcine tissue provided by the University of Minnesota's Visible Heart Lab (VHL). Whenever a sample tissue was provided, each person was assigned one of the four roles designated in the testing protocol, see Appendix A, and the testing procedure outlined in the testing protocol was carried out as outlined.

During this time, minor quality of life changes were made to the grasper device in order to more streamline the testing process. The first of these changes included extending the connecting wires

from the grasper device to the NI-DAQ board on the cart in order to provide the individual grasping more freedom to move without the cart interfering. In order to ensure this wire extension did not cause additional noise in the data signals, testing was carried out comparing noise using the previous shorter wires to noise experienced with extended wires. It was determined that the change in overall noise was negligible and had no impact on overall data. The second quality of life change that was made to the grasper was swapping the DC DC converter boost Buck to a dedicated power supply for signal amplification. During the dry runs of porcine tissue testing it was discovered that the DC DC converters would occasionally overheat and fail temporarily forcing testing to pause while they cooled. The dedicated power supply had no overheating issues and could be used indefinitely without any noticeable effect and would be used throughout in-vivo porcine tissue testing.

Testing Summary

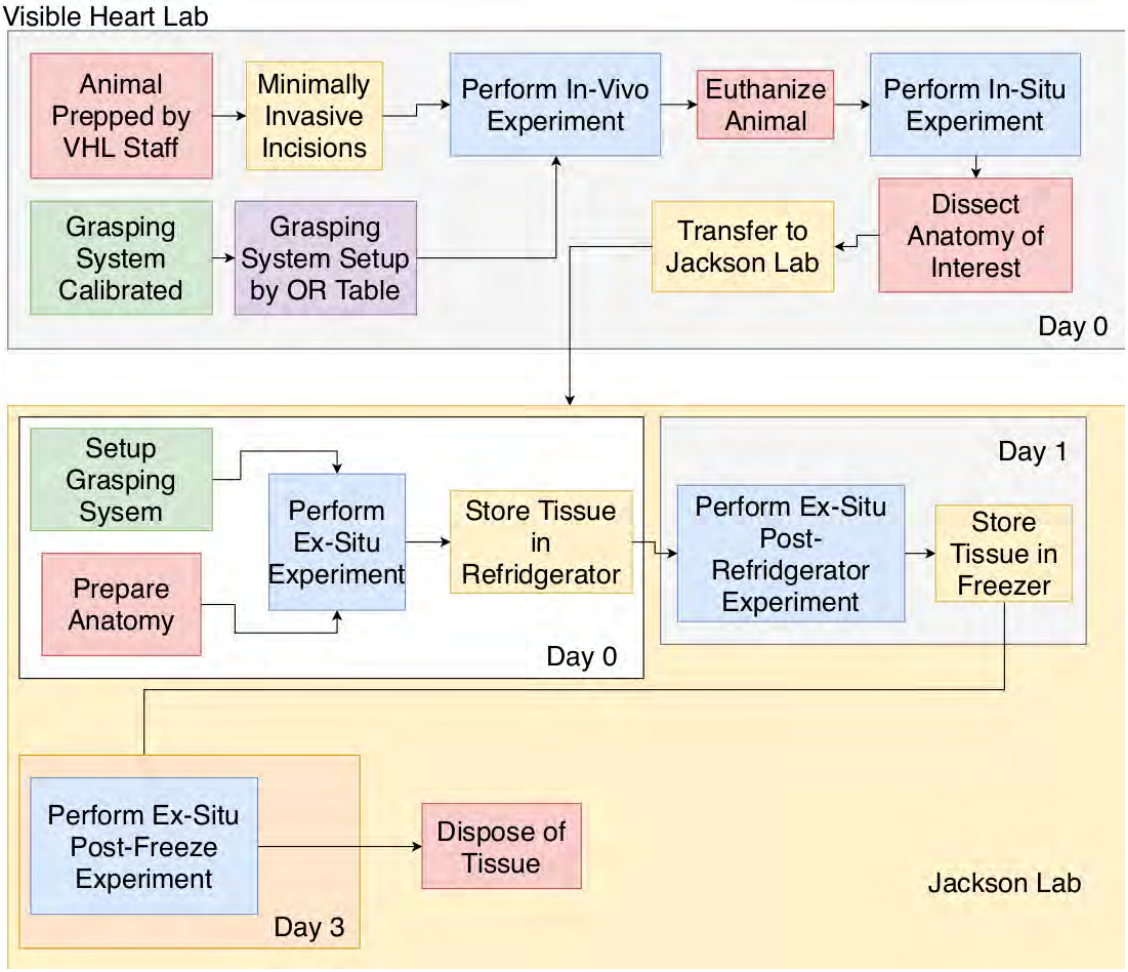


Figure 1: Tissue testing procedure for a single animal

In-Vivo and Postmortem

The animal was sedated and laid on its back. An incision is made starting in the abdominal cavity (to access the Liver, Spleen, and “Belly” tissues), and then later expanded to the chest cavity (to access Lung and Aorta). The body temperature of the animal is measured via a rectal thermometer and recorded.

After completing the data collection process on the desired tissues, the animal was euthanized, and the animal’s heart was extracted for separate testing occurring within the Visible Heart Lab. We continued with data collection immediately after euthanasia, on the same tissues as tested during in-vivo conditions, as outlined in Figure 1.

Ex-Vivo, Refrigeration, and Freezing

After Postmortem testing the tissue was extracted from the carcass and placed in a plastic bag with 25 mL of 1x PBS/pen-strep antibiotic solution to minimize tissue decay. The liver was placed in a bag with 75 mL of solution due to its larger size.

The plastic bags containing the tissues were placed in water baths at 37.5 degrees Celsius until data collection for the ex-vivo stage such that the tissue was approximately body temperature. After ex-vivo data collection, the tissues were replaced in their respective plastic bags (still containing the antibiotic solution) and then placed into a refrigerator maintaining approximately 4 degrees Celsius.

After approximately 24 hours of refrigeration, the tissues were removed from the refrigerator and re-heated in a water bath at 37.5 to 38 degrees C for 2-4 hours, or until they achieved roughly body temperature. Again, the tissues remained in their plastic bags with the antibiotic solution, until they were temporarily removed for data collection. Once data collection was completed, the tissues were placed in their respective plastic bags. The tissue may be placed in a new plastic bag with more antibiotic solution if the previous bag was saturated with fluids from the tissue.

The tissue was placed in a freezer at -18 degrees C for approximately 72-120 hours, after which the tissues was removed and placed in a water bath at 38 degrees C for 4-5 hours, or until the tissue achieved roughly normal body temperature. Data was collected according to protocol from the tissues for a final time, after which the tissues were disposed of.

Grasping Device Operation Setup

The grasping device was first placed into a medical glove such that the load cells were placed into the fingers of the glove: the top jaw went into the “index finger” and the lower jaw went into the “pinky finger”. The remaining fingers of the glove were taped against the device in order to prevent interference during testing.

The grasping device was connected to a National Instruments DAQ, which was then connected to a computer. This signal path is presented in a flow chart below in Figure 2. The computer ran the “Scissors Console” application, which samples the voltage and encoder data at 1000 Hz and stored the data in CSV files defined by the user. The application also displays a graph of the data collected in real-time and produces a series of tones at fixed time intervals in order to produce consistent grasp timings, which is shown in Figures 3 and 4 respectively.

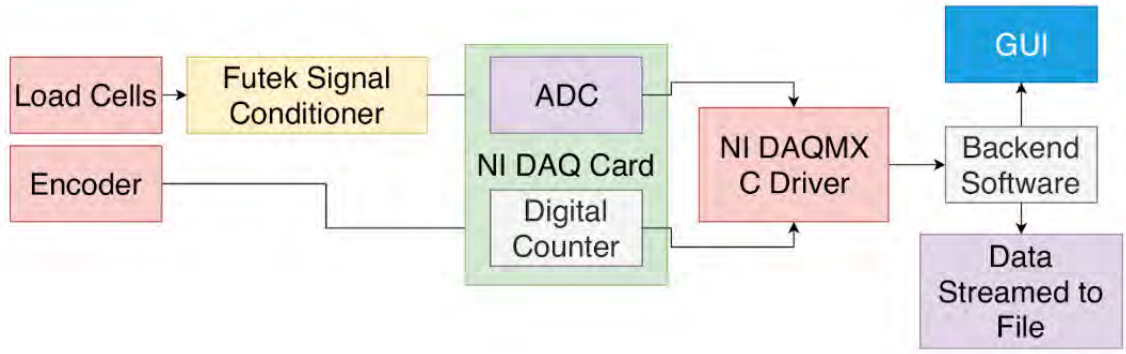


Figure 2: Data signal path from grasper device to stored data

Figure 3: Screenshot of the scissors console application. Plot displays data collected in real-time. See Appendix B for detailed process to acquire code for GUI

Calibration

First, the Futek Signal Conditioners were adjusted such that the voltage readings from the load cells were within +/- 0.1 V when the grasping device “closed” and held horizontally (the “top” load cell points towards the floor, while the “bottom” load cell points towards the ceiling without touching each other).

The grasping device was then placed on a horizontal surface such that a load cell was oriented perpendicular to the floor. The voltage data was recorded as a baseline, then dead weights with known masses (varying from 10 mg to 500 mg) were placed on the load cell, and the voltage data was recorded for each mass. The weighting process was repeated for the other load cell, producing voltage data for every weight for both load cells.

The full calibration process was repeated at the beginning and end of each day of data collection and both before and after changing testing locations.

The data was later used to convert from the recorded voltage data to force data. This was done by applying a linear regression to a plot of Force (converted from weight) to Voltage. The resulting fit coefficients were then applied to the recorded voltage data to obtain force data from voltage readings.

Data Collection

An individual designated as the “Tissue Holder” measured the tissue temperature, and adjusted the tissue into a suitable position for data collection; ideally holding the tissue horizontally with space for both jaws of the grasping device to freely move without touching any other objects. The tissue holder also indicated where the tissue was to be grasped - making sure to avoid the edges of the tissue and any surface irregularities.

Once the tissue was in position, the data collection process was initiated. The Scissors Console application produced a series of tones to dictate the timing of the data collection process. Initially, the console produced 3 beeps, each 1 second long, with 0.75 seconds between beeps as the grasping device was “stretched” open and closed in order to ensure that the index pulse was triggered on the encoder ensuring accurate and consistent encoder results.

Once the stretching period had elapsed, the console waits for 3 seconds and then produces a different set of tones.

First, the console produced a short beep for 200 milliseconds as the user moved into position on the grasp site. After a 500-millisecond delay, the console produced continuous tone for 1500 milliseconds, and the user grasped the tissue site throughout the duration of the tone. Once the beep stopped, the user released the tissue from the grasp and the console waited for 1000 milliseconds to repeat the cycle.

This cycle was repeated 5 to 10 times at a single grasp site, using the beeps to ensure consistent grasp duration and frequency. Once the desired number of grasps were completed at the site, the “Stop” button was pressed on the console application, and the grasp site was marked using o a meat marker. The Data Collection process was repeated at several grasp sites throughout the tissue, for each tissue. However, certain tissues such as the aorta and esophagus could only be effectively grasped at a single grasp site during in-vivo testing, due to physical limitations. The temperature of the tissue is taken again on the surface of the tissue, near the location of the trials. This entire procedure was conducted as outlined in testing protocol, see Appendix A.

Data Analysis

Data Handling and Preprocessing

Preprocessing

The raw data from the CSV files are first read by a MATLAB script and collected into one large data structure which organizes individual trials by animal, tissue, test stage (in-vivo, ex-vivo, postmortem, etc.), and finally by test site. During this process, the script discards all data from the initial “stretching” period and converts recorded voltage data into force.

Grasp Detection/Segmentation

Once the data has been organized into a MATLAB data structure, another script analyzes each trial and automatically detects individual grasps from a trial.

First, the script obtains a moving mean of the force using a running average of 100 samples. It then calculates the difference in force between each subsequent sample and applies a threshold limit to find the samples where force is increasing in both load cells. Thereafter, periods where the force is increasing are defined as grasps.

The grasps are further refined to combine “broken regions” where a single grasp has been detected as multiple smaller grasps - usually because the grasping device temporarily stops increasing the force applied to tissue.

The automatic grasp detection is then further refined by manually determining the start of the grasp. This is done by plotting the force and manually selecting the time stamp when both jaws have begun contact with tissue, determined by when the force on both load cells have increased greater than the baseline (typically <0.1 N). During this process, we discard any trials that had technical issues that may have previously been unnoticed. These issues usually stemmed from a loose wire connection with the encoder or either load cell.

The script then converts the encoder data into the angle between jaws, and then into the linear distance between the jaws. This is used to define the thickness of the tissue, which is the distance between the jaws on first contact. It then calculates the change in tissue thickness from the beginning of the grasp. From this data, we calculate the “pseudo-strain” of the tissue, by dividing the change in tissue thickness by the initial thickness, producing the normalized displacement.

Finally, the script also creates data fields for testing information. This information includes tissue temperature, location on the tissue (if available), testing type (whether it was a site that had previously been tested or not), and any miscellaneous testing notes. These data are entered manually from a checklist filled out during testing.

Data Analysis

Grasper Device Validation

Grasper device testing results validation was conducted by analyzing all of puck grasping datasets to one another. Ideally, there should be not change in stiffness for the synthetic puck, thus the resulting force-strain relationship at a given applied force should be identical across all puck grasping datasets. Upon plotting all of the first grasps on the synthetic puck across all testing scenarios, it was quickly evident that with the automatically segmented grasps, we were not seeing similar stiffness across all trials, as shown in Figure 4 below:

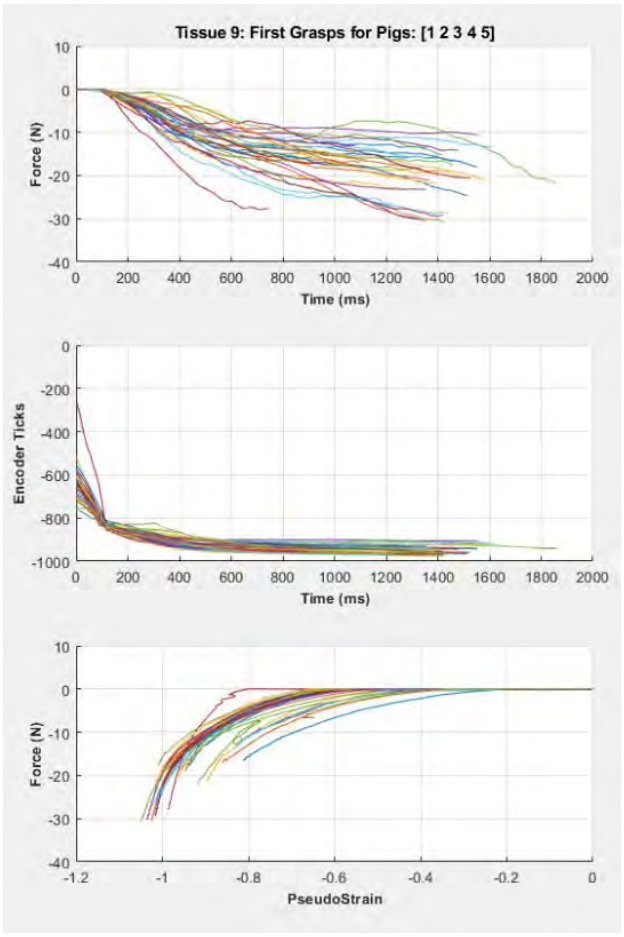


Figure 4: Synthetic puck first grasp raw data plots across all testing session. First plot is applied force with respect to time, second plot is encoder tick with respect to time, and third plot is force with respect to pseudo strain

Upon seeing this, there were two potential sources where this change in stiffness could be coming from. The first of these was from an improper voltage to force conversion, which could cause variation in the applied force dataset. To analyze this, each of the jaw load cell calibration datasets was loaded into a MATLAB function individually and analyzed. From this relationship of applied weight to voltage readings, it was possible to apply a line of best fit to the dataset, in which the slop would be the resulting voltage to force conversion. The results from this study are presented on the next page, in Figure 5. From this plot it is clear that the voltage to force conversion remains fairly consistent across all testing scenarios for each load cell independently with a resulting average conversion rate of 4.901N/V for the load cell and 5.076N/V for the bottom load cell. In addition to analyzing the conversion ratio directly, it was also necessary to consider that a linear best fit line was appropriate for the conversion. When analyzing the R-squared values across each load cell calibration dataset the minimum R-squared value was 0.996674 which reinforces the notion that the voltage to force conversion is a linear relationship, thus a linear line of best fit is an appropriate method for determining the conversion ratio.

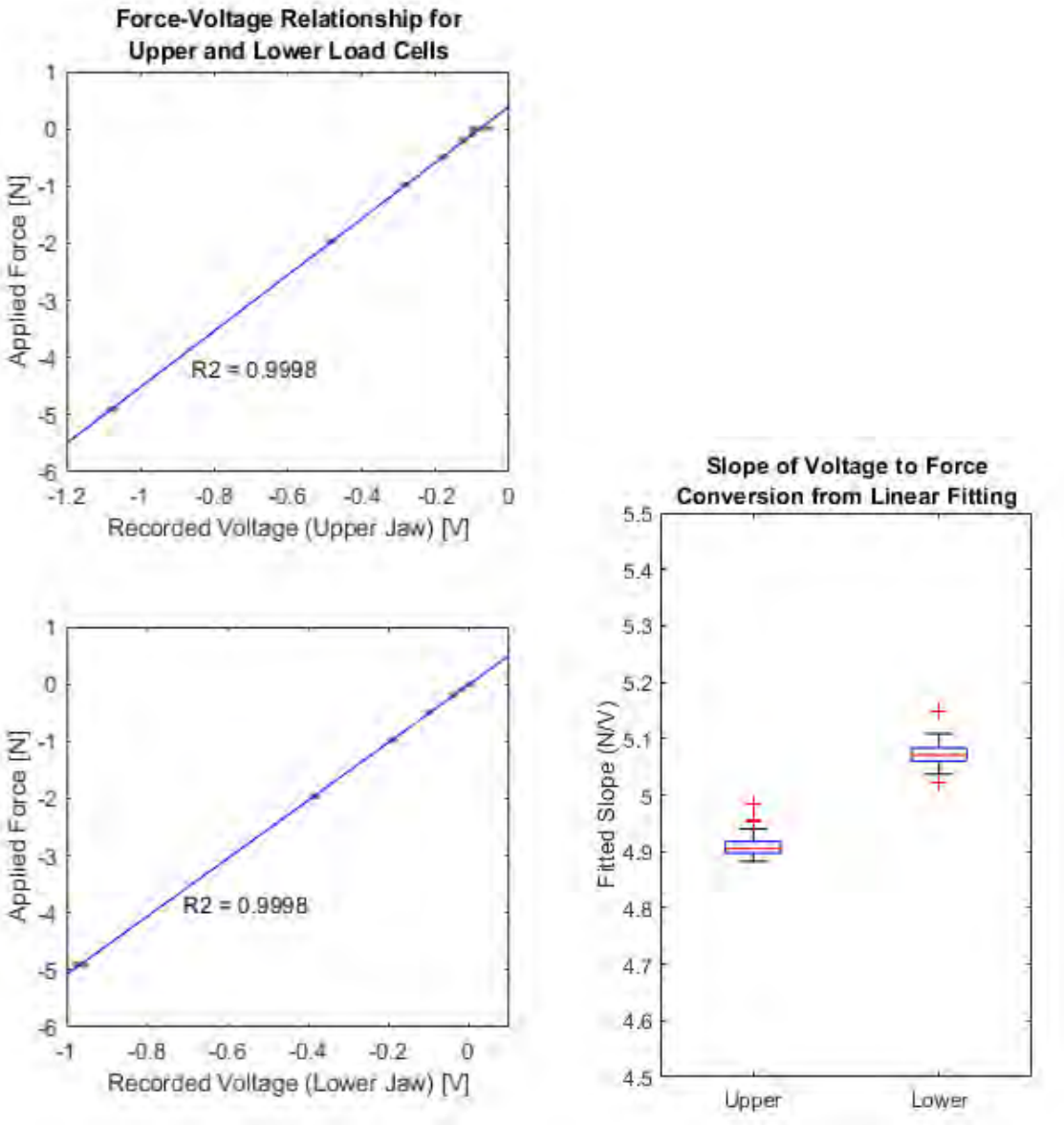


Figure 5: Voltage to force linear fit example plots (left) and overall fitted slope voltage to force conversation ratio for both jaw load cells (right)

This process of validating the voltage to force conversion ratio, while not directly solving the issue of varied results in puck calibration stiffness, did ensure that the proper conversion ratio is being used throughout the rest of the data analysis process. The second potential source of stiffness error was determined to be an issue with the automated grasp segmentation process. After analyzing individual grasp raw date, it was noted that the automated grasp segmentation process was setting the start points of grasps to early in the data set, which caused the linear region during the first 100ms of the force with respect to time plot of Figure 6. To combat, the team went in a manually set the starting timestamp for each first grasp to the puck and generated new force with respect to pseudo strain plots. This plot is presented below in Figure 7.

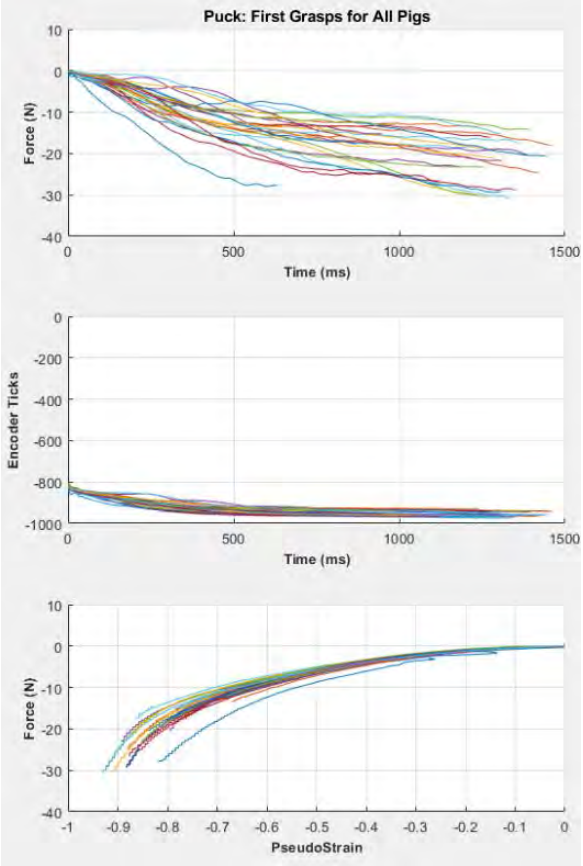


Figure 6: Synthetic puck first grasp manually segmented grasp start timestamps plots across all testing session. First plot is applied force with respect to time, second plot is encoder tick with respect to time, and third plot is force with respect to pseudo strain

From this figure it is clear that the slope of the force with respect to pseudo strain plot was much more consistent across trials, thus signifying more consistent stiffness readings. The next step in grasper device validation was to the then review the stiffness of the puck at various applied forces, which ideally will be the same for a given applied force across all sessions. These plots are presented below in Figures 7 and 8:

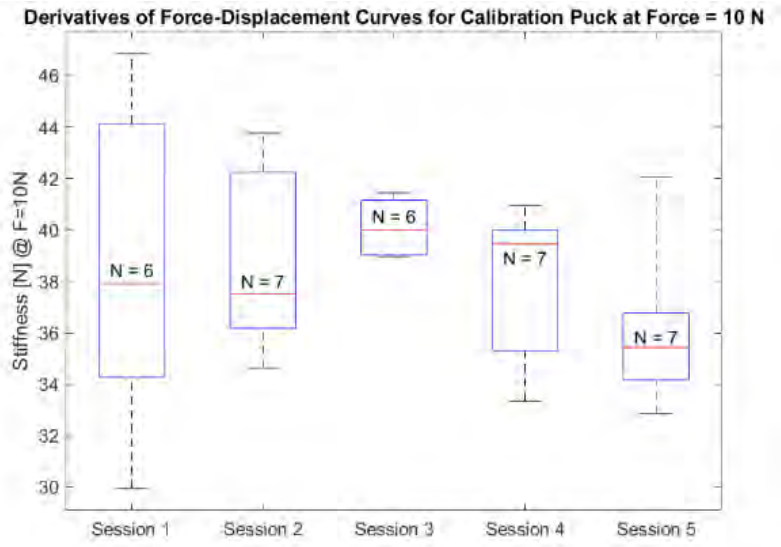


Figure 7: Synthetic puck stiffness at 10N applied force across all testing sessions

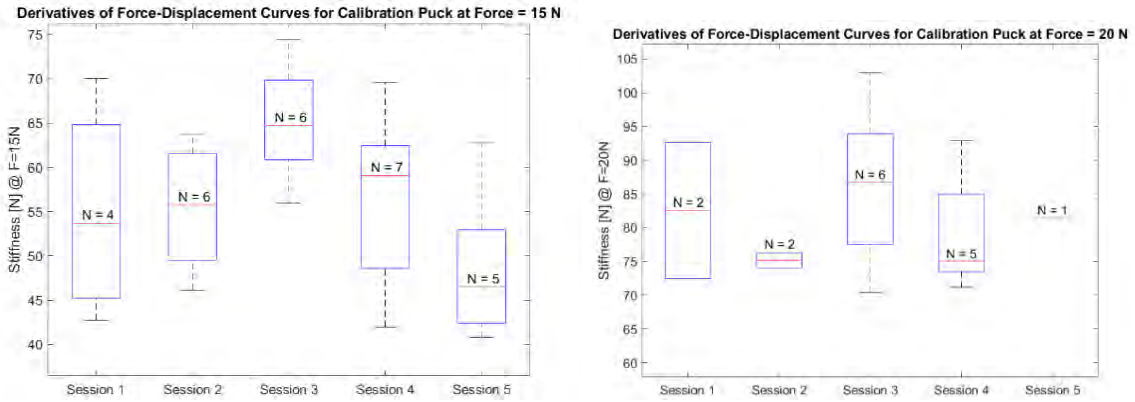


Figure 8: Synthetic puck stiffness at 15N (left) and 20N (right) applied forces across all testing sessions

From these figures, it is clear that the resulting stiffness is now much more consistent at a given applied force. With this said, the calibration puck data shows some variation in stiffness across sessions and within sessions, despite being the exact same material with the same measurement methods. Variation can be partially explained by grasping near the edges of the calibration since material stiffness can change dur to edge effects and grasping the puck at an angle. These sources of error, and general uncertainty of the grasper device readings due to noise, were now within an acceptable range to continue with data analysis on entire set of porcine tissue testing data.

Curve Fitting

The following is the equation used for curve-fitting:

$$F(t) = \alpha * \left(e^{\beta * \epsilon (t-1)^2}\right) \qquad (1)$$

Where (alpha) and (beta) are the curve-fit parameters to be determined; while F(t) and (t) are the force and (normalized) displacement of a grasp, respectively.

We utilize both the “trust-region-reflective” algorithm as well as the “levenberg-marquardt” algorithm with a step tolerance of 10^{-13}, as well as function tolerance and optimality tolerance of 10 (default MATLAB settings).

The initial guesses for curve-fitting were defined for each tissue as follows:

Tissue	α	β
Liver (Default)	-2	5
Spleen	-1	2.5
Aorta	-0.25	4
Lung	-0.03	6.5
Belly	-0.5	6.5
Calibration Puck	-20	1

Table 1: Initial guess coefficients for iterative curve-fitting

Curve-fitting is run multiple times for each grasp; each time, the residual for the function is calculated. If the squared sum of the error (SSE) is lower than the previous best sum, it is selected as the solution. The current best solution is then fed into the curve-fitting algorithm as a new initial guess. The script switches between the two solvers on each subsequent iteration, until both solvers are run 3 times each.

After obtaining the best solution, we calculate a “quality” metric for the solution similar to how an R-squared value is calculated for a linear fit. First, we calculate the sum of squared total:

$$SST = \sum_{t=1}^{n} (F(t) - \bar{F})^2 \qquad (2)$$

Where F(t) is the force at time t and $\bar{F}$ is the mean force. We can then calculate the quality metric as:

$$Quality = 1 - SSE \div SST \qquad (3)$$

The generated alpha and beta values are saved along with the generated quality metric and other data from each grasp.

Data Analysis

We calculate a “stiffness” metric for the tissue in each trial by computing the analytical derivative of the curve-fit Force-Displacement curve of each grasp. The derivative of the curve is given as follows:

$$stiffness = \frac{dF(t)}{d\epsilon(t)} = 2\alpha\beta\epsilon(t) * e^{\beta * \epsilon(t)^2} \qquad (4)$$

Thus, we calculate the stiffness from the curve-fit parameters and the displacement of the tissue at a point in the grasp where the force applied is equal to some desired force value. Consequently, any grasps that did not reach the desired force value cannot produce a stiffness metric with this method.

