KNEE TORQUE TEST JIG:

Gathering Torque Profiles About the Knee Joint of a Biomimetic Humanoid Robot

Lindie Burgess
burgel89@gmail.com

Table of Contents

List of Tables and Figures

Nomenclature

Biomimetic: relating to or denoting synthetic methods which mimic biochemical processes.

Onyx Material: a micro carbon fiber filled 3D printable nylon that yields accurate parts with high quality surface finish. It offers high strength, toughness, and chemical resistance when printed alone, and can be reinforced with Continuous Fibers to yield aluminum-strength parts.

Pneumatic: containing or operated by air or gas under pressure.

Tibia: the inner and typically larger of the two bones between the knee and the ankle (or the equivalent joints in other terrestrial vertebrates), parallel with the fibula.

Muscle Origin: A skeletal muscle attaches to bone (or sometimes other muscles or tissues) at two or more places. If the place is a bone that remains immobile for an action, the attachment is called an origin.

Muscle Insertion: A skeletal muscle attaches to bone (or sometimes other muscles or tissues) at two or more places. If the place is on the bone that moves during the action, the attachment is called an insertion.

Pneumatic Artificial Muscles (PAMs): are contractile or extensional devices operated by pressurized air filling a pneumatic bladder. In an approximation of human muscles, PAMs are usually grouped in pairs: one agonist and one antagonist.

Torque: a twisting force that tends to cause rotation.

Actuation: the action of causing a machine or device to operate.

Flexion: refers to a movement that decreases the angle between two body parts.

Extension: refers to a movement that increases the angle between two body parts.

Patella: the kneecap.

Femoral Condyles: The femoral condyles are the two rounded prominences at the end of the femur; they are called the medial and the lateral femoral condyle, respectively. The motions of the condyles include rocking, gliding and rotating.

Distal: situated away from the center of the body or from the point of attachment.

Proximal: situated nearer to the center of the body or the point of attachment.

1 Introduction

Much can be learned about material properties, kinematics, dynamics, and control of biological systems through the design of robots which mimic biology. Additionally, biologically inspired robots can offer platforms on which to test controllers and to study assistive technology. Steady progress has been made in Portland State University's Agile and Adaptive Robotics Lab in developing a Biomimetic Humanoid Bipedal Robot. Artificial skeletal features are 3D printed in Onyx and artificial muscles are implemented using Festo brand pneumatic actuators[1], [2]. A knee joint, whose center of rotation translates as the tibia rotates about the knee, was previously developed to have similar range of motion as a human[3]. Accompanying the physical design, algorithms were written to determine origin and insertion points for the Pneumatic Artificial Muscles (PAMs) so that the resulting torque profiles about the knee joint closely matches the torque profiles of these same muscles in a human[4].

The primary goal of this project was to design and fabricate a test jig which facilitates the acquisition of the true torque profile resulting from the actuation of the PAMs about the knee of the bipedal humanoid robot. Data collected on this jig will be used to validate and refine the models used to design PAM attachment locations.

The following details methods by which data will be collected once the design and fabrication steps of this test jig have been completed. Also included are very preliminary results generated from early testing in March 2021.

2 Methods:

The PAMs on the robot leg assembly are configured such that they allow the maximum range of motion of the knee joint. The femur is statically connected to the test jig while the tibia can rotate, in controlled intervals, about the knee. At the limit of each interval, a load cell quantifies the force induced by isometric contraction of the PAM and an image is taken to gather angle measurements through post processing. Torque data is compiled in Excel and Matlab is used to generate Torque Profiles.

2.1 Leg Assembly Configuration:

PAMs interface to the leg assembly by pin joint at the origin point on the femur and by artificial tendon, bike cable, at the insertion point. The Flexion and Extension PAMs have been sized to the longest lengths possible, allowing for the largest possible range of motion of the knee.

For the flexor PAM, the length is limited by the distance between its origin and insertion points when the leg assembly is fully extended. A length of artificial tendon exists between the end of the flexor PAM and the insertion point and is only long enough for sufficient clearance between the PAM end cap and insertion point through the range of motion of the knee.

FIGURE 1: SETTING THE FLEXOR PAM LENGTH ON A FULLY EXTENDED LEG ASSEMBLY.

Choosing the extensor PAM length is slightly more complicated due to wrapping around the knee. When the leg assembly is fully flexed, the extensor PAM is unpressurized, and the patella should be positioned on the femoral condyles. This patellar position is the determining factor for the length of the artificial tendon between the distal side of the patella and the tibial insertion point. At the same patellar position, the extensor PAM should span as much of the distance between the femoral origin and proximal patellar insertion point, without kinking the PAM. A slight bend in the BPA is permissible. The remaining length between PAM endcap and proximal patellar insertion point is spanned by artificial tendon.

FIGURE 2: SETTING THE EXTENSOR PAM LENGTH ON THE FULLY FLEXED LEG ASSEMBLY.

2.2 Test Stand Configuration:

A test jig, shown below in Figure 3, was designed for the primary purpose of acquiring the true torque profile resulting from the actuation of the PAMs about the knee of the robot.

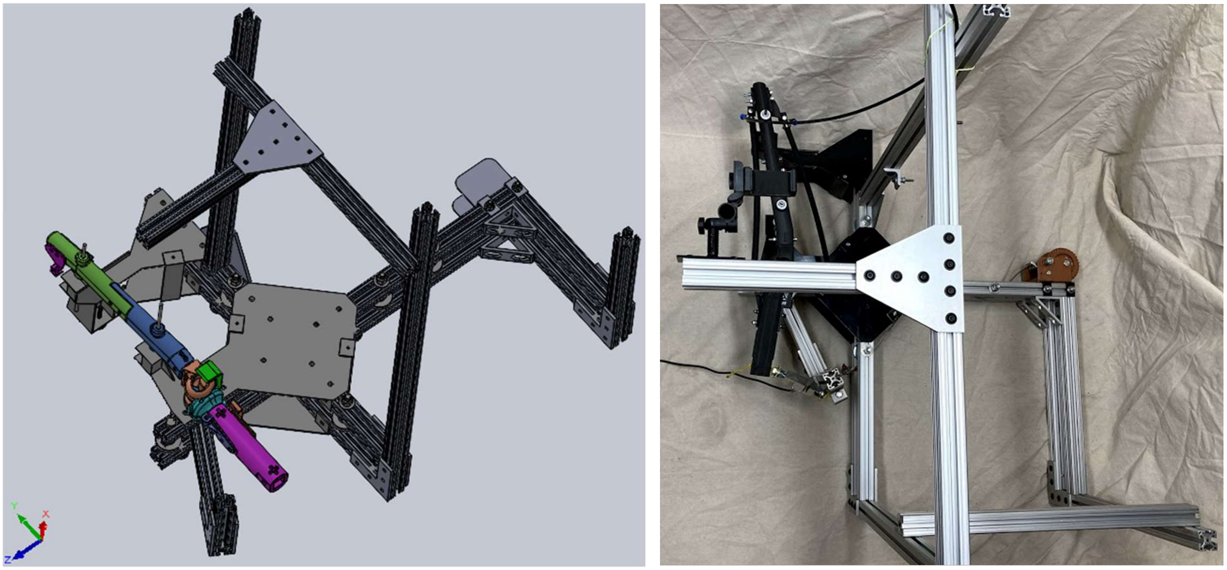

Figure 3: Test jig design for gathering the torque profile produced by PAMs about the knee axis of a biomimetic, humanoid, robot leg assembly.

The femur is fixed in two locations to the femur jig (Figure 4.a), while the tibia rotates through a horizontal plane, in controlled intervals, about the knee. The tibia is connected via a load cell to the Jig Swing (Figure 4.b). The angular intervals traveled by the Jig Swing are controlled by a Dutton-Lainson hand winch with a 900lb first layer load capacity, wound with an ultra-flexible, 3/16 diameter wire rope with a capacity of 600lbs. The winch is very simply modeled (Figure 4.c). At the limit of each angular interval, a load cell quantifies the force induced by isometric contraction of the PAM and an image is taken to gather distance and angular measurements through post processing.

