Effect of the Supraspinatus and Coracohumeral Ligament on Humeral Motion and Tendon Strain

by

Justin Gabriel Buce

Bachelor of Science in Mechanical Engineering, University of Pittsburgh, 2021

Submitted to the Graduate Faculty of the

Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science in Mechanical Engineering

University of Pittsburgh

2023

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Justin Gabriel Buce

It was defended on

April 7, 2023

and approved by

Mark C. Miller PhD, Associate Research Professor, Mechanical Engineering and
Materials Science

Jeen-Shang Lin, Associate Professor, Civil and Environmental Engineering

Thesis Advisor: Patrick Smolinski PhD, Associate Professor, Mechanical Engineering and
Materials Science

Effect of the Supraspinatus and Coracohumeral Ligament on Humeral Motion and Tendon Strain

Justin Gabriel Buce, M.S.

University of Pittsburgh, 2023

The supraspinatus (SS) is a rotator cuff (RC) muscle primarily responsible for initiating abduction and stabilizing the glenohumeral joint. It is a bipennate tendon, meaning it can be split into an anterior cord and posterior strap. Anterior to the SS is the coracohumeral ligament (CHL) which is also responsible for stabilizing the shoulder. A biomechanical cadaveric study was performed to determine the effects of these structures on humeral head translation, tendon strain, and tear gap width. It was hypothesized that simulated tears of these structures would cause humeral head migration, and that tendon strain would increase or decrease according to the loading patterns.

Twenty cadaveric specimens were loaded with physiological loads in a shoulder simulator and tested at 0° and 30° of shoulder abduction. All twenty specimens underwent five different loading cases with the loads of the SS cord and SS strap varied. Following the loading cases, the specimens were evenly divided into a CHL and SS cord first cut groups where the tendon was physically released. Humeral head translation and tendon strain in the SS was measured for the loading cases, and the same measurements were made with the cutting cases in addition to tear gap width size.

The humeral head did not significantly translate within the glenoid for all loading and cutting cases, and translations were less than 2 mm. Tendon strain changed according to the

loading cases, but strain in the tendon did not significantly increase or decrease when compared to each other. The tear gap width became significantly large once the entire SS was released.

Although patterns were seen in humeral head translation and tendon strain, there were no significant changes. This does not support the claim that the SS stabilizes the glenohumeral joint and that tendon strain would significantly change according to tendon load. While the results imply that these structures can be treated conservatively clinically unless the entire SS is torn, more work is needed to determine the full biomechanical responsibilities of the SS and CHL.

Table of Contents

List of Tables

List of Figures

Preface

I would like to thank both of my advising professors, Dr. Mark Miller and Dr. Patrick Smolinski, and Dr. Christopher Schmidt for their guidance, leadership, and academic support. This research opportunity offered by them has been invaluable, and I am thankful for their help. Special thanks to my fellow researchers and friends I made along the way: Austin Cook, Sean Cooke, Dr. Joshua Dworkin, and Dr. Omar Rodriguez. I am glad to have worked with them and wish them the best in their future endeavors. I would also like to thank my family and friends for supporting me in my academic career.

1.0 Introduction

1.1 Motivation

The rotator cuff (RC) is a group of muscles and tendons that surround the shoulder joint, providing stability and arm movement. A RC tear is a common injury that can occur due to traumatic events, repetitive overhead movements, or degenerative changes in the tendon. Risk factors for developing RC tears include age, frequent overhead lifting or throwing, a history of shoulder injuries, and genetic factors[3,4]. RC tears can range from mild to severe, and symptoms may include pain, weakness, stiffness, and limited range of motion in the shoulder. The majority of tears start in the anterior portion of the RC, and small tears can start out asymptomatic[5]. Tears can propagate and increase in size overtime to surrounding tendons and ligaments which lead to symptoms. Conservative treatments such as rest, physical therapy, and anti-inflammatory medications may be effective in managing mild size tears. However, severe tears may require surgical intervention to repair the tendon. RC tears can lead to long-term shoulder dysfunction and disability if left untreated.

1.2 Goals

Determining the effects of RC tear size and location can have clinical impacts on patient care and treatment. This study focuses on the anterior portion of the RC, which consists of the supraspinatus (SS) and coracohumeral ligament (CHL). The SS was also divided into an anterior

1

cord and posterior strap sections[15]. Applying different muscle loading conditions was used to compare simulated RC tears between the SS cord and SS strap. Physical transection of tendon/ligament off of bone was used to compare simulated tears between the SS cord and CHL. Humeral movement relative to the scapula and strain within the SS tendon were the measured outcomes, and they were measured through the use of a motion tracking system. The clinical relevance of RC tears can be found by analyzing the effects of simulated RC tears in the SS and CHL on humeral movement and tendon strain.

2.0 Background

2.1 Anatomic Definitions

Anatomic definitions are a standard way of describing the position and orientation of the human body. They are important for clear and accurate communication among healthcare professionals and for precise instructions in medical procedures. In anatomic position, the body is standing upright, facing forward, with the arms at the sides and the palms facing forward (Figure 1). Anterior and posterior describes the front and back of the body, respectively. Distal describes portions further from the body or a structure whereas proximal is closer. Medial is closer to the mid-line of the body and lateral is further. Superior describes higher up the body while inferior is lower. Deep is further away from the surface of the body and superficial is towards the surface. The coronal plane describes a plane oriented through the sides of the human body, the transverse through the mid-section, and the sagittal plane from front to back (Figure 2).
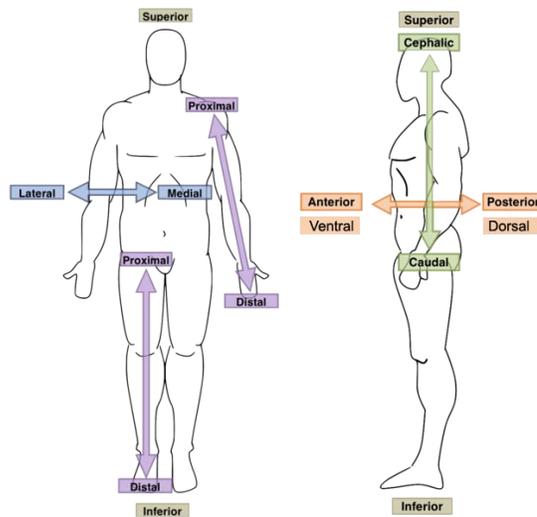
Figure 1: Anatomic directions (https://www.coursehero.com/study-guides/ap1x94x1/anatomical-orientation-and-directions/)
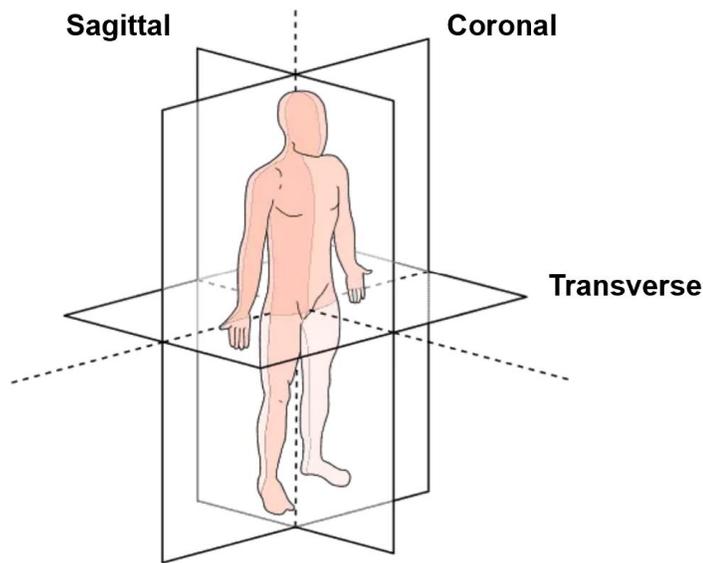
Figure 2: Anatomic planes (https://human-memory.net/anatomical-planes-of-body/)

2.2 Shoulder Anatomy

The shoulder is a ball and socket joint, which means that rotation is allowed in all directions. The scapula and the humerus are the two main bones that make up the shoulder joint (Figure 3). The humeral head, which is the superior portion of the humerus, is the ball that rotates within the joint. The glenoid, which is the lateral portion of the scapula and medial to the humerus, is the socket that holds the humeral head.
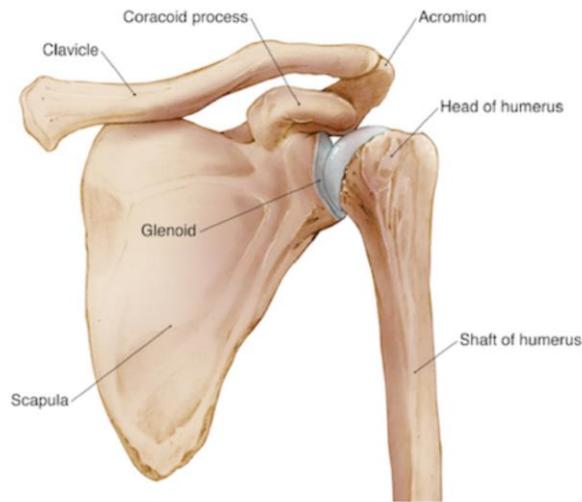
Figure 3: Shoulder skeletal anatomy (https://anatomy.lexmedicus.com.au/collection/shoulder)

The main anatomic shoulder movements are flexion/extension, abduction/adduction, and internal/external rotation (Figure 4). Flexion is movement of the arm forward and anterior to the body, while extension is in the opposite direction. Abduction is lifting of the arm lateral to the body whereas adduction is bringing the arm medial to the body. Internal and external rotation describe the rotation of the arm axially with the respective directions shown in Figure 4.
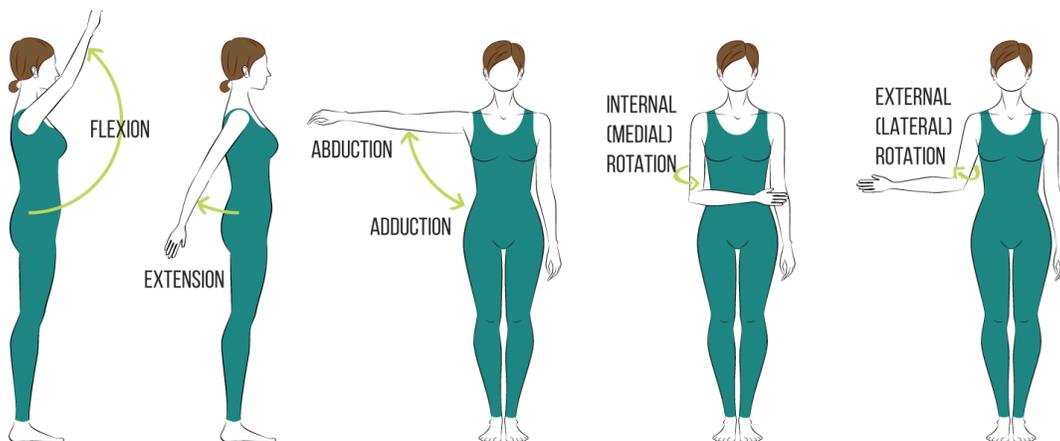
Figure 4: Anatomic shoulder movements (https://sequencewiz.org/2016/03/16/loosen-up-your-shoulders/)

The RC is comprised of four muscles: subscapularis (SC), supraspinatus (SS), infraspinatus (IS), and teres minor (TM)[2]. These muscles are primarily responsible for the anatomic shoulder movements shown in Figure 4, and they are found deep to the deltoid which spans the glenohumeral joint and is shown in Figures 5. The SS is known to be responsible for initiating humeral abduction as well as stabilizing the glenohumeral joint[4,17,20]. The IS and TM are responsible for external rotation, and the SC is responsible for internal rotation.

The SS is bipennate muscle, which means it consists of two distinct structures with muscle fibers going in different directions (Figure 6A and 6B). A previous study focused on the SS architecture described these structures as one a thicker, anterior cord-like structure and the other a flatter, posterior strap-like structure[15]. The mechanical contribution of the SS cord and SS strap as individual units is unknown, and because of this was one of the reasons for this study.

The coracohumeral ligament (CHL) is a broad supportive structure that is anterior to the SS and posterior to the SC. Since it is a ligament, the CHL is thought to not bear any force load and instead contribute to stabilizing the glenohumeral joint. Due to the proximity of the CHL to the SS, they are often blended together on their insertion to the humerus. The CHL is also often torn for RC tears that start anteriorly.
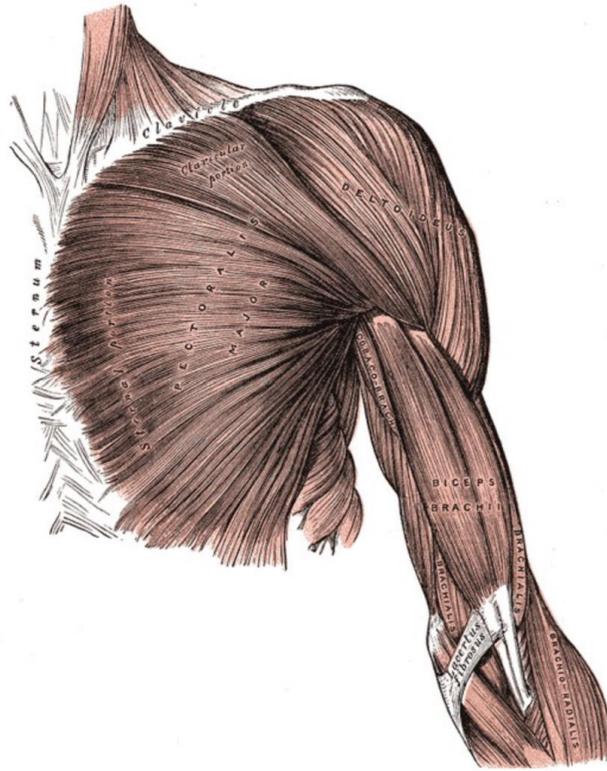
Figure 5: Anterior view of deltoid[11]
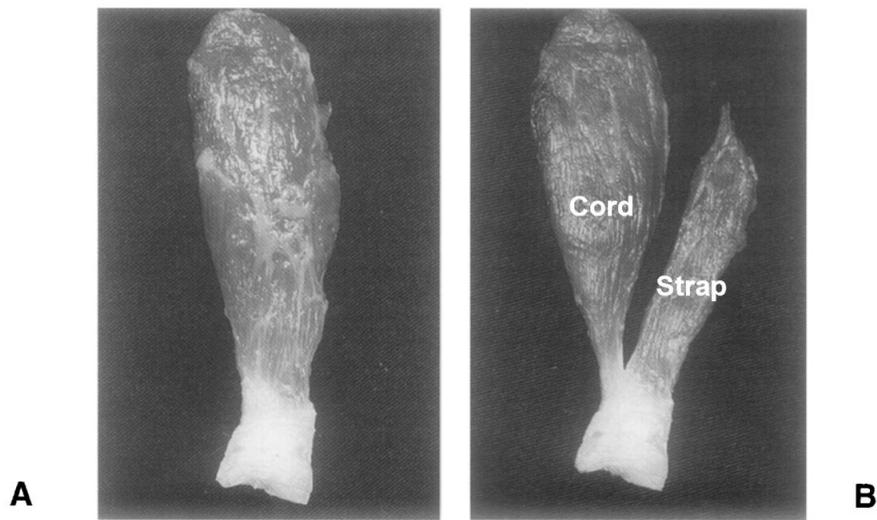
Figure 6A and 5B: Dissected supraspinatus tendon and muscle

5A shows the musclotendinous unit as a whole. 5B shows the cord and strap separated[15]

2.3 Biomechanics of the Rotator Cuff

Mechanical and clinical studies show that anterior supraspinatus tendon tears are associated with increased glenohumeral translation and decreased abduction motion compared to posterior supraspinatus tendon tears. Measuring humeral head movement and strain in the tendon can quantify the humeral head translation and force transmission throughout the tendon.

2.3.1 Coordinate Systems

Coordinate systems are used to identify positions of points in space or relative to a body. A three-dimensional cartesian coordinate system can be created with three unique points by defining two vectors from these three points. Normalizing these vectors into unit vectors and calculating the cross-product creates a third orthogonal vector. Normalizing this new orthogonal vector and taking the cross-product with one of the first two unit vectors creates a second orthogonal vector, and repeating the cross-product and normalizing step with the two new found unit vectors creates a third orthogonal vector. These three resulting orthogonal unit vectors in matrix format define a three-dimensional coordinate system. Putting these vectors in a matrix creates a transformation matrix, which is also known as a rotation matrix. Multiplying a point by the rotation matrix and adding any displacements will transform the point from its original coordinate system to the new coordinate system defined by the rotation matrix. These steps are outlined in Figure 7A and 7B.

A global coordinate system defines points in space, while a local coordinate system defines points relative to a body or an object. Local coordinate systems are dependent on the position and rotation of the object defined. Coordinates from one coordinate system can be

transformed to another coordinate system by multiplying the matrices together, with importance on order of multiplication. The use of multiple coordinate systems is useful in tracking position, translation, and rotation of points relative to different bodies (Figure 8).