Ongoing Data Analysis

Liver

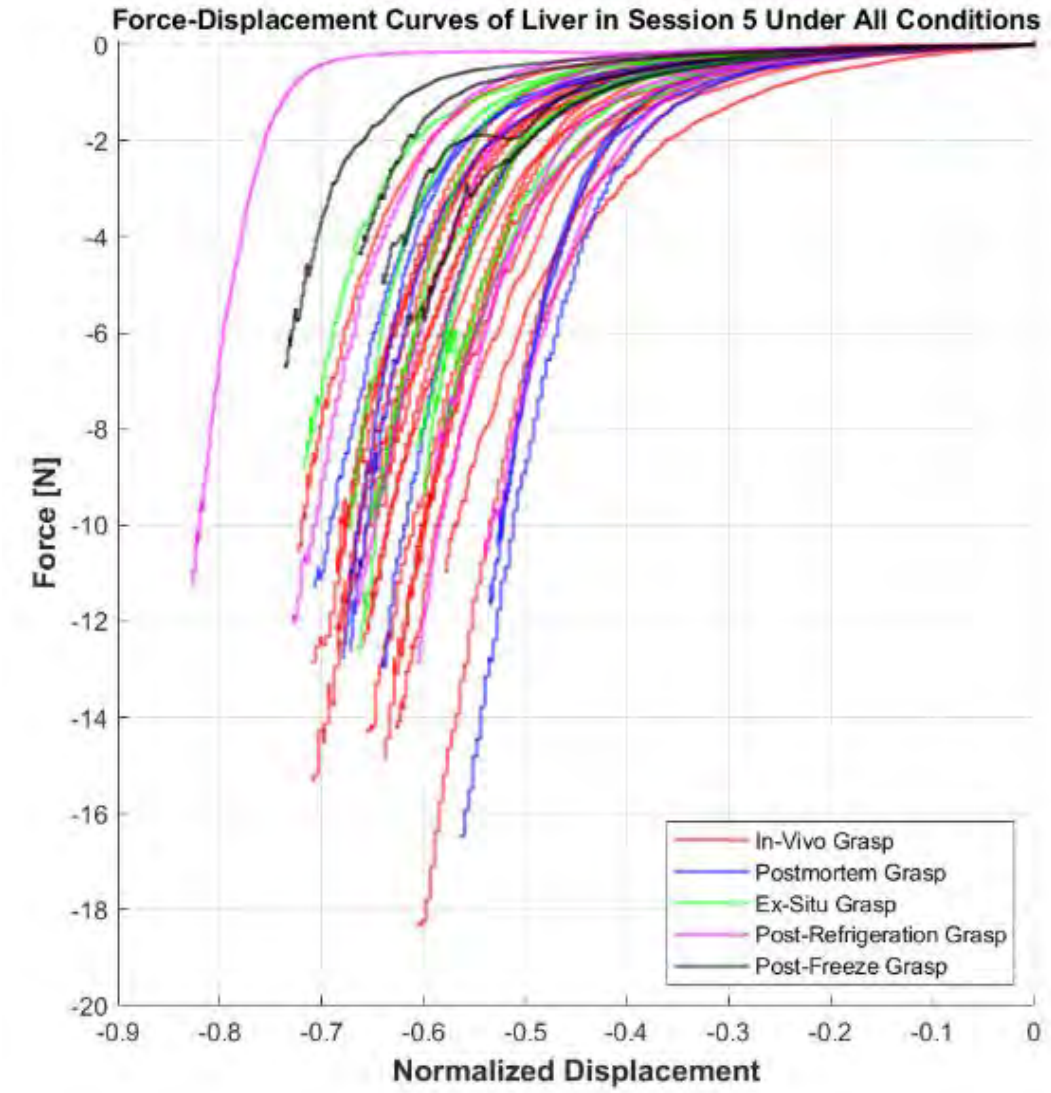


Figure 9: Porcine liver force-normalized displacement curve for all first grasps from session/animal 5

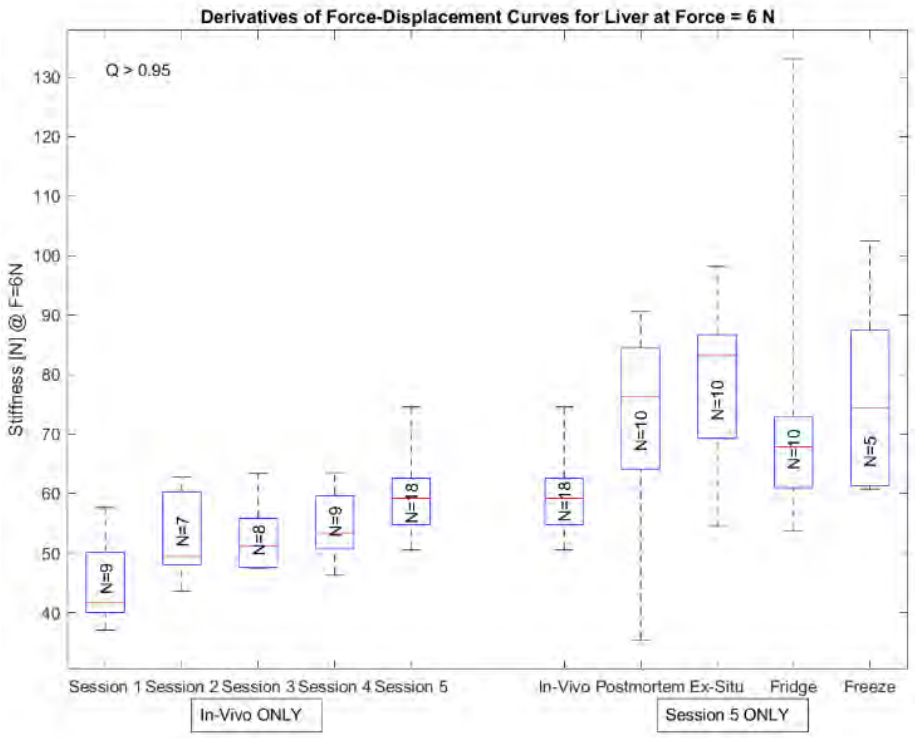


Figure 10: Porcine liver stiffness at an applied force of 10N across all sessions in-vivo (left) and across all states during session 5 (right)

Plot of Liver stiffnesses measured via the Curve-Fit-Derivative approach. The first 5 box plots compare stiffnesses across sessions (different animals) as measured during the In-Vivo experiments; the last 5 box plots show the measured stiffnesses across the different test conditions. The “N =” value indicates how many grasps are included in each set; and each grasp has a quality value of at least 0.95. All outliers are included in the whiskers: outlier detection is disabled.

We observe increased stiffness in the liver after euthanasia, until we refrigerate (and re-heat) the tissue. Unfortunately, the post-freeze data is more limited with only 5 samples; many of the samples were “lost” due to insufficient maximum force applied to the tissue.

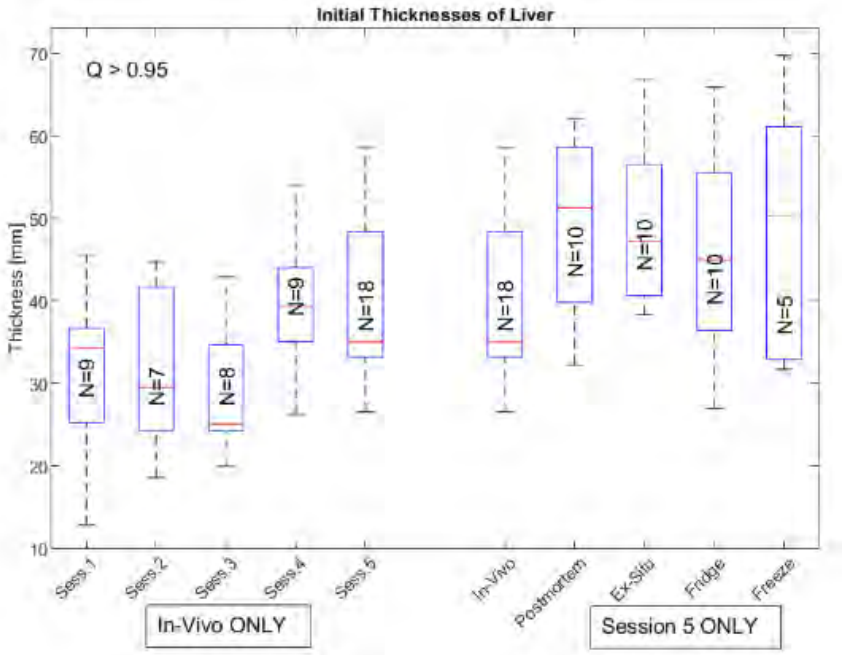


Figure 11: Porcine liver initial thickness across all sessions in-vivo (left) and across all states during session 5 (right)

All of these results portray primarily for a single animal across all five tissue states. Presented in the figure below are similarly formatted plots; however, for the different tissue states tissue from all sessions are included across each of the five animals.

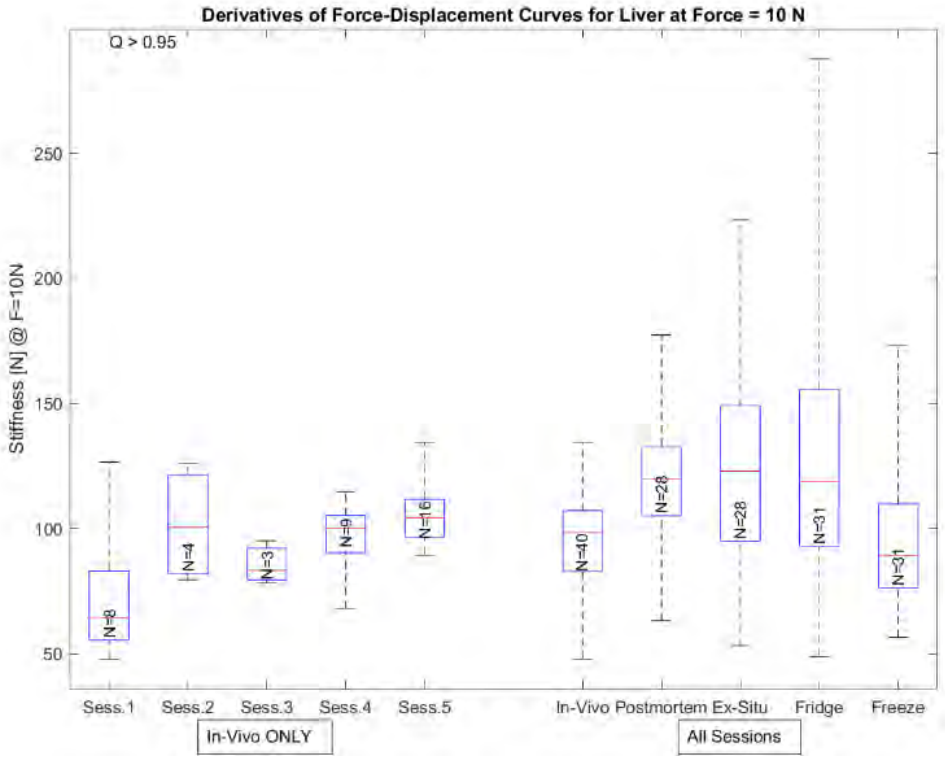


Figure 12: Porcine liver stiffness at an applied force of 10N across all sessions in-vivo (left) and across all states across all sessions (right)

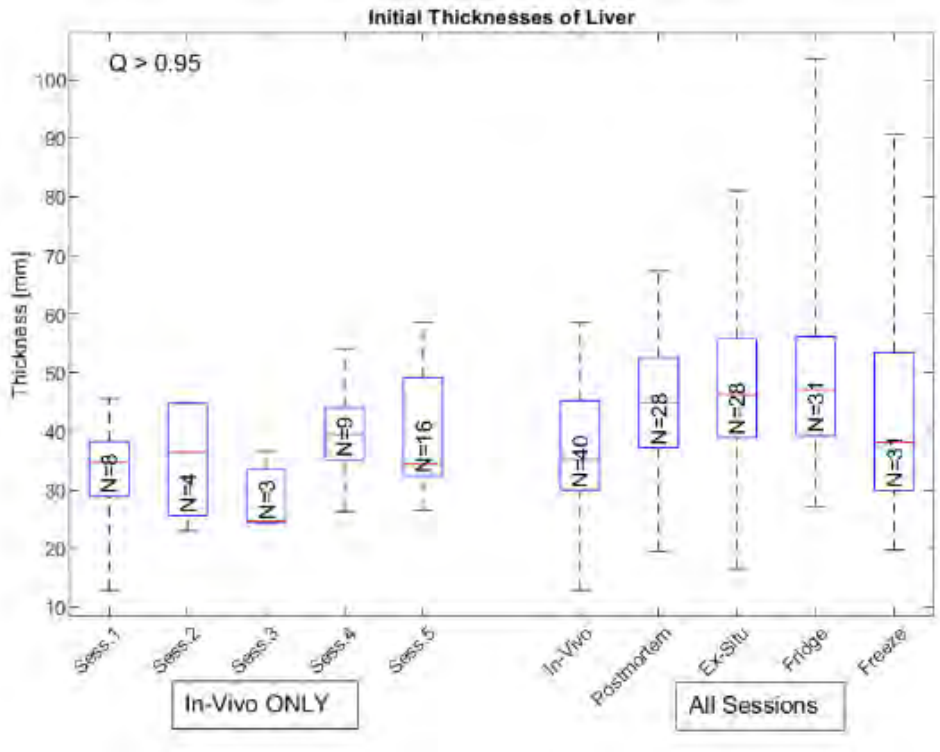


Figure 13: Porcine liver initial thickness across all sessions in-vivo (left) and across all states at across all sessions (right)

Overall liver data does not exhibit significant differences in stiffness from postmortem and onwards. Post-freeze stiffness returns to values closer to those observed during in-vivo. There is largely an increase in variation after in-vivo, which may be explained by variations in temperature.

Spleen

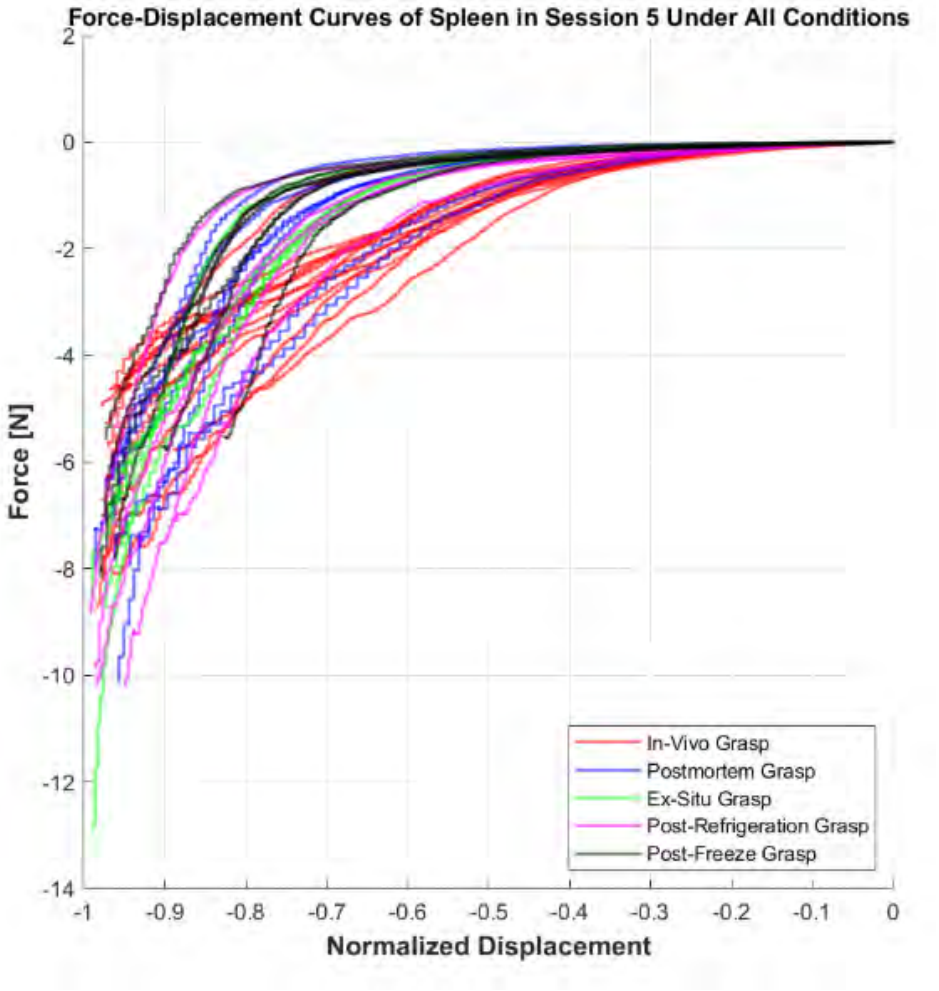


Figure 14: Porcine spleen force-normalized displacement curve for all first grasps from session/animal 5

From observations during our testing and preliminary data analysis, we believe the spleen is a special case that is dominated by fluid/viscous effects. This is due to certain behavior observed in the Force-Displacement curve of the spleen: if a constant (nonzero) force is applied to the spleen, the displacement increases despite no additional effort applied by the grasper. This effect can be so pronounced that a reduction in force applied can still produce increased displacement. An example of this is shown in the figure below:

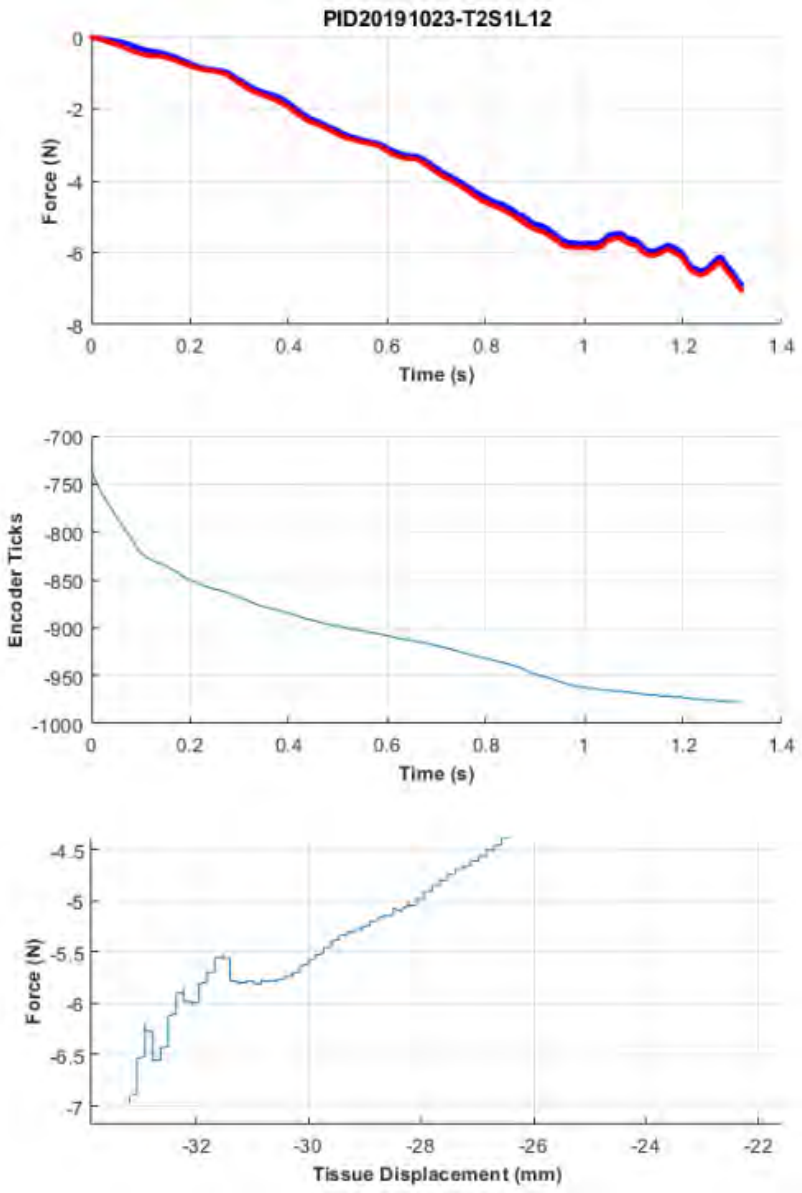


Figure 15: Porcine spleen example first grasp depicting change in tissue thickness at constant applied force resulting in poor force with respect to tissue displacement relationship

Figure 15 shows force in top plot (red line being the upper jaw, blue line being lower jaw); the encoder ticks in the middle plot, and the resulting force-displacement curve in the bottom plot. Around the 1 second mark, the applied force remains roughly constant at just under 6 Newtons, but the grasper continues to close, as evidenced by the encoder ticks decreasing. Shortly after, the applied force is reduced, but the grasper continues to close.

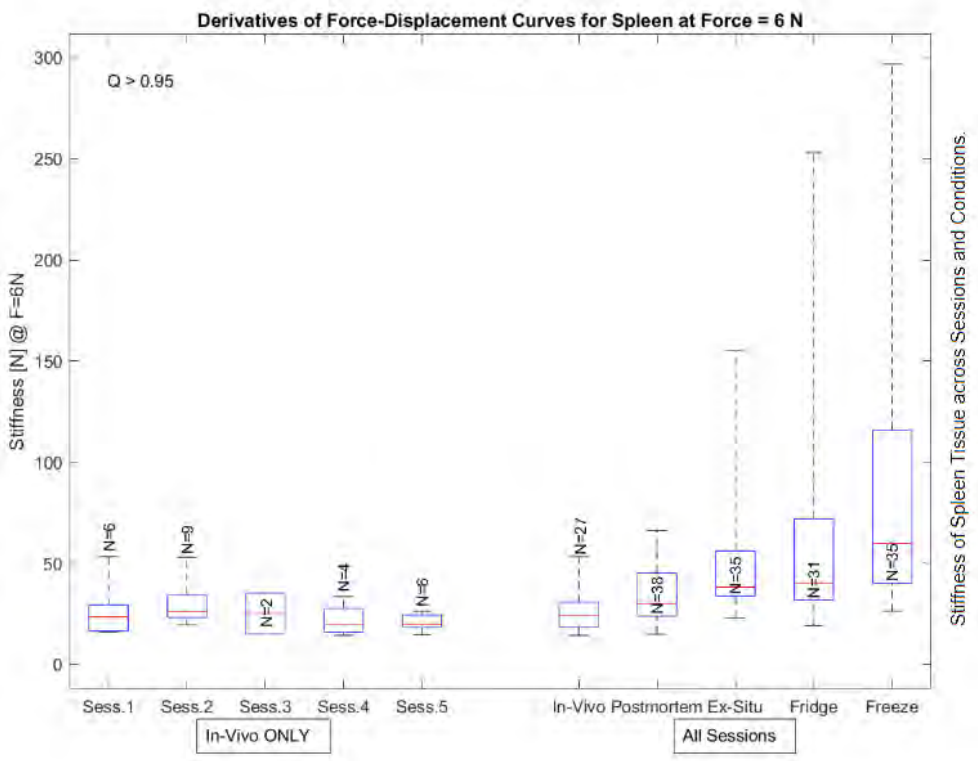


Figure 16: Porcine spleen stiffness at an applied force of 6N across all sessions in-vivo (left) and across all states across all sessions (right)

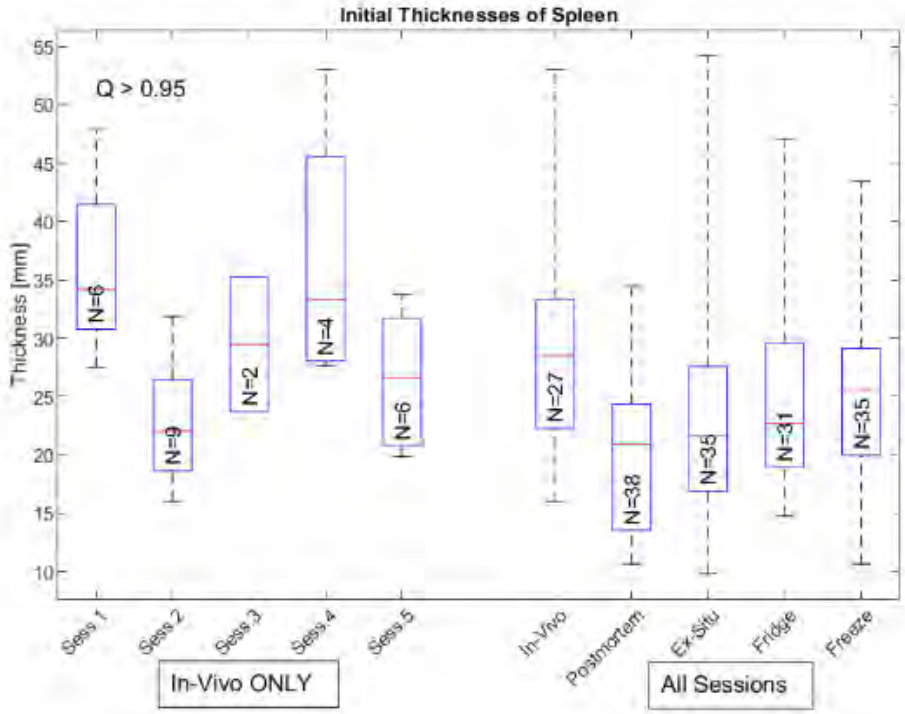


Figure 17: Porcine spleen initial thickness across all sessions in-vivo (left) and across all states at across all sessions (right)

The spleen exhibits consistent, increased stiffness beginning with Postmortem and increasing stiffness after the tissue is extracted and subjected to the later conditions. This further supports the notion that the stiffness of the spleen is at least partially determined by fluid/viscous effects, as the fluid in the spleen is evacuated as time goes on (after death).

Spleen data displays strong viscous effects during the In-Vivo condition and becomes increasingly stiff with the later conditions. This is unsurprising, as the spleen contains a significant amount of blood in life and begins to “drain” starting with postmortem. By the time the tissue is the Ex-Situ condition, a significant amount of blood has evacuated from the spleen, resulting in significantly reduced size of the organ, as well as increasingly stiff measurements, as the tissue becomes thin and the grasper presses against itself.

