2007

Development and Validation of a Computational Musculoskeletal Model of the Elbow Joint

Justin Paul Fisk
Virginia Commonwealth University

Follow this and additional works at: https://scholarscompass.vcu.edu/etd
Part of the Biomedical Engineering and Bioengineering Commons

© The Author

Downloaded from https://scholarscompass.vcu.edu/etd/1434

This Thesis is brought to you for free and open access by the Graduate School at VCU Scholars Compass. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of VCU Scholars Compass. For more information, please contact libcompass@vcu.edu.
© Justin Paul Fisk, 2007

All Rights Reserved
DEVELOPMENT AND VALIDATION OF A COMPUTATIONAL MUSCULOSKELETAL MODEL OF THE ELBOW JOINT

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Biomedical Engineering at Virginia Commonwealth University.

by

Justin Paul Fisk
B.S., Rensselaer Polytechnic Institute, 2005

Director: Jennifer S. Wayne, Ph.D
Departments of Biomedical Engineering and Orthopaedic Surgery

Virginia Commonwealth University
Richmond, Virginia
August, 2007
ACKNOWLEDGEMENTS

I am very grateful to the many people who have helped to make this research possible. I would first like to thank my advisor, Dr. Jennifer Wayne, for providing me an opportunity to work in the Orthopaedic Research Laboratory at VCU. Her guidance, support, and wisdom have been indispensable. For taking the time and effort to serve on my committee, I want to acknowledge Dr. Douglas Boardman III and Dr. Tom Haas.

I also want to thank all members of the Orthopaedic Research Laboratory, fellow VCU graduate students, and past RPI classmates for their friendship and generosity. In addition, the experiences and flexibility provided by TriMech Solutions has been truly incredible. For this, I would like to recognize Jim Reidy, Mike Voll, Kelly Judson, Emmanuel Kim, and the technical support team.

I would particularly like to thank my fiancé, Brigid. Words cannot describe how appreciative I am for her encouragement, compassion, and sacrifice. For precisely the same reasons, I want to acknowledge my parents, Vicki and Dick. For all they have done and continue to do, I am grateful. I am also very fortunate to have the support and advice of a former biomedical engineer and older brother, Jesse. And for helping me to remember that life outside work does exist and can be quite enjoyable, I would like to thank my sister, Jenessa.

I would also like to thank Dr. Curtis Hayes and the VCU Department of Radiology for providing the CT scans which played such a central role in this research.
# TABLE OF CONTENTS

ACKNOWLEDGEMENTS........................................................................................................ ii

LIST OF TABLES................................................................................................................ vi

LIST OF FIGURES ............................................................................................................... vii

ABSTRACT............................................................................................................................ x

1. INTRODUCTION ........................................................................................................... 1
   1.1 MODELING OF MUSCULOSKELETAL SYSTEMS ........................................... 1
   1.2 COMPUTER METHODS IN MUSCULOSKELETAL MODELING .................... 5
   1.3 MODELING OF THE ELBOW JOINT COMPLEX ........................................... 8
   1.4 OBJECTIVE ............................................................................................................. 8

2. BACKGROUND ............................................................................................................. 10
   2.1 BONY ANATOMY .............................................................................................. 10
   2.2 JOINT MOTION .................................................................................................. 14
   2.3 LIGAMENTOUS RESTRAINTS .......................................................................... 17
   2.4 MUSCLE STRUCTURES .................................................................................. 22

3. BONY GEOMETRY RECONSTRUCTION .............................................................. 24
   3.1 OVERVIEW ........................................................................................................ 24
   3.2 COMPUTED TOMOGRAPHY ACQUISITION.................................................... 24
   3.3 VOXEL SEGMENTATION ................................................................................ 26
   3.4 SURFACE MODIFICATIONS .......................................................................... 32
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2.1</td>
<td>Skeletal Geometry</td>
<td>77</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Soft Tissue</td>
<td>78</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Testing Apparatus and Procedure</td>
<td>79</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Data Acquisition</td>
<td>81</td>
</tr>
<tr>
<td>6.3</td>
<td>RESULTS</td>
<td>82</td>
</tr>
<tr>
<td>6.4</td>
<td>DISCUSSION</td>
<td>93</td>
</tr>
<tr>
<td>7</td>
<td>CONCLUSION</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>APPENDIX A: MATLAB PROGRAM FOR GEOMETRY FITTING</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>APPENDIX B: COSMOSMOTION TIMER API</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>APPENDIX C: MATLAB PROGRAM FOR SDA FITTING</td>
<td>114</td>
</tr>
<tr>
<td></td>
<td>APPENDIX D: ELBOW DISSECTIONS</td>
<td>116</td>
</tr>
<tr>
<td></td>
<td>REFERENCES</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td>VITA</td>
<td>132</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 4.2.1: Solver parameters ................................................................. 48
Table 4.4.1: Ligament fiber properties ...................................................... 60
Table 4.6.1: Contact parameters .............................................................. 65
Table 5.3.1: Elbow flexion-extension range of motion ......................... 69
Table 6.3.1: Simulation solve time ......................................................... 93
LIST OF FIGURES

Figure 2.1.1: Upper extremity in 30 degrees of elbow flexion ................................. 10
Figure 2.1.2: Elbow in full extension ........................................................................ 11
Figure 2.1.3: Axial view of distal humerus ............................................................... 12
Figure 2.2.1: Screw displacement axis definition ...................................................... 16
Figure 2.3.1: Medial collateral ligament complex ..................................................... 18
Figure 2.3.2: Lateral ligament complex ................................................................. 19
Figure 2.3.3: Interosseous membrane ..................................................................... 21
Figure 3.2.1: CT positioning apparatus .................................................................. 25
Figure 3.3.1: Sagittal CT slice of elbow ................................................................. 27
Figure 3.3.2: Mimics mask and polylines ............................................................... 29
Figure 3.3.3: Fit surface of proximal humerus ....................................................... 30
Figure 3.3.4: CT slice depicting branching ............................................................. 31
Figure 3.4.1: Original and corrected surface representing distal radius ............... 33
Figure 3.4.2: Branch surface correction at distal humerus ...................................... 35
Figure 3.4.3: Hybridized skeletal structures in CT position ................................. 37
Figure 3.5.1: Geometry fitting graphical environment ......................................... 44
Figure 4.2.1: COSMOSMotion timer user interface .............................................. 49
Figure 4.3.1: Flexion-extension screw displacement axis ..................................... 53
Figure 4.3.2: Elbow flexion in CT position ........................................................... 54
Figure 4.3.3: Pronation-supination axis of rotation .......................................................... 55
Figure 4.4.1: Distal radioulnar and medial collateral ligament fibers .............................. 56
Figure 4.4.2: Annular ligament solid and lateral collateral ligament fibers ..................... 58
Figure 4.4.3: Interosseous membrane fibers ..................................................................... 59
Figure 4.5.1: Biceps, brachialis, and triceps muscle insertions ........................................ 63
Figure 5.3.1: Model in predicted extension position ........................................................ 67
Figure 5.3.2: Model in predicted flexion position ............................................................ 68
Figure 5.3.3: Contributions to joint stability at flexion and extension limits ................... 70
Figure 6.2.1: Coronoid process resection ......................................................................... 78
Figure 6.2.2: Soft tissue fibers of varus study model ....................................................... 79
Figure 6.2.3: Varus displacement test apparatus .............................................................. 80
Figure 6.2.4: Varus study computational model............................................................... 81
Figure 6.3.1: Characteristic axial load measured at varus displacement apparatus........... 83
Figure 6.3.2: Percent constraining load at 30 degrees flexion ........................................... 84
Figure 6.3.3: Percent constraining load at 45 degrees flexion .......................................... 84
Figure 6.3.4: Percent constraining load at 60 degrees flexion .......................................... 85
Figure 6.3.5: Percent constraining load at 90 degrees flexion .......................................... 85
Figure 6.3.6: Percent constraining load at 120 degrees flexion ......................................... 86
Figure 6.3.7: Mean constraining load ............................................................................... 88
Figure 6.3.8: Proximal ulnar translation ........................................................................... 89
Figure 6.3.9: Stabilizing force of the annular ligament ..................................................... 91
Figure 6.3.10: Contact force between the radial head and capitulum.............................. 92
Figure 6.4.1: Stabilization of elbow provided by radial head

Figure D.1: Anterior view of specimen 1

Figure D.2: Biceps brachii tendon

Figure D.3: Radial collateral ligament origin and insertion, separated

Figure D.4: Radial collateral ligament origin and insertion

Figure D.5: Radial head

Figure D.6: Annular ligament, specimen 1

Figure D.7: Cut annular ligament, specimen 1

Figure D.8: Medial collateral ligament origin

Figure D.9: Medial collateral ligament insertions

Figure D.10: Annular ligament, specimen 2

Figure D.11: Annular ligament, specimen 2

Figure D.12: Cut annular ligament, specimen 2
ABSTRACT

DEVELOPMENT AND VALIDATION OF A COMPUTATIONAL MUSCULOSKELETAL MODEL OF THE ELBOW JOINT

by Justin Paul Fisk, B.S.

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Biomedical Engineering at Virginia Commonwealth University.

Virginia Commonwealth University, 2007

Director: Jennifer S. Wayne, Ph.D
Departments of Biomedical Engineering and Orthopaedic Surgery

Musculoskeletal computational modeling is a versatile and effective tool which may be used to study joint mechanics, examine muscle and ligament function, and simulate surgical reconstructive procedures. While injury to the elbow joint can be significantly debilitating, questions still remain regarding its normal, pathologic, and repaired behavior. Biomechanical models of the elbow have been developed, but all have assumed fixed joint axes of rotation and ignored the effects of ligaments. Therefore, the objective of this thesis was to develop and validate a computational model of the elbow joint whereby joint kinematics are dictated by three-dimensional bony geometry contact, ligamentous constraints, and muscle loading.
Accurate three-dimensional bone geometry was generated by acquiring CT scans, segmenting the images to isolate skeletal features, and fitting surfaces to the segmented data. Ligaments were modeled as tension-only linear springs, and muscle were represented as force vectors with discrete attachment points. Bone contact was modeled by a routine which applied a normal force at points of penetration, with a force magnitude being a function of penetration depth. A rigid body dynamics simulator was used to predict the model’s behavior under particular external loading conditions.

The computational model was validated by simulating past experimental investigations and comparing results. Passive flexion-extension range of motion predicted by the model correlated exceptionally well with reported values. Bony and ligamentous structures responsible for enforcing motion limits also agreed with past observations. The model’s varus stability as a function of elbow flexion and coronoid process resection was also investigated. The trends predicted by the model matched those of the associated cadaver study.

This thesis successfully developed an accurate musculoskeletal computational model of the elbow joint complex. While the model may now be used in a predictive manner, further refinements may expand its applicability. These include accounting for the interference between soft tissue and bone, and representing the dynamic behavior of muscles.
1. INTRODUCTION

1.1 MODELING OF MUSCULOSKELETAL SYSTEMS

Computational modeling is a powerful and useful tool, with many applications in the field of biomechanics. Such models reduce the need for expensive experimental tests and a large pool of anatomic specimens. They are capable of measuring and calculating important parameters that are difficult if not impossible to capture experimentally. These virtual representations are also easily manipulated, seamlessly allowing the researcher to vary the parameters of interest.

Musculoskeletal models have been developed for a wide range of applications. Many researchers have used their models to study the contributions of specific muscles on joint moments [9, 47, 57, 63, 73, 96] and body segment motion [8, 45, 74, 80, 97]. Such mathematical models are particularly useful in these situations, as the indeterminate problem caused by muscle redundancy can be solved through optimization [8, 36, 46, 47, 74, 84, 85, 91, 97]. Joint stability and injury mechanisms have also been investigated through modeling approaches [22, 44, 61, 70, 104]. While joint contact areas and pressure distributions are difficult to measure experimentally, computational modeling has been adopted as an effective alternative [26, 44, 53, 60]. Likewise, in vivo ligament strains and forces have proven difficult to evaluate experimentally, but can be accurately predicted with musculoskeletal modeling [44, 53, 64, 91, 100]. Quantitative biomechanical evaluation of surgical reconstructive procedures through computational
modeling has also shown to be a valuable tool [26, 28, 59, 63]. Although not currently available, many researchers see important applications in patient-specific preoperative planning, computer-aided surgery, and computer-aided rehabilitation [20, 33, 54, 60, 101].

Commonly, biomechanical models are implemented through either finite element analysis (FEA) [44, 64, 97, 104] or rigid body dynamics (RBD) [8, 10, 22, 26, 36, 46, 47, 53, 59, 60, 63, 67, 70, 74, 80, 82, 84, 85, 91, 95, 100]. FEA is a widely used and highly valuable research tool, often applied to study joint contact areas and stress distributions. Both finite element analysis and rigid body dynamics methods are effective for investigating soft tissue function, but the latter is also used frequently to examine limb kinematics and joint behavior. While FEA models are based on concepts of continuum mechanics, RBD models take a different approach. Here, the equations of motion are applied, given a set of undeformable solid bodies under user-defined constraints. Thus the modeling technique used is largely dictated by the type of results the investigation is seeking.

While research objectives highly influence model choice, feasibility and time constraints may also factor into this decision. Unfortunately, the system of equations associated with finite element models typically contains an extraordinary number of unknown variables. As a result, such a simulation may require an impractical amount of time to complete, particularly if the model is solved over the time domain. Alternately, rigid body dynamics algorithms are highly efficient and often converge to a solution considerably faster than the corresponding FEA model [60]. Therefore, in situations
where bone deformations may be considered negligible and stress computations on joint surfaces unnecessary, rigid body dynamics is a particularly appealing approach to modeling a musculoskeletal structure.

Computational models based on rigid body motion may be constructed so as to solve the forward dynamics, inverse dynamics, or mixed dynamics problem. Under the forward dynamics problem, external forces/moments acting on the rigid bodies are prescribed, and resultant positions/orientations are solved for over the time domain. Conversely, given each body’s time-dependent position, the inverse dynamics problem determines the forces/moments required to produce such motion. Mixed dynamics models are more complex and require both the forward and inverse problems to be solved in combination [107]. Here, a portion of both force/moment and motion constraints are imposed, and unspecified parameters are solved for.

Many researchers have developed musculoskeletal models that pose the inverse dynamics problem [10, 67, 70, 87, 95, 97]. The required kinematic data are frequently gathered through optical motion capture techniques, whereby a subject fitted with an array of markers performs the movement under investigation. While such models have generated valuable results, their dependence on the capabilities of the subject makes it difficult to use the model in a robust or predictive manner. The study of pathomechanics and surgical reconstructive procedures through such means is particularly problematic, as the subject must be exposed to the factors under investigation. Furthermore, the accuracy of inverse dynamics models is often compromised by skin motion artifacts [35, 68].
Inherent to all musculoskeletal models are assumptions and simplifications which can potentially affect the accuracy of results. In particular, many researchers have constrained joint degrees of freedom and ignored the effects of ligaments [8-10, 22, 28, 36, 41, 46, 47, 54, 63, 67, 73, 74, 85, 91, 95]. Although some circumstances may warrant such generalizations, it is not always appropriate to presume a human joint will function in the same manner as a mechanical joint [12]. For example, the knee is often modeled as a hinge joint, despite known internal/external rotation and anteroposterior translation. In addition to decreasing the validity of model results, these simplifications prevent investigation of ligament function, joint laxity, and injury mechanisms.

Fewer biomechanical models have been developed where joint motion is dictated by articular surface contact, ligamentous constraints, and muscle loading [12, 13, 26, 44, 53, 59, 60, 64, 65, 80, 100]. A three-dimensional rigid body model of the knee developed by Wismans et al [100] was among the first to incorporate such features. Representing ligaments and condyle surface contact resulted in a more realistic description of knee behavior. However, satisfying the surface contact constraint was an absolute requirement, even if bone separation would normally occur. Consequently, evaluation of certain joint kinematics such as varus-valgus motions was not appropriate. Hirokawa [53] developed a similar model, but with the addition of geodesic lines to accurately depict how soft tissues wrap around bony geometry. Kwak et al [60] described a three-dimensional rigid body model where contact force was a function of surface overlap. Here, the greater the overlap, the greater the reaction force between the two interfering bodies. Comparable to this, Iwasaki et al. [59] used the discrete element analysis (DEA)
technique to model and predict the intricate behavior of carpal bones and ligaments. This approach maintains body rigidity, but uses a grid of compressive springs distributed over the articular surface to more accurately simulate and analyze contact stresses.

Regardless of specific modeling assumptions, the importance of model validation has been stressed by many [20, 27, 33, 60, 101]. Before a given computational model can be used in a predictive and meaningful manner, model results must be compared with experimental data. Only after there is agreement between these data can it be assured that the model is an accurate representation of the physical system. In its most basic form, geometry interference and range of motion assessments may be completed. A more thorough validation may include comparing the results of an experimentally-performed mechanical simulation with the equivalent test carried out on the model.

1.2 COMPUTER METHODS IN MUSCULOSKELETAL MODELING

The vast majority of musculoskeletal models require underlying computer code to develop and implement. This code is often exceptionally complex and demands the programmer to have comprehensive knowledge of the model’s constitutive equations and solution algorithms. In the case of rigid body dynamics, the task is to formulate and solve the differential algebraic equations dictated by the model’s inputs and constraints [18]. To date, most musculoskeletal simulation software is written in-house [53, 60, 84, 97, 100], thereby making it difficult to assess and utilize the model outside the laboratory where it was created. Furthermore, these programs require considerable time and resources to develop, unavoidably shifting the focus away from biomechanical analyses.
A small number of software applications specifically designed for the study of musculoskeletal systems have been created. Virtual Interactive Musculoskeletal System (VIMS) integrates a database of three-dimensional anatomic models with a collection of analysis tools, allowing a graphical, interactive environment to manipulate models and view results. The application is capable of representing bone, muscle, ligament, and any arbitrarily-shaped rigid geometry. This system has been used in a wide variety of studies including simulation of shoulder throwing activities, calculation of joint articular contact stresses, external fixator evaluation, and range of motion examination [19]. Although several types of analyses can be performed with VIMS, it requires the use of other proprietary software packages, several of which are no longer commercially available [20]. While VIMS is currently distributed among nine institutions [20], it is not accessible to the general public at this time.

Software for Interactive Musculoskeletal Modeling (SIMM) is another software application providing a graphical interface with which three-dimensional musculoskeletal models may be developed and studied. The system has been primarily utilized to elucidate the roles of specific muscles and how these roles vary under different geometric conditions [9, 27, 54, 73]. While the program is useful for static investigations, all joints are constrained to have particular degrees of freedom and dynamic analysis is not possible without additional proprietary software.

