5-5-2012

Determining Upper Extremity Posture Using a Simplified Marker Configuration for Biomechanical Risk Evaluation during Tool Use

Tarek A. Tantawy
University of Connecticut, tarek.tantawy@engr.uconn.edu

Determining Upper Extremity Posture Using a Simplified Marker Configuration for Biomechanical Risk Evaluation during Tool Use

Tarek Ahmed Tantawy

B.S., University of Connecticut, 2009

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Masters of Science

at the

University of Connecticut

2012

APPROVAL PAGE

Masters of Science Thesis

Determining Upper Extremity Posture Using a Simplified Marker Configuration for Biomechanical Risk Evaluation during Tool Use

Presented by

Tarek Ahmed Tantawy, B.S.

Major Advisor _____

Donald R. Peterson, PhD., M.S.

Associate Advisor _____

Angela Kueck, MD

Associate Advisor _____

John C. Bennett Jr., PhD., M.S.

Acknowledgements

I would like to thank my major advisor, Dr. Peterson, for the opportunity, guidance and support along this process. I would also like to thank my associate advisors, Dr. Bennett and Dr Kueck, for their assistance.

I would also like to acknowledge the other students at the Biodynamics Lab, Shane Tornifoglio, Katelyn Burkhart, Beau Knapp, Takafumi Asaki and Maria Qadri, for their continued help and encouragement throughout my project.

I would like to express my particular gratitude to Eric Bernstein and Drew Seils for all of their tremendous efforts working alongside me during this process.

Finally, I would like to thank all of my family and friends for their encouragement and support.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

ABSTRACT

Determining Upper Extremity Posture Using a Simplified Marker Configuration for Biomechanical Risk Evaluation during Tool Use

By

Tarek Ahmed Tantawy

Representing upper extremity posture has posed a great challenge in the field of analytical biomechanics due to their great freedom of motion, which allows for large rotations in multiple directions. The Euler and quaternion methods, which are two commonly used methods for describing joint angles and body segment postures and movement, have great limitations when dealing with upper extremities. The planar projection method has also been used to represent joint angles and body segment postures and movement. The advantages of this method over other common methods are that it yields rotations with anatomical meaning that make it easy to understand from a clinical standpoint, it is not affected by Gimbal lock phenomenon, and is more accurate at large multi-axis rotations, which make it a good candidate for representing upper extremity posture. The disadvantages of this method are that it has some accuracy limitations when dealing with multi-axis rotations beyond 60 degrees, with complex rotation of the body

structures inside of soft tissue, and has error associated with it when projecting onto global planes. This thesis investigated a modified planar projection approach using a simplified opto-electronic marker configuration to improve the accuracy when calculating upper extremity orientations. One modification of the method was that anatomical planes were calculated by defining orthogonal axes at the chest and rotating the axes to account for the slope of the chest. By defining the anatomical axes using anatomical landmarks of the upper torso, the body can move freely without disturbing the anatomical plane calculations. Projecting onto planes that are defined locally for different body segments rather than global segments, eliminated the angle errors previously associated with the planar projection method. This modified method was tested using human and mechanical tests and was applied to data pertaining to laparoscopic surgical hand tool use. The biomechanical risk exposure of the tasks was quantified from the postural results using a threshold technique and a novel approach to the Rapid Upper Limb Assessment.

1 Background and Significance

Opto-electronic motion capture (OEMC) is a powerful tool in analytic biomechanics. OEMC systems have the ability to track the three-dimensional position of infrared-reflective markers. Methods have been developed to determine anatomical orientations and postures using marker configurations and calculation methods. When calculating the orientation of a rigid body, a unit orthogonal axis is defined using three markers whose positioning is defined by anatomical landmarks (Peterson and Bronzino, 2008). Once the orthogonal axis is defined, there are several calculation methods developed to define rigid body orientation. The planar projection, Euler/Cardan angle, attitude vector, and screw axis methods are used to represent anatomical orientation (Peterson, 1999; Chao, 1980; Tupling and Pierrynowski, 1987; Woltring, 1994; Cheng, 2000). All of these methods use a 3 x 3 rotation matrix to define the orientation of the three unit coordinate axes of the anatomical body segment. The axes are defined in vector notation in relation to the global coordinate system.

The Euler (or Cardan) method converts this rotation matrix into three sequential rotations. The three rotations can be any combination of x, y and z. Euler rotations can have one rotation axis repeat, while Cardan angles have three rotations about unique axes. These three rotations are defined in terms of anatomical orientation of the body segments as flexion/extension, abduction/adduction and medial/lateral rotation. These allow for a simplified understanding for the clinical use of the results. One major issue with the Euler rotation method is that errors occur when there are multiple large rotations of the segment (Chao, 1980; Tupling and Pierrynowski, 1987; Coates, 2007). The other major issue with the Euler rotation method is the occurrence of Gimbal lock phenomenon

that occurs when a rotation about one axis causes the other two axes to align, which yields sequential rotation about the axes. Mathematically, this is associated with the sine and cosine characteristics at 90 degrees. In Biomechanics, the Euler rotation method has been accepted as the best technique for defining orientation of lower extremities where large multiple rotations and Gimbal lock parameters during gait are inherently avoided. For upper extremity postures with a greater amount of rotational freedom, there is a greater amount of error in posture calculation using the Euler approach (Chao, 1980; Tupling and Pierrynowski, 1987; Coates, 2007).

The attitude vector method uses quaternion mathematics to convert the rotation matrix into anatomical rotations, which involves defining a single rotation about an axis and is converted to three Euler angles. The advantages of this method include avoidance of Gimbal lock and erroneous joint orientations (Woltring, 1994). This method involves difficult calculations and interpretation and has also been found to not have accurate anatomical meaning (Dura et al., 2011).

The planar projection method involves projecting the unit coordinate axes of the body segments onto anatomical planes, in order to calculate their orientation with respect to anatomical axes of other body segments. This has been used to define complex upper extremity movement (Peterson, 1999). With the planar projection method, errors occur when the reference vector on the projection plane is rotated away from the global axes and the value of the rotation increases. The error that occurs when this happens can be quantified and calculated along with the angle values (Peterson, 1999). This method avoids Gimbal lock phenomenon, has known angle errors, and is easy to interpret for clinical understanding of the results.

1.1 Anatomical Posture Calculation of Upper Extremities

Chao (1980) first implemented a method of anatomical segment orientation calculation from a rotational matrix when he introduced a triaxial goniometer for measuring three-dimensional angular motions by use of Euler angles. Chao's methods provided unique motion patterns of the joint from one position to another. The Euler angle method was found to be beneficial because it defines rotations about axes that are, by definition, the anatomical axes; however, the work showed that different combinations of x, y and z yield different results. In addition to these errors, Chao found that the Euler method is limited due to errors in calculation that arise from the Gimbal lock phenomenon and, when there are rotations greater than 30 degrees in multiple directions, cross-talk can occur. The results of the study suggested that the Euler method is beneficial in calculating anatomical orientation of body segments below 30 degrees, which is useful for lower extremity calculation, but less useful for upper extremity calculation.

Tupling and Pierrynowski (1987) proposed a method for tracking any rigid human body segment using Cardan angle calculations. The results of this study support Chao's (1980) claims that different rotation sequences yield different results and that the calculations become inaccurate beyond 30 degrees in multiple directions. Tupling and Pierrynowski also suggested that different rotations methods should be used for different movement patterns and that the use of Cardan angles is good because it describes motion using anatomical movement patterns. Both the study by Tupling and Pierrynowski and by Chao found Euler or Cardan angles to be beneficial for representing anatomical postures below 30 degrees in multiple directions.

Woltring (1994) proposed a new convention for unambiguous and easily interpretable three-dimensional joint angles based on attitude vector in Euler's theorem. The attitude vector method defines a segment's orientation as one rotation about one vector. The three components of this vector along with the rotation magnitude make up a quaternion that can be broken down into three rotations about anatomical axes that represent flexion/extension, abduction/adduction and medial/lateral rotation. An advantage of the quaternion method is that it does away with Gimbal lock issues; however, Woltring showed that Cardan angles have issues at great deviations due to cross-talk and suggested that the non-vectorial behavior of three-dimensional rotations are the reason why the Euler sequences are limited for use on the upper extremities.

Dura et al. (2011) investigated the advantages of attitude vector and Euler angle methods by observing the influence of different Euler angles versus attitude vectors for hip, knee, and ankle. Results of the study indicated that, while the attitude vector method had mathematical advantages (no Gimbal lock), its results did not have accurate anatomical meaning.

Peterson (1999) developed a vector-based planar projection method to calculate distal upper extremity posture and movement. The method used by Peterson defines vectors for each distal upper extremity, projects them onto a global plane, and calculates the angle between a moving vector and a reference vector. This method assumed that the reference vector did not rotate relative to the global plane and, if it did, the error associated with the angle calculations was defined by Peterson. The angle error is found to depend on the magnitude of the rotation and the calculated angle and is quantified in a table that is used to look up the error based on those two factors (see Table 1.1). The

planar projection method is beneficial because it does not have Gimbal lock issues and closely follows and isolates the anatomical rotation conventions (i.e. flexion/extension, abduction adduction and medial/lateral rotation).

Table 1.1: Angle error associated with projection angles as reference vector moves away from global axes (Peterson, 1999)

Projection Angles: Rotation Angle Between -j and Reference Vector	Projection Angle Between Reference Vector and Moving Vector							
	10	20	30	40	50	60	70	80
10	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0
30	1	1	2	2	2	2	1	1
40	2	3	4	4	4	3	3	1
50	5	10	12	13	12	10	7	4
60	9	16	19	19	17	14	10	5
70	17	27	29	28	24	19	13	7
80	35	44	43	38	32	24	16	8

Peterson and Cherniack (2001) used the planar projection method and angle error method to calculate the orientation and movement of upper extremities and a hammer during a hammering task. Using a simplified marker system, the planar projection method was capable of calculating upper extremity postures during the hammering movements but lacked the investigation of calculating rotations of the shoulder.

1.2 Planar Vector Projection Calculation

The planar projection method uses the OEMC markers to define axes of the moving body segment as well as a reference segment that can be used to project moving segment vectors onto for orientation calculation. In order to calculate the orthogonal axis of each segment, three markers are required. One of the three orthogonal axes is defined as the vector from one marker to another, as seen in Equation 1, where M_1 and M_2 are the markers that define the vector, $\overrightarrow{V_1}$. Another vector on the plane is also defined as seen in

Equation 2, where M_3 and M_2 are the markers that define the vector, $\overrightarrow{V_o}$. Both vectors are normalized, as shown in Equations 3 and 4, where $\overrightarrow{eV_1}$ and $\overrightarrow{eV_o}$ are the normalized vectors. The cross product of these two vectors will yield a vector that is normal to the plane, as seen in Equation 5, where $\overrightarrow{V_2}$ is the calculated vector. This is considered the second of the three orthogonal axes and this vector is also normalized, as shown in Equation 6, where $\overrightarrow{eV_2}$ is the normalized vector. The cross product of the two calculated orthogonal axes (i.e. $\overrightarrow{eV_1}$ and $\overrightarrow{eV_2}$) yields the third orthogonal axis, $\overrightarrow{eV_3}$, as seen in Equation 7. This vector does not need to be normalized because it is the cross product of two normalized orthogonal vectors.

$$\overrightarrow{V_1} = (M_{1x} - M_{2x})\hat{\imath} + \left(M_{1y} - M_{2y}\right)\hat{\jmath} + (M_{1z} - M_{2z})\hat{k} \quad (1)$$

$$\overrightarrow{V_o} = (M_{3x} - M_{2x})\hat{\imath} + \left(M_{3y} - M_{3y}\right)\hat{\jmath} + (M_{3z} - M_{2z})\hat{k} \quad (2)$$

$$\overrightarrow{eV_1} = \frac{\overrightarrow{V_1}}{|V_1|} = \frac{V_1\hat{\imath}+V_1\hat{\jmath}+V_1\hat{k}}{\sqrt{(V_1\,\hat{\imath})^2+(V_1\,\hat{\jmath})^2+(V_1\,\hat{k})^2}} \quad (3)$$

$$\overrightarrow{eV_o} = \frac{\overrightarrow{Vo}}{|Vo|} = \frac{Vo\,\hat{\imath}+Vo\,\hat{\jmath}+Vo\,\hat{k}}{\sqrt{(Vo\,\hat{\imath})^2+(Vo\,\hat{\jmath})^2+(Vo\,\hat{k})^2}} \quad (4)$$

$$\overrightarrow{V_2} = \overrightarrow{eV_1} \times \overrightarrow{eV_o} \quad (5)$$

$$\overrightarrow{eV_2} = \frac{\overrightarrow{V_2}}{|V_2|} = \frac{V_2\hat{\imath}+V_2\hat{\jmath}+V_2\hat{k}}{\sqrt{(V_2\,\hat{\imath})^2+(V_2\,\hat{\jmath})^2+(V_2\,\hat{k})^2}} \quad (6)$$

$$\overrightarrow{eV_3} = \overrightarrow{eV_1} \times \overrightarrow{eV_2} \quad (7)$$

Each body segment had its calculated coordinate axes projected onto an anatomical reference coordinate system (see Figure 3.14). More specifically, the calculated coordinate axes of the upper arm is projected onto the coordinate system

defined for the torso, while the coordinate axes of the hand and tool are projected onto the the forearm and hand, respectively.

After the orthogonal axes are calculated for a body segment and for the segment it will be projected onto, the vectors can be projected onto planes. In order to project a vector onto a plane, the cross product of the vector being projected and the vector normal to the plane is taken and then the cross product of the vector normal to the plane and the result of the previous calculation is taken. This calculation is shown in Equation 8, where $\overrightarrow{V_n}$ is the vector normal to the plane, $\overrightarrow{V_1}$ is the vector being projected, and $\overrightarrow{V_p}$ is the projection. The vector projection is normalized, as shown in Equation 9, where $\overrightarrow{eV_p}$ is the normalized vector.

$$\overrightarrow{V_p} = \overrightarrow{V_n} \times \left(\overrightarrow{V_1} \times \overrightarrow{V_n}\right) \tag{8}$$

$$\overrightarrow{eV_p} = \frac{\overrightarrow{V_p}}{|V_p|} = \frac{V_p\hat{\imath}+V_p\hat{\jmath}+V_p\hat{k}}{\sqrt{(V_p\hat{\imath})^2+(V_p\hat{\jmath})^2+(V_p\hat{k})^2}} \tag{9}$$

In order to represent an anatomical rotation, the angle between the vector projection and a corresponding axis of the plane it is projected on is calculated. This is calculated by taking the arccosine of the dot product of the two vectors and is shown in Equation 10, where $\overrightarrow{eV_{np}}$ is the normalized coordinate axis on the plane and θ is the angle between the vectors.

$$\theta = \cos^{-1}(\overrightarrow{eV_p} \cdot \overrightarrow{eV_{np}}) \tag{10}$$

1.3 Proposed Work

This study aims to further develop, test and implement a method for calculating upper extremity and tool orientation using Peterson's (1999) simplified marker system. In previous studies by Peterson (1999, 2001), the planes that the vectors are projected

onto are set by attempting to align the subject's anatomical segment planes with global coordinate planes. This study investigates a method of defining the subject's anatomical planes using a simplified marker system on the chest to allow for free movement of the subject without disturbing shoulder posture calculations. This study also investigates a method of calculating upper extremity rotations by only projecting onto anatomical segment planes. This method was tested using human and mechanical validation tests. It was also tested using an actual application on laparoscopic instrument use.