Belly Peritoneal Lining

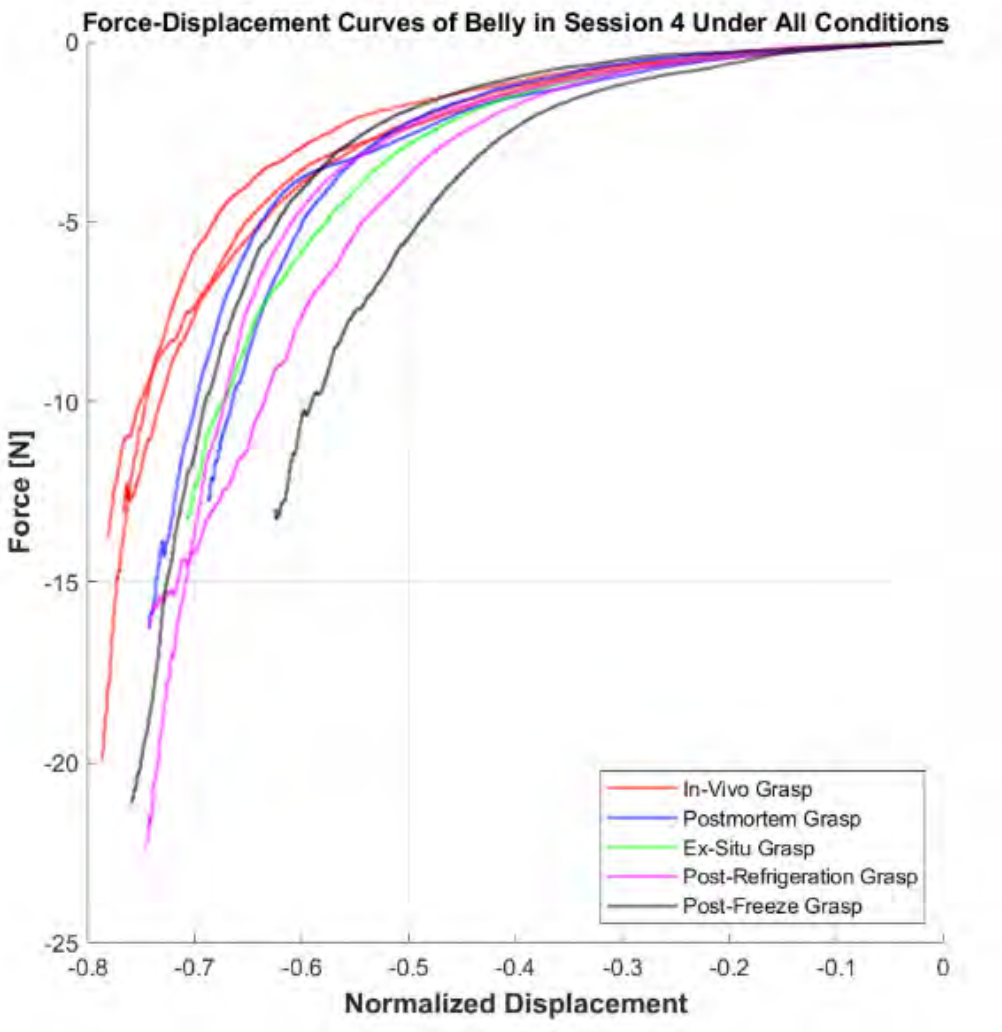


Figure 18: Porcine belly/peritoneal lining force-normalized displacement curve for all first grasps from session/animal 4

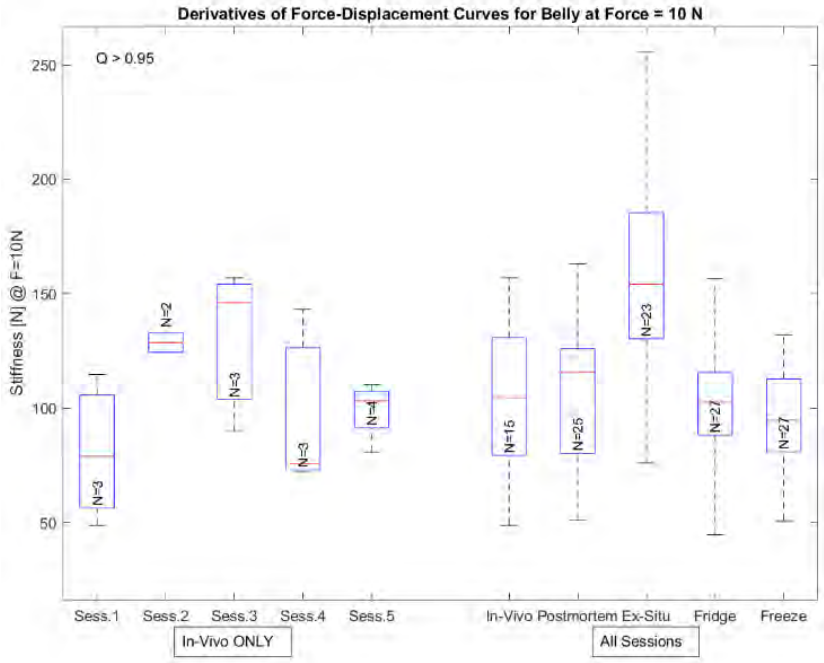


Figure 19: Porcine belly stiffness at an applied force of 10N across all sessions in-vivo (left) and across all states across all sessions (right)

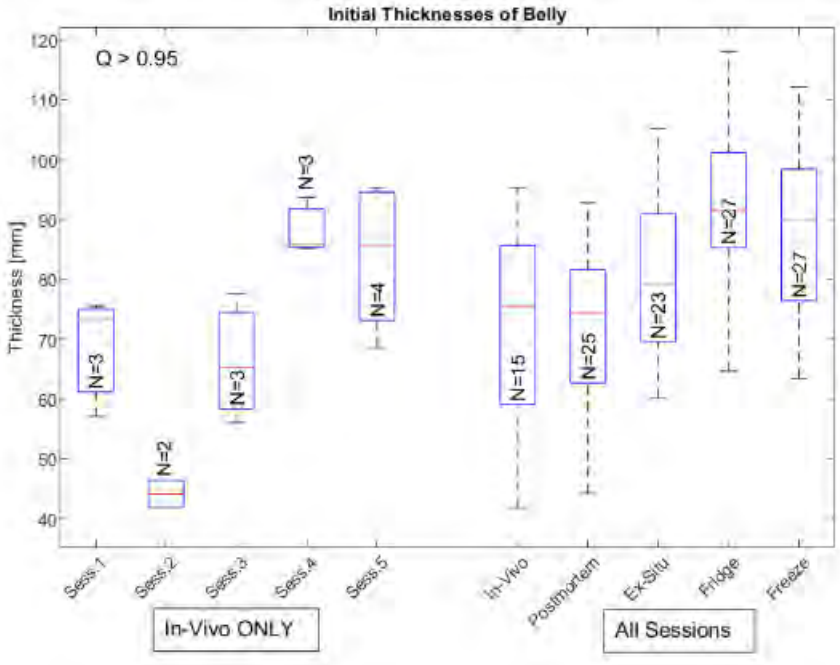


Figure 20: Porcine belly initial thickness across all sessions in-vivo (left) and across all states at across all sessions (right)

“Belly” data shows almost no variation in across testing conditions, except for the Ex-Situ condition. The cause of this discrepancy is unknown, and its cause could simply be due to methodology and/or temperature effects.

Lung

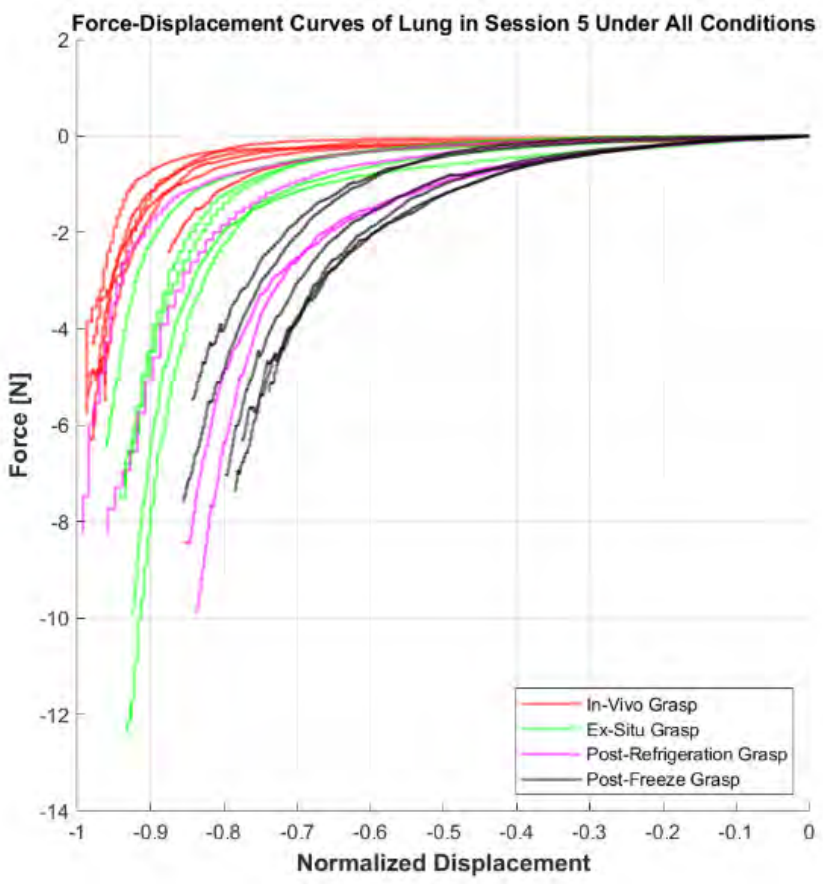


Figure 21: Porcine ling tissue force-normalized displacement curve for all first grasps from session/animal 5

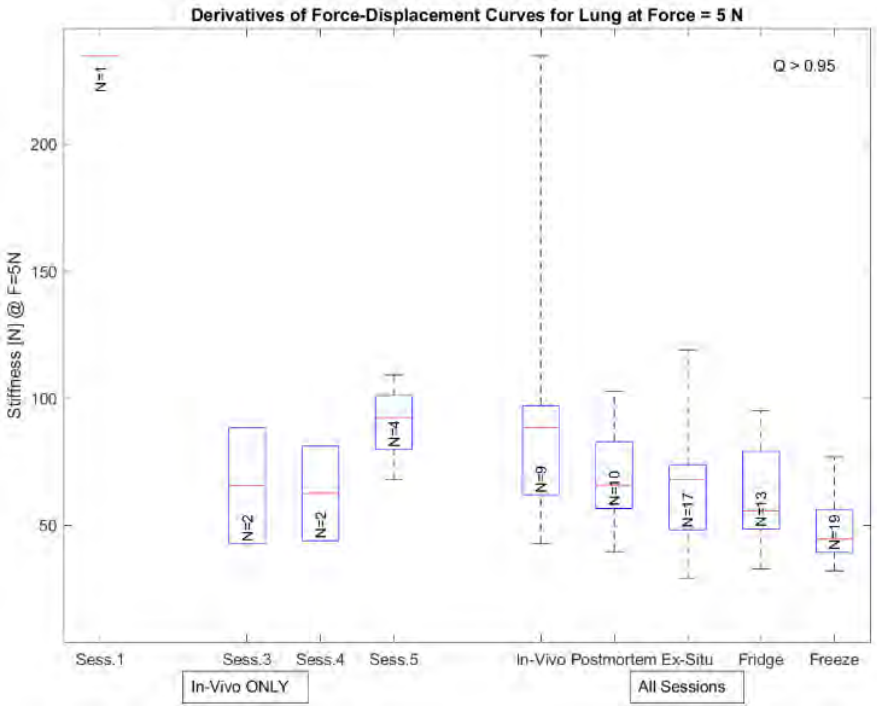


Figure 22: Porcine lung stiffness at an applied force of 5N across all sessions in-vivo (left) and across all states across all sessions (right)

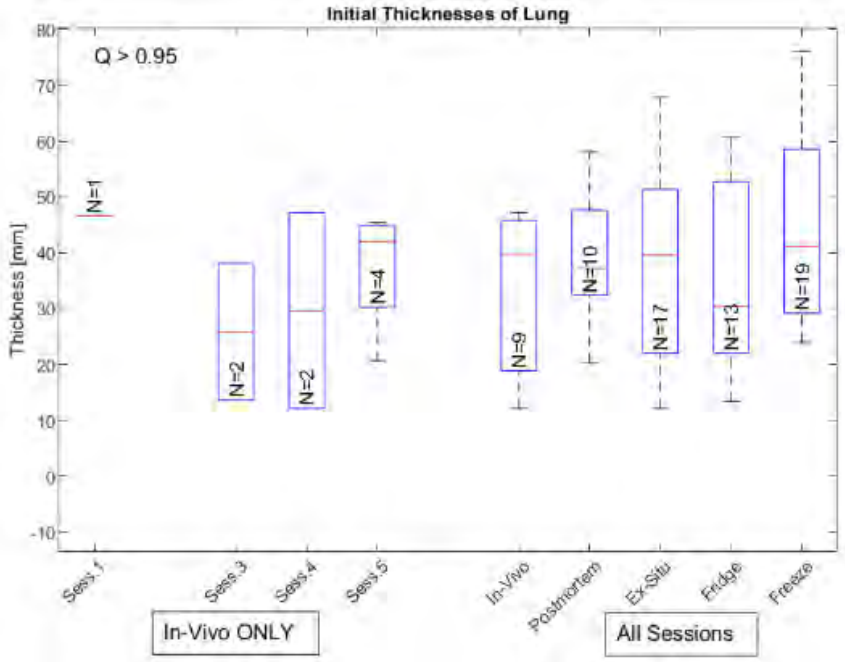


Figure 23: Porcine lung initial thickness across all sessions in-vivo (left) and across all states at across all sessions (right)

Lung data indicates a somewhat steady decline in stiffness as time goes on. The cause of this “softening” effect is unknown and may merit further review. However, the significant apparent variation in stiffness during In-Vivo is likely caused by the active expansion and contraction of the lungs as the animal breathes. While the breathing effects are unlikely to cause significant effects to the true stiffness of lung tissue, the expansion and contraction of the lungs significantly impact

the handling of the tissue and other tissues around it; therefore, an accurate simulator must simulate lung expansion and contraction as the “patient” breathes.

Updates and Facilities Status

IT and Data Storage

All raw and processed data is stored within the Medical Robotics and Devices Lab’s google drive as well as on a backup physical hard drive within the lab. All supporting documents are also stored withing the lab’s google drive and as physical copies within the lab. Additionally, all data analysis code is stored within the with lab’s GitHub repository.

Equipment Status

The grasper device, and all relevant equipment, have been shipped to CREST at the University of Washington for potential future tissue testing and data collection. An additional backup grasper device remains at the University of Minnesota for potential future tissue testing, data collection, or similar projects.

Sean Moen, Special projects director under the school of medicine and the department of neurosurgery continues to co-operate out of our lab.

General Lab Operations

STAFFING

Faizan Malik and Bradley Drahos were hired to work full time on the tissue collection cooperative project as part of data collection, data analysis, and publication work.

Upcoming Events

Final Tasks

- Share data via PLOS ONE or Dryad publish data repository pending acceptance of PLOS ONE in-vivo paper
    - Consider alternative data hosting
- Generation of final data analysis results figures
- Statistical analysis review with statistician for major publication(s)
- Completion of primary publication analyzing tissue mechanical response of porcine tissue across animals and across tissue states for liver, spleen, and peritoneum tissues
- Potential additional publication(s) pending statistical analysis review

Publications

- Drahos, B., Safdari, A., Malik, F., Smith, R., Kubala, M., Norfleet, J., Parsey, C., Goodwin, S., Kowalewski, T., (2020). Design of a Handheld Tissue Grasping Device to Measure Tissue Mechanical Properties In-Vivo or in a Laboratory Setting. In 2020 *Design of Medical Devices Conference*, *in press*. American Society of Mechanical Engineers, ASME Digital Collection, 2020.
- **“**Design of a Handheld Tissue Grasping Device to Measure Tissue Mechanical Properties In-Vivo or in a Laboratory Setting”, Design of Medical Devices Conference Virtual Rapid-Fire Presentation Session, Minneapolis, MN, June 2020
- Xiaoyin Ling, Amer Safdari, Michael B. Tradewell, Robert M. Sweet, and Timothy M. Kowalewski. Dynamics of Foley Catheter Insertion: A Cadaver Study. In *Engineering and Urology Society, 34th EUS Annual Meeting, San Francisco, CA*, 2019.

- Amer Safdari, Xiaoyin Ling, Michael B. Tradewell, Timothy M. Kowalewski, and Robert M. Sweet. Practical, Non-Invasive Measurement of Urinary Catheter Insertion Forces and Motions. In *2019 Design of Medical Devices Conference*, pages V001T06A016--V001T06A016. American Society of Mechanical Engineers, ASME Digital Collection, 2019.
- Amer Safdari. Towards More Accurate Medical Simulation via Procedural Instrumentation. Master's thesis, The University of Minnesota Twin Cities, 2019.
- Xiaoyin(Catherine) Ling. Dynamics of Male Urethra Catheterization: Simulator and Cadaveric Study. Master's thesis, The University of Minnesota Twin Cities, 2019.
- Xiaoyin Ling, Michael Tradewell, Robert M. Sweet, and Timothy M. Kowalewski. A catheter insertion force assessment tool: Design and preclinical results. In *Engineering and Urology Society, 33rd EUS Annual Meeting, San Francisco, CA*, 2018.

Publications in Progress

- Drahos, B., Norfleet, J., Parsey, C., Malik, F., Safdari, A., Kowalewski, T., (2020). Determining Potential Changes in Tissue Mechanical Properties of Porcine Liver and Spleen from In-Vivo to a Laboratory Setting Port-Refrigeration and Post-Freeze to Facilitate More Accurate Medical Simulators. In 2020 Military Health System and Research Symposium, *[ACCEPTED, CONFERENCE CANCELLED]*.

Expenditures to Date

Spending on Current Project Budgets

Sponsored Project Detail

Report Run Date: Friday,7/24/202
Data as of Friday,07/24/2020 02:49 AM

Sponsored Project: 00075541
Accounting Date Through: 7/24/2020
Encumbered Facilities and Admin: Included
Sponsor: U.S. Army Research Laboratory
Award: CON000000045559
Department: 11135
Project Start Date: 9/18/2018
Project End Date: 9/17/2020
PI: Kowalewski,Timothy Mariusz

Project Budget: $349,176.00
Program Income: $0.00
Tip Income: $0.00
F&A: RSRCH MTDC1 54.00%
F&A Distribution: 11135 MechEngAdm 100.00%

	Sponsored:	Sponsored: 07/01/2020 - 07/24/2020		Sponsored: Inception Through 07/24/2020			Cost Share:	Cost Share: 07/01/2020 - 07/24/2020		Cost Share: Inception Through 07/24/2020		
Detail Account	A Budget Amount	B Pre-Enc & Enc	C Expenses	D Pre-Enc & Enc	E Expenses	F=A-D-E Available Balance	G Budget Amount	H Pre-Enc & Enc	I Expenses	J Pre-Enc & Enc	K Expenses	L=G-J-K Available Balance
700100 Salaries-Faculty-Fin Bdg Only	27,560.00	0.00	0.00	0.00	14,000.83	13,559.17	0.00	0.00	0.00	0.00	0.00	0.00
700200 Salaries-P/A/Police-FinBdgOnly	48,750.00	0.00	0.00	0.00	0.00	48,750.00	0.00	0.00	0.00	0.00	0.00	0.00
700210 Salaries-Post Doc-FinBd	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
700310 Salaries-Grd/Pr - FinBdg	27,969.00	0.00	0.00	0.00	85,585.33	-57,616.33	0.00	0.00	0.00	0.00	0.00	0.00
700500 Salaries-Civil Service-Fin Bdg	17,018.00	0.00	0.00	0.00	25,581.22	-8,563.22	0.00	0.00	0.00	0.00	0.00	0.00
700530 Salaries-Temp/Casual-Fin Bdg	10,000.00	0.00	0.00	0.00	9,827.51	172.49	0.00	0.00	0.00	0.00	0.00	0.00
710100 Fringe - Faculty-Fin Bdg Only	9,426.00	0.00	0.00	0.00	5,040.30	4,385.70	0.00	0.00	0.00	0.00	0.00	0.00
710200 Fringe-P/A/Police-FinBdgOnly	17,550.00	0.00	0.00	0.00	0.00	17,550.00	0.00	0.00	0.00	0.00	0.00	0.00
710210 Fringe-Post Doc-FinBd	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
710300 Fringe-Grad/Pr-w/Tuitn-FinBdg	15,522.00	0.00	0.00	0.00	15,378.89	143.11	0.00	0.00	0.00	0.00	0.00	0.00
710310 Fringe-Grad/Pr No Tuitn-FinBdg	4,951.00	0.00	0.00	0.00	14,024.87	-9,073.87	0.00	0.00	0.00	0.00	0.00	0.00
710500 Fringe-Civil Service-Fin Bdg	6,628.00	0.00	0.00	0.00	7,461.73	-833.73	0.00	0.00	0.00	0.00	0.00	0.00
710530 Fringe-Temp/Casual-Fin Bdg	770.00	0.00	0.00	0.00	792.74	-22.74	0.00	0.00	0.00	0.00	0.00	0.00
720100 Gen Oper Supplies-FinBdg Only	200.00	0.00	0.00	0.00	270.07	-70.07	0.00	0.00	0.00	0.00	0.00	0.00
720200 Lab/Med Supplies-FinBdg Only	15,306.00	0.00	0.00	0.00	12,200.39	3,105.61	0.00	0.00	0.00	0.00	0.00	0.00
720300 Gen Oper Services-Fin Bdg Only	3,231.00	0.00	0.00	0.00	175.65	3,055.35	0.00	0.00	0.00	0.00	0.00	0.00
720400 Lab/Medical Svcs-FinBdg Only	8,000.00	0.00	0.00	0.00	2,866.42	5,133.58	0.00	0.00	0.00	0.00	0.00	0.00
720600 Travel/Mileage/Mov-FinBdg Only	8,000.00	0.00	0.00	0.00	10,067.28	-2,067.28	0.00	0.00	0.00	0.00	0.00	0.00
750100 NC Bldgs/Equip-Fin Bdg Only	6,300.00	0.00	0.00	0.00	19,030.56	-12,730.56	0.00	0.00	0.00	0.00	0.00	0.00
850100 Capital Equip - Fin Bdg Only	8,246.00	0.00	0.00	0.00	15,173.07	-6,927.07	0.00	0.00	0.00	0.00	0.00	0.00
Direct Cost Total	**$235,427.00**	$0.00	**$0.00**	**$0.00**	**$237,476.86**	**-$2,049.86**	**$0.00**	**$0.00**	**$0.00**	**$0.00**	**$0.00**	**$0.00**
810500 F/A Costs-Fin Bdg Only	113,749.00	0.00	0.00	2,009.47	111,739.53	0.00	0.00	0.00	0.00	0.00	0.00	0.00
F&A Total	**$113,749.00**	$0.00	**$0.00**	**$2,009.47**	**$111,739.53**	**$0.00**	**$0.00**	**$0.00**	**$0.00**	**$0.00**	**$0.00**	**$0.00**
Project Total	**$349,176.00**	**$0.00**	**$0.00**	**$2,009.47**	**$349,216.39**	**-$2,049.86**	**$0.00**	**$0.00**	**$0.00**	**$0.00**	**$0.00**	**$0.00**

Encumbered Facilities and Admin is included on this report.

Appendix A:

ARL In-Vivo Testing Protocol

Study Overview

Each subject pig will undergo mechanical characterization on both the spleen and liver over 5 experimental conditions. The objective is to measure and quantify the difference in tissue response between in-vivo to ex-vivo, along key testing conditions that may capture the decay of tissue over its post-mortem lifecycle. A mechanical grasping tool instrumented with load cell arms and an encoder will perform the measurement. VHL personnel will assist in providing anatomical access in-vivo, in a manner that is consistent with our animal protocol and minimizes bleeding. Additionally, in-vivo VHL lab personnel will assist in identifying regions that are accessible for the grasping tool. The liver lobe(s) accessible in-vivo will be noted, and any further tests on those lobes will be indicated on the notes sheet as info for analysis.

Following the in-vivo experiments, the procedure will be replicated in-situ after the animal has been euthanized. We will wait a minimum of 15 minutes after the conclusion of the in-vivo tests to ensure the organs have had a chance to mechanically relax into their native state. Post euthanasia, the animal will receive a large abdominal incision to allow full organ access for in-situ testing. Throughout our testing every grasp site is dried and labeled to prevent retesting on the same spot. After the conclusion of the in-situ experiments, the organs will be harvested and transported to another lab for further testing ex-vivo ex-situ as soon as possible.

Experiments are similarly conducted on these excised organs, and after completion the liver is dissected into its constituent lobes to allow for easier handling and thawing. Each liver lobe, and the spleen will receive 25 mL of 1x PBS/pen-strep solution and stored in a zip lock bag in a fridge at 40° F. After 24 hours (tolerance 18-36h), the organs will be heated in a water bath for 2 hours at physiological temperature and further tested. The organs are replaced into their original bags containing the 1x PBS/pen-strep solution, and frozen for 3-5 days. This last experimental condition will test on spots that have been previously tested on, identified by any remaining and visible prior markings.

Pre-study Preparation

Calibration

1) On the morning of testing day, calibrate load cells using the array of known weights in the blue container.
2) Start by running the data collection software, and using the signal conditioners to tare both load cells within +/- 0.01 V
    a) Use the small futek screwdriver to adjust tare offset
3) Collect data on the 10, 20 (both), 50, 100, 200 (both), and 500 g weights. Save this to the calibration folder with the date of the testing.

Stock Cart and Prepare Supplies

Stock cart with supplies indicated in checklist. Ensure 100x pen-strep solution is thawing in the fridge the morning of the experiment.

Study Preparation

1) Ensure load cells were calibrated
2) Setup hardware and software
    a) NI DAQ breakout connected to PC
    b) Futek conditioners connected to NI DAQ breakout
    c) Scissors connected to Futek conditioners
    d) Confirm all 3 data channels collecting and in range: Force & Position, both jaws
3) Setup video collection
    a) Make sure GoPro camera is charged
    b) Setup tripod
4) Open software
    a) Located in `~/Documents/scissors_console`
5) User sets destination directory for data
    a) Usually `~/Documents/[today's date eg. 20190220]/`
6) Install XL nitrile gloves over ends of scissors

Study Role Sheets

Indicated below.

Computer person

1) Set filename for trial [XYZ] - timestamps automatically recorded
    a) Study coordinator will indicate this to you
    b) X - tissue
        i) 1 = liver
        ii) 2 = spleen
        iii) 9 = Calibration synthetic (red)
    c) Y - experimental setup
        i) 0 = in-vivo, in-situ
        ii) 1 = ex-vivo, in-situ
        iii) 2 = ex-vivo, ex-situ, fresh
        iv) 3 = ex-vivo, ex-situ, post 24 hour refrigeration cycle in 1x PBS/pen-strep solution (tolerance minimum 18 hours, maximum 36 hours)
            (1) Duration is defined as time put into fridge, and time at which tissue is completely heated and at physiological temperature (37 C) in water bath
        v) 4 = ex-vivo, ex-situ, post >72 hour freeze-thaw cycle (tolerance minimum 72 hours, maximum 5 days)
            (1) Duration defined similarly as above
    d) Z - trial number
        i) 0 - 9
2) Start test when indicated by others
    a) Confirm start of data acquisition
    b) Confirm index pulse was triggered (encoder values look consistent from trial to trial usually ranging between 100 and -900)
    c) Confirm end of data acquisition
    d) Confirm filename of last recorded trial
3) Stop test when indicated by others
4) Monitor sensor feed for anomalies (sensor flatlining, erratic noise, large (> 0.1 V) offset between load cell values, etc etc)
    a) Indicate to study coordinator if present

Scissors wielder

1) Setup
    a) Setup computer and software for computer person
        i) Open program
        ii) Set folder paths and filenames
        iii) Ensure operator understands their role
    b) Install gloves onto scissors
    c) Mount Futek signal conditioners to arm
    d) Install biohazard bag over scissors, futek conditioners, and arm
        i) Allow small hole in bag for scissors to extend through
2) Calibration Grasp
    a) Do 10 “practice grasps” on synthetic tissue standard (for reference point before a trial)
    b) Conduct this on the Red tissue puck that is located with the data collection cart
    c) Do this prior to starting or ending with an organ
3) Perform a single Trial: Perform a target of **10** grasps on tissue in specific site - region indicated by the tissue holder
    a) Wait for tissue holder and study coordinator to signal readiness
    b) Begin trial by flexing scissors from fully open to fully closed (through air)
        i) Do this at least 3 times
        ii) Confirm with computer person that the index pulse was triggered
    c) Grasp at a speed such that 3 grasps (open -> close -> open = 1 grasp) are completed in roughly 5 seconds
    d) Within a grasp, continue squeezing tissue until the onset of a “membrane squishing/popping” sensation is felt and then backoff for the next grasp
    e) Count grasps out loud
    f) Minimum acceptable grasps without significant anomalies is **7** for liver, **5** for spleen.
        i) Significant slippage, significant tissue tearing/damage, equipment failures, sensor flatlining, other unforeseen trouble etc
4) Once complete, signal to others that trial is done; confirm computer person stopped data collection
5) Blot grasping site clean, and mark with designated marker
    a) Black on in-vivo test, silver on all others
6) Prompt study coordinator for new site; room permitting
7) Repeat steps 5-7 until tissue holder confirms completion of trials on the particular organ
8) At the conclusion of testing all organs, conduct a final calibration grasp on the synthetic

Tissue holder

1) Stand on the opposite side of the OR table relative to the scissors wielder
2) Identify tissue of interest
    a) Liver
    b) Spleen
3) Prompt study coordinator for next tissue site
4) Hold and expose tissue such that a “lobe” can be grasped within the region of interest
    a) **MINIMIZE Palpation, and handling of tissue. Ideally spot that grasper goes on should NEVER be touched beforehand.**
    b) Do not hold tissue artificially taut i.e. do not stretch
    c) Also do not compress/squish tissue
5) Identify target tissue
6) Indicate grasping site
    a) Criteria Are:
        i) Avoid slippery spots
        ii) Accessible
        iii) Avoid extraneous anatomy (gallbladder)
        iv) Diversity in location
        v) Spots that aren’t too close to edge (1 cm away)
        vi) Spots that haven’t been grasped before (1 cm away)
        vii) Use up available real estate
    b) Mesentery on spleen
        i) Ok to test over
        ii) Can also cut in-situ if necessary for access
7) Signal readiness to others
8) Wait for scissors wielder to finish test
9) Work with scissors wielder to mark the most recently grasped site
10) Prepare for next grasp at direction of study coordinator

Study Coordinator

1) Arrange for all pre-study checklists to be completed for day of study
    a) See experimental checklist document
2) Ensure computer person and tissue holder understand their roles
    a) Make sure tissue handler understands handling criteria i.e. as soft and gentle as possible, while providing access
3) What progress of study protocol, ensure enforcement
    a) Use Experimental Checklist document to record compliance with protocol
4) Ensure calibration grasp is conducted
5) As scissors wielder is counting out grasps (1 to 10) check camera footage for focus, clear field of view, and lighting.
6) Ensure scissors wielder blots and marks last grasp spot with marker
7) Note any remarks by scissors wielder, or any observations of anomalies, etc on experimental checklist

Data Exclusion Criteria

1) Anomalous Grasps due to sensor issues or unusual trajectories by user:
    a) Grasps with excessive mean speeds are not considered for analysis
    b) Grasps with exceedingly slow mean speeds are not considered for analysis
    c) Grasps with peak forces exceeding 10V are not considered for analysis
    d) Grasps with peak forces below 1V are not considered for analysis
    e) Grasps on tissues below thickness resulting in less than 100 encoder ticks are not considered for analysis
    f) Grasps on tissues above thickness resulting in greater than 1500 encoder ticks are not considered for analysis
    g) Incomplete trials or trials where video review confirms non-compliant conditions: tissue slip or shear, incomplete grasps or unintended events like ‘bumps’ or ‘pulls’, lost grip, etc.
    h) Anomaly of sensation reported by scissors wielder on video during experiment

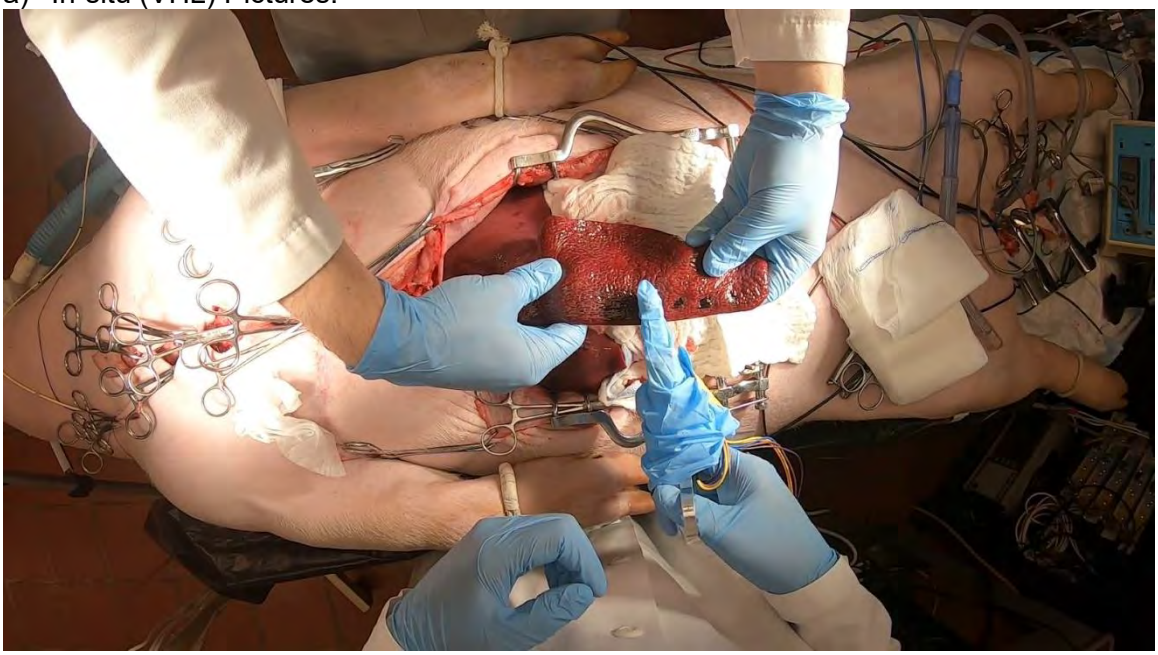
Inclusion Criteria

1) Organs of interest (in order of priority)
    a) Liver
    b) Spleen
    c) Artery?