The modeling and analysis of musculoskeletal systems through commercially-available software is an appealing alternative to previously stated methods. Commercial software is often easily implemented, robust, widely accessible, and well-documented.
Taking advantage of these existing powerful graphical interfaces and proven FEA and
dynamics algorithms may allow the researcher to be focused on manipulating the model,
obtaining results, and making conclusions/predictions.

Recently, the use of SolidWorks (SolidWorks Corp, Concord, MA) in conjunction
with an add-in called COSMOSMotion (Structural Research and Analysis Corp, Santa
Monica, CA) has shown to be a valuable tool for accurately simulating joint behavior
[65]. While SolidWorks is a mainstream computer-aided design (CAD) tool,
COSMOSMotion provides three-dimensional rigid body dynamics simulation through
use of the ADAMS/Solver software package (MSC Software Corp, Santa Ana, CA).
Since its inception in the mid 1970’s [21], Automatic Dynamic Analysis of Mechanical
Systems (ADAMS) has been the basis of numerous musculoskeletal models [22, 63, 65,
80]. Because COSMOSMotion functionality is fully integrated within SolidWorks,
model development and simulation are easily performed, and results are readily
accessible in both graphical and numerical formats.

Importantly, COSMOSMotion is capable of modeling three-dimensional contact
between a set of arbitrarily-shaped bodies [2]. This provides the means to investigate
musculoskeletal structures where joint degrees of freedom are not restricted, but instead
articular surface geometry, ligamentous constraints, and applied loading are used to
dictate joint kinematics. Such an approach has the potential to increase model accuracy
and greatly expand applications of commercially-available rigid body dynamics software
in musculoskeletal systems.
1.3 MODELING OF THE ELBOW JOINT COMPLEX

The elbow has been recognized as the most important joint of the upper extremity [71]. This articulation greatly enhances the positions attainable by the hand, and is involved in many important tasks. Consequently, pain, injury, and motion impingement can be significantly debilitating. With this, the complexity of posttraumatic elbow instability [89], the high frequency of elbow dislocations [99], and the operative complications associated with elbow trauma [88] make this joint an important structure to study.

Many musculoskeletal models including the elbow have been developed [10, 36, 40, 41, 45-47, 54, 57, 63, 67, 73, 84, 85, 95], but all have constrained the articulations to have particular degrees of freedom and ignored the effects of ligaments. The majority of these studies investigated muscles and their interplay with body segment motion. Neglecting the effects of ligaments and off-axis body rotations reduced the complexity of the models to a manageable size, but also limited model usefulness. In addition to being potential sources of error, these assumptions prevent examination of elbow laxity, ligament function, and injury mechanisms. As such, the use of existing elbow models as a clinical tool for preoperative planning is doubtful.

1.4 OBJECTIVE

The objective of this thesis is to develop and validate a computational model of the elbow whereby joint behavior is dictated by three-dimensional bony geometry, ligamentous constraints, and muscle loading. The model will utilize commercially-available software to accurately represent musculoskeletal features and experimental
testing devices. A fully-integrated and highly efficient rigid body dynamics simulator will be implemented, capable of solving mixed dynamics problems. Model validation will include comparing the results of experimental studies with results obtained from the computational model.
2. BACKGROUND

2.1 BONY ANATOMY

Careful examination of the upper extremity’s skeletal structure is important, due to its significant influence on joint behavior. Such effects are exceptionally prominent at the elbow, as this is one of the most congruous joints in the body [4]. As depicted in figure 2.1.1, the elbow joint complex (EJC) incorporates all three long bones of the arm; the humerus, ulna, and radius. While the humerus is involved in shoulder and elbow function, the ulna and radius are both part of the elbow and wrist joints. These three skeletal structures and their associated articulations play an integral role in positioning and orienting the hand in three-dimensional space.

![Figure 2.1.1: View of the right upper extremity in 30 degrees of elbow flexion.](image-url)
The humerus is the longest bone in the arm and is located proximal to the ulna and radius. The humeral head forms a half spheroid and is covered with hyaline cartilage, allowing articulation with the glenoid fossa. While the shaft is cylindrical proximally, it becomes prismatic distally. As shown in figure 2.1.2, the humerus widens inferiorly, with the most mediolateral extents defined as the medial and lateral epicondyles.

The distal humerus contains several structures important to the elbow joint complex. Adjacent to the more prominent medial epicondyle is the trochlea. This surface resembles a pulley, as it is circular in sagittal cross-section and contains a central trochlear sulcus with medial and lateral lips (figure 2.1.3). Cartilage covers the trochlea.
over an arc of approximately 300 degrees [71], allowing articulation with the ulna’s
trochlear notch. The capitulum is positioned lateral to the trochlea, and approximates the
shape of a half sphere. This structure’s surface contains anterior and inferior
components, with its posterior limit demarked by a tubercle, as shown in figure 2.1.3.
The capitulum is covered with hyaline cartilage and articulates with the concave proximal
surface of the radial head. Located on the anterior aspect of the humerus, superior to the
trochlea and capitulum are the coronoid fossa and radial fossa. These depressions
provide clearance for the coronoid process and radial head at high angles of elbow
flexion. Likewise, the olecranon fossa on the posterior distal humerus accommodates the
ulna’s olecranon as the elbow becomes extended.

![Image](image.png)

**Figure 2.1.3: Axial view of the right distal humerus. Source: [71]**

The proximal ulna includes several features, many of which crucial to proper
elbow function. Most superiorly, the ulna projects forward, approximating the shape of a
beak. This formation’s posterior aspect is defined as the olecranon, while its anterior
surface is covered with hyaline cartilage and represents the most proximal portion of the
trochlear notch. Distal to this is the coronoid process, which projects anteriorly. The superior aspect of the coronoid process defines the distal limit of the trochlear notch. Cartilage covers this surface which continues proximally until reaching a narrow transverse strip. This rough non-articular strip delimits proximal and distal portions of the trochlear notch. From a lateral viewpoint, the trochlear notch forms an elliptical arc of approximately 190 degrees [71]. Its topography closely matches that of the trochlea, thereby permitting the ulna to articulate with the humerus. As shown in figure 2.1.2, the radial notch is located lateral and slightly distal to the coronoid process. This shallow, cartilage-covered depression has an arc of approximately 70 degrees [71] and allows articulation with the radial head.

As alluded to above, structures of the proximal radius are also important to elbow function. Most superiorly is the discoid-shaped radial head. The axis of this structure and the adjacent neck form a 15 degree angle with the radius’ diaphysis, oriented in a direction opposite the radial tuberosity [71]. The proximal aspect of the radial head is concave and forms a shallow cup. Hyaline cartilage envelops this surface which abuts and articulates with the capitulum. The radial head’s circumference is also covered with cartilage, continuous with that of the proximal depression. The perimeter of the radial head is congruent with the radial notch of the ulna, thus allowing the radial head rotation necessary for pronation and supination.

Also influencing elbow behavior are skeletal features distal to the joint. The shafts of the ulna and radius are triangular in cross section, and run roughly parallel to one another when the forearm is supinated. Interosseous borders are present on the
medial and lateral margins of the supinated radius and ulna shafts, respectively. While
the ulna tapers distally until reaching its head, the radius expands until terminating at its
distal limit. The lateral aspect of the ulnar head is convex and covered with cartilage.
This surface articulates with the concave ulnar notch located on the medial side of the
distal radius. This joint, along with those of the proximal radial head, permit forearm
supination and pronation.

2.2 JOINT MOTION

The upper extremity contains diarthrodial joints at the elbow and distal forearm.
These include the ulnohumeral, radiohumeral, proximal radioulnar, and distal radioulnar
articulations, which together are described as a trochoginglymoid joint [71]. The
resulting motion is considered to be in two discrete forms, known as flexion-extension
and pronation-supination.

Elbow flexion and extension is primarily dictated by the ulnohumeral joint, with
the trochlea of the distal humerus articulating with the trochlear notch of the proximal
ulna. While this relative motion is often characterized as a hinge joint, the accuracy and
finer details of this assumption are controversial. Traditionally, the ulnohumeral joint is
defined as a fixed axis of rotation (AOR) intersecting the lateral epicondyle and the
anteroinferior aspect of the medial epicondyle [16]. Beyond assuming the axis is
independent of flexion angle, this description has high error sensitivity because of the
relatively short distance between epicondyles. The International Society of
Biomechanics has offered two AOR approximations [103]. The more favored definition
is described as perpendicular to the plane formed by two line segments; the first segment
passing between the glenohumeral rotation center and the midpoint of the line connecting
the medial and lateral epicondyles, and the second segment connecting this midpoint to
the ulnar styloid when the elbow is flexed at 90 degrees.

More accurate flexion-extension kinematics have been realized by incorporating
the concept of a screw displacement axis (SDA). Depicted in figure 2.2.1, the
transformation of a body subject to this kinematic constraint can be decomposed into a
simultaneous rotation about and translation along the SDA, with a helix describing the
trajectory of any point fixed on the body. While some investigators believe a fixed screw
displacement axis is sufficient [15, 98, 105], others have observed considerable variation
of the instantaneous SDA with respect to elbow flexion angle [30, 32, 48]. In addition,
these studies have shown inter-specimen variability, and differences between active and
passive flexion. However, regardless of the fixed or instantaneous definition, many
researchers agree that the elbow functions as a “loose” hinge joint, with the average
rotation axis consistently passing through the geometric centers of the trochlea and
capitulum [15, 32, 71, 98, 105].
Forearm pronation and supination incorporates the radiohumeral, proximal radioulnar, and distal radioulnar articulations. At the elbow, the radiohumeral joint includes motion between the capitulum and the superior aspect of the radial head, while the proximal radioulnar joint is described as rotation of the radial head in the radial notch. Inferior to the elbow, the distal radioulnar joint is characterized by rotation of the distal radius about the ulnar head. The axis of rotation associated with pronation and supination is widely accepted as passing proximally through the center of the radial head and distally through the ulnar head [4, 71, 75, 98]. Extending this axis proximally, it intersects the ulnohumeral rotation axis at the center of the capitulum. As a result, elbow flexion and extension is independent of forearm supination and pronation [71, 105]. Unlike that of the ulnohumeral joint, forearm instantaneous axes of rotation show little variation with respect to forearm position [75, 98, 105]. While controversy exists,
rotation and translation of the ulna during supination-pronation is generally thought to be insignificant compared to movement of the radius [71, 75, 105]. Although the primary motion of the radius is rotatory, some investigators have observed a proximal translation with respect to the ulna during pronation [34, 71]. These findings suggest that pronation-supination motion would be more accurately represented by a screw displacement axis than a revolute joint.

2.3 LIGAMENTOUS RESTRAINTS

The humerus, ulna, and radius are stabilized by several passive structures. Particularly important, elbow motion is restrained by medial and lateral ligament complexes. Contributions to forearm stability are provided by a strong interosseous membrane between the radius and ulna. Located at the distal radioulnar joint, the triangular fibrocartilage complex can also impact elbow behavior.

The medial collateral ligament (MCL) complex consists of anterior, posterior, and transverse segments [25, 38, 72]. The anteroinferior surface of the medial epicondyle marks the origin of both the anterior and posterior bundles [72], with no appreciable attachments to the adjacent trochlea [79]. As depicted in figure 2.3.1, the anterior ligament has its insertion on a tubercle located at the medial margin of the coronoid process. The posterior bundle inserts more distally and broadly onto the medial aspect of the olecranon. Importantly, the anterior and posterior bundles of the medial collateral ligament have a significant elbow stabilizing effect in response to valgus displacement [71]. Some have also suggested the structures play a role in limiting elbow flexion and
extension. Conversely, the transverse portion is often well integrated with the joint capsule and provides negligible contributions to joint stability [72].

Figure 2.3.1: Medial collateral ligament complex. Source: [71]

The lateral collateral ligament (LCL) complex is less consistent between individuals, but often includes four distinct structures [71, 72]. The annular ligament attaches to the anterior margin of the radial notch, encircles approximately 80% of the radial head, and inserts on the posterior margin of the radial notch. This strong collagenous band is circular in axial cross-section and tapers distally, thus closely resembling the shape of a funnel. While some authors classify the accessory collateral ligament as part of the annular ligament or otherwise disregard it [49, 52, 76], others insist upon its discrete existence [17, 71, 72]. As shown in figure 2.3.2, this structure attaches to the crista supinatoris and blends proximally with the distal lateral rim of the annular ligament. The radial collateral ligament (RCL) originates from the anteroinferior aspect of the lateral epicondyle, fans out as it courses distally, and inserts
indistinguishably into the lateral portion of the annular ligament. While variable forms of
the lateral ulnar collateral ligament (LUCL) have been documented [94], this bundle is
consistently present. The LUCL and RCL both originate on the inferior surface of the
lateral epicondyle, and their fibers are not readily separable proximal to the annular
ligament. Located posterior to the RCL’s insertion, the lateral ulnar collateral ligament
superficially blends with the annular ligament. While posterior fibers may share their
insertion with the annular ligament, an appreciable portion of the LUCL attaches distally
and slightly posteriorly on the crista supinatoris. Because the components of the lateral
ligament complex are so intimately related, individual contributions to joint stability are
still controversial [52]. However, it is well known that the annular ligament stabilizes the
radial head by maintaining its contact with the radial notch. Furthermore, LCL
insufficiency often results in posterolateral rotatory instability (PLRI) [78].

Figure 2.3.2: Lateral ligament complex. Source: [71]
The interosseous membrane (IOM) is a strong thin collagenous sheet which inserts along the interosseous borders of the radius and ulna. This structure consists of a proximal band (also known as the oblique cord), central band (CB), accessory bands, and distal membranous portion [69, 92], as shown in figure 2.3.3. With the exception of the proximal band, the interosseous membrane runs distally and medially from its radial origin to its ulnar insertion. On average, its fibers are oriented 21 degrees to the long axis of the ulna [92]. Located more proximally and directed perpendicular to these fibers, the oblique cord is not consistently present [49]. Sharing its origin with the oblique cord, the central band fans out slightly as it progresses to its insertion. This portion is approximately twice as thick as all other regions [55]. Although variable in number, several accessory bands are located inferior to the central band. Between and distal to these bands are areas of membranous tissue. Among other structures, the interosseous membrane prevents proximal migration of the radius, with the majority of its strength attributed to the central band [55, 69]. It has also been suggested that the IOM acts as a stabilizer of the distal radioulnar joint [90]. Conversely, the oblique cord has been described as having limited functional significance [49, 71].
Figure 2.3.3: Medial view of the interosseous membrane of the forearm.

The triangular fibrocartilage complex (TFCC) contains several structures important to distal radioulnar articulation. The triangular fibrocartilage component of the TFCC is a triangular disk that separates the distal ulna from the carpus. This disk has dorsal and palmar thickenings composed of longitudinally oriented collagen fibers, defined as the dorsal and palmar distal radioulnar ligaments (DRULs) [58, 90]. These ligaments have a common attachment on the fovea of the ulna. The dorsal and palmar components of the DRUL insert along the dorsal and palmar margins of the ulnar notch, respectively. During pronation and supination, these structures glide over the distal end of the ulnar head and play a substantial role in stabilizing and guiding the distal radioulnar joint [90]. In addition, longitudinal stability of the radius is provided in part by the TFCC [55].
2.4 MUSCLE STRUCTURES

While 24 distinct muscles cross the elbow joint, only a select subset most significantly influencing the elbow will be discussed here. According to An et al [7], muscles with the largest flexion-extension moment potentials include the triceps brachii, brachialis, and biceps brachii.

The triceps brachii occupies the majority of the space posterior to the humerus and has three distinct origins. The long head has the most medial attachment, originating from the infraglenoid tubercle of the scapula. The lateral head attaches to the humerus along a thin linear strip superior to the radial groove. This band is directed longitudinally, with its proximal extent at the surgical neck. The medial head has a broad origin on the entire posterior surface of distal humerus, with proximal attachments toward the medial margin. These three segments begin to converge in the middle of the muscle, and ultimately insert on the posterior, proximal surface of the olecranon. The triceps is the largest extensor of the elbow.

The brachialis originates broadly on the anterior, distal half of the humerus. This region extends to the sharp medial and lateral borders of the humerus and inferiorly to within a few centimeters of the elbow joint capsule. Its fibers converge and insert more discretely on the ulnar tuberosity and base of the coronoid process. The brachialis acts as a flexor of the elbow. Although its large cross sectional area permits significant force generation, the muscle’s close proximity to the ulnohumeral axis of rotation hinders the resultant flexion moment.
Positioned more superficially on the anterior aspect of the upper arm, the biceps brachii has two origins. Attaching within the capsule of the shoulder joint, the long head wraps around the humeral head and descends in the intertubercular sulcus of the humerus. The short head originates from the apex of the coracoid process and blends with the long head approximately 7 centimeters proximal to the elbow. The biceps brachii has a well-defined insertion on the rough posterior aspect of the radial tuberosity, but also includes a broad medial attachment on the deep fascia of the flexor muscles known as the bicipital aponeurosis. In addition to being a powerful supinator, the biceps brachii is a significant elbow flexor, particularly when the forearm is supinated [7, 49].
3. BONY GEOMETRY RECONSTRUCTION

3.1 OVERVIEW

Because the present musculoskeletal modeling technique integrates articular contact, accurate three-dimensional (3D) reproduction of the upper extremity’s bones was crucial. This was accomplished by obtaining high resolution computed tomography (CT) scans from an anatomic specimen, isolating three-dimensional bony geometry, and fitting polynomial surfaces to this geometry. Advanced surfacing methods were then applied to translate these surfaces into a solid 3D representation of each bone. Bony landmarks and geometry fitting algorithms were utilized as a means to quantitatively describe upper extremity features and reference frames.

3.2 COMPUTED TOMOGRAPHY ACQUISITION

A fresh human cadaveric right upper extremity was obtained from a 91 year old female donor. The specimen was disarticulated at the shoulder joint, with all structures distal to the humeral head left intact. Passive flexion-extension and supination-pronation indicated normal elbow behavior, and no gross anatomic deformities were observed. The latter was subsequently confirmed by the CT scans.

A custom designed positioning apparatus was used to properly orient and stabilize the specimen as the CT scans were being performed. As shown in figure 3.2.1, the specimen was placed on the plywood base, lateral aspect facing down. The approximate
ulnohumeral axis of rotation was aligned with the nylon bolt, and the long axes of the humerus and forearm were oriented parallel with the proximal and distal bases, respectively. Dowels and cable ties were used to secure the specimen on the positioning device, maintaining the forearm in neutral rotation. Flexion angle markings on the base allowed the upper extremity to be accurately placed in the desired position.

Figure 3.2.1: Isometric view of positioning apparatus in 30 degrees of elbow flexion.