Step 1: Create vectors
$$P_1 = [x\ y\ z] \quad P_2 = [x\ y\ z] \quad P_3 = [x\ y\ z]$$
$$V_1 = P_2 - P_1 = [i\ j\ k] \quad V_2 = P_3 - P_1 = [i\ j\ k]$$

Step 2: Normalize vectors
$$V_1, = \frac{V_1}{|V_1|} \qquad V_2, = \frac{V_2}{|V_2|}$$

Step 3: Create first orthogonal unit vector
$$V_1, \times V_2, = V_3 \qquad V_i = \frac{V_3}{|V_3|}$$

Step 4: Create second orthogonal unit vector
$$V_i \times V_2, = V_4 \qquad V_j = \frac{V_4}{|V_4|}$$

Step 5: Create third orthogonal unit vector
$$V_i \times V_j = V_5 \qquad V_k = \frac{V_5}{|V_5|}$$

Step 6: Combine vectors and create rotation matrix
$$R = \begin{bmatrix} V_i \\ V_j \\ V_k \end{bmatrix} = \begin{bmatrix} i_1 & j_1 & k_1 \\ i_2 & j_2 & k_2 \\ i_3 & j_3 & k_3 \end{bmatrix}$$

A

Original Point   Rotation Matrix   Displacement
$$P_{original} = [x\ y\ z] \qquad R = \begin{bmatrix} i_1 & j_1 & k_1 \\ i_2 & j_2 & k_2 \\ i_3 & j_3 & k_3 \end{bmatrix} \qquad d = [x_d\ y_d\ z_d]$$

$$P_{new} = R * P'_{original} + d$$

$$P_{new} = [x_{new}\ y_{new}\ z_{new}]$$

B

Figure 7A and 7B: Equations for coordinate system transformation

7A: Steps to create rotation matrix from three unique points. 7B: Transforming a point from one coordinate system to another

Global Coordinate System

Scapular Coordinate System

Humeral Coordinate System
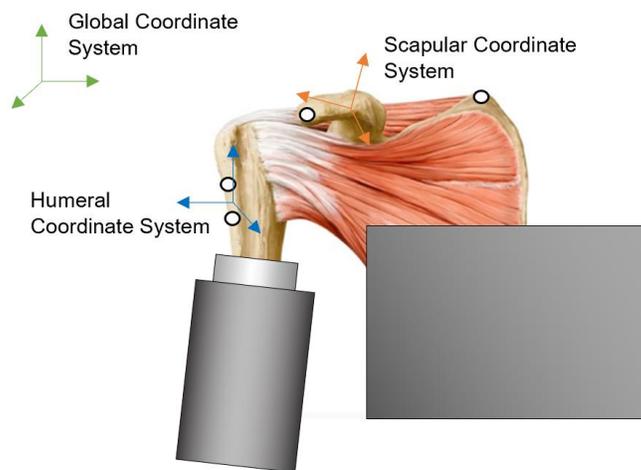
Figure 8: Multiple coordinate systems on a specimen diagram

2.3.2 Joint Kinematics

Posterior-superior migration of the humeral head is a common outcome of RC tears, and movement of the humerus out of the glenoid socket can cause pain and limit mobility[7,12,14,19]. Measuring the movement of the humeral head relative to the glenoid was done in previous studies by focusing on the humeral head apex (HHA)[9,12,14]. The HHA is defined as the most prominent point with respect to being the geometric center of the humeral head. A local coordinate system was created for the humerus, and the position of the HHA was found with respect to the humeral head (Figure 9). Similarly, a local coordinate system was also created for the glenoid fossa. Points along the perimeter edge of the glenoid were identified and the centroid of the glenoid was identified (Figure 10). A similar procedure was used in this study.
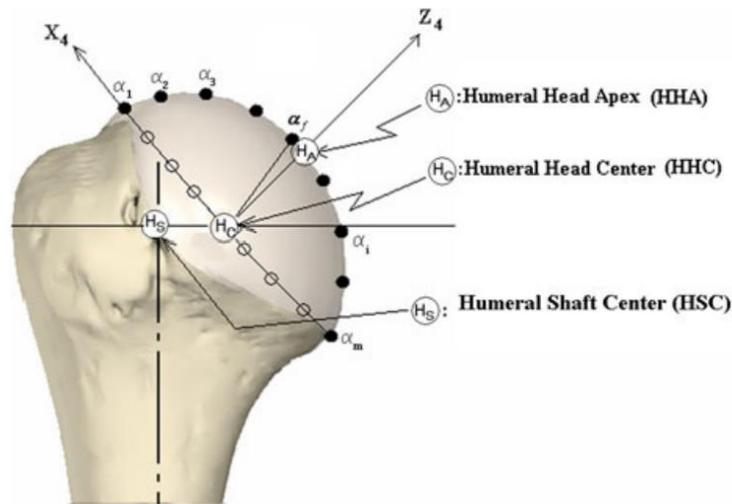
Figure 9: Humeral head local coordinate system[9]
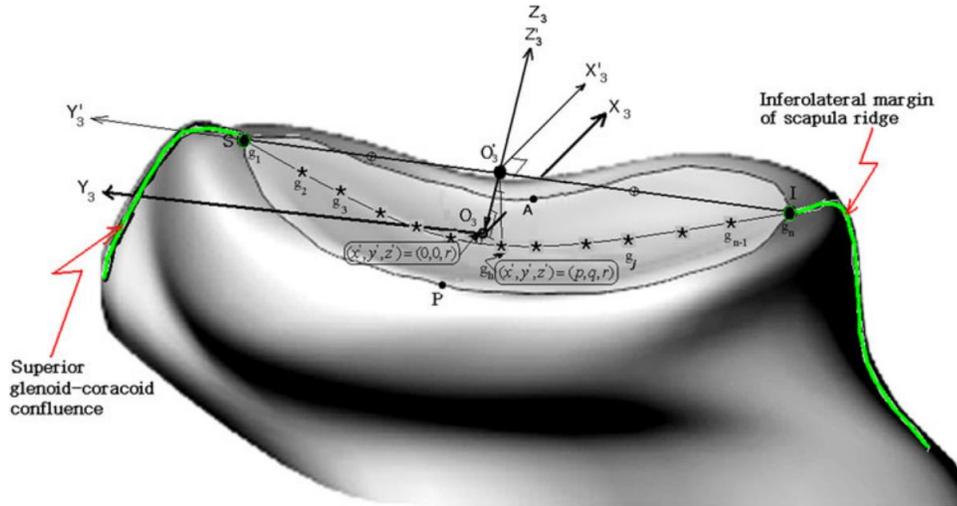
Figure 10: Glenoid local coordinate system[9]

2.3.3 Tendon Strain

Strain describes the deformation in a body in the direction of the force applied. It is defined as the change in length over the original length. Measuring strain within a tendon is useful in determining if the tendon is under tension and therefore under load. Previous biomechanical studies focused on simulating SS tears with strain as the primary measured output have found conflicting results on whether there is a significant increase[18]. These studies also used more detailed methods for measuring strain such as digital image correlation (DIC) whereas in this study a different method was used.

3.0 Methods

3.1 Overview

Twenty fresh-frozen cadaveric specimens were used in this study. Each specimen was dissected down to bone leaving the RC muscles, and markers were placed onto the SS tendon, scapula, and humerus. Specimens were fixed into a shoulder simulator, and all the RC muscles were cyclically loaded with physiological forces. Specimens underwent four different loading cases where the SS cord and SS strap were variably and randomly loaded. Following the loading cases, all twenty specimens were randomly and equally divided up into a CHL first-cut group and SS cord first-cut group. The cuts for these cut cases were performed by a trained orthopedic fellow. Sequential release of the remaining RC tendons followed each first cut. The position of the markers was tracked during loading and cut cases, and the variables measured were glenohumeral translation and tendon strain. These measures were made at 0° and 30° of abduction, in which the supraspinatus was the main shoulder abductor.

3.2 Specimen Preparation

Forty-five fresh-frozen human cadaveric arms from scapula-to-fingertip were procured for this study. Any specimen that was determined to have glenohumeral arthritis and/or RC tears by a trained orthopedic fellow were excluded from the study, leaving a total of twenty specimens with average age 69±11 years (9 males). For each specimen, the humerus was osteotomized 5 cm

distal to the deltoid insertion. The deltoid and other soft tissue was transected off, with each specimen being dissected down to the level of the RC structures. The clavicle and acromion were osteotomized from the scapula in order to better view and access the muscle insertions.

The anteroposterior insertion borders of the SS, IS, and CHL were identified under loupe magnification and outlined with a waterproof marker (Figure 11A). The SS insertion was split into SS cord and SS strap insertions. After identification of each RC muscle and CHL, a locking #2 braided suture was sown proximal to the musculotendinous junction of the following RC structures: SS cord, SS strap, IS, TM and upper and lower SSc. The SSc was split into two sections due to its larger muscle mass relative to the other RC muscles. All muscle medial to the sutures were transected off. Eyelet screws were drilled and secured into the scapula for the SS cord, SS strap, IS, and upper SSc. The respective sutures were fed through the eyelet screws to ensure correct anatomic lines of action. The scapula was fixed into a custom-built fixture using bolts and polyester resin (Bondo, 3M), and the humerus was fixed into a PVC pipe with the same polyester resin. The specimen was removed from the custom-built fixture and secured into the shoulder simulator. Each of the six sutures were connected to their respective individual actuators (Parker Hannifin Corp., Cleveland, OH) using low friction loading cables. Eight custom white spherical markers (four on the humerus and four on the scapula) were fixed to each specimen before mechanical testing and were used to track motion of the scapula and humerus (Figure 10). The four markers on the humerus were fixed below the humeral head, two on the anterior side and two on the posterior side. For the scapular markers, two markers were placed on the acromion, one on the clavicle, and one on the scapular ridge. Three rows of five black spherical markers were glued to the center of the SS cord, SS strap, and their adjacent border to measure tendon strain in each respective structure (Figure 11B).
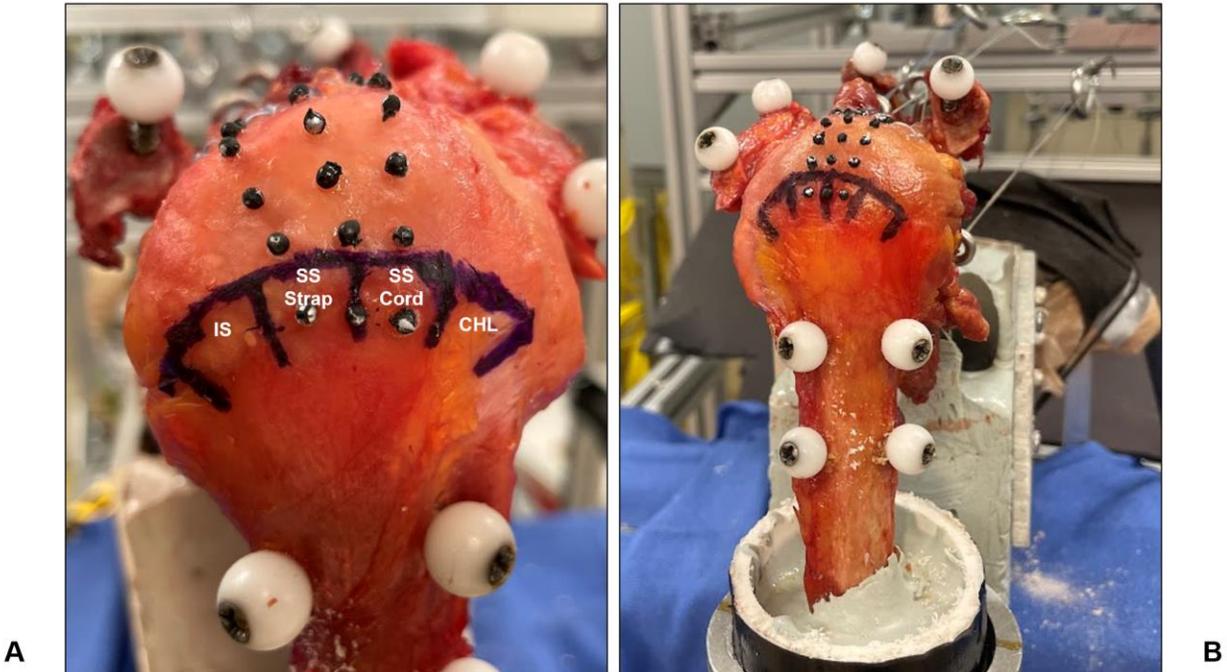
Figure 11A and 11B: Prepared specimen

11A showing outlined tendons with IS, SS strap, SS cord, and CHL marked. 11B showing placement

of motion tracking markers

3.3 Shoulder Simulator

The custom-built shoulder simulator was used to apply simulated abduction forces via the aforementioned sutures (Figure 11). The aluminum frame of the shoulder simulator consisted of multiple T-slot aluminum bars (80/20®). Within the frame were six servo actuators with each actuator connected to a single-DOF load cells (MLP-100, Transducer Techniques, Inc., Temecula, CA). The sutures were attached to the actuators with cables, and the placement of the cables were adjusted through the use of pulleys. This allowed for precise loading of the six distinct RC structures. The abduction angle could be variably fixed using the custom-made arc

attached to the simulator. Each specimen was positioned within the machine so that the glenohumeral joint was appropriately placed at the center of the arc. The humerus was fixed only in abduction and flexion, but was free to rotate axially.
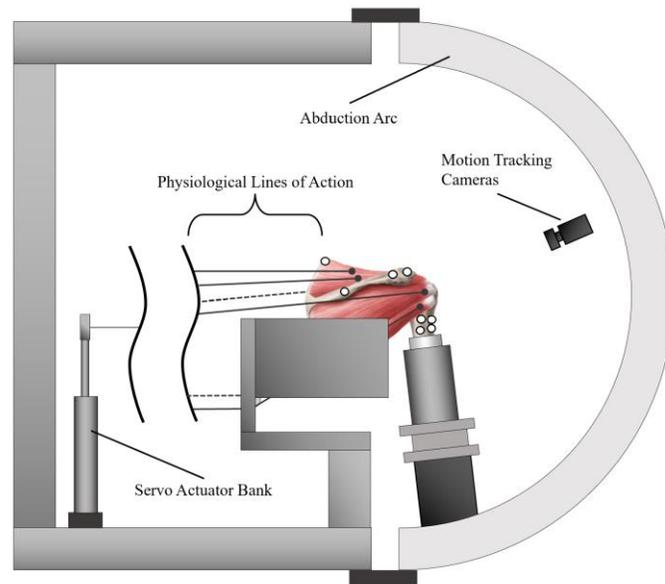
Figure 12: Shoulder Simulator Diagram

3.4 Specimen Testing

Specimen testing protocol began with centering the humeral head in the glenoid and confirming position through C-arm imaging. Once centered, each of the six RC structures were preloaded to 10N, and the preload was maintained for five minutes to eliminate hysteresis. Each specimen underwent cyclic loading from 10N to physiological load with a frequency of 0.25 Hz. The physiological load was based on muscle cross-sectional area and electromyographic activity (Table 1)[6,13]. Four complete cycles were run, with output values measured at the peak of the

fourth cycle. This was done to ensure that the muscles did not undergo excessive sustained loading and to eliminate hysteresis from the load cell readings.

Table 1: Physiological Loads for Rotator Cuff structures

Muscle	SS Cord	SS Strap	IS	TM	Upper SSc	Lower SSc
Actuator Load	56 N	24 N	90 N	97 N	127 N	108 N

This procedure was repeated for both the loading cases and cut cases. The SS cord and SS strap force values were varied for the five loading cases, and can be seen in Table 2. The IS, TM, upper SSc, and lower SSc for values were kept the same for each loading case. The order of the trials was randomized (Appendix A.1). Cases 1 and 3 simulate removal of the entire musculotendinous unit with no load transfer to the other tendon. Cases 2 and 4 simulate a partial SS tendon tear with full load transfer of the entire muscle force into the remaining intact tendon.

Table 2: Loading Cases

Loading Case	SS Cord Load (N)	SS Strap Load (N)
Native	56	24
Case 1	0	24
Case 2	0	80
Case 3	56	0
Case 4	80	0

The cut cases followed the loading cases, and all 20 specimens were randomly divided into two groups: a CHL first-cut group and a SS cord first-cut group (Table 3) (Appendix A.1). The CHL-cut first group was released in the order of CHL, SS cord, SS strap, IS. The SS cord-

cut first group was released in the order of SS cord, CHL, SS strap, IS. All loading cases and cut cases were performed at 0° and 30° of abduction, with abduction fixed on the custom arc.

Table 3: Cutting Cases

First Cut Group	Cut 1	Cut 2	Cut 3	Cut 4
CHL-first	CHL	SS cord	SS strap	IS
SS cord-first	SS cord	CHL	SS strap	IS

3.5 Motion Tracking System

A two-camera motion tracking system (Spicatek, SPICA Technology Corp., Fox Island, WA) (accuracy 0.2 mm) was used to track the movement of the humeral, scapular, and tendon markers. The two cameras were mounted on tripods facing down on the humeral head of the mounted specimen so that all the humeral and scapular markers were visible to both cameras. The motion tracking system was calibrated for each specimen set-up to create a three-dimensional global coordinate system. Images of the markers were recorded at 10 Hz during testing. The captured videos were analyzed within the motion tracking system to output three-dimensional coordinates of the markers in a global cartesian coordinate system over the duration of each test.

3.5.1 Humeral Head Apex Translation and Rotation

The white humeral and scapular markers were used to track HHA translation relative to the glenoid. Following testing of the loading and cutting cases, the RC tendons were transected off both the humerus and the scapula, revealing the articular side of the humeral head and the glenoid fossa. The humerus was placed in a separate area and the cameras were readjusted and recalibrated so that the humeral head was in view. The HHA was identified, and a custom pointer was used to mark the position of the HHA while the motion tracking system recorded (Figure 13). A similar procedure was followed for the scapula to mark sixteen points along the perimeter of the glenoid, and the protocol is outlined in Table 4 and shown in Figure 14A,B and C.

Table 4: HHA translation and rotation motion tracking protocol

Steps	Description
1	Track all humeral and scapular markers during loading and cutting cases
2	Transect tendons and separate humerus and scapula
3	Point to HHA, track pointer and humeral markers
4	Point to glenoid perimeter, track pointer and humeral markers

The position of the markers for the pointer, humerus, and scapula was tracked using the motion tracking system. The three-dimensional coordinates from the motion tracking system were inputted into a custom MATLAB function to create local coordinate systems and therefore transformation matrices for the pointer, humerus, and scapula (Appendix A.2). Local coordinate

systems were able to be created because the markers were fixed to each rigid body. The MATLAB function followed a similar procedure previously shown in Figure 7. The inputs for the function were the coordinates of the markers of the two structures transforming to and from, and the outputs were the rotation matrix and displacement vector.