Appendix B:

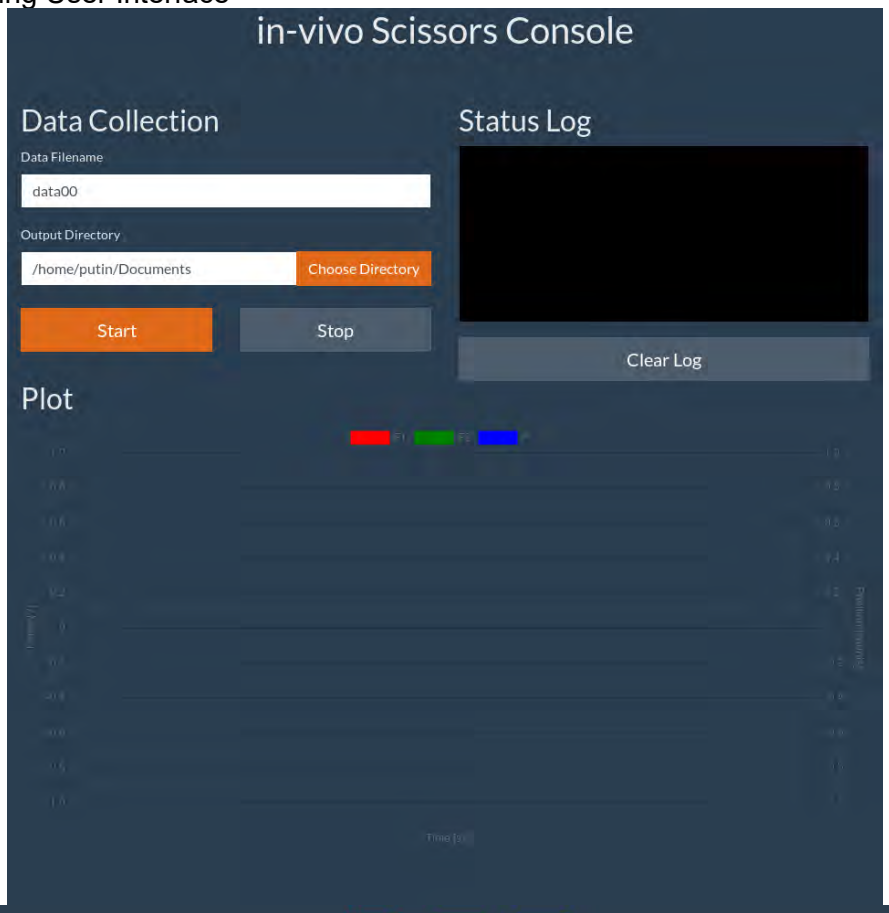
Winter 2019/2020 In-Vivo Update

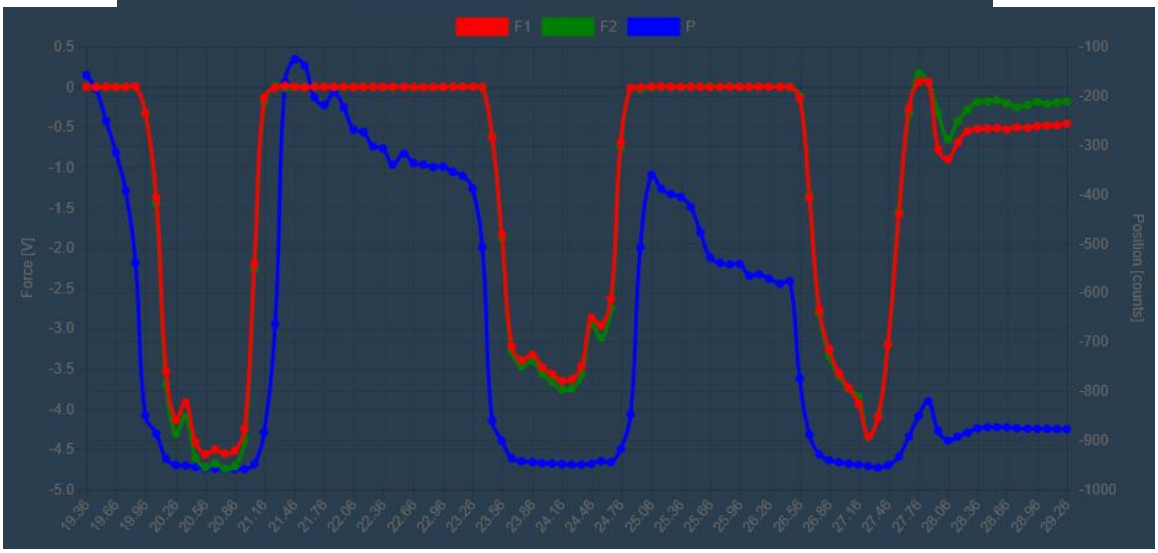
1) Major Points
   a) All testing complete
   b) Data analysis moving forward
   c) Paper submitted and accepted to DMD conference regarding grasper
2) Testing
   a) In-situ (VHL) Pictures:



   b) In-situ (VHL) Pictures:

c) Testing User Interface

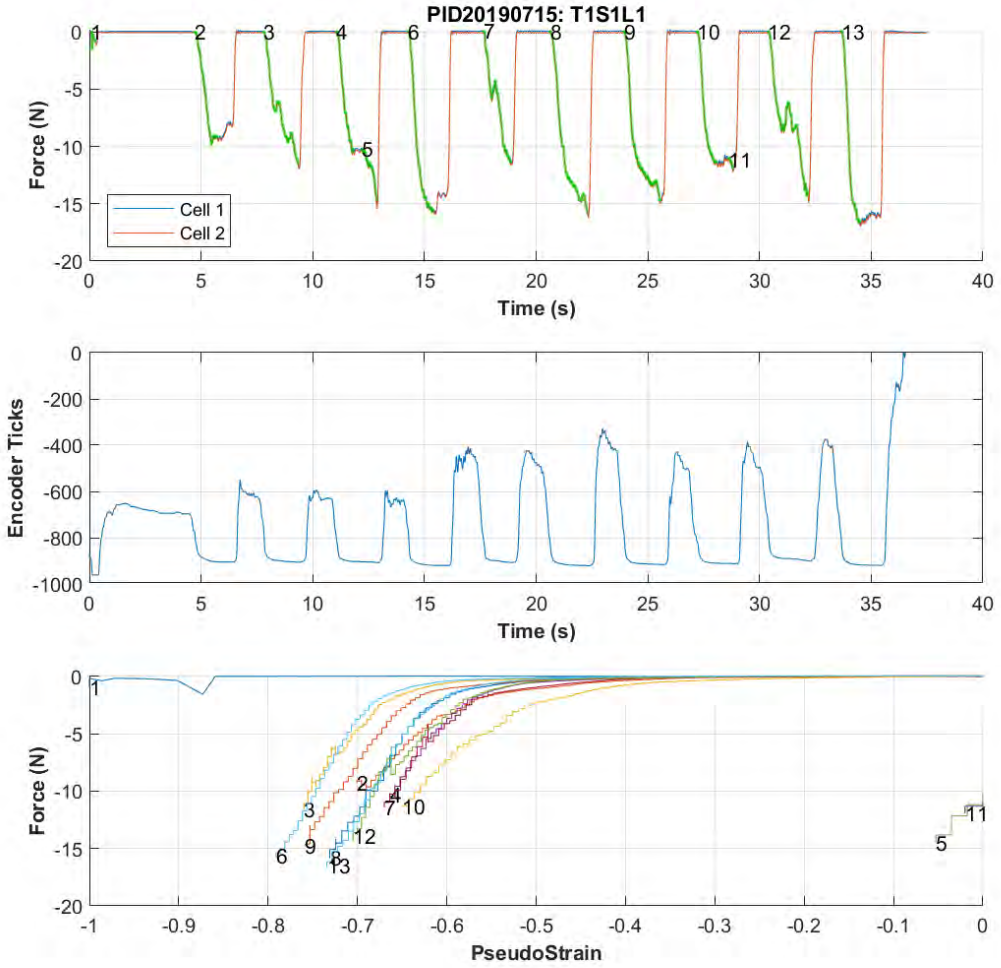




3) Data analysis
   a) Still in the process of completing data analysis. The code that was initially made and testing using the calibration puck runs into issues when analyzing testing completed on tissue. The code segments an entire trial of 10 grasps and then automatically flags grasps based on changes in force and grasper angle, but we are finding that it is flagging small bumps as grasps even though they weren't

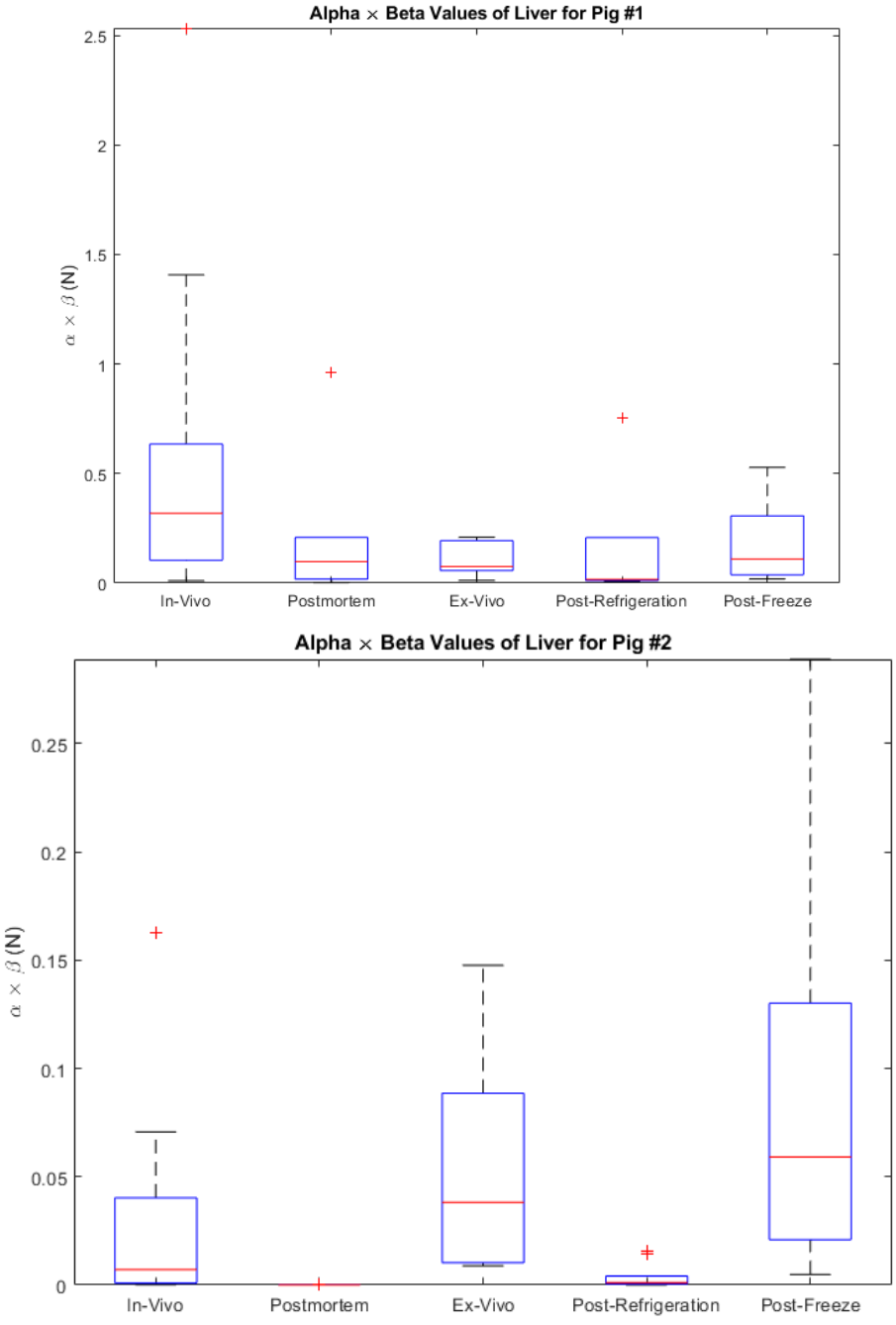
testing grasps. What we are doing now is refining how we flag grasps in the code so we can rerun the code for every trial we did across all five pigs instead of having to manually unflag the poor grasps and reflag correct grasps that were split due to improper grasp flagging. Shown below is a trial where our grasp flagging code misflagged some grasps.

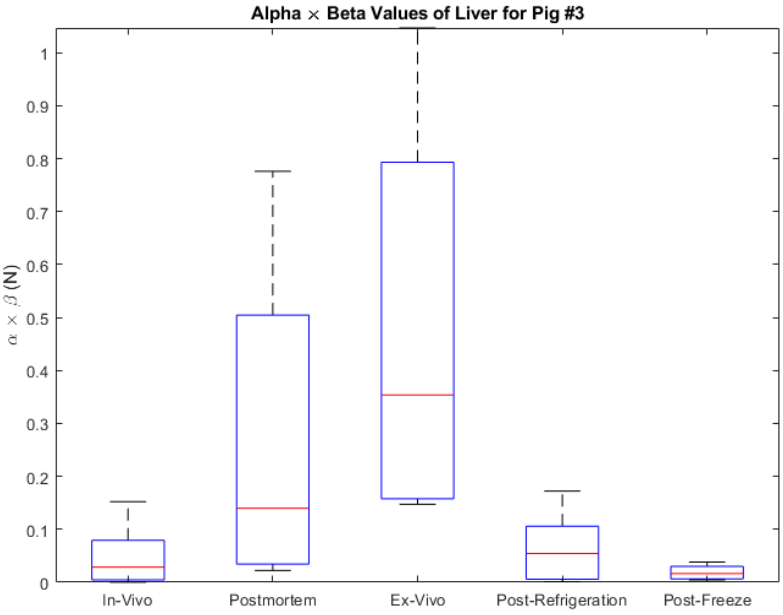
(a) By looking at the force plot you can see that there are 13 grasps flagged but only 10 actual grasps. What is happening is that its flagging a small jump in force, probably due to closing the grasper fully or bumping tissue, at the very beginning as a grasp. Grasps 5 and 11 are also misflagged grasps that we need to filter out which are happening because the grasps before are not continually increasing in force, so the program things that there is another grasp in the middle of grasps 4 and 10. We just found out about this issue last week and we have ideas on how to work around this issue which we should be able to implement within a week or
so.

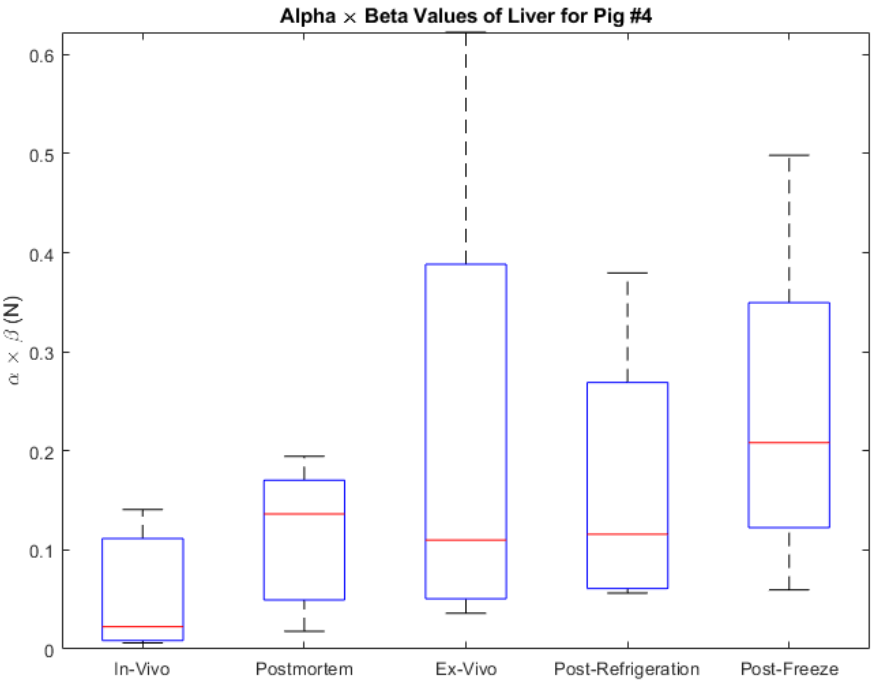


b) In regard to generating figures, we are still processing all of the data so there are not many updates on that front. Shown below are plots that will be similar to the

final ideal plots. We want to generate histograms of the alpha times beta values for each tissue/state/pig to note any differences between tissues, states, or pigs. You might have already seen these figures last time you were in town. Keep in mind that these plots do not account for the exclusion of poor trials or outliers.







4) Papers and Conferences
   a) We have submitted a technical brief paper to the DMD conference at the University of Minnesota and it was recently accepted. This paper focuses on the grasper itself, its design, and potential uses. This will allow us a reference when writing the future papers regarding tissue data collection.
   b) We are currently trying to come up with a good submission to the MHSRS conference in a month or so
   c) Once data analysis is finally complete, we are going to focus on writing the ⅘ papers we initially had planned.

Appendix B:

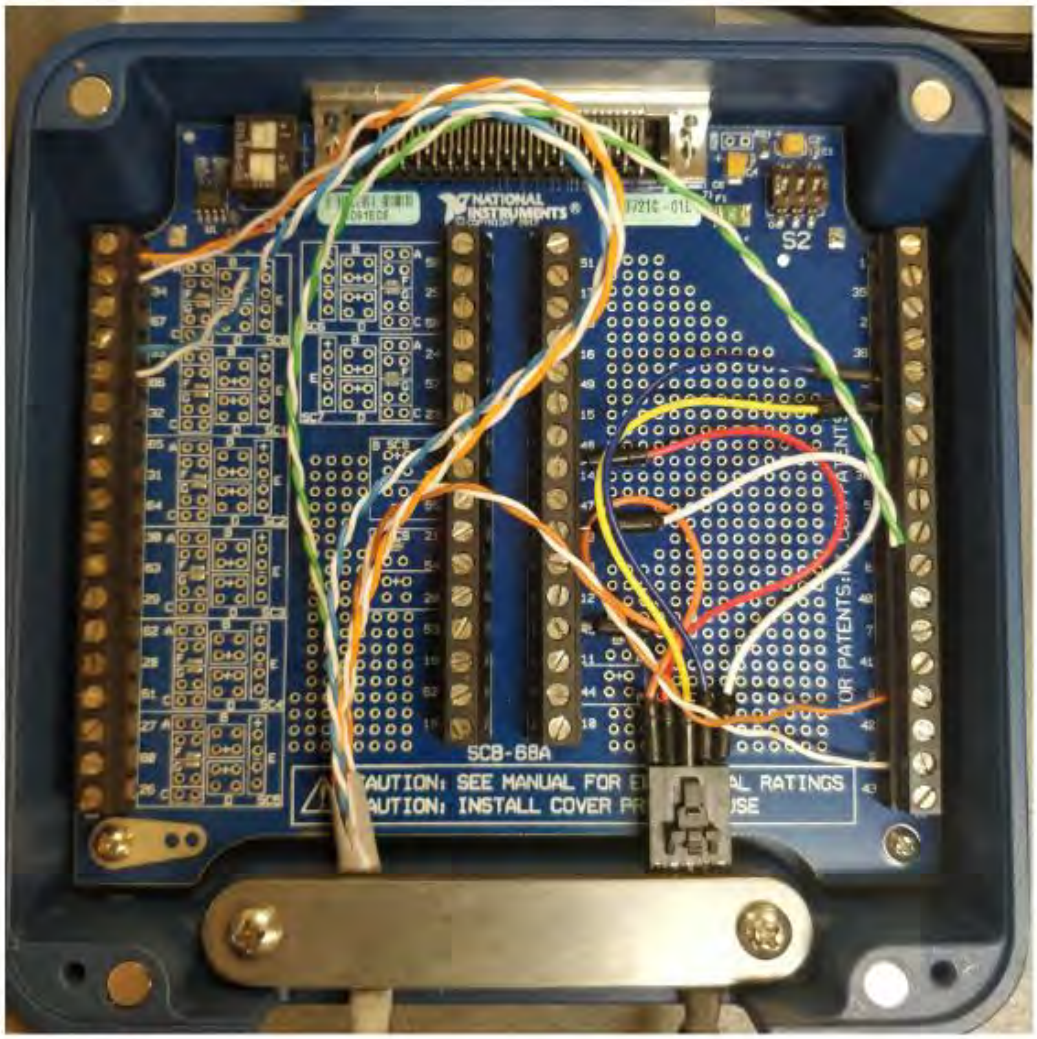
RUST Data Acquisition GUI and MATLAB Data Analysis Code Access:

To access code necessary to run grasper device GUI console for data acquisition or the data analysis scripts see github.com/labmrd for necessary repositories:

RUST Data Acquisition GUI Setup

1. See Scissors_Console repository on github.com/labmrd or go directly to github.com/labmrd/Scissors_Console
2. Clone repository
3. Ensure Rust compiler is set up including VS 2019 build tools if on Windows 10
4. Follow rust building guidelines outlined in README.me which outlines how to generate scissors console executable
5. Install version 18.6 of the NIDAQ-mx drivers as outlined in README.me
    a. Linux – Install Webkit2GTK 2.8 from your distro's package manager. Then follow the instructions to download and install the RPM file from NI on a Radhat based OS
    b. Windows – NI drivers can be installed by downloading and installing the NI package manager
6. Prior to running the scissors console executable, ensure all wiring connections are connected as presented in the figure on the next page:

NIDAQ Pinout



1. ADC0
    a. AI 0 (AI 0+)
    b. AI 8 (AI 0-)
2. ADC1
    a. AI 1 (AI 1+)
    b. AI 9 (AI 1-)
3. Encoder
    a. Pin 14 (5V)

MATLAB Data Analysis Setup

1. See InVivo_Data_Processing repository on github.com/labmrd or go directly to github.com/labmrd/InVivio_Data_Processing

2. Clone repository

3. Follow procedural instructions as outlined in README.txt to generate necessary .mat file necessary for data analysis

Appendix C:

MATLAB Data Analysis Code:

7/30/20 6:03 PM G:\MRD\InVivo_Data_Processing\Pr...\ReadDataNew.m 1 of 3

```
clc
clear all
close all

% Specify pig ID # to add to data structure
%G:\Shared drives\MRD\in-vivo\in-vivo\pig_20190807\data_20190807
%G:\Shared drives\MRD\in-vivo\in-vivo
% {'20190715', '20190724', '20190807', '20190918', '20191023'}
PigID = {'20190724','20190807','20190918','20191023'};

APPEND_TO_EXISTING_DATA = 0;

% PigID = {'20190807'};
% 1 = liver, 2 = spleen, 3 = IVC, 4 = Aorta, 5 = Lung, 6 = Belly, 7 = Esophagus, 9 = Puck
TissueID = 1:9;
% TissueID = [1,2];

% 0 = in-vivo; 1 = postmortem in-situ; 2 = ex-situ; 3 = post-refrigeration; 4 = post-freeze
TestID = [0:4] + 1;

TestSiteID = 1:20;
SAMPLE_PERIOD = .001; % Seconds; from NI Card;

if APPEND_TO_EXISTING_DATA
    load 'TissueData.mat';
end

offset1 = [];
slope1 = [];

offset2 = [];
slope2 = [];


for pig = 1:1:size(PigID,2)
    pigFolder = strcat( 'G:/Shared drives/MRD/in-vivo/in-vivo/pig_' , PigID{pig})

    CalCoeffs{1} = calibrateLoadCell(strcat(pigFolder, '/day 0 calibration/Pre' ));   % First calibration

    calDir = strcat(pigFolder, '/day 0 calibration/Mid 0-1' );
    if exist(calDir,'dir')  % If intermediate calibration data exists, use it
        CalCoeffs{2} = calibrateLoadCell(calDir);
    else                    % Otherwise, revert to previous calibration data.
        CalCoeffs{2} = CalCoeffs{1};
    end

    CalCoeffs{3} = calibrateLoadCell(strcat(pigFolder, '/day 0 calibration/Post' ));

    CalCoeffs{4} = calibrateLoadCell(strcat(pigFolder, '/day 1 calibration/Pre' ));
    CalCoeffs{5} = calibrateLoadCell(strcat(pigFolder, '/day 3 calibration/Pre' ));

    dataFolder = strcat(pigFolder, '/data_', PigID{pig}, '/');

    vals = CalCoeffs{1,1};
    offset1 = [offset1 vals(1,1,1)];
    slope1 = [slope1 vals(1,2,1)];
```

```
    offset2 = [offset2 vals(1,1,2)];
    slope2 = [slope2 vals(1,2,2)];


    for Tissue_Index = 1:1:size(TissueID,2)
        Tissue = TissueID(Tissue_Index);

        for Test = 1:1:size(TestID,2)

            for TestSite = 1:1:size(TestSiteID,2)
                % Convert from MATLAB Index System to our ID System, and generate the folder directory.
                ID = strcat(num2str(TissueID(Tissue_Index)), num2str(TestID(Test)-1), num2str(TestSiteID(TestSite)-1));
                baseFolder = strcat(dataFolder, ID);

                if exist(baseFolder, 'dir')  % Check if data folder actually exists.
                    fprintf( 'Loading Data Folder: %s' , ID);
                    folder = dir(baseFolder);
                    filesInDir = folder(~([folder.isdir]));
                    for i=1:1:size(filesInDir,1)     % Scan files in directory for the 'enc' and 'adc' files.
                        if strfind(filesInDir(i).name, 'enc')
                            PositionFile = strcat(baseFolder, '/',filesInDir(i).name);
                        elseif strfind(filesInDir(i).name, 'adc')
                            ForceFile = strcat(baseFolder, '/',filesInDir(i).name);
                        end
                    end
                    rawPositionData = load(PositionFile);     % Load raw data into variables.
                    rawForceData = load(ForceFile);

                    fprintf( '.\n'); % Done loading files

% We know that the encoder and voltage data streams are parallel, and their
% data indices are synchronized. However, the encoder (position) data
% begins recording slightly before the force data, so we have to
% synchronize the two. First, we remove the  first 8 seconds of the force
% data (stretching period). Then we remove all entries with indices smaller
% than the first index of the resulting force data; this ensures that the
% first index of the two streams are identical. Then we shorten the data so
% that they are both the same length. The force and encoder data can then
% be placed "alongside" each other without any need for interpolation.
% THIS IS ONLY VALID FOR TESTS USING scissors_console v0.6.1+ (Pigs 2+).

                    tempForceData = rawForceData(floor(8/SAMPLE_PERIOD):end,:);
                    tempPositionData = rawPositionData(tempForceData(1,1):end,:);

                    drl = min(size(tempForceData,1),size(tempPositionData,1));   % Data Record Length: the smallest of the two data streams.
                    tempForceData = tempForceData(1:drl,:);
                    tempPositionData = tempPositionData(1:drl,:);


                    X = tempPositionData(:,1);       % Time index
                    X = (X - tempPositionData(1,1)) * SAMPLE_PERIOD;      % Start at 0, convert to seconds
```

```
                    forceCoeffs = CalCoeffs{Test};
                    actualForceData1 = tempForceData(:,2).*forceCoeffs(1,1,1) + forceCoeffs ↙
(1,2,1);
                    actualForceData2 = tempForceData(:,3).*forceCoeffs(1,1,2) + forceCoeffs ↙
(1,2,2);

                    syncedDataTemp = [X, tempPositionData(:,2), actualForceData1, ↙
actualForceData2];
                    allData.(strcat( 'PID',PigID{pig})).TissueID(Tissue).TestID(Test).TestSiteID ↙
(TestSite).syncedData = syncedDataTemp;
                    allData.(strcat( 'PID',PigID{pig})).TissueID(Tissue).TestID(Test).TestSiteID ↙
(TestSite).testType = 1;
                else
                    %fprintf('Folder does not exist: %s\n',baseFolder{1})
                    allData.(strcat( 'PID',PigID{pig})).TissueID(Tissue).TestID(Test).TestSiteID ↙
(TestSite).syncedData = [];
                    allData.(strcat( 'PID',PigID{pig})).TissueID(Tissue).TestID(Test).TestSiteID ↙
(TestSite).testType = 0;
                end
            end

            allData.(strcat( 'PID',PigID{pig})).TissueID(Tissue).TestID(Test).temperatures = {[]; ↙
[]};
        end
    end
end

save('TissueData_Raw.mat','allData')
```

```
load TissueData.mat
PigID = fieldnames(allData);

params.RunningAvgSize = 100;

% Thresholds for data segmentation
params.LowDiffThreshold = 0.001; %0.00001;
params.LowForceThreshold = -0.0;
params.DisplacementThreshold = -400;  % (Encoder value)
params.DisplacementThreshold = 0;  % (Encoder value)

% Radius of scissors (used for determining strain)
params.r = 90; % mm

params.encoderOffset = -964; %-956;


fig = figure(3);
set(gcf, 'Position', [0, 0, 750, 700]);

PigStartIndex = 1;       % !!!!!!!!!!!!!!!!!!Adjust this to skip entries
for iPigNum = PigStartIndex:1:size(PigID,1)
    numTissue=size(allData.(PigID{iPigNum}).TissueID,2);

    TissueStartIndex = 1;    % !!!!!!!!!!!!!!!!!!Adjust this to skip entries
    for jTissueNum = TissueStartIndex:1:numTissue
        numTest=size(allData.(PigID{iPigNum}).TissueID(jTissueNum).TestID,2);

        TestStageStartIndex = 1;     % !!!!!!!!!!!!!!!!!!Adjust this to skip entries
        for kStageNum = TestStageStartIndex:1:numTest
            % We remove any invalid TestSiteID entries
            lSiteNum = 1;            % !!!!!!!!!!!!!!!!!!Adjust this to skip entries
            while (lSiteNum <= length(allData.(PigID{iPigNum}).TissueID(jTissueNum).TestID ↙
(kStageNum).TestSiteID))
                Sample_in = allData.(PigID{iPigNum}).TissueID(jTissueNum).TestID(kStageNum). ↙
TestSiteID(lSiteNum);

                % Check if entry is valid
                if allData.(PigID{iPigNum}).TissueID(jTissueNum).TestID(kStageNum).TestSiteID ↙
(lSiteNum).testType
                    fprintf( 'Processing %i,%i,%i\n' ,jTissueNum,kStageNum,lSiteNum);

%                     PlotRawData(Sample_raw.syncedData);

                    Sample = ProcessData(Sample_in,params);

                    titleText = [PigID{iPigNum}, ': T',num2str(jTissueNum), 'S',num2str ↙
(kStageNum),'L',num2str(lSiteNum)];

                    PlotGraspData(Sample, fig, titleText);

                    % Add data. Add breakpoint here to verify
                    allData.(PigID{iPigNum}).TissueID(jTissueNum).TestID(kStageNum).TestSiteID ↙
(lSiteNum).syncedData = Sample.syncedData;
                    allData.(PigID{iPigNum}).TissueID(jTissueNum).TestID(kStageNum).TestSiteID ↙
(lSiteNum).syncedClose= Sample.syncedClose;
                    allData.(PigID{iPigNum}).TissueID(jTissueNum).TestID(kStageNum).TestSiteID ↙
(lSiteNum).syncedOpen= Sample.syncedOpen;
```

```
                    allData.(PigID{iPigNum}).TissueID(jTissueNum).TestID(kStageNum).TestSiteID
(lSiteNum).Grasp = Sample.Grasp;
                    allData.(PigID{iPigNum}).TissueID(jTissueNum).TestID(kStageNum).TestSiteID
(lSiteNum).Thickness = Sample.Thickness;
                    allData.(PigID{iPigNum}).TissueID(jTissueNum).TestID(kStageNum).TestSiteID
(lSiteNum).notes = '';
                    allData.(PigID{iPigNum}).TissueID(jTissueNum).TestID(kStageNum).TestSiteID
(lSiteNum).location = '';

%                     saveas(fig,[PigID{iPigNum},'-',num2str(jTissueNum),num2str(kStageNum),
num2str(lSiteNum),'.png']);


%                     plot(Sample.syncedData(:,1), Sample.syncedData(:,3),'b.')
%                     %ylim([-7 1])
%                     hold on
%                     %plot(Sample.syncedData(:,1), Sample.syncedData(:,4))
%                     title(PigID{ii})
%                     plot(Sample.syncedClose(:,1), Sample.syncedClose(:,3),'r.')
%                     hold off

                    %pause(0.05)
                    lSiteNum = lSiteNum + 1;
                else
                    allData.(PigID{iPigNum}).TissueID(jTissueNum).TestID(kStageNum).TestSiteID
(lSiteNum) = [];
                end
                clear Sample Sample_raw Sample_in data2
            end
        end
    end
end

save('ProcessedData.mat', 'allData', '-v7.3')
```

```
%% Preamble

if ~exist('allData', 'var')
    fprintf('Loading Data... ');
    load('ProcessedData.mat');
    fprintf('Done\n');
end

fitType = 2;

PID = fieldnames(allData);


% Symbolic function used by Least Squares Curve Fitting
% x is a vector containing the alpha and beta values.
% y is the input data
F = @(x,y) x(1) * (exp(x(2).*y) - 1);
ab_init = [-0.7 -0.17];    % Initial Guess for regular curve


if fitType == 2
    ab_init = [-2 5];    % Initial Guess for squared displacement curve
end

%

% Increase solver limits, select algorithms
solver1 = optimoptions('lsqcurvefit');  % Default algorithm: 'trust-region-reflective'
solver1.MaxFunctionEvaluations = inf;
solver1.MaxIterations = inf;
solver1.StepTolerance = 1e-13;
% solver1.FunctionTolerance = eps;

solver2 = solver1;
solver2.Algorithm = 'levenberg-marquardt';

solvers = [solver1 solver2];

solverStats = zeros(1,length(solvers)+1);    % Just to keep track of which solver was used for ↙
final solution


for Pig_Num = 1:1:length(PID)                                                    % Go through ↙
all Pigs
    for Tissue_Num = 1:1:length(allData.(PID{Pig_Num}).TissueID)                 % Go through ↙
all Tissues

        switch Tissue_Num   % Initial guess based on tissue for squared displacement
            case 2                          % Spleen
                ab_init = [-1 2.5];      % normalized

            case 4                          % Aorta
%                ab_init = [-0.05 0.05];
                ab_init = [-0.25 4];     % normalized

            case 5                          % Lung
%                ab_init = [-1 0.001];
                ab_init = [-0.03 6.5];   % normalized
```

```
            case 6                              % Belly
%                ab_init = [-0.3 0.0007];
                 ab_init = [-0.5 6.5];

            case 9                              % Puck
%                ab_init = [-300 0.00015];
                 ab_init = [-20 1];

            otherwise                           % Liver and others
%                ab_init = [-1 0.005];           % Plain displacement
                 ab_init = [-2 5];              % Normalized displacement
        end

        for stage = 1:1:length(allData.(PID{Pig_Num}).TissueID(Tissue_Num).TestID)    % Go through ↙
all testing stages
%         for stage = 1:1:1 % in vivo only
            TestSites = allData.(PID{Pig_Num}).TissueID(Tissue_Num).TestID(stage).TestSiteID;
            GraspData.(PID{Pig_Num}).TissueID(Tissue_Num).TestID(stage).temperatures = allData.   ↙
(PID{Pig_Num}).TissueID(Tissue_Num).TestID(stage).temperatures;

            for site = 1:1:length(TestSites)    % Check all testing locations
%                for grasp = 1:1:length(TestSites(site).Grasp)  % All grasps
                for grasp = 1:1:1  % First grasp only
                    force1 = TestSites(site).Grasp(grasp).allData(:,3);
                    force2 = TestSites(site).Grasp(grasp).allData(:,4);

                    force1 = force1 - force1(1);
                    force2 = force2 - force2(1);

                    % Create a "graspInfo" object, store the displacement and force.
                    graspInfo.displacement = TestSites(site).Grasp(grasp).allData(:,8);
                    graspInfo.force = (force1 + force2) / 2;

%                     idx = find(graspInfo.force < 17);
                    idx = 1:1:length(graspInfo.force);

                    if (any(isnan(graspInfo.displacement)) || any(isnan(graspInfo.force)))
                        ab_fit = [-1 -1];
                        SSresid = -1;
                        quality = -1;
                    else
                        % Curve-fit the exponential equation to obtain alpha-beta values
                        ab_fit = ab_init;
                        SSresid = inf;
                        selectedSolver = length(solverStats);

                        curveFit_X = graspInfo.displacement(idx);
                        curveFit_Y = graspInfo.force(idx);

                        if fitType == 2
                            curveFit_X = curveFit_X .^ 2;
                        end

                        for i = 1:3     % Run the solvers multiple times
                            for j = 1:1:length(solvers)     % Iterate through the solvers
                                [ab_fit_test,SSresid_test,resid] = lsqcurvefit(F,ab_fit,   ↙
curveFit_X,curveFit_Y, [], [], solvers(j));
```

```
                            if SSresid_test < SSresid    % Check if current solution is better
                                ab_fit = ab_fit_test;        % Update best solution
                                SSresid = SSresid_test;      % SSE: Sum of squared error
                                selectedSolver = j;
                            end
                        end
                    end
                    solverStats(selectedSolver) = solverStats(selectedSolver)+1;

                    SStotal = sum((curveFit_Y - mean(curveFit_Y)).^2);    % SST: Sum of squared ↙
total

                    quality = 1 - SSresid/SStotal;    % "Pseudo" R-squared value. Normalized ↙
residual/error
                end

                % Store relevant data
                graspInfo.alpha = ab_fit(1);
                graspInfo.beta = ab_fit(2);
                graspInfo.AxB = ab_fit(1)*ab_fit(2);      % alpha times beta
                graspInfo.resNorm = SSresid;              % Sum of Squares of the residual
                graspInfo.quality = quality;              % "Quality" metric. Used to filter ↙
out "bad" grasps
%                 graspInfo.grasp = TestSites(site).Grasp(grasp).allData;

                % Store the Grasp Info into a new structure
                GraspData.(PID{Pig_Num}).TissueID(Tissue_Num).TestID(stage).TestSiteID(site).   ↙
Grasp(grasp) = graspInfo;
                clear graspInfo;
            end
            % Transfer the "test type" as well
            GraspData.(PID{Pig_Num}).TissueID(Tissue_Num).TestID(stage).TestSiteID(site).   ↙
testType = TestSites(site).testType;
            GraspData.(PID{Pig_Num}).TissueID(Tissue_Num).TestID(stage).TestSiteID(site).   ↙
Thickness = TestSites(site).Thickness;
            GraspData.(PID{Pig_Num}).TissueID(Tissue_Num).TestID(stage).TestSiteID(site).   ↙
notes = TestSites(site).notes;
            GraspData.(PID{Pig_Num}).TissueID(Tissue_Num).TestID(stage).TestSiteID(site).   ↙
location = TestSites(site).location;
        end
    end
end

GraspData.fitType = fitType;
GraspData.info = 'All Pigs, In-Vivo ONLY, Pucks ONLY, First Grasp ONLY' ;

disp('Processing complete!');

%%
disp('Saving...')
save('GraspData_new.mat','GraspData');
disp('Done!');
```

```
if ~exist('GraspData', 'var')
    fprintf('Loading Data... ');
    load('GraspData.mat');
    fprintf('Done\n');
end

PID = fieldnames(GraspData);

Tissue_Lookup = {
    'Liver'     % 1
    'Spleen'    % 2
    'IVC'       % 3
    'Aorta'     % 4
    'Lung'      % 5
    'Belly'     % 6
    'Esophagus' % 7
    ''
    'Calibration Puck'   % 9
};

Stage_Lookup = {
   'In-Vivo'             % 1
   'Postmortem'          % 2
   'Ex-Vivo'             % 3
   'Post-Refrigeration'  % 4
   'Post-Freeze'         % 5
};


% Configuration
des_force = 5;
TissueNum = 5;  % Which Tissue. Set to only a single tissue.

min_quality = 0.95; % Minimum fit quality


Plot_Position1 = [0 0 1200 700];
Plot_Position2 = [0 0 1200 700];
Plot_Position3 = [0 0 1200 500];


% Standard Letter Page: 8.5" x 11"
% 1 in margins on each side: 6.5" x 9"
FIG_W = 9.0;     % Image width
FIG_H = 6.5;     % Image height
FIG_UNITS = 'inches'; % units for W&H

% Plot_Position1 = [0 0 FIG_W FIG_H];


% Boxplot labels
boxlabels_Stages = { 'In-Vivo','Postmortem','Ex-Situ','Fridge','Freeze'};
% boxlabels_Pigs = {'Animal 1','Animal 2','Animal 3','Animal 4','Animal 5'};
% boxlabels_Pigs = {'Session 1','Session 2','Session 3','Session 4','Session 5'};
boxlabels_Pigs = { 'Sess.1','Sess.2','Sess.3','Sess.4','Sess.5'};

boxlabels = [boxlabels_Pigs boxlabels_Stages];
boxplot_spacing = [1:5 7:11];
```

```
% boxlabels = boxlabels_Pigs;
% boxplot_spacing = 1:5;


textbox_pos1 = [0.245,0.035,0.080,0.037];
textbox_txt1 = "In-Vivo ONLY";

textbox_pos2 = [0.674,0.035,0.073,0.037];
textbox_txt2 = "All Sessions";

close all;
% Collect data for box plots

Derivatives = [];
Thicknesses = [];
Qualities = [];

Avg_vels = [];
Max_vels = [];

groups = [];

for PigNum = 1:5

    if ~isa(GraspData.(PID{PigNum}), 'struct')
        continue;
    end

    fprintf('\nBegin Pig %i\n',PigNum);

    for StageNum = 1:length(GraspData.(PID{PigNum}).TissueID(TissueNum).TestID)
%     for StageNum = 1
        TestSites = GraspData.(PID{PigNum}).TissueID(TissueNum).TestID(StageNum).TestSiteID;
        for loc = 1:1:length(TestSites)
            curThickness = TestSites(loc).Thickness;
            for grasp = 1
%             for grasp = 1:1:length(TestSites(loc).Grasp)
%                 curGrasp = TestSites(loc).Grasp(grasp);
                curGrasp = TestSites(loc).Grasp;

                if (curGrasp.quality < min_quality)
%                     fprintf('Fit quality is less than threshold: %.3f\n', curGrasp.quality);
                    continue;
                end
%                 fprintf('T%i S%i L%i G%i: ',TissueNum,StageNum,loc,grasp);

                curForce = curGrasp.force;

                if (min(curForce) > -des_force)
%                     fprintf('Maximum force is less than desired force: %.2f\n',-min(curForce));
                    continue;
                end
                curDisp = curGrasp.displacement;

                alpha = curGrasp.alpha;
                beta = curGrasp.beta;
                AxB = curGrasp.AxB;
```

```
                [~,idx] = min(abs(curForce + des_force));    % Find the index where the force is ↙
closest to the desired force

                des_disp = curDisp(idx);     % (Nearest) Displacement at the desired force value

                % Calculate the derivative near the desired force
                if (GraspData.fitType == 2)
                    derivative = 2 * AxB * des_disp * exp(beta * des_disp.^2);     % Fit 2
                else
                    derivative = AxB * exp( beta * des_disp );      % Fit 1
                end


                if (derivative > 200)
%                     fprintf('!!!!!Found derivative = %.0f with P%i S%i L%i\n',derivative,  ↙
PigNum, StageNum, loc);
                end


                % Segmenting by ANIMAL:
                if (StageNum == 1)      % Show in-vivo ONLY when segmenting by animal
                    groups(end+1) = PigNum;
                    Derivatives(end+1) = derivative;
                    Thicknesses(end+1) = curThickness;
                    Qualities(end+1) = curGrasp.quality;
                end


                % Segmenting by STAGE:
                if (PigNum <= 5)    % Change to `== 5` if you want to only show pig 5, for ↙
example
                    groups(end+1) = 5+StageNum;   % Segment by stage
                    Derivatives(end+1) = derivative;
                    Thicknesses(end+1) = curThickness;
                    Qualities(end+1) = curGrasp.quality;
                end


%                 avgVel = -curDisp(end)/length(curDisp);
%                 maxVel = -min(diff(curDisp));
%
%                 Avg_vels = [Avg_vels avgVel];
%                 Max_vels = [Max_vels maxVel];

%                 fprintf('derivative %.3f\t avg vel %.4f\t max vel %.4f\n', derivative, avgVel,  ↙
maxVel);
            end
        end
    end
end


Derivatives_segmented = cell(1,length(boxlabels));
Thicknesses_segmented = Derivatives_segmented;
Qualities_segmented = Derivatives_segmented;
for ii = 1:length(groups)
```

```
    Derivatives_segmented{groups(ii)}(end+1) = Derivatives(ii);
    Thicknesses_segmented{groups(ii)}(end+1) = Thicknesses(ii);
    Qualities_segmented{groups(ii)}(end+1) = Qualities(ii);
end

% This part is necessary for when there isn't any data in a certain
% grouping and MATLAB complains about extra x-labels and stuff for a
% nonexistent grouping instead of being intelligent and ignoring it.
ii = 1;
while (ii <= length(boxplot_spacing))
    if (isempty (Derivatives_segmented{ii}))
        boxlabels(ii) = [];
        boxplot_spacing(ii) = [];
        Derivatives_segmented(ii) = [];
        Thicknesses_segmented(ii) = [];
        Qualities_segmented(ii) = [];
    else
        ii = ii+1;
    end
end



% NeedsTextBoxes = ~ ishandle(7);
NeedsTextBoxes = 1;


%%%%%%%%%%%% DERIVATIVES PLOTTING
fig = figure(7);

set(gcf, 'Position', Plot_Position1);



boxplot(Derivatives, groups, 'positions', boxplot_spacing, 'whisker', 9999);
title(['Derivatives of Force-Displacement Curves for ' , Tissue_Lookup{TissueNum}, ' at Force = ',↙
num2str(des_force), ' N']);
ylabel(['Stiffness [N] @ F=' ,num2str(des_force), 'N']);
set(gca,'xtick',boxplot_spacing,'xticklabel',boxlabels)

if (NeedsTextBoxes)
    annotation(fig, 'textbox', textbox_pos1, 'String', textbox_txt1, 'FitBoxToText', 'on')
    annotation(fig, 'textbox', textbox_pos2, 'String', textbox_txt2, 'FitBoxToText', 'on')
end

for ii = 1:length(boxlabels)
    txt_handle = text(boxplot_spacing(ii)-0.0, quantile(Derivatives_segmented{ii},0.25)+1 , ↙
sprintf('N=%i',length(Derivatives_segmented{ii})) , 'FontSize',9 );
    txt_handle.Rotation = 90;
end

yl = ylim;

text(1,yl(2)-5, sprintf( 'Q > %.2f',min_quality))
```

```
set(gcf,'PaperPositionMode','auto', 'units', FIG_UNITS)
Plot_Position1 = get(gcf, 'position');
Plot_Position1(3:end) = [FIG_W FIG_H];
set(gcf, 'position', Plot_Position1);
% tmptxt = 'Puck_F20_Stiff';    savefig(tmptxt);print(tmptxt, '-dpng')



%%%%%%%%%%%%%% THICKNESS PLOTTING


fig = figure(8);
set(gcf, 'Position', Plot_Position2);
boxplot(Thicknesses, groups, 'positions', boxplot_spacing, 'whisker', 9999);
title(['Initial Thicknesses of ', Tissue_Lookup{TissueNum}]);
ylabel('Thickness [mm]');

set(gca, 'xtick',boxplot_spacing, 'xticklabel',boxlabels, 'FontSize',8, 'XTickLabelRotation',↙
45 )

if (NeedsTextBoxes)
    annotation(fig, 'textbox', [0.223 0.0234 0.167 0.0625], 'String', textbox_txt1,↙
'FitBoxToText', 'on')
    annotation(fig, 'textbox', textbox_pos2, 'String', textbox_txt2, 'FitBoxToText', 'on')
end

for ii = 1:length(boxlabels)
    txt_handle = text(boxplot_spacing(ii)-0.0, quantile(Thicknesses_segmented{ii},0.25)+0.5 , ↙
sprintf('N=%i',length(Thicknesses_segmented{ii})) , 'FontSize',9 );
    txt_handle.Rotation = 90;
end

yl = ylim;

text(1,yl(2)-5, sprintf( 'Q > %.2f',min_quality))



set(gcf,'PaperPositionMode','auto', 'units', FIG_UNITS)
Plot_Position1 = get(gcf, 'position');
Plot_Position1(3:end) = [6.5 4.5];
set(gcf, 'position', Plot_Position1);

return


%%%%%%%%%%%%%% QUALITY PLOTTING

fig = figure(9);
set(gcf, 'Position', Plot_Position3);
boxplot(Qualities, groups, 'positions', boxplot_spacing, 'whisker', 9999);
title(['Curve-Fitting Quality for ', Tissue_Lookup{TissueNum}]);
ylabel('Normalized Residual');
set(gca,'xtick',boxplot_spacing,'xticklabel',boxlabels)
if (NeedsTextBoxes)
    annotation(fig, 'textbox', textbox_pos1, 'String', textbox_txt1, 'FitBoxToText', 'on')
```

```
    annotation(fig, 'textbox', textbox_pos2, 'String', textbox_txt2, 'FitBoxToText', 'on')
end
```

```
function CalCoeffs = calibrateLoadCell(calFolder)

CalCoeffs = [];

internalFolders = [{ 'zip_tie'} {'10'} {'20_0'} {'20_1'} {'50'} {'100'} {'200_0'} {'200_1'}
{'500'}];
mass = [0 10 20 20 50 100 200 200 500];      % grams
LoadData01 = [];
LoadData02 = [];


for j=1:1:size(internalFolders,2)
    massFolder = dir(strcat(calFolder, '/adcbottom/',internalFolders{j}));
    for k = 1:1:size(massFolder,1)
        if contains(massFolder(k).name, 'adc')
            LoadCellFilename =  strcat(strcat(massFolder(1).folder,  '/'),massFolder(k).name);
            LoadCellFile = load(LoadCellFilename);
            LoadData01 = [LoadData01; LoadCellFile(:,2), -(mass(j)./1000).*9.80665.*ones(length
(LoadCellFile(:,2)),1)];
        end
    end
end


for j=1:1:size(internalFolders,2)
    massFolder = dir(strcat(calFolder, '/adctop/',internalFolders{j}));
    for k = 1:1:size(massFolder,1)
        if contains(massFolder(k).name, 'adc')
            LoadCellFilename =  strcat(strcat(massFolder(1).folder,  '/'),massFolder(k).name);
            LoadCellFile = load(LoadCellFilename);
            LoadData02 = [LoadData02; LoadCellFile(:,3), -(mass(j)./1000).*9.80665.*ones(length
(LoadCellFile(:,3)),1)];
        end
    end
end


fit01 = polyfit(LoadData01(:,1),LoadData01(:,2),1);
val01 = polyval(fit01,LoadData01(:,1));

fit02 = polyfit(LoadData02(:,1),LoadData02(:,2),1);
val02 = polyval(fit02,LoadData02(:,1));

% figure;
% subplot(1,2,1)
% plot(LoadData01(:,1), LoadData01(:,2),'r.')
% hold on
% plot(LoadData01(:,1),val01)
%
% subplot(1,2,2)
% plot(LoadData02(:,1), LoadData02(:,2),'b.')
% hold on
% plot(LoadData02(:,1),val02)

CalCoeffs(:,:,1) = fit01;
CalCoeffs(:,:,2) = fit02;

end
```

```
function Sample = ProcessData(Sample,params)

RunningAvgSize=params.RunningAvgSize;

% Thresholds for data segmentation
LowDiffThreshold=params.LowDiffThreshold; %0.00001;
LowForceThreshold=params.LowForceThreshold;
DisplacementThreshold = params.DisplacementThreshold;  % (Encoder Value)

% Radius of scissors (used for determining strain)
r=params.r; % mm

encoderOffset =params.encoderOffset; %-956;

% Account for force offset
Sample.syncedData(:,3) = Sample.syncedData(:,3) - max(Sample.syncedData(1,3));
Sample.syncedData(:,4) = Sample.syncedData(:,4) - max(Sample.syncedData(1,4));

% Find theta (in degrees) from encoder values.
Sample.syncedData(:,5) =   (360/8192)*(Sample.syncedData(:,2) - encoderOffset);

% Find distance between jaws
Sample.syncedData(:,6) = 2*r.*sind(Sample.syncedData(:,5));

% Smooth/average force measurements to find regions where values
% measurements are increasing/decreasing
Sample.meanForce01 = movmean(Sample.syncedData(:,3),RunningAvgSize);
Sample.meanForce01 = movmean(Sample.meanForce01,RunningAvgSize);
Sample.meanForce01 = movmean(Sample.meanForce01,RunningAvgSize);
Sample.meanforceDiff01 = diff(Sample.meanForce01);
Sample.meanforceDiff01(end+1) = Sample.meanforceDiff01(end);

Sample.meanForce02 = movmean(Sample.syncedData(:,4),RunningAvgSize);
Sample.meanForce02 = movmean(Sample.meanForce02,RunningAvgSize);
Sample.meanForce02 = movmean(Sample.meanForce02,RunningAvgSize);
Sample.meanforceDiff02 = diff(Sample.meanForce02);
Sample.meanforceDiff02(end+1) = Sample.meanforceDiff02(end);

% Find index values corresponding to each of the threshold values
zeroIndex01=[]; zeroIndex02 = []; zeroIndex03 = [];
zeroIndex04=[]; zeroIndex05 = []; zeroIndex06 = [];
zeroIndex = [];
zeroIndex01 = find(abs(Sample.meanforceDiff01)>LowDiffThreshold);
zeroIndex02 = find(abs(Sample.meanforceDiff02)>LowDiffThreshold);
zeroIndex03 = find(Sample.meanForce01<LowForceThreshold);
zeroIndex04 = find(Sample.meanForce02<LowForceThreshold);
zeroIndex05 = find(Sample.syncedData(:,2)<DisplacementThreshold);
%             zeroIndex02 = zeroIndex01;
% Find indices which satisfy each of the threshold criteria
zeroIndex = intersect(intersect(zeroIndex01, zeroIndex02), intersect(intersect(zeroIndex03,  ↙
zeroIndex04), zeroIndex05));


% Find regions where force values are decreasing (corresponds to
% jaws closing)
indexClose01 = []; indexClose02 = []; indexClose = [];
indexClose01 = find(Sample.meanforceDiff01<-10^(-4) );
indexClose02 = find(Sample.meanforceDiff02<-10^(-4));
```

```
indexClose = intersect(indexClose01, indexClose02);

% Find indices which satisfy threshold criteria as well as jaws
% closing criteria
indexClose = intersect(indexClose, zeroIndex);


% Segment into grasps
EndGraspIndexTemp = []; StartGraspIndexTemp = [];
StartGraspIndexTemp = [1; find(diff(indexClose)>1)+1]';
EndGraspIndexTemp = [find(diff(indexClose)>1); length(indexClose)]';
NumberOfGrasps = length(StartGraspIndexTemp);


% Fill in broken regions

% If the previous region ended within 1000 ms of the beginning of the current region,
% we consider both regions to be part of a single grasp. This is to prevent two parts
% of the same grasp to be considered as two different grasps, because there may be a
% short period where force stops increasing during the grasp.
% Fortunately, we know that there is approximately 1.7 seconds between grasps due to
% the beeper, which implements a 1 second delay between grasp cycles, followed by a
% 200 ms beep, and another 500 ms delay to move into position. The grasp tone itself
% beeps for 1.5 seconds.

fillBroken = 1; % Continuously run the check until no more broken regions are detected
% It is possible that a "sub-broken" region may not actually have greater force, but a
% subsequent region it is combined with does. Thus, it will show up as a
% separate grasp. Re-running it helps combine the sub-broken regions.
while fillBroken
    RegionIndex = [];
    for i=1:1:(NumberOfGrasps-1)
        if(indexClose(StartGraspIndexTemp(i+1)) - indexClose(EndGraspIndexTemp(i)) < 1000 )
            minForce1 = min(Sample.syncedData(indexClose(StartGraspIndexTemp(i)):indexClose
(EndGraspIndexTemp(i)),3));
            minForce2 = min(Sample.syncedData(indexClose(StartGraspIndexTemp(i+1)):indexClose
(EndGraspIndexTemp(i+1)),3));
            if minForce2 < minForce1
                RegionIndex = [RegionIndex i];
            end
        end
    end
    StartGraspIndexTemp(RegionIndex+1) = [];
    EndGraspIndexTemp(RegionIndex) = [];
    NumberOfGrasps = length(StartGraspIndexTemp);

    if isempty(RegionIndex)
        fillBroken = 0;
    end
end


% Cut out short, irrelevant grasps, only if the change in force is small
RegionIndex = [];
for i=1:1:(NumberOfGrasps)
    if length(EndGraspIndexTemp)>=i && length(StartGraspIndexTemp)>=i
        if length(indexClose)>=EndGraspIndexTemp(i) && length(indexClose)>=StartGraspIndexTemp(i)
```

```
            if(indexClose(EndGraspIndexTemp(i)) - indexClose(StartGraspIndexTemp(i)) < 750 )
                forceRange = range(Sample.syncedData(indexClose(StartGraspIndexTemp(i)):
indexClose(EndGraspIndexTemp(i)),3));
                if forceRange < 3    % 3 Newton force delta
                    RegionIndex = [RegionIndex i];
                end
            end
        end
    end
end

% Combine results of previous segments of code
StartGraspIndexTemp(RegionIndex) = [];
EndGraspIndexTemp(RegionIndex) = [];
NumberOfGrasps = length(StartGraspIndexTemp);
indexCloseTemp = [];

StartGraspIndex = [];
EndGraspIndex = [];
for i=1:1:length(StartGraspIndexTemp)
    indexCloseTemp = [indexCloseTemp indexClose(StartGraspIndexTemp(i)):1:indexClose
(EndGraspIndexTemp(i))];
    StartGraspIndex(i) = find(indexCloseTemp == indexClose(StartGraspIndexTemp(i)));
    EndGraspIndex(i) = find(indexCloseTemp == indexClose(EndGraspIndexTemp(i)));
end

indexClose = indexCloseTemp;


% Find regions where force values are increasing (corresponds to
% jaws closing)
indexOpen01 = find(Sample.meanforceDiff01>0 );
indexOpen02 = find(Sample.meanforceDiff02>0 );
indexOpen = intersect(indexOpen01, indexOpen02);

% Find indices which satisfy threshold criteria as well as jaws
% closing criteria
indexOpen = intersect(indexOpen, zeroIndex);

% Divide data into "close" and "open" segments
Sample.syncedClose = Sample.syncedData(indexClose,:);
Sample.syncedOpen = Sample.syncedData(indexOpen,:);


Sample.StartGrasp = Sample.syncedClose(StartGraspIndex,:);
Sample.EndGrasp = Sample.syncedClose(EndGraspIndex,:);
%
OldDisplacement =0;

% Determine "strain" for each segment, and append to data structure
for i=1:1:length(indexClose)
    for j=1:1:NumberOfGrasps
        if(indexClose(i)>indexClose(StartGraspIndex(j)) && indexClose(i)<(indexClose
(EndGraspIndex(j))+1))
            GraspIndexRange = (StartGraspIndex(j):(EndGraspIndex(j)))';
            InitialDisplacement = max(Sample.syncedClose(GraspIndexRange,6));
            if OldDisplacement~=max(Sample.syncedClose(GraspIndexRange,2))
```

```
                OldDisplacement = max(Sample.syncedClose(GraspIndexRange,2));
            end
            Sample.syncedClose(GraspIndexRange,7) = (Sample.syncedClose(GraspIndexRange,6)-
InitialDisplacement);
            Sample.syncedClose(GraspIndexRange,8) = (Sample.syncedClose(GraspIndexRange,6)-
InitialDisplacement)./InitialDisplacement;
        end
    end
end

if NumberOfGrasps>=1
    for i=1:1:NumberOfGrasps
        GraspIndexRange = (StartGraspIndex(i):(EndGraspIndex(i)))';
        Sample.Grasp(i).allData = Sample.syncedClose(GraspIndexRange,:);
    end
    % Thickness is the distance between jaws at first contact
    Sample.Thickness = Sample.Grasp(1).allData(1,6);
else
    Sample.Grasp.allData = [];
end


end
```