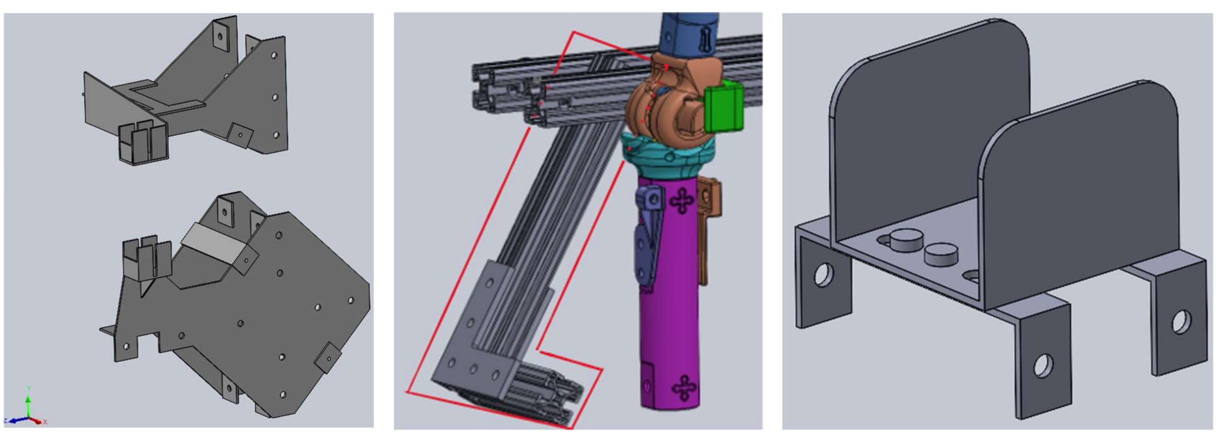

Figure 4: Test jig detail descriptions. From left to right, (A) Upper and Lower femur Jig, (B) Jig swing is outlined in red, and (C) A very simple model representation of a Dutton-Lainson hand winch.

The cable can be rerouted, as shown in Figure 5, to allow for data collection from the flexor and extensor PAM, without removing the leg assembly.

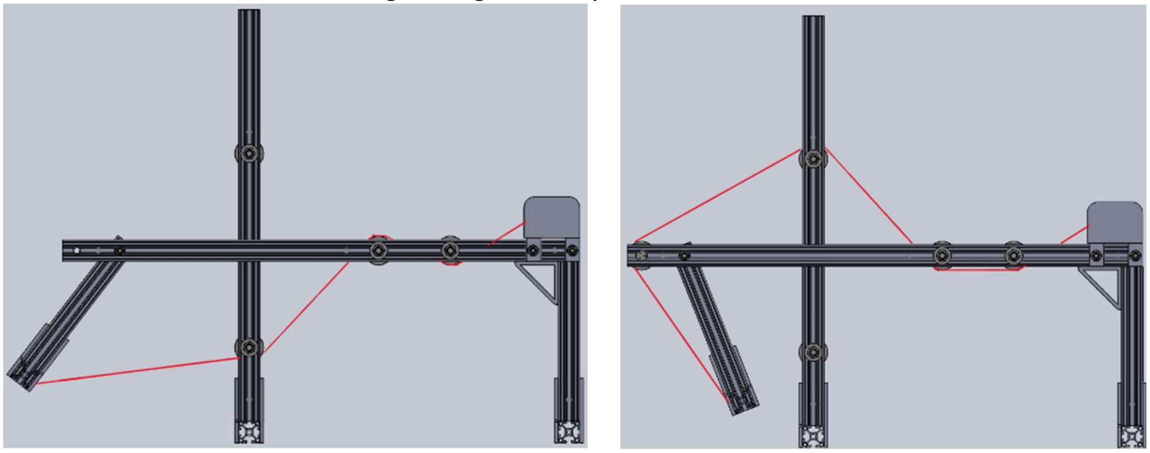

Figure 5: Cable wrapping for tests in flexion and extension. On the left, the cable (modeled in red) is configured for data collection from the flexor PAM. On the Right, cabling is configured for data collection from the extensor PAM.

For more details about the construction of the Test Jig, please see Appendix 1.

2.3 Camera Calibration:

A 2020 iPhone SE was used to collect image data for these tests. The rear facing camera features a Sony IMX315 Exmor RSA sensor with an f/2.2 aperture and image resolution of 12.19 MP. A generic remote shutter was connected via Bluetooth to the iPhone to ensure consistent camera orientation through testing.

Hugin, an open-source software solution, is used to find the internal lens parameters. Before any tests are run, a calibrating tool including at least two straight lines (shown in Figure 6.a) is positioned in approximately the same plane through which the tibia will rotate. The location at which the load cell attaches to the jig swing is employed to set the position of the calibration tool (Figure 6.b).

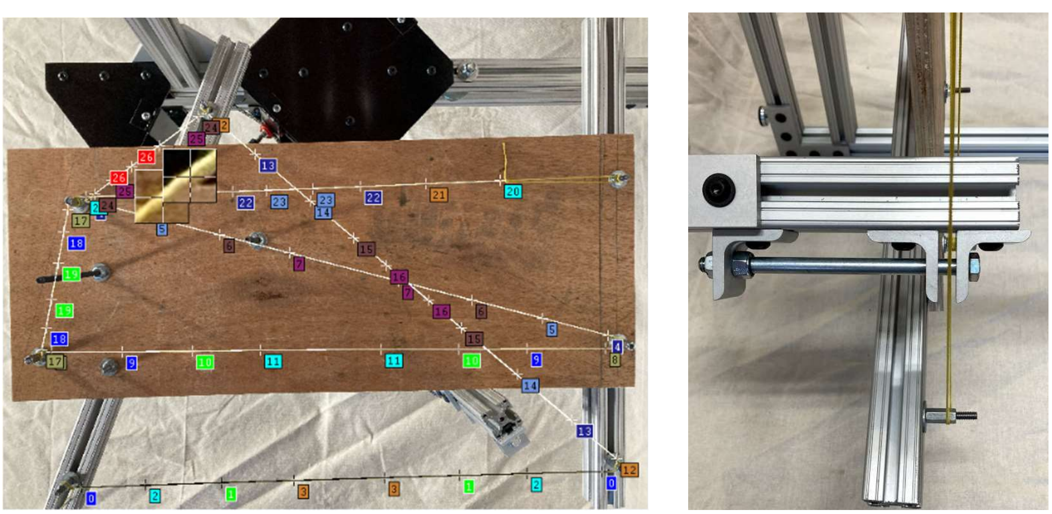

Figure 6: Calibration details. (A) lens calibration tool with straight line control points selected (left). (B) positioning the calibration tool using the test jig swing (right).

Uploading an image of the calibration tool into Hugin allows for the selection of control points on the straight lines, as shown in Figure 6.A. An optimization protocol is run to generate lens parameters which can be saved and applied to any subsequent image taken by the same camera at a similar distance from the image of interest. See Appendix 2 for additional notes about Hugin and the camera calibration process.

2.4 Load Cell Setup:

A CALT brand S-style load cell, in series with a HX711 Load Cell Amplifier and an Arduino Due microcontroller was used to collect force data for these tests.

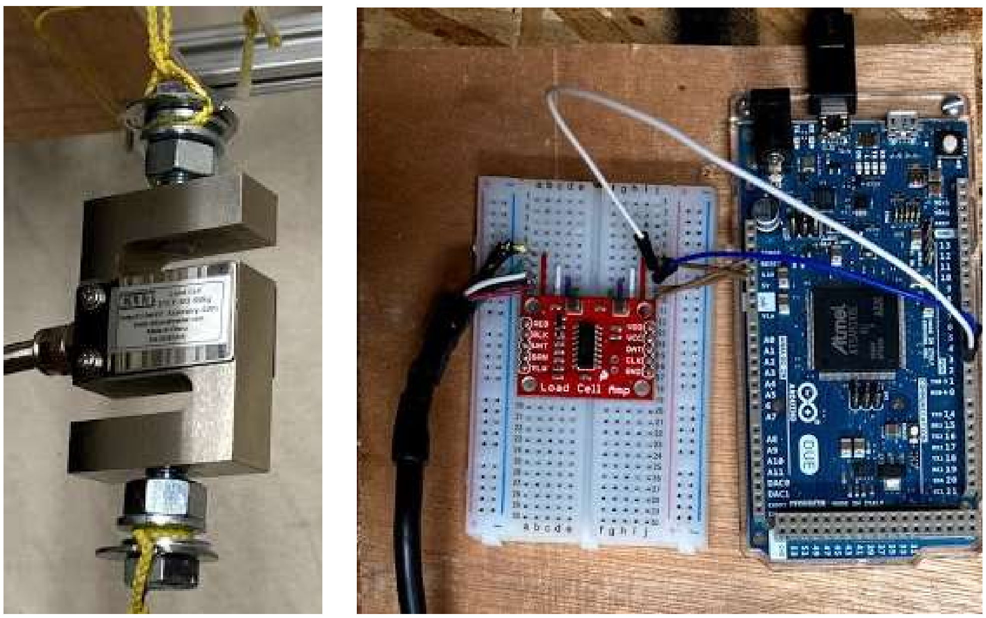

FIGURE 7: LOAD CELL SETUP. LOAD CELL (LEFT) IS CONNECTED IN SERIES WITH AN HX711 LOAD CELL AMPLIFIER AND ARDUINO DUE MICROCONTROLLER (RIGHT)

Data collection was controlled by Matlab through serial connection to the Arduino. For Code and Load Cell Calibration technique, please see Appendix 3.