A total of three sets of computed tomography scans were acquired with a SOMATOM Sensation 64 helical scanner (Siemens AG, Forchheim, Germany). The protocol produced a series of 512 by 512 pixel images of the upper extremity, approximately axial in orientation. The first scan was of the entire specimen in 30 degrees of elbow flexion and neutral forearm rotation, with an interpolated slice distance of 2.0 mm. A second scan was performed with the specimen in precisely the same
orientation, but concentrated on the elbow joint. This set included data from 10 cm proximal to 10 cm distal of the elbow with a slice distance of 0.4 mm. The last scan was of the same resolution and also concentrated on the elbow region, but with the specimen at 90 degrees of flexion and neutral forearm rotation.

3.3 VOXEL SEGMENTATION

Several important steps were necessary to accurately segment and reproduce the bones of the upper extremity based on acquired CT data. For this purpose, a commercially available application called Mimics 9.1 (Materialise, Ann Arbor MI) was utilized. This software was able to automatically interpret each set of 2D DICOM (Digital Imaging and Communications in Medicine) files generated by the CT scanner. Because three CT sets were obtained, the following approach was repeated for each set.

This procedure began by isolating voxels corresponding to bone tissue. A voxel is the three dimensional equivalent of a pixel, where the grayscale value (proportional to the CT number in this case) within each cuboid is constant. Because the x-ray linear attenuation coefficient of bone is higher than that of other tissue, voxels representing bone were initially identified by applying a threshold. To do this, it was first necessary to create a mask in Mimics. Each mask has a binary on/off, enabled/disabled definition for every voxel in the CT set. A global threshold of 180 was applied, automatically turning on any mask voxels where the corresponding CT number was equal to or above this value.

Additional techniques were then applied to address the shortcomings of global thresholding. Because there is no definitive threshold value yielding solely bone, the
resulting mask contained many unwanted voxels in regions outside the bone cortex (figure 3.3.1). This issue was solved through use of the region growing tool. Here, the user selects a mask and an enabled voxel, and the protocol propagates voxels that are on in the original mask and continuous in three-dimensional space with the selected voxel. Also important, a separate mask was needed for each bone. Particularly around the elbow joint, the various bones were close enough in proximity (figure 3.3.1) that basic region growing would produce a single mask including all skeletal structures. Consequently, before a specific bone was isolated with region growing, adjacent enabled mask voxels belonging to other bones were manually turned off. A unique mask was produced for each bone by repeatedly applying this method.

Figure 3.3.1: Sagittal CT slice of the elbow showing unwanted voxels (green) and region grown voxels (pink). Note the close proximity of the humerus and ulna.
Automated and manual tools were used to define the outer perimeter of each bone. For a given mask, Mimics is able to analyze each CT slice and produce a set of 2D contours called polylines that enclose enabled voxels. Using this feature, the resultant polylines were consistently well defined near the bone’s diaphysis, but often required manual editing elsewhere. Particularly at epiphyses, bone porosity caused highly localized regions of decreased CT number. As shown in figure 3.3.2, such areas introduced cavitations in the cortical polylines. To correct this, each CT slice was examined and a lower local mask threshold was manually applied to such areas. This was followed by updating the polylines to reflect the modified mask.

Cortical polylines were then isolated from the smaller unwanted medullary polylines (figure 3.3.2). This process was automated by use of the polyline growing tool. Starting with a user-selected contour, the routine creates a new polyline set by sequentially analyzing adjacent CT slices and adding polylines similar in shape, size, and position. In this manner, the majority of the cortical contours for a given bone were automatically copied to a new polyline set. This set was then manually appended to also include any cortical polylines not recognized by the polyline growing protocol.
An optimum three-dimensional surface was fit to each set of cortical polylines. As shown in figure 3.3.3, the resultant surface contour at each CT slice plane illustrates how the surface is a smoothed approximation of the actual polylines. Mimics utilizes the NURBS (non-uniform rational B-spline) surface representation, which requires specific properties to be defined by the user. These parameters include the order and number of control points in both the u (longitudinal) and v (transverse) directions. Unfortunately, optimum sets of values varied with the geometry being represented. While lower values could not capture small features and sharp edges, high values produced undulating and sometimes self-intersecting surfaces. It was therefore necessary to apply a trial and error approach for each surface. On average, fifth order surfaces with 43 and 36 control points
in the u and v directions respectively were used to represent the long bones of the upper extremity.

**Figure 3.3.3:** View of proximal humerus showing polylines (magenta), fit surface (yellow), and contours of fit surface (black). For clarity, intermediate polylines and contours are not shown.

Several surface fitting limitations required the use of novel approaches. While Mimics was capable of generating a surface connected in the v direction, it was unable to close the surface at the two longitudinal limits of the polyline set. Because the CT slices were aligned approximately transverse to the arm, this meant that the proximal and distal extents of each surface were left open. Methods to correct this shortcoming are discussed in section 3.4.
More problematic, Mimics is unable to account for branching. This phenomenon is equivalent to the way in which a tree’s trunk splits off into several unique branches. Tracking the cortical polylines distally or proximally from the bone diaphysis, the CT slices often transformed from containing one closed contour to containing several, depending on the complexity of the bone surfaces within the joint (figure 3.3.4). It was therefore necessary to create multiple surfaces for each bone. The primary surface extended proximally and distally up to the onset of branching. Beyond this, individual surfaces were fit to each branch. To ensure C1 continuity between these surfaces and their adjacent primary surface, the branch polyline sets, and thus the fit surfaces, included several CT slices also belonging to the primary surface. Similar to branching, internal cavities required the use of additional surfaces. Methods of producing a single bone based on these various surfaces are presented in section 3.4.

Figure 3.3.4: CT view of the distal humerus depicting branching. Visible are polylines of the capitulum (yellow), trochlea (teal, tan), and olecranon (black).
3.4 SURFACE MODIFICATIONS

A commercially-available computer-aided design (CAD) package known as SolidWorks 2006 (SolidWorks Corporation, Concord, MA) provided the means to import and modify geometry developed in Mimics. Through use of the IGES (Initial Graphics Exchange Specification) file format as an intermediary, all polylines and surfaces generated by Mimics were successfully interpreted by SolidWorks. Because this data was produced with respect to the CT reference frame, polylines and surfaces from a given CT set were superimposed without requiring repositioning. Each bone was modeled as a separate SolidWorks part file and then assembled.

Limitations of Mimics discussed in section 3.3 necessitated the use of SolidWorks’ advanced surfacing techniques. To close each skeletal structure’s surface at its longitudinal extents, the trim surface and filled surface features were primarily utilized. Because the surface fitting protocol used in Mimics had no information beyond the last polyline, the surface tangency at this CT slice plane was consistently incorrect, as depicted in figures 3.4.1 and 3.4.2A. It was therefore necessary to remove this inaccurate geometry prior to patching. Surface regions outside a plane offset inward from the last polyline’s slice plane were removed. Although the offset distance varied with geometry, less than 1mm in the longitudinal direction was normally trimmed away. The resulting edges were then used to create a filled surface, thereby closing the longitudinal extent of the bone. To ensure the surface accurately represented the actual skeletal geometry, two constraints were enforced. Firstly, the filled surface was defined to be tangent to the adjacent existing surface. Furthermore, it was required to be in contact with one or more
guide curves. As shown in figure 3.4.1, these curves were created in the form of 3D splines. Viewed from the transverse plane, each spline began on the edge of the original surface, intersected the bone’s apex (defined transversely as the center of the last polyline), and ended again on the edge of the original surface, but in a position opposite the first endpoint. In three-dimensional space, all splines were defined to be C1-continuous, at their endpoints, with the adjacent existing surface. Importantly, as each spline crossed a cortical polyline, its path was constrained to be in as close proximity as possible to the polyline.

Figure 3.4.1: View of a branch at the distal radius showing original (left) and corrected (right) surface. Visible are the unmodified surface (dark yellow), surface cap (red), CT polylines (solid black), two guide curves (solid yellow) and surfaces contours (dashed blue).

As explained in section 3.3, branching of CT polylines was resolved in Mimics by creating a primary surface and an additional surface for each branch. SolidWorks was
used to merge these various surfaces together in order to represent each bone as a single solid body. While the techniques to accomplish this were geometry-specific, several generic steps were frequently applied. First, the longitudinal extents of each branch were cut back and closed, as described previously. For reasons explained in section 3.3, unmodified primary and branch surfaces always contained a region of overlap (figure 3.4.2A). Within these areas, contours of the primary surface and branch surface did not exactly coincide, as all surfaces were polynomial approximations. However minor, these differences required the creation of a transition region. The primary and branch surfaces were trimmed back in their respectful directions, leaving a 1 to 2mm longitudinal gap. The mutual trim tool was then applied to remove the remaining overlapping surface portions which existed between branches. This resulted in sharp inter-branch edges, as shown in figure 3.4.2B. To more accurately define the smooth transition between branches, each surface was cut back approximately 1mm from the intersection edge, and one or more filled surface features with C1 continuity were produced.
Figure 3.4.2: View of the distal humerus showing steps taken to correct branched surfaces. Visible are the primary surface (dark yellow), surface branches (shades of blue), surface caps (red), inter-branch transition surfaces (green), transition surfaces (pink), branch intersection contours (dashed black), and guide curves (solid yellow).

The transition region between branches and the primary surface was also smoothed by means of the filled surface tool. Because of SolidWorks limitations, a single transition surface closed in the transverse direction could not be created. As a result, a series of smaller surface patches coursing circumferentially and extending from
branch to primary surface were created. Shown in figure 3.4.2D, the distal humerus required 12 such surfaces. Analogous to the capping method described previously, guide curves allowed control of transition surface patches, and C1 continuity was prescribed at the borders of neighboring surfaces. The path of each guide curve was dictated by CT cortical polylines.

Beyond controlling guide curves as described above, CT cortical polylines helped to qualitatively assess the level of fit provided by imported surfaces. Although rare, some localized surface regions deviated significantly from these polylines. As illustrated in figures 3.4.2C and D, surface trimming was necessary at a lateral area on the lateral branch of the humerus and the posteromedial aspect of the medial trochlear lip. By following this removal with patching described above, inaccurate surface correction and transition region smoothing were accomplished simultaneously.

It should be noted that the above method is a brief overview of the surfacing techniques employed. Additional complexity was frequently encountered. For example, the proximal ulna required simultaneous transitioning of the ulnar shaft, distal extension of the olecranon, coronoid process, and cavity formed by the medial portion of the trochlear notch. Also challenging, the distal radius contained four ill demarcated branches whose internal boundaries combined to form a single depression within which the carpal bones resided. Substantial manual alterations were necessary to accurately represent this transition.

In order to completely enclose each bone, the preceding techniques were repeatedly applied. The knit surface command was then utilized to merge all surfaces
and produce a solid three-dimensional body. After every bone was knit, models originating from the two CT scans in 30 degrees of flexion were combined. As discussed in section 3.2, one scan was of lower resolution but included the entire upper extremity, while the other was of higher resolution but focused on the elbow joint. By removing the elbow region in the complete scan and splicing in the more refined elbow geometry, a hybridized model was created. This optimum model included the complete long bone structures, but with greater detail at the elbow (figure 3.4.3). Because the orientation of the specimen in the global CT reference frame was consistent between 30 degree scans, only a 3D translation was necessary to properly align geometry. Although this splicing process was applied to each SolidWorks part file individually, the same translation was used in all cases.

Figure 3.4.3: Anteromedial view of the hybridized skeletal structures in CT position. Visible are the humerus (red), ulna (blue), and radius (yellow). Geometry originating from the 2mm slice interval CT set is shown in green.
For computational efficiency, parametric model information was removed by exporting and again importing each hybrid bone. The possibility of data translation errors was minimized by use of the parasolid format, as SolidWorks files are based off the same kernel. These reimported solid bodies were then combined by creating a SolidWorks assembly, thereby forming the skeletal structures used throughout this research.

3.5 GEOMETRY FITTING

In order to accurately describe certain skeletal features, a program was developed to fit primitive three-dimensional shapes to freeform geometry. More specifically, MATLAB R14 (The MathWorks, Inc, Natick, MA) was used to create an optimization tool capable of fitting a sphere, 3D line, cylinder, or plane to point cloud data. While similar functionality is available in the MedCAD module of Mimics [3], its capabilities are significantly limited.

This program was successfully used to accurately describe upper extremity skeletal features. The long axis of a bone was calculated as the centerline of a cylinder fit to data points positioned along the surface of the diaphysis (figure 3.5.1B). This axis was determined for the humerus, radius, and ulna. In addition, independent spheres were fit to the humeral head (figure 3.5.1A) and capitulum. Sphere, cylinder, and plane optimizations were all applied to portions of the radial head, with each description providing its own unique information. The orientation of the trochlear notch was also quantitatively defined. This was accomplished by fitting a plane to the structure’s prominent central ridge.
The developed geometry fitting routine requires a Microsoft Excel spreadsheet containing the Cartesian coordinates of all data points (see appendix A). This file may be created from any source and may include any conceivable combination of points. In SolidWorks, a single 3D sketch containing arbitrary points constrained to coincide with the bone’s surface was produced. These points were easily added, removed, and modified, thus allowing the geometry of interest to be isolated. A SolidWorks application programming interface (API) was then used to export these sketch point coordinates to Excel. The position vector of the $i^{th}$ point was defined as:

$$\vec{P}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$$

(1)

This routine is able to calculate the optimum geometric parameters through nonlinear unconstrained minimization of an objective function $\Phi$. The program makes use of the Nelder-Mead Simplex Method [62] provided in MATLAB’s optimization toolbox. An initial guess dependent upon geometry type is required by the user, as described in appendix A. However, the geometry fitting program produced the same solution regardless of initial guess value.

For a sphere, the sum of the squared distances from each point to the sphere surface was minimized. This was accomplished by calculating the distance from the sphere center $C$ to the point $P_i$, and then subtracting the radius $R$. The objective function $\Phi$ was therefore dependent upon choice of data points $[x_i, y_i, z_i]$, sphere center $[C_x, C_y, C_z]$, and sphere radius $R$: 
\[
\text{Min} \left[ \Phi(C_x, C_y, C_z, R) \right], \quad \Phi = \sum_{i=1}^{n} \left( \vec{P}_i - \vec{C} \right)^2
\] (2)

where,
\[
\vec{C} = \begin{bmatrix}
C_x \\
C_y \\
C_z 
\end{bmatrix}
\]

For line and cylinder fitting, the parametric equation of a line was utilized. If a given point \( P_0 \) on the line and the orientation of the line \( V \) are known, this equation allows an infinite number of points \( P_t \) on the line to be evaluated by varying the independent variable \( t \) as follows:
\[
\vec{P}_t = \vec{P}_0 + \vec{V}t
\] (3)

where,
\[
\vec{P}_t = \begin{bmatrix}
P_{tx} \\
P_{ty} \\
P_{tz}
\end{bmatrix}, \quad \vec{P}_0 = \begin{bmatrix}
P_{0x} \\
P_{0y} \\
P_{0z}
\end{bmatrix}, \quad \vec{V} = \begin{bmatrix}
V_x \\
V_y \\
V_z
\end{bmatrix}
\]

Although the 3D line fitting function was not applied to the current computational model, it formed the basis of the cylinder fitting algorithm and may have future uses. This routine minimized the sum of the squared distances from each point to the line. Because the parametric line equation can be defined with any \( P_0 \) and any magnitude of \( V \), constraints were imposed allowing the solver to converge on a unique set of parameters. By setting \( P_{0z} \) equal to zero, \( P_0 \) was restricted to lie on the xy plane. A vector identical in direction to \( V \), but with a z-component of unity was defined as \( V_2 \) and used in the optimization protocol. It is important to note that this formulation prevents a precise
solution if the line is exactly parallel to the xy plane. While equation 4 may not be in its most compact form, the terms of the objective function have straightforward geometric meaning. While the first two terms result in a vector from \( P_0 \) to \( P_i \), the latter terms give a vector from \( P_0 \) to the projection of \( P_i \) onto the line. Taking the difference yields the normal distance between point \( P_i \) and the line. The objective function \( \Phi \) was therefore dependent upon choice of data points \([x_i, y_i, z_i]\) and reduced properties of the 3D line \((P_{0x}, P_{0y}, V_{2,1}, V_{2,2})\):

\[
\min \left[ \Phi(P_{0x}, P_{0y}, V_{2,1}, V_{2,2}) \right], \quad \Phi = \sum_{i=1}^{n} \left| \vec{P}_i - \vec{P}_0 - \frac{\vec{V}_2}{\|\vec{V}_2\|} \left( \frac{\vec{P}_i - \vec{P}_0}{\|\vec{P}_i - \vec{P}_0\|} \right) \right|^2
\]  

(4)

where,

\[
\vec{P}_0 = \begin{bmatrix} P_{0x} \\ P_{0y} \\ 0 \end{bmatrix}, \quad \vec{V}_2 = \begin{bmatrix} V_{2,1} \\ V_{2,2} \\ V_{2,3} \end{bmatrix} = \begin{bmatrix} V_x/V_z \\ V_y/V_z \\ 1 \end{bmatrix}
\]

Comparable to the 3D line, cylinder optimization minimized the sum of the squared distances from each point to the cylinder surface. Here, the cylinder is represented as a line with a constant perpendicular radius \( R \). By modeling the cylinder centerline in the same manner as the above 3D line, the resulting objective function only differed by the term \( R \). By subtracting the radius scalar from the perpendicular distance before squaring and summing, the optimization maximized the cylinder surface’s proximity to the set of data points. The objective function \( \Phi \) was dependent upon choice of data points \([x_i, y_i, z_i]\), reduced centerline properties \((P_{0x}, P_{0y}, V_{2,1}, V_{2,2})\), and radius \( R \):
Min[Φ(P_{0x}, P_{0y}, V_{2,1}, V_{2,2}, R)], \quad Φ = \sum_{i=1}^{n} \left( \frac{\vec{P}_i - \vec{P}_0}{|\vec{V}_2|} - \left( \frac{\vec{P}_i - \vec{P}_0}{|\vec{V}_2|} \cdot \frac{\vec{V}_2}{|\vec{V}_2|} \right) - R \right)^2 \quad (5)

where,

\[
\begin{bmatrix}
P_{0x} \\
\vdots \\
0
\end{bmatrix}, \quad \begin{bmatrix}
V_{2,1} \\
V_{2,2} \\
\vdots \quad \vdots
\end{bmatrix} = \begin{bmatrix}
V_x / V_z \\
V_y / V_z \\
\vdots \\
1
\end{bmatrix}
\]

The plane optimization routine made use of the plane equation. Variables A, B, and C constitute the components of vector V oriented normal to the plane, whereas x, y, and z are the coordinates of a point lying on the plane.