Appendix D:

DMD 2020 Conference Paper Accepted: “”

Proceedings of the 2020 Design of Medical Devices Conference
DMD2020
April 6, 7-9, 2020, Minneapolis, MN, USA

DMD2020-10082

DESIGN OF A HANDHELD TISSUE GRASPING DEVICE TO MEASURE TISSUE MECHANICAL PROPERTIES IN-VIVO OR IN A LABORATORY SETTING

Bradley Drahos[α], Amer Safdari MS[β], Faizan Malik[α], Rebecca Smith[α], Matt Kubala[α], Jack Norfleet [γ], Conner Parsey[γ]
Shikha Goodwin PhD, Timothy M. Kowalewski PhD[1]

Medical Robotics and Devices Lab
Department of Mechanical Engineering[α], Biomedical Engineering[β]
University of Minnesota
Minneapolis, Minnesota 55455

U.S. Army Futures Command[γ]
CCDC Soldier Center Simulation and Training Technology Center
12423 Research Parkway
Orlando, Florida 32826

ABSTRACT

With medical institutions increasing the use of medical simulators for educational purposes it is detrimental that the knowledge gap regarding tissue mechanical properties be researched further in depth. The grasper device discussed throughout this paper aims to provide researchers a handheld device capable of testing soft organs and tissue in-vivo and ex-situ in a laboratory setting. The device consists of two load cells on the inner jaws of the grasper to measure compressive force and an encoder to monitor the graspers angular position which yields tissue position and strain. Accompanying the grasper is a GUI written in Rust which is capable of data file management, and providing a 10 second live feed of load cell and encoder readings. The grasper device is currently being employed in a study testing the tissue mechanical response of porcine tissue at states ranging from in-vivo to ex-situ post freeze. The results from this test, and subsequent tests using the grasper have the capability of providing much needed knowledge regarding tissue mechanical properties to improve medical simulators and medical education as a whole.

[1]Contact author: timk@umn.edu.

INTRODUCTION

The Association of American Medical College (AAMC) estimates that there will be a shortage of over 100,000 physicians in the United States by 2032 [1]. One method to combat this physician shortage is through improving medical education. With medical institutions across the globe pushing towards the use of medical simulators for education, it is imperative that the next generation of medical simulators are as accurate to human tissue as possible. More accurate medical simulators will minimize the adjustment needed for new physicians to transfer their training from medical simulators to the patient. One common issue with current generation medical simulators is that they are based on properties of pre-conditioned tissue in a laboratory setting rather than in-vivo tissue from a living human being. Even though there have been numerous studies conducted in the past aiming to bridge the gap between tissue mechanical properties from the tissue of living patients to preconditioned tissue in a laboratory setting, there is still a need in the medical device field for a device capable of testing basic tissue mechanical response across various settings including in-vivo conditions.

One of the first comparisons between in-vivo and ex-vivo tissue was conducted by Rosen et al. [2]. Through their work, they

developed and tested a handheld grasping device that recorded the stress and strain applied to a small tissue sample. The primary drawback to this device is that by measuring stress and strain directly, the information that can be derived from testing results is limited to strictly stress and strain over time. Years later subsequent studies were conducted using an indentation instrument to characterize tissue mechanical properties laparoscopically by Samur et al. and Lim et al. [3,4]. While these studies were successful in fitting resulting stress strain relationships for tissue, the tissues used in these studies were against a hard surface (such as bone). A hard surface behind the tissue is necessary for such an indentation style device; however, there are various tissues where a rigid surface will not be available thus a more generally applicable design is necessary. While the current body of literature has explored a vast set of topics for various applications, there is still a need for a small handheld device capable of testing tissue basic mechanical properties across various settings [5,6].

This paper demonstrates the design of a grasper device that is capable of monitoring basic tissue mechanical response through the use of applied compressive forces via the grasper jaws and encoder position of the grasper jaws. This device allows for tissue mechanical responses to be measured in settings previously impossible based on current tools and devices.

METHODS

DESIGN REQUIREMENTS The design requirements for this device were primarily driven by the need for a testing methodology that would yield consistent results from an in-vivo setting to ex-situ laboratory settings. Common mechanical testing methodologies such as tension (uniaxial/biaxial) and compression (free/confined), while incredibly robust and ubiquitous, were deemed unsuitable due to the difficulty of being able to implement and replicate the exact same testing conditions on the lab bench as well as in-vivo. With suitability for in-vivo testing as the primary criteria in mind, it was important to ensure all necessary design requirements were met.

The primary design requirement for the grasper device was that it had to be capable of recording the compressive force applied (within +/- 5mN) as well as the displacement over time of the grasper jaws contact with the tissue. It was necessary that each of these data streams (load cell voltages and encoder position) were sampled at a high enough rate to ensure the entire grasping motion is covered in enough detail for analysis (sampling frequency of 1kHz). It was necessary to ensure that the data streams could be monitored live during testing while also being recorded in a usable state. With these data acquisition requirements met, it was necessary to then make the device realistically usable for the desired cases.

The ultimate goal of the device was for it to be capable of being used to test organ samples from a state in-vivo to a state ex-vivo in a laboratory setting. To make this possible, the device had to be handheld such that it could allow for limited nearby tissue damage in order to access target tissue access during in-vivo testing while still allowing the user to manipulate the tissue in tight spaces. In addition to being accessible for in-vivo usage and in a laboratory setting, it was necessary for the device and all auxiliary components to be capable of being shielded from tissue matter or be reasonably distant from any experiment such that there is no concern regarding sterilization or contamination. The final design requirement also came down to accessibility, such that testing of a single organ at a single state (in-vivo or ex-vivo) would take no longer than 20 minutes. Any longer and the device would no longer be viable for testing multiple tissues across multiple states for various animals. The final grasper device was designed with all of these design considerations in mind, and was capable of satisfying each of the necessary design requirements.

FIGURE 1: Grasper device hardware including load cells on jaw ends, hemispherical attachments near jaw tips, and encoder attached to metal body

HARDWARE A custom aluminum handheld grasping device, loosely intended to mimic a pair of surgical forceps, was instrumented with two 5 kg load cells (TAL220B, HT Sensor Technology) along each end, with 10 mm hemispherical attachments serving as the end effectors for tissue contact. Hemispherical attachments were chosen because it allows the grasping experiment to be comparable to indentation of an infinite half-space under certain conditions. The grasper device, load cells, hemispherical attachments, and encoder is shown in Figure 1. Each load cell was connected to a signal conditioner (IAA100, Futek Inc). The grasper jaw angle was measured using a quadrature rotary encoder with index pulse functionality (AMT102-V, CUI Inc) with a resolution of 0.7°. Both sets of instrumentation were connected to a NI PCIe 6320 DAQ card; data was acquired at 1 kHz using the NI-DAQMX C API on a Linux based PC. Data acquisition of force and jaw angle was controlled and monitored via a custom GUI display which is described in the following section.

FIGURE 2: Graphical user interface used during data acquisition for file selection, titling, and providing live feedback of load cell and encoder readings

SOFTWARE In an effort to make the entire data acquisition pipeline more robust, and enable higher throughput for future projects that wish to use this platform, National Instrument DAQ cards were selected to interface with the grasper. National Instruments provides a C-based API to communicate with their various DAQ hardware. Instead of writing the data acquisition console directly in C, a Rust wrapper over the NI-DAQMX C API was created and used to communicate with the DAQ hardware. The Rust-based application then outputs the analog voltage readings, as well as the encoder data into separate CSV files in a folder specified by the user. The indices of the voltage and encoder data are synchronized down to the millisecond.

The user interface is a simple web application written in HTML/CSS (with theming provided by Bootstrap 4.0) which displays the voltage and force data in a live graph, by communicating with the Rust backend.

APPLICATION The grasper device is currently being employed for in-vivo through ex-situ post freeze testing on porcine liver, spleen, peritoneum, and lung tissue samples, shown in Figure 3. The purpose of this study is to quantify the discrepancy of tissue mechanical properties from porcine tissue between in-vivo, ex-vivo in-situ, ex-vivo ex-situ, ex-situ post refrigeration, and ex-vivo post freeze states [7]. The final deliverables

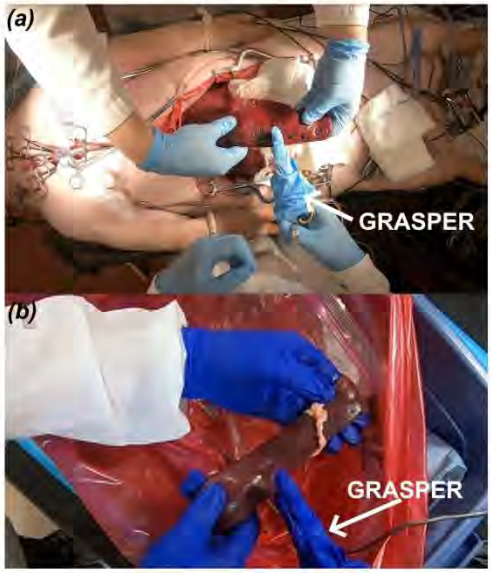


FIGURE 3: Head mounted GoPro footage of porcine spleen grasping tests: (a) in-vivo and (b) ex-situ post-freeze

of the study include the generation of a decay curve for tissue mechanical properties throughout these five states, the effects of grasping a single location on an organ multiple times on tissue mechanical properties, and determining variation of a single organs mechanical tissue properties in-vivo. Ultimately the results from this study will be used for the creation of more accurate medical simulators, specifically for combat medic training. The grasper was initially designed for this specific study and use; however, it was designed such that it is not limited to this study and can be used for various other applications.

For a single organ, a total of 10 individual locations on the organ are grasped a total of 10 times each. Prior to conducting any grasps on the tissue, it was necessary to conduct a voltage to force calibration. To do so, various loads of known weights, 10g up to 500g, were applied to each load cell jaw of the grasper. For each applied calibration load the readings from the load cell were recorded for 3 seconds. This calibration data was later used to determine the load cell voltage to the applied force correlation. This correlation is dependent of ambient temperature, thus it was necessary to complete this calibration before and after every testing session in a single location. Once calibration was complete, an additional calibration set of grasps was recommended on a

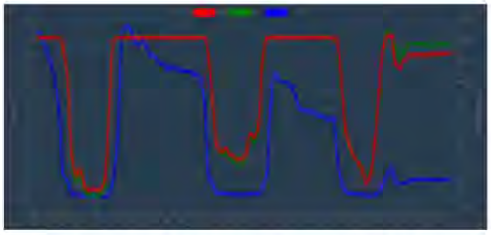


FIGURE 4: Upper jaw load cell (red), lower jaw load cell (green), and encoder (blue) live feed recordings for three tissue grasps

synthetic puck to ensure encoder and load cells are reading appropriate values. The load cells should each read between -2V and -7V while the encoder values should range from -1500 ticks to 1000 ticks (170° range of motion). It is recommended to periodically conduct synthetic puck calibration grasps in case of potential voltage drift during testing.

Upon completion of the synthetic puck grasps, tissue grasping could commence. Prior to starting the data acquisition process the person grasping the tissue moved into position near the tissue. At this point the person controlling the computer initiated the data acquisition process. A tone repeated three times where the grasper was to be fully opened and closed three times. This was to ensure the encoder index pulse was reached. After the three tones a short break was allowed for the grasper to move into position. For a single grasp a short tone was played to prime the grasper followed by a longer tone where the tissue was to be grasped. The tissue was grasped with as much force as possible without causing any tissue damage or puncturing. Once the desired number of grasps were completed, the individual controlling the computer stopped the data acquisition process. A screenshot of the load cell and encoder live readings is presented in Figure 4. Once the desired number of grasps were completed, the individual controlling the computer stopped the data acquisition process. A screenshot of the load cell and encoder live readings is presented in Figure 5. From this relationship it was clear that a discrepancy exists between liver basic mechanical response in terms of a force-strain relationship from an in-vivo state to ex-stiu state. At this point it was necessary to conduct the set of calibration weights to verify the force voltage relationship determined prior to tissue grasping.

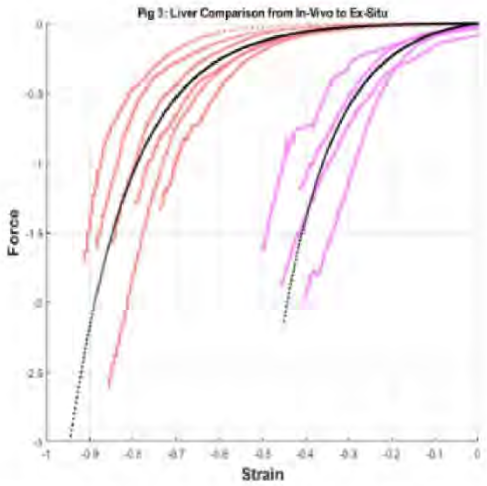


FIGURE 5: Grasper compressive force strain relationship on porcine liver tissue in-vivo (red) comparative to ex-situ (pink)

RESULTS

Prior to implementing the grasper device in the study previously discussed, it was necessary to first validate the graspers ability to monitor and record applied tissue forces and angular position of the encoder during the grasping process. In order to validate the grasper device's performance a silicon puck was printed and was used as a control for grasping. This silicon puck was grasped 10 times in succession, just as any tissue would be treated in a study. The ultimate goal of this control testing was to validate the grasper device via the generation of a force-strain relationship for the silicon puck.

The resulting force-strain relationship for the silicon puck control testing is shown in Figure 6. In the first plot of force (N) with respect to time (s) each of the 10 grasps are clearly recognized via the green overlay without encompassing the release period of the grasp or time between grasps. The encoder ticks with respect to time during the grasping process fell within the desirable range of encoder values while presenting the local minimums during peak grasping force. The most important result was the final plot of the force-strain relationship for the silicon puck. Since the silicon puck has consistent steady mechanical properties, the force-strain relationship remains consistent between grasps. This consistent force-strain relationship was monitored, thus further validating the performance of the grasper device.

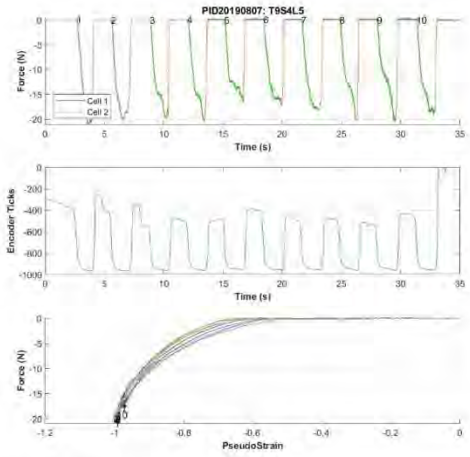


FIGURE 6: Calibration puck grasp force and encoder results over time, and resulting force-strain relationship for each grasp

CONCLUSION

This paper explored the design and application of a new handheld tissue grasping device for measuring tissue mechanical properties. The purpose of this device is to allow for in-vivo, ex-vivo, in-situ, and ex-situ testing of soft organs to determine tissue mechanical properties and create a force to pseudo-strain relationship for the tissue. This device aims to provide researchers a device capable of providing freedom for future biomechanics studies in-vivo through ex-situ.

The device itself consists of a scissors-like grasper with load cells on both jaws such that closing of the grasper on tissue contact is made solely on the load cells. Also present on the grasper is an encoder capable of monitoring the angle of the grasper at any point which can be used to determine the thickness of the tissue at any moment. Also presented is the necessary data acquisition hardware (NI PCIe 6320 DAQ card) with accompanying software to carry out data acquisition with the grasper. This software is accompanied by a pre-built GUI and data acquisition/storage capabilities. With the grasper device and accompanying software, the grasper device can be implemented and basic tissue mechanical properties can be monitored and recorded in future studies.

The grasper device is currently being implemented to test basic mechanical properties of porcine tissue in-vivo to ex-situ post freeze. The purpose of this study is to construct a decay curve of various porcine tissue quantifying the discrepancies in basic tissue mechanical properties in-vivo, ex-vivo, ex-situ, post refrigeration, and post freeze. While the grasper is proving successful in monitoring basic tissue mechanical properties and responses due to applied compressive loads, the device has been designed to be adopted in future studies on both animal and human tissue.

The ultimate purpose of this device is to aid in bridging the knowledge gap of tissue mechanical properties in-vivo and ex-situ in a laboratory setting. This knowledge can then be used to improve the accuracy of next generation medical simulators, primarily for use by combat medics.

ACKNOWLEDGMENTS

The authors would like to thank the U.S. Army Futures Command (Department of Defense) for sponsoring the design of the grasper and current porcine tissue grasping study. We would also like to thank Paul Iazzo, Tinen Isles, and all of the staff at the Visible Heart Lab (VHL) at the University of Minnesota for providing grasper design input and all animal tissue used to verify the graspers performance and for the current porcine tissue grasping study. Without the funding provided by the U.S. Army Research Laboratory and assistance provided by the Visible Heart Lab this device would not have been possible to design, construct, and implement.

REFERENCES

[1] Dall, T., Reynolds, R., Consultant, S., and Jones, K., 2019. The Complexities of Physician Supply and Demand. Tech. rep.

[2] Rosen, J., Brown, J. D., De, S., Sinanan, M., and Hannaford, B., 2008. "Biomechanical Properties of Abdominal Organs In Vivo and Postmortem Under Compression Loads". *Journal of Biomechanical Engineering,* ***130***.

[3] Samur, E., Sedef, M., Basdogan, C., Avtan, L., and Duzgun, O., 2007. "A robotic indenter for minimally invasive measurement and characterization of soft tissue response". *Medical Image Analysis,* ***11***(4), pp. 361–373.

[4] Lim, Y. J., Deo, D., Singh, T. P., Jones, D. B., and De, S., 2009. "In situ measurement and modeling of biomechanical response of human cadaveric soft tissues for physics-based surgical simulation". *Surgical Endoscopy and Other Interventional Techniques,* ***23***(6), pp. 1298–1307.

[5] Brouwer, I., Ustin, J., Bentley, L., Sherman, A., Dhruv, N., and Tendick, F., 2001. "Measuring In Vivo Animal Soft Tissue Properties for Haptic Modeling in Surgical Simulation". *Medicine Meets Virtual Reality*.

[6] Ocal, S., 2010. "Effect of Preservation Period on the Viscoelastic Material Properties of Soft Tissues With Implications for Liver Transplantation". *Journal of Biomechanical Engineering,* ***132***(10), p. 101007.

[7] Safdari, A., 2019. "Towards More Accurate Medical Simulation via Procedural Instrumentation". Master's thesis, University of Minnesota, Minneapolis, MN.

Appendix E:

MHSRS 2020 Conference (cancelled) Abstract Accepted:

Abstract Title (255 Chars):
Determining Potential Changes in Tissue Mechanical Properties of Porcine Liver and Spleen from In-Vivo to a Laboratory Setting Post-Refrigeration and Post-Freeze to Facilitate More Accurate Medical Simulators

Abstract (8000 Chars):

Military medical personnel master critical lifesaving skills on simulators and live tissue. The continued use of live tissue is an indictment of the realism and effectiveness of current simulation technologies. To reduce reliance on live tissue, the DoD has invested heavily to better understand and apply the properties and behaviors of the tissues and structures being simulated. A majority of this research has been and is currently based on preconditioned deceased tissue in a laboratory setting with the assumption that the tissue's mechanical properties are similar to living (in-vivo) conditions. This assumption is fairly common throughout medicine; however, it is important to test that assumption by comparing the mechanical responses of various tissues from in-vivo to ex-situ.

This paper discusses a study in which a handheld, scissors-like grasper device capable of measuring applied forces and tissue thickness, via angular displacement grasper jaws, was implemented to obtain stress-strain relationships of porcine liver and spleen at various tissue states. The tissue states tested in this study include an in-vivo state, a postmortem state, ex-situ immediately after removing the organs from the body, post tissue refrigeration, and post tissue freeze. At each state the tissue was grasped at various locations throughout the organ, including both previously grasped and undisturbed locations.

The force and angle readings were used to create a stress-strain relationship for the first grasp at each location on the tissue. A curve-fitting approach was applied to these stress strain relationships using the exponential equation:

$$F(t)=*([*(t)]-1)$$

Fitted alpha and beta values were used to compare basic tissue mechanical properties. A one-way anova test was run to compare the product of the resulting alpha and beta values from the curve fitting functions between tissue states for both the porcine liver and spleen. The anova analyses showed that there was no significant difference (smallest p value = 0.5413) between the mean product of the alpha and beta constants across the five states for porcine liver. There was also no significant difference, but at a lower confidence (smallest p value = 0.0839), between the alpha/beta product for the porcine spleen.

The results from these tests detected no significant difference in mean stress strain relationship between a porcine liver and spleen tissue from in-vivo through ex-vivo to a post-freeze state This supports the initial assumption that tissue mechanical properties do not change drastically from a living state to a dead state. This also strengthens arguments that medical simulation materials based on cadaveric tissue measures can realistically simulate live biologic tissues. What has not been considered thus far is the variance in tissue mechanical properties between states. This study analyzed the mean tissue mechanical property values and variance will be analyzed at a later date.

Abstract Disclaimer (350 Chars):

Only one method of data analysis has been implemented via the curve fitting approach. An additional approach makes use of required force to achieve various levels of strain and will be implemented for the final publication presentation.

Learning Objectives:

1. Understand that live and cadaveric (in-vivo and ex-vivo) tissues may feel differently (exhibit different mechanical responses) and that such differences are tissue-specific and potential shortcomings of current medical simulators as a result.
2. Gain insight into whether the tissues tested (animal liver and spleen) do or do not change between live and cadaveric states.
3. Describe the assumption that living tissue has similar tissue mechanical properties to deceased pre-conditioned tissue was verified from this study

Appendix F:

Publication in Progress:

Porcine Organ Tissue Property Variability from In-vivo to Ex-Situ post Freeze

Aimed towards medical simulators accuracy and how they compare to the real thing

- Target publications:
  - PLOS
  - TBME
  - JBE (JBME)
  - Journal Med Devices
- Authors
  - Faizan Malik
  - Brad Drahos
  - Amer Safdari
  - Rebecca SMith
  - Matt Kubala
  - Victor Barocas
  - Mark Mazzeo
  - Jack Norfleet
  - Rob Sweet
  - Tim Kowalewski

- Abstract
  - ***Why it's important***
    - ***Including Previous literature***
    - ***Initial idea that tissue properties change from in-vivo to ex-situ***
    - ***Goal to describe/quantify how tissue properties change***
  - ***Mention how liver compares to literature***
  - ***Sentence on each tissue***

  Keywords -
- Introduction
  - Mechanical Response
    - What is this? Stress and strain - When the force is applied to a tissue, the tissue response is measured in the form of strain. This is important as it helps us determine how the tissue will behave when dealt with.
    - How do we define it?
  - Porcine Tissue
    - Why porcine tissue?
      - Look into this. Conner might have answers
      - Air force prefers this
    - Rough anatomy?
    - How does it compare to humans
    - History of using porcine tissue
  - *Analysis and result details-*

- *How to select alpha and beta*

- Previous work and knowledge
  - Amers thesis
    - Have amer write a short abstract of his thesis
  - Previous study on liver
    - Found difference in liver mechanical properties from in-vivo to ex-vivo

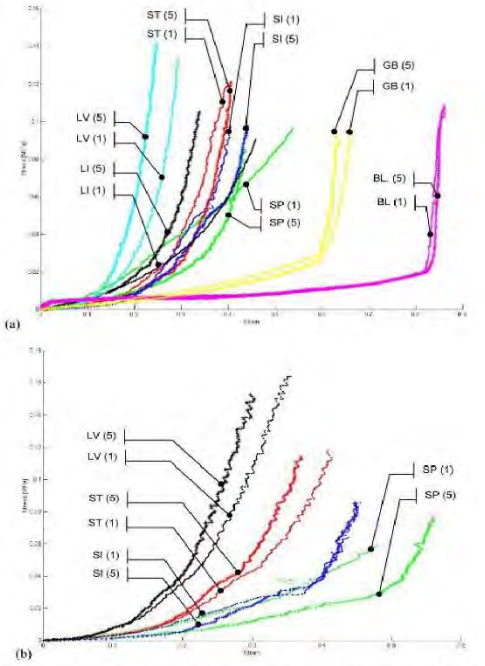
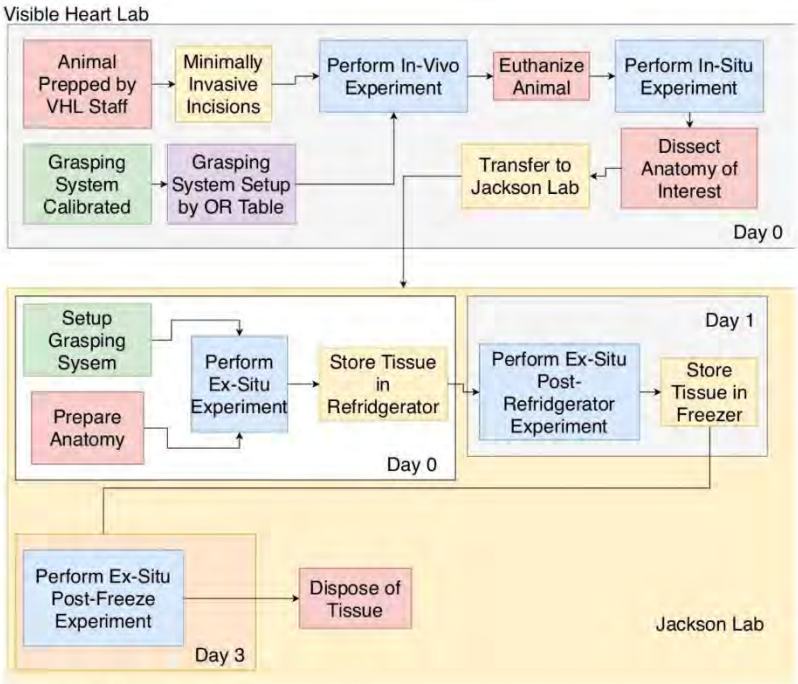


Fig. 3 Example stress-strain curves for all organs under study, as measured with the MEG at 5.4 mm/s loading velocity (first and fifth cycles shown): (*a*) in vivo and (*b*) ex corpus. Organs' legends: BL=bladder, GB=gallbladder, LI=large intestine, LV=liver, SI=small intestine, SP =spleen, and ST=stomach. The loading cycle number (1 or 5) is defined in the brackets.