More specifically, it was used to assess the biomechanical risk associated with laparoscopic surgical tasks. Previous studies have implemented some form of OEMC in order to assess the biomechanical risk involved with laparoscopic tasks (Person et al., 2001; Lee and Park, 2008; Lee et al., 2010); however, none of these studies assessed biomechanical risk of upper extremity posture using planar projection and a thresholding technique.

2 Preliminary Studies

2.1 PATH Study

PATH is a subjective tool used for measuring biomechanical risk. The method operates on the assumption that biomechanical risk factors depend on the task performed. The risks associated with a task and the frequencies with which they occur are calculated (Buchholz et al., 1996). PATH was primarily used to assist with the experimental design of the application to the surgeons and the laparoscopic tools.

The PATH profiles generated intitially in preparation for the application of this solely looked at the upper extremity positions; specifically, every degree of freedom for the shoulder, wrist, forearm, and elbow of surgeons who were previously recorded on video. The arm that used the tool and the tool itself is presented in this study. The PATH procedure used in this study was to take a 'snapshot' observation of a laparoscopic surgical task every 30 seconds and record the postures and actions observed in that snapshot. A score of 1 or 0 is given for each posture based on whether the subject is, or is not, beyond a certain threshold.

It should be emphasized that PATH is a quantifiable subjective process that is not error free. The error associated with PATH is not standard and depends on user error, sampling rate and duration of measurement. To reduce the error, multiple observers review the same video recordings. As a limiting factor in this study, the amount of video attained for each tool was unbalanced, since there were more videos available of the manual device than the powered device. Another limiting factor was the variation in the

surgical tasks, especially since the surgical tasks observed were not standardized and the tools were used by different surgeons whom are carrying out different surgical tasks.

The PATH results presented in Table 2.1 are for the amount of time a surgical tool is used during the surgical procedures examined. The percentage given is the percentage of time the surgeon was in certain postures.

The results of the subjective PATH study are to be compared to the objective results of the OEMC planar projection methods.

Table 2.1: Results of PATH study

	Powered Device	Manual Device
POSTURE	**% of Time in Posture**	**% of Time in Posture**
SHOULDER		
Neutral (0^o to 60^o for All Positions)	97.1%	91.7%
Adduction $>60^o$	0.0%	0.0%
Abduction $>60^o$	2.9%	8.3%
Flexion $>60^o$	0.0%	0.0%
Extension $>60^o$	0.0%	0.0%
WRIST		
Neutral (0^o to 15^o for All Positions)	77.8%	58.3%
Extension $>15^o$	11.1%	16.7%
Flexion $>15^o$	0.0%	8.3%
Ulnar Deviation $>15^o$	11.1%	25.0%
Radial Deviation $>15^o$	0.0%	0.0%
FOREARM		
Neutral (0^o to 45^o)	86.1%	58.3%
Rotation > ± 45^o	13.9%	41.7%
ELBOW		
Neutral (60^o to 120^o for All Positions)	94.4%	91.7%
Flexion < 60^o	5.6%	0.0%
Extension $>120^o$	0.0%	8.3%

3 Methods

Many different pieces of equipment, calculation techniques, and testing methods are implemented in this study, in order to accomplish the goal of using the planar projection approach and a simplified marker system to quantify upper extremity posture.

3.1 OEMC System

The 24 opto-electronic motion capture cameras used are Natural Point OptiTrack V100R2 cameras, where each is connected to one of four Natural Point OptiTrack OptiHubs via USB 2.0 . Each hub has six USB downlink ports that each connect to one camera and has one USB uplink port that connects each hub to one USB 2.0 port on a computer. This transmits the data from the Cameras to the computer at 1.5, 12, or 480 Mbps. The power input to the hubs is 12 Volts (V) at 3 amps (A) and the power output per port to the cameras is 3.5 Watts (W) or 700 mA at 5 V. Status LEDs indicate power, uplink port status, download port status, and sync activity. The hubs are synced together by daisy chain starting from the 'hub sync out' port of the first hub and ending at the 'hub sync in' port of the last hub. The 'external sync in' port of the first, or master hub, is connected to a manually-controlled 9 V battery switch via a BNC cable for synchronized triggering of multiple systems to run simultaneously.

The cameras were optimally positioned, in order to track all desired markers. The goal of this setup was to arrange the cameras for the greatest amount of overlapping fields of view. These overlapping fields are known as the capture volume and can be seen in Figure 3.1.

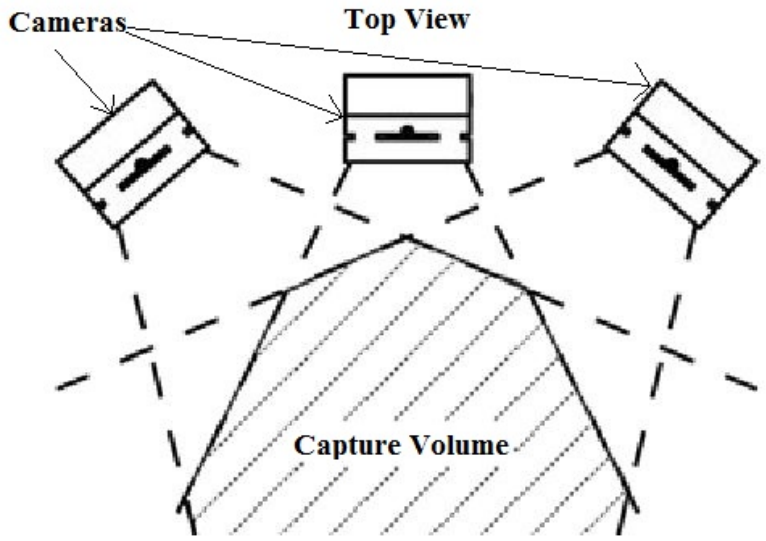


Figure 3.1: Capture volume created by overlapping fields of view of OEMC cameras

A Windows 7 based HP Compaq 8200 Elite SFF PC computer, which has an Intel i7-2600 processor with 3.4 GHz and 16 GB of RAM, is used for OEMC capture, in order to allow for fast calculation and processing of information from each OEMC hub. It is also equipped with eight USB 2.0 ports on four different USB hubs in order to allow for streaming of information from each OEMC hub on a separate USB hub.

The OEMC system runs on Tracking Tools (version 2.3.2) and Arena (version 1.7.1000) commercial software (Natural Point). During this study, Arena, which is software used for animation, was used only for calibration of the system because of its superior calibration capability. Calibration of the system is the method by which the system determines the position of all cameras with respect to each other in three-dimensional space. This is done by waving a wand, having three markers separated by known distances, in front of a frame that also has three markers separated by known distances. The frame and wand must be in a part of the capture volume that is visible by all 24 cameras. Once the position of all the cameras relative to each other is known, the

computer is able to read the two-dimensional image of each camera and convert them into several markers with three-dimensional coordinates associated with each. The calibration is done until the final error is less than 0.15 mm for each camera. The origin and orthogonal axes of the global volume is also calculated and the calibration of the system is saved as a .CAL file that can be opened either in Arena or Tracking Tools. It is important to note that, whenever a camera is moved, the system must be recalibrated.

Tracking Tools software is used for capturing and exporting data collected with the OEMC system. A new Tracking Tools project is started using the .CAL file and the following settings for all trials associated with this study are 175 Threshold, 5-8 Illumination, 3-7 Exposure, and 50 frames per second (fps). The Tracking Tools project is saved in a .TTP format, which saves the camera settings and calibration file and each capture is saved in a .TIM (timeline) file that allows for watching the marker movement in a three-dimensional volume. From the timeline, three-dimensional coordinate data can be exported with *x*-, *y*- and *z*- coordinates of all markers for each frame (50 fps in this case). This exported data file is in .CSV (comma separated values) format and a screen shot of data captured using Tracking Tools is shown in Figure 3.2.

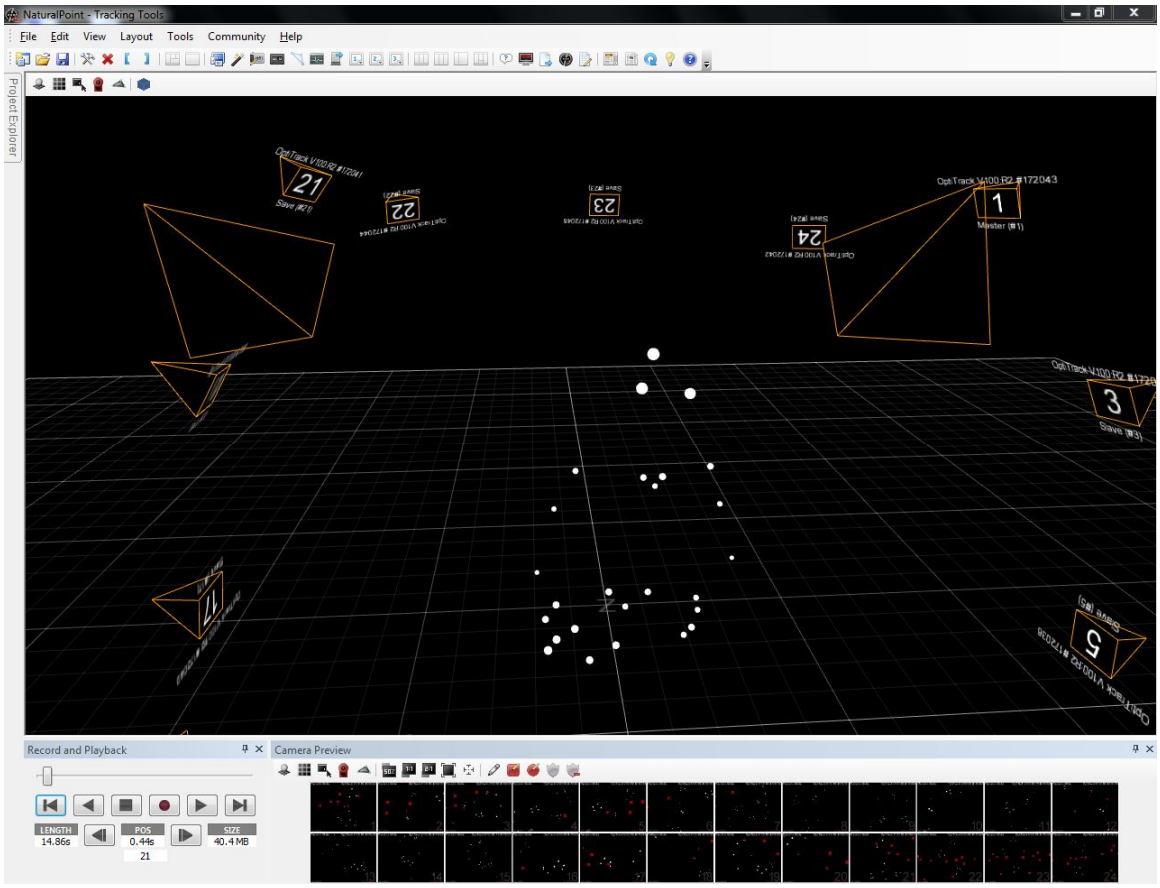


Figure 3.2: Tracking Tools screenshot

Post-processing of the exported data was done using custom MATLAB (version R2007b) code on a Sony Vaio VPCF1190X Laptop with 6GB of RAM and an Intel Core i7 processor with 1.60 GHz.

3.2 Marker Setup

The markers used are custom made from semispherical beads that are 14 mm in diameter. The beads are covered in retro-reflective tape (3M 8830 IR), which reflects the IR light emitted from the LEDs on the OEMC cameras and allows them to be tracked by the system. The markers are cleaned before and after each use to maintain reflective capacity and are placed on subjects using hypoallergenic double-sided tape.

This simplified marker scheme requires the minimum amount of markers in order to calculate the orientation of a plane for each body segment. In this study, there are three markers placed on the torso, three on the upper arm, four on the hand, and three on the tool being used. The markers on the torso are placed on the medial end of the left and right clavicles and on the munubrium sterni. This marker setup can be seen in Figure 3.3. The markers on the upper arm are placed on the fulcrum point of the shoulder, lateral epicondyle, and on the muscle belly of the lateral head of the triceps. The shoulder fulcrum is identified as the spot on the skin of the glenohumeral joint that acts as a fulcrum during shoulder flecion and extension. In order to place an offsetting marker on the posterior side of the lateral surface of the upper arm, a non-fatty spot was identified that does not move excessively when the upper arm is in motion. This marker setup can be seen in Figure 3.4. The markers on the hand are placed on the ulnar styloid, radial styloid, second metacarpophalangeal (MCP) joint, and fifth MCP joint and can be seen in Figure 3.5. The three markers that are placed on the tool are placed on a rigid L-shaped piece of wood, which was attached to the tool as seen in Figure 3.6.

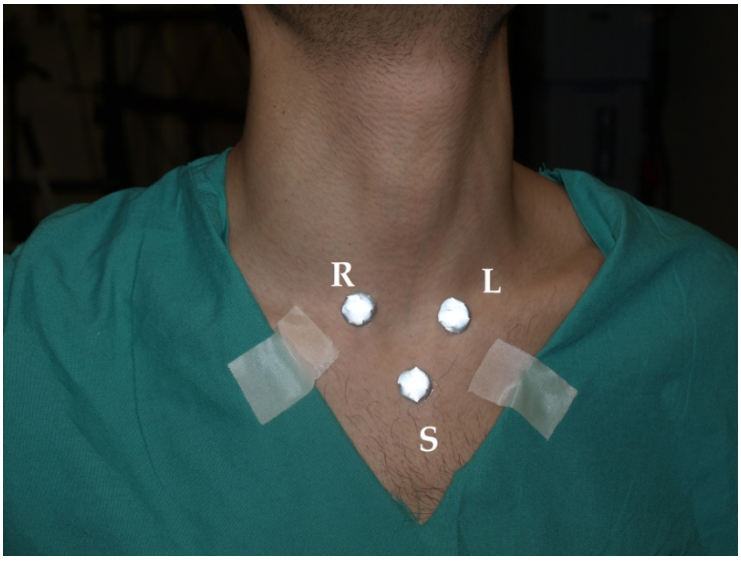


Figure 3.3: Marker setup of the torso

Figure 3.4: Marker setup of the upper arm

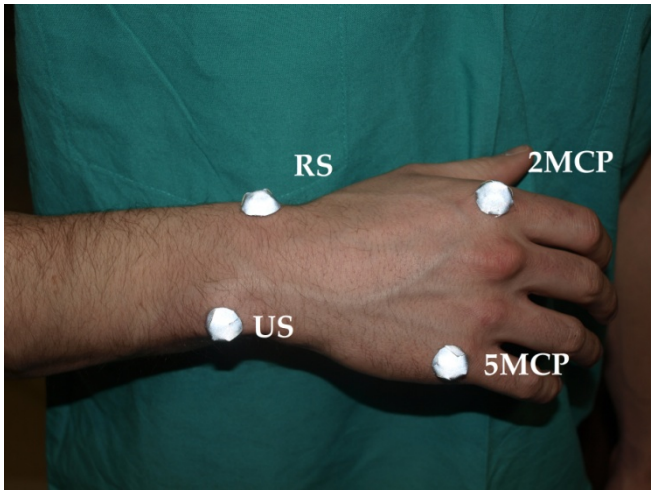


Figure 3.5: Marker setup of the hand

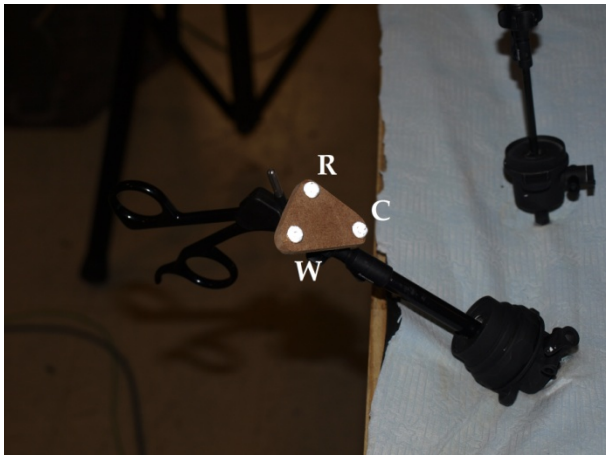


Figure 3.6: Sample tool fitted with markers

3.3 Vector Calculations

For each body segment used in this study, the markers were used to calculate and establish the three-dimensional vectorsthat were used to define the three orthogonal axes needed to represent a body segment in terms of x-, y- and z-axes. The calculation of these axes is explained further in the following sections.