2.5 Testing:

A Porter and Cable, 6-gallon, 150PSI air compressor, in series with the Festo brand pneumatic manifold shown in Figure 8, was used to pressurize each PAM. The first image collected in each series of data, corresponds to the orientation of the leg assembly when the muscle being tested is unpressurized. For the Flexor series, image 1 corresponds to the leg assembly in full extension. For the Extensor series, image 1 corresponds to the leg assembly in full flexion.

FIGURE 8: FESTO PNEUMATIC MANIFOLD

The following steps are performed in iteration, throughout the range of motion of the knee, allowable by each PAM:

1. Set knee angle. Between each iteration, the winch should be allowed to advance 5 gear teeth.
2. Begin data collection with the load cell
    a. Code used for data collection can be found in Appendix 3. Currently, 6000 data points are collected per iteration, for which a Force vs. time plot is generated, and basic statistics are given for the last 500 data points. A typical result of data collection is shown below in Figure 9.
3. Manually open the solenoid valve on the pneumatic manifold to pressurize the PAM
4. Ensure that regulated pressure stabilizes at 90psi (approx. 6 bar).
    a. If pressure does not reach 90, reset the tank and start these steps from the beginning. Otherwise, continue.
5. Image the system.
6. Save all data.
    a. Transfer statistics to excel.
7. Repeat.

2.6 Post Processing:

Post testing, three stages of data processing are necessary. First, the angle measurements are taken from the images generated during testing. Each image is first uploaded into Hugin and correct for lens distortion. Then, Fiji (Fiji is Just ImageJ) is the open-source software employed to collect angular data, as shown in Figure 10.

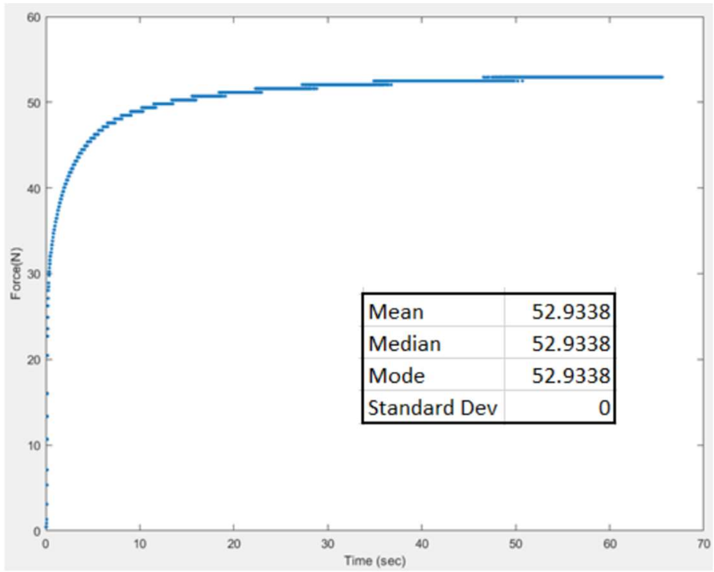

FIGURE 9: TYPICAL FORCE VS. TIME RESPONSE OF THE FLEXOR PAM. INSET STATISTICS ARE GENERATED WITH THE LAST 500 DATA POINTS OF THE 6000 SET SERIES.

Then Moment arm distances and relational angles are collected in Solidworks, a 3D modeling software used to create the leg assembly.

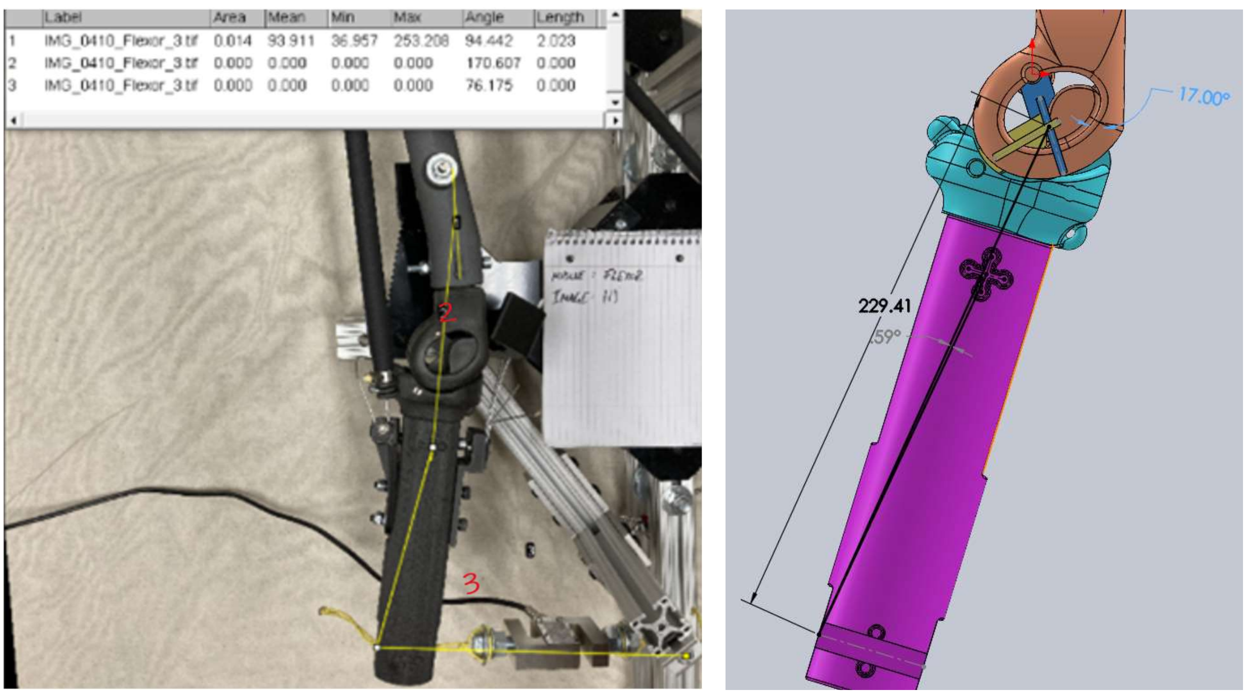

FIGURE 10: GATHERING ANGLE MEASUREMENTS IN FIJI AND SOLIDWORKS. WITHIN THE LEFT IMAGE, MARKER 2 IS THE ANGLE OF THE KNEE. MARKER 3 IS THE ANGLE MEASURED BETWEEN THE LINE OF ACTION AND A LINE BETWEEN A KNOWN REFERENCE POINT AND THE POINT AT WHICH THE RESULTANT FORCE ACTS ON THE TIBIA. IN SOLIDWORKS (RIGHT), THE MOMENT ARM DISTANCE AND ANGLE BETWEEN IT AND THE KNOWN REFERENCE POINT ARE COLLECTED.

Torque data is ultimately calculated in Excel using the data collected during testing and image analysis. Torque profiles are generated in Matlab.

3. Results:

The preliminary tests run in March of 2021 were intended to (1) ensure that the test jig and newly designed leg assembly did not break during testing, (2) Root out design features that need to be addressed before more rigorous testing can occur, (3) Work out kinks in the testing procedure and (4) to Gather a small set of quantifiable data. For the preliminary testing, a 10mm PAM was used, with insertion and origin points that were not biomimetic.

The range of motion available to the robot knee is from +10° in full extension through -150° in full flexion. Generally, the range of motion of a human knee is from +10° in full extension through -120° in full flexion. The range of motion of a human knee is our range of interest.