A series of coordinate transformations between the local and global coordinate systems of the pointer, humerus, scapula, and motion tracking system was performed using a MATLAB script (Appendix A.3). All points were transformed into the scapular coordinate system to determine the marked points from the pointer relative to each other in a singular coordinate system. The equations used and order of transformations are shown in Figure 15. The centroid of all sixteen glenoid points was calculated, and the HHA was found relative to the centroid of the glenoid. Translation and rotation of the HHA was calculated for all loading cases and cutting cases relative to the respective native condition at both 0° and 30° of abduction.

Figure 13: Custom pointer with array of markers and pointer tip
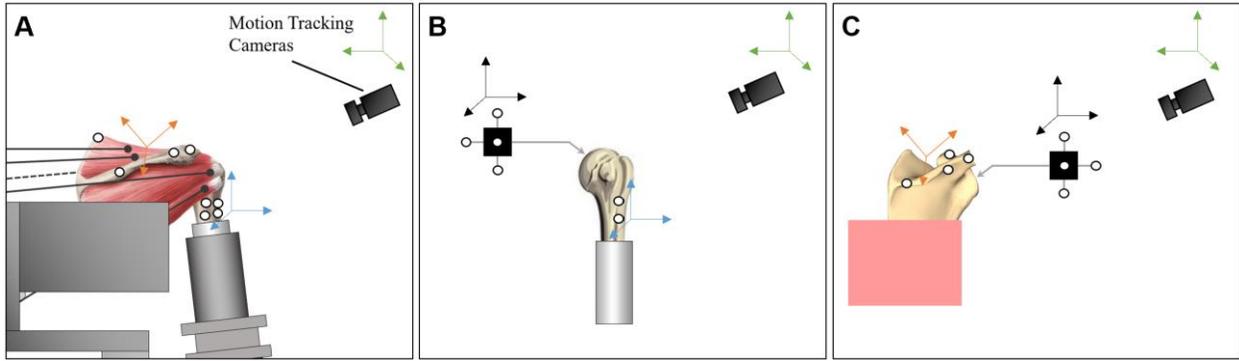
Figure 14A, 14B, and 14C: Motion tracking measurement stages

14A: Tracking all markers during test. 14B: Tracking HHA on humerus. 14C: Tracking glenoid

perimeter on scapula
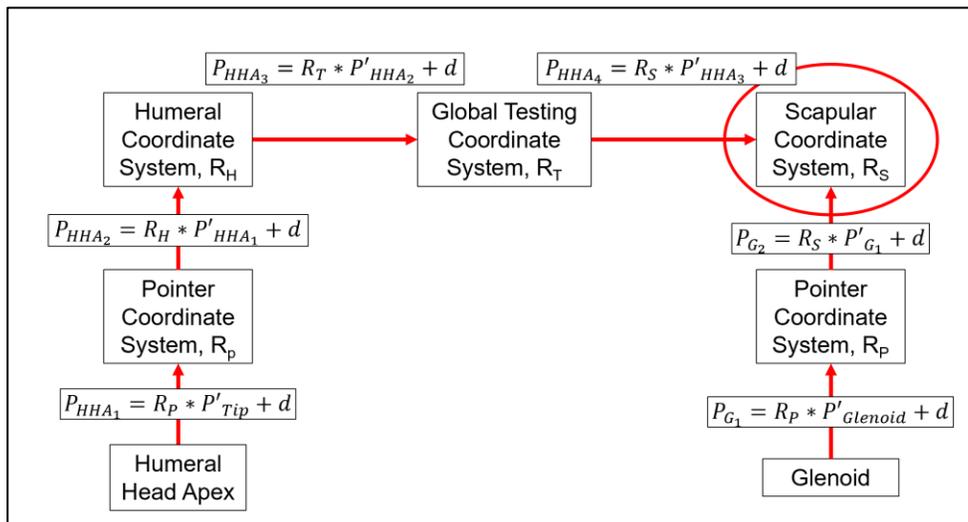
Figure 15: Diagram showing coordinate transformations and equations

3.5.2 Integrated Tendon Strain and Gap Widths

The same motion tracking system was used to track the movement of the three columns of five black markers placed on the SS cord, SS strap, and the adjacent border between them (Figure 16A and 16B). The markers were placed on these structures to determine the force

transmission in the SS tendon for both the loading and cutting cases. The length between each

adjacent marker in each column was measured. The strain was integrated along the total length

of the markers by summing the total length between all five markers. The strain was calculated

using the native condition as the reference length and every loading or cutting case as the new

length (Figure 17) (Appendix A.4). Gap widths were only measured for the cutting cases, and the

width was defined as the distance between the marker distal to the cut and the marker directly

proximal (Figure 18).
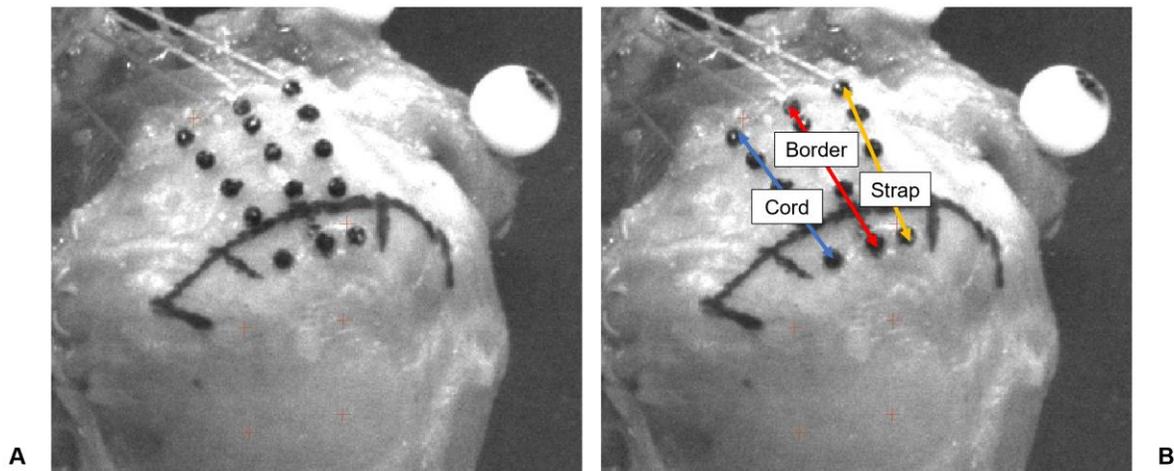
Figure 16A and 16B: Picture of strain markers

16A shows markers on SS tendon. 16B has markers labeled and total length shown
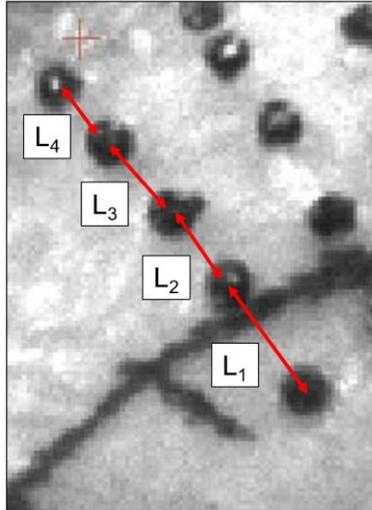
$$L = L_1 + L_2 + L_3 + L_4$$

Length calculated for native (L_N) and every loading and cutting case (L_C)

$$Strain = \varepsilon = \frac{L_C - L_N}{L_N}$$

Figure 17: Strain integrated over multiple markers
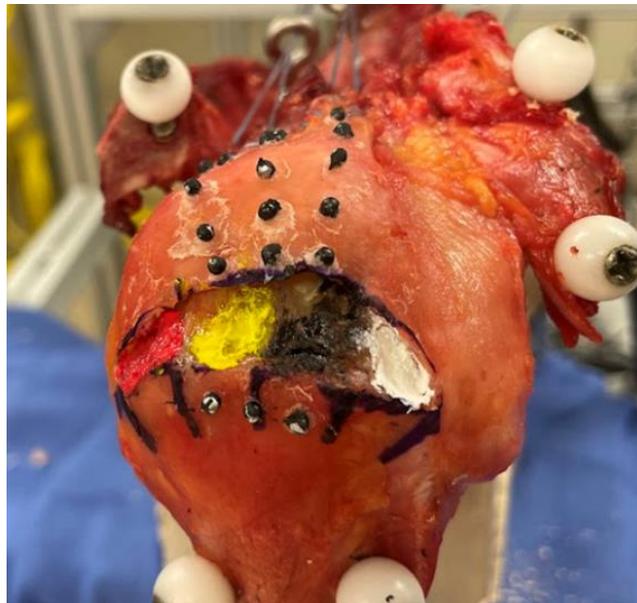
Figure 18: Specimen gap width after rotator cuff tendons released

3.6 Statistical Analysis

Multifactor repeated measures ANOVAs followed by post hoc analysis using Bonferroni correction if statistical significance occurred (SPSS, IBM) ($p<0.05$) were used to analyzed HHA translation and rotation, integrated tendon strain, and gap widths. Statistical analysis was performed for both the loading cases and the two cut groups, meaning there were three ANOVAs for each measurement. The three factors for HHA translation were abduction angle, direction relative to glenoid centroid, and loading/cutting case. The two factors for the HHA rotation were abduction angle and loading/cutting case. There were only two factors for tendon strain, and they were strain location and loading/cutting case. For gap widths, the three factors were abduction angle, tendon location, and loading/cutting case.

4.0 Results

Motion tracking analysis was not completed for one of the twenty specimens; there was a motion tracking system malfunction and data file corruption. This left nineteen total specimens analyzed for the loading cases. The nineteen specimens were also split into nine specimens for the CHL first-cut group and ten specimens for the SS cord first-cut group.

4.1 Humeral Head Apex Translation and Rotation

The three-dimensional coordinates outputted by the motion tracking system were transformed by the custom MATLAB script into a two-dimensional coordinate system. The centroid of the glenoid was made as the origin, with the x-axis as the transverse plane and the y-axis as the coronal plane (Figure 19A and 19B). The HHA translation relative to native case for both loading cases and cutting cases are shown in bar charts in Figures 20-25. Translation was not plotted on the average glenoid due to magnitude less than 2mm in both the anteroposterior and superoinferior directions. Abduction angle, loading/cutting case, and direction were not statistically significant ($p > 0.231$). This shows that simulated CHL, SS cord, and SS strap tears and abduction angle had no detectable effect on humeral head migration in either direction.
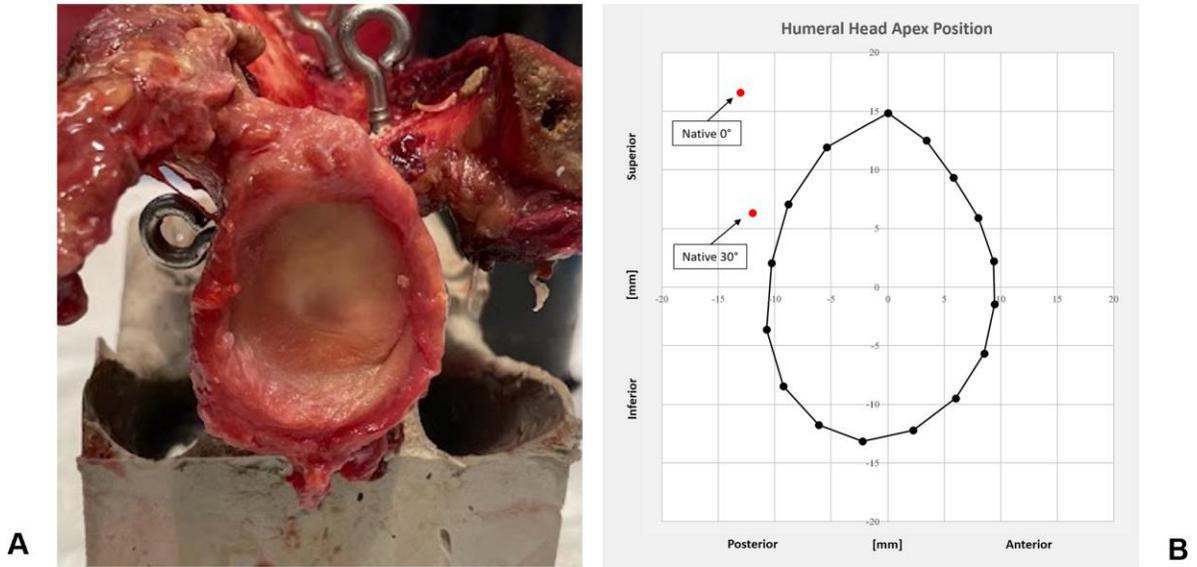
Figure 19A and 19B: Glenoid fossa

19A: Dissected glenoid. 19B: Glenoid transformed to 2D plane, with average humeral head apex points of