3.3.1 Anatomical Orthogonal Axes Convention

The marker positions were used to calculate the orthogonal axes for each body segment, which was assumed to be a rigid body. A convention was used in this study to calculate the local orthogonal axes of any body part described in White et al. (1975) but with different directions. The orthogonal axes that were calculated for each body segment do not move relative to their assigned segment and the directions of the axes were determined using a standardized orientation convention where the position of the overall body is characterized by a typical orientation of the *x*-, *y*- and *z*-axes. More specifically, the standard position of the subject was considered to be a standing position with the arms at the subjects' side and the thumbs facing in the anterior direction. The positive *x*-axis was always calculated to point to the subject's left during standard position, the *y*-axis was always calculated to point in the subject's posterior direction, and the *z*-axis was always calculated to point in the subject's superior direction. This typical orientation convention can be seen in Figure 3.7, where the subject is in the neutral position. In addition, the orientation conventions of each body segment can be seen in Figures 3.8 through 3.12.

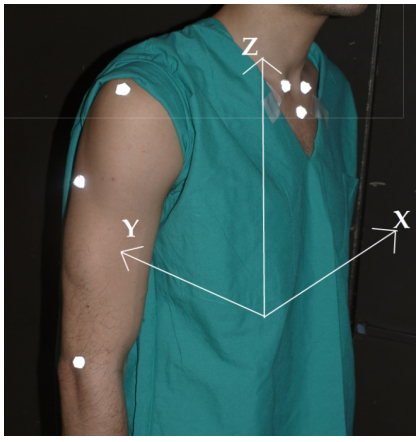


Figure 3.7: Standardized orientation convention of the orthogonal axes for the entire body

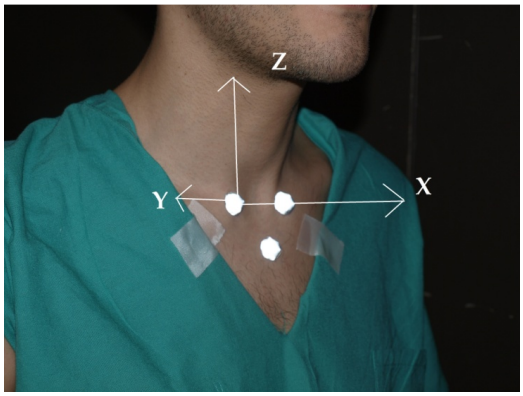


Figure 3.8: Standardized orientation convention of the orthogonal axes calculated for the torso

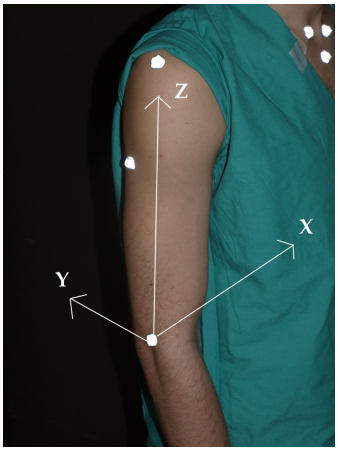


Figure 3.9: Standardized orientation convention of the orthogonal axes calculated for the upper arm

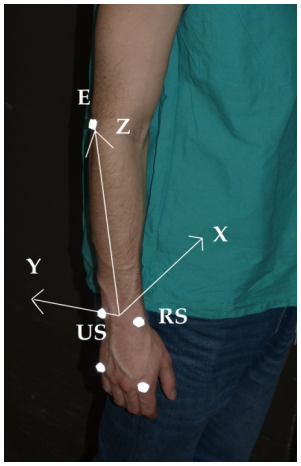


Figure 3.10: Standardized orientation convention of the orthogonal axes calculated for the forearm

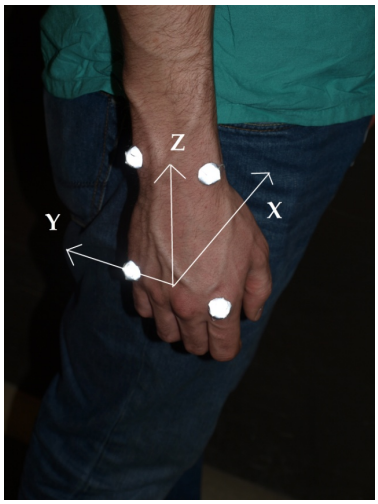


Figure 3.11: Standardized orientation convention of the orthogonal axes calculated for the hand

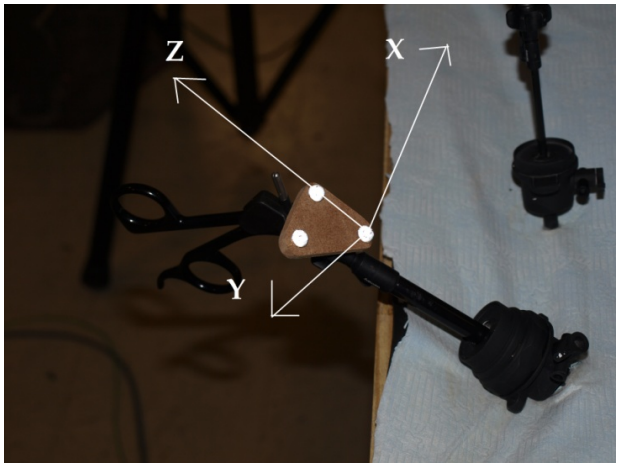


Figure 3.12: Standardized orientation convention of the orthogonal axes calculated for the tool

3.3.2 Vector Math to Calculate Orthogonal Axes

At least three markers, placed on anatomical landmarks, are used to define a relative plane and, ultimately, axes of a body segment. Any vector can be defined by subtracting the position of one marker from another and any two vectors on the same plane can also be determined this way. By taking the cross product of two vectors on the same plane, the resulting vector will be normal to the plane. By crossing this resultant vector and one of the vectors on the original plane, the result is a third vector that

completes the orthogonal arrangement. The markers are set up on a body segment in a manner that allows for the calculation of a vector that serves as a particular coordinate axis (*x*, *y,* or *z*) following the orientation convention previously described. There is also an offsetting marker on the same plane that allows for the calculation of another vector and, by using the cross product method described, the two vectors are used to establish two orthogonal axes. The three orthogonal axes are considered the orthogonal axes of the body segment and all orthogonal vectors are normalized to become unit vectors. The process of calculating the orthogonal axes for each body segment are described in more detail in the following sections.

3.3.3 Torso Orthogonal Axes Calculations

For calculation of the orthogonal axes associated with the torso, shown in Figure 3.8, the first step was to calculate the vectors from markers L to R and S to R. These calculations were done using Equations 11 and 12 and the vectors were normalized using Equations 13 and 14, where vector $\overrightarrow{eV_x}$ is the *x*-axis of the torso and $\overrightarrow{eV_o}$ is the reference vector. The normalized result of the cross product of the two vectors yielded the *y*-axis of the torso, $\overrightarrow{eV_y}$ (Equations 15 and 16). The order of the cross product is determined by the right-hand rule so that the *y*-axis points in the posterior direction and the *z*-axis is the cross product of the *x*- and *y*-axes (Equation 17), which was not normalized since it is the cross product of two normalized orthogonal vectors. The torso orthogonal axes were adjusted to account for the slope of the sternum, which is explained in more detail later in the methods.

$$\overrightarrow{V_x} = (L_x - R_x)\hat{\imath} + \left(L_y - R_y\right)\hat{\jmath} + (L_z - R_z)\hat{k} \tag{11}$$

$$\overrightarrow{V_o} = (S_x - R_x)\hat{\imath} + \left(S_y - R_y\right)\hat{\jmath} + (S_z - R_z)\hat{k} \tag{12}$$

$$\overrightarrow{eV_x} = \frac{\overrightarrow{V_x}}{|V_x|} = \frac{V_x\hat{\imath}+V_x\hat{\jmath}+V_x\hat{k}}{\sqrt{(V_x\hat{\imath})^2+(V_x\,\hat{\jmath})^2+(V_x\,\hat{k})^2}} \tag{13}$$

$$\overrightarrow{eV_o} = \frac{\overrightarrow{V_o}}{|V_o|} = \frac{V_o\hat{\imath}+V_o\hat{\jmath}+V_o\hat{k}}{\sqrt{(V_o\,\hat{\imath})^2+(V_o\,\hat{\jmath})^2+(V_o\,\hat{k})^2}} \tag{14}$$

$$\overrightarrow{V_y} = \overrightarrow{eV_x} \times \overrightarrow{eV_o} \tag{15}$$

$$\overrightarrow{eV_y} = \frac{\overrightarrow{V_y}}{|V_y|} = \frac{V_y\hat{\imath}+V_y\hat{\jmath}+V_y\hat{k}}{\sqrt{(V_y\hat{\imath})^2+(V_y\,\hat{\jmath})^2+(V_y\,\hat{k})^2}} \tag{16}$$

$$\overrightarrow{eV_z} = \overrightarrow{eV_x} \times \overrightarrow{eV_y}, \tag{17}$$

where the torso *x*-, *y*-, and *z*-axes are $\overrightarrow{eV_x}$, $\overrightarrow{eV_y}$, and $\overrightarrow{eV_z}$, respectively.

3.3.4 Upper Arm Orthogonal Axes Calculations

For calculation of the orthogonal axes associated with the upper arm, shown in Figure 3.9, the first step was to calculate the vectors from markers S to E and O to E. These calculations were done using Equations 18 and 19 and the vectors were normalized using Equations 20 and 21, where vector $\overrightarrow{eV_z}$ is the *z*-axis of the upper arm and $\overrightarrow{eV_o}$ is the reference vector. The normalized result of the cross product of the two vectors yielded the *x*-axis of the upper arm, $\overrightarrow{eV_x}$ (Equation 22 and 23). The order of the cross product was determined by the right-hand rule so that the *x*-axis is pointing in the proper direction by the conventions and the *y*-axis is the cross product of the *z*- and *x*-axes (Equation 24), which was not normalized since it is the cross product of two normalized orthogonal vectors.

$$\overrightarrow{V_z} = (S_x - E_x)\hat{\imath} + \left(S_y - E_y\right)\hat{\jmath} + (S_z - E_z)\hat{k} \tag{18}$$

$$\overrightarrow{V_o} = (O_x - E_x)\hat{\imath} + \left(O_y - E_y\right)\hat{\jmath} + (O_z - E_z)\hat{k} \tag{19}$$

$$\overrightarrow{eV_z} = \frac{\overrightarrow{V_z}}{|V_z|} = \frac{V_z\hat{\imath}+V_z\hat{\jmath}+V_z\hat{k}}{\sqrt{(V_z\hat{\imath})^2+(V_z\,\hat{\jmath})^2+(V_z\,\hat{k})^2}} \tag{20}$$

$$\overrightarrow{eV_o} = \frac{\overrightarrow{V_o}}{|V_o|} = \frac{V_o\hat{\imath}+V_o\hat{\jmath}+V_o\hat{k}}{\sqrt{(V_o\,\hat{\imath})^2+(V_o\,\hat{\jmath})^2+(V_o\,\hat{k})^2}} \tag{21}$$

$$\overrightarrow{V_x} = \overrightarrow{eV_z} \times \overrightarrow{eV_o} \tag{22}$$

$$\overrightarrow{eV_x} = \frac{\overrightarrow{V_x}}{|V_x|} = \frac{V_x\hat{\imath}+V_x\hat{\jmath}+V_x\hat{k}}{\sqrt{(V_x\hat{\imath})^2+(V_x\,\hat{\jmath})^2+(V_x\,\hat{k})^2}} \tag{23}$$

$$\overrightarrow{eV_y} = \overrightarrow{eV_z} \times \overrightarrow{eV_x}, \tag{24}$$

where the upper arm *x*-, *y*-, and *z*-axes are $\overrightarrow{eV_x}$, $\overrightarrow{eV_y}$, and $\overrightarrow{eV_z}$, respectively.

3.3.5 Forearm Orthogonal Axes Calculations

For calculation of the orthogonal axes associated with the forearm, shown in Figure 3.10, the first step was to calculate the midpoint of the styloid markers, U and R. This was done using Equation 25, which yielded styloid midpoint, W. The vectors from markers E to W and W to U were calculated using Equations 26 and 27 and the vectors were normalized using Equations 28 and 29, where vector $\overrightarrow{eV_z}$ is the *z*-axis of the forearm and $\overrightarrow{eV_o}$ is the reference vector. The normalized result of the cross product of the two vectors yielded the *x*-axis of the forearm, $\overrightarrow{eV_x}$ (Equation 30 and 31). The order of the cross product was determined by the right-hand rule so that the *x*-axis is pointing in the proper direction by the conventions and the *y*-axis is the cross product of the *z*- and *x*-

axes (Equation 32), which was not normalized since it is the cross product of two normalized orthogonal vectors.

$$\overrightarrow{W} = \frac{(U_x+R_x)}{2}\hat{\imath} + \frac{(U_y+R_y)}{2}\hat{\jmath} + \frac{(U_z+R_z)}{2}\hat{k} \tag{25}$$

$$\overrightarrow{V_z} = (E_x - W_x)\hat{\imath} + (E_y - W_y)\hat{\jmath} + (E_z - W_z)\hat{k} \tag{26}$$

$$\overrightarrow{V_o} = (U_x - W_x)\hat{\imath} + (U_y - W_y)\hat{\jmath} + (U_z - W_z)\hat{k} \tag{27}$$

$$\overrightarrow{eV_z} = \frac{\overrightarrow{V_z}}{|V_z|} = \frac{V_z\hat{\imath}+V_z\hat{\jmath}+V_z\hat{k}}{\sqrt{(V_z\hat{\imath})^2+(V_z\,\hat{\jmath})^2+(V_z\,\hat{k})^2}} \tag{28}$$

$$\overrightarrow{eV_o} = \frac{\overrightarrow{V_o}}{|V_o|} = \frac{V_o\hat{\imath}+V_o\hat{\jmath}+V_o\hat{k}}{\sqrt{(V_o\,\hat{\imath})^2+(V_o\,\hat{\jmath})^2+(V_o\,\hat{k})^2}} \tag{30}$$

$$\overrightarrow{V_x} = \overrightarrow{eV_o} \times \overrightarrow{eV_z} \tag{31}$$

$$\overrightarrow{eV_x} = \frac{\overrightarrow{V_x}}{|V_x|} = \frac{V_x\hat{\imath}+V_x\hat{\jmath}+V_x\hat{k}}{\sqrt{(V_x\hat{\imath})^2+(V_x\,\hat{\jmath})^2+(V_x\,\hat{k})^2}} \tag{32}$$

$$\overrightarrow{eV_y} = \overrightarrow{eV_z} \times \overrightarrow{eV_x}, \tag{33}$$

where the forearm *x*-, *y*-, and *z*-axes are $\overrightarrow{eV_x}$, $\overrightarrow{eV_y}$, and $\overrightarrow{eV_z}$, respectively.

