2008

Modeling upper body kinematics while using a transradial prosthesis

Derek J. Lura
University of South Florida

Follow this and additional works at: http://scholarcommons.usf.edu/etd
Part of the American Studies Commons

Scholar Commons Citation

This Thesis is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.
Modeling Upper Body Kinematics While Using a Transradial Prosthesis

by

Derek J. Lura

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Mechanical Engineering Department of Mechanical Engineering College of Engineering University of South Florida

Major Professor: Rajiv Dubey, Ph.D.
Craig Lusk, Ph.D.
M. Jason Highsmith, D.P.T.
Stephanie Carey, Ph.D.

Date of Approval:
November 7, 2008

Keywords: Robotics, Simulation, Transradial, Inverse Kinematics, Compensatory Motion, Activities of Daily Living (ADL)

© Copyright 2008, Derek J. Lura
Acknowledgments

This project was funded in part by the U.S. Department of Education, Rehabilitation Services Administration Award# H235J050020, Project# 6140100000 “Demonstration Project on Prosthetics and Orthotics”.
# Table of Contents

List of Tables iv  
List of Figures v  
Abstract vi  

Chapter 1: Introduction 1  
1.1 Anatomy and Approximations of the Human Upper Body 2  
1.1.1 Torso 2  
1.1.2 Shoulder Complex 2  
1.1.3 Upper Arm 4  
1.1.4 Forearm 5  
1.1.5 Hand 6  
1.2 Other Models of the Human Upper Body 6  
1.3 Range of Movement and Nomenclature 8  
1.4 Transradial Prostheses 11  

Chapter 2: The Model 13  
2.1 Denavit and Hartenberg Parameters 13  
2.2 Parameters of Model Links 14  
2.3 Bilateral Configuration 15  

Chapter 3: Analysis of the Model 16  
3.1 Tasks & Trajectories 16  
3.1.1 Opening a Door 17  
3.1.2 Drinking From a Cup 17  
3.1.3 Turning a Steering Wheel 17  
3.1.4 Lifting a Box 18  
3.2 Inverse Kinematics 18  
3.2.1 Forward Kinematics 18  
3.2.2 Jacobian & Pseudo Inverse 19
A.1.3 Virtual Reality Initialization ‘VRModel.m’ 50
A.1.4 Opening a Door Trajectory Generator ‘Door.m’ 51
A.1.5 Drinking From a Cup Trajectory Generator ‘Drinking.m’ 52
A.1.6 Lifting a Box Trajectory Generator ‘Box.m’ 52
A.1.7 Turning a Steering Wheel Trajectory Generator ‘Turn.m’ 53
List of Tables

Table 1: Bending of the Spine Joint Limits 8
Table 2: Joint Limits of the Glenohumeral Range of Motion 10
Table 3: Elbow Joint Limits 10
Table 4: Wrist Joint Limits 11
Table 5: Description of Denavit and Hartenberg Parameters 13
Table 6: Model Parameters 14
Table 7: Marker Descriptions for Carey et al.[5] 26
Table 8: Average Joint Angle Difference (Degrees / Joint) 29
Table 9: Weighting Factors for Opening a Door Task 31
List of Figures

Figure 1: Division of the Human Upper Body into Specified Regions 1
Figure 2: Torso Range of Movement 8
Figure 3: Scapular Range of Movement 9
Figure 4: Glenohumeral Range of Movement 9
Figure 5: Elbow Range of Movement 10
Figure 6: Wrist Range of Movement 11
Figure 7: Terminal Devices 11
Figure 8: Diagram of a Link 14
Figure 9: Layout of the Model 15
Figure 10: Graphical User Interface 23
Figure 11: Vicon Model Layout [5] 25
Figure 12: Left Joint Angle Difference 29
Figure 13: Right Joint Angle Difference 29
Figure 14: Glenohumeral Joint Angles 30
Figure 15: Elbow Angles 31
Figure 16: Translation of the Torso in the Z (L1) and Y (L2) Directions 31
Figure 17: Screen Capture of the Turning a Wheel Task 32
Figure 18: Plot of Scapular Elevation / Depression Angles 32
Figure 19: Plot of Torso Sideways Bending 33
Figure 20: Screen Capture of the Lifting a Box Task 34
Figure 21: Plot of Forward / Backward Bending of the Torso 34
Figure 22: Plot of Scapular Protraction / Retraction Angle 35
Figure 23: Screen Capture of the Model at 70% of Task Completion 35
Modeling Upper Body Kinematics While Using a Transradial Prosthesis

Derek J. Lura

ABSTRACT

The prostheses used by the majority of persons with upper limb amputations today offer a limited range of motion. Relative to anatomical joints transradial (below the elbow) prosthesis users lose at least two of the three degrees of freedom provided by the wrist and forearm. Some myoelectric prostheses currently allow for forearm pronation and supination (rotation about an axis parallel to the forearm) and the operation of a powered prosthetic hand. Body-powered prostheses, incorporating hooks and other cable driven terminal devices, have even fewer active degrees of freedom. In order to perform activities of daily living, an amputee must use a greater than normal range of movement from other anatomical body joints to compensate for the loss of movement caused by the amputation. By studying this compensatory motion of prosthetic users, the mechanics of how they adapt to the loss of range of motion in a given limb and specific tasks were analyzed. The purpose of this study is to create a robotic based kinematic model that can predict the compensatory motion of a given task using given subject data in select tasks. The tasks used in this study are the activities of daily living: opening a door, drinking from a cup, lifting a box, and turning a steering wheel.

For the model the joint angles necessary to accomplish a task are calculated by a simulation for a set of prostheses and tasks. The simulation contains a set of configurations that are represented by parameters that consist of the joint degrees of freedom provided by each prosthesis, and a set of task information that includes joint constraints and trajectories. In the simulation the hand or prosthesis follows the trajectory to perform the task. Analysis of tasks is done by attaching prosthetic constraints to one of the arms of the upper body model in the simulation, other arm maintains an anatomical configuration. By running the model through this simulation with different configurations the compensatory motions were found. Results
can then be used to select the best prosthesis for a given user, design prostheses that are more effective at selected tasks, and demonstrate some possible compensations given a set of residual joint limitations with certain prosthetic components, by optimizing the configuration of the prostheses to improve their performance.
Chapter 1: Introduction

The complex motions of the human upper body can be approximated in several ways. The number of articulations in the upper body makes modeling, manipulating, and analyzing all of them difficult. In order to decrease the difficulty of modeling the upper body some articulations were approximated. The first approximation made is to consider the bones to be rigid bodies. The bones and their articulations form the basis of the upper body movement. Groups of related articulations were approximated into simpler motions. Figure 1 shows the upper body divided into the regions of interest which were approximated into segments. In this study the upper body is divided into five segments, the torso, shoulder, upper arm, forearm, and hand. The articulations of these segments are developed based on anatomical data, commonly measured degrees of freedom, and models developed for motion analysis. The anatomical articulations will be generalized to fit these segments while remaining as close to anatomical movements as possible. This chapter overviews the anatomy of the human upper body, the approximations made in the model, other, models of the human upper limb, the range of motion of the model articulations, and some transradial prosthetics.

Figure 1: Division of the Human Upper Body into Specified Regions
1.1 Anatomy and Approximations of the Human Upper Body

The motions of the upper body were analyzed before they were modeled. This section reviews the anatomy of the sections of the body that were considered in the model and the approximations made.

1.1.1 Torso

Movement of the torso is largely defined by lower body, the pelvis, and spine. The base of the spine is the origin of the model. The degrees of freedom of the lower body are approximated as gross movements. Because the goal of this study is to understand compensatory motion, the lower body should not be ignored, as it is a potential source of motion. However, including detail in the lower body would greatly complicate the model and therefore it was determined that the range of motion provided by the lower body would be modeled by allowing the origin of the model to translate in all three directions, allowing the model to move in space. Rotation of the pelvis and bending of the spine were considered to comprise the rotational components of the degrees of freedom of the torso.

In the human body the spinal column is comprised of three sections; the cervical, thoracic, and lumbar. The motion of the cervical spine accounts for the degrees of freedom of the head which is not to be included in the model and is ignored in the current design of the model. The thoracic and the lumbar spine segments contribute to what is considered the bending of the torso. The spine is supported by a series of ligaments that run between vertebrae along the long axis of the spine. Additionally there is an intervertebral disc between each vertebra that supports the spine and allows for articulation. Each joint between vertebrae has its own range of movement [19] and contributes to the overall motion of the torso. The lumbar segments have a greater range of movement in forward and backward bending than the thoracic region. Lateral bending is almost evenly distributed between thoracic and lumbar sections. The range of rotation is greater in the thoracic segments. The relative motions of the spine during activities of daily living are considered to be small enough that the base of the spine will be considered the center of rotation for the torso segment of the model. For large displacements this will result in some error where the motion of the spine becomes significant.

1.1.2 Shoulder Complex

The human shoulder is a very complex mechanism. It is the joint with the greatest range of movement in the human body, Gray et al. [10]. The shoulder consist of three major bones; the humerus (upper arm
bone), the clavicle (collar bone), and the scapula (shoulder blade). There are four joints in the shoulder complex. The sternoclavicular joint attaches the clavicle to the sternum. The acromioclavicular joint connects the clavicle to the acromion (a protrusion on the topmost region of the scapula). The scapulothoracic joint holds the scapula in place with respect to the thorax. And the glenohumeral joint, a ball and socket joint between the scapula and the humerus. In the model the shoulder complex is considered to be dependent on the scapula and the clavicle, while the upper arm segment is dependent on the glenohumeral joint.

The sternoclavicular joint is a double arthrodial joint, also called a double plane joint. The joint has limited range of motion in three directions. The clavicle is held in place on the sternal side primarily by the anterior and posterior sternal ligaments, and the costoclavicular ligament. The motion of the clavicle carries the scapula, therefore motion of the glenohumeral joint is closely centered about the articulation point of the sternoclavicular joint. The articular disk distributes loads and absorbs impacts, the synovial membranes provide lubrication to the clavicle to allow it to move respective to the sternum.

The acromioclavicular joint is an arthrodial or plane joint between the acromial end of the clavicle and the medial margin of the acromion of the scapula. The joint is able to slide and rotate in the plane parallel to the joint. This joint allows the clavicle and the scapula to move together without moving the clavicle away from the sternum. The articular capsule of the acromioclavicular joint is similar to that of the sternoclavicular joint, it surrounds the joint, and is strengthened above and below by the superior and inferior acromioclavicular ligaments. The superior acromioclavicular ligament is a quadrilateral band; it extends between the upper side of the end of the clavicle and the upper end of the acromion. The coracoclavicular ligament connects the clavicle with the coracoid process of the scapula and maintains contact in the articulation; it is composed of two fasciculi called the trapezioid and conoid ligaments.

The scapulothoracic joint holds the scapula to the thoracic region of the torso. The scapula is supported by the surrounding muscles and can move in a large number of directions. The majority of the movement of the scapula is along the surface of the rib cage, however small amounts of rotation out of the plane of the rib cage can also be accomplished.

The inclusion of the shoulder complex in the model increases the range of movement of the arm by magnifying the positioning options of the glenoid fossa. The movement of the glenohumeral joint with
respect to the torso is approximated by the shoulder complex within the model. This can also be described as the joint angles of the sternoclavicular joint along the clavicle and the angle of the acromioclavicular joint along the distance from the joint to the glenohumeral joint. Because the distance along the clavicle is nearly constant, the change in the acromioclavicular joint angle, and the distance between the acromioclavicular and glenohumeral joint centers is small, in the model it is therefore assumed that the distance between the sternoclavicular and the glenohumeral joints is constant. The motion of the shoulder complex is described by rotations about a projected center of the sternoclavicular joint. The segment which represents the shoulder complex in the model is referred to as the clavicle segment. The length of the clavicle segment is the approximated distance between the projected joint center and the glenohumeral joint center.

1.1.3 Upper Arm

The upper arm is in some respects the simplest segment of the upper body. It is composed of one bone, the humerus, and has three rotational degrees of freedom provided by the glenohumeral joint. Often referred to as the shoulder joint the glenohumeral joint is a ball and socket joint. It is able to rotate in almost any direction. The head of the humerus is larger than the socket or glenoid fossa of the scapula, which gives the articulation a large range of motion but requires the ligaments and tendons to stabilize the humerus.