Ax + By + Cz + D = 0 \quad (6)

where,

\[
\vec{V} = \begin{bmatrix}
A \\
B \\
C
\end{bmatrix}
\]

The plane optimization minimized the sum of the squared distances from each data point to the plane. Identical to the parametric equation of a line, vector V required its magnitude to be constrained. This was once more accomplished by requiring a z-component of unity. To provide a geometrically meaningful definition for D, it has been written in terms of D_{off}, the offset distance of the plane from the origin. If there is one sample point (n=1) and the objective function is zero (ideal convergence), algebraic substitution and expansion of equation 7 reveals that it is equivalent to the plane equation given in equation 6. The objective function Φ was dependent upon choice of data points.
\([x_i, y_i, z_i]\), components of the reduced normal vector \((V_{2,1}, V_{2,2})\), and the distance from the origin to the plane along the direction of \(V\) \((D_{\text{off}})\):

\[
\text{Min} \Phi\left(V_{2,1}, V_{2,2}, D_{\text{off}}\right) = \sum_{i=1}^{n} \left( \frac{V_2}{\|V_2\|} \cdot P_i - D_{\text{off}} \right)^2
\]

(7)

where,

\[
\vec{V}_2 = \begin{bmatrix} V_{2,1} \\ V_{2,2} \\ V_{2,3} \end{bmatrix} = \begin{bmatrix} A/C \\ B/C \\ 1 \end{bmatrix}, \quad D_{\text{off}} = -\frac{D}{\|V\|}
\]

In addition to providing numerical solutions, this geometry fitting program contains an intelligent system to simultaneously view the input data points and optimized geometry (figure 3.5.1). This subroutine rotates the view one revolution, and then allows the user to interactively pan, zoom, and rotate. While the objective function’s value at convergence is also always provided, this graphical environment allows for an efficient qualitative examination of geometry suitability.
Figure 3.5.1: View of the geometry fitting graphical environment showing a sphere fit to the humeral head (A) and a cylinder fit to the humeral shaft (B).

3.6 REFERENCE FRAMES

Generating frames of reference for the various upper extremity bones proved to be an important step affecting subsequent model definitions. Such frames aided in the creation of soft tissue attachments, characterization of joint parameters, placement and orientation of external perturbations, and capture of dynamic simulation data.

The anteroposterior (AP) plane of the humerus was defined as containing the glenohumeral rotation center and the most caudal points of the medial and lateral epicondyles. Because this rotation center coincides with the geometric center of the humeral head [98], methods described in section 3.5 were applied to determine this point. The mediolateral (ML) plane of the humerus was produced by rotating the
anteroposterior plane 90 degrees about the long axis of the humerus. This axis was defined as passing between the glenohumeral center and the midpoint of the line connecting the medial and lateral epicondyles. While the fit cylinder representation could have instead been used, the angle between these two long axis descriptions deviated by a mere 2.14 degrees. The transverse plane was then taken as normal to the long axis and containing the humerus center of mass. The orientations of these planes coincide with the humerus coordinate system suggested by the International Society of Biomechanics (ISB) [103].

A reference frame for the ulna was also created. The AP plane was defined as containing the long axis of the ulna and aligned parallel to the ulnohumeral screw displacement axis calculated in section 4.3. In this case, the long axis was characterized by the centerline of a cylinder fit to the ulnar shaft through methods discussed in section 3.5. The mediolateral plane was created by a 90 degree rotation of the AP plane about the ulna long axis. Consistent with that of the humerus, the transverse plane was defined as normal to the long axis and containing the center of mass. Because a limited number of bony landmarks exist in the forearm and ISB coordinate systems are defined by such markers [103], the reference frame definitions created here are thought to be of greater meaning and utility.

While a general reference frame was produced for the radius, it had limited application in the current work. The long axis was again defined through cylinder fitting, with the transverse plane normal to the centerline at the center of mass.
4. COMPUTATIONAL MODEL FORMULATION

4.1 OVERVIEW

By integrating and supplementing the geometry developed in chapter 1, a computational musculoskeletal model of the elbow and forearm was generated. While SolidWorks 2006 (SolidWorks Corporation, Concord, MA) provided the means to create, alter, reposition, and visualize model structures, COSMOSMotion 2006 (Structural Research & Analysis Corp, Santa Monica, CA) was implemented to prescribe force/motion constraints and simulate rigid body dynamics (RBD) of the resulting model. By defining idealized elbow and forearm joints in SolidWorks, the initial position of the RBD model could be adjusted as needed. While ligaments were represented by linear tension-only dampened springs, muscles were modeled as constant magnitude force vectors. To represent the normal force produced when two bones collide, three-dimensional contact constraints were defined. COSMOSMotion was then utilized to repetitively solve and integrate the equations of motion in order to simulate rigid body dynamics. By allowing each bone to move in all six degrees of freedom, joint motion was dictated by three-dimensional bony geometry contact, ligaments, muscles, and prescribed external perturbations.
4.2 COSMOSMOTION

A commercially-available software package known as COSMOSMotion was used to generate and simulate the upper extremity musculoskeletal model though use of three-dimensional rigid body dynamics. Because this application interfaced exceptionally well with SolidWorks, 3D geometry was easily transferred to COSMOSMotion. For this to occur, it was first necessary to model each bone as a distinct part file and then combine the structures in the form of a SolidWorks assembly. The position of the parts in the assembly dictated the initial position of the rigid body dynamics model. By identifying at least one body as “moving” in COSMOSMotion, any combination of force and motion constraints could be defined.

At the core of COSMOSMotion is the ADAMS/Solver software package (MSC Software, Santa Ana, CA). Internally, this software is responsible for simulating the dynamic behavior of the computational model. While several solvers are available, the widely-accepted and well-documented GSTIFF integrator was chosen. This efficient predictor-corrector method is based off the work of Gear [42, 43]. By representing the system with Lagrange’s equation of the first kind [106], the iterative routine is able to numerically solve the equations of motion by reducing the time step size until solution error falls below a particular threshold [43]. It was therefore important to specify integrator parameters appropriately. By trial and error, the values given in table 4.2.1 were identified as those that resulted in the highest accuracy without causing integration failures or requiring an unrealistic solution time. Rigid body mass properties were found to be directly related to such integration failures. As a result, the density of the annular
ligament solid developed in section 4.4.1 was increased considerably prior to finalizing solver parameters.

Table 4.2.1: Solver parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrator</td>
<td>GSTIFF</td>
</tr>
<tr>
<td>Maximum Iteration</td>
<td>50</td>
</tr>
<tr>
<td>Initial Time Step</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td>Minimum Time Step</td>
<td>$10^{-9}$</td>
</tr>
<tr>
<td>Maximum Time Step</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>Accuracy</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td>Jacobian Pattern</td>
<td>100%</td>
</tr>
<tr>
<td>Adaptivity</td>
<td>0</td>
</tr>
</tbody>
</table>

Simulation accuracy was ensured by appropriately selecting solver parameters. The minimum time step was set to the lowest possible value, thus allowing short-lived events, such as rigid body contact, to be captured. The accuracy parameter corresponds to the acceptable amount of solution error at each time step, and was therefore set as low as possible. By setting the Jacobian pattern to 100%, the Jacobian matrix was updated after every iteration. To minimize the possibility of the solver missing a rigid body contact, adaptivity was disabled by setting its value to zero.

After simulating a given model, its results were easily accessible in both visual and numerical formats. Animations allowed a qualitative evaluation of system behavior, while the software’s Excel export option provided detailed data on specific model elements. In addition, a simple API was created in Microsoft Visual Basic to precisely measure the amount of time required by COSMOSMotion to solve a given simulation.
This program’s code and associated user interface are provided in appendix B and figure 4.2.1, respectively. It should be noted that the desired number of simulation frames are specified by the user in the COSMOSMotion timer application, but the solver automatically adjusts this value to obtain the desired simulation accuracy.

![COSMOSMotion timer user interface](image)

**Figure 4.2.1: COSMOSMotion timer user interface.**

Several modeling assumptions were made with respect to COSMOSMotion. Although densities of the rigid bodies were not realistically defined, such discrepancies were disregarded for two reasons. Firstly, no investigations of dynamic model behavior were performed, and secondly, the effects of gravity were neglected. Furthermore, the frictional effects between bodies in contact, as defined in section 4.6, were assumed to be negligible. This approximation was justified based on the exceptionally low coefficient of friction found within synovial joints [37, 83].
4.3 JOINT PARAMETERIZATION

As discussed in section 3.2, geometry of the upper extremity was only captured in two positions; 30 and 90 degrees of elbow flexion, with forearm rotation approximately neutral. In order to reposition the model in any combination of flexion-extension and pronation-supination, idealized motion constraints were defined. Bone positions that resulted from these three-dimensional transformations were assumed to be those of the passive arm and were used as the initial positions in subsequent rigid body dynamics simulations. In addition, ligaments required passive upper extremity kinematics as a means of determining fiber stress-free lengths, as described in section 4.4.2.

4.3.1 Elbow Flexion-Extension

While many elbow flexion-extension definitions exist (see section 2.2), this motion was described in the current model through use of a screw displacement axis (SDA). By assuming this motion is independent of forearm rotation, the SDA was calculated by a comparison of the humerus and ulna in positions acquired from the CT scans described in section 3.2. This methodology was chosen because joint parameters derived from experimental motion data have been shown to be more accurate than definitions based solely on anatomical landmarks [87].

Motion between the humerus and ulna was quantified by first defining each bone’s position and orientation in both 30 and 90 degree CT scans. A centerline was fit to each humeral shaft by methods discussed in section 3.5. A vector perpendicular to this line and containing the medial epicondyle was then defined. Similarly, a centerline was fit to each ulnar shaft, followed by the creation of a perpendicular vector containing the
tip of the coronoid process. Based on these vectors, the same 3 points were identified on each model of the ulna. In SolidWorks, the 30 and 90 degree upper extremity assemblies were then combined to create a higher-level assembly. With the 30 degree model fixed, the 90 degree model was repositioned by aligning its humerus vectors with those of the 30 degree humerus. This allowed the relative movement of the ulna from 30 to 90 degrees of flexion to be evaluated.

The screw displacement axis was calculated using methods described by Beggs [11], which have also been applied to the elbow by Bottlang et al [15]. The transformation of point 1 \([P1_x, P1_y, P1_z]\) to point 2 \([P2_x, P2_y, P2_z]\) by applying a rotation of angle \(\sigma\) about the SDA and translation of magnitude \(s\) along the SDA is given by:

\[
\begin{bmatrix}
P2_x \\
P2_y \\
P2_z
\end{bmatrix}
=
\begin{bmatrix}
1 & 0 & 0 & 0 \\
S_{21} & S_{22} & S_{23} & S_{24} \\
S_{31} & S_{32} & S_{33} & S_{34} \\
S_{41} & S_{42} & S_{43} & S_{44}
\end{bmatrix}
\begin{bmatrix}
P1_x \\
P1_y \\
P1_z
\end{bmatrix}
\]

where,

\[
S_{21} = |\hat{s}|(\hat{i} \cdot \hat{s}) - o_x (S_{22} - 1) - o_y (S_{23} - 1) - o_z (S_{24} - 1)
\]

\[
S_{22} = \cos(\sigma) + \text{versin}(\sigma) (\hat{i} \cdot \hat{s})^2
\]

\[
S_{23} = -\sin(\sigma) (\hat{k} \cdot \hat{s}) + \text{versin}(\sigma) (\hat{i} \cdot \hat{s})(\hat{j} \cdot \hat{s})
\]

\[
S_{24} = \sin(\sigma) (\hat{j} \cdot \hat{s}) + \text{versin}(\sigma) (\hat{k} \cdot \hat{s})(\hat{i} \cdot \hat{s})
\]

\[
S_{31} = |\hat{s}|(\hat{j} \cdot \hat{s}) - o_x (S_{32} - 1) - o_y (S_{33} - 1) - o_z (S_{34} - 1)
\]

\[
S_{32} = \sin(\sigma) (\hat{k} \cdot \hat{s}) + \text{versin}(\sigma) (\hat{i} \cdot \hat{s})(\hat{j} \cdot \hat{s})
\]
\[ S_{33} = \cos(\sigma) + \text{versin}(\sigma)(\hat{j} \cdot \hat{s})^2 \]

\[ S_{34} = -\sin(\sigma)(\hat{i} \cdot \hat{s}) + \text{versin}(\sigma)(\hat{j} \cdot \hat{s})(\hat{k} \cdot \hat{s}) \]

\[ S_{41} = [\hat{i} \cdot \hat{s}]o_x S_{42} - o_y S_{43} - o_z (S_{44} - 1) \]

\[ S_{42} = -\sin(\sigma)(\hat{j} \cdot \hat{s}) + \text{versin}(\sigma)(\hat{k} \cdot \hat{s})(\hat{i} \cdot \hat{s}) \]

\[ S_{43} = \sin(\sigma)(\hat{i} \cdot \hat{s}) + \text{versin}(\sigma)(\hat{j} \cdot \hat{s})(\hat{k} \cdot \hat{s}) \]

\[ S_{44} = \cos(\sigma) + \text{versin}(\sigma)(\hat{k} \cdot \hat{s})^2 \]

where \([o_x, o_y, o_z]\) is any point lying on the SDA, \(\hat{s}\) is the unit vector defining the direction of the SDA, and \(\text{versin}(\theta) = 1 - \cos(\theta)\).

To determine the flexion-extension screw matrix \([S]\) of the ulna with respect to the global CT reference frame, the points previously defined on the ulna in its 30 and 90 degree positions were required. These 6 points were used as the inputs of a MATLAB program capable of calculating an optimum SDA associated with the initial and final positions of exactly three points (appendix C). This straightforward routine follows the method presented by Beggs [11], and as such does not require any type of iterative optimization. Properties of the flexion-extension screw displacement axis calculated by MATLAB are:
\[
\hat{s} = \begin{bmatrix} 0.0878 \\ -0.9944 \\ -0.0580 \end{bmatrix}, \quad \sigma = \begin{bmatrix} 0 \\ 1672.79 \\ -386.88 \end{bmatrix}, \quad |\hat{s}| = 0.1170\text{mm}, \quad \sigma = -1.09\text{rads} = -62.17\text{degs},
\]

\[
S = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 502.8121 & 0.4710 & -0.0978 & -0.8767 \\ 51.6954 & 0.0047 & 0.9941 & 0.1084 \\ -127.0371 & -0.8821 & -0.0469 & 0.4687 \end{bmatrix}
\]

These results and their graphical equivalent given in figure 4.3.1 were consistent with what was expected. The SDA deviated 6 degrees from the vertical direction that the positioning apparatus of section 3.2 sought to align with elbow flexion-extension. Also, the calculated rotation about the SDA was only 2.17 degrees from the expected value of 60 (90 degrees – 30 degrees). Qualitatively, the axis ran approximately through the centers of the capitulum and trochlea. It was therefore concluded that the calculated screw displacement axis could be accurately used to position the upper extremity at any elbow flexion angle.

Figure 4.3.1: Anterior (left) and transverse (right) views of the distal humerus showing the flexion-extension screw displacement axis.
Elbow flexion angle was defined as the angle between the projection of the long axes of the humerus and ulna onto a plane normal to the SDA. These centerlines were calculated by geometry fitting methods described in section 3.5. While the three-dimensional angle between these axes could have instead been used, the projection method minimized small contributions attributed to carrying angle. Although the CT positioning apparatus attempted to achieve a flexion angle of 30 degrees, this angle calculated in the model was 36 degrees, as shown in figure 4.3.2.

![Figure 4.3.2: Elbow flexion in CT position, as viewed normal to the SDA.](image)

4.3.2 Forearm Pronation-Supination

Because all CT scans held the forearm fixed in approximate neutral position, pronation-supination joint parameters could not be calculated based on kinematic data. Instead, this motion was modeled as a hinge joint whose axis intersected the center of the radial head proximally and the fovea of the ulnar head distally. The proximal center point was defined using sphere fitting, while the distal point was identified by inspection.
As shown in figure 4.3.3, this axis of rotation (AOR) is oblique, inferiorly running posteromedially.

Forearm rotation was defined as the angular deviation from neutral position about the AOR, with a positive value arbitrarily representing supination. Neutral rotation was calculated as the position of the radius causing the line between ulnar and radial styloids to be parallel with the mediolateral plane of the humerus, as constructed in section 3.6. Because this angle varied slightly with elbow flexion, it was arbitrarily defined with the humerus and ulna in CT position (36 degrees of elbow flexion).

Figure 4.3.3: Anterior (top) and lateral (bottom) views of the forearm in CT position, showing the pronation-supination axis of rotation.

4.4 LIGAMENT CONSTRAINTS

After describing passive joint kinematics, the behavior of soft tissue was incorporated into the rigid body dynamics model. The effects of ligaments, select muscles, and the interosseous membrane were modeled, while other soft tissues were
assumed to have negligible contributions to biomechanical function. Attachments of these structures were defined, followed by characterization of their mechanical properties.

4.4.1 Skeletal Attachments

Skeletal attachments of these soft tissues were defined through a primarily qualitative approach. Regions of origin and insertion identified by anatomical atlases [49, 76], published studies [38, 52, 58, 69, 71, 72, 79, 90, 92, 94], and dissection (see appendix D) were correlated to the computational model through use of bony landmarks. In most cases, the effective attachments were characterized by the centroid of each region. Soft tissue lines of action were then defined as passing directly between attachments (figure 4.4.1).

Figure 4.4.1: Views of the distal radioulnar ligaments (left) and medial collateral ligaments (right).
The annular ligament could not be modeled by methods described above. Because it wraps around the radial head (figure 2.3.2) with the ulna providing both origin and insertion, a direct ligament line of action would be ineffective. Therefore, an additional solid body was developed, as shown in figure 4.4.2A. This annular ligament solid (ALS) was constructed by revolving a thickened sketched arc 225 degrees about an axis. The geometry was then added to the elbow model’s assembly and positioned by aligning the revolve axis with the centerline fit to the radial head in section 3.5. Anterior and posterior ligament segments were further divided into proximal and distal portions, resulting in 4 lines of action used to emulate the function of the annular ligament. Ulnar attachment points were identified through the approach given above, with each fiber inserting on its appropriate ALS vertex (figure 4.4.2B). A three-dimensional contact constraint, as defined in section 4.6, was then applied between the annular ligament geometry and radius. By making the internal surface of the ALS slightly larger than the radial head, the ligament model allowed the radius to supinate and pronate, but limited anteroposterior and mediolateral translation. Representing portions of the annular ligament as solid geometry also permitted radial collateral ligament insertions to be defined. Because this ligament attaches diffusely onto the annular ligament, 3 RCL fibers were defined which attached to the lateral aspect of the ALS at anterior, central, and posterior positions.
Figure 4.4.2: Superomedial view of annular ligament solid (A) and posterolateral view of the radial collateral ligament fibers (black), lateral ulnar collateral ligament (pink), and posterior fibers of the annular ligament (green) (B).