3.3.6 Hand Orthogonal Axes Calculations

For calculation of the orthogonal axes associated with the hand, shown in Figure 3.11, the first step was to calculate the midpoint of the styloid markers, U and R. The next step was to calculate the midpoint of the MCP markers, M_2 and M_5, which is done using Equations 33 and 34, where the *W* is the styloid midpoint and *M* is the MCP midpoint. The vectors from markers *W* to *M* and M_5 to *M* were calculated using

Equations 35 and 36, and normalized using Equations 37 and 38, where vector $\overrightarrow{eV_z}$ is the *z*-axis of the hand and $\overrightarrow{eV_o}$ is the reference vector. The normalized result of the cross product of the two vectors yielded the *x*-axis of the hand, $\overrightarrow{eV_x}$, as shown in Equations 39 and 40. The order of the cross product was determined by the right-hand rule so that the *x*-axis is pointing in the proper direction by the conventionsand the *y*-axis is the cross product of the *z*- and *x*-axes (Equation 41), which was not normalized since it is the cross product of two normalized orthogonal vectors.

$$\overrightarrow{W} = \frac{(U_x+R_x)}{2}\hat{\imath} + \frac{(U_y+R_y)}{2}\hat{\jmath} + \frac{(U_z+R_z)}{2}\hat{k} \tag{33}$$

$$\overrightarrow{M} = \frac{(M_{2x}+M_{5x})}{2}\hat{\imath} + \frac{(M_{2y}+M_{5y})}{2}\hat{\jmath} + \frac{(M_{2z}+M_{5z})}{2}\hat{k} \tag{34}$$

$$\overrightarrow{V_z} = (W_x - M_x)\hat{\imath} + (W_y - M_y)\hat{\jmath} + (W_z - M_z)\hat{k} \tag{35}$$

$$\overrightarrow{V_o} = (M_{5x} - M_x)\hat{\imath} + (M_{5y} - M_y)\hat{\jmath} + (M_{5y} - M_y)\hat{k} \tag{36}$$

$$\overrightarrow{eV_z} = \frac{\overrightarrow{V_z}}{|V_z|} = \frac{V_z\hat{\imath}+V_z\hat{\jmath}+V_z\hat{k}}{\sqrt{(V_z\hat{\imath})^2+(V_z\,\hat{\jmath})^2+(V_z\,\hat{k})^2}} \tag{37}$$

$$\overrightarrow{eV_o} = \frac{\overrightarrow{V_o}}{|V_o|} = \frac{V_o\hat{\imath}+V_o\hat{\jmath}+V_o\hat{k}}{\sqrt{(V_o\,\hat{\imath})^2+(V_o\,\hat{\jmath})^2+(V_o\,\hat{k})^2}} \tag{38}$$

$$\overrightarrow{V_x} = \overrightarrow{eV_o} \times \overrightarrow{eV_z} \tag{39}$$

$$\overrightarrow{eV_x} = \frac{\overrightarrow{V_x}}{|V_x|} = \frac{V_x\hat{\imath}+V_x\hat{\jmath}+V_x\hat{k}}{\sqrt{(V_x\hat{\imath})^2+(V_x\,\hat{\jmath})^2+(V_x\,\hat{k})^2}} \tag{40}$$

$$\overrightarrow{eV_y} = \overrightarrow{eV_z} \times \overrightarrow{eV_x}, \tag{41}$$

where the hand *x*-, *y*-, and *z*-axes are $\overrightarrow{eV_x}$, $\overrightarrow{eV_y}$, and $\overrightarrow{eV_z}$, respectively.

3.3.7 Tool Orthogonal Axes Calculations

For calculation of the orthogonal axes associated with the tool marker label, shown in Figure 3.12, the first step was to calculate the vectors from markers R to C and W to C. These calculations were done using Equations 42 and 43 and were normalized using Equations 44 and 45. Vector $\overrightarrow{eV_z}$ is the *z*-axis of the tool and $\overrightarrow{eV_o}$ is the offsetting vector on the plane. The normalized result of the cross product of the two vectors yielded the *y*-axis of the tool, $\overrightarrow{eV_y}$, shown in Equations 46 and 47. The order of the cross product was determined by the right-hand rule so that the *y*-axis is pointing in the posterior direction by our conventions and the *x*-axis is the cross product of the *y*- and *z*-axes (Equation 48), which was not normalized since it is the cross product of two normalized orthogonal vectors.

$$\overrightarrow{V_z} = (R_x - C_x)\hat{\imath} + \left(R_y - C_y\right)\hat{\jmath} + (R_z - C_z)\hat{k} \tag{42}$$

$$\overrightarrow{V_o} = (W_x - C_x)\hat{\imath} + \left(W_y - C_y\right)\hat{\jmath} + (W_z - C_z)\hat{k} \tag{43}$$

$$\overrightarrow{eV_z} = \frac{\overrightarrow{V_z}}{|V_z|} = \frac{V_z\hat{\imath}+V_z\hat{\jmath}+V_z\hat{k}}{\sqrt{(V_z\hat{\imath})^2+(V_z\,\hat{\jmath})^2+(V_z\,\hat{k})^2}} \tag{44}$$

$$\overrightarrow{eV_o} = \frac{\overrightarrow{V_o}}{|V_o|} = \frac{V_o\hat{\imath}+V_o\hat{\jmath}+V_o\hat{k}}{\sqrt{(V_o\,\hat{\imath})^2+(V_o\,\hat{\jmath})^2+(V_o\,\hat{k})^2}} \tag{45}$$

$$\overrightarrow{V_y} = \overrightarrow{eV_o} \times \overrightarrow{eV_z} \tag{46}$$

$$\overrightarrow{eV_y} = \frac{\overrightarrow{V_y}}{|V_y|} = \frac{V_y\hat{\imath}+V_y\hat{\jmath}+V_y\hat{k}}{\sqrt{(V_y\hat{\imath})^2+(V_y\,\hat{\jmath})^2+(V_y\,\hat{k})^2}} \tag{47}$$

$$\overrightarrow{eV_x} = \overrightarrow{eV_y} \times \overrightarrow{eV_z}, \tag{48}$$

where the tool *x*-, *y*-, and *z*-axes are $\overrightarrow{eV_x}$, $\overrightarrow{eV_y}$, and $\overrightarrow{eV_z}$, respectively.

3.4 Torso Rotation

Before capturing the motion of a subject during a task, a neutral capture was recorded, where the subject was instructed to stand up straight with their hands at their sides and to look forward. The subject was positioned so that their anatomical planes align closely with the orthogonal planes of the global coordinate system and was recorded for approximately five seconds. The goal of this capture was to have a reference of the neutral slope of the chest because the torso *xz*-plane is defined as the plane of the chest as shown in Figure 3.13 below. Since the plane of the chest is never vertical, this method was used to find the angle between the chest and global vertical axes.

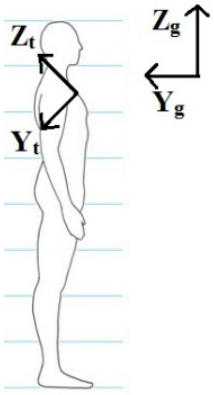


Figure 3.13: Torso and global *z*- and *y*-axes

In order to make this calculation, the vertical *z*-axis of the torso was projected onto the global plane that is approximately parallel to the sagittal plane of the subject's body (i.e., the *yz*-plane in this study). This was done using Equation 49, where G_n is the

axis normal to the global *yz*-plane, $\overrightarrow{z_t}$ is the *z*-axis of the chest and $\overrightarrow{Z_p}$ is the projection of the $\overrightarrow{z_t}$ onto the *yz*-plane. This projection was normalized as shown in Equation 50, where $\overrightarrow{eZ_p}$ is the normalized vector, and the angle between vertical axes was found using Equation 51, where θ is the angle between $\overrightarrow{eZ_p}$ and $\overrightarrow{Z_g}$, which is the global *z*-axis and rotation matrix, *R*, was generated using θ, as shown in Equation 52, and the chest orthogonal axes were rotated about its own *x*-axis by the magnitude of the calculated angle. When subjects were captured during the tasks of interest, the coordinates of each axis of the chest were rotated using the rotation matrix as shown in Equation 53, where *i, j,* and *k* are the original coordinates of the axis and *i', j',* and *k'* are the rotated coordinates.

$$\overrightarrow{Z_p} = \overrightarrow{\mathrm{G_n}} \times \left(\overrightarrow{z_t} \times \overrightarrow{\mathrm{G_n}}\right) \tag{49}$$

$$\overrightarrow{eZ_p} = \frac{\overrightarrow{Z_p}}{|Z_p|} = \frac{Z_p\hat{\imath}+Z_p\hat{\jmath}+Z_p\hat{k}}{\sqrt{(Z_p\hat{\imath})^2+(Z_p\hat{\jmath})^2+(Z_p\,\hat{k})^2}} \tag{50}$$

$$\theta = cos^{-1}(\overrightarrow{eZ_p} \cdot \overrightarrow{Z_g}) \tag{51}$$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\,\theta & sin\,\theta \\ 0 & -sin\,\theta & cos\,\theta \end{bmatrix} \tag{52}$$

$$\begin{bmatrix} i' \\ j' \\ k' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\,\theta & sin\,\theta \\ 0 & -sin\,\theta & cos\,\theta \end{bmatrix} \begin{bmatrix} i \\ j \\ k \end{bmatrix} \tag{53}$$

Once the axes were rotated, the torso orthogonal axes can be considered to represent the true anatomical axes, which allowed for projection onto the torso planes and angle measurement to be representative of the true values.

3.5 Projection onto Planes

Projecting a vector onto a plane is essentially taking the component of a vector in three-dimensional space and redefining it on a two-dimensional plane using only the components that lie in the plane. A simple way to visual this is the image of a light projector pointed directly at a surface and putting an arrow somewhere in the line of light. This arrow will have a two-dimensional shadow projected onto the surface. Figure 3.14, below, illustrates a vector, *A,* being projected onto a plane with a normal vector, *B*. This process can be mathematically conducted, in order to project a vector in space onto a plane by knowing the components of the vector being projected and the components of the vector that is normal to the plane. Certain axes of the calculated orthogonal axes are projected onto planes of other anatomical segments, in order to calculate the angle between the projected vector and the reference vector on the projection plane.

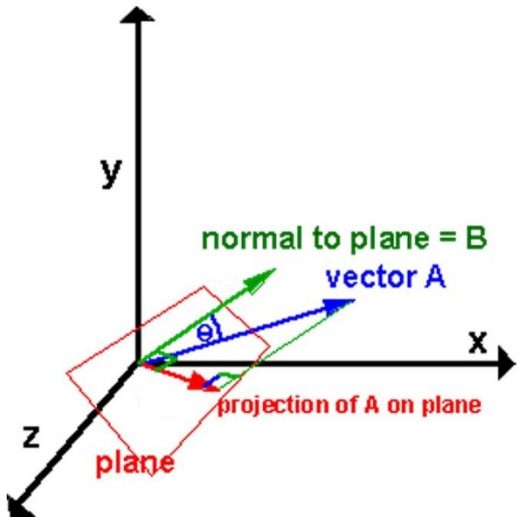


Figure 3.14: Projecting a vector onto a plane

3.5.1 Upper Arm Flexion and Extension Projection

In order to calculate the upper arm flexion and extension, the *z*- axis of the upper arm was projected onto the *yz*-plane of the torso and then normalized. This was done using Equations 54 and 55, where $\overrightarrow{Z_p}$ is the projection of the upper arm *z*-axis onto the torso's *yz*-plane, $\overrightarrow{T_n}$ is the vector normal to the torso *yz*-plane, which is also known as the torso *x*-axis, and $\overrightarrow{eZ_p}$ is the normalized *z*-axis.

$$\overrightarrow{Z_p} = \overrightarrow{T_n} \times \left(\overrightarrow{Z_u} \times \overrightarrow{T_n}\right) \tag{54}$$

$$\overrightarrow{eZ_p} = \frac{\overrightarrow{Z_p}}{|Z_p|} = \frac{Z_p\hat{\imath}+Z_p\hat{\jmath}+Z_p\hat{k}}{\sqrt{(Z_p\hat{\imath})^2+(Z_p\hat{\jmath})^2+(Z_p\,\hat{k})^2}} \tag{55}$$

3.5.2 Upper Arm Abduction and Adduction Projection

In order to calculate the upper arm abduction and adduction, the *z*-axis of the upper arm was projected onto the *xz*-plane of the torso and then normalized. This was done using Equations 56 and 57, where $\overrightarrow{Z_p}$ is the projection of the upper arm *z*-axis onto the torso's *xz*-plane, $\overrightarrow{T_n}$ is the vector normal to the torso *xz*-plane, which is the *y*-axis of the torso, and $\overrightarrow{eZ_p}$ is the normalized *z*-.

$$\overrightarrow{Z_p} = \overrightarrow{T_n} \times \left(\overrightarrow{Z_u} \times \overrightarrow{T_n}\right) \tag{56}$$

$$\overrightarrow{eZ_p} = \frac{\overrightarrow{Z_p}}{|Z_p|} = \frac{Z_p\hat{\imath}+Z_p\hat{\jmath}+Z_p\hat{k}}{\sqrt{(Z_p\hat{\imath})^2+(Z_p\hat{\jmath})^2+(Z_p\,\hat{k})^2}} \tag{57}$$

3.5.3 Upper Arm Medial and Lateral Rotation Projection

In order to calculate the upper arm medial and lateral rotation, the *x*-axis of the upper arm as projected onto the *xy*-plane of the torso and then normalized. This was done using Equations 58 and 59, where $\overrightarrow{X_p}$ is the projection of the upper arm *x*-axis onto the torso's *xy*-plane, $\overrightarrow{T_n}$ is the vector normal to the torso *xy*-plane, which is the *z*-axis of the torso and eX_p is the normalized *x*- projection.

$$\overrightarrow{X_p} = \overrightarrow{T_n} \times \left(\overrightarrow{X_u} \times \overrightarrow{T_n}\right) \tag{58}$$

$$\overrightarrow{eX_p} = \frac{\overrightarrow{X_p}}{|X_p|} = \frac{X_p\hat{\imath}+X_p\hat{\jmath}+X_p\hat{k}}{\sqrt{(X_p\hat{\imath})^2+(X_p\hat{\jmath})^2+(X_p\,\hat{k})^2}} \tag{59}$$

3.5.4 Forearm Rotation Projection

In order to calculate the forearm rotation, or pronation/supination, the *x*-axis of the upper arm was projected onto the *xy*-plane of the forearm and normalized. This was done using Equations 60 and 61, where $\overrightarrow{X_p}$ is the projection of the upper arm *x*-axis onto the forearm's *xy*-plane, $\overrightarrow{F_n}$ is the vector normal to the forearm *xy*-plane, which is the *z*-axis of the forearm, and $\overrightarrow{eX_p}$ is the normalized *x*- projection.