Native 0° and Native 30°
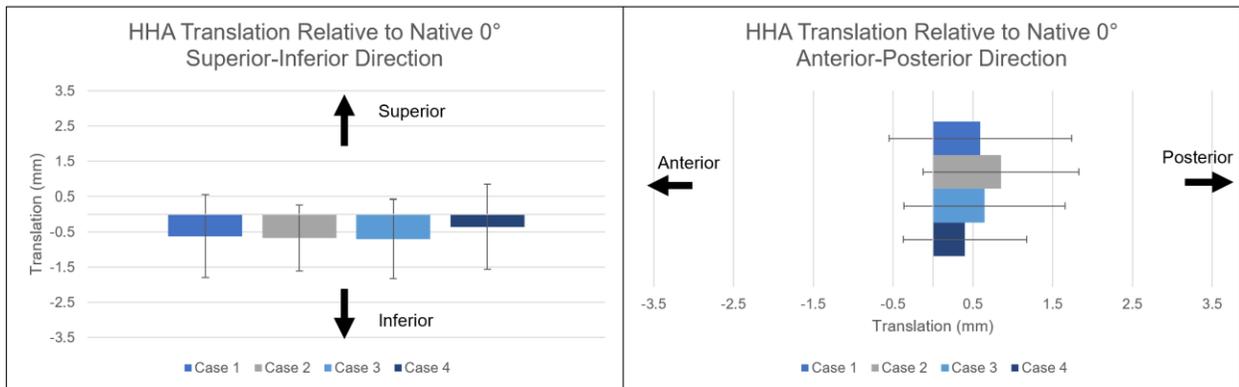
Figure 20: HHA translation for 0° load cases
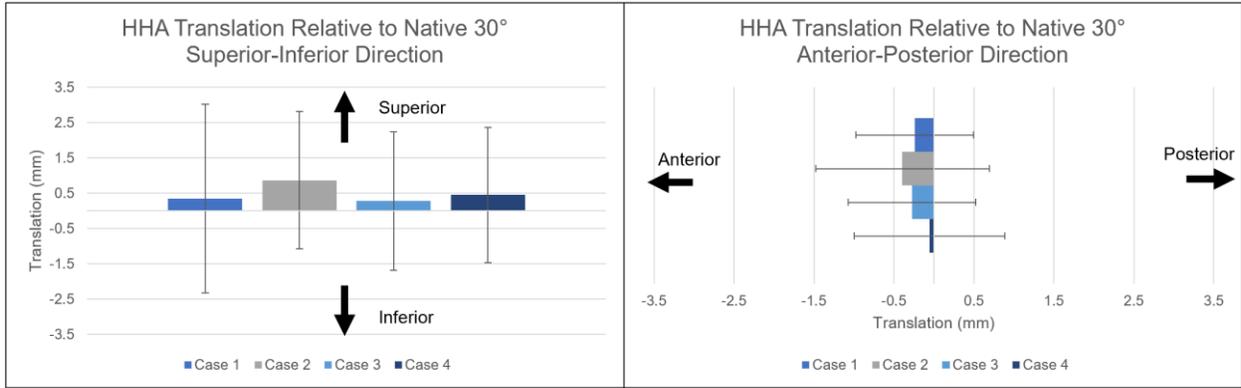
Figure 21: HHA translation for 30° load cases
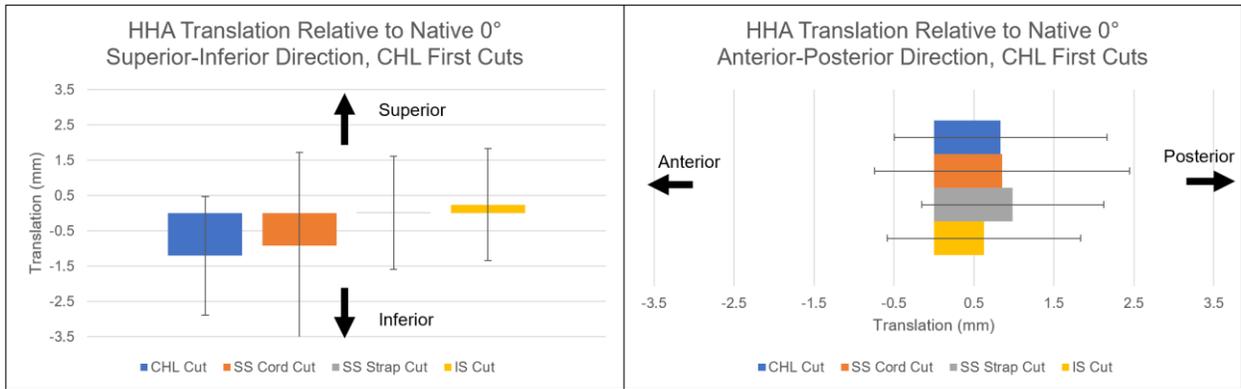
Figure 22: HHA translation for 0° CHL first cut cases
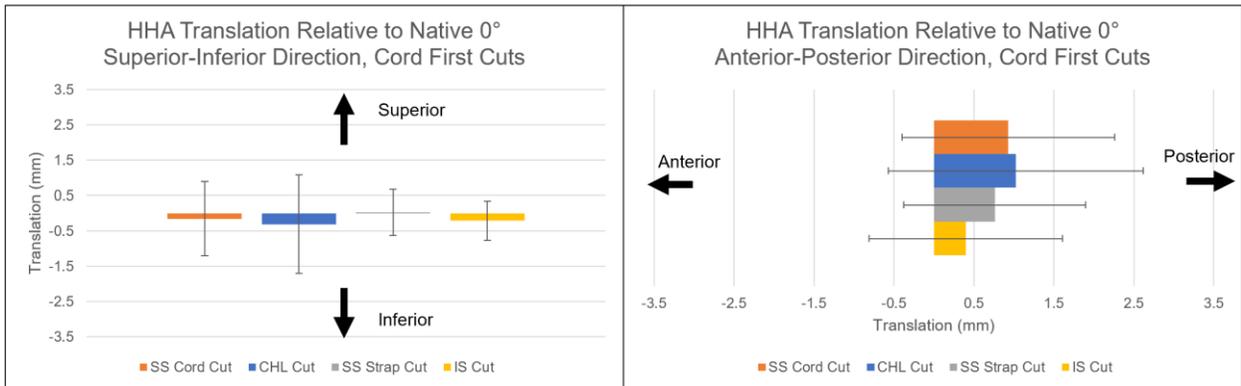
Figure 23: HHA translation for 0° SS Cord first cut cases
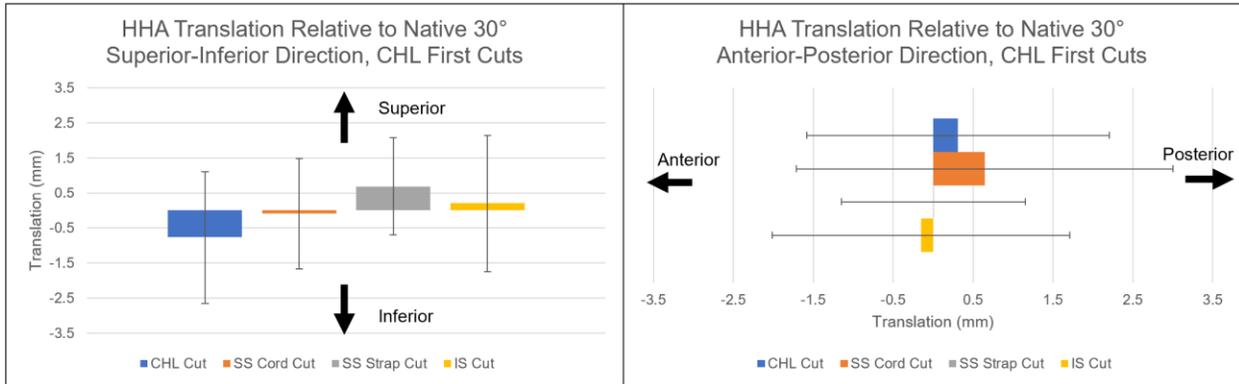
Figure 24: HHA translation for 30° CHL first cuts
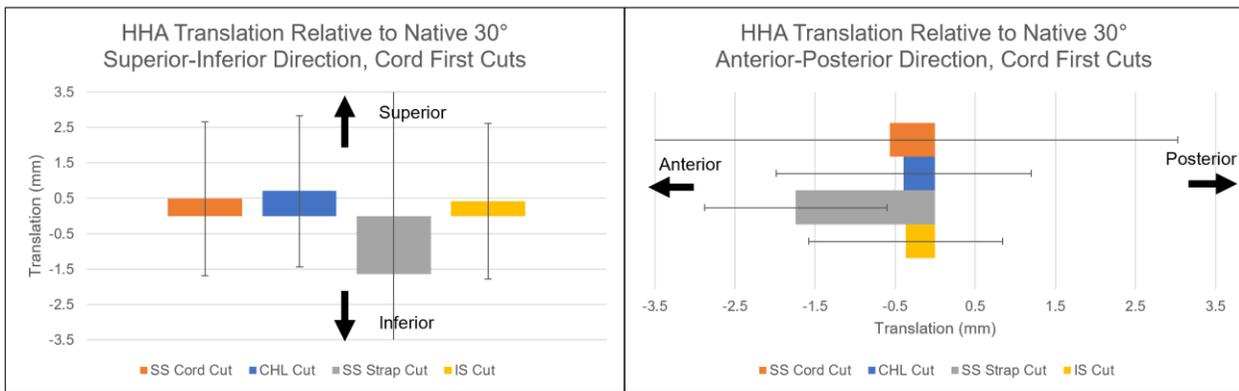
Figure 25: HHA translation for 30° SS Cord first cuts

The HHA rotation relative to native case for both loading cases and cutting cases are shown in bar charts in Figures 26-28. In all loading and cutting cases, the humeral head internally rotated with average magnitudes reaching almost 20°. Abduction angle and loading/cutting cases were not statistically significant (p>0.305). This shows that specimens did not significantly internally rotate during testing compared to the native cases.
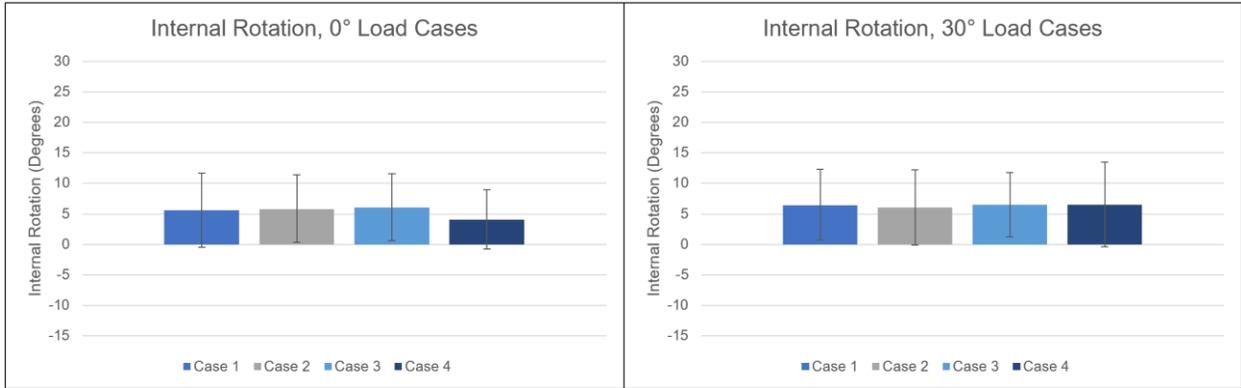
Figure 26: Internal rotation of load cases at 0° and 30°
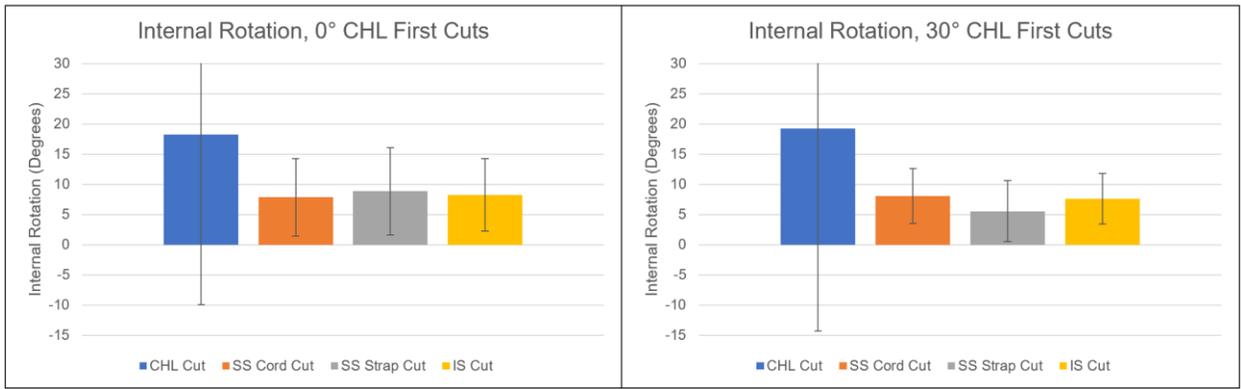
Figure 27: Internal rotation of CHL first cuts at 0° and 30°
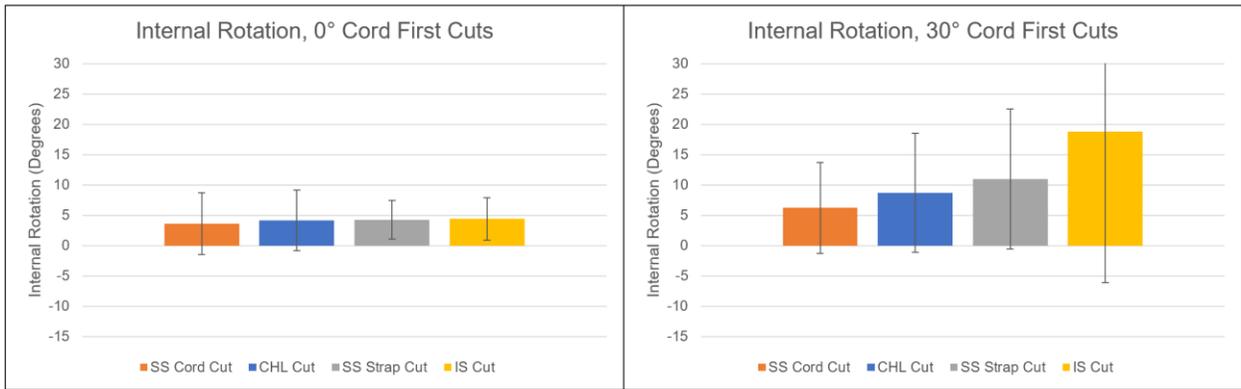
Figure 28: Internal rotation of SS cord first cuts at 0° and 30°

4.2 Integrated Tendon Strain

Tendon strain was calculated only for 0° of abduction; the tendon markers were not reliably visible in all of the 30° abduction cases. The tendon markers were also not visible for all of the cutting cases due to tendon shrinkage and large gap widths from the simulated releases. In total, nineteen specimens were analyzed for the loading cases, five specimens for the CHL first cutting cases, and six specimens for the SS cord first cutting cases. The results can be seen in Figures 29-31.

Both the loading cases and cutting cases followed the expected strain pattern resulting from load variation and tendon release, respectively. For the load cases, strain in the SS cord decreased compared to the native condition for cases 1 and 2; the SS cord was not loaded during these cases. A similar pattern followed for the SS strap for cases 3 and 4. The SS cord also decreased in strain magnitude when released in both cutting groups. Negative strain appeared in for the SS cord in the loading cases and in the latter cuts in both cutting groups, which indicates that the tendon contracted in total length. While visually there appears to be significant differences among strain in both loading and cutting cases, none of the factors were statistically significant ($p > 0.057$).
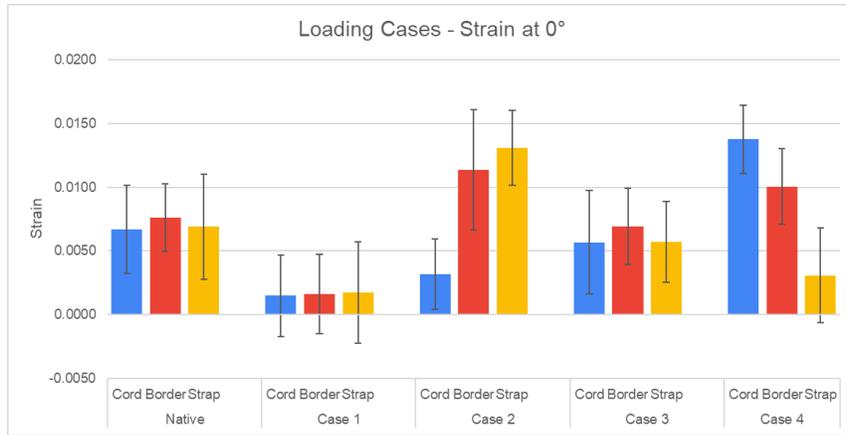
Figure 29: Loading case strain results
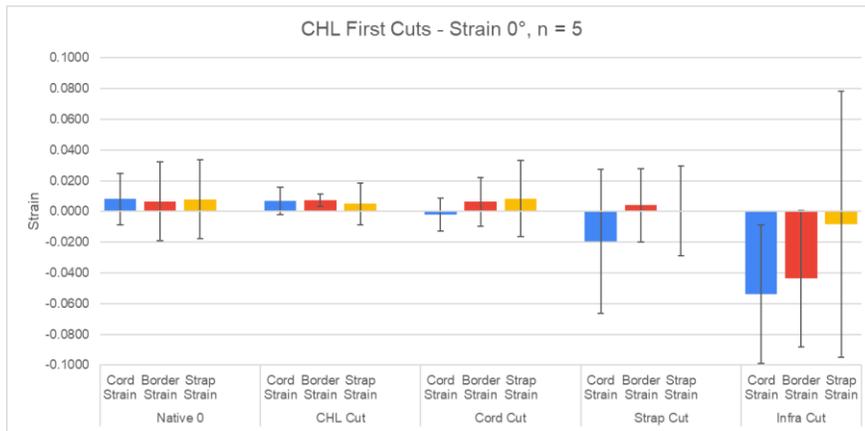
Figure 30: CHL first cut strain results
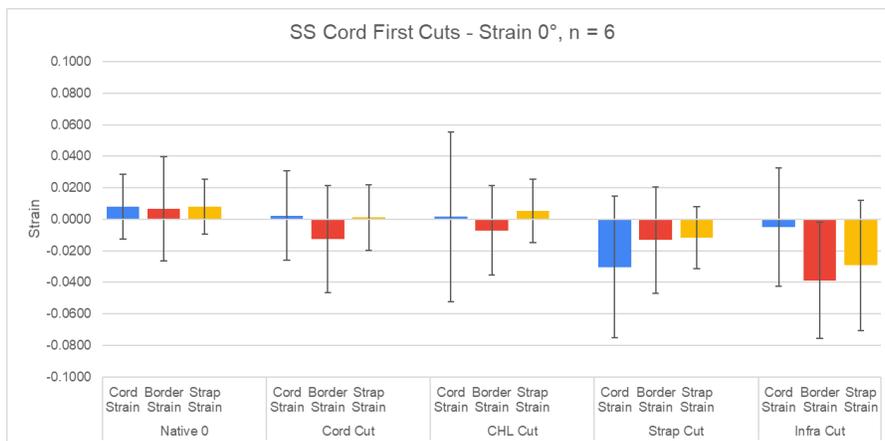
Figure 31: SS Cord first cut strain results

4.3 Gap Widths

Gap width measurements were completed for all motion tracking specimens for the cutting cases at both 0° and 30° of abduction (n=9 for CHL first cut group and n=10 for SS cord first cut group). The results can be seen in Figures 32 and 33. Abduction angle (p<0.014) and cutting case (p<0.009) were significant for both CHL first cut and SS cord first cut groups, but tendon location was not (p>0.298). The red stars on Figures 32 and 33 indicate significant difference between cut case and native.
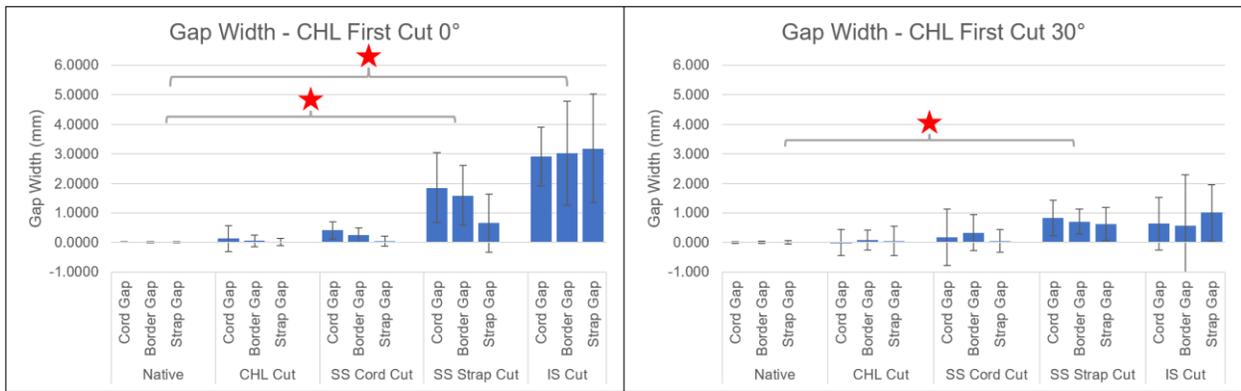
Figure 32: Gap width results for CHL first cut at 0° and 30°
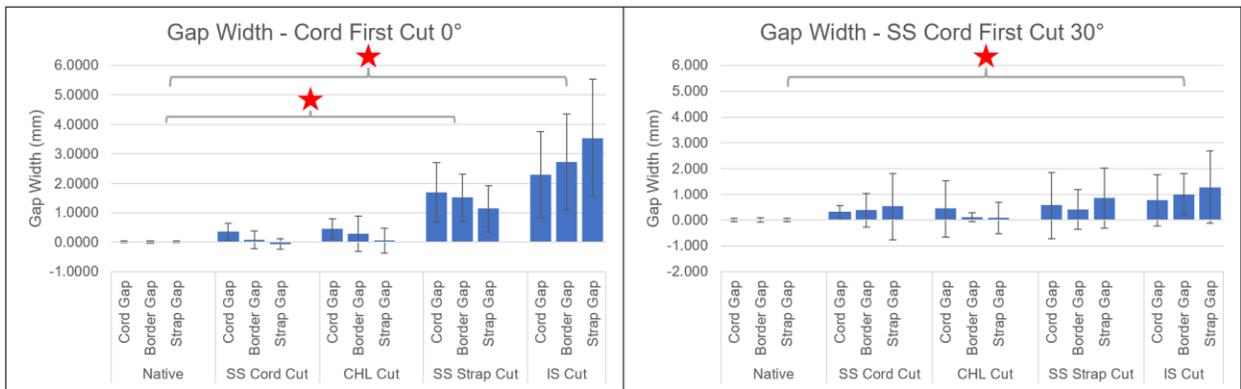
Figure 33: Gap width results for SS cord first cut at 0° and 30°

5.0 Discussion and Conclusion

Previous mechanical and clinical studies have shown that anterior SS tendon tears decreased abduction motion and increased glenohumeral translation compared to posterior SS tears[1,8,12,17,19]. The results of this study show there is increased average translation of the HHA relative to the native loading condition for loading cases 1 and 2, where the anterior SS cord was unloaded, compared to cases 3 and 4, where the SS cord was loaded. While there was an increase in translation, total translation in both the anteroposterior and superoinferior directions relative to the native case was less than 1mm, and there was no statistical difference among any of the loading cases. The rotation results also did not follow the loading patterns and there was no statistical difference.