3.1 Test Jig Design Issues and Proposed Solutions

The test jig and leg assembly held up well under these initial testing conditions. However, there are several design considerations that should be addressed before more rigorous testing is conducted:

For these tests, the Flexor PAM moved the knee joint from +10° through -35°. The test jig facilitated the acquisition of 15 images and force readings on the range of -6° through -35° series as shown in Figure 11.

FIGURE 11: FLEXION IMAGING. LEG ASSEMBLY AT -6° (LEFT) AND AT -35° (RIGHT)

Test Jig Design Issue #1: In its current configuration the test stand has interference between the jig swing and the lower vertical members of the main test frame so that it is not possible to collect force data between +10° and -6°.

Possible solutions for Test Jig Design Issue #1: There are two quickly available and relatively simple solutions for this interference issue.

(1) Reduce the length of the load cell by any combination of the following:
   a. Reduce the length of twine between load cell and tibia
   b. Replace the current "thick" nut with a thinner nut (Figure 12). Then, it also becomes possible to shorten the length of the bolt.

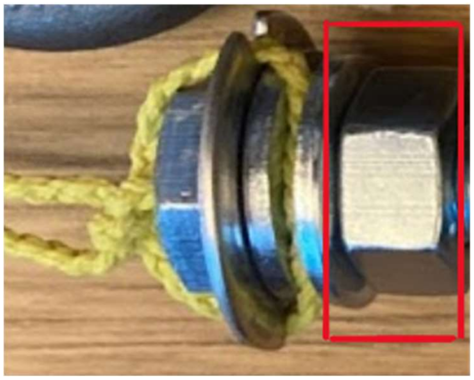

FIGURE 12: TEST JIG DESIGN SOLUTION. REPLACE NUT (OUTLINED IN RED) WITH NUT OF THINNER PROFILE.

(2) The femur jig should be thought of not only as providing two stable mounting locations on which to fix the femur. Adapters can be fitted to these mounting positions so that it is possible to reposition the femur as necessary, or in the future, to attach other parts of the robot assembly.
   a. To address the current issue, an adapter should be designed and printed from onyx, to be added to the top femur mounting position. A simple adapter, such as the sketch in Figure 13, will allow the femur to slightly rotate clockwise about the lower femur mounting position. This solution keeps the knee joint roughly on axis with the pivot axis of the test jig swing, as is necessary.

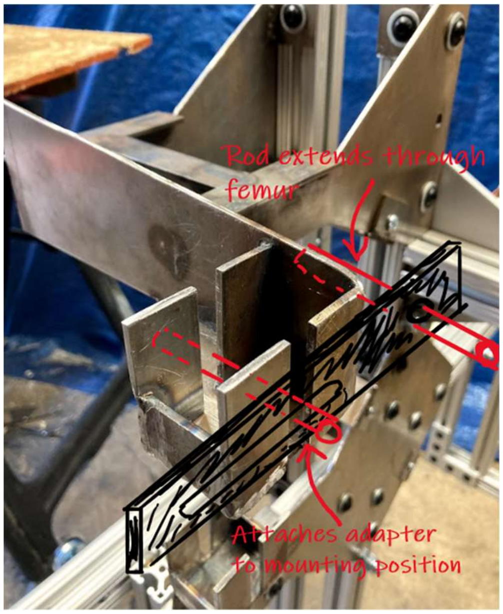

FIGURE 13: TEST JIG DESIGN SOLUTION. ADAPTER ADDED TO UPPER MOUNTING POSITION.

The Extensor PAM moved the knee joint from -150° through -102° as shown in Figure 14. The test jig facilitated the acquisition of 9 images and usable force readings on the range of -137° through -102°.

FIGURE 14: EXTENSOR IMAGING. LEG ASSEMBLY AT -102° (LEFT) AND -137° (RIGHT).

Test Jig Design Issue #2: In its current configuration the test stand has interference between the load cell and the extensor PAM between -137° and -150°. Since, the robot knee has a larger range of motion (+10° through -150°) than a human (+10° through -120°), the interference between PAM and load cell may not end up being an issue.

Possible solutions for Test Jig Design Issue #2: If in the future it becomes desirable to collect data from the robot on the range of -137° through -150°, an adapter with a new load cell attachment point can be added to the jig swing so that, effectively, the jig swing has a longer reach. Then, the length of the twine can be increased between the tibia and the load cell. The solution might look something like the quick sketch in Figure 15.

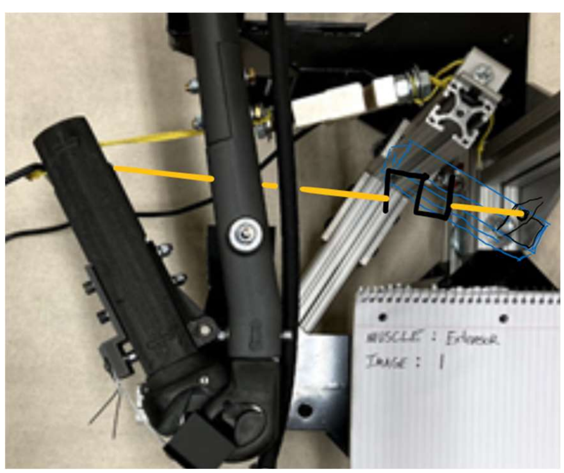

FIGURE 15: PROPOSED TEST JIG DESIGN SOLUTION FOR INTERFERENCE BETWEEN LOAD CELL AND EXTENSOR PAM

Test Jig Design Issue #3: As it currently exists, the wire rope is overdesigned for this test jig application. The design specifications for the wire rope suggest that it should not be wrapped around a diameter of less than around 4 in. The winch drum is around 0.5 in. The effect is that the wire rope (1) acts as a spring pulling on the jig swing, and (2) tends to unwind from the winch if left unloaded while adjusting the winch between imaging and (3) tends to unwind if left unloaded during the rerouting of wire rope between testing the leg assembly in flexion/extension.

Possible solutions for Test Jig Design Issue #3:

(1) Even using the largest possible PAM (40mm), the largest torque induced by PAMs acting about the knee axis translate to around 100ft_lbs where the wire rope interfaces with the jig swing. The wire rope has a capacity of 600lbf; a factor of safety of 6. This means that the diameter of the wire rope can be reduced to avoid all of the issues listed above. It is possible that the wire rope can even be replaced with string for future torque tests induced by 10mm PAM future 10mm PAM tests

(2) To prevent unspooling of the wire rope when reconfiguring the wire rope between tests of extension/flexion, clamp the wire rope to the main test frame, as seen in Figure 17.

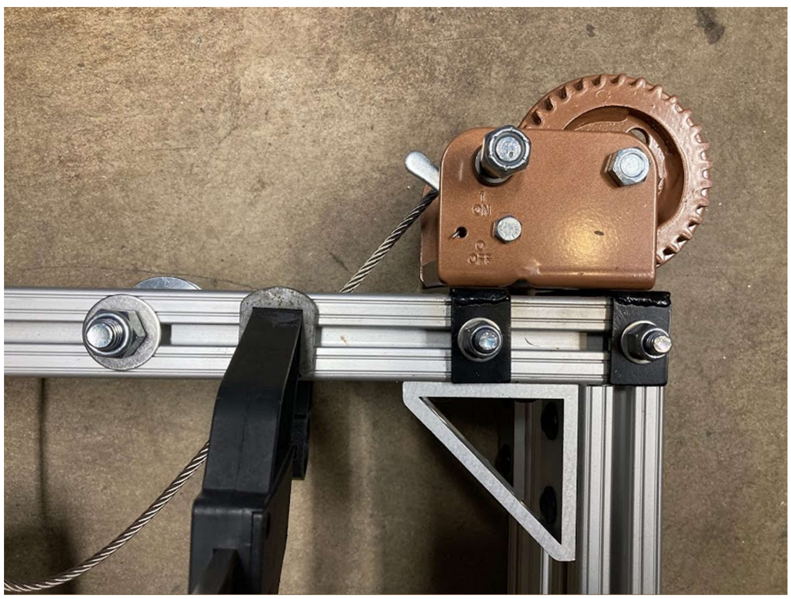

FIGURE 16: TEST JIG DESIGN SOLUTION PREVENTING UNSPOOLING OF WINCH WHEN REROUTING WIRE ROPE

3.2 Leg Assembly Design Issues to Address

Leg Assembly Design Issue #1: Insertion points

The washers that were 3D printed to clamp the artificial tendon at the insertion point are already wearing. It will be beneficial to replace these washers with longer lasting washers with more clamping capability, especially when it comes to testing PAMs that can develop higher forces than the 10mm PAMs which were used in preliminary testing.