$$\overrightarrow{X_p} = \overrightarrow{F_n} \times \left(\overrightarrow{X_u} \times \overrightarrow{F_n}\right) \tag{60}$$

$$\overrightarrow{eX_p} = \frac{\overrightarrow{X_p}}{|X_p|} = \frac{X_p\hat{\imath}+X_p\hat{\jmath}+X_p\hat{k}}{\sqrt{(X_p\hat{\imath})^2+(X_p\hat{\jmath})^2+(X_p\,\hat{k})^2}} \tag{61}$$

3.5.5 Wrist Flexion and Extension Projection

In order to calculate the wrist flexion and extension, the *z*-axis of the hand is projected onto the *xz*-plane of the forearm and normalized. This is done using Equations 62 and 63, where $\overrightarrow{Z_p}$ is the projection of the hand *z*-axis onto the forearm's *xz*-plane, $\overrightarrow{F_n}$ is the vector normal to the forearm's *xz*-plane which is the *y*-axis of the forearm, and $\overrightarrow{eZ_p}$ is the normalized *z*- projection.

$$\overrightarrow{Z_p} = \overrightarrow{F_n} \times \left(\overrightarrow{Z_h} \times \overrightarrow{F_n}\right) \tag{62}$$

$$\overrightarrow{eZ_p} = \frac{\overrightarrow{Z_p}}{|Z_p|} = \frac{Z_p\hat{\imath}+Z_p\hat{\jmath}+Z_p\hat{k}}{\sqrt{(Z_p\hat{\imath})^2+(Z_p\hat{\jmath})^2+(Z_p\,\hat{k})^2}} \tag{63}$$

3.5.6 Wrist Ulnar and Radial Deviation Projection

In order to calculate the wrist ulnar and radial deviation, the *z*-axis of the hand was projected onto the *yz*-plane of the forearm and normalized. This was done using Equations 64 and 65, where $\overrightarrow{Z_p}$ is the projection of the hand *z*-axis onto the forearm's *yz*-plane, $\overrightarrow{F_n}$ is the vector normal to the forearm's *yz*-plane, which is the *x*-axis of the forearm, and $\overrightarrow{eZ_p}$ is the normalized *z*- projection.

$$\overrightarrow{Z_p} = \overrightarrow{F_n} \times \left(\overrightarrow{Z_h} \times \overrightarrow{F_n}\right) \tag{64}$$

$$\overrightarrow{eZ_p} = \frac{\overrightarrow{Z_p}}{|Z_p|} = \frac{Z_p\hat{\imath}+Z_p\hat{\jmath}+Z_p\hat{k}}{\sqrt{(Z_p\hat{\imath})^2+(Z_p\hat{\jmath})^2+(Z_p\,\hat{k})^2}} \tag{65}$$

3.5.7 Tool Yaw Projection

In order to calculate the tool yaw, the *z*-axis of the tool was projected onto the *xz*-plane of the hand and normalized. This was done using Equations 66 and 67, where $\overrightarrow{Z_p}$ is the projection of the tool *z*-axis onto the hand's *xz*-plane, $\overrightarrow{H_n}$ is the vector normal to the hand's *xz*-plane, which is the *y*-axis of the hand, and $\overrightarrow{eZ_p}$ is the normalized *z*- projection.

$$\overrightarrow{Z_p} = \overrightarrow{H_n} \times \left(\overrightarrow{Z_t} \times \overrightarrow{H_n}\right) \tag{66}$$

$$\overrightarrow{eZ_p} = \frac{\overrightarrow{Z_p}}{|Z_p|} = \frac{Z_p\hat{\imath}+Z_p\hat{\jmath}+Z_p\hat{k}}{\sqrt{(Z_p\hat{\imath})^2+(Z_p\hat{\jmath})^2+(Z_p\,\hat{k})^2}} \tag{67}$$

3.5.8 Tool Pitch Projection

In order to calculate the tool pitch, the *z*-axis of the tool was projected onto the *yz*-plane of the hand and normalized. This was done using Equations 68 and 69, where $\overrightarrow{Z_p}$ is the projection of the tool *z*-axis onto the hand's *yz*-plane, $\overrightarrow{H_n}$ is the vector normal to the hand's *yz*-plane, which is the *x*-axis of the hand, and $\overrightarrow{eZ_p}$ is the normalized *z*- projection.

$$\overrightarrow{Z_p} = \overrightarrow{H_n} \times \left(\overrightarrow{Z_t} \times \overrightarrow{H_n}\right) \tag{68}$$

$$\overrightarrow{eZ_p} = \frac{\overrightarrow{Z_p}}{|Z_p|} = \frac{Z_p\hat{\imath}+Z_p\hat{\jmath}+Z_p\hat{k}}{\sqrt{(Z_p\hat{\imath})^2+(Z_p\hat{\jmath})^2+(Z_p\,\hat{k})^2}} \tag{69}$$

3.5.9 Tool Roll Projection

In order to calculate the tool roll, the *x*-axis of the tool was projected onto the *xy*-plane of the hand and normalized. This was done using Equations 70 and 71, where $\overrightarrow{X_p}$ is

the projections of the tool *x*-axis onto the hand's *xy*-plane, $\overrightarrow{H_n}$ is the vector normal to the hand's *xy*-plane, which is the *z*-axis of the hand, and $\overrightarrow{eX_p}$ is the normalized *x*- projection.

$$\overrightarrow{X_p} = \overrightarrow{H_n} \times \left(\overrightarrow{X_t} \times \overrightarrow{H_n}\right) \tag{70}$$

$$\overrightarrow{eX_p} = \frac{\overrightarrow{X_p}}{|X_p|} = \frac{X_p\hat{\imath}+X_p\hat{\jmath}+X_p\hat{k}}{\sqrt{(X_p\hat{\imath})^2+(X_p\hat{\jmath})^2+(X_p\,\hat{k})^2}} \tag{71}$$

3.6 Angle Calculations

Anatomical rotation angles were calculated as the angle between the vector projections and the reference vectors, which is found by the dot product of the two vectors. The reference vector was the corresponding axis on the projection plane, where, if the two vectors are parallel, there would be no rotation. Since the dot product only gives the magnitude of an angle, direction was determined by redefining the base coordinate system of the projected vector as the orthogonal axes of the reference body segment (Chen, 1999) and by finding the sign of a certain component of the redefined vector. The details of these processes are explained in detail in the following sections for each rotation calculated.

3.6.1 Shoulder Flexion and Extension Calculation

To calculate the shoulder flexion and extension, the dot product was used to find the angle between the upper arm vector projection and the torso coordinate axes. Shoulder flexion and extension was calculated as the angle between the projection of the upper arm *z*-axis and the torso *z*-axis, as shown in Equation 72.

$$\theta_z = cos^{-1}(\overrightarrow{eZ_p} \cdot \overrightarrow{Z_t}) \tag{72}$$

The directions of the angles were determined by re-defining the base coordinate system for the projected vector to the torso coordinate axes and the sign of the *y*-component of the vector was used to evaluate the sign of the rotation. If the sign of the *y*-component of the vector was positive the rotation was flexion and if the sign of the *y*-component was negative, the rotation was extension. Using the right-hand rule and the axes sign convention, flexion was considered a negative rotation about the *x*-axis and extension was considered a positive rotation. The projected vector was redefined as shown in Equations 73, where B_t is the 3x3 matrix of the torso's coordinate axes (x is the first column, y is the second column and z is the third), B_n is an identity matrix, P is the product of B_n and the inverse of B_t, Z_p is the upper arm *z*- projection, and V_c is the redefined vector used to check the sign.

$$P = B_n * {B_t}^{-1}$$

$$V_c = P * \begin{bmatrix} Z_{px} \\ Z_{py} \\ Z_{pz} \end{bmatrix} \tag{73}$$

3.6.2 Shoulder Abduction and Adduction Calculation

To calculate the shoulder abduction and adduction, the dot product was used to find the angle between the upper arm vector projections and the torso coordinate axes. Shoulder abduction and adduction were calculated as the angle between the projection of the upper arm *z*-axis and the torso *z*-axis, as shown in Equation 74.

$$\theta_z = cos^{-1}(\overrightarrow{eZ_p} \cdot \overrightarrow{Z_t}) \tag{74}$$

The directions of the angles were determined by re-defining the base coordinate system for the projected vector to the torso coordinate axes and the sign of the *x*-component of the vector was used to evaluate the sign of the rotation. If the sign of the *x*-component of the vector was positive the rotation as abduction and if the sign of the *x*-component was negative, the rotation was adduction. Using the right-hand rule and the axes sign convention, abduction was considered a negative rotation about the *y*-axis and adduction was considered a positive rotation. The projected vector is redefined, as shown in Equations 75, where B_t is the 3x3 matrix of the torso's coordinate axes, B_n is an identity matrix, P is the product of B_n and the inverse of B_t, Z_p is the upper arm *z*-projection, and V_c is the redefined vector used to check the sign.

$$P = B_n * {B_t}^{-1}$$

$$V_c = P * \begin{bmatrix} Z_{px} \\ Z_{py} \\ Z_{pz} \end{bmatrix} \tag{75}$$

3.6.3 Shoulder Medial and Lateral Rotation Calculation

To calculate the shoulder medial and lateral rotation, the dot product was used to find the angle between the upper arm vector projections and the torso coordinate axes. Shoulder medial and lateral rotation was calculated as the angle between the projection of the upper arm *x*-axis and the torso *x*-axis, as shown in Equation 76.

$$\theta_x = cos^{-1}(\overrightarrow{eX_p} \cdot \overrightarrow{X_t}) \tag{76}$$

The directions of the angles were determined by re-defining the base coordinate system for the projected vector to the torso coordinate axes and the sign of the *y*-

component of the vector was used to evaluate the sign of the rotation. If the sign of the *y*-component of the vector as positive, the rotation was medial and if the sign of the *y*-component was negative, the rotation was lateral. Using the right-hand rule and the axes sign convention, medial rotation was considered a positive rotation about the *z*-axis and lateral rotation as considered a negative rotation. The projected vector was redefined as shown in Equations 77, where B_t is the 3x3 matrix of the torso's coordinate axes, B_n is an identity matrix, P is the product of B_n and the inverse of B_t, X_p is the upper arm *x*-projection, and V_c is the redefined vector used to check the sign.

$$P = B_n * {B_t}^{-1}$$

$$V_c = P * \begin{bmatrix} X_{px} \\ X_{py} \\ X_{pz} \end{bmatrix} \tag{77}$$

3.6.4 Forearm Rotation Calculation

To calculate the forearm rotation, the dot product was used to find the angle between the upper arm vector projections and the forearm coordinate axes. Forearm rotation was calculated as the angle between the projection of the upper arm *x*-axis and the forearm *x*-axis, as shown in Equation 78.

$$\theta_x = cos^{-1}(\overrightarrow{eX_p} \cdot \overrightarrow{X_f}) \tag{78}$$

The directions of the angles were determined by re-defining the base coordinate system for the projected vector to the forearm coordinate axes and the sign of the *y*-component of the vector as used to evaluate the sign of the rotation. If the sign of the *y*-component of the vector was positive, the rotation was supination and if the sign of the *y*-

component was negative, the rotation was pronation. Using the right-hand rule and the axes sign convention, pronation was considered a positive rotation about the *z*-axis and supination was considered a negative rotation. The projected vector was redefined as shown in Equations 79, where B_f is the 3x3 matrix of the forearm's coordinate axes, B_n is an identity matrix, P is the product of B_n and the inverse of B_f, X_p is the upper arm *x*-projection, and V_c is the redefined vector used to check the sign.

$$P = B_n * {B_f}^{-1}$$

$$V_c = P * \begin{bmatrix} X_{px} \\ X_{py} \\ X_{pz} \end{bmatrix} \tag{79}$$

3.6.5 Wrist Flexion and Extension Calculation

To calculate the wrist flexion and extension, the dot product was used to find the angle between the hand vector projections and the forearm coordinate axes. Wrist flexion and extension was calculated as the angle between the projection of the hand *z*-axis and the forearm *z*-axis, as shown in Equation 80.

$$\theta_z = cos^{-1}(\overrightarrow{eZ_p} \cdot \overrightarrow{Z_f}) \tag{80}$$

The directions of the angles were determined by re-defining the base coordinate system for the projected vector to the forearm coordinate axes and the sign of the *x*-component of the vector was used to evaluate the sign of the rotation. If the sign of the *x*-component of the vector was positive, the rotation was extension and if the sign of the *x*-component was negative, the rotation was flexion. Using the right-hand rule and the axes sign convention, extension was considered a positive rotation about the *y*-axis and flexion

was considered a negative rotation. The projected vector was redefined, as shown in Equations 81, where B_f is the 3x3 matrix of the forearm's coordinate axes, B_n is an identity matrix, P is the product of B_n , and the inverse of B_f, Z_p is the hand z- projection, and V_c is the redefined vector used to check the sign.

$$P = B_n * {B_f}^{-1}$$

$$V_c = P * \begin{bmatrix} Z_{px} \\ Z_{py} \\ Z_{pz} \end{bmatrix} \tag{81}$$

3.6.6 Wrist Ulnar and Radial Deviation Calculation

To calculate the wrist ulnar and radial deviation, the dot product was used to find the angle between the hand vector projections and the forearm coordinate axes. Wrist ulnar and radial deviation was calculated as the angle between the projection of the hand z-axis and the forearm z-axis, as shown in Equation 82.

$$\theta_z = cos^{-1}(\overrightarrow{eZ_p} \cdot \overrightarrow{Z_f}) \tag{82}$$

The directions of the angles were determined by re-defining the base coordinate system for the projected vector to the forearm coordinate axes and the sign of the y-component of the vector was used to evaluate the sign of the rotation. If the sign of the y-component of the vector was positive, the rotation was radial and if the sign of the y-component as negative, the rotation is ulnar. Using the right-hand rule and the axes sign convention, ulnar deviation was considered a positive rotation about the x-axis and radial deviation was considered a negative rotation. The projected vector was redefined, as shown in Equations 83, where B_f is the 3x3 matrix of the forearm's coordinate axes, B_n is

an identity matrix, P is the product of B_n and the inverse of B_f, Z_p is the hand z-projection, and V_c is the redefined vector used to check the sign.

$$P = B_n * {B_f}^{-1}$$

$$V_c = P * \begin{bmatrix} Z_{px} \\ Z_{py} \\ Z_{pz} \end{bmatrix} \tag{83}$$

3.6.7 Tool Yaw Calculation

To calculate the tool yaw, the dot product was used to find the angle between the tool vector projections and the hand coordinate axes. Tool yaw was calculated as the angle between the projection of the tool z-axis and the hand z-axis as shown in Equation 84.

$$\theta_z = cos^{-1}(\overrightarrow{eZ_p} \cdot \overrightarrow{Z_h}) \tag{84}$$

The directions of the angles were determined by re-defining the base coordinate system for the projected vector to the hand coordinate axes and the sign of the x-component of the vector was used to evaluate the sign of the rotation. If the sign of the x-component of the vector was positive, the rotation was positive and if the sign of the x-component as negative, the rotation was negative. The positive and negative designations are done by using the right hand rule with the axis sign conventions of this study. The projected vector was redefined as shown in Equations 85, where $B_h$is the 3x3 matrix of the hand's coordinate axes, B_n is an identity matrix, P is the product of B_n and the inverse of B_h, Z_p is the tool z- projection, and V_c is the redefined vector used to check the sign.

$$P = B_n * {B_h}^{-1}$$

$$V_c = P * \begin{bmatrix} Z_{px} \\ Z_{py} \\ Z_{pz} \end{bmatrix} \tag{85}$$

3.6.8 Tool Pitch Calculation

To calculate the tool pitch, the dot product was used to find the angle between the vector projections and the hand coordinate axes. Tool pitch was calculated as the angle between the projection of the tool *z*-axis and the hand *z*-axis, as shown in Equation 86.