The main comparison in the cutting cases was between the CHL and the SS cord to determine if there was any humeral head migration when simulating RC tears. There were no discernable visual HHA translation or rotation patterns when examining the CHL first cut and SS cord first cut data. All translations relative to the respective native cases were less than 2mm. Although internal rotation reached almost 20° and occurred when the CHL was cut in the CHL first cut cases, there was no statistical difference among any of the factors for both translation and rotation. Both the cutting case and loading case results show that the humeral head remained within the socket of the glenoid and did not move a significant amount when simulating tears through physical release or tendon load variation. This does not follow the results seen from other biomechanical or clinical studies[17,19].

The tendon strain results show where the location of force transmission occurred for both loading and cutting cases. For the loading cases, a modeled cord tear with full load transfer to the

SS strap (case 2) resulted in a 52% drop in SS cord strain and an 89% increase in SS strap strain compared to the strain in the native case at 0° of abduction. Similarly for the opposite load transfer (case 4), there was a 106% increase in SS cord strain and a 55% decrease in SS strap strain. Although there was a large difference in strain, none of the loading cases were determined to be significant ($p > 0.057$). These results indicate there isn't full force transmission from the SS cord to the SS strap and vice versa, highlighting the fact that these two tendinous structures of the bipennate SS tendon are crosslinked.

The strain results from the cutting case, although not statistically significant, show that force transmission throughout the SS cord or SS strap decreases once physically released. The large negative strain values following the SS strap cut show that the tendon shrinks since it is no longer attached to its humeral insertion.

The gap width is significant once the SS strap is released for the both first cut groups. This means that there a significant gap between the tendon and humerus forms once the entire SS tendon and CHL is released. The gap widths were less conclusive when the specimens were tested at 30° of abduction. These results make sense physically as the tendon has a smaller distance to contract when the arm is in a higher abduction state.

The results from the HHA translation, HHA rotation, tendon strain, and gap width measurements provide more information on the impact the CHL, SS cord, and SS strap have on shoulder function. The HHA translation and rotation results indicate that the humeral head does not significantly move within the glenoid socket. The tendon strain results show that while strain decreases in the SS cord or SS strap when they are unloaded or physically released, there is no significant difference in strain. Force is transmitted throughout the entire SS tendon. The gap width results show that a significant gap forms once the entire SS is released.

There are potential clinical impacts from interpreting the results of this study. The results show anterior RC tears does not cause humeral head migration, abduction force is transmitted to surrounding tendons, and the gap of the tear is not significantly large. This supports the claim that anterior tears of the SS including the CHL can be treated conservatively without the need for surgical intervention.

5.1 Limitations and Future Work

There are a few limitations to this study that must be considered when evaluating the results. All of the motion tracking results had large standard deviations which could have had an effect on the statistical analysis and significance. Factors such as marker placement, possible small movements of the specimen, and specimen variation could have been responsible for the larger standard deviation. The deltoid muscle was also not loaded and is used primarily in abduction. Not having a deltoid force is a limitation on the results. Nevertheless, simplifying the protocol to include only the rotator cuff muscles limits confounding variables. The findings may also not apply to other arm positions or loading conditions.

This motion tracking system can be improved for use in future shoulder studies. The white markers on the humerus and scapula were assumed as fixed, but their design could be improved in future iterations. The placement of the markers, specifically on the scapula, could also be changed to create more accurate local coordinate systems that reflect anatomic planes. Overall, this initial motion tracking set up showed reasonable results and is step in the right direction.

Appendix A MATLAB Code

Appendix A.1 Loading and Cutting Case Randomization

```
%Generate the order of loading cases for shoulder abduction
%1 = Native, 2 = Load 1, 3 = Load 2, 4 = Load 3, 5 = Load 4
sequence = randperm(5,5);
disp(sequence)

%Generate random number determining whether the CHL or Cord is cut first
%1 = CHL, 2 = SS Cord
cutF = randperm(2,2);
disp(cutF)
```

Appendix A.2 Soder Function

```
% soder.m: Matlab function to determine rigid body rotation & translation
% From:
% I. Soederqvist and P.A. Wedin (1993) Determining the movement of the
skeleton
%   using well-configured markers. J. Biomech. 26:1473-1477.
% Same algorithm is described in:
% J.H. Challis (1995) A prodecure for determining rigid body transformation
%   parameters, J. Biomech. 28, 733-737.
% The latter also includes possibilities for scaling, reflection, and
% weighting of marker data.
%
% Written by Ron Jacobs (R.S. Dow Neurological Institute, Porland OR),
% adapted by Ton van den Bogert (University of Calgary).
%
% Input:
% x: 3-D marker coordinates in position 1 (3 columns, one row for each
marker)
% y: 3-D marker coordinates in position 2 (same format)
%
% Output:
% R: rotation matrix
% d: translation vector
% rms: the root mean square fit error of the rigid body model
```

```
%
% the rigid body model is: y = R*x + d
%
function[R,d,rms]=soder2(x,y)

[nmarkers,ndimensions]=size(x);
% we could give an error message if ndimensions is not 3

mx=mean(x);
my=mean(y);

% construct matrices A and B, subtract the mean so there is only rotation
for i=1:nmarkers,
  A(i,:)=x(i,:)-mx;
  B(i,:)=y(i,:)-my;
end
A = A';
B = B';

% The singular value decomposition to calculate R with det(R)=1
C=B*A';
[P,T,Q]=svd(C);
R=P*diag([1 1 det(P*Q')])*Q';

% Calculate the translation vector from the centroid of all markers
d=my'-R*mx';

% calculate RMS value of residuals
sumsq = 0;
for i=1:nmarkers
  ypred = R*x(i,:)' + d;
  sumsq = sumsq + norm(ypred-y(i,:)')^2;
end
rms = sqrt(sumsq/3/nmarkers);
```

Appendix A.3 Motion Tracking Code

```
% Motion Tracking of Apex of Humeral Head Relative to Glenoid
% 2/1/2023
close all; clear; clc

%% Abbreviations
% Sty = Stylus, Pointed tip of Pointer
% PA = Pointer Array, 5 white balls on array of pointer
% HM = Humerus Markers, 4 white balls on humerus
% ScM = Scapula Markers, 4 white balls on scapula
% GE = Glenoid Edge, Stylus pointing to Glenoid Edge
% CMM = Coordinate Measuring Machine
```

```matlab
%% Loading in Data
% Input coordinates of new pointer #1 from measurements from CMM
CMM = [266.822 69.544 -383.424; % Top Ball
    256.181 113.491 -385.244; % Short arm
    278.281 97.159 -424.465; % Following are markers clockwise
    239.199 64.583 -423.411;
    219.516 85.052 -382.484];
Sty1 = [173.374 188.178 -472.785]; % Stylus point relative to pointer marks
from CMM

% Spicatek coordinates; stylus touching humeral apex
HumApexTemp = xlsread('HumApex.xlsx',1,'B5:P5'); % PA Coordinates
HumApex = reshape(HumApexTemp,3,5)';
HM1temp = xlsread('HumApex.xlsx',1,'AC5:AN5'); % HM Coordinates
HM1 = reshape(HM1temp,3,4)';

% Spicatek coordinates; stylus touching glenoid edges
GEp1temp = xlsread('Glenoid.xlsx',1,'B5:P5'); % PA Coordinates Point 1
GEp2temp = xlsread('Glenoid.xlsx',1,'B6:P6'); % PA Coordinates Point 2
GEp3temp = xlsread('Glenoid.xlsx',1,'B7:P7'); % PA Coordinates Point 3
GEp4temp = xlsread('Glenoid.xlsx',1,'B8:P8'); % PA Coordinates Point 4
GEp5temp = xlsread('Glenoid.xlsx',1,'B9:P9'); % PA Coordinates Point 5
GEp6temp = xlsread('Glenoid.xlsx',1,'B10:P10'); % PA Coordinates Point 6
GEp7temp = xlsread('Glenoid.xlsx',1,'B11:P11'); % PA Coordinates Point 7
GEp8temp = xlsread('Glenoid.xlsx',1,'B12:P12'); % PA Coordinates Point 8
GEp9temp = xlsread('Glenoid.xlsx',1,'B13:P13'); % PA Coordinates Point 9
GEp10temp = xlsread('Glenoid.xlsx',1,'B14:P14'); % PA Coordinates Point 10
GEp11temp = xlsread('Glenoid.xlsx',1,'B15:P15'); % PA Coordinates Point 11
GEp12temp = xlsread('Glenoid.xlsx',1,'B16:P16'); % PA Coordinates Point 12
GEp13temp = xlsread('Glenoid.xlsx',1,'B17:P17'); % PA Coordinates Point 13
GEp14temp = xlsread('Glenoid.xlsx',1,'B18:P18'); % PA Coordinates Point 14
GEp15temp = xlsread('Glenoid.xlsx',1,'B19:P19'); % PA Coordinates Point 15
GEp16temp = xlsread('Glenoid.xlsx',1,'B20:P20'); % PA Coordinates Point 16
GEp1 = reshape(GEp1temp,3,5)';
GEp2 = reshape(GEp2temp,3,5)';
GEp3 = reshape(GEp3temp,3,5)';
GEp4 = reshape(GEp4temp,3,5)';
GEp5 = reshape(GEp5temp,3,5)';
GEp6 = reshape(GEp6temp,3,5)';
GEp7 = reshape(GEp7temp,3,5)';
GEp8 = reshape(GEp8temp,3,5)';
GEp9 = reshape(GEp9temp,3,5)';
GEp10 = reshape(GEp10temp,3,5)';
GEp11 = reshape(GEp11temp,3,5)';
GEp12 = reshape(GEp12temp,3,5)';
GEp13 = reshape(GEp13temp,3,5)';
GEp14 = reshape(GEp14temp,3,5)';
GEp15 = reshape(GEp15temp,3,5)';
GEp16 = reshape(GEp16temp,3,5)';
```

```matlab
ScM1p1temp = xlsread('Glenoid.xlsx',1,'Q5:AB5'); % ScM Coordinates Point 1
ScM1p2temp = xlsread('Glenoid.xlsx',1,'Q6:AB6'); % ScM Coordinates Point 2
ScM1p3temp = xlsread('Glenoid.xlsx',1,'Q7:AB7'); % ScM Coordinates Point 3
ScM1p4temp = xlsread('Glenoid.xlsx',1,'Q8:AB8'); % ScM Coordinates Point 4
ScM1p5temp = xlsread('Glenoid.xlsx',1,'Q9:AB9'); % ScM Coordinates Point 5
ScM1p6temp = xlsread('Glenoid.xlsx',1,'Q10:AB10'); % ScM Coordinates Point 6
ScM1p7temp = xlsread('Glenoid.xlsx',1,'Q11:AB11'); % ScM Coordinates Point 7
ScM1p8temp = xlsread('Glenoid.xlsx',1,'Q12:AB12'); % ScM Coordinates Point 8
ScM1p9temp = xlsread('Glenoid.xlsx',1,'Q13:AB13'); % ScM Coordinates Point 9
ScM1p10temp = xlsread('Glenoid.xlsx',1,'Q14:AB14'); % ScM Coordinates Point
10
ScM1p11temp = xlsread('Glenoid.xlsx',1,'Q15:AB15'); % ScM Coordinates Point
11
ScM1p12temp = xlsread('Glenoid.xlsx',1,'Q16:AB16'); % ScM Coordinates Point
12
ScM1p13temp = xlsread('Glenoid.xlsx',1,'Q17:AB17'); % ScM Coordinates Point
13
ScM1p14temp = xlsread('Glenoid.xlsx',1,'Q18:AB18'); % ScM Coordinates Point
14
ScM1p15temp = xlsread('Glenoid.xlsx',1,'Q19:AB19'); % ScM Coordinates Point
15
ScM1p16temp = xlsread('Glenoid.xlsx',1,'Q20:AB20'); % ScM Coordinates Point
16
ScM1p1 = reshape(ScM1p1temp,3,4)';
ScM1p2 = reshape(ScM1p2temp,3,4)';
ScM1p3 = reshape(ScM1p3temp,3,4)';
ScM1p4 = reshape(ScM1p4temp,3,4)';
ScM1p5 = reshape(ScM1p5temp,3,4)';
ScM1p6 = reshape(ScM1p6temp,3,4)';
ScM1p7 = reshape(ScM1p7temp,3,4)';
ScM1p8 = reshape(ScM1p8temp,3,4)';
ScM1p9 = reshape(ScM1p9temp,3,4)';
ScM1p10 = reshape(ScM1p10temp,3,4)';
ScM1p11 = reshape(ScM1p11temp,3,4)';
ScM1p12 = reshape(ScM1p12temp,3,4)';
ScM1p13 = reshape(ScM1p13temp,3,4)';
ScM1p14 = reshape(ScM1p14temp,3,4)';
ScM1p15 = reshape(ScM1p15temp,3,4)';
ScM1p16 = reshape(ScM1p16temp,3,4)';

% % Spicatek coordinates; test file of Native, 0 deg at start
% HM2temp = xlsread('Native-0.xlsx',1,'AC5:AN5'); % HM Coordinates
% HM2 = reshape(HM2temp,3,4)';
% ScM2temp = xlsread('Native-0.xlsx',1,'Q5:AB5'); % ScM Coordinates
% ScM2 = reshape(ScM2temp,3,4)';

% Spicatek coordinates; test file of Native, 0 deg at end
HM3temp = xlsread('Native-0.xlsx',1,'AC6:AN6'); % HM Coordinates
```

```matlab
HM3 = reshape(HM3temp,3,4)';
ScM3temp = xlsread('Native-0.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM3 = reshape(ScM3temp,3,4)';

% % Spicatek coordinates; test file of Native, 30 deg at start
% HM4temp = xlsread('Native-30.xlsx',1,'AC5:AN5'); % HM Coordinates
% HM4 = reshape(HM4temp,3,4)';
% ScM4temp = xlsread('Native-30.xlsx',1,'Q5:AB5'); % ScM Coordinates
% ScM4 = reshape(ScM4temp,3,4)';

% Spicatek coordinates; test file of Native, 30 deg at end
HM5temp = xlsread('Native-30.xlsx',1,'AC6:AN6'); % HM Coordinates
HM5 = reshape(HM5temp,3,4)';
ScM5temp = xlsread('Native-30.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM5 = reshape(ScM5temp,3,4)';

% Spicatek coordinates; test file of Cut 1, 0 deg at end
HM6temp = xlsread('Cut1-0.xlsx',1,'AC6:AN6'); % HM Coordinates
HM6 = reshape(HM6temp,3,4)';
ScM6temp = xlsread('Cut1-0.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM6 = reshape(ScM6temp,3,4)';

% Spicatek coordinates; test file of Cut 1, 30 deg at end
HM7temp = xlsread('Cut1-30.xlsx',1,'AC6:AN6'); % HM Coordinates
HM7 = reshape(HM7temp,3,4)';
ScM7temp = xlsread('Cut1-30.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM7 = reshape(ScM7temp,3,4)';

% Spicatek coordinates; test file of Cut 2, 0 deg at end
HM8temp = xlsread('Cut2-0.xlsx',1,'AC6:AN6'); % HM Coordinates
HM8 = reshape(HM8temp,3,4)';
ScM8temp = xlsread('Cut2-0.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM8 = reshape(ScM8temp,3,4)';

% Spicatek coordinates; test file of Cut 2, 30 deg at end
HM9temp = xlsread('Cut2-30.xlsx',1,'AC6:AN6'); % HM Coordinates
HM9 = reshape(HM9temp,3,4)';
ScM9temp = xlsread('Cut2-30.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM9 = reshape(ScM9temp,3,4)';

% Spicatek coordinates; test file of Cut 3, 0 deg at end
HM10temp = xlsread('Cut3-0.xlsx',1,'AC6:AN6'); % HM Coordinates
HM10 = reshape(HM10temp,3,4)';
ScM10temp = xlsread('Cut3-0.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM10 = reshape(ScM10temp,3,4)';

% Spicatek coordinates; test file of Cut 3, 30 deg at end
HM11temp = xlsread('Cut3-30.xlsx',1,'AC6:AN6'); % HM Coordinates
HM11 = reshape(HM11temp,3,4)';
```

```matlab
ScM11temp = xlsread('Cut3-30.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM11 = reshape(ScM11temp,3,4)';

% Spicatek coordinates; test file of Cut 4, 0 deg at end
HM12temp = xlsread('Cut4-0.xlsx',1,'AC6:AN6'); % HM Coordinates
HM12 = reshape(HM12temp,3,4)';
ScM12temp = xlsread('Cut4-0.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM12 = reshape(ScM12temp,3,4)';

% Spicatek coordinates; test file of Cut 4, 30 deg at end
HM13temp = xlsread('Cut4-30.xlsx',1,'AC6:AN6'); % HM Coordinates
HM13 = reshape(HM13temp,3,4)';
ScM13temp = xlsread('Cut4-30.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM13 = reshape(ScM13temp,3,4)';

% Spicatek coordinates; test file of Unload1, 0 deg at end
HM14temp = xlsread('Unload1-0.xlsx',1,'AC6:AN6'); % HM Coordinates
HM14 = reshape(HM14temp,3,4)';
ScM14temp = xlsread('Unload1-0.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM14 = reshape(ScM14temp,3,4)';

% Spicatek coordinates; test file of Unload1, 30 deg at end
HM15temp = xlsread('Unload1-30.xlsx',1,'AC6:AN6'); % HM Coordinates
HM15 = reshape(HM15temp,3,4)';
ScM15temp = xlsread('Unload1-30.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM15 = reshape(ScM15temp,3,4)';

% Spicatek coordinates; test file of Unload2, 0 deg at end
HM16temp = xlsread('Unload2-0.xlsx',1,'AC6:AN6'); % HM Coordinates
HM16 = reshape(HM16temp,3,4)';
ScM16temp = xlsread('Unload2-0.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM16 = reshape(ScM16temp,3,4)';

% Spicatek coordinates; test file of Unload2, 30 deg at end
HM17temp = xlsread('Unload2-30.xlsx',1,'AC6:AN6'); % HM Coordinates
HM17 = reshape(HM17temp,3,4)';
ScM17temp = xlsread('Unload2-30.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM17 = reshape(ScM17temp,3,4)';

% Spicatek coordinates; test file of Unload3, 0 deg at end
HM18temp = xlsread('Unload3-0.xlsx',1,'AC6:AN6'); % HM Coordinates
HM18 = reshape(HM18temp,3,4)';
ScM18temp = xlsread('Unload3-0.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM18 = reshape(ScM18temp,3,4)';

% Spicatek coordinates; test file of Unload3, 30 deg at end
HM19temp = xlsread('Unload3-30.xlsx',1,'AC6:AN6'); % HM Coordinates
HM19 = reshape(HM19temp,3,4)';
ScM19temp = xlsread('Unload3-30.xlsx',1,'Q6:AB6'); % ScM Coordinates
```

```matlab
ScM19 = reshape(ScM19temp,3,4)';

% Spicatek coordinates; test file of Unload4, 0 deg at end
HM20temp = xlsread('Unload4-0.xlsx',1,'AC6:AN6'); % HM Coordinates
HM20 = reshape(HM20temp,3,4)';
ScM20temp = xlsread('Unload4-0.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM20 = reshape(ScM20temp,3,4)';

% Spicatek coordinates; test file of Unload4, 30 deg at end
HM21temp = xlsread('Unload4-30.xlsx',1,'AC6:AN6'); % HM Coordinates
HM21 = reshape(HM21temp,3,4)';
ScM21temp = xlsread('Unload4-30.xlsx',1,'Q6:AB6'); % ScM Coordinates
ScM21 = reshape(ScM21temp,3,4)';

%% Finding Humeral Head Apex relative to scapula
% Step 1
% To find the position of Sty relative to PA, use Soder
[Rhapa,dhapa] = soder2(CMM,HumApex); % Rhapa = Rotation matrix of humeral
apex to pointer array, dhapa = displacement..
% Calculating new Sty position
Sty2 = Rhapa*Sty1' + dhapa;

% Step 2
% To find position of Sty relative to HM in the native test file, use Soder
% [RHM2,dHM2] = soder2(HM1,HM2);
[RHM3,dHM3] = soder2(HM1,HM3);
% [RHM4,dHM4] = soder2(HM1,HM4);
[RHM5,dHM5] = soder2(HM1,HM5);
[RHM6,dHM6] = soder2(HM1,HM6);
[RHM7,dHM7] = soder2(HM1,HM7);
[RHM8,dHM8] = soder2(HM1,HM8);
[RHM9,dHM9] = soder2(HM1,HM9);
[RHM10,dHM10] = soder2(HM1,HM10);
[RHM11,dHM11] = soder2(HM1,HM11);
[RHM12,dHM12] = soder2(HM1,HM12);
[RHM13,dHM13] = soder2(HM1,HM13);
[RHM14,dHM14] = soder2(HM1,HM14);
[RHM15,dHM15] = soder2(HM1,HM15);
[RHM16,dHM16] = soder2(HM1,HM16);
[RHM17,dHM17] = soder2(HM1,HM17);
[RHM18,dHM18] = soder2(HM1,HM18);
[RHM19,dHM19] = soder2(HM1,HM19);
[RHM20,dHM20] = soder2(HM1,HM20);
[RHM21,dHM21] = soder2(HM1,HM21);

% Calculating new Sty position
% Sty3 = RHM2*Sty2 + dHM2;
Sty3b = RHM3*Sty2 + dHM3;
% Sty3c = RHM4*Sty2 + dHM4;
```