FIGURE LEGEND - What colors represent ?? and what is x axis and y axis

Methods

Tissue Handling



In-Vivo and Postmortem

The animal is sedated and laid on its back. An incision is made starting in the abdominal cavity (to access the Liver, Spleen, and "Belly" tissues), and then later expanded to the chest cavity (to access Lung and Aorta). The body temperature of the animal is measured via a rectal thermometer and recorded.

After completing the data collection process on the desired tissues, the animal is euthanized, and the animal's heart is extracted. We continue with data collection immediately after euthanasia, on the same tissues as tested during *in-vivo* conditions.

Ex-Vivo, Refrigeration, and Freezing

After Postmortem testing, we extract the tissues from the carcass and place each tissue in a plastic bag with 25 mL of 1x PBS/pen-strep antibiotic solution. The liver is placed in a bag with 75 mL of solution due to its larger size.

The plastic bags containing the tissues are placed in water baths at 37.5 degrees Celsius until data collection for the *ex-vivo* stage. After *ex-vivo* data collection, the tissues are replaced in their respective plastic bags (still containing the antibiotic solution) and then placed into a refrigerator maintaining approximately 4 degrees Celsius.

After approximately 24 hours of refrigeration, the tissues are removed from the refrigerator and re-heated in a water bath at 37.5 to 38 degrees C for 2-4 hours. Again, the tissues remain in their plastic bags with the antibiotic solution, until they are temporarily removed for data collection. Once data collection is completed, the tissues are replaced in plastic bags. The tissue may be placed in a new plastic bag with more antibiotic solution if the previous bag was saturated with fluids from the tissue.

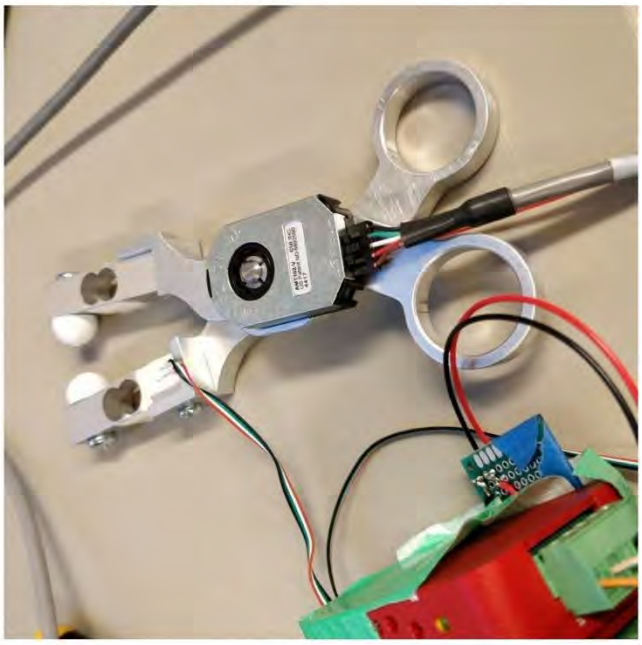
The tissue is placed in a freezer at -18 degrees C for approximately 48 hours, after which the tissues are removed and placed in a water bath at 38 degrees C for 4-5 hours. Data is collected from the tissues for a final time, after which the tissues are disposed.

- Device (reference DMD paper)
  - Discussion of indenter then move to grasper
  - - indenter type design

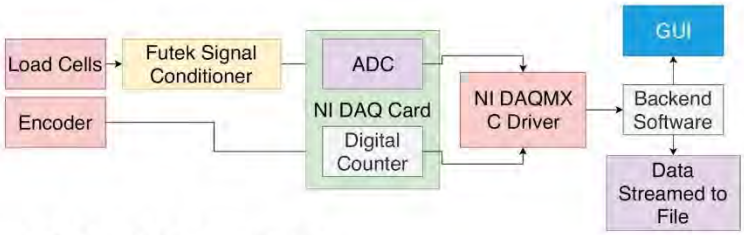
Grasping Device Design

The grasping device used is the device detailed in "DESIGN OF A HANDHELD TISSUE GRASPING DEVICE TO MEASURE TISSUE MECHANICAL PROPERTIES IN-VIVO OR IN A LABORATORY SETTING" by Drahos, et al. It contains two opposing load cells (jaws) in a scissor-like fashion, with 3d-printed hemispherical plastic tips consisting as the grasping surface. The device also contains an encoder to measure the angle as the device opens

- Detailed discussion of grasper-
- Load cell, encoder.



- Code discussion
  - Discussion of code used to record data and how its recorded
  - How grasping device interfaces with daq board into the computer



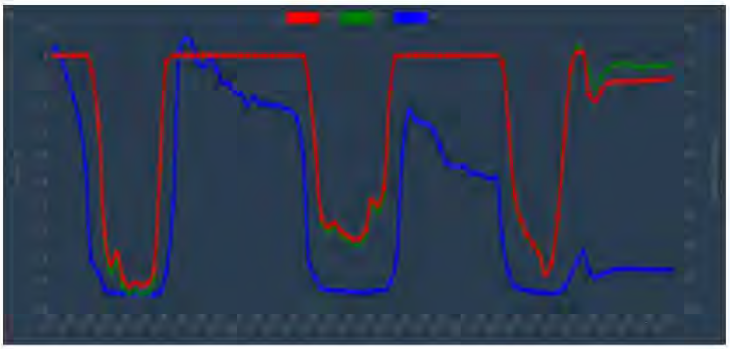
Grasping Device Operation

Setup

The grasping device is first placed into a medical glove such that the load cells are placed into the fingers of the glove: the top jaw goes into the “index finger” and the lower jaw goes into the “pinky finger”. The remaining fingers of the glove are taped against the device in order to prevent interference during testing.

The grasping device is connected to a National Instruments DAQ, which is then connected to a computer. The computer runs the “Scissors Console” application, which samples the voltage and encoder data at 1000 Hz, and stores the data in CSV files defined by the user. The application also displays a graph of the data collected in real-time, and produces a series of tones at fixed time intervals in order to produce consistent grasp timings.

Screenshot of the Scissors Console application. The plot displays data collected in real-time.



Calibration

First, the Futek Signal Conditioners are adjusted such that the voltage readings from the load cells are within +/- 0.1 V when the grasping device “closed” and held horizontally (the “top” load cell points towards the floor, while the “bottom” load cell points towards the ceiling without touching each other).
The grasping device is then placed on a horizontal surface such that a load cell is oriented perpendicular to the floor. The voltage data is recorded as a baseline, then dead weights with known masses (varying from 10 mg to 500 mg) are placed on the load cell, and the voltage data is recorded for each mass. The weighting process is repeated for the other load cell, producing voltage data for every weight for both load cells.

The full calibration process is repeated at the beginning and end of each day of data collection.

The data is later used to convert from the recorded voltage data to force data. This is done by applying a linear regression to a plot of Force (converted from weight) to Voltage. The resulting fit coefficients are then applied to the recorded voltage data to obtain force data.

Data Collection

An individual designated as the “Tissue Holder” measures the tissue temperature, and adjusts the tissue into a suitable position for data collection; ideally holding the tissue horizontally with space for both jaws of the grasping device to freely move without touching any other objects. The tissue holder also indicates where the tissue is to be grasped - making sure to avoid the edges of the tissue and any surface irregularities.

Once the tissue is in position, the data collection process is initiated. The Scissors Console application produces a series of tones to dictate the timing of the data collection process. Initially, the console produces 3 beeps, each 1 second long, with 0.75 seconds between beeps

as the grasping device is “stretched” open and closed in order to ensure that the index pulse is triggered on the encoder.
Once the stretching period has elapsed, the console waits for 3 seconds and then produces a different set of tones.
First, the console produces a short beep for 200 milliseconds as the user moves into position on the grasp site. After a 500 millisecond delay, the console beeps continuously for 1500 milliseconds, and the user grasps the tissue site throughout the duration of the beep. Once the beep stops, the user releases the tissue from the grasp and the console waits for 1000 milliseconds to repeat the cycle.

This cycle is repeated 5 to 10 times at a single grasp site, using the beeps to ensure consistent grasp duration and frequency. Once the desired number of grasps have been completed at the site, the “Stop” button is pressed on the console application, and the grasp site is marked using a permanent marker and/or a meat marker.

The Data Collection process is repeated at several grasp sites throughout the tissue, for each tissue. However, certain tissues such as the aorta and esophagus can only be effectively grasped at a single grasp site during *in-vivo* testing, due to physical limitations. The temperature of the tissue is taken again on the surface of the tissue, near the location of the trials.

Data Processing

Preprocessing

The raw data from the CSV files are first read by a MATLAB script and collected into one large data structure which organizes individual trials by animal, tissue, test stage (*in-vivo*, *ex-vivo*, *postmortem*, etc.), and finally by test site. During this process, the script discards all data from the initial “stretching” period and converts recorded voltage data into force.

Grasp Detection/Segmentation

Once the data has been organized into a MATLAB data structure, another script analyzes each trial and automatically detects individual grasps from a trial.

First, the script obtains a moving mean of the force using a running average of 100 samples. It then calculates the *difference* in force between each subsequent sample, and applies a threshold limit to find the samples where force is increasing in both load cells. Thereafter, periods where the force is increasing are defined as grasps.
The grasps are further refined to combine “broken regions” where a single grasp has been detected as multiple smaller grasps - usually because the grasping device temporarily stops increasing the force applied to tissue.

The automatic grasp detection is then further refined by manually determining the start of the grasp. This is done by plotting the force and manually selecting the time stamp when **both** jaws

have begun contact with tissue, determined by when the force on both load cells have increased greater than the baseline (typically <0.1 N). During this process, we discard any trials that had technical issues that may have previously been unnoticed. These issues usually stemmed from a loose wire connection with the encoder or either load cell.

The script then converts the encoder data into the angle between jaws, and then into the linear distance between the jaws. This is used to define the thickness of the tissue, which is the distance between the jaws on first contact. It then calculates the change in tissue thickness from the beginning of the grasp. From this data, we calculate the "pseudo-strain" of the tissue, by dividing the change in tissue thickness by the initial thickness, producing the normalized displacement.

Finally, the script also creates data fields for testing information. This information includes tissue temperature, location on the tissue (if available), testing type (whether it was a site that had previously been tested or not), and any miscellaneous testing notes. These data are entered manually from a checklist filled out during testing.

Curve Fitting

The following is the equation used for curve-fitting:

$$F(t) = \alpha * (\exp(\beta * \varepsilon(t)^2) - 1)$$

Where α (alpha) and β (beta) are the curve-fit parameters to be determined; while $F(t)$ and $\varepsilon(t)$ are the force and (normalized) displacement of a grasp, respectively.

We utilize both the "trust-region-reflective" algorithm as well as the "levenberg-marquardt" algorithm with a step tolerance of 10^{-13}, as well as function tolerance and optimality tolerance of 10^{-6} (default MATLAB settings).

The initial guesses for curve-fitting were defined for each tissue as follows:

Tissue	α	β
Liver (Default)	-2	5
Spleen	-1	2.5
Aorta	-0.25	4
Lung	-0.03	6.5
Belly	-0.5	6.5
Calibration Puck	-20	1

Initial guesses for curve-fitting

Curve-fitting is run multiple times for each grasp; each time, the residual for the function is calculated. If the squared-sum of the error (SSE) is lower than the previous best sum, it is selected as the solution. The current best solution is then fed into the curve-fitting algorithm as a new initial guess. The script switches between the two solvers on each subsequent iteration, until both solvers are run 3 times each.

After obtaining the best solution, we calculate a "quality" metric for the solution similar to how an R-squared value is calculated for a linear fit. First, we calculate the sum of squared total:

$$SST = \sum_{t=1}^{n} (F(t) - \bar{F})^2$$

Where $F(t)$ is the force at time t and $\bar{F}$ is the mean force. We can then calculate the quality metric as:

$$Quality = 1 - SSE \div SST$$

The generated alpha and beta values are saved along with the generated quality metric and other data from each grasp.

Data Analysis

We calculate a "stiffness" metric for the tissue in each trial by computing the analytical derivative of the curve-fit Force-Displacement curve of each grasp. The derivative of the curve is given as follows:

$$stiffness = \frac{dF(t)}{d\varepsilon(t)} = 2\alpha\beta\varepsilon(t) * \exp(\beta * \varepsilon(t)^2)$$

Thus, we calculate the stiffness from the curve-fit parameters and the displacement of the tissue at a point in the grasp where the force applied is equal to some desired force value. Consequently, any grasps that did **not** reach the desired force value cannot produce a stiffness metric with this method.

In this paper, we only analyze the very first grasp at a particular location as the act of grasping tissue can result in permanent deformation of the tissue at the location. Thus, only the first grasp in a trial is analyzed, and any trials at a marked location (previously grasped) are not analyzed.

Statistical Analysis

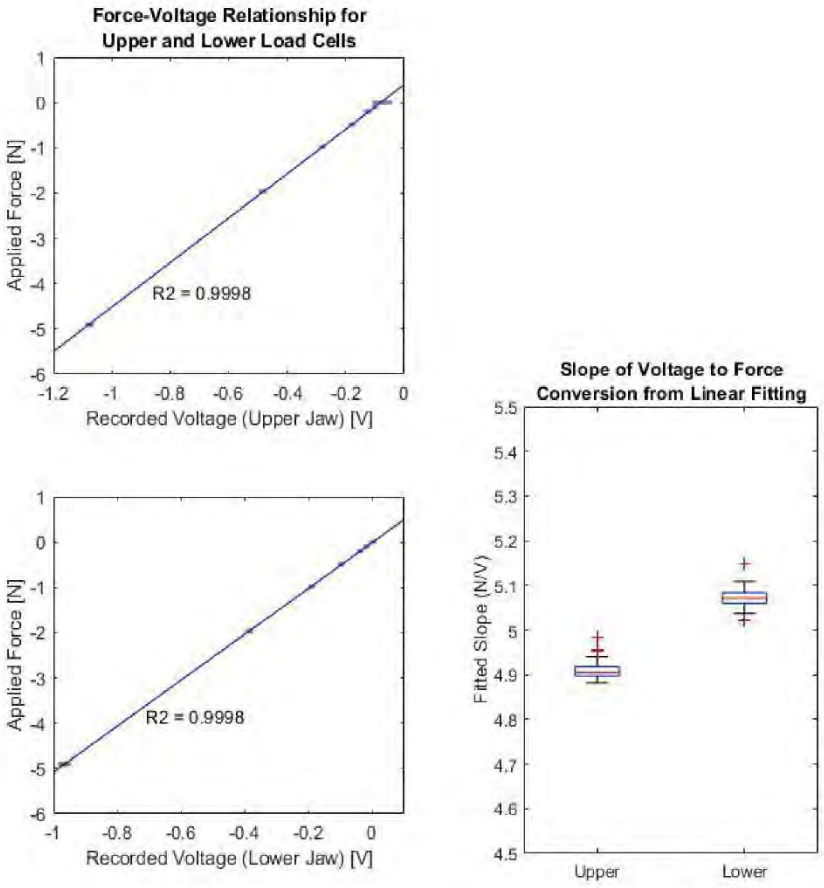
TBD

Data Validation

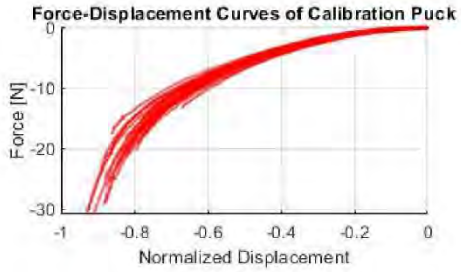
Worst fit: 20190715.D0.Mid before 2

Top Load Cell Slope = 4.901 worst-case R2 = 0.997862

Bottom Load Cell Slope = 5.076 worst-case R2 = 0.996674



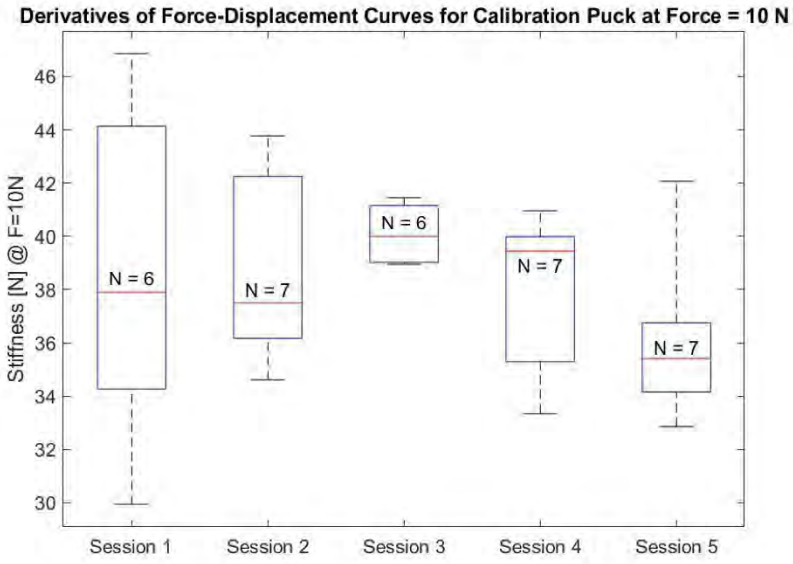
Voltage to Force linear fits have R-squared values above 0.99 for all fits, and the slope of the conversion (in Newtons per Volt) is consistent across calibration sessions.

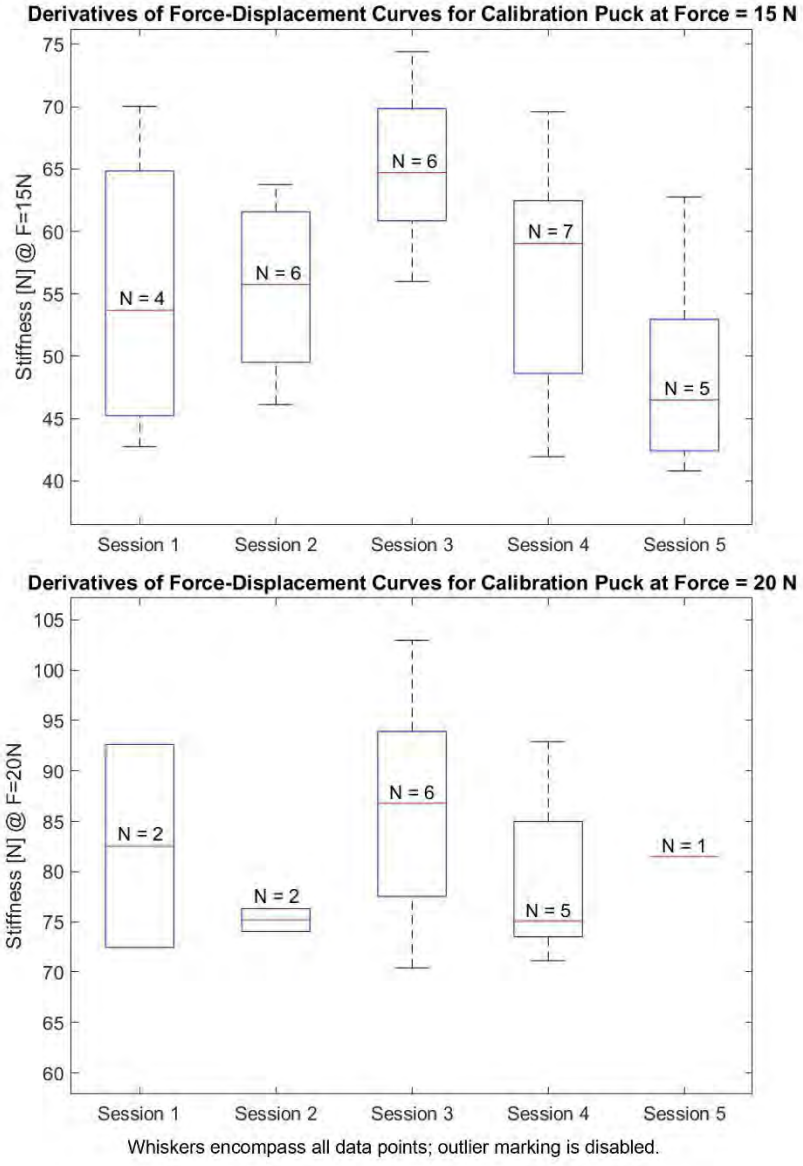


Plot of the "raw" Force vs. Displacement for the Calibration puck. The apparent "stair-stepping" of the curves is caused by quantization noise from the encoder.

Calibration Puck data shows some variation, despite being the exact same material with same measurement methods. Variation can be partially explained by grasping near the edges (apparent material stiffness can change due to edge effects), and grasping the puck at an angle.

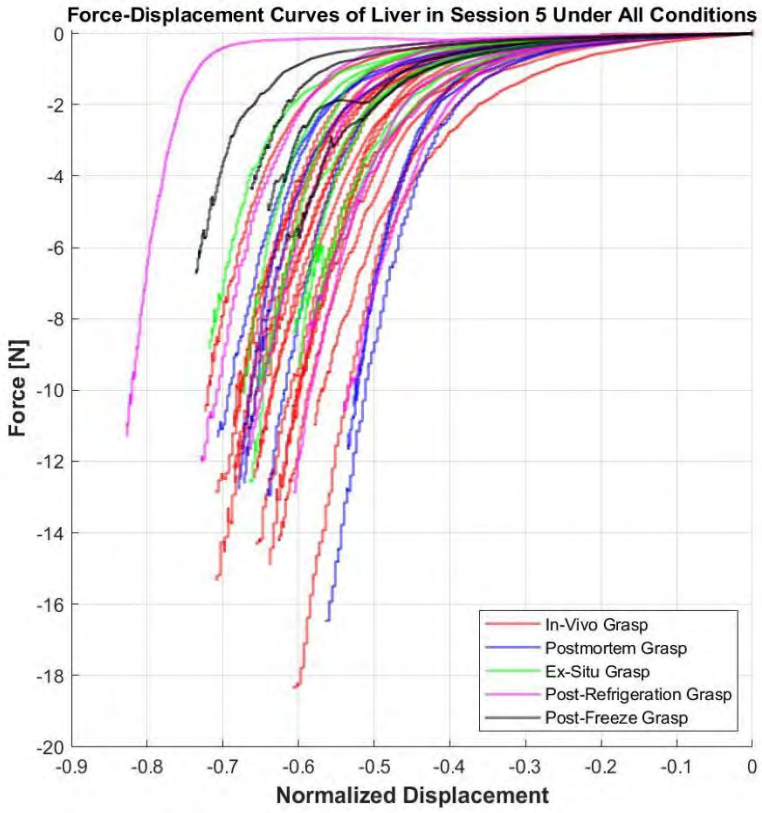
Next, we quantify the stiffness of the puck at various force values.

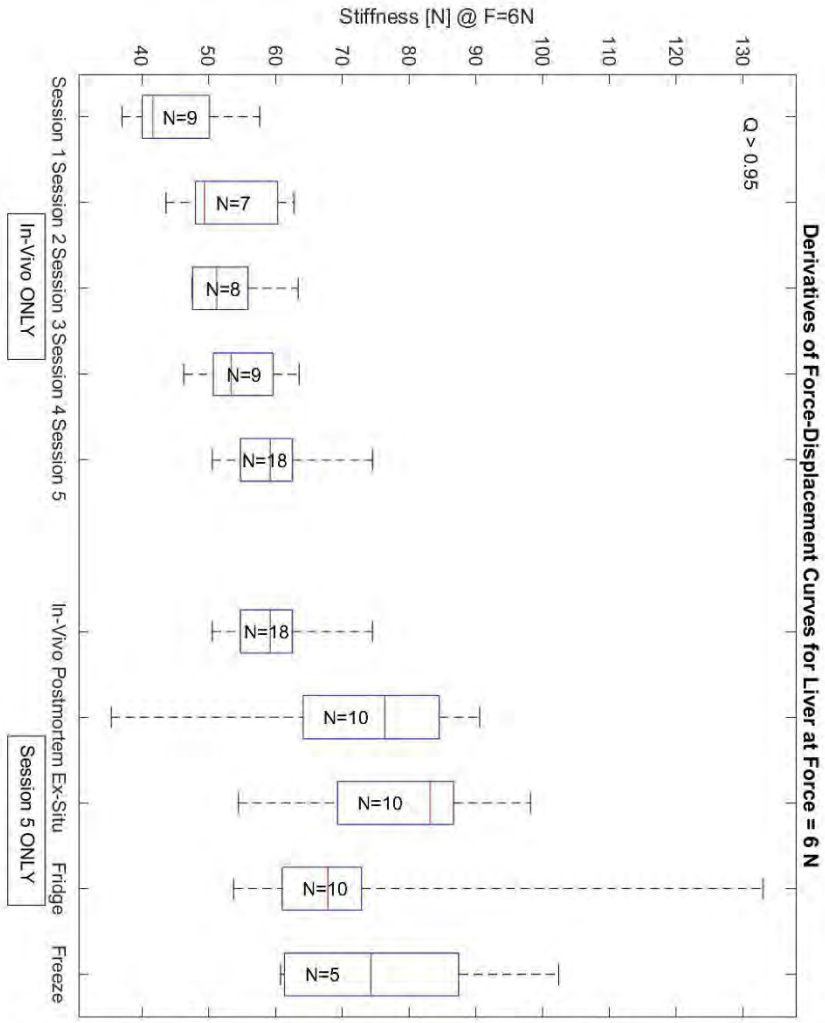



Derivatives of Force-Displacement Curves for Calibration Puck at Force = 15 N
Stiffness [N] @ F=15N
75
70
65
60
55
50
45
40
N = 4
N = 6
N = 6
N = 7
N = 5
Session 1
Session 2
Session 3
Session 4
Session 5
Derivatives of Force-Displacement Curves for Calibration Puck at Force = 20 N
Stiffness [N] @ F=20N
105
100
95
90
85
80
75
70
65
60
N = 2
N = 2
N = 6
N = 5
N = 1
Session 1
Session 2
Session 3
Session 4
Session 5
Whiskers encompass all data points; outlier marking is disabled.