Detailed published literature regarding the antebrachial interosseous membrane [69, 92] allowed its origin and insertion to be defined in a more precise manner. Because of its substantial size, this structure was modeled as 4 discrete lines of action oriented along the membrane’s fiber directions. While these attachment points were confined to lie on the interosseous borders in three-dimensional space, accurate descriptions were provided by defining their positions along the ulna’s long axis (figure 4.4.3). The central band was represented as 2 fibers located at the region’s proximal and distal margins. More distally, the cumulative effects of accessory bands were modeled as proximal and distal fibers. The proximal fiber was constrained to form a 21 degree angle with the ulna, as this was the average IOM fiber orientation given in the literature [92]. The angle of the distal accessory band fiber was relaxed in order to account for longitudinal
attachment dimensions. This resulted in an orientation more consistent with anatomical atlases [49, 76].

Figure 4.4.3: Medial view of the interosseous membrane fibers.

4.4.2 Mechanical Properties

The mechanical properties of ligament fibers were determined from the literature and estimated when required (table 4.4.1). The stiffness of individual model fibers was calculated by dividing the experimentally measured ligament stiffness by the number of fibers used to represent the structure. Stiffnesses of the MCL, RCL, and LUCL were taken directly from a study carried out by Regan et al [86]. Because the mechanical properties of the annular ligament have not been investigated, this structure’s stiffness was estimated as the mean of the adjacent medial and lateral ligament complexes. Stiffnesses of the interosseous membrane’s central band were taken from Stabile et al [93]. These values were consistent with the findings of Pfaeffle et al [81]. Although mechanical properties of the accessory band region have not been directly studied, its
stiffness was estimated as 29% of the central band based on work done by Hotchkiss et al [55]. Dorsal and palmar distal radioulnar ligament stiffnesses were taken from results presented by Schuind et al [90].

Table 4.4.1: Ligament fiber properties.

<table>
<thead>
<tr>
<th>Fiber</th>
<th>Description</th>
<th>Stiffness (N/mm)</th>
<th>In Situ Strain (%)</th>
<th>Stress-Free Length (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALAD</td>
<td>Annular, Anterior, Distal</td>
<td>28.5</td>
<td>2.0</td>
<td>8.5</td>
</tr>
<tr>
<td>ALAP</td>
<td>Annular, Anterior, Proximal</td>
<td>28.5</td>
<td>2.0</td>
<td>4.6</td>
</tr>
<tr>
<td>ALPD</td>
<td>Annular, Posterior, Distal</td>
<td>28.5</td>
<td>2.0</td>
<td>5.7</td>
</tr>
<tr>
<td>ALPP</td>
<td>Annular, Posterior, Proximal</td>
<td>28.5</td>
<td>2.0</td>
<td>9.8</td>
</tr>
<tr>
<td>DRULD</td>
<td>Distal Radioulnar, Dorsal</td>
<td>13.2</td>
<td>2.0*</td>
<td>9.3</td>
</tr>
<tr>
<td>DRULP</td>
<td>Distal Radioulnar, Palmar</td>
<td>11.0</td>
<td>2.0*</td>
<td>12.7</td>
</tr>
<tr>
<td>IOMABD</td>
<td>Interosseous Mem, Accessory, Distal</td>
<td>18.9</td>
<td>0.6*</td>
<td>32.2</td>
</tr>
<tr>
<td>IOMABP</td>
<td>Interosseous Mem, Accessory, Proximal</td>
<td>18.9</td>
<td>0.6*</td>
<td>43.4</td>
</tr>
<tr>
<td>IOMCBD</td>
<td>Interosseous Mem, Central, Distal</td>
<td>65.0</td>
<td>0.5*</td>
<td>30.0</td>
</tr>
<tr>
<td>IOMCBP</td>
<td>Interosseous Mem, Central, Proximal</td>
<td>65.0</td>
<td>0.8*</td>
<td>21.6</td>
</tr>
<tr>
<td>LUCL</td>
<td>Lateral Ulnar Collateral</td>
<td>57.0</td>
<td>n/a</td>
<td>40.3</td>
</tr>
<tr>
<td>MCLA</td>
<td>Medial Collateral, Anterior</td>
<td>72.3</td>
<td>n/a</td>
<td>17.2</td>
</tr>
<tr>
<td>MCLP</td>
<td>Medial Collateral, Posterior</td>
<td>52.2</td>
<td>n/a</td>
<td>13.2</td>
</tr>
<tr>
<td>RCLA</td>
<td>Radial Collateral, Anterior</td>
<td>15.5</td>
<td>n/a</td>
<td>24.2</td>
</tr>
<tr>
<td>RCLC</td>
<td>Radial Collateral, Central</td>
<td>15.5</td>
<td>n/a</td>
<td>16.7</td>
</tr>
<tr>
<td>RCLP</td>
<td>Radial Collateral, Posterior</td>
<td>15.5</td>
<td>n/a</td>
<td>29.7</td>
</tr>
</tbody>
</table>

*value in neutral forearm rotation

For several ligaments, their stress-free fiber lengths were calculated based on experimental in situ strain measurements taken with the upper extremity in a specific position. Through use of idealized motion defined in section 4.3, the computational model was repositioned accordingly and fiber lengths were measured. By assuming the
model to be in a passive state, these model fiber lengths $L_p$ and experimental strain values $\varepsilon$ were used to determine the ligament fiber’s stress-free length $L_0$:

$$L_0 = \frac{L_p}{1 + \varepsilon}$$  

For the interosseous membrane’s central band, proximal and distal *in situ* strains experimentally measured by Gabriel et al [39] in neutral forearm rotation were used. Because values have not been established at the accessory band, *in situ* strains in these fibers were estimated to be that of the mean central band. Consistent with Liacouras [65, 66], ligaments with no available *in situ* strain data were designated to have 2 percent strain [102]. Such structures include the annular ligament and distal radioulnar ligaments.

Stress-free fiber lengths of the MCL, RCL, and LUCL were determined without experimental *in situ* strain values. By moving elbow specimens through their range of motion, Regan et al [86] was able to quantify the flexion angles at which ligament tension develops. For each fiber, the computational model was repositioned at this transition angle, and the distance between fiber attachments was defined as the stress-free length.

### 4.4.3 Model Integration

Each skeletal attachment defined in section 4.4.1 was created as a sketch point in the SolidWorks part file of the associated bone. As a result, attachment points remained fixed to the bone as translations and rotations occurred. In COSMOSMotion, a distinct action-reaction force was defined for each ligament fiber through selection of appropriate sketch points. An expression representing the ligament fiber’s force was then entered,
based on properties given in section 4.4.2. The equation was that of a tension-only
dampened spring, as given in equation 10. The equivalent COSMOSMotion expression
required to represent equation 10 is given in equation 11.

\[
F = \begin{cases} 
  0, & D_M - L_0 < 0 \quad \text{(compression)} \\
  0, & D_M - L_0 = 0 \quad \text{(stress-free length)} \\
  -K(D_M - L_0)^n - C(V_R), & D_M - L_0 > 0 \quad \text{(tension)} 
\end{cases} \quad (10)
\]

\[
F = \text{IF}(D_M(M_1, M_2) - L_0 : 0, 0, -K*(D_M(M_1, M_2) - L_0)^n - C*(V_R(M_1, M_2))) \quad (11)
\]

where,

\[
F = \text{action-reaction force (positive = compression)}
\]

\[
M_i = \text{attachment point i}
\]

\[
D_M = \text{distance between attachments}
\]

\[
L_0 = \text{stress-free fiber length}
\]

\[
K = \text{fiber stiffness}
\]

\[
n = \text{spring order}
\]

\[
C = \text{damping coefficient}
\]

\[
V_R = \frac{dD_M}{dt} = \text{relative velocity between attachments}
\]

Because literature defining elbow ligament properties is limited, several
assumptions were made. While ligaments functioning in their physiologic range possess
a nonlinear load-elongation relationship [37, 102], ligament fibers were modeled as linear
(in equations 10 and 11, n=1). In addition, fiber damping coefficients were all set to 0.1
N-s/mm. This value was not intended to accurately capture realistic ligament
viscoelasticity. Instead, damping was incorporated as a means of bringing the dynamic
model to a steady-state position faster. Because results were only captured after the model had equilibrated, damping parameter did not influence results.

4.5 MUSCLE CONSTRAINTS

Identical to the methods of section 4.4.1, muscle origins and insertions were defined by consulting the literature [7, 49, 76], identifying attachment regions, and representing each region by one or more discrete attachment points. While biceps brachii and brachialis insertions were each described by a single point, 3 attachment points were used to characterize the triceps brachii (figure 4.5.1). The origins of individual muscle heads were also identified, but never utilized in this thesis. The complex attachment of the bicipital aponeurosis was not modeled. In addition, this computational model was not able to represent the way in which muscles occasionally wrap around bones and/or adjacent soft tissue.
No effort was made to characterize the biomechanical behavior of muscles. For the scope of this research, it was sufficient to represent these structures as force vectors of constant magnitude.

4.6 RIGID BODY CONTACT

While the initial position of each bone was dictated by joints calculated in section 4.3, these constraints were removed prior to model simulation. As a result, rigid bodies were allowed to move in all 6 degrees of freedom as permitted by bony and ligamentous constraints. To model the interaction between bones at articular surfaces, 3D contact sets were created in COSMOSMotion. One set was defined between the humerus, radius, and ulna, while another set included contact between the radius and annular ligament solid (see section 4.4.1).

COSMOSMotion was able to enforce three-dimensional contact constraints through use of interference detection and penalty regularization [1]. At each time step, the software determined if interference existed between bodies of the same contact set. For this calculation, COSMOSMotion created a faceted representation of each body. In all simulations, the accuracy used when calculating this simplified geometry was set to the highest possible value. If an interference was detected, the centroid of the interfering volume was used as the point of contact, and the outward normal direction was calculated. The contact constraint was then enforced by applying equal and opposite normal forces $F_n$ to the appropriate bodies. The magnitude of this force was computed as a function of the amount of penetration $g$ and penetration velocity:
\[ F_n = kg^e + f \left( \frac{dg}{dt}, c_{\text{max}}, d_{\text{max}} \right) \]  

(12)

with model constants given in table 4.6.1 and the damping function defined by [2].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometry Accuracy</td>
<td>100%</td>
</tr>
<tr>
<td>Contact Type</td>
<td>Impact</td>
</tr>
<tr>
<td>Stiffness (k)</td>
<td>8,000 N/mm</td>
</tr>
<tr>
<td>Exponent (e)</td>
<td>2</td>
</tr>
<tr>
<td>Maximum Damping (c_{\text{max}})</td>
<td>400 N-s/mm</td>
</tr>
<tr>
<td>Penetration at Max Damping (d_{\text{max}})</td>
<td>0.001 mm</td>
</tr>
<tr>
<td>Friction</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 4.6.1: Contact parameters

By moving the upper extremity bones about their computed joints, small interferences occasionally resulted. To prevent initial contact in the COSMOSMotion simulation, these bones were manually repositioned.
5. ELBOW RANGE OF MOTION STUDY

5.1 OVERVIEW

Model validation was performed in part by evaluating the computational model’s range of motion (ROM). By imposing a moment representing the action of the distal upper extremity under gravity, passive elbow flexion and extension limits were determined. This predicted range of motion was then compared to that found in the literature. In addition, the ligamentous and bony structures responsible for these motion constraints were acquired from the musculoskeletal model and analyzed.

5.2 METHODS

Anthropometric data collected by Veeger et al [98] allowed for calculation of the average torque induced at the elbow as a result of the forearm and hand under gravity. By idealizing these structures as a point mass of 1.55kg located 117.8mm distal to the lateral epicondyle and assuming the flexion-extension axis of rotation to pass through the lateral epicondyle, its was concluded that the horizontally-aligned forearm exerts a 1.79N-m moment on the elbow. In the computational model, this moment was oriented in the mediolateral plane of the humerus and was applied to the ulna. By defining the moment with respect to the global coordinate system, its direction remained constant as the ulna rotated in three-dimensional space.
The simulation protocol positioned the forearm in neutral pronation-supination, approximately 20 degrees from its expected flexion-extension motion limit. A cubic polynomial was used to ramp the moment magnitude from 0 to 1.79N-m. This ramp was applied over 5 seconds, followed by a 10 second period allowing the system to stabilize. Repositioning the forearm and reversing the moment’s direction allowed the opposite motion limit to be assessed.

5.3 RESULTS

After applying the flexion-extension moment, all bodies came to rest within 5 seconds. The musculoskeletal model’s final positions are depicted in figures 5.3.1 and 5.3.2. No dislocations were observed at the ulnohumeral, radiohumeral, proximal radioulnar, or distal radioulnar articulations. However, slight separation occurred between the inferior trochlea notch and trochlea in full extension. A gap of approximately 6.9mm was present between the radial head and capitulum in full extension, but diminished to 0.3mm in flexion. In both simulations, contact occurred between the humerus and ulna, and ulna and radius. In extension, contacts were present between the olecranon fossa and medial and lateral margins of the olecranon. In flexion, several points of contact existed between the coronoid process and coronoid fossa.

![Figure 5.3.1: Medial view of model after application of extension moment.](image-url)
The angle of flexion was measured as the three-dimensional angle between the long axes of the humerus and ulna. The computational model predicted an extension limit of -10.1 degrees (hyperextension) and a flexion limit of 150.4 degrees (table 5.3.1, figures 5.3.1 and 5.3.2). Flexion and extension limits and overall range of motion were within one standard deviation of the passive results obtained by Gunal et al [51] for the right (dominant) arm. Gunal et al’s maximum extension and ROM for the left arm were also within one standard deviation of the model’s results. Values obtained by Boone et al [14] through active elbow flexion showed a smaller range of motion than that determined by the model. Although standard deviations were not provided, the range of values for flexion-extension limits provided by the American Academy of Orthopaedic Surgeons [5] contained the values predicted by the computational model.
Table 5.3.1: Predicted and published elbow flexion-extension range of motion.

<table>
<thead>
<tr>
<th>Source</th>
<th>Extension Limit*</th>
<th>Flexion Limit*</th>
<th>Range of Motion*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computational Model</td>
<td>-10.1</td>
<td>150.4</td>
<td>160.5</td>
</tr>
<tr>
<td>Gunal et al (right) [51]</td>
<td>-3.8 (11.9)</td>
<td>142.8 (8.4)</td>
<td>146.6 (14.6)</td>
</tr>
<tr>
<td>Gunal et al (left) [51]</td>
<td>-6.0 (10.2)</td>
<td>145.6 (3.1)</td>
<td>151.6 (10.7)</td>
</tr>
<tr>
<td>Boone et al [14]</td>
<td>0.6 (3.1)</td>
<td>142.9 (5.6)</td>
<td>142.3 (6.4)</td>
</tr>
<tr>
<td>AAOS [5]</td>
<td>-15 to 0</td>
<td>150</td>
<td>150 to 165</td>
</tr>
</tbody>
</table>

*values given in the form of mean (SD).

During both simulations, the forearm was observed to move from neutral rotation to pronation. At maximum flexion, the forearm became pronated by 35.6 degrees. At maximum extension, a pronation angle of 40.5 degrees was observed.

As the elbow reached its steady-state positions, significant forces were observed in ligamentous structures and at points of bony contact. The magnitudes of these forces at flexion and extension limits are given in figure 5.3.3. For each simulation, contact between the humerus and ulna (BONE) produced the largest reaction forces. High force values were also observed in the anterior and posterior bundles of the medial collateral ligament (MCLA, MCLP). While the force in the anterior fiber was relatively consistent between flexion extremes, a dramatic increase of 202N was seen in the posterior fiber with flexion. Forces developed in the central fiber of the radial collateral ligament (RCLC) were less than those of the medial ligament complex, but still significant. Forces in the annular ligament (AL) were similar in magnitude to the RCLC, but showed a decrease with flexion as compared to the increase observed in the RCLC. The lateral ulnar collateral ligament (LUCL) and anterior and posterior segments of the radial
collateral ligament (RCLA, RCLP) were not taught in either simulation. Forces in the distal radioulnar ligaments (DRUL) and interosseous membrane (IOM) were small and consistent between flexion and extension limits.

![Diagram showing forces in different structures at flexion and extension limits](image)

**Figure 5.3.3:** Structures contributing to joint stability at flexion and extension limits.

### 5.4 DISCUSSION

Elbow range of motion has been described with varying results by several authors [6, 14, 23, 31, 50, 51, 71, 105]. An important difference among the literature relates to the method used to evaluate range of motion, which was not always stated. Active ROM assessment is particularly straightforward; the subject uses their muscles to flex and
extend the joint as far as possible. Passive ROM is evaluated by the investigator applying a “slight overpressure” and sensing the “end feel” at motion limits [23]. The computational model emulated the passive approach by applying a “pressure” (i.e. moment) representing the gravitational effects of the distal upper extremity. This was considered to be the best approximation of the highly qualitative clinical description of passive ROM evaluation.

The musculoskeletal model predicted flexion-extension motion limits consistent with passive values reported in a 1,000-subject investigation by Gunal et al [51]. For both maximum flexion and maximum extension, the model’s results were exceptionally close to the means of the angle ranges provided by the American Academy of Orthopaedic Surgeons [5]. However, the ROM computed by the model was greater than that observed by Boone et al [14]. This difference was expected, due to the fact that the experimental study investigated active flexion-extension as compared to passive. These two methods have been observed to result in different elbow ROM values, with a smaller range of motion for active flexion-extension [23, 51].

While the model’s results correlated well with previous findings, slightly increased laxity was observed at both elbow motion extremes. This behavior may be attributed to several factors. First, all subjects used in these studies were male, while the bony geometry of the computational model was that of a female. Increased hyperextension has been observed in women, due to the reduced size of their olecranon [23]. In addition, the passive moment applied to the ulna was calculated from anthropometric data based on male specimens. The appropriate forearm-hand mass for
the model is therefore expected to be lower, with a center of mass located closer to the elbow axis of rotation. As a result, the elbow joint of the model would normally experience a lower moment magnitude during passive ROM evaluation, thus producing a smaller flexion-extension range of motion.

Contributions to elbow stability may have also been provided, to a lesser extent, by structures not represented in the model. While there is disagreement among researchers [71, 77], the elbow joint capsule is generally thought to have small but observable effects on elbow kinematics. In addition, passive ROM studies may have been affected by tension in muscles crossing the elbow joint. In this study, muscle forces were not modeled.