The articular capsule covers the entire joint and is attached above the glenoid fossa and at the anatomical neck of the humerus. It is thicker at the top and bottom and in the middle, and is loose enough to not interfere with motion in any direction. It is strengthened by the supraspinatus muscle above, by the long head of the triceps brachii below, by the tendons of the infraspinatus and teres minor behind; and in front by the tendon of the subscapularis. The coracohumeral ligament is a broad band which strengthens the superior aspect of the articular capsule. It is attached to the lateral border of the coracoid process, and passes obliquely downward and laterally to the anterior aspect of the greater tubercle of the humerus, blending with the tendon of the supraspinatus muscle. The glenohumeral ligaments are three bands that help to strengthen the articular capsule. One passes from the medial edge of the glenoid cavity to the lower part of the lesser tubercle of the humerus. The second extends from the lower part of the glenoid cavity to the lower part of the anatomical neck of the humerus. The third attaches to the scapula above the top of the
glenoid cavity and passes down along the medial edge of the tendon of the biceps brachii, and is attached to a small depression above the lesser tubercle of the humerus. The articular capsule is also strengthened by two bands from the tendons of the pectoralis major and the teres major respectively. The transverse humeral ligament is a broad band passing from the lesser to the greater tubercle of the humerus, and lies above the epiphysial line. It converts the intertubercular groove into a canal that contains the tendon of the long head of the biceps brachii. The glenoidal labrum is a fibrocartilaginous rim attached around the margin of the glenoid cavity. It deepens the articular cavity, and protects the edges of the bone. The synovial membrane covers the inner surface of the articular capsule, and is reflected from the margin of the glenoid cavity over the labrum. The bursae are fluid filled sacs that protect the tendons and muscles that move against each other.

The movement of the upper arm is described by the rotations about the center of the glenohumeral joint. The movement of the upper arm also includes some of the rotational components of the scapula which are neglected in the shoulder segment. Most measures of upper arm range of motion are taken relative to the torso. This is one of the reasons that rotations of the scapula are considered to be part of the upper arm segment’s range of motion rather than that of the clavicle segment. The length of the upper arm segment is equal to the distance between the glenohumeral and elbow joint centers.

1.1.4 Forearm

The forearm consists of two bones; the radius and the ulna. Between them there exist four joints, the humeroulnar, humeroradial, proximal radioulnar, and distal radioulnar joint. Each of these joints contributes to the structure of the forearm and allow for stability throughout movement. The humeroulnar joint is often described a hinge joint that allows for the flexion and extension of the forearm. Contours of the distal end of the humerus and the proximal end of the ulna fit together tightly and do not allow for significant rotational movement in more than one direction. The humeroradial joint provides support for the forearm by helping to hold the ulna against the humerus and helps transferring force along the arm. The proximal and distal radioulnar joints control the pronation and supination of the forearm. During pronation the radius crosses the ulna. The radioulnar joints allow the radius and ulna to slide against each other without separating.
The center of rotation for the flexion of the forearm is approximated at the distal end of the humerus. The axis of rotation for the pronation of the forearm is approximated as the center of the forearm along its length. In most individuals there is an angle between the forearm and the humerus when the arm is fully extended in the frontal plane, which is referred to as the carrying angle. Because the carrying angle is normally small, varies between persons, and can be compensated for by upper arm rotation for elbow positions of moderate flexion, it is excluded from the model.

1.1.5 Hand

The normal articulations of the hand and wrist allow for the flexion, extension, radial ulnar deviations, and grasping dexterity. The bones of the hand are separated into the carpals, metacarpals, proximal phalanges, intermediate phalanges, and distal phalanges. The large number of bones and joints in the hand make it necessary to greatly simplify the movements of the hand for use in the model. Each of the bones in the hand has its own range of motion. The movements of the wrist are incorporated by the joints between the carpals, radius, and ulna. The wrist is held in place by the radiocarpal ligaments and ulnar and radial collateral ligaments. The metacarpals and phalanges are used to give the hand the necessary dexterity to grasp a wide variety of objects. The motions of the metacarpals and phalanges are omitted from the complexity of the hand in the model due to the fact that grasping strategies of the anatomical hand is outside the scope of this study.

1.2 Other Models of the Human Upper Body

Various studies have produced models of the human upper body; these models differ greatly in degree of complexity and configuration. Most upper body models simplify anatomical joints into a combination of single degree of freedom revolute and prismatic joints that are commonly used in robotic manipulators. For instance the shoulder is often simplified as three revolute joints that have orthogonal intersecting axes. More detailed models are often used to simulate muscle action, and have articulations that resemble anatomical movement with greater accuracy, but these often limit themselves to one joint or segment. In this section several models of the upper body used in different fields are reviewed.

In motion analysis systems, markers are placed on subjects and used to find points and calculate joint locations. Models typically consist of a hierarchy of segments each of which is defined by a number of
markers which are placed on specified positions on a subject’s body. In motion analysis segments are defined by a given set of markers, usually three or more. Three markers are required to establish the location and rotation of a segment and define its coordinate system. The Euler angles between coordinate systems are calculated by the system. Other angles and distances can be calculated based on sets of code and input data such as subject measurements and marker positions. In these systems the complexity of the model used is often determined by the number of markers used. A greater number of markers increases the difficulty of setting up trials and keeping track of individual markers. Models tend to consist of a limited series of segments, i.e. torso, upper arm, forearm, and hand. The number and order of the segments in motion analysis is determined by where markers can be placed, and what the desired measurements are. Motion analysis models are used to retrieve information from a physical system.

Complex simulation models like Maurel [13] are used in systems that include muscle and other soft tissue mechanics usually include definitions that make them as close as possible to anatomical systems. As many articulations as can be handled are included. However, even these relatively complex models often use rotations as defined about a fixed point on a segment. The accuracy of a model is mostly dependent on how well the segments are modeled and the number and complexity of the joints included. Similar but less complex models can be seen in Torres et al. [26] where limited spherical joint are used to simulate arm movements. The degree of complexity of these models is usually depended on a specific aspect to be studied.

Models developed for simulations often use robotic methods. [27, 1, 2] All describe their models as a series of one degree of freedom joints, where higher degree of freedom joint are separated into collections of one degree of freedom joints. It is common for upper arm models to originate at the shoulder. More complex models that include the torso such as Abdel-Malek et al. [1] do exist. The inclusion of specific robotic parameters such as the Denavit and Hartenberg parameters is less common. The use of robotic parameters was chosen for our model because it makes the manipulation of the model less complex reducing computational requirements, and the model itself can be quantitatively described using the robotic parameters. Upper body simulation models are often used to find the stress distributions in segments, or find the motion of segments given various sets of constraint.
1.3 Range of Movement and Nomenclature

It is helpful to have a nomenclature that describes the movement of each joint. Below are a set of pictures that describe the movement of anatomical joints. The movements of joints in the model resemble the movements shown as closely as possible. The movements while performing tasks are complex, and consist of a combination of the movements shown. This may cause the definitions shown to be less clear. The range of motion for every joint given is based on values found in literature and will vary between people. Joint limits given in this section are taken from the American Academy of Orthopedic Surgeons, and can be found in Neumann et al. [18].

Translation of the torso is described relative to the global coordinate system. At initial position the Z axis is the vertical direction, the Y axis describes forward and backward movement, and the X axis describes movement to the left and right.

![Figure 2: Torso Range of Movement](image)

Figure 2 shows the movement of the torso, including some contribution due to changes in the hip rotation. Table 1 shows the joint limits for the spine, it does not include the range of motion provided by the rotation of the hip.

<table>
<thead>
<tr>
<th>Motion</th>
<th>Joint Limit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward/Backward Bending</td>
<td>80° / 25°</td>
<td>Anterior / Posterior bending of the spine.</td>
</tr>
<tr>
<td>Sideways Bending</td>
<td>35° (right &amp; left)</td>
<td>Lateral bending of the spine to the right or left.</td>
</tr>
<tr>
<td>Rotation</td>
<td>45° (right &amp; left)</td>
<td>Rotation about the length of the spine.</td>
</tr>
</tbody>
</table>
Figure 3 shows the movement of the scapula. Movements are considered to be rotations about the approximated sternoclavicular joint. There are no limits given for these scapular movements. The motions of the glenohumeral are described in Figure 4, the joint limits are given in Table 2. Flexion and extension, abduction and adduction, and others are rotational pair and describe opposite directions of the same movement. The nomenclature shown is to help clarify the joint limits by providing a point of reference.
### Table 2: Joint Limits of the Glenohumeral Range of Motion

<table>
<thead>
<tr>
<th>Motion</th>
<th>Joint Limit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abduction</td>
<td>180°</td>
<td>Lateral elevation of the upper arm.</td>
</tr>
<tr>
<td>Flexion / Extension</td>
<td>180° / 60°</td>
<td>Anterior / Posterior elevation of the upper arm.</td>
</tr>
<tr>
<td>Lateral / Medial Rotation</td>
<td>90° / 70°</td>
<td>Rotation about the length of the humerus.</td>
</tr>
</tbody>
</table>

The joint limits are dependent on the orientation of proximal joints in the kinematic chain. If the angle of one of the components of the joint is not zero the joint limits may change. The shoulder joint has a large degree of freedom that causes the nomenclature to become less clear, for instance when abduction is 90° then flexion rotates the upper arm in the transverse plane. Therefore it is important to remember that the order of joint rotations effects the resultant position.

### Figure 5: Elbow Range of Movement

The range of motion of the forearm segment are given in Figure 5, the joint limits are given in Table 3. The complexities of two degree of freedom joints such as the elbow and wrist are less than that of the three degree of freedom joints. The axes of rotation in the joint can remain orthogonal in all positions.

### Table 3: Elbow Joint Limits

<table>
<thead>
<tr>
<th>Motion</th>
<th>Joint Limit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexion</td>
<td>150°</td>
<td>Moving the forearm towards / away from the upper arm.</td>
</tr>
<tr>
<td>Pronation / Supination</td>
<td>80° / 80°</td>
<td>Rotation about the length of the forearm.</td>
</tr>
</tbody>
</table>
Figure 6: Wrist Range of Movement

Figure 6 shows the range of movement of the wrist. Abduction and adduction of the wrist are sometimes referred to as radial and ulnar deviation. Table 4 shows the joint limits of the wrist. In the model the motions of the segments is dependent on the proximal degrees of freedom.

<table>
<thead>
<tr>
<th>Motion</th>
<th>Joint Limit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexion / Extension</td>
<td>80° / 70°</td>
<td>Forward / Backward bending of the wrist.</td>
</tr>
<tr>
<td>Abduction / Adduction</td>
<td>80° / 80°</td>
<td>Bending of the wrist toward the radius / ulna.</td>
</tr>
</tbody>
</table>

1.4 Transradial Prostheses

Transradial prostheses can be divided into, the terminal device, wrist, forearm, socket, and control device. The terminal device’s main purpose is to grasp objects. Hooks and electrically powered grippers are commonly used terminal devices, see Figure 7. A wrist unit allows for the positioning of the terminal device. Most prostheses are equipped with a fixed wrist or a manually adjustable wrist. These wrists can be set to one or more positions but do not allow for the wrist to move while performing tasks.