$$\theta_z = cos^{-1}(\overrightarrow{eZ_p} \cdot \overrightarrow{Z_h}) \tag{86}$$

The directions of the angles were determined by re-defining the base coordinate system for the projected vector to the hand coordinate axes and the sign of the *y*-component of the vector was used to evaluate the sign of the rotation. If the sign of the *y*-component of the vector was positive, the rotation is negative and if the sign of the *y*-component was negative, the rotation was positive. The positive and negative designations were done by using the right hand rule with the axis sign conventions of this study. The projected vector was redefined, as shown in Equations 87, where B_h is the 3x3 matrix of the hand's coordinate axes, B_n is an identity matrix, P is the product of B_n and the inverse of B_h, Z_p is the tool *z*- projection, and V_c is the redefined vector used to check the sign.

$$P = B_n * {B_h}^{-1}$$

$$V_c = P * \begin{bmatrix} Z_{px} \\ Z_{py} \\ Z_{pz} \end{bmatrix} \tag{87}$$

3.6.9 Tool Roll Calculation

To calculate the tool roll, the dot product was used to find the angle between the vector projections and the hand coordinate axes. Tool roll was calculated as the angle between the projection of the tool *x*-axis and the hand *x*-axis, as shown in Equation 88.

$$\theta_x = cos^{-1}(\overrightarrow{eX_p} \cdot \overrightarrow{X_h}) \tag{88}$$

The directions of the angles were determined by re-defining the base coordinate system for the projected vector to the hand coordinate axes and the sign of the *y*-component of the vector was used to evaluate the sign of the rotation. If the sign of the *y*-component of the vector was positive, the rotation was positive and if the sign of the *y*-component was negative, the rotation was negative. The positive and negative designations were done by using the right hand rule with the axis sign conventions of this study. The projected vector was redefined as shown in Equations 89, where B_h is the 3x3 matrix of the hand's coordinate axes, B_n is an identity matrix, P is the product of B_n and the inverse of B_h, X_p is the tool *x*- projection, and V_c is the redefined vector used to check the sign.

$$P = B_n * {B_h}^{-1}$$

$$V_c = P * \begin{bmatrix} X_{px} \\ X_{py} \\ X_{pz} \end{bmatrix} \tag{89}$$

3.7 Validation Protocols

Many mechanical and human tests were conducted in an attempt to validate the methods proposed in this study.

3.7.1 System Validation

In order to use the OEMC system for testing, its viability must be validated. The system's uncertainty was estimated using static and dynamic tests. The static test used two very small reflective markers placed six inches apart on a ruler, capturing them with the OEMC system, and calculating the distance between them to compare to the actual value.

In order to conduct the dynamic test, the two very small markers were first attached to a rubber band and the OEMC system captured the marker positions as the rubber band was stretched to two, three and four inches apart. These distances were monitored using a ruler that was placed underneath the rubber band. When trying to stretch the rubber band to exact distances, there is a level of human error involved with this dynamic test method. Very small markers were used because the OEMC system calculates the centroid of a marker and the calculation error would be very small. After these tests were conducted, the distances calculated were compared to the distances measured.

3.7.2 Mechanical Testing

In order to validate the planar projection calculation methods, especially for the shoulder, a static mechanical test was conducted. The test used a plywood board with

three markers on it that represented a marker set from the upper arm and another three-marker board that represented the torso markers as seen in Figure 3.15. The torso plane was originally placed perpendicular to the upper arm plane while the upper arm plane was attached to an adjustable ball and socket joint that allowed for the upper arm plane to be set in any rotation combination to represent flexion or extension, abduction or adduction, and medial or lateral rotation. The wooden upper arm was captured in static postures using combinations of 0, 30 and 60 degrees for the three shoulder rotations, which were measured using a manual goniometer. The calculation technique was applied to these recordings and the results were compared to the measured angles.

a) b)

Figure 3.15: Mock-up marker configuration of the a) upper arm segment and b) torso segment

3.7.3 Rotation Direction Validation

The method for re-defining the coordinate system of a vector, described in Section 3.6, was tested before use by applying the algorithm to several vectors and comparing the results of the algorithm to the expected values. This test found the method to be accurate.

3.7.4 Human Testing

In order to verify the methods designed, human testing was required to test how accurately the methods calculated posture angles on actual subjects.

Initially, the effect of hip flexion on shoulder posture calculations was tested by setting the subject's shoulder posture to known angles and altering their hip flexion, where the angles were set using a goniometer. The subject started in a neutral standing posture and was instructed to bend forward at the waist as far as possible while not moving their upper arm with respect to their torso. The shoulder posture calculations were expected to show minimal change as the subject's hip flexion changed because the markers on the torso should ensure that the shoulder rotation calculations only change as the upper arm moves with respect to the torso.

Shoulder flexion calculation method was tested by starting the subject in a specified shoulder posture with a known flexion and having the subject lower their arm to a neutral posture. This was done with six different combinations of shoulder abductions and lateral rotations. The shoulder flexion plots were expected to start at the initial flexion level and end at a neutral posture.

Shoulder rotation calculation method was tested by starting the subject in a specified shoulder posture with a known medial rotation and having the subject rotate to a neutral rotation posture. The shoulder medial and lateral rotation plot were expected to begin at the starting posture and end at a neutral rotation posture.

Figure 3.16 is the experimental setup for the shoulder rotation test. In both shoulder studies, a wooden board marked with measured lines from 0 to 90 degrees in 10 degree increments was used to aid the subject and improve accuracy. This allowed the subject to visualize how to place their arm and what lines to follow.

Figure 3.16: Experimental setup for the shoulder rotation test

Forearm rotation, wrist flexion/extension and ulnar/radial deviation calculation methods were tested by capturing a subject's movement from a neutral posture to their range of motion in each direction. The subject's range of motion in these four postures was measured using a goniometer. The calculations were expected to yield the same maximum and minimum as the measured values.

3.8 Application

The methods described in this study are applied to OEMC data pertaining to laparoscopic device use in a simulated environment. A manual and a powered device were used by 10 subjects that were fit with the marker configuration described in this study and captured using each device nine times while performing a simulated surgical task on a tissue surrogate made out of foam. Of the nine tasks, six were performed in a neutral arm posture and three in an awkward arm posture. The neutral arm postures were meant to represent a standard position that a surgeon could face during tool use. The awkward posture was meant to represent an extreme scenario that surgeon may experience during tool use. The simplified marker scheme was essential in this study because the subjects are also fit with force and surface EMG sensors. Postural calculations were made for all trials including threshold testing and RULA scores as described in the following sections (3.9 and 3.10).

3.9 Threshold Testing

In order to quantify intensity of biomechanical risk exposure due to posture, a threshold calculation was applied to the angular displacement plots (Peterson, 1999). The PATH conventions defined by Buchholz et al. (1996) incorporate posture thresholds for different joint rotations. They hypothesize that postures beyond these thresholds carry biomechanical risks that could lead to neuromuscular injury. Their defined thresholds are shown in Table 3.1. For the angular displacement of each joint, the areas below the curve beyond the thresholds were calculated. These areas carry units of angle*time and are used as a comparison between trials. In addition, the percentage of task time spent

beyond threshold was calculated and could be used to interpret the duration of biomechanical risk exposure during a task due to posture. In addition, the percentage of time at risk was compared to the results of the preliminary study involving PATH, for contrast.

Table 3.1: Posture angle thresholds for joints considered

Joint Rotation	Threshold
Shoulder Flexion	$>60^0$
Shoulder Extension	$>60^0$
Shoulder Abduction	$>60^0$
Shoulder Adduction	$>60^0$
Forearm Pronation	$>45^0$
Forearm Supination	$<45^0$
Wrist Ulnar Deviation	$>15^0$
Wrist Radial Deviation	$>15^0$
Wrist Flexion	$>15^0$
Wrist Extension	$>15^0$

The threshold method was verified by applying it to one period of a vertically-shifted sine wave,

$$[\sin\theta + 10]_0^{2\pi}, \tag{90}$$

and comparing the predicted and calculated results. A value of 10 was added to the sine wave in order to test for a threshold of 10, where the results expected to be

greater than 10 from 0 to π. The expected area and percent area of the sine wave were found using Equations 91 and 92 and, based on theory, was expected to yield an area of 2 radians, a percent time of 50% and a percent area of 3.18%.

$$\int_0^{\pi} sin(\theta)\, d\theta \tag{91}$$

$$Area = 2 / (10 * 2\pi) \tag{92}$$

3.10 RULA Posture Scores

In order to interpret the results for biomechanical assessment, the RULA (Hedge, 2000) hand and wrist posture scores were calculated using four main factors: upper arm, lower arm, wrist and wrist twist. Upper arm factor was dependent on the shoulder flexion/extension and adduction/abduction calculations and can range from one to six. The lower arm factor is dependent on elbow flexion/extension calculations and can range from one to three. Wrist factor is dependent on wrist flexion/extension and ulnar/radial deviation calculations and could range from one to four. Wrist twist was dependent on forearm rotation calculations and can be one or two. The four factors were used to look up the RULA posture score in Table 3.2 and the exact criteria for RULA scoring can be seen in Appendix K.

Table 3.2: RULA arm and wrist posture score using the upper arm, lower arm, wrist and wrist twist factors (Hedge, 2000)

RULA Arm and Wrist Posture									
		Wrist							
		1		2		3		4	
Upper Arm	Lower Arm	Wrist Twist		Wrist Twist		Wrist Twist		Wrist Twist	
		1	2	1	2	1	2	1	2
1	1	1	2	2	2	2	3	3	3
	2	2	2	2	2	3	3	3	3
	3	2	3	2	3	3	3	4	4
2	1	2	2	2	3	3	3	4	4
	2	2	2	2	3	3	3	4	4
	3	2	3	3	3	3	4	4	5
3	1	2	3	3	3	4	4	5	5
	2	2	3	3	3	4	4	5	5
	3	2	3	3	4	4	4	5	5
4	1	3	4	4	4	4	4	5	5
	2	3	4	4	4	4	4	5	5
	3	3	4	4	5	5	5	6	6
5	1	5	5	5	5	5	6	6	7
	2	5	6	6	6	6	7	7	7
	3	6	6	6	7	7	7	7	8
6	1	7	7	7	7	7	8	8	9
	2	7	8	8	8	8	9	9	9
	3	9	9	9	9	9	9	9	9

4 Results

4.1 System Validation Tests

The system validation was done to ensure that any errors found in this method can be attributed to the method and not by errors in tracking markers with the OEMC system.

The static system validation test was conducted by attaching two reflective markers 6 inches apart on a ruler and capturing them with the OEMC system for about five seconds. The average distance between the markers was calculated to be 6.00 +/- 0.01 inches. The error was equivalent to 0.0321millimeters, which is less than the 0.150 millimeters that was the target during calibration.

The dynamic system validation test was conducted by attaching two markers to a rubber band and capturing the stretching of the rubber band so that the markers are two, three and four inches apart which was measured using a ruler. The average distance between the markers, while the rubber band was held at 2, 3, and 4 inches is 2.054, 3.029, and 4.048 inches, respectively. This means that the average error was 1.11 millimeters.

4.2 Mechanical Testing

The shoulder calculations were tested using a mechanical method. The plywood triangle was rotated about its three axes in all combinations of 0, 30 and 60 degrees. The difference between the measured and calculated angles was found in order to further understand the inaccuracy of this shoulder posture calculation method. Figure 4.1 shows the difference of each trial broken down into nine subplots. Table 4.1 shows the different combinations of rotations, regardless of order, and the difference between measured and calculated results associated with them. When one or fewer rotations were made, the average difference was 3.53 degrees. When there were two rotations made, the average

differences increased as the magnitudes increase from two 30 degree rotations to two 60 degree rotations. When there were three rotations made, the average differences increased as the magnitudes increased from three 30 degree rotations to three 60 degree rotations.

There was a level of uncertainty that was associated with this test that originates from the human element of setting up the segments into the postures and from the accuracy of the goniometer measurements. The uncertainty associated with the human error was estimated to be about 5 degrees and the uncertainty associated with the goniometer measurement was estimated to be about 1 degree. The square root of the sum of the squares of all the contributors to the uncertainty yields an estimated total of 5 degrees. The differences between the measured and calculated results were calculated. The differences were found to be within 1 degree or less of the estimated uncertainty when there were no 60 degree rotations.

Table 4.1: Different combinations of rotations influence differences between measured and calculated angles

Rotation Combinations	Average Difference (Degrees)
One or fewer rotations	3.53
Two 30 and one 0 degree rotations	4.38
Three 30 degree rotations	5.82
Two 30 and one 60 degree rotations	11.16
Two 60 and one 30 degree rotations	12.24
One 30 , one 60 and one 0 degree rotation	14.71
Two 60 and one 0 degree rotations	15.11
Three 60 degree rotations	16.57

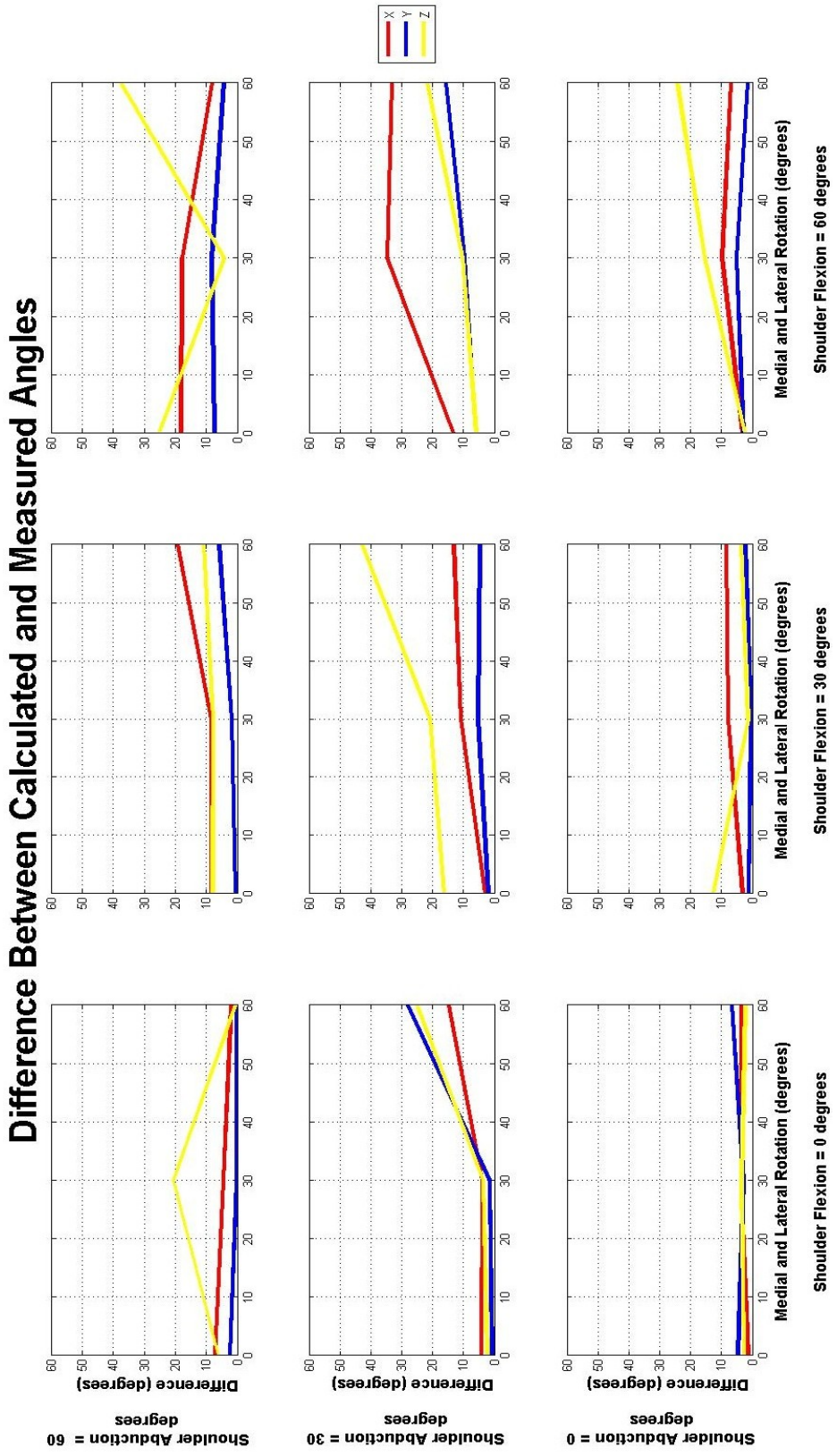


Figure 4.1: Differences between measured and calculated angles are plotted for all combinations of 0, 30 and 60 degrees for flexion, abduction and rotation

4.3 Human Testing

4.3.1 Effect of Hip Flexion on Shoulder Measurements

The purpose of defining the orthogonal axis of the torso was to project the axes of the upper arm onto its planes for shoulder calculations. By projecting onto the torso planes, as opposed to global planes, the subject's orientation with respect to the global planes does not matter and allows for free movement of the subject during capture without an effect on the outcome of shoulder calculations. When the subject changes their torso orientation with respect to the global coordinates, while keeping the upper arm stable with respect to the torso, the shoulder posture should remain static. In this test, the subject started at a neutral hip posture and bent forward at the hip as far possible while trying not to vary their upper arm posture. The shoulder flexion in this test went from 10 to 60 degrees in 10 degree increments and the change in shoulder postures was plotted for these six trials in Figure 4.2. The average absolute changes in shoulder flexion, adduction, and medial rotation were 5.4, 1.7, and 1.8 degrees, respectively. The greatest changes in flexion, adduction, and rotation were 20.2, 7.8, and 12.6 degrees, respectively.