Furthermore, in situ strain values for the elbow’s lateral and medial ligament complexes are not currently available. Consequently, these structures’ stress-free lengths were estimated based on a single study’s qualitative observation of ligament tautness, as described in section 4.4.2. This could have potentially caused over- or underestimation of ligament in situ strain. The latter would have resulted in additional elbow laxity.

Interestingly, the forearm of the computational model was observed to pronate as the upper extremity equilibrated, although no external loading was applied directly to the radius. This behavior is attributed to preloads in the annular ligament pressing the radial head against the lesser sigmoid notch. The resulting forearm rotation, coupled with ligament force data, indicated the efficacy of the annular ligament modeling technique.

The annular ligament solid, as described in section 4.4.1, was developed to fulfill two needs. First, this structure, in combination with the 4 annular ligament fibers, was
required to limit radial head translation while allowing radial head rotation. Secondly, the solid body was to provide insertion sites for the radial collateral ligament, thereby offering a pathway for forces to be transmitted between the lateral epicondyle and radius/lateral ulna. Because free rotation of the radius was observed in combination with forces in the RCL and annular ligament, this lateral ligament representation was believed to be an accurate reflection of the actual complex.

Bone and ligament contributions to elbow range of motion limits were consistent with published findings. The anterior bundle of the medial collateral ligament and contact between the olecranon and olecranon fossa has been observed as limiting elbow extension [71]. As shown in figure 5.3.3, these structures showed the largest forces when compared to all other extension values. In addition, contact between the coronoid process and coronoid fossa is considered to be the significant flexion constraint. The computational model predicted ulnohumeral contact to be the largest force in flexion, due in part to the multiple contact points observed between the coronoid process and coronoid fossa.

Assuming ligament cross-sectional areas and ultimate strengths provided by Regan et al [86], the stresses observed in all ligament fibers were well within their allowable limits, except for the posterior bundle of the medial collateral ligament. At its flexion limit, stresses were calculated to be higher than the structure’s ultimate strength. Inspection of the model at maximum flexion showed the ulna to be translated medially and only in contact with the medial facet of the trochlea. While the ulnohumeral joint attempted to reduce MCLP force by displacing mediolaterally, the high level of bony
congruity only permitted finite translations. This behavior is shown quantitatively in figure 5.3.3, as bone contact forces became exceptionally high with elbow flexion. As explained above, MCL \textit{in situ} strain values have not been reported in the literature, and therefore had to be approximated. In this instance, the stress-free length of the medial collateral ligament’s posterior bundle appears to have been underestimated.

While the results of this study are promising, several differences exist between the model and published investigations. Most significantly, the literature reported findings on a group of subjects, whereas the simulated model is representative of a single individual. With respect to soft tissues, cartilage was not modeled and ligaments were assumed to behave in a linear fashion. In addition, bony geometry was approximated with surface-fitting techniques.

This study demonstrated the computational model’s capacity to accurately predict elbow range of motion. Because these results were dependent upon articular contact, ligamentous constraints, and externally applied loading, realistic articular behavior was realized without relying on experimental ROM data or making assumptions about joint axes of rotation. In contrast, other upper extremity mathematical models limit the achievable body segment positions by hard-coding ROM data into the model definition [63].

Furthermore, the computational model allowed important biomechanical parameters to be easily and accurately obtained. While articular contact forces are exceedingly difficult to measure in cadaver studies, mathematical modeling provided an acceptable alternative. Similarly, ligament length and stress values were accurately
collected with the click of a button. Joint flexion angles and other kinematic data were also readily available.
6. VARUS ELBOW STABILITY STUDY

6.1 OVERVIEW

Validity of the developed computational elbow model was further assessed by simulating an experimental study and comparing results. More specifically, the model emulated an investigation carried out by Hull et al entitled, “Role of the coronoid process in varus osteoarticular stability of the elbow” [56]. By applying a varus displacement to the distal ulna and measuring constraining load, this experiment was able to quantify coronoid process contributions to elbow stability. The use of this study as a validation tool was particularly appropriate, as no other computational elbow model has had the capacity to predict such parameters.

In summary of the experimental study, eleven fresh human cadaveric upper extremities were tested with a biaxial servohydraulic materials testing machine. All soft tissue was excised with the exception of the distal radioulnar ligament, annular ligament, interosseous membrane, and biceps, triceps, and brachialis muscle insertions. The humerus was secured in place with its shaft oriented horizontally and medial epicondyle vertically. Muscle actions were simulated by applying tension to monofilament lines attached to the muscle insertions. The Instron actuator was coupled to the distal ulna through a spherical rod end bearing which allowed the forearm to freely rotate in internal-external directions and translate proximally and distally (figure 6.2.3). At elbow flexion angles of 30, 45, 60, 90, and 120 degrees, a 1cm varus displacement was applied to the
distal ulna, and the actuator’s restraining force was measured. The coronoid process was then incrementally resected and retested for cut levels of 25%, 33%, 40%, 50%, 75%, and 100%.

6.2 METHODS

Small modifications and additions to the computational model formulated in chapter 1 were necessary to reproduce the experimental testing protocol. This included changes in bone geometry, removal of particular ligaments, characterization of muscle forces, and creation of test devices.

6.2.1 Skeletal Geometry

Geometry of the ulna developed in chapter 1 was modified in order to represent incremental resection of the coronoid process. To model this feature, the orientation of the coronoid process cut plane was first described. This was accomplished by fitting a plane to a set of points positioned along the central ridge of the trochlear notch, using methods described in section 3.5. The cut plane was then defined as being perpendicular to the fit plane and parallel to the ulna’s proximal long axis. The depth of this cut plane was then varied appropriately to achieve different levels of coronoid process resection, as shown in figure 6.2.1. While this approach was believed to be the best representation of the experimental procedure used by Hull et al, a very similar method has also been applied by Doornberg et al [29].
Figure 6.2.1: Anteromedial view of the coronoid process shown intact (left), 50% resected (center), 100% resected (right).

6.2.2 Soft Tissue

Because the cadaver study resected the medical collateral ligament, radial collateral ligament, and lateral ulnar collateral ligament, the corresponding ligament fibers were disabled in the computational model. Effective muscle origins were defined in a manner consistent with the experimental approach, using the medial epicondyle and the long axis of the humerus as references. Muscles were then represented as force vectors directed from insertion to origin (figure 6.2.2). Assuming the study’s pulleys to be frictionless, constant magnitude force vectors of 19.6N, 19.6N, and 39.2N were created for the biceps, brachialis, and triceps, respectively. In the triceps, this force was distributed evenly among the three attachments.
Figure 6.2.2: Lateral view of soft tissue fibers (defined in section 4.4.1), muscle force vectors, and point of varus displacement (magenta circle).

6.2.3 Testing Apparatus and Procedure

To mimic the test device, an additional solid body and two mechanical joints were constructed. As shown in figure 6.2.3, the materials testing machine was modeled as a cylindrical solid body with its lateral end tapering to a point. A translational joint oriented along the cylinder’s centerline was created, thereby preventing all motion except displacement along the centerline. As described below, this degree of freedom was controlled in the testing procedure. By medially offsetting the ulna’s long axis, the shoulder bolt’s central axis was modeled. This centerline was fixed to the ulna, but also constrained to be in contact with the endpoint of the computational model’s varus displacement apparatus. The second condition was enforced through use of an in-line
joint primitive, which constrained the point to lie along the centerline while allowing free rotation.

Figure 6.2.3: Experimental (left) and model (right) test apparatus, showing allowable (green arrows) and prescribed (red arrow) directions of motion.

The cadaver study’s experimental procedure was reproduced by controlling the position of the computational model’s varus displacement apparatus with respect to the fixed humerus (figure 6.2.4). Several parameters were necessary to determine its initial position. The central axis was aligned vertically and defined to be 19.5cm from the medial epicondyle. The position of the varus displacement apparatus’ lateral endpoint was dictated by the in-line joint described above, which was attached to the ulna. The initial position of the ulna was dictated by the flexion-extension screw displacement axis (SDA) defined in section 4.3.1. This last constraint was removed prior to running simulations. For the first 6 seconds, the varus displacement apparatus remained in its
initial position, thereby allowing the musculoskeletal model to stabilize. Over the next second, the solid body of the apparatus was smoothly displaced 1cm along its translational joint in a varus direction. Lastly, the varus displacement apparatus remained fixed in its final position for an additional 6 seconds, again allowing the system to equilibrate. This simulation sequence was repeated for the same combinations of elbow flexion and coronoid process resection level as was done in the cadaver study, thus yielding 35 unique simulations.

![Figure 6.2.4: View of varus study computational model in 90 degrees of flexion.](image)

6.2.4 Data Acquisition

Through use of COSMOSMotion’s Excel export function, various force values were obtained for each simulation time step. These included ligament fiber reaction forces, forces resulting from bony contact, and the reaction force required to maintain the
prescribed position of the varus displacement apparatus. Steady-state forces were calculated by taking a 0.5s mean value. In addition, the COSMOSMotion timer API developed in section 1.1 was used to monitor simulation solve time.

6.3 RESULTS

While the axial load magnitude measured at the varus displacement apparatus varied with flexion angle and coronoid process resection level, a characteristic plot of force versus time is given in figure 6.3.1. In all simulations, initial force values were particularly large and oscillatory. After several seconds, the model was able to equilibrate and arrive at a steady actuator force magnitude. At 6 seconds into the simulation, the apparatus began to apply the varus displacement. The effect of this was apparent in the actuator’s reaction force, as its magnitude increased considerably. The force continued to rise until simulation time reached approximately 7 seconds, which correlated with the time varus displacement ceased. In many cases, the force would take several additional seconds to stabilize, sometimes reducing slightly before reaching a steady-state value. To quantitatively determine if the system had reached equilibrium, a line was fit to each set of data points used to calculate mean force values (at 5.5s–6.0s and again at 12.5s–13s). The absolute value of the slope of this line was consistently less than 0.05N/s.
Figure 6.3.1: Characteristic axial load measured at the translational joint of the varus displacement apparatus.

For each flexion angle, the elbow’s percent constraining load was calculated as the force measured at the actuator after varus displacement divided by that measured with the coronoid process intact. Therefore, at 0% resection, the constraining load was 100% by definition. Results obtained from the cadaver study and computational model are given in figures 6.3.2 through 6.3.6. The horizontal axis shows the percent coronoid process resection, while the vertical axis shows percent constraining load. Error bars indicate the standard deviations of the cadaver study.
Figure 6.3.2: Constraining load vs coronoid process resection at 30 degrees flexion.

Figure 6.3.3: Constraining load vs coronoid process resection at 45 degrees flexion.
Figure 6.3.4: Constraining load vs coronoid process resection at 60 degrees flexion.

Figure 6.3.5: Constraining load vs coronoid process resection at 90 degrees flexion.
Figure 6.3.6: Constraining load vs coronoid process resection at 120 degrees flexion.

The constraining load observed in the experimental investigation declined from 100% for the intact coronoid process to approximately 22% for the completely resected coronoid process. This general trend of declining constraining load with increasing resection was predicted by the computational model at all flexion angles except 120 degrees. At this flexion angle, the model’s constraining load was significantly less than that of the experimental investigation for coronoid process cut levels of 25%, 33%, 40%, and 50%. At higher resection, the constraining load increased, with values well within one standard deviation of the experimental findings. At 30 and 45 degrees of flexion and 25% and 33% resection, the constraining load predicted by the model was slightly higher than 100%. This same phenomenon was observed in the cadaver study at the same resection levels, but for all flexion angles with the exception of 120 degrees.
Several important observations were made at resection levels above 33%. At flexion angles less than 90 degrees, the computational model predicted a noticeable decline in constraining load at approximately 40% resection. While the experimental results also showed the constraining load to decrease at this resection level, the amount of reduction was less pronounced. At all flexion angles except 120 degrees, the model predicted the constraining load at 50% resection exceptionally close to the results of the cadaver study. At flexion angles less than 90 degrees, a dramatic drop in constraining load was predicted by the computational model at 75% coronoid process cut level. The signs of these constraining loads were negative, indicating that a compressive load was being applied to the varus displacement apparatus. For these flexion angles, the constraining load became more negative as the coronoid was entirely resected. At 90 degrees of elbow flexion, the model predicted the relationship between percent constraining load and resection level extremely well. The model’s results were within one standard deviation of the experimental results for all resection levels except 100%.

As indicated above, percent constraining load was calculated with the force measured at the actuator after varus displacement. These loads, averaged over all flexion angles, are given in figure 6.3.7. Percent coronoid process resection is shown on the horizontal axis while constraining load in newtons is given on the vertical axis. This plot reveals that the computational model predicted a very similar trend in constraining load, but significantly offset from experimental results. On average, the model’s load for a given resection level was 5.03N less than the load measured by the cadaver study. The standard deviation associated with this value was 0.49N, which indicated an appreciable
similarity between the shapes of the model and experimental curves. However, small differences existed between the two curves. At low resection levels, the cadaver study recorded a slight increase in constraining load, while the model predicted a slight decrease. At 50% resection, the experimental results showed a noticeable drop in load, while the model’s constraining load remained approximately constant. At 100% resection, the average load of the computational model dropped below zero, while the experimental results remained positive.

![Constraining Load vs Coronoid Process Resection](image)

**Figure 6.3.7:** Constraining load (in newtons) vs coronoid process resection, averaged over all flexion angles.

In the cadaver study, several elbow dislocations were observed after applying tendon loads but before varus displacement. This behavior was seen at 50%, 75%, and
100% resection for 30 degrees of flexion, and at 75% and 100% resection for 45 degrees. In the computational model, degree of dislocation was evaluated by more quantitative means. The amount of ulnar proximal translation, as measured along the ulna’s long axis, before varus displacement but after muscle loading is shown in figure 6.3.8. Individual curves are given for each elbow flexion angle, and ulnar translation is plotted as a function of coronoid process resection. Consistent with experimental observations, the most amount of translation occurred at 30 and 45 degrees of flexion at high levels of coronoid process resection. At 90 and 120 degrees of flexion, negligible changes in proximal translation occurred with increased levels of coronoid process resection.

Figure 6.3.8: Proximal ulnar translation before varus displacement vs coronoid process resection level.
The annular ligament’s stabilizing force (calculated as the sum of the 4 ligament fibers) was also measured with the computational model. Similar to ulnar translation, these values were recorded after muscle force application but before varus displacement. An equivalent observation was not made in the experimental investigation. The results are given in figure 6.3.9. Again, individual curves are given for each elbow flexion angle, and stabilizing force is plotted as a function of coronoid process resection. At low levels of coronoid resection, the stabilizing force provided by the annular ligament increased linearly with flexion angle (note that the change in angle is not consistent between curves). Compared to intact values, the largest increase in load occurred in 30 degrees of elbow flexion at 75% and 100% coronoid process resection. To a lesser extent, annular ligament stabilizing load was observed to increase in 45 degrees of flexion at 75% and 100% resection.
Contact force between the radial head and capitulum was calculated for the computational model, as shown in figure 6.3.10. This parameter was again evaluated after muscle force application but before varus displacement. Elbow flexion angle is shown on the horizontal axis while the magnitude of the contact force is given on the vertical axis. Contact between the humerus and radius did not exist at coronoid process resection levels below 75%. This contact force was the largest at 30 degrees of flexion and steadily declined as the elbow was flexed. At 120 degrees of flexion, contact between the radial head and capitulum did not occur. At most flexion angles, contact forces were greater for 100% resection than for 75%.

Figure 6.3.9: Stabilizing force of the annular ligament to the radial head vs coronoid process resection level.
Time required to solve each rigid body dynamics simulation was recorded using the COSMOSMotion timer developed in section 1.1. These values were relatively independent of elbow flexion angle, but varied considerably with coronoid process resection. Table 6.3.1 gives these results, averaged over flexion angle. The average time required to solve simulations at 100% coronoid resection was dramatically larger than that of other resection levels.
Table 6.3.1: Time required to solve model, averaged over all flexion angles.

<table>
<thead>
<tr>
<th>Resection Level</th>
<th>Simulation Solve Time (mins)</th>
<th>Simulation Frames</th>
<th>Average Frame Solve Time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>64.03</td>
<td>2069</td>
<td>1.85</td>
</tr>
<tr>
<td>25%</td>
<td>37.41</td>
<td>2076</td>
<td>1.08</td>
</tr>
<tr>
<td>33%</td>
<td>46.58</td>
<td>2082</td>
<td>1.34</td>
</tr>
<tr>
<td>40%</td>
<td>41.64</td>
<td>2076</td>
<td>1.20</td>
</tr>
<tr>
<td>50%</td>
<td>33.52</td>
<td>2095</td>
<td>0.96</td>
</tr>
<tr>
<td>75%</td>
<td>45.18</td>
<td>2156</td>
<td>1.24</td>
</tr>
<tr>
<td>100%</td>
<td>1352.77</td>
<td>2108</td>
<td>38.59</td>
</tr>
</tbody>
</table>

Values given in the table represent the entire 20s simulation.

6.4 DISCUSSION

In this study, the computational model was able to accurately predict the relationship between varus constraining load and coronoid process resection based on a cadaver study carried out by Hull et al [56]. A characteristic drop in percent constraining load was observed above 50% resection, consistent with findings of the experimental investigation. In addition, the cadaver study observed that at 75% and 100% resection, the percent constraining load increased as the elbow became flexed. This same trend was predicted by the musculoskeletal model (figures 6.3.2 through 6.3.6).

As shown in figure 6.3.7, the shapes of the constraining load versus coronoid resection curves were similar between the model and cadaver study. However, the model data were offset by approximately 5N. This observation may be attributed to several differences between approaches. In the experimental study, each cadaver was positioned with its medial aspect facing upward. As a result, gravity was pulling the forearm down during mechanical testing, thus increasing the load measured at the actuator. In the
computational model, the effects of gravity were not represented. To illustrate the potential effect gravity may have on model results, an additional simulation was carried out at 90 degrees of elbow flexion with the coronoid process intact. Forces approximating the effects of gravity on the ulna and radius, based on data from Veeger et al [98], were added to the model. Constraining force measured at the varus displacement apparatus after varus displacement increased from 4.41N without gravity to 7.53N with gravity.

Furthermore, the model’s passive position for each flexion angle was determined by the screw displacement axis defined in section 4.3.1. Any inaccuracies of this motion could have caused the ulna to have greater valgus angulation than is normally observed. Because the initial position of the varus displacement apparatus was defined with respect to the ulna, valgus angulation would have caused the actuator’s position to be less varus than expected. With this, the screw displacement axis was determined from segmented CT data. Close inspection of the CT positioning apparatus shown in figure 3.2.1 reveals that the thickness of the plywood, in combination with the constraints of the cable ties, may have caused the specimen to be in slight valgus alignment.