Figure 7: Terminal Devices. Touch Bionics i-LIMB (Left), Hosmer Model 88X (Right)
Newly developed wrist units are now allowing for greater range movement in the wrist, and wrists with powered rotation are now available. The forearm of the prosthesis adds any necessary length to the prosthesis to create symmetry and expand the reach of the arm; it can also contain battery and control units. A socket is used to attach the device to the residual limb. In some designs, a harness system is used to hold the prosthesis suspended to the body and/or to facilitate function of the terminal device. The control device operates any actuations that exist on the prosthesis. There are two commonly used control devices, body and externally powered. Body powered controls operate the prosthesis by cables that attach to harnesses usually on the shoulders. These controls are routinely used with hook type terminal devices and fixed wrists. Myoelectric controls, the most common type of externally powered system, are operated by electromyographic (EMG) sensors that detect muscle activation. Built in processors analyze the EMG signals and then operate the prosthesis. Myoelectric controls are used with electrically powered prostheses.
Chapter 2: The Model

The model is a kinematic chain with five segments: the torso, clavicle, upper arm, forearm, and hand. The model is repeated with mirrored initial angles for the bilateral tasks. The length of each segment is determined by the subject’s height [R14]. In the basic model there are also five joints, the hip, sternoclavicular, glenohumeral, elbow, and wrist. The hip joint allows for six degrees of freedom, translation and rotation in the three orthogonal directions. The inclusion of the hip is one of the biggest differences between this model and other upper body models, which often start at the glenohumeral joint. The sternoclavicular joint is considered to have two degrees of freedom, and allows for movement of the glenohumeral joint about the sternum. The glenohumeral joint is a three degree of freedom ball and socket type joint. The elbow includes two degrees of freedom: forearm flexion / extension, and pronation / supination. The wrist joint is a two degree of freedom joint for an anatomical limb with adduction / abduction, and flexion / extension. For prosthesis with wrist rotation configuration the degrees of freedom in the wrist are removed. The standard prosthesis configuration also removes forearm pronation and supination.

2.1 Denavit and Hartenberg Parameters

Serial manipulators are often described by Denavit and Hartenberg parameters. These parameters describe serial manipulators by a series of links each of which has one degree of freedom. The parameters for each link are described in Table 5.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link Length</td>
<td>Distance along the line normal to both axes</td>
</tr>
<tr>
<td>Link Twist</td>
<td>Angle between the current link axis and the next link axis</td>
</tr>
<tr>
<td>Link Offset</td>
<td>Distance between the center of the current link and the next along the link axis.</td>
</tr>
<tr>
<td>Joint Angle</td>
<td>Rotation of the link about its axis</td>
</tr>
<tr>
<td>Type</td>
<td>Type of joint Prismatic (p) or Rotational (r)</td>
</tr>
</tbody>
</table>

Table 5: Description of Denavit and Hartenberg Parameters

Figure 8 shows an example of a link to help describe each of the parameters. For a prismatic joint the link offset, d, is the variable, for a rotational joint the link angle, θ, is the variable.
2.2 Parameters of Model Links

The model is created in MATLAB using the Robotics Toolkit [R13], by defining a link for each degree of freedom. The links are defined by the Denavit and Hartenberg parameters listed in Table 6. Joints with more than one DoF have a number of zero length links equal to the number of degrees of freedom of the joint minus one. In joints with two links the axes of rotation will remain 90° apart. In three link joints the angle between the first and second, and the second and third axis of rotation will remain 90° apart, however the angle between the first and third axis of rotation is dependent on the rotation of the second link in the joint. Because of this it is possible for a singularity to occur, where two links share the same axis or revolution, reducing the degrees of freedom of the joint.

<table>
<thead>
<tr>
<th>Link</th>
<th>Description</th>
<th>α twist</th>
<th>A length</th>
<th>Θ angle</th>
<th>d offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>Translation of the hip joint in the Z direction</td>
<td>π/2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>L2</td>
<td>Translation of the hip joint in the Y direction</td>
<td>π/2</td>
<td>0</td>
<td>π/2</td>
<td>0</td>
</tr>
<tr>
<td>L3</td>
<td>Translation of the hip joint in the X direction</td>
<td>π/2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>L4</td>
<td>Torso Bending Backward (+) / Forward (-)</td>
<td>π/2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>L5</td>
<td>Torso Sideways Bending Right (+) / Left (-)</td>
<td>π/2</td>
<td>0</td>
<td>π/2</td>
<td>0</td>
</tr>
<tr>
<td>L6</td>
<td>Torso Rotation Left (+) / Right (-)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.288*h</td>
</tr>
<tr>
<td>L7</td>
<td>Scapular Retraction (+) / Protraction (-)</td>
<td>π/2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>L8</td>
<td>Scapular Depression (+) / Elevation (-)</td>
<td>0</td>
<td>0.129*h</td>
<td>π</td>
<td>0</td>
</tr>
<tr>
<td>L9</td>
<td>Glenohumeral Adduction (+) / Abduction (-)</td>
<td>π/2</td>
<td>0</td>
<td>π/2</td>
<td>0</td>
</tr>
<tr>
<td>L10</td>
<td>Glenohumeral Extension (+) / Flexion (-)</td>
<td>π/2</td>
<td>0</td>
<td>π/2</td>
<td>0</td>
</tr>
<tr>
<td>L11</td>
<td>Glenohumeral Medial Rotation Inward (+)/Outward (-)</td>
<td>π/2</td>
<td>0</td>
<td>π</td>
<td>0.186*h</td>
</tr>
<tr>
<td>L12</td>
<td>Elbow Extension (+) / Flexion (-)</td>
<td>π/2</td>
<td>0</td>
<td>π/2</td>
<td>0</td>
</tr>
<tr>
<td>L13</td>
<td>Forearm Pronation (+) / Supination (-)</td>
<td>π/2</td>
<td>0</td>
<td>π/2</td>
<td>0.146*h</td>
</tr>
<tr>
<td>L14</td>
<td>Wrist Flexion (+) / Extension (-)</td>
<td>π/2</td>
<td>0</td>
<td>π/2</td>
<td>0</td>
</tr>
<tr>
<td>L15</td>
<td>Wrist Adduction (+) / Abduction (-)</td>
<td>π/2</td>
<td>0.108*h</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The model is composed of a 15 link chain. Figure 9 shows links 4 through 15, and a plot of the links of model within MATLAB with its elbow flexed 90 degrees. In the MATLAB plot, each link is labeled by its axis of rotation. The initial angle of rotation of the second link in the three DoF joints is $90^\circ$ so that three degree of freedom joints are orthogonal in the original position. This is done to help avoid singularities caused by links sharing the same axis of rotation. The range of movement in the model is controlled by the inverse kinematic function, see section 3.2. The prosthetic configurations of the model are achieved by a modifier matrix $D$ which restricts or prevents the movement of specified links while solving the inverse kinematics. After the model is run, animations of the model performing the tasks can be generated.

2.3 Bilateral Configuration

In order to perform bilateral task the model is repeated with revised joint angles so that it can represent the other arm. For unilateral task only the limb under consideration is analyzed. For bilateral tasks we apply the prosthetic constraints to the appropriate side. The movement of the hip is connected to the side that is given the prosthetic constraints. The anatomical limb still has nine degrees of freedom from the wrist, elbow, glenohumeral, and sternoclavicular joints. The details of the configurations are described in the chapter 3.
Chapter 3: Analysis of the Model

This chapter deals with the use of the model to analyze human movements, and specifically compensatory motions. The first section defines the general motions that the model must make, and describes the specified paths or trajectories that were created which represent the selected tasks. The second section reviews the inverse kinematics used to control the model during the task simulation. The third section reviews the procedures used to optimize the inverse kinematics increasing the models resemblance to human motions.

3.1 Tasks & Trajectories

The tasks selected for use in this study are activities of daily living: opening a door, drinking from a cup, turning a steering wheel, and lifting a box. Of these tasks, opening a door and drinking from a cup are unilateral, and turning a steering wheel and lifting a box are bilateral. In the simulation, the performance of the model is determined by how similar it is to a human motion the better the correlation between human movement and the model the higher the performance. The tasks in the simulation are much simpler than tasks in real life. The simulated tasks are composed of relatively simple trajectories with a number of constraints. In real life, there are an infinite number of trajectories that could accomplish a given task; however, in the simulation, the trajectories remain constant for all of the trials. The goal of this study was not to task trajectory analysis, but to predict compensatory motions due to the limitations of the prosthesis. A simple single trajectory for each tasks makes allows for direct analysis of compensatory motion. In this study idealized trajectories that are as simple as possible to accomplish the given tasks were used. The idealized trajectories will also make the difference between a real life task and the simulated task easier to compare. Each trajectory has 100 points or samples where the inverse kinematics were calculated; the points are equally spaced along the trajectory. Restriction and prevention of joint motion is accomplished by the joint weighting factors, see section 3.2.3.
3.1.1 Opening a Door

The opening a door task has the least number of constraints on the model. Translation and rotation of the hip were restricted but not prevented. For this task, there are many different consideration, for example, the type of door, the direction it is opening, and the handle. For this study, a household interior door with a round knob was chosen as the basis for the task, so that the end effector must rotate about the knob’s axis. The height of the knob was 0.93m from the ground and the distance from the hinge axis to the knob center was 0.71m. These distances are based on measurements of home interior doors and are not necessarily standard or average for their type. The door task was modeled opening the door toward the model body and was run twice, once for a right hung door, and once for a left hung door. The model was not run for both the left and right arms as the joint angles for a right hand opening the right hung door will be the same as a left hand model opening a left hung door, assuming mirrored configurations and the same weighting factors are used.

The trajectory used for the opening a door task starts with the rotation on the door knob, and is followed by the swing of the door. Turning the knob rotates the end effector about its x-axis 90°. The swing of the door follows a 0.71m radius arc with the x-axis of the end effector remaining tangent to the path of the arc and travels so that the door would open 90°.

3.1.2 Drinking From a Cup

For the drinking from a cup task there are two steps. The first step starts with elbow at 90° and the z-axis in a vertical position. While maintaining the z-axis’ vertical orientation, the end effector’s position is raised to a position near the mouth. In the second step, the cup is held in position and rotated 90° from the vertical orientation towards the mouth of the simulation. The points along the trajectory are a function of height so that the position of the hand is relative to the length of the model segments and the joint angles become independent of subject height. For this task, movement and rotation of the hip joint is prevented because the position of the cup’s position relative to the body is required for the task.

3.1.3 Turning a Steering Wheel

For the turning a steering wheel task, the model starts with its elbows bent and the hand located 0.50m in front of the models coronal (frontal) plane, and 0.50m apart on the edge of the steering wheel at the three and nine o’clock positions. The hands must then rotate around the center of the wheel which forms a
circular path in a plane parallel to the coronal plane at the initial position. The wheel rotates 90° in one direction, 180° in the other direction, and then returns to its original position. The orientation of the hands remains tangent to the edge of the wheel. Translation of the hip is prevented in this trial to simulate the model sitting in a fixed position.

3.1.4 Lifting a Box

For the lifting a box task, the task starts with the arms in front of the model, the hands 1 meter from the ground, on both sides of a box. The hands move toward the model 0.3m, then up 0.6m, then away from the model 0.3m. The distance between the hands, and the orientation of the hands remains constant for the entire trial. The hip is prevented from moving in this task to increase the similarity to the recorded task in previous studies [5].

3.2 Inverse Kinematics

Inverse kinematics is the method of solving for joint parameters given the desired position of the end effector. The model of the upper limb is a 15 degree of freedom system. This means that there are 15 independent variables that describe the position of the model. The end effector is completely described by 6 variables, 3 for position, and 3 for orientation. The extra degrees of freedom are redundant. This redundancy allow for an infinite number of joint positions to exist for a given end effector position, so long as that position is within the workspace of the model. This redundancy also means that there is no direct solution to solve the inverse kinematics of the model. In order to solve the inverse kinematics of the system, additional constraints must used that allow for a single solution among all of the possible solutions to be found. It is also necessary to find a method that will mimic human behavior as closely as possible so that the motions produced are an accurate prediction of compensatory movement.

3.2.1 Forward Kinematics

Before solving the inverse kinematics the forward kinematics must be defined. If \( \mathbf{r} \) is defined as the end effector position and orientation of the model, the forward kinematic solution states:

\[
\mathbf{r} = f(\mathbf{\theta})
\]

where \( \mathbf{\theta} \) is the joint parameter vector of the model and \( f \) is the forward kinematic function. The forward kinematic function is calculated by computing the product of the transform for each link based on the
Denavit and Hartenburg parameters and the joint parameter $\theta$. For this study the ‘fkine’ function from the robotics toolkit [R13] was used to solve the forward kinematics.