A level of uncertainty was present and likely due to the change due to the subject's inability to maintain the measured shoulder posture during the test and the human error in positioning the upper arm segment. The uncertainty associated with the error of placing the segment in a certain posture was estimated to be about 5 degrees, while the uncertainty associated with the subject's inability to maintain a proper posture during extreme hip flexion was estimated to be about 10 degrees and about 1 degree for the goniometer measurement. The square root of the sum of the squares of all the

contributors to the uncertainty yields an estimated total uncertainty of about 12 degrees. The average changes in shoulder rotations were all lower than the estimated uncertainty of 12 degrees. The highest abduction and rotation differences were within one degree or less of the estimated uncertainty, while the greatest difference in the flexion was about 8 degrees beyond the uncertainty.

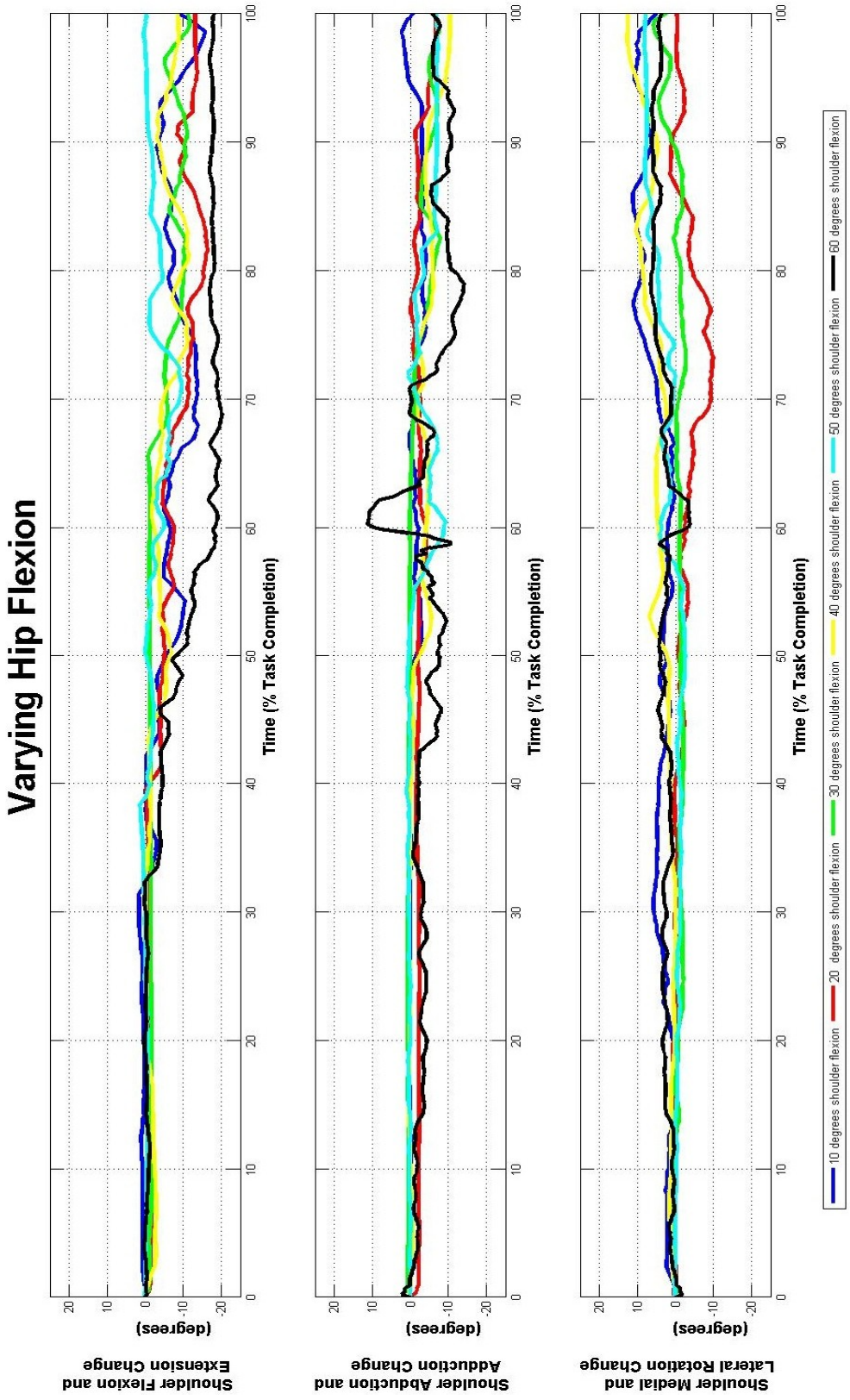


Figure 4.2: Change in shoulder postures as the hip flexion changes

4.3.2 Shoulder Flexion with Varying Rotation and Abduction

In order to validate the method of shoulder flexion and extension calculations, a second human test was performed to compare a known shoulder flexion motion with different shoulder rotations and abductions. In each trial, the shoulder flexion went from -60 degrees to a neutral posture of about -10 degrees (measured using goniometer) while starting at different rotations and abductions. By starting at different postures, the effect of changing the shoulder rotation and abduction on the shoulder flexion measurement was observed. The shoulder flexion measurement results of the six trials are shown in Figure 4.3. The profiles of all the plots go from about -60 to -10 degrees and are all within 8 degrees of -60 and 4 degrees of -10.

The level of uncertainty in this test was due to inaccuracy in upper arm segment setup and the subject's inability to maintain the exact postures they are put in. The uncertainty associated with the error of placing the segment in a certain posture was estimated to be about 5 degrees, the uncertainty associated with the subject's inability to maintain a proper posture while changing shoulder flexion was estimated to be about 5 degrees, and about 1 degree for the goniometer measurement. The square root of the sum of the squares of all the contributors to the uncertainty yields an estimated total uncertainty of about 7 degrees. The results are within 1 degree or less of the estimated 7 degree uncertainty.

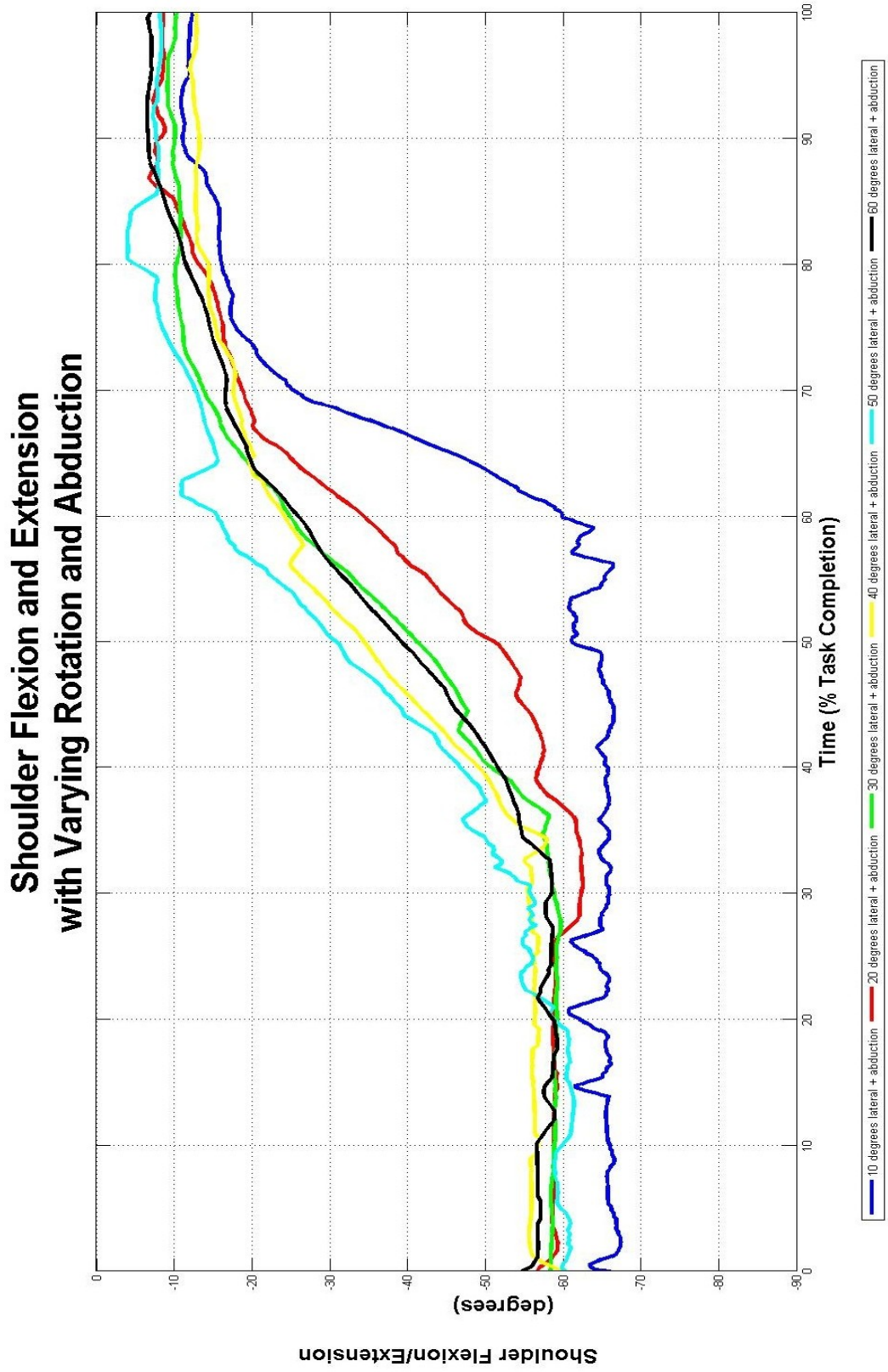


Figure 4.3: Shoulder flexion from -60 to -10 degrees for varying rotation and abduction

4.3.3 Shoulder Medial and Lateral Rotation Test

In order to test the calculation method of shoulder medial and lateral rotation, a third human test was conducted, where the subject rotated their shoulder from 90 to 0 degrees of medial rotation using a wooden board with angle markers on it. The trial was done three times and the shoulder rotation was calculated and plotted in Figure 4.4. The calculated shoulder rotation was expected to start at 90 degrees and come down to 0. During the motion from 90 to 0 degrees, the calculations didn't start following the expected trend until the subject was below 60 degrees. This was due to the offset marker of the upper arm being mounted on the tricep, which has limited movement during humeral rotation.

The level of uncertainty caused by the inability of the subject to accurately achieve the desired postures was estimated to be about 7 degrees and all three trials were observed to be within the 7 degrees estimated uncertainty at the end of the trial (i.e., greater than 80 percent task completion as seen in Figure 4.4) .

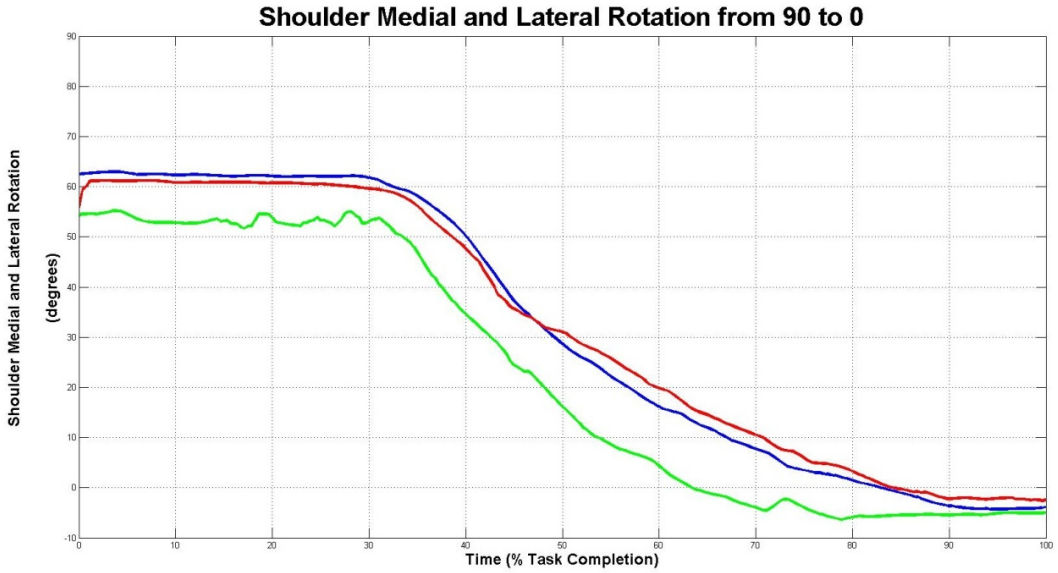


Figure 4.4: Shoulder rotation calculation for 90 to 0 degree rotation

4.3.4 Forearm Rotation Test

The forearm rotation calculation methods were tested by conducting a fourth human test where a subject was captured as they start from neutral, pronate to their range of motion, and then supinate to their range of motion. The results of the three trials conducted are shown in Figure 4.5 and the measured range of motion for the subject was 75 degrees of supination and 64 degrees of pronation. The average calculated values were 73 degrees of supination and 60 degrees of pronation and the difference between the measured and calculated values of supination and pronation were 2 and 4 degrees, respectively.