41

```matlab
Sty3d = RHM5*Sty2 + dHM5;
Sty3e = RHM6*Sty2 + dHM6;
Sty3f = RHM7*Sty2 + dHM7;
Sty3g = RHM8*Sty2 + dHM8;
Sty3h = RHM9*Sty2 + dHM9;
Sty3i = RHM10*Sty2 + dHM10;
Sty3j = RHM11*Sty2 + dHM11;
Sty3k = RHM12*Sty2 + dHM12;
Sty3l = RHM13*Sty2 + dHM13;
Sty3m = RHM14*Sty2 + dHM14;
Sty3n = RHM15*Sty2 + dHM15;
Sty3o = RHM16*Sty2 + dHM16;
Sty3p = RHM17*Sty2 + dHM17;
Sty3q = RHM18*Sty2 + dHM18;
Sty3r = RHM19*Sty2 + dHM19;
Sty3s = RHM20*Sty2 + dHM20;
Sty3t = RHM21*Sty2 + dHM21;


%Used in the Testing System plot

% Step 3
% To find position of Sty relative to ScM in pointing test, use Soder
% Assuming Scapula is stationary throughout glenoid pointing test, just using
ScMp1
% [R1,d1] = soder2(ScM2,ScM1p1);
[R2,d2] = soder2(ScM3,ScM1p1);
% [R3,d3] = soder2(ScM4,ScM1p1);
[R4,d4] = soder2(ScM5,ScM1p1);
[R5,d5] = soder2(ScM6,ScM1p1);
[R6,d6] = soder2(ScM7,ScM1p1);
[R7,d7] = soder2(ScM8,ScM1p1);
[R8,d8] = soder2(ScM9,ScM1p1);
[R9,d9] = soder2(ScM10,ScM1p1);
[R10,d10] = soder2(ScM11,ScM1p1);
[R11,d11] = soder2(ScM12,ScM1p1);
[R12,d12] = soder2(ScM13,ScM1p1);
[R13,d13] = soder2(ScM14,ScM1p1);
[R14,d14] = soder2(ScM15,ScM1p1);
[R15,d15] = soder2(ScM16,ScM1p1);
[R16,d16] = soder2(ScM17,ScM1p1);
[R17,d17] = soder2(ScM18,ScM1p1);
[R18,d18] = soder2(ScM19,ScM1p1);
[R19,d19] = soder2(ScM20,ScM1p1);
[R20,d20] = soder2(ScM21,ScM1p1);


% Calculating new Sty position
% Sty4a = R1*Sty3 + d1; % Native 0 begin
Sty4b = R2*Sty3b + d2; % Native 0 end
% Sty4c = R3*Sty3c + d3; % Native 30 begin
```

```matlab
Sty4d = R4*Sty3d + d4;  % Native 30 end
Sty4e = R5*Sty3e + d5;  % Cut1 0 end
Sty4f = R6*Sty3f + d6;  % Cut1 30 end
Sty4g = R7*Sty3g + d7;  % Cut2 0 end
Sty4h = R8*Sty3h + d8;  % Cut2 30 end
Sty4i = R9*Sty3i + d9;  % Cut3 0 end
Sty4j = R10*Sty3j + d10;  % Cut3 30 end
Sty4k = R11*Sty3k + d11;  % Cut4 0 end
Sty4l = R12*Sty3l + d12;  % Cut4 30 end
Sty4m = R13*Sty3m + d13;  % Unload1 0 end
Sty4n = R14*Sty3n + d14;  % Unload1 30 end
Sty4o = R15*Sty3o + d15;  % Unload2 0 end
Sty4p = R16*Sty3p + d16;  % Unload2 30 end
Sty4q = R17*Sty3q + d17;  % Unload3 0 end
Sty4r = R18*Sty3r + d18;  % Unload3 30 end
Sty4s = R19*Sty3s + d19;  % Unload4 0 end
Sty4t = R20*Sty3t + d20;  % Unload4 30 end

% Now have Sty position from CMM -> HumApex Pointing Test -> Actual Test ->
Glenoid Pointing Test

% Converting HM Markers to same glenoid pointing test as above

%% Finding glenoid edge points relative to scapula
% Step 1
% To find the position of Sty relative to PA, use Soder
% Reshaped raw data points, no longer temp variable
% Rgepa = Rotation matrix of glenoid edge to pointer array, points 1,2,3 etc.
dgepa = displacement,..., points 1,2,3,etc
[Rgepa1,dgepa1] = soder2(CMM,GEp1);
[Rgepa2,dgepa2] = soder2(CMM,GEp2);
[Rgepa3,dgepa3] = soder2(CMM,GEp3);
[Rgepa4,dgepa4] = soder2(CMM,GEp4);
[Rgepa5,dgepa5] = soder2(CMM,GEp5);
[Rgepa6,dgepa6] = soder2(CMM,GEp6);
[Rgepa7,dgepa7] = soder2(CMM,GEp7);
[Rgepa8,dgepa8] = soder2(CMM,GEp8);
[Rgepa9,dgepa9] = soder2(CMM,GEp9);
[Rgepa10,dgepa10] = soder2(CMM,GEp10);
[Rgepa11,dgepa11] = soder2(CMM,GEp11);
[Rgepa12,dgepa12] = soder2(CMM,GEp12);
[Rgepa13,dgepa13] = soder2(CMM,GEp13);
[Rgepa14,dgepa14] = soder2(CMM,GEp14);
[Rgepa15,dgepa15] = soder2(CMM,GEp15);
[Rgepa16,dgepa16] = soder2(CMM,GEp16);
% Calculating new Sty positions for each glenoid edge point
GE2p1 = Rgepa1*Sty1' + dgepa1;
GE2p2 = Rgepa2*Sty1' + dgepa2;
GE2p3 = Rgepa3*Sty1' + dgepa3;
```

```
GE2p4 = Rgepa4*Sty1' + dgepa4;
GE2p5 = Rgepa5*Sty1' + dgepa5;
GE2p6 = Rgepa6*Sty1' + dgepa6;
GE2p7 = Rgepa7*Sty1' + dgepa7;
GE2p8 = Rgepa8*Sty1' + dgepa8;
GE2p9 = Rgepa9*Sty1' + dgepa9;
GE2p10 = Rgepa10*Sty1' + dgepa10;
GE2p11 = Rgepa11*Sty1' + dgepa11;
GE2p12 = Rgepa12*Sty1' + dgepa12;
GE2p13 = Rgepa13*Sty1' + dgepa13;
GE2p14 = Rgepa14*Sty1' + dgepa14;
GE2p15 = Rgepa15*Sty1' + dgepa15;
GE2p16 = Rgepa16*Sty1' + dgepa16;

% Step 2
% To find position of Sty -> ScM system of test, use Soder
% Note: All GE points are transformed into the same scapula system from the
test file
[RScM2p1,dScM2p1] = soder2(ScM1p1,ScM3); % Was using ScM2 before, change to 3
[RScM2p2,dScM2p2] = soder2(ScM1p2,ScM3);
[RScM2p3,dScM2p3] = soder2(ScM1p3,ScM3);
[RScM2p4,dScM2p4] = soder2(ScM1p4,ScM3);
[RScM2p5,dScM2p5] = soder2(ScM1p5,ScM3);
[RScM2p6,dScM2p6] = soder2(ScM1p6,ScM3);
[RScM2p7,dScM2p7] = soder2(ScM1p7,ScM3);
[RScM2p8,dScM2p8] = soder2(ScM1p8,ScM3);
[RScM2p9,dScM2p9] = soder2(ScM1p9,ScM3);
[RScM2p10,dScM2p10] = soder2(ScM1p10,ScM3);
[RScM2p11,dScM2p11] = soder2(ScM1p11,ScM3);
[RScM2p12,dScM2p12] = soder2(ScM1p12,ScM3);
[RScM2p13,dScM2p13] = soder2(ScM1p13,ScM3);
[RScM2p14,dScM2p14] = soder2(ScM1p14,ScM3);
[RScM2p15,dScM2p15] = soder2(ScM1p15,ScM3);
[RScM2p16,dScM2p16] = soder2(ScM1p16,ScM3);
% Finding new Sty position by multiplying R matrices found
GE3p1 = RScM2p1*GE2p1 + dScM2p1;
GE3p2 = RScM2p2*GE2p2 + dScM2p2;
GE3p3 = RScM2p3*GE2p3 + dScM2p3;
GE3p4 = RScM2p4*GE2p4 + dScM2p4;
GE3p5 = RScM2p5*GE2p5 + dScM2p5;
GE3p6 = RScM2p6*GE2p6 + dScM2p6;
GE3p7 = RScM2p7*GE2p7 + dScM2p7;
GE3p8 = RScM2p8*GE2p8 + dScM2p8;
GE3p9 = RScM2p9*GE2p9 + dScM2p9;
GE3p10 = RScM2p10*GE2p10 + dScM2p10;
GE3p11 = RScM2p11*GE2p11 + dScM2p11;
GE3p12 = RScM2p12*GE2p12 + dScM2p12;
GE3p13 = RScM2p13*GE2p13 + dScM2p13;
GE3p14 = RScM2p14*GE2p14 + dScM2p14;
```

```
GE3p15 = RScM2p15*GE2p15 + dScM2p15;
GE3p16 = RScM2p16*GE2p16 + dScM2p16;


% In summary, have Sty -> PA -> ScM system, need to do this for 16 glenoid
points to form "circle"
% Resulting coordinates should now all be ScM system of the test file


%% 3D Plots
% Plotting All Points
figure('Name','Glenoid Points and Hum Apex');
title('Glenoid Edge and Humeral Apex Points at 0 and 30 Deg');
%Glenoid Points
scatter3(GE2p1(1,1),GE2p1(2,1),GE2p1(3,1),100,'green')
text(GE2p1(1,1),GE2p1(2,1),GE2p1(3,1),'G1')
hold on
scatter3(GE2p2(1,1),GE2p2(2,1),GE2p2(3,1),100,'green')
text(GE2p2(1,1),GE2p2(2,1),GE2p2(3,1),'G2')
scatter3(GE2p3(1,1),GE2p3(2,1),GE2p3(3,1),100,'green')
text(GE2p3(1,1),GE2p3(2,1),GE2p3(3,1),'G3')
scatter3(GE2p4(1,1),GE2p4(2,1),GE2p4(3,1),100,'green')
text(GE2p4(1,1),GE2p4(2,1),GE2p4(3,1),'G4')
scatter3(GE2p5(1,1),GE2p5(2,1),GE2p5(3,1),100,'green')
text(GE2p5(1,1),GE2p5(2,1),GE2p5(3,1),'G5')
scatter3(GE2p6(1,1),GE2p6(2,1),GE2p6(3,1),100,'green')
text(GE2p6(1,1),GE2p6(2,1),GE2p6(3,1),'G6')
scatter3(GE2p7(1,1),GE2p7(2,1),GE2p7(3,1),100,'green')
text(GE2p7(1,1),GE2p7(2,1),GE2p7(3,1),'G7')
scatter3(GE2p8(1,1),GE2p8(2,1),GE2p8(3,1),100,'green')
text(GE2p8(1,1),GE2p8(2,1),GE2p8(3,1),'G8')
scatter3(GE2p9(1,1),GE2p9(2,1),GE2p9(3,1),100,'green')
text(GE2p9(1,1),GE2p9(2,1),GE2p9(3,1),'G9')
scatter3(GE2p10(1,1),GE2p10(2,1),GE2p10(3,1),100,'green')
text(GE2p10(1,1),GE2p10(2,1),GE2p10(3,1),'G10')
scatter3(GE2p11(1,1),GE2p11(2,1),GE2p11(3,1),100,'green')
text(GE2p11(1,1),GE2p11(2,1),GE2p11(3,1),'G11')
scatter3(GE2p12(1,1),GE2p12(2,1),GE2p12(3,1),100,'green')
text(GE2p12(1,1),GE2p12(2,1),GE2p12(3,1),'G12')
scatter3(GE2p13(1,1),GE2p13(2,1),GE2p13(3,1),100,'green')
text(GE2p13(1,1),GE2p13(2,1),GE2p13(3,1),'G13')
scatter3(GE2p14(1,1),GE2p14(2,1),GE2p14(3,1),100,'green')
text(GE2p14(1,1),GE2p14(2,1),GE2p14(3,1),'G14')
scatter3(GE2p15(1,1),GE2p15(2,1),GE2p15(3,1),100,'green')
text(GE2p15(1,1),GE2p15(2,1),GE2p15(3,1),'G15')
scatter3(GE2p16(1,1),GE2p16(2,1),GE2p16(3,1),100,'green')
text(GE2p16(1,1),GE2p16(2,1),GE2p16(3,1),'G16')


% HumApex Points Below
% Native 0 End
scatter3(Sty4b(1,1),Sty4b(2,1),Sty4b(3,1),100,'Red')
```

```
text(Sty4b(1,1),Sty4b(2,1),Sty4b(3,1),'Native0')
% Native 30 End
scatter3(Sty4d(1,1),Sty4d(2,1),Sty4d(3,1),100,'Red')
text(Sty4d(1,1),Sty4d(2,1),Sty4d(3,1),'Native30')
% Cut1 0 End
scatter3(Sty4e(1,1),Sty4e(2,1),Sty4e(3,1),100,'Red')
text(Sty4e(1,1),Sty4e(2,1),Sty4e(3,1),'Cut1-0')
% Cut1 30 End
scatter3(Sty4f(1,1),Sty4f(2,1),Sty4f(3,1),100,'Red')
text(Sty4f(1,1),Sty4f(2,1),Sty4f(3,1),'Cut1-30')
% Cut2 0 End
scatter3(Sty4g(1,1),Sty4g(2,1),Sty4g(3,1),100,'Red')
text(Sty4g(1,1),Sty4g(2,1),Sty4g(3,1),'Cut2-0')
% Cut2 30 End
scatter3(Sty4h(1,1),Sty4h(2,1),Sty4h(3,1),100,'Red')
text(Sty4h(1,1),Sty4h(2,1),Sty4h(3,1),'Cut2-30')
% Cut3 0 End
scatter3(Sty4i(1,1),Sty4i(2,1),Sty4i(3,1),100,'Red')
text(Sty4i(1,1),Sty4i(2,1),Sty4i(3,1),'Cut3-0')
% Cut3 30 End
scatter3(Sty4j(1,1),Sty4j(2,1),Sty4j(3,1),100,'Red')
text(Sty4j(1,1),Sty4j(2,1),Sty4j(3,1),'Cut3-30')
% Cut4 0 End
scatter3(Sty4k(1,1),Sty4k(2,1),Sty4k(3,1),100,'Red')
text(Sty4k(1,1),Sty4k(2,1),Sty4k(3,1),'Cut4-0')
% Cut4 30 End
scatter3(Sty4l(1,1),Sty4l(2,1),Sty4l(3,1),100,'Red')
text(Sty4l(1,1),Sty4l(2,1),Sty4l(3,1),'Cut4-30')
% Unload1 0 End
scatter3(Sty4m(1,1),Sty4m(2,1),Sty4m(3,1),100,'Red')
text(Sty4m(1,1),Sty4m(2,1),Sty4m(3,1),'%Unload1-0')
% Unload1 30 End
scatter3(Sty4n(1,1),Sty4n(2,1),Sty4n(3,1),100,'Red')
text(Sty4n(1,1),Sty4n(2,1),Sty4n(3,1),'%Unload1-30')
% Unload2 0 End
scatter3(Sty4o(1,1),Sty4o(2,1),Sty4o(3,1),100,'Red')
text(Sty4o(1,1),Sty4o(2,1),Sty4o(3,1),'Unload2-0')
% Unload2 30 End
scatter3(Sty4p(1,1),Sty4p(2,1),Sty4p(3,1),100,'Red')
text(Sty4p(1,1),Sty4p(2,1),Sty4p(3,1),'Unload2-30')
% Unload3 0 End
scatter3(Sty4q(1,1),Sty4q(2,1),Sty4q(3,1),100,'Red')
text(Sty4q(1,1),Sty4q(2,1),Sty4q(3,1),'Unload3-0')
% Unload3 30 End
scatter3(Sty4r(1,1),Sty4r(2,1),Sty4r(3,1),100,'Red')
text(Sty4r(1,1),Sty4r(2,1),Sty4r(3,1),'Unload3-30')
% Unload4 0 End
scatter3(Sty4s(1,1),Sty4s(2,1),Sty4s(3,1),100,'Red')
text(Sty4s(1,1),Sty4s(2,1),Sty4s(3,1),'Unload4-0')
```

```matlab
% Unload4 30 End
scatter3(Sty4t(1,1),Sty4t(2,1),Sty4t(3,1),100,'Red')
text(Sty4t(1,1),Sty4t(2,1),Sty4t(3,1),'Unload4-30')
view(0,19)
% Use this to make XYZ scales the same
%
h = get(gca,'DataAspectRatio');
if h(3)==1
      set(gca,'DataAspectRatio',[1 1 1/max(h(1:2))])
else
      set(gca,'DataAspectRatio',[1 1 h(3)])
end
%

%% Finding Avg, Range, STD, Variance
% Finding Centroid
GE2 =
[GE2p1';GE2p2';GE2p3';GE2p4';GE2p5';GE2p6';GE2p7';GE2p8';GE2p9';GE2p10';GE2p1
1';GE2p12';GE2p13';GE2p14';GE2p15';GE2p16'];
GE2x = 0; GE2y = 0; GE2z = 0;
for i=1:size(GE2,1)
    GE2x = (GE2x + GE2(i,1));
    GE2y = (GE2y + GE2(i,2));
    GE2z = (GE2z + GE2(i,3));
    GE2temp = [GE2x GE2y GE2z];
    GE2centroid = rdivide(GE2temp,size(GE2,1)); % 3D Centroid of glenoid

%     GE3c = zeros(size(GE3,1),3)+GE3centroid; % creating centroid matrix
same size as # of data points
%     GE3c2 = GE3 - GE3c; % Each glenoid point subtract the centroid
end

% Finding Avg distance each point is from centroid
HApoints = [Sty4b'; Sty4d'; Sty4e'; Sty4f'; Sty4g'; Sty4h'; Sty4i'; Sty4j';
Sty4k'; Sty4l'; Sty4m'; Sty4n'; Sty4o'; Sty4p'; Sty4q'; Sty4r'; Sty4s';
Sty4t'];
% ^^this one includes only the ends of all the cuts and unloads
for i=1:size(HApoints,1)
    GE2c = zeros(size(HApoints,1),3)+GE2centroid; % creating centroid matrix
same size as # of hum apex points
    GE2d = zeros(size(GE2,1),3)+GE2centroid; % creating centroid matrix same
size as # of glenoid points
    HApts = HApoints - GE2c; % Each HumApex point subtract the centroid
    GE2new = GE2 - GE2d; % Each glenoid point subtract the centroid
end
avg = mean(HApts); % Avg distance each HumApex point is away from centroid
stdev = std(HApts); % Std for above
variance = var(HApts); % Variance of all HumApex points
maximum = max(HApts); % Max distance away from centroid
```

```matlab
minimum = min(HApts); % Min distance away from centroid
range = maximum-minimum; % Range (max-min)

% Creating new coordinate system
unit1 = [GE2new(1,1)/sqrt((GE2new(1,1))^2+(GE2new(1,2))^2+(GE2new(1,3))^2)
    GE2new(1,2)/sqrt((GE2new(1,1))^2+(GE2new(1,2))^2+(GE2new(1,3))^2)
    GE2new(1,3)/sqrt((GE2new(1,1))^2+(GE2new(1,2))^2+(GE2new(1,3))^2)];
unit2temp =
[GE2new(14,1)/sqrt((GE2new(14,1))^2+(GE2new(14,2))^2+(GE2new(14,3))^2) %
Change 14 to whichever one looks most orthogonal to G1
    GE2new(14,2)/sqrt((GE2new(14,1))^2+(GE2new(14,2))^2+(GE2new(14,3))^2)
    GE2new(14,3)/sqrt((GE2new(14,1))^2+(GE2new(14,2))^2+(GE2new(14,3))^2)];
unit3temp = cross(unit1,unit2temp);
unit3 =
[unit3temp(1,1)/sqrt((unit3temp(1,1))^2+(unit3temp(2,1))^2+(unit3temp(3,1))^2
)

unit3temp(2,1)/sqrt((unit3temp(1,1))^2+(unit3temp(2,1))^2+(unit3temp(3,1))^2)

unit3temp(3,1)/sqrt((unit3temp(1,1))^2+(unit3temp(2,1))^2+(unit3temp(3,1))^2)
];
unit2temp2 = cross(unit1,unit3);
unit2 =
[unit2temp2(1,1)/sqrt((unit2temp2(1,1))^2+(unit2temp2(2,1))^2+(unit2temp2(3,1
))^2)

unit2temp2(2,1)/sqrt((unit2temp2(1,1))^2+(unit2temp2(2,1))^2+(unit2temp2(3,1)
)^2)

unit2temp2(3,1)/sqrt((unit2temp2(1,1))^2+(unit2temp2(2,1))^2+(unit2temp2(3,1)
)^2)];
newS = [unit1'; unit2'; unit3'];
newGE = [(newS*GE2new(1,:)')'; (newS*GE2new(2,:)')'; (newS*GE2new(3,:)')';
(newS*GE2new(4,:)')'; (newS*GE2new(5,:)')'; (newS*GE2new(6,:)')';
    (newS*GE2new(7,:)')'; (newS*GE2new(8,:)')'; (newS*GE2new(9,:)')';
(newS*GE2new(10,:)')'; (newS*GE2new(11,:)')'; (newS*GE2new(12,:)')';
    (newS*GE2new(13,:)')'; (newS*GE2new(14,:)')'; (newS*GE2new(15,:)')';
(newS*GE2new(16,:)')'];
newHApts = [(newS*HApts(1,:)')'; (newS*HApts(2,:)')'; (newS*HApts(3,:)')';
(newS*HApts(4,:)')'; (newS*HApts(5,:)')'; (newS*HApts(6,:)')';
(newS*HApts(7,:)')'; (newS*HApts(8,:)')'; (newS*HApts(9,:)')';
    (newS*HApts(10,:)')'; (newS*HApts(11,:)')'; (newS*HApts(12,:)')';
(newS*HApts(13,:)')'; (newS*HApts(14,:)')'; (newS*HApts(15,:)')';
(newS*HApts(16,:)')'; (newS*HApts(17,:)')'; (newS*HApts(18,:)')'];

%% Rotation Calculations
% Determining rotation from Native0 end to Native30 end
HM3n1 = [HM3(2,1) HM3(2,2) HM3(2,3)] - [HM3(1,1) HM3(1,2) HM3(1,3)];
HM3nu1 = HM3n1/norm(HM3n1);
```

```matlab
HM3n2 = [HM3(3,1) HM3(3,2) HM3(3,3)] - [HM3(1,1) HM3(1,2) HM3(1,3)];
HM3nu2 = HM3n2/norm(HM3n2);
HM3n3 = cross(HM3nu1,HM3nu2);
HM3nu3 = HM3n3/norm(HM3n3);
HM3n4 = cross(HM3nu1,HM3nu3);
HM3nu4 = HM3n4/norm(HM3n4);
% HM3nu1 = abduction, HM3nu3 = "forward flexion", HM3nu4 = IE rotation
HM3r = [HM3nu1; HM3nu3; HM3nu4];
HM3new = [(HM3r*HM3(1,:)')'; (HM3r*HM3(2,:)')'; (HM3r*HM3(3,:)')';
(HM3r*HM3(4,:)')'];

% HM5new1 = [(HM3r*HM5(1,:)')'; (HM3r*HM5(2,:)')'; (HM3r*HM5(3,:)')';
(HM3r*HM5(4,:)')'];
% % Soder to get all in same system
% [R3to5,d3to5] = soder2(HM3new,HM5new1);
% %
% rotx3to5 = acosd(R3to5(3,3)); %abd
% rotz3to5 = acosd(R3to5(1,1)); %IE rot, +I

% Creating rotation matrix for Native30 end
HM5n1 = [HM5(2,1) HM5(2,2) HM5(2,3)] - [HM5(1,1) HM5(1,2) HM5(1,3)];
HM5nu1 = HM5n1/norm(HM5n1);
HM5n2 = [HM5(3,1) HM5(3,2) HM5(3,3)] - [HM5(1,1) HM5(1,2) HM5(1,3)];
HM5nu2 = HM5n2/norm(HM5n2);
HM5n3 = cross(HM5nu1,HM5nu2);
HM5nu3 = HM5n3/norm(HM5n3);
HM5n4 = cross(HM5nu1,HM5nu3);
HM5nu4 = HM5n4/norm(HM5n4);
% HM5nu1 = abduction, HM5nu3 = "forward flexion", HM5nu4 = IE rotation
HM5r = [HM5nu1; HM5nu3; HM5nu4];
HM5new = [(HM5r*HM5(1,:)')'; (HM5r*HM5(2,:)')'; (HM5r*HM5(3,:)')';
(HM5r*HM5(4,:)')'];

% HM3r and HM5r are the rotation matrices for native 0 and 30 end
respectively
% Multiplying each 0 and 30 case by R matrices, rotation values below:
% 0 Deg cases
HM6new = [(HM3r*HM6(1,:)')'; (HM3r*HM6(2,:)')'; (HM3r*HM6(3,:)')';
(HM3r*HM6(4,:)')']; % Cut1-0
HM8new = [(HM3r*HM8(1,:)')'; (HM3r*HM8(2,:)')'; (HM3r*HM8(3,:)')';
(HM3r*HM8(4,:)')']; % Cut2-0
HM10new = [(HM3r*HM10(1,:)')'; (HM3r*HM10(2,:)')'; (HM3r*HM10(3,:)')';
(HM3r*HM10(4,:)')']; % Cut3-0
HM12new = [(HM3r*HM12(1,:)')'; (HM3r*HM12(2,:)')'; (HM3r*HM12(3,:)')';
(HM3r*HM12(4,:)')']; % Cut4-0
HM14new = [(HM3r*HM14(1,:)')'; (HM3r*HM14(2,:)')'; (HM3r*HM14(3,:)')';
(HM3r*HM14(4,:)')']; % Unload1-0
HM16new = [(HM3r*HM16(1,:)')'; (HM3r*HM16(2,:)')'; (HM3r*HM16(3,:)')';
(HM3r*HM16(4,:)')']; % Unload2-0
```

```
HM18new = [(HM3r*HM18(1,:)')'; (HM3r*HM18(2,:)')'; (HM3r*HM18(3,:)')';
(HM3r*HM18(4,:)')']; % Unload3-0
HM20new = [(HM3r*HM20(1,:)')'; (HM3r*HM20(2,:)')'; (HM3r*HM20(3,:)')';
(HM3r*HM20(4,:)')']; % Unload4-0


[R3to6,d3to6] = soder2(HM3new,HM6new);
[R3to8,d3to8] = soder2(HM3new,HM8new);
[R3to10,d3to10] = soder2(HM3new,HM10new);
[R3to12,d3to12] = soder2(HM3new,HM12new);
[R3to14,d3to14] = soder2(HM3new,HM14new);
[R3to16,d3to16] = soder2(HM3new,HM16new);
[R3to18,d3to18] = soder2(HM3new,HM18new);
[R3to20,d3to20] = soder2(HM3new,HM20new);


rotx3to6 = acosd(R3to6(3,3)); %abd
rotz3to6 = acosd(R3to6(1,1)); %IE rot, +I
rotx3to8 = acosd(R3to8(3,3)); %abd
rotz3to8 = acosd(R3to8(1,1)); %IE rot, +I
rotx3to10 = acosd(R3to10(3,3)); %abd
rotz3to10 = acosd(R3to10(1,1)); %IE rot, +I
rotx3to12 = acosd(R3to12(3,3)); %abd
rotz3to12 = acosd(R3to12(1,1)); %IE rot, +I
rotx3to14 = acosd(R3to14(3,3)); %abd
rotz3to14 = acosd(R3to14(1,1)); %IE rot, +I
rotx3to16 = acosd(R3to16(3,3)); %abd
rotz3to16 = acosd(R3to16(1,1)); %IE rot, +I
rotx3to18 = acosd(R3to18(3,3)); %abd
rotz3to18 = acosd(R3to18(1,1)); %IE rot, +I
rotx3to20 = acosd(R3to20(3,3)); %abd
rotz3to20 = acosd(R3to20(1,1)); %IE rot, +I

% Internal Rotation Matrice 0 Deg, +Internal
rotation0IE = [rotz3to6; %Cut 1-0
    rotz3to8; %Cut 2-0
    rotz3to10; %Cut 3-0
    rotz3to12; %Cut 4-0
    rotz3to14; %Unload 1-0
    rotz3to16; %Unload 2-0
    rotz3to18; %Unload 3-0
    rotz3to20]; % Unload 4-0

% 30 deg cases
HM7new = [(HM5r*HM7(1,:)')'; (HM5r*HM7(2,:)')'; (HM5r*HM7(3,:)')';
(HM5r*HM7(4,:)')']; %Cut1-30
HM9new = [(HM5r*HM9(1,:)')'; (HM5r*HM9(2,:)')'; (HM5r*HM9(3,:)')';
(HM5r*HM9(4,:)')']; %Cut2-30
HM11new = [(HM5r*HM11(1,:)')'; (HM5r*HM11(2,:)')'; (HM5r*HM11(3,:)')';
(HM5r*HM11(4,:)')']; %Cut3-30
```

```matlab
HM13new = [(HM5r*HM13(1,:)')'; (HM5r*HM13(2,:)')'; (HM5r*HM13(3,:)')';
(HM5r*HM13(4,:)')']; %Cut4-30
HM15new = [(HM5r*HM15(1,:)')'; (HM5r*HM15(2,:)')'; (HM5r*HM15(3,:)')';
(HM5r*HM15(4,:)')']; %Unload1-30
HM17new = [(HM5r*HM17(1,:)')'; (HM5r*HM17(2,:)')'; (HM5r*HM17(3,:)')';
(HM5r*HM17(4,:)')']; %Unload2-30
HM19new = [(HM5r*HM19(1,:)')'; (HM5r*HM19(2,:)')'; (HM5r*HM19(3,:)')';
(HM5r*HM19(4,:)')']; %Unload3-30
HM21new = [(HM5r*HM21(1,:)')'; (HM5r*HM21(2,:)')'; (HM5r*HM21(3,:)')';
(HM5r*HM21(4,:)')']; %Unload4-30


[R5to7,d5to7] = soder2(HM5new,HM7new);
[R5to9,d5to9] = soder2(HM5new,HM9new);
[R5to11,d5to11] = soder2(HM5new,HM11new);
[R5to13,d5to13] = soder2(HM5new,HM13new);
[R5to15,d5to15] = soder2(HM5new,HM15new);
[R5to17,d5to17] = soder2(HM5new,HM17new);
[R5to19,d5to19] = soder2(HM5new,HM19new);
[R5to21,d5to21] = soder2(HM5new,HM21new);


rotx5to7 = acosd(R5to7(3,3)); %abd
rotz5to7 = acosd(R5to7(1,1)); %IE rot, +I
rotx5to9 = acosd(R5to9(3,3)); %abd
rotz5to9 = acosd(R5to9(1,1)); %IE rot, +I
rotx5to11 = acosd(R5to11(3,3)); %abd
rotz5to11 = acosd(R5to11(1,1)); %IE rot, +I
rotx5to13 = acosd(R5to13(3,3)); %abd
rotz5to13 = acosd(R5to13(1,1)); %IE rot, +I
rotx5to15 = acosd(R5to15(3,3)); %abd
rotz5to15 = acosd(R5to15(1,1)); %IE rot, +I
rotx5to17 = acosd(R5to17(3,3)); %abd
rotz5to17 = acosd(R5to17(1,1)); %IE rot, +I
rotx5to19 = acosd(R5to19(3,3)); %abd
rotz5to19 = acosd(R5to19(1,1)); %IE rot, +I
rotx5to21 = acosd(R5to21(3,3)); %abd
rotz5to21 = acosd(R5to21(1,1)); %IE rot, +I

% Internal Rotation Matrice 30 Deg, +Internal
rotation30IE = [rotz5to7; %Cut 1-30
    rotz5to9; %Cut 2-30
    rotz5to11; %Cut 3-30
    rotz5to13; %Cut 4-30
    rotz5to15; %Unload 1-30
    rotz5to17; %Unload 2-30
    rotz5to19; %Unload 3-30
    rotz5to21]; % Unload 4-30
```