### 3.2.2 Jacobian & Pseudo Inverse

One of the methods to solve the inverse kinematics of a redundant manipulator is to use the pseudo inverse of the Jacobian. The Jacobian is a matrix of first order partial derivatives of a vector valued function, and specifies a mapping from velocities in joint space to velocities in Cartesian space [7]. The change in end effector position, $\Delta r$, can be described by the Jacobian, $J(\theta)$, which is $\frac{\partial r}{\partial \theta}$ and the change in joint parameter vector, $\Delta \theta$.

$$\Delta r \approx J(\theta)\Delta \theta$$  (2)

With some initial joint configuration, $\theta_t$, and a desired end effector position, $r_{t+1}$, the solution of the joint angles, $\theta_{t+1}$, at the desired end effector position can be found. First the forward kinematics are used (1) to solve $r_t$. Then the desired change in the end effector position is found.

$$r_{t+1} - r_t = \Delta r$$  (3)

With $\Delta r$ a known value from (3) the solution of $\Delta \theta$ is found by taking the inverse of the Jacobian. However, because the Jacobian is a $6 \times 15$ matrix it is not invertible, therefore the pseudo inverse $J^+(\theta)$ must be used to find the solution.

$$J^+(\theta)\Delta r \approx \Delta \theta$$  (4)

$$\theta_{t+1} = \theta_t + \Delta \theta$$  (5)

Because this is not an exact solution, for large changes in position iterations need to be performed where $\theta_t$ is replaced by $\theta_{t+1}$ and the process is repeated until $\Delta \theta$ becomes less than a specified tolerance. For the model the change in end effector position is small enough that the error can be ignored. This method also uses joints based on how effectively they move the end effector, minimizing the norm of $\Delta \theta$, so it tends to move proximally located joints rather than distal located ones. This method is can be solved quickly it does not account for any obstacle avoidance, or account for joint limitations.

### 3.2.3 Weighted Pseudo Inverse

To improve the pseudo inverse method, a weighting matrix is added to the equation. The weighting matrix is a diagonal matrix that is $n$ by $n$ where $n$ is the number of joints of the robot, in this case 15. The
diagonal values of the matrix are the weighting factors for each link where 1 is normal movement and 0 is no movement in that link. For this study the weighting matrix is referred to as \( D \).

\[
D[(J(\theta)D)^+\Delta r] \approx \Delta \theta
\]  

(6)

The matrix \( D \) has to post multiply the Jacobian before the pseudo inverse is taken, and pre-multiply the new modified pseudo inverse of the Jacobian. This method allows for a good deal of control of the model while performing the inverse kinematics by controlling the rate at which a joint moves. However, the rate is only controlled with respect to itself and therefore low weighting factors are often necessary to make the changes apparent. This method is very good for removing degrees of freedom from the model; any link that needs to be removed can be given a weighting factor of zero, and will act as a rigid segment. The use of dynamic weighting factors to control joint limits was attempted. However, it caused a large increase in computational time and caused joints to lock near their limit. The use of a large number of low weighting factors also effectively reduced degrees of freedom and can lead to non-convergence errors in the iterative process. It was determined that the weighting matrix is best used on a limited number of links and that other methods will be better for controlling joint limits, and for obstacle avoidance.

3.3 Optimization Functions

3.3.1 Exploiting Redundancy

Because we have a redundant manipulator, there are an infinite number of solutions to the inverse kinematic problem. In the basic equation for solving with the pseudo inverse (4), the solution that has the minimum \( \theta \) is found. This redundancy can be used to manipulate the arm in null space, the range where the joints of the arm can move without changing the end effector position. We can specify a function or even a series of functions which we wish to minimize: \( H(\theta) \).

\[
\Delta \theta \approx J^+(\theta)\Delta r - (I - J^+(\theta)J(\theta))\nabla H(\theta)
\]  

(7)

Where the gradient vector \( \nabla H(\theta) \) is defined as (8), and \( I \) is a \( n \) by \( n \) identity matrix.

\[
\nabla H(\theta) = \begin{bmatrix}
\frac{\partial H}{\partial \theta_1} \\
\vdots \\
\frac{\partial H}{\partial \theta_n}
\end{bmatrix}
\]  

(8)
The next step is to develop a series of equations to optimize the inverse kinematics. To allow for multiple optimization functions we can combine a series of equations.

\[ H(\theta) = k_a A(\theta) + k_2 B(\theta) \ldots + k_n N(\theta) \]  \hspace{1cm} (9)

The \( k \) coefficients allow for priority among the separate optimization functions to be established. For this model we can include optimizations for joint limits, and functions that help simulate human like movements.

### 3.3.2 Joint Limits

The first optimization function used was for maintaining the joint limits of the model. Because the joints are anthropomorphic the joint limits are not simple ranges, so our function needs to work with any combination of minimum and maximum joint angles.

\[ A(\theta) = \sum^n \left[ \left( \frac{\theta_{\text{max}} - \theta_{\text{min}}}{2} \right)^2 - \left( \theta_i - \frac{\theta_{\text{max}} + \theta_{\text{min}}}{2} \right)^2 \right]^{-1} \]  \hspace{1cm} (10)

The basis for equation (10) is found in [16]. The function is minimized at the average value of \( \theta_{\text{max}} \) and \( \theta_{\text{min}} \), the function also approaches infinity as \( \theta_i \) approaches the joint limits. The next step is to find the gradient of the function. We can simplify the function first to make finding the gradient easier.

Because \( \theta_{\text{max}} \) and \( \theta_{\text{min}} \) are constants we can make the following simplifications.

\[ \left( \frac{\theta_{\text{max}} - \theta_{\text{min}}}{2} \right) = \theta_{\text{diff}} \quad \& \quad \left( \frac{\theta_{\text{max}} + \theta_{\text{min}}}{2} \right) = \theta_{\text{avg}} \]  \hspace{1cm} (11)

\[ A(\theta) = \sum^n \left[ \theta_{\text{diff}}^2 - (\theta_i - \theta_{\text{avg}})^2 \right]^{-1} \]  \hspace{1cm} (12)

\[ A(\theta) = \sum^n \left[ \theta_{\text{diff}}^2 - \theta_i^2 + 2\theta_i\theta_{\text{avg}} - \theta_{\text{avg}}^2 \right]^{-1} \]  \hspace{1cm} (13)

\[ \frac{\partial A(\theta)}{\partial \theta_i} = (2\theta_i - 2\theta_{\text{avg}}) \left[ \theta_{\text{diff}}^2 - \theta_i^2 + 2\theta_i\theta_{\text{avg}} - \theta_{\text{avg}}^2 \right]^{-2} \]  \hspace{1cm} (14)

Equation (14) shows the gradient form of the equation.
Chapter 4: Model Programming

This section covers the programming of the model and the operation of the simulation. In addition to the files developed for this study the model must be run on a computer with MATLAB (version 2007a or newer), the robotics toolkit developed by Craig [R18], and the virtual reality toolkit. The virtual reality toolkit is only needed to run the 3D simulation. The calculations are done in the background and processing time will depend on hardware specifications.

4.1 MATLAB File Library

This section contains a list of the files developed for this study with descriptions of their purpose.

4.1.1 ModelGUI.m

This file initializes the model and the graphical user interface. It contains the code to define the links, model, graphics, and callbacks. Links are defined by the robotics toolkit command ‘link’ and the Denavit and Hartenburg parameters. The links are used to create a serial manipulator with the ‘robot’ command and an array of the links. Graphics used for the elements in the interface are native and require no additional files. Each of the elements in the graphical user interface has a callback. A callback tells the program to perform some action given some user input. For example, a button callback can perform some function when the button is pressed. ModelGUI is a function file, coded in MATLAB; the callbacks are nested functions within ModelGUI. To activate the graphical user interface, the directory that contains the files must be set to the working directory or added to the MATLAB path directories using the ‘path’ command. The interface can be opened by typing ModelGUI into the workspace. Any errors that occur while running the program will be displayed in the MATLAB workspace.

Figure 10 shows a screen shot of the model’s graphical user interface, each section is labeled to help describe the interface. Sections 5 – 8 are hidden until the solution to the inverse kinematics is found for the selected components. A detailed description of each function is given below in Figure 10.
1. The menu reads all of the *.mat files in the trajectories folder, and allows the user to select the trajectory for the task they wish to use.

2. The menu reads all of the *.mat files in the configurations folder, and allows for a preset configuration to be selected.

3. Allows the diagonal elements of the weighting matrix to be defined manually.

4. Loads the trajectory and configuration files selected in boxes 1 and 2, overwrites weighting matrix if the box in 3 is checked, and runs the inverse kinematic function ‘Dkine’ with the selected variables. Generates parts 5 – 8 on the interface.

5. Shows the results using a wireframe of the model, uses the robotics toolkit ‘plot’ command.

6. Initializes the virtual reality simulation of the model of the upper limb.

7. Allows the user to select and plot the angle of a link.

8. Saves the variables created by the program into a *.mat file.
4.1.2 /trajectories

This folder contains the task trajectories that we can use to analyze the model. Trajectories for ‘drinking from a cup,’ ‘opening a door,’ ‘turning a steering wheel,’ and ‘lifting a box’ which were described in chapter 3 are available. The tasks are given in *.mat files that contain two trajectories each, one for the left and right configurations. Any *.mat files that are added to this directory will show up on the selection menu of the graphical user interface, however if the *.mat files does not contain the variables TR and TL which are the trajectories of the right and left arm respectively it will return an error. The files can also contain any constraint variables that are prevalent to the task such as modifications to the weighting matrix.

4.1.3 /configurations

This folder is similar to the trajectories folder; however it contains the files that define preset anatomical configurations. The configurations are given in *.mat files as well and the graphical user interface loads them to the configuration menu in the same manner as the trajectories. The files for configuration contains the joint modifier matrix with diagonal values of one or zero, to keep a joint activate or make it inactive. The selection of the ‘Manually Edit Weighting Parameters’ box causes the values given by the configuration files to be overwritten by the manually entered values.

4.1.4 Virtual Reality

The virtual reality model is a 3D representation of the upper body. It was created with VRML 2.0, and contains segments for each link. The skin of the model was taken from an open source model of an adult male and remodeled to match the segments of the model. Because the segments of the model are all rigid bodies there are some intersecting surfaces within the model and various positions may cause the model to look less anatomical. VRML 2.0 was used due to its integration into MATLAB with the virtual reality toolkit. This model is purely for visualization and is not involved in any of the calculations. The model does not scale to size but operates off of the joint variable outputs from the inverse kinematics, and therefore is independent of link length.
Chapter 5: Verifying the Model

In order to verify the model it was compared with subject data from Carey et al. [5]. In Carey et al.’s study the kinematic and kinetic data of subjects performing activities of daily living tasks was recorded using the Vicon motion analysis system. The model used in the Carey et al. study was somewhat different than the one used in this study, the Carey et al. model did not include a segment for the clavicle. However the markers used in the study are sufficient for matching to our model. The marker data was used to recreate the model in Vicon, and make the necessary changes to allow the data to be meaningfully compared to the solutions of the model trajectories found by the inverse kinematic methods used in this study.

5.1 The Previous Model

The previous model developed by Carey et al. [5] for motion analysis is shown in Figure 11. Carey’s model uses the markers described in Table 7 to record the subject upper body movements. The markers are placed on the skin of the subject and are recorded by an eight camera array. The Vicon software analyzes the images from the cameras and computes the location of each marker in 3D space. Simple mathematical equations can be entered into the Vicon software to calculate joint center locations. There are native functions that allow for the defining of segments based on any calculated point or marker.

Figure 11: Vicon Model Layout [5]
Figure 11 shows a screen capture of the Carey et al.’s model as it appears in one of the displays available in the Vicon software. The white spheres represent recorded marker positions, the blue spheres are calculated joint centers, the green lines represent a particular segment’s axes, and the red lines are used to show grouping of markers, which makes it easier to tell which markers are which but aren’t generally used in any calculations.