It should be noted that if the artificial tendon is not wrapped around the insertion point, there is not currently sufficient clamping force to stop the tendon slipping. Wrapping the tendon at the insertion point has been sufficient. Washers with higher clamping capacity will also help.

Leg Assembly Design Issue #2: Patella tracking
While testing the extensor PAM, a design issue with the patella tracking on the femoral condyles became apparent. There is a position of the leg assembly at which the length of the muscle and tendon between femur origin and patella insertion becomes too long for the patella to stay on the femoral condyles. This becomes an issue for accurate data collection, because when the extensor is pressurized, it settles into a position of least resistance. This position does not serve the intended purpose of extending the leg assembly. See Figure 17 (left).

FIGURE 17: LEG ASSEMBLY DESIGN ISSUE TO ADDRESS - PATELLA TRACKING. PATELLA SLIPS OFF THE FEMORAL CONDYLES (LEFT). EXCESS TENDON USED TO KEEP THE PATELLA IDEALLY POSITIONED (RIGHT).

A workable solution for the current leg assembly configuration is shown in Figure 17 (right). Excess Tendon is used to keep the patella ideally positioned during extensor PAM pressurization.

Leg Assembly Design Issue #3: Settling action of the tendon endcap when PAM is pressurized
While testing the flexor pam, a design issue between the artificial tendon and muscle cap interface became apparent. There is a point at which, upon being pressurized, the shoulder of the artificial tendon catches on the edge of the muscle cap. As the PAM continues to come to equilibrium, there is a point at which the artificial tendon "pops" into place. These two stages are shown below in Figure 18.

FIGURE 18: LEG ASSEMBLY DESIGN ISSUE, TENDON SETTLING WHILE PAM PRESSURIZES. SHOULDER OF TENDON CATCHING ON MUSCLE ENDCAP (LEFT). TENDON "POPPING" INTO PLACE (RIGHT).

3.3 Issues in the Testing Procedure

Testing Procedure Issue #1: Adding to code to ensure that data collection occurs through to system equilibrium

As it is, when Matlab commands Arduino to collect data from the load cell, a definite number of data points are collected. This method begs a question about whether the system has actually reached equilibrium at the point when data stops being collected.

A wonderful addition to the code would be an active interface where the user was allowed to collect data until the system has definitely reached equilibrium.

Testing Procedure Issue #2: Camera Calibration

The camera calibration process that was used for preliminary testing in March 2021 is outlined in Appendix 2. Using this process resulted in length measurement error in FIJI (around -10mm from the expected value). This could be because incorrect camera sensor information was used as input during the process of finding camera lens parameters.

The camera calibration process outlined in this document should be used as a starting point for other users of this test jig, but the calibration process needs to be improved before it can be incorporated with high confidence for image analysis.

3.4 Data Collected During Preliminary Testing

A small amount of data was collected during preliminary testing in March 2021. The torque profiles which were generated from this preliminary testing are shown in Figure 19.

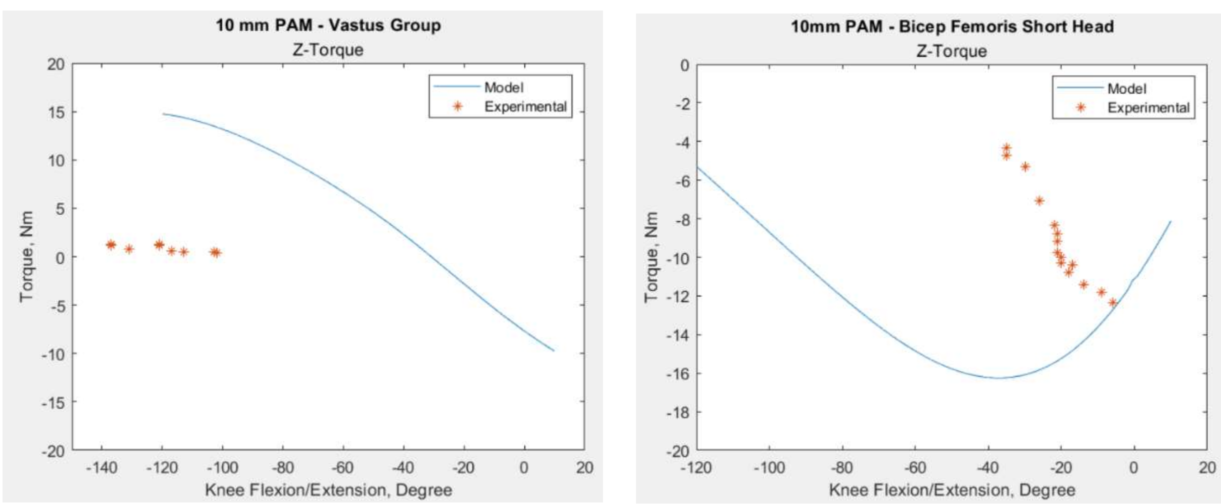

FIGURE 19: EXPERIMENTAL TORQUE PROFILES VS. MODEL TORQUE PROFILES. EXPERIMENTAL TORQUE PROFILES WERE GENERATED USING DATA COLLECTED DURING PRELIMINARY TESTING IN MARCH 2021.

Preliminary data shows that ultimately, this test jig will serve as an integral tool in verifying and refining the model which is currently used to predict torque profiles about the joints of the biomimetic humanoid robot. There are certainly issues that exist in the testing process which mean that there are problems when comparing experimental data with the model. For instance, during the preliminary tests of the extensor muscle (vastus group shown in Figure 19), data was only collected during the transient system response. Had data been collected from the system at equilibrium, the magnitude of experimentally produced torque values may have been more consistent with the model values.

Additionally, it is known that there were errors introduced during the camera calibration process used for gathering these preliminary data points. This means that when looking at results of from testing the flexor PAM (Figure 19, Bicep Femoris Short Head), there is not high confidence in the match between torque value and the angle at which that torque value exists.

However, there is value in comparing these preliminary experimental results with the model results. Since the torque components of Force and Moment Arm Length were gathered outside image analysis, it is promising to see that the range of torque magnitudes in the experimental results match relatively well with the torque magnitudes predicted by the model.

Additionally, the test jig is producing conclusive results already that the model will need to be developed. While the model predicts that the range of motion for the knee will be on +10° through -120° for the 10mm PAM at the exact origin and insertion points used for the leg assembly, the range of motion produced by the actual PAMs were much less than expected.

4. Conclusion:

A test jig has been designed and fabricated with the purpose of collecting true torque profiles resulting from the actuation of the PAMs about the knee of the bipedal humanoid robot. Preliminary data has been collected which shows proof of concept, and that there are design details for the test jig as well as the leg assembly and testing process that should be addressed before more robust data collection can take place.

Works Cited

[1] A. G. Steele, "Biomimetic Design and Construction of a Bipedal Walking Robot," p. 95.

[2] B. P. Bolen and A. J. Hunt, "Determination of Artificial Muscle Placement for Biomimetic Humanoid Robot Legs," in *Biomimetic and Biohybrid Systems*, vol. 11556, U. Martinez-Hernandez, V. Vouloutsi, A. Mura, M. Mangan, M. Asada, T. J. Prescott, and P. F. M. J. Verschure, Eds. Cham: Springer International Publishing, 2019, pp. 15–26.

[3] A. G. Steele, A. Hunt, and A. C. Etoundi, "Biomimetic Knee Design to Improve Joint Torque and Life for Bipedal Robotics," in *Towards Autonomous Robotic Systems*, vol. 10965, M. Giuliani, T. Assaf, and M. E. Giannaccini, Eds. Cham: Springer International Publishing, 2018, pp. 91–102.

[4] C. Morrow, B. Bolen, and A. Hunt, "Optimization of Artificial Muscle Placements for a Humanoid Bipedal Robot," presented at the Living Machines, Jul. 2020.

Appendix 1: Important Notes about Test Jig Construction

1. Femur Jig cross brackets and winch sled brackets are not interchangeable. Femur Jig Cross Brackets have been scored as shown in Figure 20 so that if it becomes necessary to disassemble the test jig, it can be reassembled.