In addition, articular cartilage was not represented in the musculoskeletal model. While this is generally thought to have negligible effects on joint kinematics, this study’s testing protocol displaced the ulna a mere 3 degrees. Because of the small magnitude of this perturbation, the slightly increased joint laxity caused by the absence of cartilage may have resulted in perceivable reductions in varus constraining load.
At low elbow flexion angles, significant proximal ulnar translation was observed at 75% and 100% coronoid process resection (figure 6.3.8). These results correlated exceptionally well with findings of Closkey et al[24]. Through use of cadaveric upper extremities, these investigators applied an axial load to ulna and observed its proximal translation as a function of coronoid process resection. They concluded that a significant amount of ulnar displacement occurred only after more than 50% of the coronoid process was removed. The axial load prescribed in this cadaver investigation was not directly represented in the computational model. However, the approximate equivalent loading condition resulted from the action of the model’s muscle forces. At low flexion angles, the muscle force vectors were more closely aligned with the ulna’s long axis. At higher flexion angles, the force of the muscles became perpendicular to the long axis of the ulna. This explains why proximal ulnar translation did not vary with coronoid process resection at higher flexion angles.

By calculating proximal ulnar displacement, annular ligament constraining force, and radiohumeral contact force, a more complete description of the elbow’s behavior under testing conditions was obtained. The computational model was able to accurately measure these parameters, while the experimental investigation could not. As a result, ulnohumeral dislocations could be assessed more precisely than was possible with visual inspection used in the cadaver study of Hull et al [56]. At 30 and 45 degrees of flexion with 75% and 100% coronoid process resection, proximal ulnar translations were large, annular ligament forces were high, and significant contact forces existed between the radial head and capitulum. From these results, it is clear that the muscle forces of the
triceps brachii and brachialis were attempting to translate the ulna proximally, and the ulna was able to freely do so in the absence of the coronoid process. However, the annular ligament coupled the ulna to the radius. As these two forearm bones began to migrate proximally, the radial head came in contact with the capitulum of the humerus and prevented further radial translation, as shown in figure 6.4.1. The ulna was then limited in its proximal translation by forces developed in the annular ligament. These model findings correlated well with the observations of the cadaver study, and gave evidence to support the role of the radial head as a secondary stabilizer of the elbow [71].

![Figure 6.4.1: Lateral view of the elbow in 30 degrees of flexion and 100% coronoid process resection after application of muscle forces. Proximal ulnar translation was limited by the annular ligament (posterior fibers shown in black). Annular ligament translation was limited by radiohumeral contact.](image)
In this study, predictions provided by the musculoskeletal computational model matched well with previously reported behavior. The efficiency of this modeling technique was also shown, as the numerical integrator was often able to solve the rigid body dynamics model at a rate greater than 0.6 frames per second. Furthermore, no other computational model has been able to predict elbow varus-valgus laxities. Because previous models incorporated idealized axes of rotation, attempting to displace the ulna in a varus or valgus direction while holding the humerus fixed would have resulted in an infinite actuator reaction force, and no such displacement would have been achieved.
7. CONCLUSION

The objective of this thesis was to develop and validate a computational model of the elbow, capable of predicting joint behavior based on three-dimensional bony geometry, ligamentous constraints, and muscle loading. Mimics software was successfully applied in order to generate highly accurate three-dimensional representations of upper extremity bones based on CT data. SolidWorks was implemented to modify these bones and incorporate into the model additional solid bodies. COSMOSMotion allowed the effects of ligaments and muscles, contact between bony features, and external loading to be represented. COSMOSMotion, in conjunction with the underlying ADAMS/Solver numerical integrator, was able to efficiently predict the behavior of the model based on rigid body dynamics.

The computational model was validated by simulating past experimental investigations and comparing results. Passive flexion-extension range of motion predicted by the model correlated exceptionally well with reported values. Bony and ligamentous structures responsible for enforcing motion limits also agreed with past observations. The model’s varus stability as a function of elbow flexion and coronoid process resection was also investigated. The trends predicted by the model matched those of the associated cadaver study.

While these results are promising, further model refinements may improve its accuracy and expand its applicability. Firstly, additional cadaver investigations are
recommended in order to determine the mechanical properties of ligaments at the elbow and forearm. Most importantly, these studies should evaluate *in situ* strains and structural stiffnesses. This information is currently not available for all elbow ligaments, and reported values assume linearity. Increased model accuracy may be possible by representing the true nonlinear load-elongation relationship consistently observed in ligamentous structures [37]. Furthermore, ligaments and muscles were represented as forces acting directly between origin and insertion. This simplification was not believed to significantly influence the studies presented in this thesis, but required the lateral ligament complex to be defined through use of the annular ligament solid. However, this assumption may not be valid in other circumstances. It is therefore recommended to model the way in which ligaments and muscles wrap around bony geometry.

While the steps taken to reproduce the bones of the upper extremity yielded highly accurate geometry, the method demanded a significant amount of manual editing by means of advanced surfacing techniques. To reduce the time necessary to generate the model’s skeletal geometry, it is recommended to incorporate an autonomous or semiautonomous segmentation procedure. In this thesis, the final model was developed by combining the segmented geometry resulting from 2 CT scans. It is highly recommended to base future skeletal geometry off a single, complete, high-resolution CT scan. Furthermore, particular joint parameters were calculated by acquiring several CT scans in varying specimen positions. While this approach is still advocated, complete bony segmentation of each scan can be avoided through use of methods proposed by Fischer et al [34].
In this thesis, results were only obtained after the model had reached a static position. However, the method used to calculate these positions and parameters was that of rigid body dynamics. Therefore, this model may be adapted to study true dynamic upper extremity behavior. This would first require each segment’s mass properties to be accurately defined. In addition, viscoelastic behavior of ligaments can readily be modeled in COSMOSMotion, as well as complex muscle mechanics, such as the popular Hill equation [37].

The musculoskeletal computational model developed in this thesis has been shown to be an accurate representation of the true elbow joint. Furthermore, it is the first forward dynamics elbow model of its kind, as all others have constrained the articulation to have particular degrees of freedom and ignored the effects of ligaments. The model has also demonstrated its powerful ability to calculate parameters that have proven difficult to capture experimentally, such as forces within ligaments, contact forces at joints, and complex three-dimensional bone kinematics. In addition, the software packages used in this thesis are commercially-available and well accepted in the engineering community. The potential medical applications for this model and modeling approach are staggering, and are anticipated to ultimately have value as a clinical tool for patient-specific preoperative planning.
APPENDIX A: MATLAB PROGRAM FOR GEOMETRY FITTING

function GEOFIT2(excel_file_name,geometry_type,initial_guess)

% TITLE..............GEOFIT2
% DESCRIPTION.....Program to fit geometry to data points
% AUTHOR.........Justin Fisk
% DATE..........07/10/2007 Revision 2
% 11/27/2006 Revision 1
% 08/20/2006 Original Release
%
% -------------------SYNTAX-------------------------
% GEOFIT2('excel_file_name',geometry_type,initial_guess)
% %
% excel_file_name: '<data_file_name>.xls'
% %
% geometry_type:
% -1 (SPHERE)
% -2 (3D LINE)
% -3 (CYLINDER)
% -4 (PLANE)
% %
% initial_guess: a vector giving an initial guess:
% -1 (SPHERE):
% initial_guess = [Xc,Yc,Zc,R]
% -2 (3D LINE):
% initial_guess = [P0x,P0y,Vx,Vy,Vz]
% NOTES: -P0x and P0y are defined when P0z = 0 (point on X-Y plane).
% -The magnitude of V does not matter.
% -3 (CYLINDER):
% initial_guess = [P0x,P0y,Vx,Vy,Vz,R]
% NOTES: -P0x and P0y are defined when P0z = 0 (point on X-Y plane).
% -The magnitude of V does not matter.
% -4 (PLANE):
% initial_guess = [A,B,C,Doff]
% NOTE: The magnitude of V does not matter.
%
% -------------------FOR SPHERE FITTING---------------------
% Pc = [Xc,Yc,Zc] = sphere center point
% R = sphere radius
%
% Pp = [Ppx,Ppy,Ppz] = data point

% PARAMETRIC EQUATIONS OF A 3D LINE:
% Ptx = P0x + Vx*t

101
% Pty = P0y + Vy*t
% Ptz = P0z + Vz*t
%
% Pt = (Ptx,Pty,Ptz) = any point lying on the line, defined by t
% P0 = (P0x,P0y,P0z) = a point lying on the line
% V = [Vx,Vy,Vz] = a vector defining the direction of the line
% t = a parameter to vary to get different points (Pt) on the line
%
% V2 = [Vx/Vz,Vy/Vz,1] = scaled direction vector (same direction as V)
% U = [Ux,Uy,Uz] = vector V normalized
% D = vector from Pt to Pp (perpendicular to line)
% Pp = [Ppx,Ppy,Ppz] = data point
%
% ------------------FOR CYLINDER FITTING------------------
% A cylinder is defined by a 3D line (representing its centerline), and a
% constant perpendicular distance from the 3D line to the cylinder's
% surface
%
% SEE 3D LINE FITTING ABOVE
% R = cylinder radius
%
% ------------------FOR PLANE FITTING------------------
% PLANE EQUATION: Ax + By + Cz + D = 0
%
% V = [A,B,C] = a vector defining the direction normal to the plane
% Pt = (x,y,z) = a point which lies on the plane
% D = -Doff*mag(V)
%
% V2 = [A/C,B/C,1] = scaled direction vector (same direction as V)
% U = vector V normalized (same direction as V)
% Doff = the distance from the origin to the plane, along V
% Pp = [Ppx,Ppy,Ppz] = data point
%
% -------------------OTHER IMPORTANT VARIABLES------------------
% distance = distance between data point and geometry
% error = sum of the distances squared (parameter to minimize)
%
% ------------------------OPTIONS------------------------
% Convergence Criteria.......................lines 159-163
% Data Point Display.........................lines 226-229
% Plot Window Size...........................lines 362-363
% Plot Rotation..............................line 376

global DATA dim_DATA; % global variables

message = warndlg('This program cannot fit an axis that is parallel to
the X-Y Plane. This program cannot fit a plane that contains the Z-
axis','PROGRAM LIMITATIONS');
waitfor(message);

if nargin == 0 % if no input arguments were specified
message = errordlg('Exactly three inputs are required (Excel file name, geometry type, and initial guess). Exiting routine.', 'INPUT ARGUMENT ERROR');
    waitfor(message);
    % This space reserved for future GUI.
    return;
end

if ~(any(nargin == [0,3])) % if an invalid number of input arguments were specified
    message = errordlg('Exactly three inputs are required (Excel file name, geometry type, and initial guess). Exiting routine.', 'INPUT ARGUMENT ERROR');
    waitfor(message);
    return;
end

% READ IN EXCEL DATA -----------------------------------------------

file_marker = fopen(excel_file_name); % -1 if file cannot be opened
if file_marker ~= -1
    fclose(file_marker); % closes file if opened successfully
end
file_extension_marker = size(intersect(excel_file_name,'.xls')); % [1,4] if file name contains '.xls'
if (file_marker ~= -1) && all(file_extension_marker == [1,4]) % if file is valid
    DATA = xlsread(excel_file_name); % reads in Excel data
else
    message = errordlg('Invalid Excel file. Exiting routine', 'FILE ERROR');
    waitfor(message);
    return;
end

% CHECK INPUTS FOR DISCREPANCIES -----------------------------------

dim_DATA = size(DATA);
if dim_DATA(2) ~= 3
    message = errordlg('Incorrect number of columns in Excel data. Exiting routine.', 'INPUT ERROR');
    waitfor(message);
    return;
end

dim_initial_guess = size(initial_guess);
if dim_initial_guess(1) ~= 1 % if the initial guess is not a vector
    message = errordlg('Incorrect dimensions of initial guess. Exiting routine.', 'INPUT ERROR');
    waitfor(message);
    return;
end
if ~(any(geometry_type == [1,2,3,4])) % if geometry_type is not 1, 2, 3, or 4.
    message = errordlg('Specified geometry type is invalid. Exiting routine.','INPUT ERROR');
    waitfor(message);
    return;
end

if geometry_type == 1 % sphere
    initial_guess_check = 4;
elseif geometry_type == 2 % 3dline
    initial_guess_check = 5;
elseif geometry_type == 3 % cylinder
    initial_guess_check = 6;
else % plane
    initial_guess_check = 4;
end

if dim_initial_guess(2) ~= initial_guess_check
    message = errordlg('Incorrect dimensions of initial guess. Exiting routine.','INPUT ERROR');
    waitfor(message);
    return;
end

% CALCULATE GEOMETRY --------------------------------------------------
--
fnc_display = 'iter'; % display option during solve routine
fnc_eval_lim = 50000; % max number of function evaluations allowed
inter_lim = 50000; % max number of iterations allowed
fnc_tol = 1e-8; % convergence criteria for function value
x_tol = 1e-8; % convergence criteria for x value

fmin_options = optimset('Display',fnc_display,...
          'MaxFunEvals',fnc_eval_lim,...
          'MaxIter',inter_lim,...
          'TolFun',fnc_tol,...
          'TolX',x_tol);

if geometry_type == 1 % sphere
    [RESULTS,fval,exitflag] = fminsearch(@SPHERE_ERROR, ...
        initial_guess, fmin_options);
    Pc = RESULTS(1:3);
    R = RESULTS(4);
elseif geometry_type == 2 % 3dline
    initial_guess_corrected = [initial_guess(1:2),
        initial_guess(3:4)/initial_guess(5)];
    [RESULTS,fval,exitflag] = fminsearch(@THREEDLINE_ERROR, ...
        initial_guess_corrected, fmin_options);
\[ P_0 = [\text{RESULTS}(1:2), 0]; \]
\[ V_2 = [\text{RESULTS}(3:4), 1]; \]
\[ U = V_2 / \text{norm}(V_2); \]
\[ \text{elseif geometry_type == 3 } % \text{for a cylinder} \]
\[ \text{initial_guess_corrected = [initial_guess}(1:2), \]
\[ \text{initial_guess}(3:4)/\text{initial_guess}(5), \text{initial_guess}(6)]; \]
\[ [\text{RESULTS}, fval, \text{exitflag}] = \text{fminsearch}(\text{@CYLINDER_ERROR}, \ldots \]
\[ \text{initial_guess_corrected, fmin_options}); \]
\[ P_0 = [\text{RESULTS}(1:2), 0]; \]
\[ V_2 = [\text{RESULTS}(3:4), 1]; \]
\[ U = V_2 / \text{norm}(V_2); \]
\[ R = \text{RESULTS}(5); \]
\[ \text{else } % \text{for a plane} \]
\[ \text{initial_guess_corrected = [initial_guess}(1:2)/\text{initial_guess}(3), \]
\[ \text{initial_guess}(4)]; \]
\[ [\text{RESULTS}, fval, \text{exitflag}] = \text{fminsearch}(\text{@PLANE_ERROR}, \ldots \]
\[ \text{initial_guess_corrected, fmin_options}); \]
\[ V_2 = [\text{RESULTS}(1:2), 1]; \]
\[ U = V_2 / \text{norm}(V_2); \]
\[ Doff = \text{RESULTS}(3); \]
\[ \text{end} \]

% REPORT RESULTS -------------------------------------------------------------
--
\[ \text{if geometry_type == 1 } % \text{sphere} \]
\[ \quad P_c \]
\[ \quad R \]
\[ \text{elseif geometry_type == 2 } % \text{3dline} \]
\[ \quad P_0 \]
\[ \quad U \]
\[ \text{elseif geometry_type == 3 } % \text{for a cylinder} \]
\[ \quad P_0 \]
\[ \quad U \]
\[ \quad R \]
\[ \text{else } % \text{for a plane} \]
\[ \quad U \]
\[ \quad Doff \]
\[ \text{end} \]

fval
exitflag

% PLOT DATA POINTS -----------------------------------------------------------
--
\[ \text{if geometry_type == 1 data_pt_disp = 'k'; } % \text{sphere point display} \]
\[ \text{elseif geometry_type == 2 data_pt_disp = '+k'; } % \text{3D line point display} \]
\[ \text{elseif geometry_type == 3 data_pt_disp = '.k'; } % \text{cylinder point display} \]
\[ \text{else data_pt_disp = '+k'; } % \text{plane point display} \]
% PLOT REGRESSION GEOMETRY ---------------------------------------------

if geometry_type == 1 % sphere
    % produce unit sphere coordinates
    sphere_x = sphere_x*R; % change sphere radius
    sphere_y = sphere_y*R; % change sphere radius
    sphere_z = sphere_z*R; % change sphere radius
    sphere_x = sphere_x + Pc(1); % move sphere's center
    sphere_y = sphere_y + Pc(2); % move sphere's center
    sphere_z = sphere_z + Pc(3); % move sphere's center

    mesh(sphere_x,sphere_y,sphere_z, ... 
        'EdgeColor','b', ... 
        'FaceAlpha',0.5); % plots sphere

elseif geometry_type == 2 % 3dline
    % goes through all data points
    for lp1 = 1:dim_DATA(1)
        Pp_test = DATA(lp1,1:3);
        t_test = dot((Pp_test-P0),U); % the projection of data point
        if t_test > t_max % finds the maximum t value in the data set
            t_max = t_test;
        end
        if t_test < t_min % finds the minimum t value in the data set
            t_min = t_test;
        end
    end

    Pt_extremes(1,:) = P0 + U*(t_min); % Pt at the minimum t value
    Pt_extremes(2,:) = P0 + U*(t_max); % Pt at the maximum t value

    plot3(Pt_extremes(:,1),Pt_extremes(:,2),Pt_extremes(:,3),'-.b', ... 
        'LineWidth',3); % plots 3D line

elseif geometry_type == 3 % for a cylinder
    % goes through all data points
    for lp1 = 1:dim_DATA(1)
        Pp_test = DATA(lp1,1:3);
        t_test = dot((Pp_test-P0),U); % the projection of data point
        if t_test > t_max % finds the maximum t value in the data set
            t_max = t_test;
        end
        if t_test < t_min % finds the minimum t value in the data set
            t_min = t_test;
        end
    end