<table>
<thead>
<tr>
<th>Code name</th>
<th>Marker description</th>
<th>Marker placement</th>
</tr>
</thead>
<tbody>
<tr>
<td>C7</td>
<td>7th Cervical vertebrae</td>
<td>Spinous process of the 7th cervical vertebrae</td>
</tr>
<tr>
<td>T10</td>
<td>10th Thoracic vertebrae</td>
<td>Spinous process of the 10th thoracic vertebrae</td>
</tr>
<tr>
<td>CLAV</td>
<td>Clavicle</td>
<td>Jugular notch where the clavicles meet the sternum</td>
</tr>
<tr>
<td>STRN</td>
<td>Sternum</td>
<td>Xiphoid process of the sternum</td>
</tr>
<tr>
<td>RBAK</td>
<td>Right back</td>
<td>Middle of the right scapula (asymmetrical)</td>
</tr>
<tr>
<td>RSHO</td>
<td>Right shoulder</td>
<td>Right acromio-clavicular joint</td>
</tr>
<tr>
<td>RUPA</td>
<td>Right upper arm</td>
<td>Right upper arm between the elbow and shoulder markers</td>
</tr>
<tr>
<td>RELB</td>
<td>Right elbow</td>
<td>Right lateral epicondyle approximating elbow joint axis</td>
</tr>
<tr>
<td>RELBM</td>
<td>Right elbow medial</td>
<td>Right medial epicondyle approximating elbow joint axis (static trial only)</td>
</tr>
<tr>
<td>RWRA</td>
<td>Right wrist A</td>
<td>Right wrist thumb side</td>
</tr>
<tr>
<td>RWRB</td>
<td>Right wrist B</td>
<td>Right wrist pinkie side – on the pisiform</td>
</tr>
<tr>
<td>RFIN</td>
<td>Right finger</td>
<td>On the dorsum of the hand just below the head of the right third metacarpal</td>
</tr>
<tr>
<td>LSHO</td>
<td>Left shoulder</td>
<td>Left acromio-clavicular joint</td>
</tr>
<tr>
<td>LUPA</td>
<td>Left upper arm</td>
<td>Left upper arm between the elbow and shoulder markers</td>
</tr>
<tr>
<td>LELB</td>
<td>Left elbow</td>
<td>Left lateral epicondyle approximating elbow joint axis</td>
</tr>
<tr>
<td>LELBM</td>
<td>Left elbow medial</td>
<td>Left medial epicondyle approximating elbow joint axis (static trial only)</td>
</tr>
<tr>
<td>LWRA</td>
<td>Left wrist A</td>
<td>Left wrist thumb side</td>
</tr>
<tr>
<td>LWRB</td>
<td>Left wrist B</td>
<td>Left wrist pinkie side</td>
</tr>
<tr>
<td>LFIN</td>
<td>Left finger</td>
<td>On the dorsum of the hand just below the head of the left third metacarpal</td>
</tr>
</tbody>
</table>

Segments in Vicon are defined by three or more markers. These makers make up the origin and two lines. The origin represents the center of that segment’s axes. The first line determines the direction of the first segment axis. The cross product of the first second line is used as the second axis, and the cross product of the first and second axes becomes the third axis. This is based on the Gram-Schmidt process of

26
orthogonalization. The first, second, and third axis are labeled according to a tag where each axis can be named x, y, or z. The joint centers were calculated in [5] for the shoulder, elbow, and wrist.

5.2 Adapting to the New Model

To make the data compatible with the model the segments in the Vicon software to defined to match the segments of the model. For most of the joints the segments and joint centers found in [5] were used for this study. An equation, calculating the position of the pseudo sternoclavicular joint center was developed to define the origin of the clavicle segment. An offset of one fourth the distance between CLAV and C7 markers was taken. A point was created from CLAV along the torso’s x-axis by this offset. This new joint center is called PCLAV. The shoulder joint center is the origin of the shoulder segment. The line from the shoulder joint center to PCLAV is the clavicle segment’s first defining line, and the line from the shoulder joint center to the CLAV marker will be the second defining line.

With matching segments in MATLAB and Vicon, the data was then transferred from one system into the other. The data from is the trials is constant; the numbers do not change and there are no inputs. The model in MATLAB is dynamic; the output is depended on the various input parameters. Therefore, the trial data was exported from Vicon to MATLAB and the comparison was made within the MATLAB interface. The specific data that was taken from the trials in Vicon were the joint angle data. Joint angle data in Vicon is described by two segments and an order of operation. The Euler angles were calculated for rotating the first segment’s coordinate system to match the second segment’s coordinate system along the x, y, and z axes in the order of operation. This gives three rotations for every joint in the Vicon model, and for every time step or sample in the trial. The first verification of the model that was performed was to make sure the sternoclavicular, elbow, and wrist joints have two non-constant rotations. This indicated that each of these joints only has two effective degrees of freedom.

After checking the consistency between the two models, the joint angles can be exported from Vicon in the form of a tab delimited array in *.txt file format. This file was then imported into MATLAB and converted into joint angle arrays that can be used directly with the MATLAB model, the file was backed up as a *.mat file for future use.
5.3 Comparing Data

A direct comparison was made using the end effector trajectory, \( r_{\text{trial}} \), generated by the joint angle data imported from the trials recorded in Vicon. \( \theta_{\text{trial}} \). The forward kinematic solution, see section 3.2.1, was used to solve for the end effector homogeneous transform for every sample in a given trial (15). The homogeneous transforms were used as a new trajectory; the first sample joint values were used as the initial configuration. A new set of joint angles were found by the inverse kinematic function (for descriptions see section 3.2 Inverse Kinematics). The difference between joint angles is the error of the model.

\[
 r_{\text{trial}} = f(\theta_{\text{trial}}) \quad (15)
\]

\[
 \text{Inverse Kinematics}(r_{\text{trial}}) = \theta_{\text{ikine}} \quad (16)
\]

\[
 \text{error} = \theta_{\text{trial}} - \theta_{\text{ikine}} \quad (17)
\]

This error is only valid for the specific trial and must be normalized to be used in a generalized form.

\[
 \text{average error} = \frac{\sum |\text{error}|}{N} \quad (18)
\]

The average error is a sum of the absolute values of all the error divided by the number of samples in the trial. With the average error for a number of trials we can verify our inverse kinematic methods according to a number of conditions. A sum total error for all tasks was found. The error for each specific task and the error when we vary the joint limit function in the inverse kinematic methods was also found.

5.3.1 Testing Task Matching

After analyzing the model and calculating the average error for each trial, the sum of the average error per link of the trials for each task and for all tasks was calculated, as shown in Table 8. Notice that the error for the left side on the Cup and Door tasks is very low for the \( K_a = 0 \) condition. Because the drinking from a cup and opening a door are unilateral tasks, and all of our subjects used their right hand for the task, there is very little movement in the left arm, resulting in the low error. The left side of the lifting a box trial was the only task where the joint limit constant \( K_a \) greater than zero resulted in an average error less than the average error for the \( K_a \) equal to zero. The higher \( K_a \) generally caused a higher error, and the error was highest for the turning the wheel task where \( K_a \) is equal to 0.1. This shows that the joint limit function decreases the performance of the model.
Table 8: Average Joint Angle Difference (Degrees / Joint)

<table>
<thead>
<tr>
<th></th>
<th>Left</th>
<th>Right</th>
<th>Left</th>
<th>Right</th>
<th>Left</th>
<th>Right</th>
<th>Left</th>
<th>Right</th>
<th>Left</th>
<th>Right</th>
<th>Left</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>5.22</td>
<td>7.35</td>
<td>12.33</td>
<td>13.43</td>
<td>16.21</td>
<td>16.24</td>
<td>17.22</td>
<td>16.62</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cup</td>
<td>0.50</td>
<td>4.61</td>
<td>8.99</td>
<td>11.36</td>
<td>11.88</td>
<td>15.31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Door</td>
<td>2.09</td>
<td>8.60</td>
<td>8.65</td>
<td>12.35</td>
<td>14.21</td>
<td>11.32</td>
<td>14.47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wheel</td>
<td>11.15</td>
<td>6.93</td>
<td>20.49</td>
<td>21.21</td>
<td>25.42</td>
<td>26.08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Box</td>
<td>12.77</td>
<td>10.92</td>
<td>11.06</td>
<td>14.76</td>
<td>11.89</td>
<td>17.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 12 and Figure 13 show the trend for the average errors.
Chapter 6: Results

To analyze the difference between the joints in each configuration, the difference between the anatomical configuration and the given prosthetic configuration within the model is defined as the degree of compensation (DoC).

\[
\text{DoC} = \theta_{\text{anatomical}} - \theta_{\text{prosthetic}}
\] (19)

The performance of a prosthetic configuration is dependent on its DoC, the higher the DoC the worse the performance of the prosthetic configuration. For most joints there is a higher DoC in the standard prosthesis configuration than in the configuration with a wrist rotator.

6.1 Drinking From a Cup

For this task only the anatomical and the prosthesis with wrist rotator configurations were able to complete the task. The prosthesis without wrist rotation was unable to turn due to insufficient degrees of freedom with the hip joint restricted from movement. In Figure 14 and Figure 15 the joint angles for the anatomical and wrist rotator configurations are shown. There is a drastic difference in the path of these joints for the 9th and 10th links, the change in angle of rotation is nearly opposite.

![Figure 14: Glenohumeral Joint Angles. Initial angles 90 degrees.](image-url)
There is a large difference between the joint angles in this task even though the task is fairly simple. In the second step, 70% through 100% of the trial, the anatomical arm configuration, has a more complex movement, while in the wrist rotator configuration only link 13, forearm rotation, changes.

### 6.2 Opening a Door

The door opening task has much more varied results than that of the drinking from a cup task. The main factors in creating the disparities in this task are the number of joints of interest, the scale of the movement, and the ambiguity of the grasp on the handle. Weighting factors were used on the hip and sternoclavicular joints and are given in Table 9.

<table>
<thead>
<tr>
<th></th>
<th>n=1</th>
<th>n=2</th>
<th>n=3</th>
<th>n=4</th>
<th>n=5</th>
<th>n=6</th>
<th>n=7</th>
<th>n=8</th>
</tr>
</thead>
<tbody>
<tr>
<td>D(n,n)</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Data given is in meters for L1-3 and degrees for L4-15 for both right and left hinged cases. The joint angle given is based on the model and should be considered with respect to the initial angle and direction data given in Table 6: Model Parameters.

![Figure 15: Elbow Angles. Initial angle is 90°, flexion is negative.](image)

The difference between the anatomical line and that of the prostheses represents the degree of compensation for each link. It is shown that the degree of compensation varies greatly for each joint. For
the majority of links the wrist rotator configuration is closer to the anatomical configuration than the standard prosthesis, thus it has a lower degree of compensation. In link 12 (elbow flexion) for the right and left hinged door tasks with the wrist rotator and standard prosthesis have similar movements.

6.3 Turning a Steering Wheel

In turning a steering wheel, the primary compensation was in the lateral flexion of the torso and hip joint. There was also a maximum DoC of 69° for the configuration with a wrist rotator and 46° for standard prosthesis configuration of depression of the sternoclavicular joint. The total amount of movement for this task made the motions rather quick compared to the lifting a box task. Figure 2 shows the first quarter of the task for the anatomical configuration.

![Figure 17: Screen Capture of the Turning a Wheel Task. Start of task on the left, end of the first turn on the right, anatomical configuration.](image)

Figure 18 and Figure 19 show the joint angles for the sternoclavicular joint in elevation and depression, and lateral torso flexion respectively. Note the difference between each model configuration.

![Figure 18: Plot of Scapular Elevation / Depression Angles. While performing the turning a steering wheel task, the initial angle is 180°, and depression is positive.](image)
As shown in Figure 18 the joint angle for the sternoclavicular joint has a higher DoC in the configuration with the wrist rotator than in the standard prosthesis configuration. This is the only case of this occurring in these tasks. The asymmetry about the middle of the task seen is the result of the unilateralism of the prosthetic configurations. In Figure 19, the standard prosthesis has a much higher DoC; this is true for most joints.

![Figure 19: Plot of Torso Sideways Bending.](image)

The range over which different compensations can occur narrows as the number of degrees of freedom is deceased. The weighting factors increase the movement of the distally located joints by restricting the proximal joints, such as the hip and sternoclavicular joints. As system’s the degrees of freedom are reduced the weighting factors become less significant because it becomes necessary for the weighted joints to move in order to find a solution to the inverse kinematics.

The peaks seen in Figure 18 and Figure 19 occur at the maximum turn angles, at 25% and 75% of task completion. The model also returns to the initial configuration at the middle and end of the task for all configurations. This return to the initial configuration, as well as the symmetry of the lateral torso flexion, shows that the attachment of the torso to the left side of the model is not causing the model to favor movement on one side.