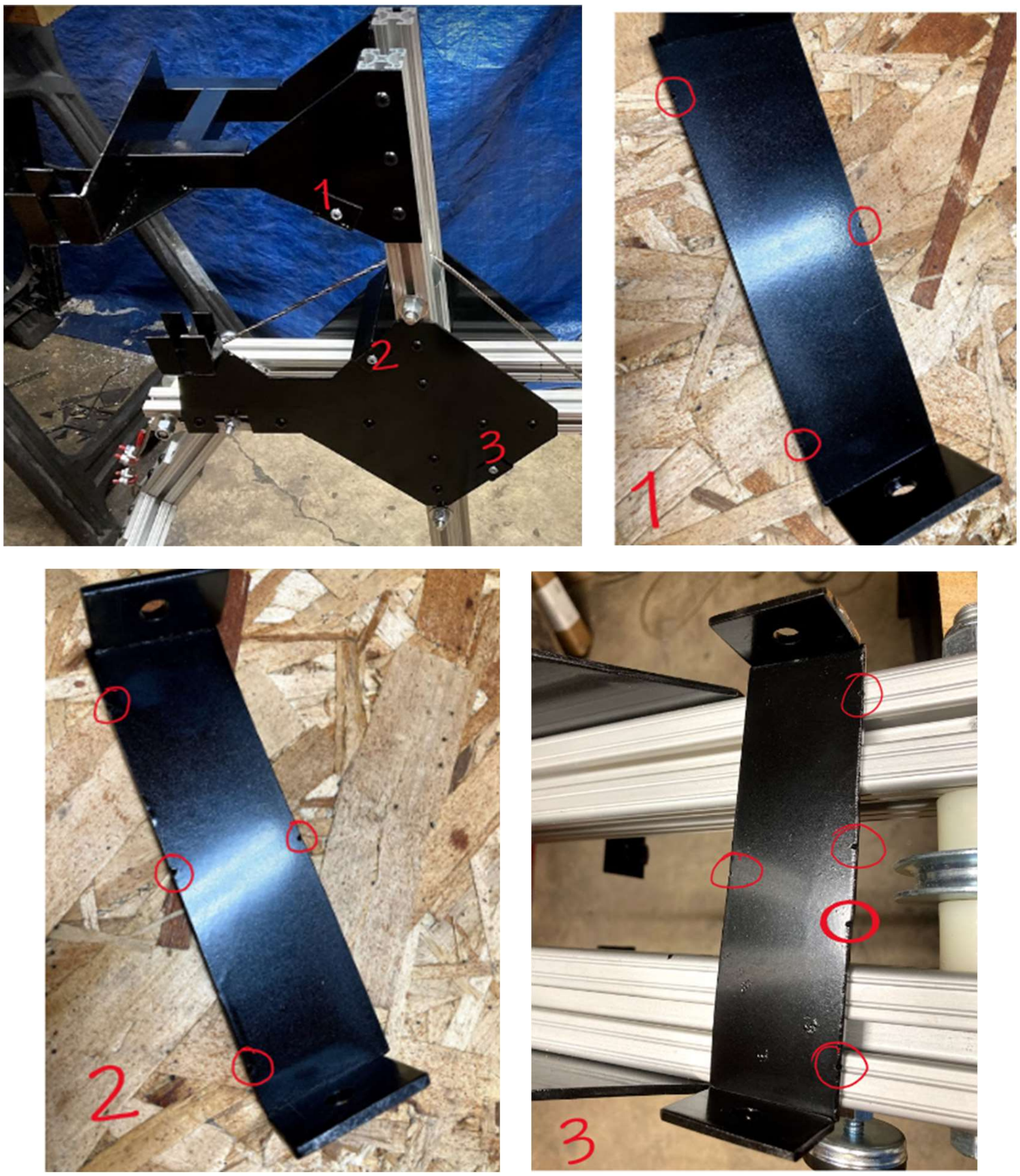

FIGURE 20: FEMUR JIG BRACKET POSITIONING (TOP, LEFT). THE SIDE OF THE BRACKET WITH A SINGLE NOTCH INDICATES THE SIDE WHICH IS CLOSEST TO THE WINCH END OF THE TEST JIG. THE OTHER SIDE OF THE BRACKET HAS A SPECIFIC NUMBER OF NOTCHES, WHICH INDICATE ITS POSITION ON THE FEMUR JIG ASSEMBLY.

2. The legs on the Test Jig (below the main horizontal of the main test frame) were intentionally left long. The distance between the center line on the main horizontal beam and the upper and lower pulleys, respectively, is the same. If more length is desired for the vertical members above the main horizontal so that, for example, a jig can be added from which to mount a pelvis, it should be simple to swap the lower vertical members for the upper members. Of course, the back legs will need to be shortened in the process.
3. In the future, it may become necessary to characterize the dynamic parameters of PAM acting about some joint. A good starting point for this exploration may be to swap out the winch for a controllable motor.

Appendix 2: Notes about Camera Calibration Process

Hugin can be downloaded here: http://hugin.sourceforge.net/download/

Hugin comes with a bunch of good user generated tutorials, including:

Simple lens calibration (http://hugin.sourceforge.net/tutorials/calibration/)

- This tutorial served as the basis for establishing a calibration tool and process for finding the lens parameters of the 2020 iPhone SE that collected preliminary data in March 2021.
- This tutorial describes the application of Hugin to obtain lens distortion parameters a, b, c. These parameters can be saved for reuse in future projects, or imported into a lens database such as lensfun.

Perspective Correction (http://hugin.sourceforge.net/tutorials/perspective/)

- This tutorial covers a basic use of Hugin — Using both horizontal and vertical control-points to remove all perspective effects from a photograph.

Appendix 3: Load Cell Calibration and Code

It is necessary to calibrate the load cell before performing testing. This section includes pertinent information about the calibration process and the associated code.

Sparkfun includes a vast amount of information about the HX711 Load Cell Amplifier, including example uses and tutorials, which can be found here: https://www.sparkfun.com/products/13879

It should be noted that the source code for the HX711 Load Cell Amplifier can be found here: https://github.com/bogde/HX711/tree/master/src

Finding the Zero Factor:
The Zero Factor is used to set the 0 value of the load cell. In permanent scale applications such as use in this Knee Torque Test Jig, setting the zero factor eliminates the need for taring the scale.

To find the Zero Factor, the load cell was hung in a horizontal configuration such that no axial load was induced upon the load cell. See Figure 21.

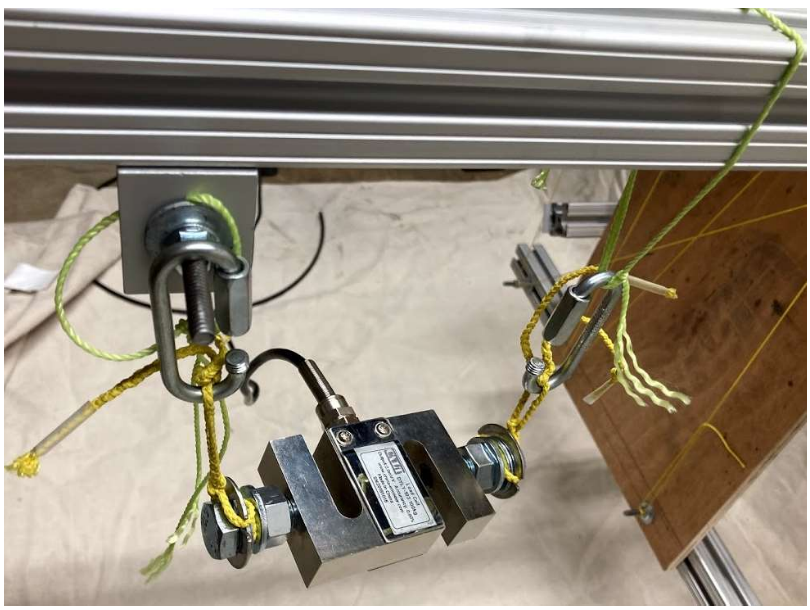

FIGURE 21: FINDING THE LOAD CELL ZERO FACTOR

The Matlab function "KneeTest.m", included in appendix 3.1 is used with the Arduino script "SLoadZero_FactorSketch", included in appendix 3.2 to find the Zero Factor. Using this set of code, data is collected at least 3 times in order to verify that an accurate Zero Factor is used.