Appendix A.4 Strain Code

```
%Strain Data Code
Clear; clc

%Import the x,y,z locations of black dots
%Below, for a R shoulder
%   o - s1   b1    c1
%   o - s2   b2    c2
%   o - s3   b3    c3
%   o - s4   b4    c4
%-----------------------------
%   o - s5   b5    c5

%Enter filename for strain
Filename = 'Native-0-S.CSV';

    %Taking in original dot locations---------------------------
    c1o = xlsread(Filename,1,'B5:D5');   %m1
    c2o = xlsread(Filename,1,'N5:P5');   %m2
    c3o = xlsread(Filename,1,'Z5:AB5');   %m3
    c4o = xlsread(Filename,1,'AL5:AN5');   %m4
    c5o = xlsread(Filename,1,'AX5:AZ5');   %m5

    b1o = xlsread(Filename,1,'BJ5:BL5');   %m6
    b2o = xlsread(Filename,1,'BV5:BX5');   %m7
    b3o = xlsread(Filename,1,'CH5:CJ5');   %m8
    b4o = xlsread(Filename,1,'CT5:CV5');   %m9
    b5o = xlsread(Filename,1,'DF5:DH5');   %m10

    s1o = xlsread(Filename,1,'DR5:DT5');   %m11
    s2o = xlsread(Filename,1,'ED5:EF5');   %m12
    s3o = xlsread(Filename,1,'EP5:ER5');   %m13
    s4o = xlsread(Filename,1,'FB5:FD5');   %m14
    s5o = xlsread(Filename,1,'FN5:FP5');   %m15

    %Taking in final dot locations-------------------------------
    c1f = xlsread(Filename,1,'B6:D6');   %m1
    c2f = xlsread(Filename,1,'N6:P6');   %m2
    c3f = xlsread(Filename,1,'Z6:AB6');   %m3
    c4f = xlsread(Filename,1,'AL6:AN6');   %m4
    c5f = xlsread(Filename,1,'AX6:AZ6');   %m5

    b1f = xlsread(Filename,1,'BJ6:BL6');   %m6
    b2f = xlsread(Filename,1,'BV6:BX6');   %m7
    b3f = xlsread(Filename,1,'CH6:CJ6');   %m8
    b4f = xlsread(Filename,1,'CT6:CV6');   %m9
```

```matlab
    b5f = xlsread(Filename,1,'DF6:DH6');   %m10

    s1f = xlsread(Filename,1,'DR6:DT6');   %m11
    s2f = xlsread(Filename,1,'ED6:EF6');   %m12
    s3f = xlsread(Filename,1,'EP6:ER6');   %m13
    s4f = xlsread(Filename,1,'FB6:FD6');   %m14
    s5f = xlsread(Filename,1,'FN6:FP6');   %m15


    %UNLOADS (USES ALL MARKERS FOR INTEGRATED STRAIN)


    %Cord Original Length (Add all 5 markers)
    cordLenA = sqrt((c2o(1,1)-c1o(1,1))^2+(c2o(1,2)-c1o(1,2))^2+(c2o(1,3)-
c1o(1,3))^2);
    cordLenB = sqrt((c3o(1,1)-c2o(1,1))^2+(c3o(1,2)-c2o(1,2))^2+(c3o(1,3)-
c2o(1,3))^2);
    cordLenC = sqrt((c4o(1,1)-c3o(1,1))^2+(c4o(1,2)-c3o(1,2))^2+(c4o(1,3)-
c3o(1,3))^2);
    cordLenD = sqrt((c5o(1,1)-c4o(1,1))^2+(c5o(1,2)-c4o(1,2))^2+(c5o(1,3)-
c4o(1,3))^2);
    cordLenOriginal = cordLenA + cordLenB + cordLenC + cordLenD;


    %Border Original Length (Add all 5 markers)
    bordLenA = sqrt((b2o(1,1)-b1o(1,1))^2+(b2o(1,2)-b1o(1,2))^2+(b2o(1,3)-
b1o(1,3))^2);
    bordLenB = sqrt((b3o(1,1)-b2o(1,1))^2+(b3o(1,2)-b2o(1,2))^2+(b3o(1,3)-
b2o(1,3))^2);
    bordLenC = sqrt((b4o(1,1)-b3o(1,1))^2+(b4o(1,2)-b3o(1,2))^2+(b4o(1,3)-
b3o(1,3))^2);
    bordLenD = sqrt((b5o(1,1)-b4o(1,1))^2+(b5o(1,2)-b4o(1,2))^2+(b5o(1,3)-
b4o(1,3))^2);
    bordLenOriginal = bordLenA + bordLenB + bordLenC + bordLenD;


    %Strap Original Length (Add all 5 markers)
    strapLenA = sqrt((s2o(1,1)-s1o(1,1))^2+(s2o(1,2)-s1o(1,2))^2+(s2o(1,3)-
s1o(1,3))^2);
    strapLenB = sqrt((s3o(1,1)-s2o(1,1))^2+(s3o(1,2)-s2o(1,2))^2+(s3o(1,3)-
s2o(1,3))^2);
    strapLenC = sqrt((s4o(1,1)-s3o(1,1))^2+(s4o(1,2)-s3o(1,2))^2+(s4o(1,3)-
s3o(1,3))^2);
    strapLenD = sqrt((s5o(1,1)-s4o(1,1))^2+(s5o(1,2)-s4o(1,2))^2+(s5o(1,3)-
s4o(1,3))^2);
    strapLenOriginal = strapLenA + strapLenB + strapLenC + strapLenD;


    %Cord Final Length (Adds all 5 markers)
    cordLenE = sqrt((c2f(1,1)-c1f(1,1))^2+(c2f(1,2)-c1f(1,2))^2+(c2f(1,3)-
c1f(1,3))^2);
    cordLenF = sqrt((c3f(1,1)-c2f(1,1))^2+(c3f(1,2)-c2f(1,2))^2+(c3f(1,3)-
c2f(1,3))^2);
```

```matlab
    cordLenG = sqrt((c4f(1,1)-c3f(1,1))^2+(c4f(1,2)-c3f(1,2))^2+(c4f(1,3)-
c3f(1,3))^2);
    cordLenH = sqrt((c5f(1,1)-c4f(1,1))^2+(c5f(1,2)-c4f(1,2))^2+(c5f(1,3)-
c4f(1,3))^2);
    cordLenFinal = cordLenE + cordLenF + cordLenG + cordLenH;


    %Border Final Length (Adds all 5 markers)
    bordLenE = sqrt((b2f(1,1)-b1f(1,1))^2+(b2f(1,2)-b1f(1,2))^2+(b2f(1,3)-
b1f(1,3))^2);
    bordLenF = sqrt((b3f(1,1)-b2f(1,1))^2+(b3f(1,2)-b2f(1,2))^2+(b3f(1,3)-
b2f(1,3))^2);
    bordLenG = sqrt((b4f(1,1)-b3f(1,1))^2+(b4f(1,2)-b3f(1,2))^2+(b4f(1,3)-
b3f(1,3))^2);
    bordLenH = sqrt((b5f(1,1)-b4f(1,1))^2+(b5f(1,2)-b4f(1,2))^2+(b5f(1,3)-
b4f(1,3))^2);
    bordLenFinal = bordLenE + bordLenF + bordLenG + bordLenH;


    %Strap Length Final (Adds all 5 markers)
    strapLenE = sqrt((s2f(1,1)-s1f(1,1))^2+(s2f(1,2)-s1f(1,2))^2+(s2f(1,3)-
s1f(1,3))^2);
    strapLenF = sqrt((s3f(1,1)-s2f(1,1))^2+(s3f(1,2)-s2f(1,2))^2+(s3f(1,3)-
s2f(1,3))^2);
    strapLenG = sqrt((s4f(1,1)-s3f(1,1))^2+(s4f(1,2)-s3f(1,2))^2+(s4f(1,3)-
s3f(1,3))^2);
    strapLenH = sqrt((s5f(1,1)-s4f(1,1))^2+(s5f(1,2)-s4f(1,2))^2+(s5f(1,3)-
s4f(1,3))^2);
    strapLenFinal = strapLenE + strapLenF + strapLenG + strapLenH;


    %Calculate the Unloading strain
    %cordStrain = (cordLenFinal-cordLenOriginal)/cordLenOriginal
    %bordStrain = (bordLenFinal-bordLenOriginal)/bordLenOriginal
    %strapStrain = (strapLenFinal-strapLenOriginal)/strapLenOriginal


    cordT1 = ['For the CORD, the original length is ',
num2str(cordLenOriginal)];
    borderT1 = ['For the BORDER, the original length is ',
num2str(bordLenOriginal)];
    strapT1 = ['For the STRAP, the original length is ',
num2str(strapLenOriginal)];
    disp(cordT1)
    disp(borderT1)
    disp(strapT1)


    cordT2 = ['For the CORD, the final length is ', num2str(cordLenFinal)];
    borderT2 = ['For the BORDER, the final length is ',
num2str(bordLenFinal)];
    strapT2 = ['For the STRAP, the final length is ',
num2str(strapLenFinal)];
    disp(cordT2)
```

```
    disp(borderT2)
    disp(strapT2)

    disp(Filename)
    AAoutput = [cordLenOriginal bordLenOriginal strapLenOriginal cordLenFinal
bordLenFinal strapLenFinal]
```