### 6.4 Lifting a Box

Figure 20 shows the anatomical configuration of the model at four points while performing the lifting a box task. The anatomical configuration is able to complete the task with very little movement of the torso.
and clavicle. The large amount of adduction of the wrist shown is a clear indicator that there will be a large DoC for this trial.

![Figure 20: Screen Capture of the Lifting a Box Task](image)

Task starts on the upper left and ends at lower right, anatomical configuration.

![Figure 21: Plot of Forward / Backward Bending of the Torso](image)

While performing the lifting a box task, the initial angle is 0°, and extension is positive.

For the lifting a box task the primary compensation was in the flexion and extension of the torso, shown in Figure 21. Similar compensation also occurred in the majority of the other joints. The plot of the scapular protraction / retraction, shown in Figure 22 and torso rotation closely resemble that of torso flexion / extension. This was the expected behavior for this task. Here the various degrees of freedom are trying to accomplish the same motion and therefore have a similar profile. The configuration with the wrist
rotator has a greater number of degrees of freedom and therefore the DoC is less. The task never returns to a previous position so no symmetry is seen. The bumps shown in Figure 21 and Figure 22 occur at 30% and 70% of task completion and are the transition points between moving the “box” towards the model, raising the “box,” and moving the “box” away from the model.

![Figure 22: Plot of Scapular Protraction / Retraction Angle. While performing the lifting a box task, the initial angle is 0, and adduction is positive.](image)

The rigidity in the wrist in the prosthetic configurations caused the arms to bunch up tightly behind the model subject, as shown in Figure 23. The right side of the model in the prosthetic configurations appeared like the movements in shown in Figure 20, with some slight variance in the configuration with the wrist rotator causing the elbow to be raised slightly.

![Figure 23: Screen Capture of the Model at 70% of Task Completion. Standard prosthesis configuration during the lifting a box task, notice the tight compaction of the left arm.](image)
Chapter 7: Discussion

The results of this study demonstrate the difference in joint angles of a human upper body model with configurations representing an anatomical arm, a prosthesis with a wrist rotator, and a standard prosthesis while performing these two unilateral and two bilateral activities of daily living. These data suggest that there is a general reduction in the degree of compensation with the addition of the wrist rotator which was predicted. One exception to this occurred at the sternoclavicular joint during the turning a steering wheel task. The compensatory movements shown in the model are large, with the maximum compensation approaching 90 degrees. The subjects in Carey et al.’s study [5] were able to complete similar real life tasks with standard prosthesis, and lower degrees of compensation. This is likely due to the fact that they can choose a trajectory that best fits their range of motion and the specified task, and that there are no strict rules for performing the task (i.e. grasp methods, stance angles, etc.). It may be better to optimize the trajectory of each task for every configuration; however this would lead to a less direct method of comparison.

The animation of the drinking from a cup trial proceeded largely as expected and resulted in a qualitatively satisfactory human-like motion. The inability of the prosthetic configuration to accomplish the task was not surprising. With only the 6 degrees of freedom of the shoulder, upper arm, and flexion of the forearm the change in orientation of the end effector is limited to specific directions and locations.

In the door opening trial a number of links in the standard prosthesis and the prosthesis with wrist rotator have very similar motions, which are different from that of the anatomical configuration. This is possibly due to the direction in which this compensation is being forced. That is, the wrist rotator can only help with movements that effect the rotation of the forearm, so for movements that require motion in a separate direction there is no difference between the standard and the configuration with wrist rotation.

In the turning a steering wheel task the model nearly reaches a DoC of 90° in lateral flexion to accomplish the task. While this is large compensation, the model is able to return to the initial configuration when the trajectory passes and returns to the initial position. Other methods of turning the wheel are
considerably more likely to be used. This shows the ability of the model to maintain a rational configuration over large deflections. The inclusion of a dynamic grip, or the removal of orientation constraints on the hands during the simulation would likely reduce the DoC, however it would not test the bilateral aspects of the model as extensively.

The lifting a box task showed an unlikely proximity of links toward the task’s end, which decreased the models ability to extend the arms again due to the method of inverse kinematics. This is a good example of the real life task being different from the modeled task. In Carey’s study [5] it can be seen that prostheses users tend to swing the box to one side while raising it. This movement decreases the need for flexion of the wrist. The model cannot select a new trajectory for the box and therefore forces itself into a disadvantageous position. This highlights some of the disadvantages of these methods. It was determined that the same task should be used for all configurations so that a direct comparison could be made, and it was expected that this might cause some problems.

The method of achieving a bilateral model was successful. By providing the extra range of motion of the torso to the arm with the prosthetic configuration, the model was able to find a solution to all points on each trajectory. The right arm was able to accomplish the right hand trajectory with the nine degrees of freedom remaining after the angles for the torso were defined.

7.1 Limitations

This study has several limitations that affect the results.

1. All segments are considered rigid bodies: in the human body no segments are truly rigid bodies.
   This approximation was made because the relative motion of the joints with respect to deformation in the segment lengths is very large.

2. Anatomical joints were approximated by constant centers of rotation: segments with a large number of articulations were reduced into gross movements with an approximated joint center.
   This causes some error in the range of motion the model is intended to represent.

3. A Limited number of task were analyzed.

4. Some anatomical features were omitted: the model excluded the carrying angle of the elbow, and did not include any motions of the head.
5. Each task was tested with only one gripping angle, changing the gripping angle will change the resulting compensatory motion.

6. Each task was only performed with one trajectory; there are an infinite number of trajectories that can perform a similar task, Carey et al. have show that the trajectory used by a person with prosthesis varies from non-prosthesis users.

7. Joint limit functions were omitted based on results from simulated tasks due to the decreased correlation between recorded and simulated trials.

8. No functions for collision avoidance were developed or tested.

9. The weighting factors for each task were determined by trial and error.

10. The model requires the use of MATLAB and the robotics toolkit, and therefore cannot be easily distributed.

7.2 Future Work

Work is being done to improve the model by decreasing the number of limitations of the model. Gradient functions for the joint limits are currently being researched for this model, with specific functions for each link, with the goal of increasing correlation between recorded and simulated movement. The inclusion of dynamic gripping constraints, and generating a larger number of task trajectories would help to improve this study. A larger number of tasks results in a better understanding of generalized compensatory motion as well as the performance of the model. The analysis of additional task would improve results, specialized task such as rock climbing, and kayaking will be added to the model.

7.3 Conclusions

The model of the upper arm was developed. Using the weighted least norm method we the joint angles were calculated for each point on the trajectories that we described for the tasks of drinking from a cup, opening a door, turning a steering wheel and lifting a box. Prosthetic constraints were placed on the model and the inverse kinematics were calculated. The difference between the results of the anatomical, standard prosthesis, and prosthesis with wrist rotator configurations of the model were analyzed.

The model was able to simulate upper limb movements, and show differences in joint angles while performing tasks with prosthetic constraints. Most joints showed that the prosthesis with the wrist rotator
performed better than the standard prosthesis. The model was able to perform bilateral tasks and maintain reasonable joint angles. It was shown that, for the majority of trials, the inclusion of the wrist rotator resulted in a reduced degree of compensation.

The model shows promise for being able to assist with predicting movement based on prosthetic configurations. As prostheses advance, task performance and degrees of freedom are likely to increase. Modeling tasks will help to make the best use of current technologies in the development, selection, and functional prognoses of new prostheses. With further development the usefulness of kinematic modeling in the simulation of activities of daily living will expand the study of compensatory motion. Study into compensatory motion will help develop more advanced prostheses, and help improve the quality of life of the individuals who utilize arm prostheses.
List of References


Appendices
Appendix 1: Matlab Code

A.1.1 Graphical User Interface ‘ModelGUI.m’

\textbf{function} varargout = GUI_export
\%initializes the graphical user interface
\n\%Generate window (figure)
\textbf{h1} = \textbf{figure}(\texttt{\textbf{Name},'Upper Body Kinematics GUI'});
\n\%Load model with robot parameters (scripts will not work in functions)
\%Script for Kinematic Model
\textbf{path}(\textbf{path,'C:\Program Files\MATLAB\R2007b\robot'});
\n\%add path for trajectory files
\textbf{path}(\textbf{path,'trajectories'});
\n\%add path for configuration files
\textbf{path}(\textbf{path,'configurations'});
\n\textbf{trajdir} = \textbf{dir}(\texttt{trajectories/*\_mat});
\textbf{trajlist} = \textbf{struct2cell}(\textbf{trajdir});
\textbf{trajname} = \textbf{trajlist}(1,:);
\n\textbf{confdir} = \textbf{dir}(\texttt{configurations/*\_mat});
\textbf{conflist} = \textbf{struct2cell}(\textbf{confdir});
\textbf{confname} = \textbf{conflist}(1,:);
\n\textbf{Dm} = \textbf{eye}(15);
\n\%Generate Links
\textbf{h} = 2;
\textbf{L1} = \textbf{link}([-\pi/2 0 0 0 1]);
\textbf{L2} = \textbf{link}([-\pi/2 0 -\pi/2 0 1]);
\textbf{L3} = \textbf{link}(0 0 0 0 1));
\textbf{L4} = \textbf{link}([\pi/2 0 0 0 0]);
\textbf{L5} = \textbf{link}([\pi/2 0 0 0 0]);
\textbf{L6} = \textbf{link}(0 0 0 0.288*h 0));
\textbf{L7} = \textbf{link}([\pi/2 0 0 0 0]);
\textbf{L8} = \textbf{link}(0 0.129*h 0 0 0 0));
\textbf{L9} = \textbf{link}((\pi/2 0 0 0 0));
\textbf{L10} = \textbf{link}([\pi/2 0 0 0 0]);
\textbf{L11} = \textbf{link}([\pi/2 0 0 0 186*h 0]));
\textbf{L12} = \textbf{link}([\pi/2 0 0 0 0]));
\textbf{L13} = \textbf{link}([\pi/2 0 0 0.146*h 0]));
\textbf{L14} = \textbf{link}([\pi/2 0 0 0 0]));
\textbf{L15} = \textbf{link}([\pi/2 0 0 0 0]));
\n\% Left Arm ROM Descriptors
\%L1 Translation of the hip joint in the Z direction
\%L2 Translation of the hip joint in the Y direction
\%L3 Translation of the hip joint in the X direction
\%L4 Torso Extension (+) / Flexion (-)
Appendix 1 (continued)

%L5 Torso Lateral Flexion Right (+) / Left (-)
%L6 Torso Rotation Left (+) / Right (-)
%L7 Shoulder Adduction (+) / Abduction (-)
%L8 Shoulder Depression (+) / Elevation (-)
%L9 Upper Arm Adduction (+) / Abduction (-)
%L10 Upper Arm Extension (+) / Flexion (-)
%L11 Upper Arm Medial Rotation Inward (+)/Outward (-)
%L12 Elbow Extension (+) / Flexion (-)
%L13 Forearm Pronation (+) / Supination (-)
%L14 Wrist Flexion (+) / Extension (-)
%L15 Wrist Adduction (+) / Abduction (-)

% Right Arm ROM Descriptors
%L1 Translation of the hip joint in the Z direction
%L2 Translation of the hip joint in the Y direction
%L3 Translation of the hip joint in the X direction
%L4 Torso Extension (+) / Flexion (-)
%L5 Torso Lateral Flexion Right (+) / Left (-)
%L6 Torso Rotation Left (+) / Right (-)
%L7 Shoulder Abduction (+) / Adduction (-)
%L8 Shoulder Depression (+) / Elevation (-)
%L9 Upper Arm Adduction (+) / Abduction (-)
%L10 Upper Arm Extension (+) / Extension (-)
%L11 Upper Arm Lateral Rotation Outward (+)/ Medial Inward (-)
%L12 Elbow Flexion (+) / Extension (-)
%L13 Forearm Supination (+) / Pronation (-)
%L14 Wrist Extension (+) / Flexion (-)
%L15 Wrist Adduction (+) / Abduction (-)

%Generate Robot
UpperBody = robot({L1, L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15});

%Generate Initial Joint Angles
%Right Configuration
qR0 = [0 0 0, 0 pi/2 0, pi pi, pi/2 pi/2 pi, -pi/2 -pi/2, pi/2 0];
%Left Configuration
qL0 = [0 0 0, 0 pi/2 0, 0 pi, pi/2 pi/2 pi, pi/2 pi/2, pi/2 0];

%%%This section contains the Initializations for the various texts and buttons.