A zero factor of -17280 was used for the preliminary torque tests performed in March 2021.

Finding the Calibration Factor:
The Calibration Factor is used to set the scale read by the load cell.

To find the Calibration Factor, a 15lb weight was connected over a pulley and to the load cell, as shown in Figure 22.

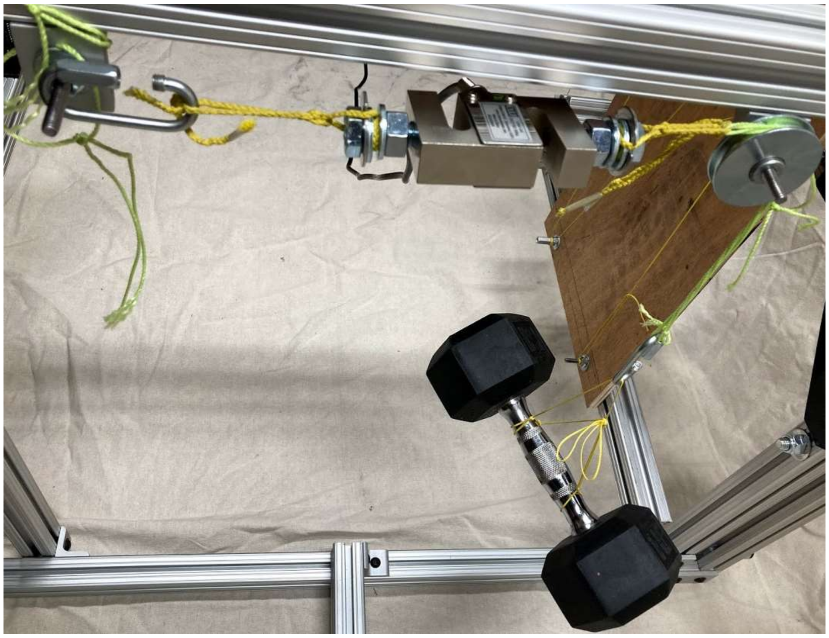

FIGURE 22: FINDING THE LOAD CELL CALIBRATION FACTOR

The Arduino script “SLoadCalibration_FactorSketch”, included in appendix 3.3 is used to find the Calibration Factor.

The calibration load is hung and then the calibration sketch is run. The calibration factor is adjusted until the force readout matches the calibration load. This process was repeated five times in order to ensure that the load cell accurately reads to within plus or minus 0.5 units of the calibration load, within the 5 seconds after being placed under load.

A calibration factor of -15400 was used for the preliminary torque tests performed in March 2021.

Appendix 3.1: KneeTest.m | Matlab

```
%This function is written to be used with 2 arduino sketches
% (1) SLoadCell_ZeroFactorSketch, when protocol_id == '1'
% (2) Knee TorqueTest, when protocol_id == '2'
% Minor Adjustments are needed to this function depending on which
% Arduino sketch it is being used with

function [Data, Stats] = KneeTest(protocol_id)

%Initialize serial port
s = serial('COM3', 'Baudrate', 9600);

%Open serial port s
fopen(s);

%Send 'protocol_id' over serial until there is data on the
%incoming buffer.  It was expected that only the
% fwrite(s,protocol_id) command should be needed, but when this is the
case, there seems to a % timing issue with Arduino.
%The While statement was constructed as a work around
while (s.BytesAvailable == 0)
    fwrite(s, protocol_id)
end


%prepare a cell array to receive ASCii data from the incoming buffer
total = 6000;
  %for protocol_id == '1', total == 150
  %for protocol_id == '3', total needs to be at least 6000 for the
  %system to reach equilibrium
svalues = cell (total,2);

tic %start timer
for i = 1:total

       svalues{i,1} = fgetl(s); %read information from buffer into
       %column 1
       svalues{i,2} = toc; %put a timestamp in column 2 for the
      %information read into column 1
end


%the following while loop converts the ascii values from column 1 of
%svalues into a numeric array
a = 1;
while ~isempty(svalues{a}) && a<total
        data(a,1) = str2num(svalues{a})*4.4482216; %%Convert lb to N
        data(a,2) = svalues{a,2};
        a = a+1;
end
```

```
%save data as function output Data
Data = data;

%Create a Force vs. time plot with the information in Data
Y = data(:,1);
X = data(:,2);
plot(X,Y,'.')
xlabel('Time (sec)')
ylabel('Force(N)')

%Use the following bit of code to find some basic statistics about the
%data that has been collected.  The operating assumption here is that
%collecting 6000 data points will be enough for the system to have
%reached equilibriam,and that the last 500 data points will exists
%outside of the transient system response.

% **NOTE: If protocol_id == '1', either comment out the following
%code,or change the range of data on which the basic statistics are
%calculated.
Stats = zeros(4,1);
Stats(1,1) = mean(data(5500:5999,1));
Stats(2,1) = median(data(5500:5999,1));
Stats(3,1) = mode(data(5500:5999,1));
Stats(4,1) = std(data(5500:5999,1));
Stats;

fclose(s)
```

Appendix 3.2: SLoadCell_ZeroFactorSketch | Arduino

```
/* sketch adapted by Lindie Burgess for use by the Agile and Adaptive Robotics Lab at Portland State
University.
This sketch is used in conjunction with Matlab function called, "KneeTest.m"
This arduino sketch is adapted from the "SparkFun_HX711_Calibration" example, which
can be found here
https://github.com/sparkfun/HX711-Load-Cell-Amplifier/tree/master/firmware
Source Code for the HX711 can be found here:
https://github.com/bogde/HX711/tree/master/src
*/

#include "HX711.h" //This library can be obtained here http://librarymanager/All#Avia_HX711
#define LOADCELL_DOUT_PIN 3 //define the Serial Data Output Pin
#define LOADCELL_SCK_PIN 2  // define the Power Down and Serial Clock Input Pin

HX711 scale;

void setup() {
Serial.begin(9600);  // initialize arduino serial communication
}

void loop()
{
  if (Serial.available()) // if information is sent over serial from matlab
  {  char choose_branch = Serial.read(); // read the serial data into variable "choose_branch"
     if (choose_branch == '1') // If choose_branch is equal to '1', iterate through the following for loop
     {  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN); // initialize load cell
        scale.set_scale(); //set the scale value to 1;
        scale.tare(); // sets OFFSET to 0
          for (int i = 0; i <150; i++)
          {
            long zero_factor = scale.read_average(); //read a baseline value into variable zero_factor
            Serial.println(zero_factor);   //print the variable zero_factor to serial
          }
     }

  }
  else  //if there is no information to read over serial from matlab, wait...
  {  while(Serial.available() <1);
  }
}
```

Appendix 3.3: SLoadCell_CalibrationFactorSketch | Arduino

```
/* sketch adapted by Lindie Burgess for use by the Agile and Adaptive Robotics Lab.

This sketch is NOT meant to be used with Matlab.  Instead, use Arduino's built in serial monitor.

This arduino sketch is adapted from the "SparkFun_HX711_Calibration" example, which can be found here:
https://github.com/sparkfun/HX711-Load-Cell-Amplifier/tree/master/firmware

Source Code for the HX711 can be found here:
https://github.com/bogde/HX711/tree/master/src

 */

#include "HX711.h" //This library can be obtained here http://librarymanager/All#Avia_HX711

#define LOADCELL_DOUT_PIN 3 //define the Serial Data Output Pin
#define LOADCELL_SCK_PIN 2  // define the Power Down and Serial Clock Input Pin

HX711 scale;

float calibration_factor = -15400; // initialize variable calibration_factor to some guess value

void setup() {

  Serial.begin(9600);  // initialize arduino serial communication
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN); // initialize load cell
scale.set_scale();
   // set scale to 1 units; this value is used to convert the raw data to "human readable" data (measure
   // units)
  scale.set_offset(-17280);
    //This sets the offset value to a known zero. There is no need for taring the scale once the zero point
    // is known for a scale in a set configuration.  The Zero Factor -17200 was found using a separate
    // Arduino Sketch called "SLoadCell_ZeroFactorSketch"
}

void loop() {
 //Protocol to read calibration_factor
	scale.set_scale(calibration_factor); //Adjust to this calibration factor

	Serial.print("Reading: ");
	Serial.print(scale.get_units(), 1);
	Serial.print(" lbs"); //Change this to kg and re-adjust the calibration factor if you follow SI units like a
sane person
	Serial.print(" calibration_factor: ");
	Serial.print(calibration_factor);
	Serial.println();
```

```
    if(Serial.available())
    // if the following characters are sent over serial, the calibration factor
    // will adjust as commanded
    {
      char temp = Serial.read();
     if(temp == '+' || temp == 'a')
        calibration_factor += 10;
     if(temp == '+' || temp == 'a')
        calibration_factor += 10;
      else if(temp == '-' || temp == 's')
        calibration_factor -= 10;
    }
}
```