Bibliography

1. Alpert, S. W., Pink, M. M., Jobe, F. W., McMahon, P. J., & Mathiyakom, W. (2000). Electromyographic analysis of deltoid and rotator cuff function under varying loads and speeds. Journal of shoulder and elbow surgery, 9(1), 47–58. https://doi.org/10.1016/s1058-2746(00)90009-0

2. Clark, J. M., & Harryman, D. T. (1992). Tendons, ligaments, and capsule of the rotator cuff. Gross and microscopic anatomy. The Journal of Bone & Joint Surgery, 74(5), 713–725. https://doi.org/10.2106/00004623-199274050-00010

3. Harryman, D. T., Mack, L. A., Wang, K. Y., Jackins, S. E., Richardson, M. L., & Matsen, F. A. (1991). Repairs of the rotator cuff. Correlation of functional results with integrity of the cuff. The Journal of Bone & Joint Surgery, 73(7), 982–989. https://doi.org/10.2106/00004623-199173070-00004

4. Itoi, E., Berglund, L. J., Grabowski, J. J., Schultz, F. M., Growney, E. S., Morrey, B. F., & An, K. N. (1995). Tensile properties of the supraspinatus tendon. Journal of orthopaedic research : official publication of the Orthopaedic Research Society, 13(4), 578–584. https://doi.org/10.1002/jor.1100130413

5. Jeong, J. Y., Min, S. K., Park, K. M., Park, Y. B., Han, K. J., & Yoo, J. C. (2018). Location of Rotator Cuff Tear Initiation: A Magnetic Resonance Imaging Study of 191 Shoulders. The American Journal of Sports Medicine, 46(3), 649–655. https://doi.org/10.1177/0363546517748925

6. Kedgley, A. E., Mackenzie, G. A., Ferreira, L. M., Drosdowech, D. S., King, G. J., Faber, K. J., & Johnson, J. A. (2007). The effect of muscle loading on the kinematics of in vitro glenohumeral abduction. Journal of biomechanics, 40(13), 2953–2960. https://doi.org/10.1016/j.jbiomech.2007.02.008

7. Kim, H. M., Dahiya, N., Teefey, S. A., Keener, J. D., Galatz, L. M., & Yamaguchi, K. (2010). Relationship of tear size and location to fatty degeneration of the rotator cuff. The Journal of bone and joint surgery. American volume, 92(4), 829–839. https://doi.org/10.2106/JBJS.H.01746

8. Largacha, M., Parsons, I. M., Campbell, B., Titelman, R. M., Smith, K. L., & Matsen, F. (2006). Deficits in shoulder function and general health associated with sixteen common shoulder diagnoses: A study of 2674 patients. Journal of Shoulder and Elbow Surgery, 15(1), 30–39. https://doi.org/10.1016/j.jse.2005.04.006

9. Lee, Y. S., & Lee, T. Q. (2010). Specimen-specific method for quantifying glenohumeral joint kinematics. *Annals of biomedical engineering*, *38*(10), 3226–3236. https://doi.org/10.1007/s10439-010-0074-7

10. McMahon, P. J., Debski, R. E., Thompson, W. O., Warner, J. J., Fu, F. H., & Woo, S. L. (1995). Shoulder muscle forces and tendon excursions during glenohumeral abduction in the scapular plane. Journal of shoulder and elbow surgery, 4(3), 199–208. https://doi.org/10.1016/s1058-2746(05)80052-7

11. Miniato MA, Anand P, Varacallo M. Anatomy, Shoulder and Upper Limb, Shoulder. [Updated 2020 Jul 31]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2020 Jan-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK536933/

12. Oh, J. H., Jun, B. J., McGarry, M. H., & Lee, T. Q. (2011). Does a critical rotator cuff tear stage exist?: a biomechanical study of rotator cuff tear progression in human cadaver shoulders. The Journal of bone and joint surgery. American volume, 93(22), 2100–2109. https://doi.org/10.2106/JBJS.J.00032

13. Omi, R., Sano, H., Ohnuma, M., Kishimoto, K. N., Watanuki, S., Tashiro, M., & Itoi, E. (2010). Function of the shoulder muscles during arm elevation: an assessment using positron emission tomography. Journal of anatomy, 216(5), 643–649. https://doi.org/10.1111/j.1469-7580.2010.01212.x

14. Pinkowsky, G. J., ElAttrache, N. S., Peterson, A. B., Akeda, M., McGarry, M. H., & Lee, T. Q. (2017). Partial-thickness tears involving the rotator cable lead to abnormal glenohumeral kinematics. Journal of shoulder and elbow surgery, 26(7), 1152–1158. https://doi.org/10.1016/j.jse.2016.12.063

15. Reed, D., Cathers, I., Halaki, M., & Ginn, K. (2013). Does supraspinatus initiate shoulder abduction? Journal of Electromyography and Kinesiology, 23(2), 425–429. https://doi.org/10.1016/j.jelekin.2012.11.008).

16. Roh, M. S., Wang, V. M., April, E. W., Pollock, R. G., Bigliani, L. U., & Flatow, E. L. (2000). Anterior and posterior musculotendinous anatomy of the supraspinatus. Journal of shoulder and elbow surgery, 9(5), 436–440. https://doi.org/10.1067/mse.2000.108387

17. Rybalko, D., Bobko, A., Amirouche, F., Peresada, D., Hussain, A., Patetta, M., Sood, A., Koh, J., & Goldberg, B. (2020). Biomechanics in an Incomplete Versus Complete Supraspinatus Tear: A Cadaveric Study. Orthopaedic journal of sports medicine, 8(12), 2325967120964476. https://doi.org/10.1177/2325967120964476

18. Schmidt, C. C., Spicer, C. S., Papadopoulos, D. V., Delserro, S. M., Tomizuka, Y., Zink, T. R., Blake, R. J., Smolinski, M. P., Miller, M. C., Greenwell, J. M., Carrazana-Suarez, L. F., & Smolinski, P. J. (2022). The Rotator Cable Does Not Stress Shield the Crescent Area During Shoulder Abduction. The Journal of bone and joint surgery. American volume, 104(14), 1292–1300. https://doi.org/10.2106/JBJS.21.01142

19. Thompson, W. O., Debski, R. E., Boardman, N. D., 3rd, Taskiran, E., Warner, J. J., Fu, F. H., & Woo, S. L. (1996). A biomechanical analysis of rotator cuff deficiency in a cadaveric model. The American journal of sports medicine, 24(3), 286–292. https://doi.org/10.1177/036354659602400307

20. Volk, A. G., & Vangsness, C. T., Jr (2001). An anatomic study of the supraspinatus muscle and tendon. Clinical orthopaedics and related research, (384), 280–285. https://doi.org/10.1097/00003086-200103000-00032