%headertext
header = uicontrol(h1,'units','characters','style','text','FontSize',12,'FontWeight','bold','Position',[24 30 60 2], 'String','Upper Body Kinematic Model');

%Trajectory popupmenu
text = uicontrol(h1,'units','characters','style','text','Position',[10 28 23 1], 'String','Select a Trajectory: ');
trajmenu = uicontrol(h1,'units','characters','style','popupmenu','BackgroundColor',[1 1 1], 'Position',[10 26.5 23 1.5], 'String',trajname);

%Configuration popupmenu
text = uicontrol(h1,'units','characters','style','text','Position',[10 24 23 1], 'String','Select a Configuration: ');
Appendix 1 (continued)

confmenu = uicontrol(h1, 'Units', 'characters', 'Style', 'popupmenu', 'BackgroundColor', [1 1 1], 'Position', [10 22.5 23 1.5], 'String', confname);

%Checkbox for manual weights
manual = uicontrol(h1, 'Units', 'characters', 'Position', [10 20 40 1.5], 'String', 'Manually Edit Weighting Parameters', 'Style', 'checkbox', 'Callback', @manual_callback);

function manual_callback(hObject, eventdata)

Dm = eye(15);

%First row of weights
el1 = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [18 18 6 1.5], 'String', '1.0', 'Style', 'edit', 'callback', @el1_Callback);

el2 = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [18 16 6 1.5], 'String', '1.0', 'Style', 'edit', 'callback', @el2_Callback);

el3 = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [18 14 6 1.5], 'String', '1.0', 'Style', 'edit', 'callback', @el3_Callback);

el4 = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [18 12 6 1.5], 'String', '1.0', 'Style', 'edit', 'callback', @el4_Callback);

el5 = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [18 10 6 1.5], 'String', '1.0', 'Style', 'edit', 'callback', @el5_Callback);

el6 = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [18 8 6 1.5], 'String', '1.0', 'Style', 'edit', 'callback', @el6_Callback);

el7 = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [18 6 6 1.5], 'String', '1.0', 'Style', 'edit', 'callback', @el7_Callback);

el8 = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [18 4 6 1.5], 'String', '1.0', 'Style', 'edit', 'callback', @el8_Callback);

%Second row of weights
el9 = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [38 18 6 1.5], 'String', '1.0', 'Style', 'edit', 'callback', @el9_Callback);

el10 = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [38 16 6 1.5], 'String', '1.0', 'Style', 'edit', 'callback', @el10_Callback);

el11 = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [38 14 6 1.5], 'String', '1.0', 'Style', 'edit', 'callback', @el11_Callback);

el12 = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [38 12 6 1.5], 'String', '1.0', 'Style', 'edit', 'callback', @el12_Callback);

el13 = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [38 10 6 1.5], 'String', '1.0', 'Style', 'edit', 'callback', @el13_Callback);

el14 = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [38 8 6 1.5], 'String', '1.0', 'Style', 'edit', 'callback', @el14_Callback);

el15 = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [38 6 6 1.5], 'String', '1.0', 'Style', 'edit', 'callback', @el15_Callback);

%fist row weight labels
tl1 = uicontrol(h1, 'Units', 'characters', 'Position', [10 18 6 1.5], 'String', 'L1 =', 'Style', 'text');

tl2 = uicontrol(h1, 'Units', 'characters', 'Position', [10 16 6 1.5], 'String', 'L2 =', 'Style', 'text');

tl3 = uicontrol(h1, 'Units', 'characters', 'Position', [10 14 6 1.5], 'String', 'L3 =', 'Style', 'text');

tl4 = uicontrol(h1, 'Units', 'characters', 'Position', [10 12 6 1.5], 'String', 'L4 =', 'Style', 'text');

tl5 = uicontrol(h1, 'Units', 'characters', 'Position', [10 10 6 1.5], 'String', 'L5 =', 'Style', 'text');

tl6 = uicontrol(h1, 'Units', 'characters', 'Position', [10 8 6 1.5], 'String', 'L6 =', 'Style', 'text');

tl7 = uicontrol(h1, 'Units', 'characters', 'Position', [10 6 6 1.5], 'String', 'L7 =', 'Style', 'text');
Appendix 1 (continued)

```matlab
tl8 = uicontrol(h1,'Units','characters','Position',[10 4 6 1.5],'String','L8 =','Style','text');
%second row weight labels
tl9 = uicontrol(h1,'Units','characters','Position',[30 18 6 1.5],'String','L9 =','Style','text');
tl10 = uicontrol(h1,'Units','characters','Position',[30 16 6 1.5],'String','L10 =','Style','text');
tl11 = uicontrol(h1,'Units','characters','Position',[30 14 6 1.5],'String','L11 =','Style','text');
tl12 = uicontrol(h1,'Units','characters','Position',[30 12 6 1.5],'String','L12 =','Style','text');
tl13 = uicontrol(h1,'Units','characters','Position',[30 10 6 1.5],'String','L13 =','Style','text');
tl14 = uicontrol(h1,'Units','characters','Position',[30 8 6 1.5],'String','L14 =','Style','text');
tl15 = uicontrol(h1,'Units','characters','Position',[30 6 6 1.5],'String','L15 =','Style','text');

function el1_Callback(hObject,eventdata)
  Dm(1,1) = str2double(get(hObject,'string'));
end
function el2_Callback(hObject,eventdata)
  Dm(2,2) = str2double(get(hObject,'string'));
end
function el3_Callback(hObject,eventdata)
  Dm(3,3) = str2double(get(hObject,'string'));
end
function el4_Callback(hObject,eventdata)
  Dm(4,4) = str2double(get(hObject,'string'));
end
function el5_Callback(hObject,eventdata)
  Dm(5,5) = str2double(get(hObject,'string'));
end
function el6_Callback(hObject,eventdata)
  Dm(6,6) = str2double(get(hObject,'string'));
end
function el7_Callback(hObject,eventdata)
  Dm(7,7) = str2double(get(hObject,'string'));
end
function el8_Callback(hObject,eventdata)
  Dm(8,8) = str2double(get(hObject,'string'));
end
function el9_Callback(hObject,eventdata)
  Dm(9,9) = str2double(get(hObject,'string'));
end
function el10_Callback(hObject,eventdata)
  Dm(10,10) = str2double(get(hObject,'string'));
end
function el11_Callback(hObject,eventdata)
  Dm(11,11) = str2double(get(hObject,'string'));
end
function el12_Callback(hObject,eventdata)
  Dm(12,12) = str2double(get(hObject,'string'));
end
function el13_Callback(hObject,eventdata)
  Dm(13,13) = str2double(get(hObject,'string'));
end
function el14_Callback(hObject,eventdata)
  Dm(14,14) = str2double(get(hObject,'string'));
end
function el15_Callback(hObject,eventdata)
```

47
Appendix 1 (continued)

Dm(15,15) = str2double(get(hObject, 'string'));
end
end

% Run Inverse Kinematics Button
runbtn = uicontrol(h1, 'Units', 'characters', 'Position', [70 25 30 2], 'String', 'Run Inverse Kinematics', 'Callback', @runbtn_callback);

% Run callback function
function [TR, TL, JR, JL] = runbtn_callback(hObject, eventdata)

Dm;
trajfile = get(trajmenu, 'value');
conffile = get(confmenu, 'value');
load(trajdir(trajfile).name);
load(confdir(conffile).name);
Dm(13,13)=D(13,13);
Dm(14,14)=D(14,14);
Dm(15,15)=D(15,15);
D = Dm;
[JR, JL] = Dkine(UpperBody, TR, TL, qR0, qL0, D);

% Buttons to appear after model is run
% Plot Results Button
plotbtn = uicontrol(h1, 'Units', 'characters', 'Position', [70 20 30 2], 'String', 'Show Results in Wireframe', 'Callback', @plotbtn_callback);

% Open Model in VR Simulation
vrbtn = uicontrol(h1, 'Units', 'characters', 'Position', [70 15 30 2], 'String', 'Show Results in VR Model', 'Callback', @vrbtn_callback);

% Select Menu for Joint Angle Plots
jplotmenu = uicontrol(h1, 'Units', 'characters', 'BackgroundColor', [1 1 1], 'Position', [65 9.85 8 2], 'String', {'L1'; 'L2'; 'L3'; 'L4'; 'L5'; 'L6'; 'L7'; 'L8'; 'L9'; 'L10'; 'L11'; 'L12'; 'L13'; 'L14'; 'L15'}, 'Style', 'popupmenu', 'Value', 1);

% Button for Joint Angle Plots
jplotbtn = uicontrol(h1, 'Units', 'characters', 'Position', [75 10 25 2], 'String', 'Plot Selected Joint Angle', 'Callback', @jplotbtn_callback);

% Save button
savebtn = uicontrol(h1, 'Units', 'characters', 'Position', [75 5 15 2], 'String', 'Save Results', 'Callback', @savebtn_callback);

function plotbtn_callback(hObject, eventdata)
figure ('name', 'Right Arm Plot');
plot(UpperBody, JR);
figure ('name', 'Left Arm Plot');
plot(UpperBody, JL);
end

end % plot end

48
Appendix 1 (continued)

```
function vrbtn_callback(hObject, eventdata)

    slider = uicontrol(h1,'units','characters','Style','slider','Max',100,'Min',1,'Value',1,'SliderStep',[.01 .10],'Position',[70 14 30 1],'Callback',@slider_callback);
    myworld = vrworld('VRModel.wrl');
    open(myworld);
    view(myworld);
    function slider_callback(hObject, eventdata)
        step = round(get(slider,'value'));
        VRmodel(myworld,JL(step,:),JR(step,:))
    end
    recbtn = uicontrol(h1,'Units','characters','Position',[60 15 10 2],'String','Run','Callback',@recbtn_callback);
    function recbtn_callback(hObject, eventdata)
        pause(5)
        for i=1:100
            VRmodel(myworld,JL(i,:),JR(i,:))
            pause(0.05)
        end
    end

function jplotbtn_callback(hObject, eventdata)
    joint = get(jplotmenu,'value');
    figure;
    plot(JR(:,joint))
    figure;
    plot(JL(:,joint))
end %Jplot

function recbtn_callback(hObject, eventdata)
    ui/save;
end %Save

end %GUI function end

A.1.2 Weighted Inverse Kinematic Function ‘Dkine.m’

% Modified from robotics toolkit 'ikine' developed by Peter I. Corke

function [qrt,qlt] = Dkine(robot, tr, tl, qr, ql, D)
    n = robot.n;
    qr = qr(:);
    ql = ql(:);
    %Dr controls restriction of hip movement to the right arm.
    Dr = D;
    Dr(1,1) = 0;
    Dr(2,2) = 0;
    Dr(3,3) = 0;
```
Appendix 1 (continued)

\[
\begin{align*}
\text{Dr}(4,4) &= 0; \\
\text{Dr}(5,5) &= 0; \\
\text{Dr}(6,6) &= 0; \\
\text{Dr}(13,13) &= 1; \\
\text{Dr}(14,14) &= 1; \\
\text{Dr}(15,15) &= 1; \\
np &= \text{size}(\text{tr}, 3); \\
\text{qlt} &= []; \\
\text{qrt} &= []; \\
\text{for } i=1:np \\
\text{TL} &= \text{tr}(:,:,i); \\
\text{el} &= \text{tr2diff}\text{(fkine(robot, ql'), TL)}; \\
\text{dql} &= \text{D} \ast ( \text{pinv} \text{(jacob0(robot, ql) } * \text{D} ) \ast \text{el}); \\
\text{ql} &= \text{ql} + \text{dql}; \\
\text{qlt} &= [\text{qlt}; \text{ql'}]; \\
\text{qr}(1) &= \text{ql}(1); \\
\text{qr}(2) &= \text{ql}(2); \\
\text{qr}(3) &= \text{ql}(3); \\
\text{qr}(4) &= \text{ql}(4); \\
\text{qr}(5) &= \text{ql}(5); \\
\text{qr}(6) &= \text{ql}(6); \\
\text{TR} &= \text{tr}(:,:,i); \\
\text{er} &= \text{tr2diff}\text{(fkine(robot, qr'), TR)}; \\
\text{dqr} &= \text{Dr} \ast ( \text{pinv} \text{(jacob0(robot, qr) } * \text{Dr} ) \ast \text{er}); \\
\text{qr} &= \text{qr} + \text{dqr}; \\
\text{qrt} &= [\text{qrt}; \text{qr'}]; \\
\text{end} \\
\text{end}
\end{align*}
\]