Appendix 3.4: KneeTorqueTest |Arduino

```
/* sketch adapted by Lindie Burgess for use by the Agile and Adaptive Robotics Lab at Portland State University.

This sketch is used in conjunction with Matlab function called, "KneeTest.m"

This arduino sketch is adapted from the "SparkFun_HX711_KnownZeroStartup" example, which
can be found here
https://github.com/sparkfun/HX711-Load-Cell-Amplifier/tree/master/firmware

Source Code for the HX711 can be found here:
https://github.com/bogde/HX711/tree/master/src


Example using the SparkFun HX711 breakout board with a scale
 By: Nathan Seidle
 SparkFun Electronics
 Date: November 19th, 2014
 License: This code is public domain but you buy me a beer if you use this and we meet someday
(Beerware license).

Most scales require that there be no weight on the scale during power on. This sketch shows how to pre-load tare values so that you don't have to clear the scale between power cycles. This is good if you have something on the scale all the time and need to reset the Arduino and not need to tare the scale.

This example code uses bogde's excellent library: https://github.com/bogde/HX711
bogde's library is released under a GNU GENERAL PUBLIC LICENSE

The HX711 does one thing well: read load cells. The breakout board is compatible with any wheat-stone bridge based load cell which should allow a user to measure everything from a few grams to tens of tons.

 Arduino pin 2 -> HX711 CLK
 3 -> DOUT
 5V -> VCC
 GND -> GND

 The HX711 board can be powered from 2.7V to 5V so the Arduino 5V power should be fine.
 */

#include "HX711.h" //This library can be obtained here http://librarymanager/All#Avia_HX711

#define LOADCELL_DOUT_PIN 3  //define the Serial Data Output Pin
#define LOADCELL_SCK_PIN 2  // define the Power Down and Serial Clock Input Pin

HX711 scale

int choose_branch = 1; // initialize the variable "choose_branch"
```

```
float calibration_factor = -15400;
  // initialized variable calibration_factor.  Calibration factor found using a separate arduino sketch
  // called SLoadCell_CalibrationFactorSketch"

void setup() {

 Serial.begin(9600);  // initialize arduino serial communication
 scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN); // initialize load cell
 scale.set_scale(calibration_factor);
  //set the scale value;  this value is used to convert the raw data to "human readable" data.  Output
  // units are in lbs
 scale.set_offset(-17200);
  //This sets the offset value to a known zero. There is no need for taring the scale once the zero point
  // is known for a scale in a  set configuration.  The Zero Factor -17200 was found using a separate
  // Arduino Sketch called "SLoadCell_ZeroFactorSketch"
}

void loop()
{
 if (Serial.available()) // if information is sent over serial from matlab
 {  char choose_branch = Serial.read(); // read the serial data into variable "choose_branch"
    if (choose_branch == '2')  // If choose_branch is equal to '2', iterate through the following for loop
    {  for (int i = 0; i <6000; i++)
      {    Serial.println(scale.get_units(), 1); //scale.get_units() returns a float
      }
    }
 }
 else  //if there is no information to read over serial from matlab, wait...
 {  while(Serial.available() <1);
 }
}
```

Appendix 4: Post Processing

4.1 Saving Images

Image format is of high importance when it comes to image analysis. JPEG uses a lossy form of compression and is not a preferred file type. TIFF and PNG work well.

The 2020 iPhone SE saves photos is the HEIC file type. For the preliminary testing that took place in March 2021, the following steps were taken from image collection to post processing in FIJI

1. Photos are converted from HEIC file type to PNG via tools onboard the iPhone: [https://www.guidingtech.com/convert-heic-images-to-png-iphone/#:~:text=Select%20Photos%20%2D%20expand%20action%20and,selected%20an%20album%20named%20PNG)](https://www.guidingtech.com/convert-heic-images-to-png-iphone/#:~:text=Select%20Photos%20%2D%20expand%20action%20and,selected%20an%20album%20named%20PNG)
2. PNG photos are uploaded to Google Photos and then downloaded to a local drive.
3. Hugin is used to apply corrections for Lens Parameters. Output is saved as TIFF.
4. TIFF is uploaded to FIJI for image analysis

4.2 Applying Corrections for Lens Parameters in Hugin

Each image collected during testing is adjusted by Hugin before being uploaded in Fiji, an image analysis software. The following brief tutorial demonstrates the process for applying a previously saved file (.ini type) containing lens parameters to images intended for image analysis:

1. Open Hugin
2. Load Image
    a. If there is not data in the MACROS about your camera, the popup window shown in Figure 23 called “Camera and Lens data” will appear

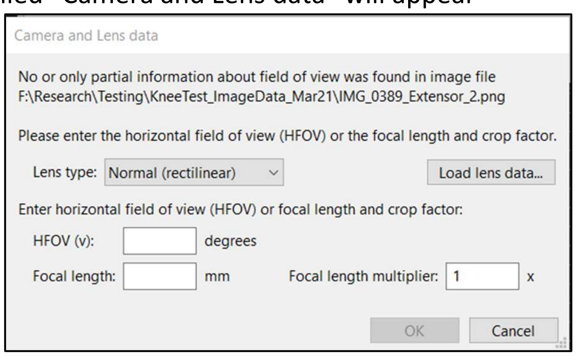

FIGURE 23: CAMERA AND LENS DATA POP-UP (HUGIN)

    b. Ensure that the Lens type is “Normal (rectilinear)”
    c. Select “Load lens data” and add the .ini file that contains your Camera Lens Calibration parameters

3. In the “Projection” tab, select “Rectilinear” from the drop down menu as shown in Figure 24

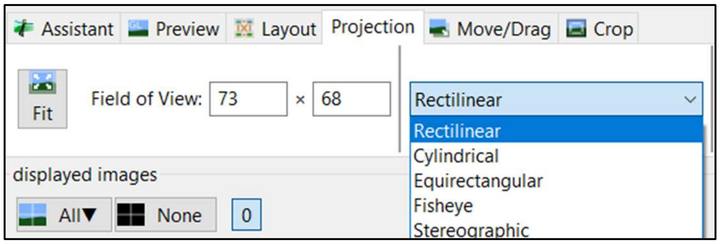

FIGURE 24: SETTING OUTPUT PROJECTION TYPE IN HUGIN

4. In the “Assistant” tab, select “2. Align”
    a. Adjust the boundaries of your photo as necessary, to encapsulate all pertinent data

5. Again, in the “Assistant” tab, select “3. Create panorama”
    a. The Assistant Wizard will walk you through saving the necessary files. Keep all suggested settings through this process.

4.3 FIJI

Fiji (Fiji Is Just ImageJ) is the open source software used to collect angle data from the imaging collected during testing. Fiji is an image processing package – a “batteries-included” distribution of ImageJ, bunling a lot of plugins which facilitate scientific image analysis.

Fiji can be downloaded here: https://imagej.net/Fiji.html#Downloads

4.4 Notes about Setting the measuring scale in FIJI

The current scale indicators on the Femur and Tibia (Figure 25) don’t work well for setting a scale because a definitive starting point and ending point on these indicators cannot be accurately selected in FIJI. Additionally, because these indicators are relatively small compared to length measurements of interest, any error introduced while using these indicators to set scale is amplified during measurements.

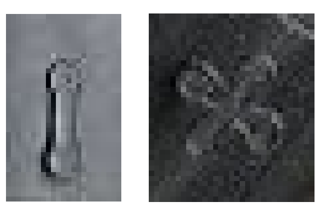

FIGURE 25: SCALE INDICATORS FOR FEMUR (LEFT) AND TIBIA (RIGHT)

To correct for this, either:

1. Use Tibia (200mm) to set the scale.
2. Print out a scale to be affixed to the tibia. The length of this scale should be 15-20 cm.