    Pt_extremes(1,:) = P0 + U*(t_min); % Pt at the minimum t value
    Pt_extremes(2,:) = P0 + U*(t_max); % Pt at the maximum t value

    plot3(Pt_extremes(:,1),Pt_extremes(:,2),Pt_extremes(:,3),'-.b', ... 
        'LineWidth',3); % plots 3D line
end
if t_test < t_min % finds the minimum t value in the data set
t_min = t_test;
end
end
Pt_extremes(1,:) = P0 + U*(t_min); % Pt at the minimum t value
Pt_extremes(2,:) = P0 + U*(t_max); % Pt at the maximum t value
plot3(Pt_extremes(:,1),Pt_extremes(:,2),Pt_extremes(:,3),'-.b', ...
'LineWidth',3); % plots centerline of cylinder

[cyl_x,cyl_y,cyl_z] = cylinder; % produces unit cylinder coordinates
cyl_x = cyl_x*R; % change cylinder radius in X direction
cyl_y = cyl_y*R; % change cylinder radius in Y direction
cyl_ht = norm(Pt_extremes(1,:) - Pt_extremes(2,:)); % required cylinder height
cyl_z = cyl_z*cyl_ht; % change cylinder height
Uk_new = U; % new frame's k unit vector (cylinder centerline)
rand_pt = DATA(1,1:3); % random point to define Uj_new
rand_pt_proj = dot(rand_pt,Uk_new); % the projection of random point onto Uk_new
rand_pt_vctr = rand_pt - Uk_new*rand_pt_proj; % vector to random point, perpendicular to Uk_new
Uj_new = rand_pt_vctr/norm(rand_pt_vctr); % new frame's j unit vector
Ui_new = cross(Uj_new,Uk_new); % new frame's i unit vector (unit vector cross unit vector with 90 degs results in a unit vector)
trans_mat = [Ui_new(1,:);Uj_new(1,:);Uk_new(1,:)]; % coordinate transformation matrix
inv_trans_mat = trans_mat'; % allows for rotation of the CYLINDER and not the COORDINATE SYSTEM
[cyl_size_1,cyl_size_2] = size(cyl_x); % the dimensions of the cylinder coordinate array
for lp1 = 1:cyl_size_1
   for lp2 = 1:cyl_size_2
      pnt_in_U_new = inv_trans_mat*([cyl_x(lp1,lp2),cyl_y(lp1,lp2),cyl_z(lp1,lp2)]'); % apply transformation to cylinder
      cyl_x(lp1,lp2) = pnt_in_U_new(1);
cyl_y(lp1,lp2) = pnt_in_U_new(2);
cyl_z(lp1,lp2) = pnt_in_U_new(3);
   end
end
cyl_x = cyl_x + Pt_extremes(1,1); % move cylinder centerline to Pt @ t_min
cyl_y = cyl_y + Pt_extremes(1,2); % move cylinder centerline to Pt @ t_min
cyl_z = cyl_z + Pt_extremes(1,3); % move cylinder centerline to Pt @ t_min

mesh(cyl_x,cyl_y,cyl_z, ...
    'EdgeColor','b', ...
    'EdgeAlpha',.6, ...
    'FaceAlpha',0.5); % plots cylinder
else % for a plane
    Pp_sum = 0;
    for lp1 = 1:dim_DATA(1) % goes through all data points
        Pp_sum = Pp_sum + DATA(lp1,1:3);
    end
    Pp_ave = Pp_sum/dim_DATA(1); % coordinates of the average point
    Pp_dist_max = 0;
    for lp1 = 1:dim_DATA(1) % goes through all data points
        Pp_dist_test = norm(Pp_ave - DATA(lp1,1:3)); % direct distance between Pp_ave and data point
        if Pp_dist_test > Pp_dist_max % finds max Pp_dist_test in data set
            Pp_dist_max = Pp_dist_test;
        end
    end

    Pp_ave_norm_dist = dot(U,Pp_ave) - Doff; % the normal distance between Pp_ave and fit plane (~0)
    Pp_ave_proj = Pp_ave - U*Pp_ave_norm_dist; % projection of Pp_ave onto plane (~Pp_ave)
    in_pln_dir1 = cross(U,[1,0,0]); % in-plane direction 1: vector U cross X-axis (on YZ-plane)
    in_pln_udir1 = in_pln_dir1/norm(in_pln_dir1); % in-plane direction 1 unit vector
    in_pln_dir2 = cross(U,in_pln_dir1); % vector U cross in-plane direction 1
    in_pln_udir2 = in_pln_dir2/norm(in_pln_dir2); % in-plane direction 2 unit vector

    pln_pts(1,:) = Pp_ave_proj + in_pln_udir1*Pp_dist_max + in_pln_udir2*Pp_dist_max;
    pln_pts(2,:) = Pp_ave_proj + in_pln_udir1*Pp_dist_max - in_pln_udir2*Pp_dist_max;
    pln_pts(3,:) = Pp_ave_proj - in_pln_udir1*Pp_dist_max - in_pln_udir2*Pp_dist_max;
    pln_pts(4,:) = Pp_ave_proj - in_pln_udir1*Pp_dist_max + in_pln_udir2*Pp_dist_max;

    pln_surf_x = [pln_pts(1,1),pln_pts(2,1);pln_pts(4,1),pln_pts(3,1)];
    pln_surf_y = [pln_pts(1,2),pln_pts(2,2);pln_pts(4,2),pln_pts(3,2)];
    pln_surf_z = (Doff - pln_surf_x.*U(1) - pln_surf_y.*U(2))/U(3); % calcs point on plane, given x,y coords
mesh(pln_surf_x,pln_surf_y,pln_surf_z, ...
    'EdgeColor','b', ...
    'EdgeAlpha',.6, ...
    'FaceAlpha',0.5); % plots plane
end

% ADJUST PLOT -------------------------------------------------------------
hor_scale = 2/3; % horizontal window scale factor
vert_scale = 2/3; % vertical window scale factor
display_props = get(0,'ScreenSize');
set(gcf,'Position',display_props(3)*((1-hor_scale)/2), ...
    display_props(4)*((1-vert_scale)/2), ...
    display_props(3)*hor_scale, ...
    display_props(4)*vert_scale)); % centers & scales plot window

axis auto; % resizes plot limits in order to fit all geometry
axis equal; % sets aspect ratio to be the same in every direction (x,y,z)
axis vis3d; % freezes aspect ratio & disables "stretch-to-fill"
axis off; % removes axis lines, marks, & labels

% ROTATE PLOT --------------------------------------------------------------
for loop1=1:360*n_rot
    pause(0.005);
    camorbit(1,0,'camera',[0 1 0]);
    drawnow;
end

% CLEAN UP -----------------------------------------------------------------
clear DATA dim_DATA;

% SPHERE_ERROR SUBFUNCTION --------------------------------------------------
function error = SPHERE_ERROR(input_vars)
    global DATA dim_DATA; % global variables
    Pc = input_vars(1:3);
    R = input_vars(4);
    error = 0;
    for loop1 = 1:dim_DATA(1) % goes through all Excel data points
        Pp = DATA(loop1,1:3);
        distance = sqrt((Pp(1) - Pc(1))^2 + (Pp(2) - Pc(2))^2 + (Pp(3) - Pc(3))^2);
        error = error + (distance - R)^2; % sum of the distances squared between sphere surface and data points
% 3DLINE_ERROR SUBFUNCTION --------------------------------------------
function error = THREEDLINE_ERROR(input_vars)

global DATA dim_DATA; % global variables

P0 = [input_vars(1:2), 0];
V2 = [input_vars(3:4), 1];
U = V2/norm(V2);
error = 0;

for loop1 = 1:dim_DATA(1) % goes through all Excel data points
    Pp = DATA(loop1,1:3);
    t = dot((Pp-P0),U); % the projection of data point Pp onto line
    D = (Pp-P0) - U*t; % vector from Pt to Pp (perpendicular to line)
    distance = norm(D); % distance from Pt to Pp
    error = error + distance^2; % sum of the distances squared, between line & data points
end

% CYLINDER_ERROR SUBFUNCTION ------------------------------------------
function error = CYLINDER_ERROR(input_vars)

global DATA dim_DATA; % global variables

P0 = [input_vars(1:2), 0];
V2 = [input_vars(3:4), 1];
R = [input_vars(5)];
U = V2/norm(V2);
error = 0;

for loop1 = 1:dim_DATA(1) % goes through all Excel data points
    Pp = DATA(loop1,1:3);
    t = dot((Pp-P0),U); % the projection of data point Pp onto line
    D = (Pp-P0) - U*t; % vector from Pt to Pp (perpendicular to line)
    distance = norm(D) - R; % distance between Pp and cylinder surface
    error = error + distance^2; % sum of the distances squared, between cylinder surface & data points
end

% PLANE_ERROR SUBFUNCTION ---------------------------------------------
function error = PLANE_ERROR(input_vars)

global DATA dim_DATA; % global variables
V2 = [input_vars(1:2), 1];
U = V2/norm(V2);
Doff = input_vars(3);
error = 0;

for loop1 = 1:dim_DATA(1) % goes through all Excel data points
    Pp = DATA(loop1,1:3);
    distance = dot(U,Pp) - Doff; % the perpendicular distance between
    Pp and plane
    error = error + distance^2; % sum of the distances squared, between
    plane & data points
end
APPENDIX B: COSMOSMOTION TIMER API

Option Explicit

Private Sub FrameBox_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Dim ch_val As String
    ch_val = Chr$(KeyAscii)
    If Not (ch_val >= "0" And ch_val <= "9") Then
        KeyAscii = 0
        MsgBox ("Please enter a number")
    End If
End Sub

Private Sub SimTimeBox_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Dim ch_val As String
    ch_val = Chr$(KeyAscii)
    If Not (ch_val >= "0" And ch_val <= "9") Then
        KeyAscii = 0
        MsgBox ("Please enter a number")
    End If
End Sub

Private Sub Reset_Button_Click()
    SimTimeBox.Locked = False
    SimTimeBox.Value = 20
    FrameBox.Locked = False
    FrameBox.Value = 2000
    SimNotesBox.Locked = False
    SimNotesBox.Value = "Optional Notes"
    StatusLabel.Caption = "Not Started"
    SolveTimeLabel.Caption = "-
End Sub

Private Sub Start_Button_Click()
    'SolidWorks variables
    Dim swApp As SldWorks.SldWorks
    Dim swModel As SldWorks.ModelDoc2
    'COSMOSMotion variables
    Dim CMAddin As CMOTIONSWAPILib.CMotionSWAddin
    Dim CMAssy As CMOTIONSWAPILib.DDMAssembly
    Dim CMMech As CMOTIONSWAPILib.DDMMechanism
    Dim CMSim As CMOTIONSWAPILib-DDMSimulation
    'Other variables
    Dim YesNoResponse As Integer
    Dim StartTime, EndTime, DeltaTime As Single

    Set swApp = Application.SldWorks

Set swModel = swApp.ActiveDoc
Set CMAddin = GetObject(, "CMOTIONSWAPI.CMOTIONSWADDIN")
Set CMAssy = CMAddin.ActiveAssembly
Set CMMech = CMAssy.Mechanism

If CMMech.Simulated Then 'if simulation exists
    YesNoResponse = MsgBox("Delete existing simulation?", vbYesNo, "CMExecute")
    If YesNoResponse = 6 Then '6="yes"
        CMMech.DeleteSimulation
    Else
        MsgBox ("Exiting application.")
    End
Else
    MsgBox ("Exiting application.")
End If

StatusLabel.Caption = "Solving"
SolveTimeLabel.Caption = "-"
SimTimeBox.Locked = True
FrameBox.Locked = True
SimNotesBox.Locked = True

StartTime = Timer
Call CMMech.Simulate(SimTimeBox.Value, FrameBox.Value)
EndTime = Timer
StatusLabel.Caption = "Completed"
If EndTime > StartTime Then
    DeltaTime = EndTime - StartTime
Else 'if simulation starts before midnight & ends after midnight
    DeltaTime = 86400 - StartTime + EndTime
End If
SolveTimeLabel.Caption = DeltaTime / 60
Beep
End Sub

Private Sub Exit_Button_Click()
    End
End Sub
function SDAFIT1(P1i,P2i,P3i,P1f,P2f,P3f)

% TITLE..........SDAFIT1
% DESCRIPTION....Program to fit screw displacement axis to data points
% AUTHOR.........Justin Fisk
% DATE...........07/01/2007 Revision 1
% 09/05/2006 Original Release
%
% ------------------------SYNTAX------------------------
% SDAFIT1(P1i,P2i,P3i,P1f,P2f,P3f)
%
% Pji = [Pjix,Pjiy,Pjiz] = initial coordinates of point j
% Pjf = [Pjfx,Pjfy,Pj fz] = final coordinates of point j
% j must equal 3
%
% ------------------------OTHER IMPORTANT VARIABLES------------------------
% S = translation along SDA
% S_hat = [Cxs,Cys,Czs] = unit vector represent the direction of the SDA
% sigma = angle of rotation around SDA (in radians)
% SDA = rotation displacement axis transformation matrix
% O = [Ox,Oy,Oz] = a point lying on the SDA (defined at Ox = 0)
%
% CALCULATE S & S_HAT ---------------------------------------------
% delt1 = P1f - P1i;
% delt2 = P2f - P2i;
% delt3 = P3f - P3i;
% A = [delt1; delt2; delt3];
% B = inv(A)^*[1;1;1];
% S = sqrt(1/(B(1)^2 + B(2)^2 + B(3)^2));
% S_hat = S*B;
%
% CALCULATE SIGMA, AND ADJUST SIGN IF NECESSARY -------------------
% Vi = P2i - P1i;
% Vf = P2f - P1f;
% crossi = cross(Vi,S_hat);
% crossf = cross(Vf,S_hat);
% sigma = acos( dot((crossi/norm(crossi)), (crossf/norm(crossf))));
% check = det([Vi;Vf;transpose(S_hat)]);
% if sign(check) == -1

114
\begin{verbatim}
sigma = -sigma;
end

% CALCULATE SDA ------------------------------------
vs_sigma = 1 - cos(sigma);
SDA(1,:) = [1;0;0;0];
SDA(2,2) = cos(sigma) + vs_sigma*S_hat(1)^2;
SDA(2,3) = -sin(sigma)*S_hat(3) + vs_sigma*S_hat(1)*S_hat(2);
SDA(2,4) = sin(sigma)*S_hat(2) + vs_sigma*S_hat(3)*S_hat(1);
SDA(3,2) = sin(sigma)*S_hat(3) + vs_sigma*S_hat(1)*S_hat(2);
SDA(3,3) = cos(sigma) + vs_sigma*S_hat(2)^2;
SDA(3,4) = -sin(sigma)*S_hat(1) + vs_sigma*S_hat(2)*S_hat(3);
SDA(4,2) = -sin(sigma)*S_hat(3) + vs_sigma*S_hat(1)*S_hat(2);
SDA(4,3) = sin(sigma)*S_hat(2) + vs_sigma*S_hat(3)*S_hat(1);
SDA(4,4) = cos(sigma) + vs_sigma*S_hat(3)^2;
C = [SDA(2,2:4);SDA(3,2:4);SDA(4,2:4)]; % det(C) should be 1.0
SDA(2:4,1) = transpose(P1f) - C*(transpose(P1i));

% CALCULATE O -------------------------------------
O(1) = 0;
O(2) = ( SDA(3,4)*SDA(4,1) - (SDA(4,4) - 1)*SDA(3,1) + S*(
        S_hat(2)*(SDA(4,4) - 1) - S_hat(3)*SDA(3,4)) )/(2*vs_sigma*S_hat(1)^2);
O(3) = ( SDA(4,3)*SDA(3,1) - (SDA(3,3) - 1)*SDA(4,1) + S*(
        S_hat(3)*(SDA(3,3) - 1) - S_hat(2)*SDA(4,3)) )/(2*vs_sigma*S_hat(1)^2);

% REPORT RESULTS ----------------------------------
SDA
S
S_hat
O
Sigma
\end{verbatim}
APPENDIX D: ELBOW DISSECTIONS

Abbreviations

CA Capitulum
CP Coronoid Process
H Humerus
IOM Interoosseous Membrane
LE Lateral Epicondyle
MCL Medial collateral ligament
ME Medial Epicondyle
OL Olecranon
R Radius
RCL Radial collateral ligament
RH Radial Head
TR Trochlea
U Ulna

Annular Ligament Measurements

<table>
<thead>
<tr>
<th>SPECIMEN</th>
<th>THICKNESS* (mm)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>POSTERIOR</td>
<td>ANTERIOR</td>
<td>AVERAGE</td>
</tr>
<tr>
<td>1</td>
<td>2.44</td>
<td>1.96</td>
<td>2.20</td>
</tr>
<tr>
<td>2</td>
<td>2.08</td>
<td>1.86</td>
<td>1.97</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>2.26</td>
<td>1.91</td>
<td>2.09</td>
</tr>
</tbody>
</table>

*values are approximate and were taken with calipers.
Figure D.1: Anterior view of specimen 1.

Figure D.2: Specimen 1; solid white arrow, biceps brachii tendon.
Figure D.3: Specimen 1; solid white arrows, RCL origin and insertion.

Figure D.4: Specimen 1; solid white arrows, RCL origin and insertion.
Figure D.5: Axial view of specimen 1.

Figure D.6: Specimen 1; solid white arrows, annular ligament; dashed white arrow, lesser sigmoid notch.
Figure D.7: Specimen 1; solid white arrows, annular ligament; dashed white arrow, lesser sigmoid notch; white centerlines, cut line.

Figure D.8: Specimen 1; solid white arrows, MCL origin.
Figure D.9: Specimen 2; solid white arrows, posterior bundle of MCL insertion; dashed white arrow, anterior bundle of MCL insertion.

Figure D.10: Specimen 2; solid white arrows, annular ligament.
Figure D.11: Specimen 2; solid white arrows, annular ligament; dashed white arrows, lesser sigmoid notch.

Figure D.12: Specimen 2; solid white arrows, annular ligament; dashed white arrow, lesser sigmoid notch; white centerlines, cut line.
REFERENCES


65. Liacouras, P.C., *Computational modeling to predict mechanical function of joints: validations and applications of lower leg simulations*, in Department of
biomedical engineering. 2006, Virginia Commonwealth University: Richmond, VA.


VITA

Justin Paul Fisk was born on November 2, 1982 in Limerick, Maine. In 2000, he attained Eagle Scout rank. In 2001, he graduated from Massabesic High School in Waterboro, Maine. Continuing on to college at Rensselaer Polytechnic Institute, Justin earned a Bachelor of Science degree in mechanical engineering in 2005. While working towards this degree, he also became a Certified SolidWorks Professional. In August 2005, he moved to Richmond, Virginia to pursue a Master of Science degree at Virginia Commonwealth University, while also working part time as an applications support engineer at TriMech Solutions. The content of this thesis was presented at the 2007 ASME Summer Bioengineering Conference, for which it won 2nd place in the Masters student competition.