A.1.3 Virtual Reality Initialization ‘VRModel.m’

\begin{verbatim}
function VRmodel(myworld,qL,qR)

% L1 Translation of the hip joint in the Z direction
TZ = qL(1)*5;

% L2 Translation of the hip joint in the Y direction
TY = qL(2)*5;

% L3 Translation of the hip joint in the X direction
TX = qL(3)*5;

myworld.L4.translation = [TX TZ TY];

% L4 Torso Extension (+) / Flexion (-) (initial = 0)
myworld.L4.rotation = [-1 0 0 qL(4)];

% L5 Torso Lateral Flexion Right (+) / Left (-) (initial = 0)
myworld.L5.rotation = [0 0 1 qL(5)-pi/2];

% L6 Torso Rotation Left (+) / Right (-) (initial = 0)
myworld.L6.rotation = [0 1 0 qL(6)];

% Left Arm ROM Descriptors
\end{verbatim}
Appendix 1 (continued)

%L7 Shoulder Adduction (+) / Abduction (-) (initial = 0)
myworld.L7.rotation = [0 1 0 qL(7)];
%L8 Shoulder Depression (+) / Elevation (-) (initial = 90)
myworld.L8.rotation = [0 0 -1 qL(8)-pi/2];

%L9 Upper Arm Adduction (+) / Abduction (-) (initial = 90)
myworld.L9.rotation = [0 0 -1 qL(9)];
%L10 Upper Arm Extension (+) / Flexion (-) (initial = 0)
myworld.L10.rotation = [-1 0 0 qL(10)-pi/2];
%L11 Upper Arm Rotation Inward (+)/Outward (-) (initial = 0)
myworld.L11.rotation = [0 1 0 qL(11)-pi];

%L12 Elbow Extension (+) / Flexion (-) (initial = -90)
myworld.L12.rotation = [-1 0 0 qL(12)-pi];
%L13 Forearm Pronation (+) / Supination (-) (initial = 90)
myworld.L13.rotation = [0 1 0 qL(13)];

%L14 Wrist Flexion (+) / Extension (-) (initial = 0)
myworld.L14.rotation = [1 0 0 qL(14)-pi/2];
%L15 Wrist Adduction (+) / Abduction (-) (initial = 0)
myworld.L15.rotation = [0 0 -1 qL(15)];

% Right Arm ROM Descriptors
%R7 Shoulder Abduction (+) / Adduction (-) (initial = 0)
myworld.R7.rotation = [0 1 0 qR(7)-pi];
%R8 Shoulder Depression (+) / Elevation (-) (initial = 90)
myworld.R8.rotation = [0 0 1 qR(8)-pi/2];

%R9 Upper Arm Adduction (+) / Abduction (-) (initial = 90)
myworld.R9.rotation = [0 0 1 qR(9)];
%R10 Upper Arm Flexion (+) / Extension (-) (initial = 0)
myworld.R10.rotation = [1 0 0 qR(10)-pi/2];
%R11 Upper Arm Rotation Lateral (+)/ Medial (-) (initial = 0)
myworld.R11.rotation = [0 1 0 qR(11)-pi];

%R12 Elbow Flexion (+) / Extension (-) (initial = 90)
myworld.R12.rotation = [1 0 0 qR(12)+pi];
%R13 Forearm Supination (+) / Pronation (-) (initial = 90)
myworld.R13.rotation = [0 1 0 qR(13)+pi];

%R14 Wrist Extension (+) / Flexion (-) Rotation (initial = 0)
myworld.R14.rotation = [1 0 0 qR(14)-pi/2];
%R15 Wrist Adduction (+) / Abduction (-) Rotation (initial = 0)
myworld.R15.rotation = [0 0 -1 qR(15)];

A.1.4 Opening a Door Trajectory Generator `Door.m`

% Opening a Door Traj (Right Hinged)
h=2;
% Establishes the initial joint angles for the model
TR1 = [0 0 1 -0.355; 1 0 0 0.71+.400; 0 1 0 (0.930-0.530*h); 0 0 0 1];
% Initial end effector position

51
Appendix 1 (continued)

\[ TR30 = \begin{bmatrix} 0 \ -1 \ 0 \ -0.355; 1 \ 0 \ 0.71+.400; 0 \ 0 \ 1 \ (0.930-0.530*h); 0 \ 0 \ 0 \ 1 \end{bmatrix}; \]

% End effector position at step 30 (knob turned 90deg)
Step1 = ctraj(TR1, TR30, 30);
% Interperlates the steps between 0 and 30
qi = ((1:70)-1)/(70-1)*pi/2;
% Creates a 1x70 array of angle q from 0 to 90deg
Step2 = zeros(4,4,0);
% Creates an empty 4x4x0 array
for q=qi
    Step2 = cat(3,Step2,
        \begin{bmatrix} \sin(q) & -\cos(q) & 0 & 0.355+2*0.355*(1-\cos(q)) \ 0 & \cos(q) & \sin(q) & 0.710*(1-\sin(q))+.4 \ 0 & 0 & 1 & (0.930-0.530*h) \ 0 & 0 & 0 & 1 \end{bmatrix});
end
% Creates an array of homogeneous transforms using the array qi. (door opening 90deg)
TL = cat(3,Step1,Step2);

TL1 = \begin{bmatrix} 0 \ -1 \ 0 \ -0.1290*h; 1 \ 0 \ 0.2540*h; 0 \ 0 \ 1 \ 0.102*h; 0 \ 0 \ 0 \ 1 \end{bmatrix};
TL70 = \begin{bmatrix} 1 \ 0 \ 0.0540*h; 0 \ 1 \ 0.1540*h; 0 \ -0 \ 1.4*h; 0 \ 0 \ 0 \ 1 \end{bmatrix};
TL100 = \begin{bmatrix} 1 \ 0 \ 0.0540*h; 0.01 & .99 & 0.01 & 0.4*h; 0 \ 0 \ 0 \ 1 \end{bmatrix};
Step1 = ctraj(TL1,TL70,70);
Step2 = ctraj(TL70,TL100,30);
TR0 = \begin{bmatrix} 0 \ -1 \ 0 \ 0.129*h; 1 \ 0 \ 0.254*h; 0 \ 0 \ 1 \ 0.102*h; 0 \ 0 \ 0 \ 1 \end{bmatrix};
TR = ctraj(TR0,TR0,100);

TL1 = \begin{bmatrix} 0 \ -1 \ 0 \ -0.250; 1 \ 0 \ 0.254*h+.15; 0 \ 0 \ 1 \ 1-0.530*h; 0 \ 0 \ 0 \ 1 \end{bmatrix};
TL33 = \begin{bmatrix} 1 \ 0 \ 0 \ 0.0540*h; 0 \ 1 \ 0.1540*h; 0 \ -0 \ 1.4*h; 0 \ 0 \ 0 \ 1 \end{bmatrix};
TL67 = \begin{bmatrix} 1 \ 0 \ 0.0540*h; 0.01 & .99 & 0.01 & 0.4*h; 0 \ 0 \ 0 \ 1 \end{bmatrix};
TL100 = \begin{bmatrix} 1 \ 0 \ 0.0540*h; 0.01 & .99 & 0.01 & 0.4*h; 0 \ 0 \ 0 \ 1 \end{bmatrix};
Step1 = ctraj(TL1,TL33,33);
Step2 = ctraj(TL33,TL67,34);
Step3 = ctraj(TL67,TL100,33);
TR = cat(3,Step1, Step2, Step3);

TL1 = \begin{bmatrix} 0 \ -1 \ 0 \ -0.250; 1 \ 0 \ 0.254*h+.15; 0 \ 0 \ 1 \ 1-0.530*h; 0 \ 0 \ 0 \ 1 \end{bmatrix};
TL33 = \begin{bmatrix} 1 \ 0 \ 0 \ 0.0540*h; 0 \ 1 \ 0.1540*h; 0 \ -0 \ 1.4*h; 0 \ 0 \ 0 \ 1 \end{bmatrix};
TL67 = \begin{bmatrix} 1 \ 0 \ 0.0540*h; 0.01 & .99 & 0.01 & 0.4*h; 0 \ 0 \ 0 \ 1 \end{bmatrix};
TL100 = \begin{bmatrix} 1 \ 0 \ 0.0540*h; 0.01 & .99 & 0.01 & 0.4*h; 0 \ 0 \ 0 \ 1 \end{bmatrix};
LStep1 = ctraj(TL1,TL33,33);
Appendix 1 (continued)

LStep2 = ctraj(TL33,TL67,34);
LStep3 = ctraj(TL67,TL100,33);
TL = cat(3,LStep1, LStep2, LStep3);

A.1.7 Turning a Steering Wheel Trajectory Generator ‘Turn.m’

%Generates trajectories for turning a wheel task
h = 2;
qi = (((1:25)-1)/(25-1))*pi/2;
% Creates a 1x25 array of angle q from 0 to 90deg
RStep1 = zeros(4,4,0);
RStep2 = zeros(4,4,0);
RStep3 = zeros(4,4,0);
RStep4 = zeros(4,4,0);
LStep1 = zeros(4,4,0);
LStep2 = zeros(4,4,0);
LStep3 = zeros(4,4,0);
LStep4 = zeros(4,4,0);
% Creates an empty 4x4x0 array
for q=qi
    RStep1 = cat(3,RStep1,[0 -cos(q) sin(q) 0.25*cos(q); 1 0 0 0.254*h; 0 sin(q) cos(q) 0.102*h-0.250*sin(q); 0 0 0 1]);
    LStep1 = cat(3,LStep1,[0 -cos(q) sin(q) -0.25*cos(q); 1 0 0 0.254*h; 0 sin(q) cos(q) 0.102*h+0.250*sin(q); 0 0 0 1]);
    RStep2 = cat(3,RStep2,[0 -cos(pi/2-q) sin(pi/2-q) 0.25*cos(pi/2-q); 1 0 0 0.254*h; 0 sin(pi/2-q) cos(pi/2-q) 0.102*h-0.250*sin(pi/2-q); 0 0 0 1]);
    LStep2 = cat(3,LStep2,[0 -cos(pi/2-q) sin(pi/2-q) -0.25*cos(pi/2-q); 1 0 0 0.254*h; 0 sin(pi/2-q) cos(pi/2-q) 0.102*h+0.250*sin(pi/2-q); 0 0 0 1]);
    RStep3 = cat(3,RStep3,[0 -cos(q-pi) sin(q-pi) 0.25*cos(q-pi); 1 0 0 0.254*h; 0 sin(q-pi) cos(q-pi) 0.102*h-0.250*sin(q-pi); 0 0 0 1]);
    LStep3 = cat(3,LStep3,[0 -cos(q-pi) sin(q-pi) -0.25*cos(q-pi); 1 0 0 0.254*h; 0 sin(q-pi) cos(q-pi) 0.102*h+0.250*sin(q-pi); 0 0 0 1]);
    RStep4 = cat(3,RStep4,[0 -cos(q-pi) sin(q-pi) 0.25*cos(q-pi); 1 0 0 0.254*h; 0 sin(q-pi) cos(q-pi) 0.102*h-0.250*sin(q-pi); 0 0 0 1]);
    LStep4 = cat(3,LStep4,[0 -cos(q-pi) sin(q-pi) -0.25*cos(q-pi); 1 0 0 0.254*h; 0 sin(q-pi) cos(q-pi) 0.102*h+0.250*sin(q-pi); 0 0 0 1]);
end
% Generates steps

% Merge steps
TR = cat(3,RStep1,RStep2,RStep3,RStep4);
TL = cat(3,LStep1,LStep2,LStep3,LStep4);