Results/Discussion

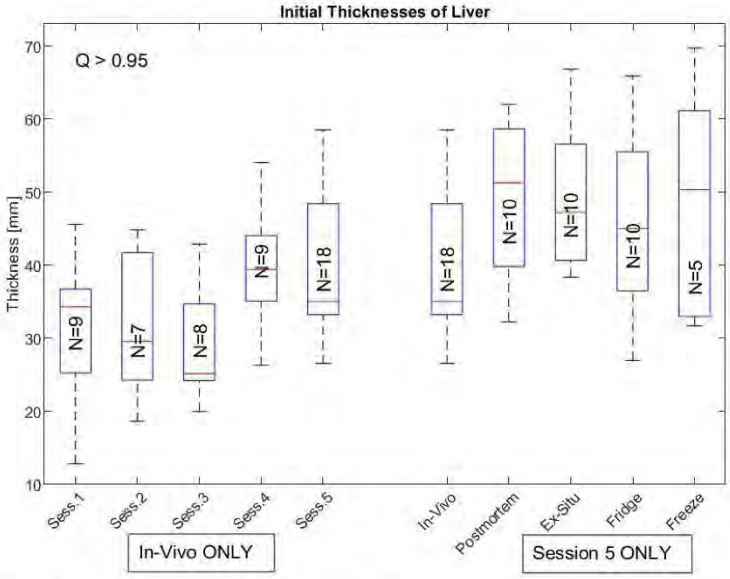
Liver





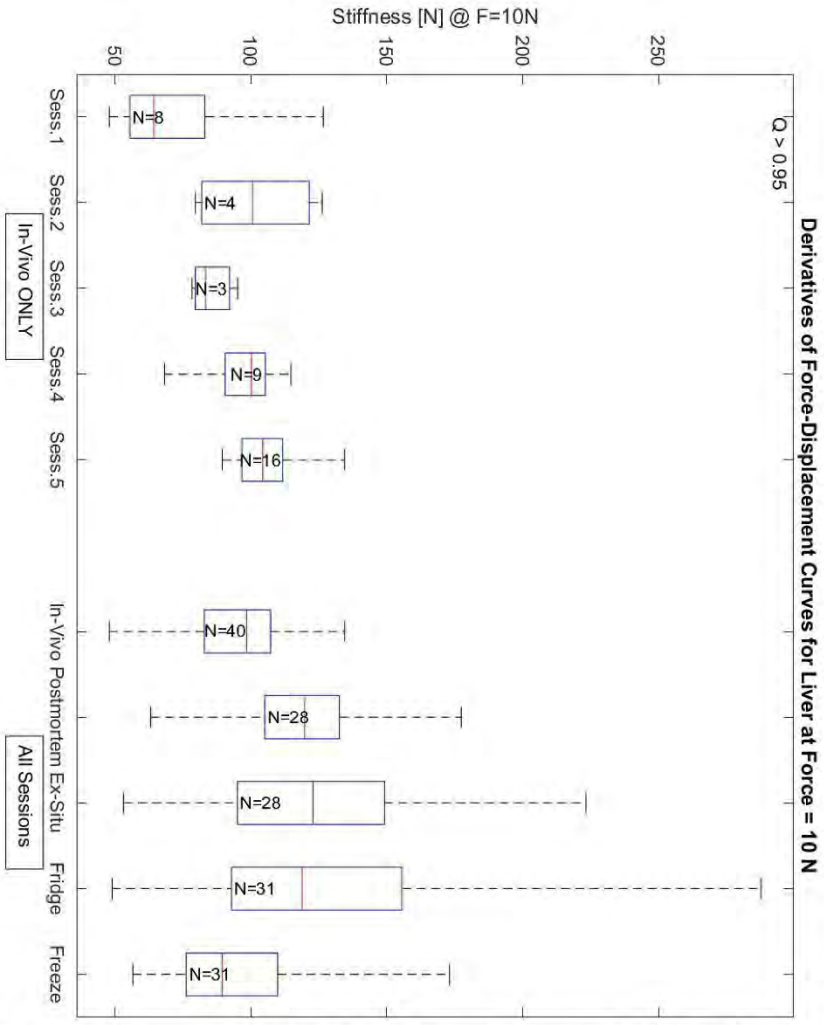
Plot of Liver stiffnesses measured via the Curve-Fit-Derivative approach. The first 5 box plots compare stiffnesses across sessions (different animals) as measured during the In-Vivo experiments; the last 5 box plots show the measured stiffnesses across the different test conditions. The “N =” value indicates how many grasps are included in each set; and each grasp has a quality value of at least 0.95. All outliers are included in the whiskers: outlier detection is disabled.

We observe increased stiffness in the liver after euthanasia, until we refrigerate (and re-heat) the tissue. Unfortunately, the post-freeze data is more limited with only 5 samples; many of the samples were “lost” due to insufficient maximum force applied to the tissue.

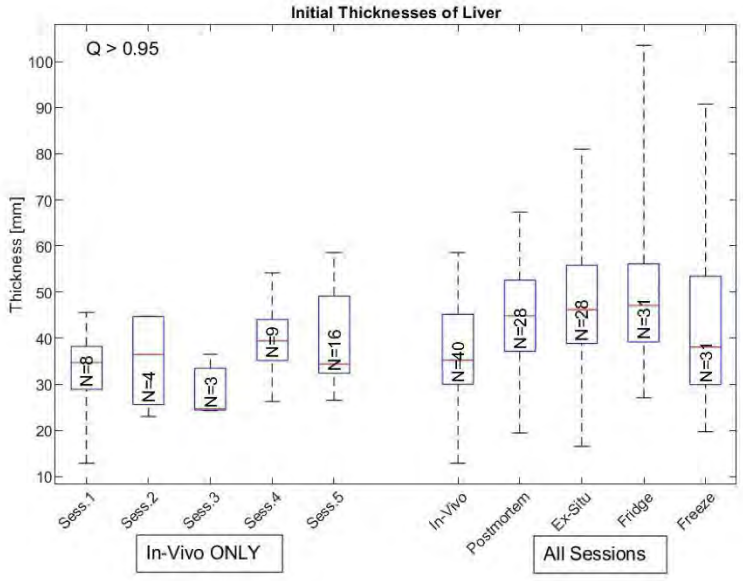


Plot of detected tissue thickness (at the beginning of the grasp). This plot uses the same format and filtering metrics as the plot of stiffnesses above.

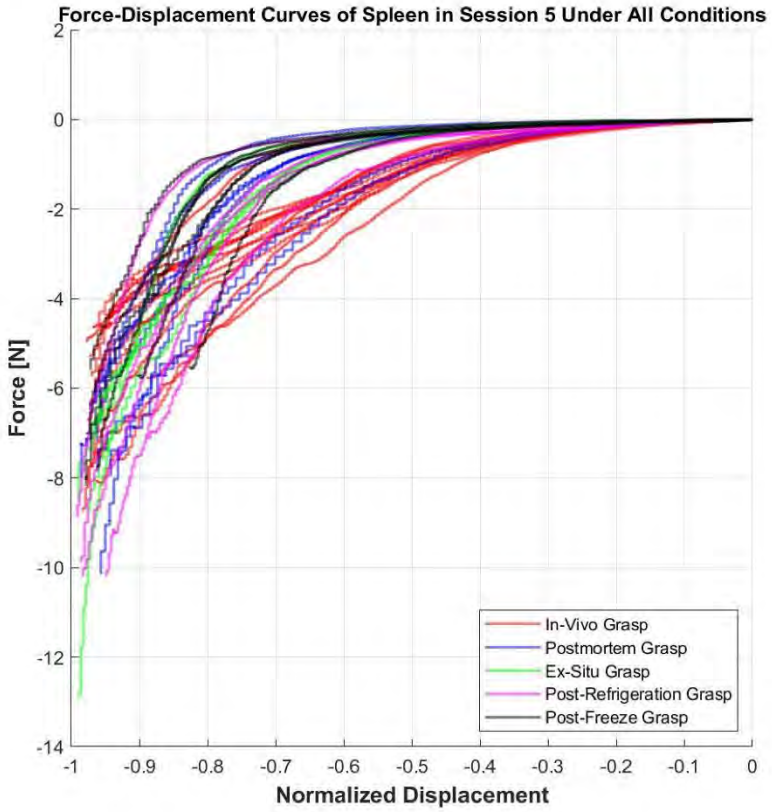
We re-create the plots above, but utilizing data from all the sessions for the tissue stages. We also include plots of different tissues:



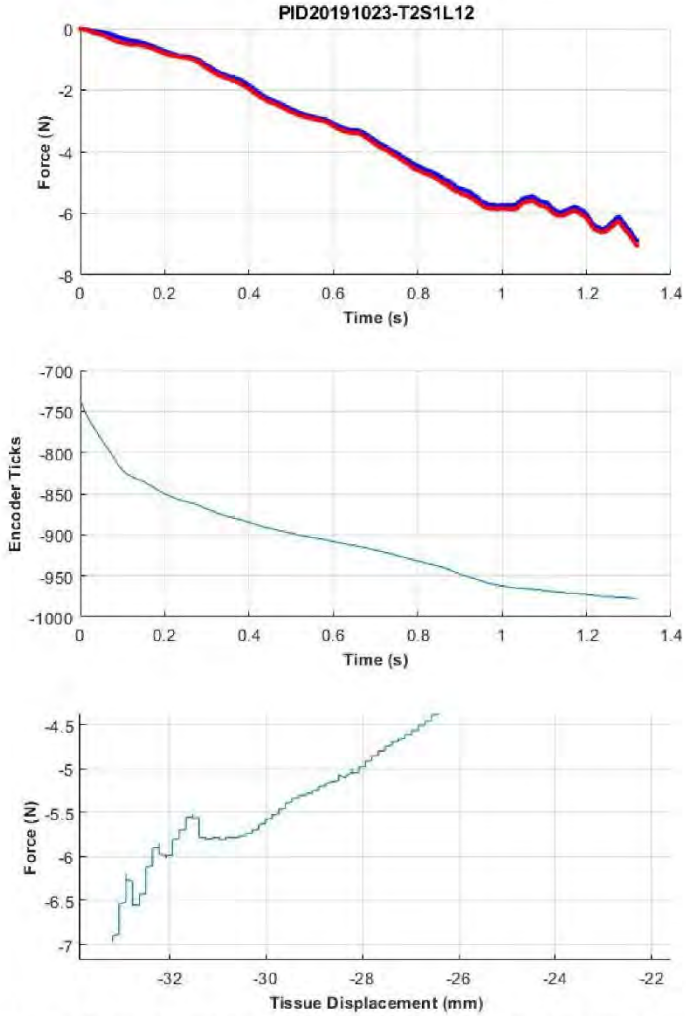
Plot of Liver Stiffnesses, similar to the plot above; however, the force is set to 10 N and data from all sessions is included in the data for different tissue conditions. Again, minimum curve-fit quality is set to 0.95.


Initial Thicknesses of Liver
Q > 0.95
Thickness [mm]
100
90
80
70
60
50
40
30
20
10
N=8
N=4
N=3
N=9
N=16
N=40
N=28
N=28
N=31
N=31
Sess.1
Sess.2
Sess.3
Sess.4
Sess.5
In-Vivo
Postmortem
Ex-Situ
Fridge
Freeze
In-Vivo ONLY
All Sessions

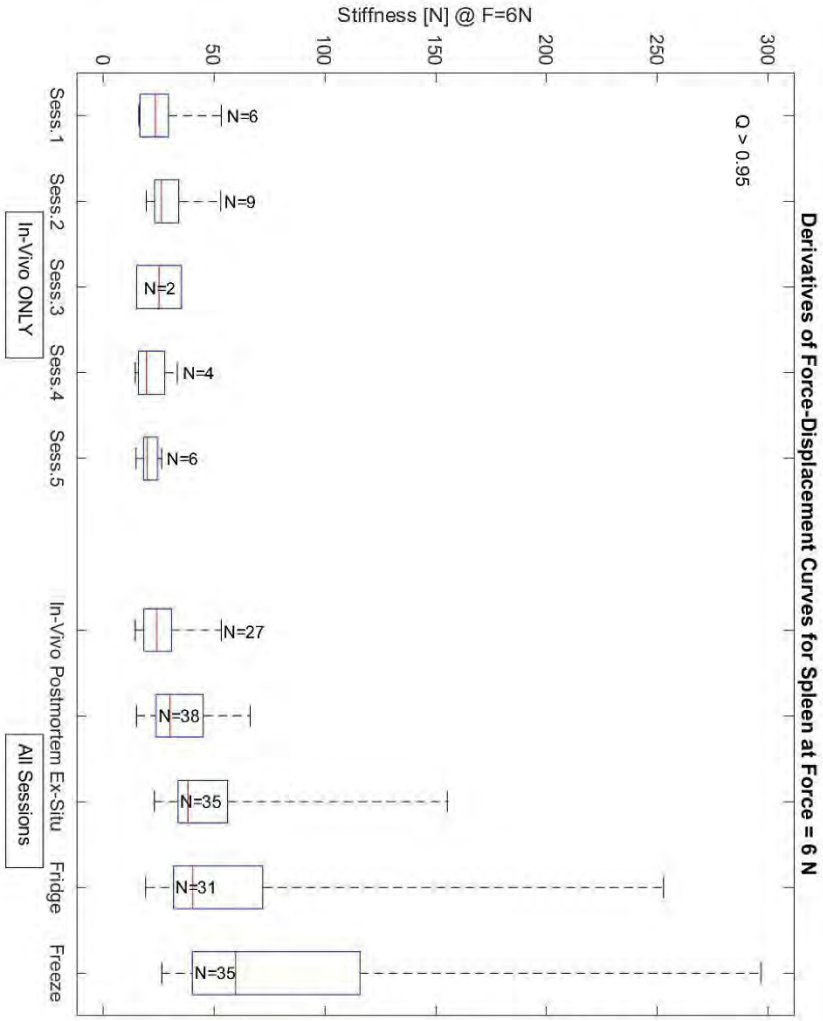
Spleen



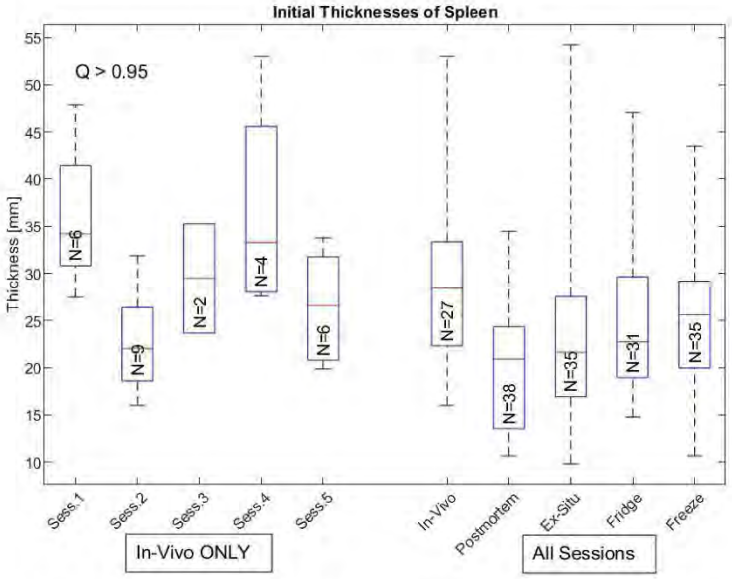
We believe the spleen is a special case that is dominated by fluid/viscous effects. This is due to certain behavior observed in the Force-Displacement curve of the spleen: if a constant (nonzero) force is applied to the spleen, the displacement increases despite no additional effort applied by the grasper. This effect can be so pronounced that a *reduction* in force applied can still produce increased displacement. An example of this is shown in the figure below:



The figure shows force in top plot (red line being the upper jaw, blue line being lower jaw); the encoder ticks in the middle plot, and the resulting force-displacement curve in the bottom plot. Around the 1 second mark, the applied force remains roughly constant at just under 6 Newtons, but the grasper continues to close, as evidenced by the encoder ticks decreasing. Shortly after, the applied force is *reduced*, but the grasper continues to close.

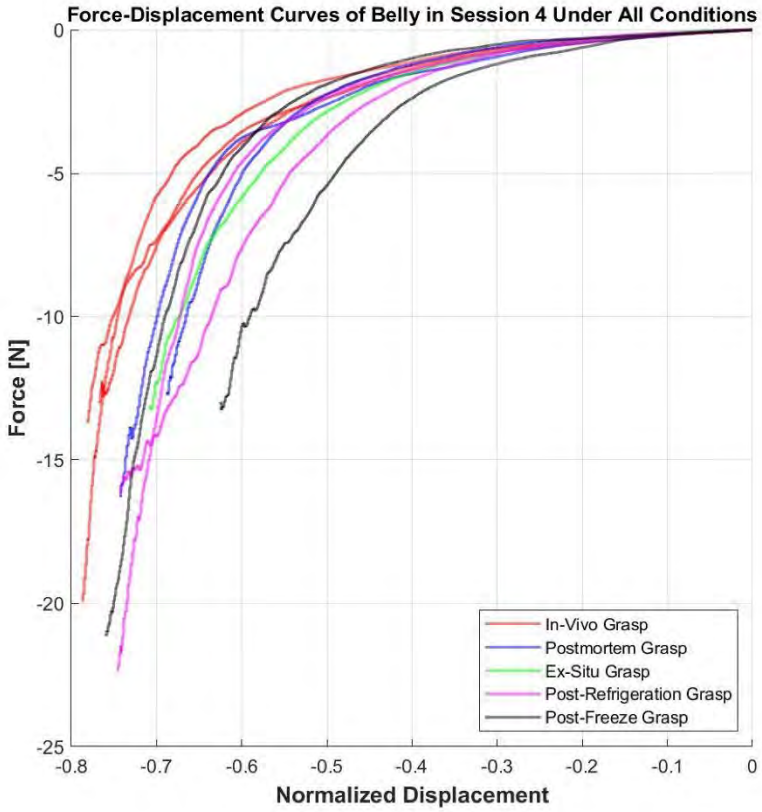


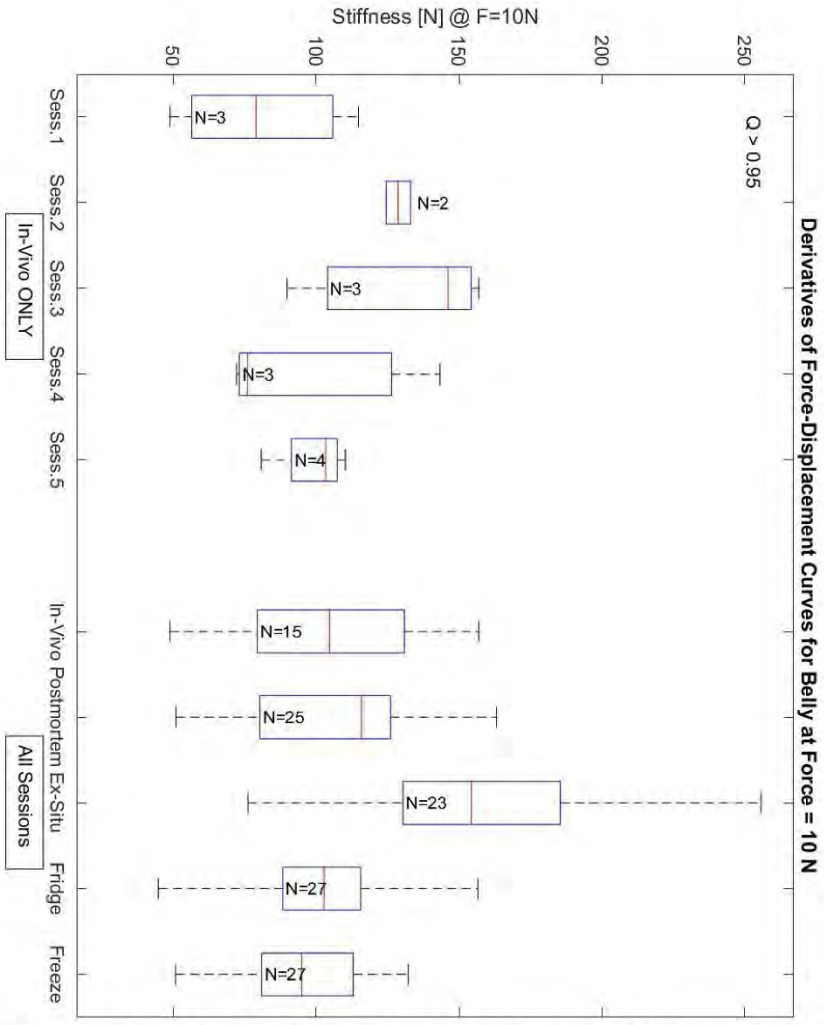
Stiffness of Spleen Tissue across Sessions and Conditions.



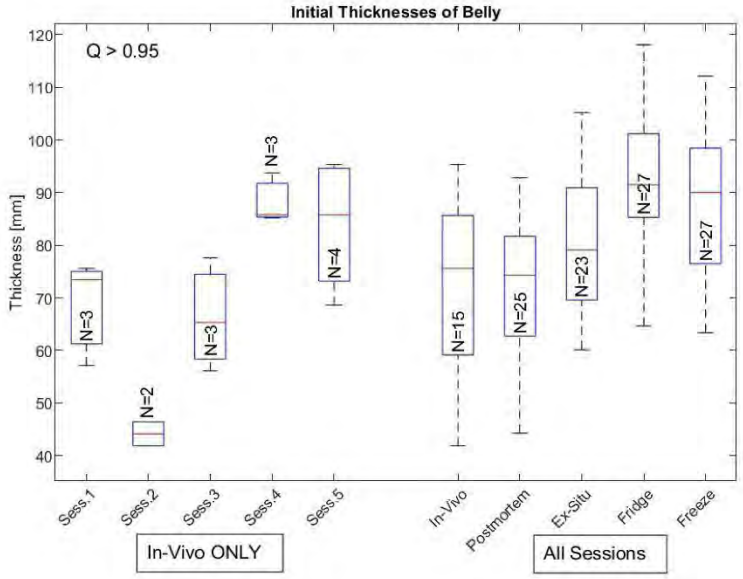
The spleen exhibits consistent, increased stiffness beginning with Postmortem and increasing stiffness after the tissue is extracted and subjected to the later conditions. This further supports the notion that the stiffness of the spleen is at least partially determined by fluid/viscous effects, as the fluid in the spleen is evacuated as time goes on (after death).

Belly/Peritoneal Lining

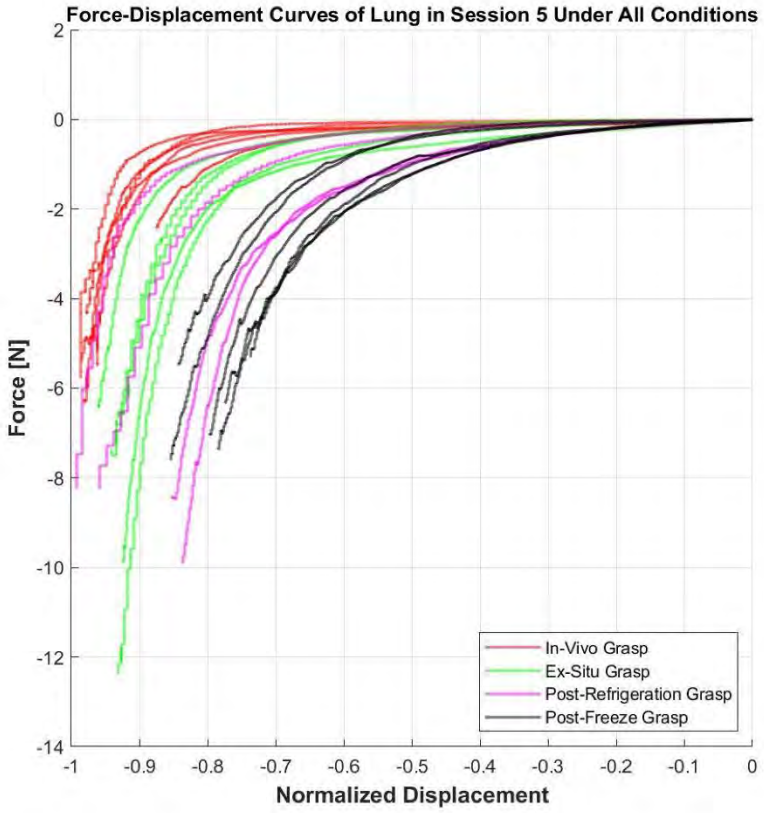




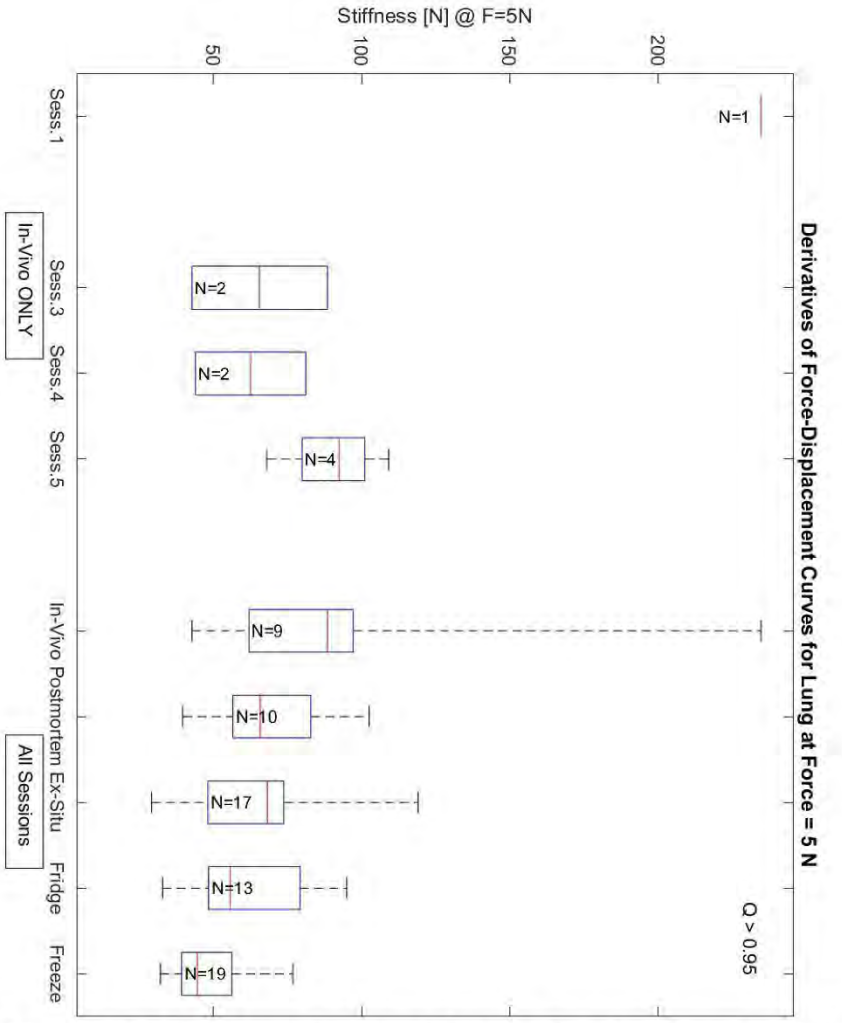
Stiffness plots for the Belly and Peritoneal Lining across Sessions and Conditions. The belly seems to remain largely consistent across trials, except for the Ex-Situ condition which seems to exhibit increased stiffness.


Initial Thicknesses of Belly
Q > 0.95
Thickness [mm]
120
110
100
90
80
70
60
50
40
N=3
N=2
N=3
N=3
N=4
N=15
N=25
N=23
N=27
N=27
Sess.1
Sess.2
Sess.3
Sess.4
Sess.5
In-Vivo
Postmortem
Ex-Situ
Fridge
Freeze
In-Vivo ONLY
All Sessions

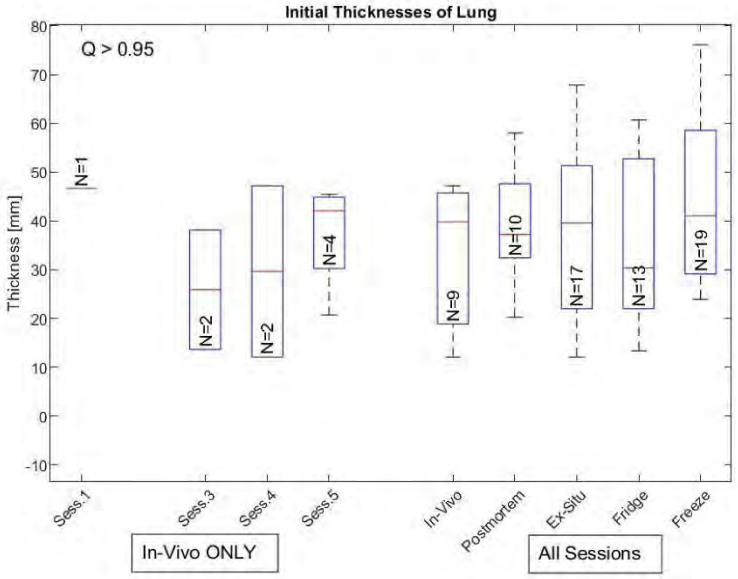
Lung



Force-Displacement curves of Lung Tissue collected during Session 5. Postmortem data was not collected.



Stiffness plots for the Lung across Sessions and Conditions. No *in-vivo* data is included from Session 2 due to insufficient force values.


Initial Thicknesses of Lung
Q > 0.95
Thickness [mm]
80
70
60
50
40
30
20
10
0
-10
N=1
N=2
N=2
N=4
N=9
N=10
N=17
N=13
N=19
Sess.1
Sess.3
Sess.4
Sess.5
In-Vivo
Postmortem
Ex-Situ
Fridge
Freeze
In-Vivo ONLY
All Sessions

Discussion

Different tissues exhibit different behavior across testing conditions.
Liver data does not exhibit significant differences in stiffness from postmortem and onwards. Post-freeze stiffness returns to values closer to those observed during in-vivo. There is largely an increase in variation after in-vivo, which may be explained by variations in temperature.

Spleen data displays strong viscous effects during the *In-Vivo* condition, and becomes increasingly stiff with the later conditions. This is unsurprising, as the spleen contains a significant amount of blood in life, and begins to "drain" starting with postmortem. By the time the tissue is the *Ex-Situ* condition, a significant amount of blood has evacuated from the spleen, resulting in significantly reduced size of the organ, as well as increasingly stiff measurements, as the tissue becomes thin and the grasper presses against itself.

"Belly" data shows almost no variation in across testing conditions, except for the *Ex-Situ* condition. The cause of this discrepancy is unknown, and its cause could simply be due to methodology and/or temperature effects.

Lung data indicates a somewhat steady decline in stiffness as time goes on. The cause of this "softening" effect is unknown and may merit further review. However, the significant apparent variation in stiffness during *In-Vivo* is likely caused by the active expansion and contraction of the lungs as the animal breathes. While the breathing effects are unlikely to cause significant effects to the *true* stiffness of lung tissue, the expansion and contraction of the lungs significantly impact the handling of the tissue and other tissues around it; therefore, an accurate simulator must simulate lung expansion and contraction as the "patient" breathes.

Conclusion (500 words)

- **What we found - brief synopsis**
- How can this information be used?
  - Potential applications in medical training
  - Better medical simulators
- Future work
  - Further tests on other tissues
  - Better human patient based test possibility
- Stiffness is one of many tissue parameters that affects perception.