The uncertainty of this test was associated with the range of motion measurements and tool use and was estimated to be about 3 degrees for range of motion measurement and about 1 degree for the goniometer. The square root of the sum of the squares of all the contributors to the uncertainty yielded a total uncertainty of about 3 degrees and the results were within 1 degree of the estimated uncertainty.

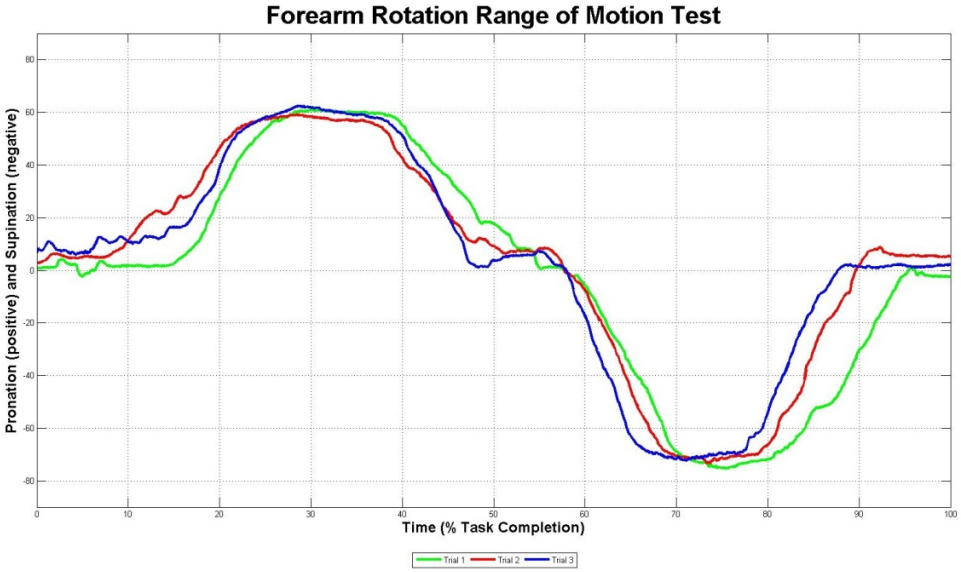


Figure 4.5: Forearm rotation range of motion

4.3.5 Wrist Flexion and Extension Test

The wrist flexion and extension calculation methods were tested by conducting a fifth human test, where a subject was captured as they start from a neutral wrist posture, flex their wrist as far as possible, and then extend their wrist as far as possible. The wrist flexion and extension was calculated and compared to the subject's ranges of motion measured using a manual goniometer. The results of the three trials, Figure 4.6, show a maximum flexion of about -57 degrees and a maximum extension of about 58 degrees while the measured range of motion was about -56 degrees of flexion and 55 degrees of extension. The differences between the measured maximum flexion and extension and the calculated were 1 and 3 degrees, respectively, which were within the estimated uncertainty of 3 degrees.

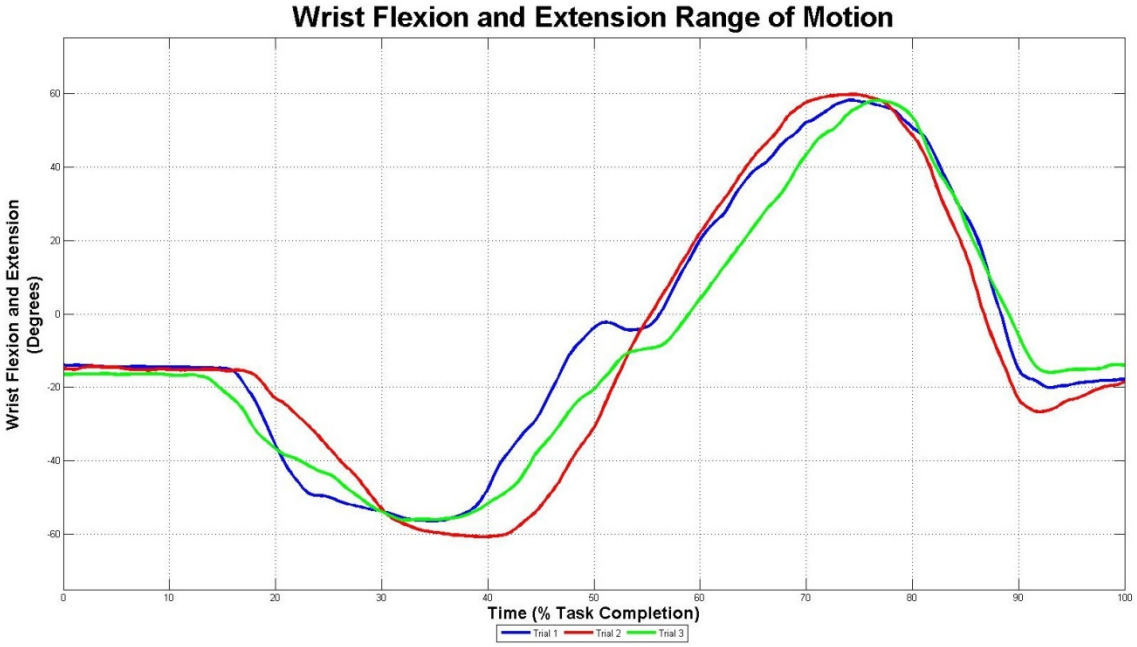


Figure 4.6: Wrist flexion and extension range of motion

4.3.6 Wrist Ulnar and Radial Deviation

The wrist ulnar and radial deviation calculation methods were tested with a sixth human test, where a subject as captured as they start from a neutral wrist posture, radially-deviate their wrist as far as possible, and then ulnarly-deviate their wrist as far as possible. The wrist ulnar and radial deviation is calculated and compared to the subject's ranges of motion measured using a goniometer. The results of the three trials, Figure 4.7, show a maximum radial deviation of about -14 degrees and a maximum ulnar deviation of about 29 degrees. The measured range of motion was about -18 degrees of radial deviation and 29 degrees of ulnar deviation. The differences between the measured maximum ulnar and radial deviations and the calculated were 4 and 0 degrees, respectively, which are within 1 degree or less of the estimated uncertainty of 3 degrees.

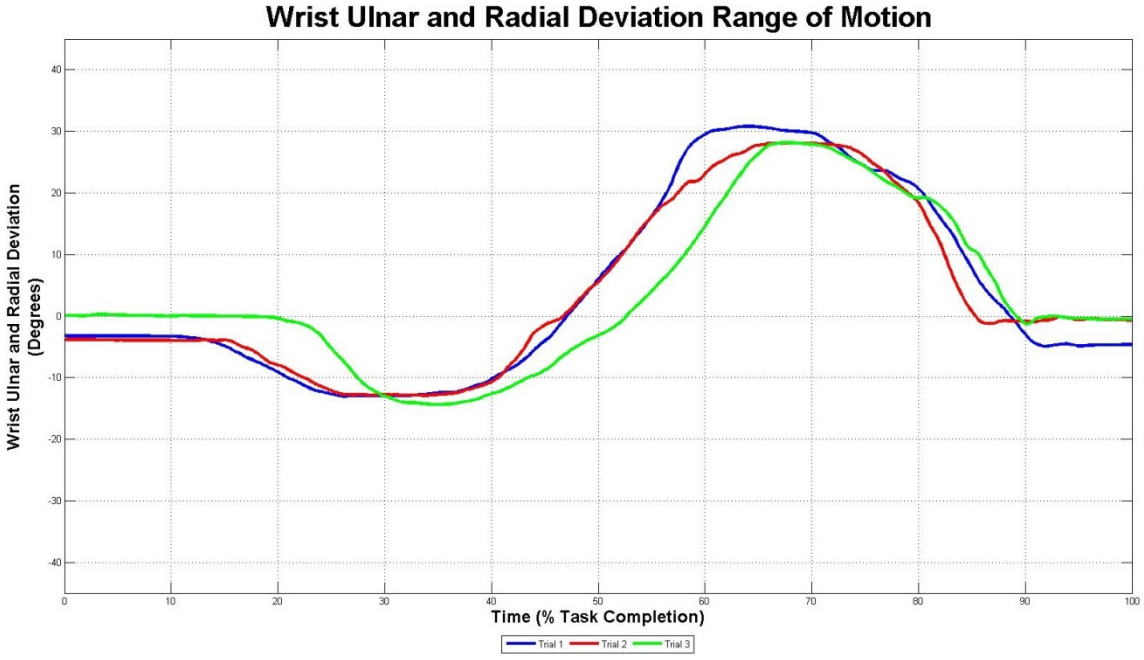


Figure 4.7: Wrist ulnar and radial deviation range of motion

4.4 Threshold Validation

In order to validate the threshold method, a sine wave (Equation 90) was centered about 10 on the vertical axis. From the sine wave, the mathematically-predicted area beyond threshold was 2 radians, the percent of time beyond threshold was 50%, and the percent area beyond threshold was 3.18%. When applying the threshold technique to the sine wave, the results of the test, shown in Table 4.2, were observed to match the mathematically-predicted values.

Table 4.2: Shows the results of the threshold test on a sine wave

Area Beyond Threshold	Percent Time	Percent Area
2.00 radians	49.92%	3.18%

4.5 Application

The methods presented to calculate joint motion of the upper extremity and tool orientation were applied to data collected on subjects during laparoscopic tool use and the threshold methodology was applied to the joint rotation data. The same thresholds were used in these calculations as for the PATH preliminary study for direct comparison. Posture calculations were split into two main categories: neutral and awkward positions. The average wrist threshold results for the neutral postures are shown in Table 4.3. The data of surgical tool use in a neutral arm position was compared to the PATH results, since the subjects were doing similar tasks in similar positions.

The results of the calculations indicate that the shoulder of the subject was in a neutral posture (i.e. all anatomical shoulder angles less than 60 degrees) 100 percent of the time for both the powered and manual devices. The PATH results indicated neutral

shoulder postures for 97.1 and 91.7 percent of the time for the powered and manual devices, respectively, which appear to be similar.

The wrist posture angle calculations indicate that the flexion and extension rotation of the wrist was beyond threshold for 46.62 and 15.94 percent of the time for the manual and powered devices, respectively, while the PATH results indicated that the flexion and extension rotation of the wrist was beyond threshold 25.0 and 11.1 percent of the time for the manual and powered devices, respectively. The calculated and PATH results for the powered device were about 20 percent different while the manual device results show a difference of about 5 percent.

The calculated results of ulnar and radial deviation of the wrist indicated that the wrist was beyond threshold for 22.39 and 53.38 percent of the time for the manual and powered devices, respectively, while the PATH results indicated that the wrist was beyond threshold 25.0 and 11.1 percent of the time for the manual and powered devices, respectively. There was about a 42 percent discrepancy between the PATH and calculated results for the powered device, with about a 3 percent difference in the results for the manual device.

Table 4.4 shows the threshold calculation results for forearm rotation. The PATH results found the forearm to be beyond threshold for 13.9 and 41.7 percent of the time for powered and manual devices, respectively, while the calculated percent of time beyond threshold was 2.20 and 5.90 percent for the powered and manual devices, respectively.

Table 4.3: Wrist threshold data for the neutral position trials

	Wrist - Neutral Position	
	Area Beyond Threshold (degrees*seconds)	
	Wrist Flexion/Extension	Wrist Ulnar/Radial Deviation
Manual	161.77	76.29
Power	22.08	76.51
	Percentage of Task Time Beyond Threshold	
	Wrist Flexion/Extension	Wrist Ulnar/Radial Deviation
Manual	46.62 %	22.39 %
Power	15.94 %	53.38 %

Table 4.4: Forearm rotation threshold calculations

	Forearm - Neutral Position	
	Area Beyond Threshold (degrees*seconds)	Percentage of Task Time Beyond Threshold
Manual	51.00	5.90 %
Power	1.73	2.20 %

The awkward position posture results for the shoulder, wrist and forearm are shown in Tables 4.5, 4.6, and 4.7. The percentage of time beyond threshold was expected to increase in the awkward position. The percentage of time the shoulder is abducted or adducted beyond threshold was found to increase for the manual device,

while the other two rotations remained the same. The percentage of time the shoulder was abducted or adducted and medially or laterally rotated increased for the powered device in an awkward position.

For the manual and powered device, the percentage of time the wrist was flexed or extended beyond threshold increased in the awkward position. The percentage of time the wrist was in ulnar or radial deviation beyond threshold for the manual device slightly increased in an awkward position and decreased for the powered device. For the forearm rotation, the manual device did not change much while the powered device increased by about 7 percent.

Table 4.5: Area and percentage of task time beyond threshold for the three shoulder rotations in the awkward position

	Shoulder - Awkward Position		
	Area Beyond Threshold (degrees*seconds)		
	Shoulder Flexion/Extension	**Shoulder Abduction/Adduction**	**Shoulder Medial/Lateral Rotation**
Manual	3.35	20.04	0.00
Power	113.49	117.06	87.86
	Percentage of Task Time Beyond Threshold		
	Shoulder Flexion/Extension	**Shoulder Abduction/Adduction**	**Shoulder Medial/Lateral Rotation**
Manual	7.60 %	27.35 %	0.00 %
Power	16.80 %	28.24 %	7.69 %

Table 4.6: Area and percentage of task time beyond threshold for the two wrist rotations in the awkward position

	Wrist - Awkward Position	
	Area Beyond Threshold (degrees*seconds)	
	Wrist Flexion/Extension	Wrist Ulnar/Radial Deviation
Manual	37.66	7.50
Power	281.46	83.87
	Percentage of Task Time Beyond Threshold	
	Wrist Flexion/Extension	Wrist Ulnar/Radial Deviation
Manual	39.82 %	14.95 %
Power	72.42 %	52.94 %

Table 4.7: Threshold calculations for forearm rotation in an awkward posture

	Forearm - Awkward Position	
	Area Beyond Threshold (degrees*seconds)	Percentage of Task Time Beyond Threshold
Manual	4.93	4.96 %
Power	71.42	9.74 %

4.6 RULA Posture Scores

The RULA posture scores were calculated for all trials and the average scores were found for all trials of each device, awkward positions of each device, and neutral positions of each device (results are shown in table 4.8). The powered device had a greater RULA posture score than the manual device overall and for both positions. The RULA posture scores of both devices were lower for the neutral positions than for the

awkward positions. The devices used weighed less than 2 kilograms, which equated to a RULA force/load score of 0, while the posture was mainly static and the action repeatedly occured four times per minute, which equates to a RULA muscle use score of 1. A table lookup yielded a final wrist and arm score of 3 or 4. Since the neck was generally in a neutral posture, the trunk position was between 0 and 20 degrees, and the legs and feet were supported and balanced, the trunk posture score was 2. Since the posture was held static and the action occured four times per minute, the final neck, trunk, and leg score was 3. This yielded a final RULA score of 3 for the laparoscopic surgical tasks, which indicates that further investigation should be done.

Table 4.8: RULA arm and wrist posture scores by device and position

RULA Arm and Wrist Posture Scores	
Overall Averages by Device	
Manual	2.50
Powered	2.94
RULA Arm and Wrist Posture Scores	
Averages By Position	
Neutral Positions	
Manual	2.30
Powered	2.73
Awkward Positions	
Manual	3.00
Powered	3.41

5 Discussion

5.1 Validation Results

The threshold method validation test showed that the threshold calculation method is accurate. The static system validation test indicated that the system is accurate to 0.0321 millimeters, which is lower than the calibration uncertainty. The dynamic system validation test was accurate to 1.11 millimeters. These differences may be attributed to test uncertainty and not a system error due to the dynamic nature of the test.

5.2 Mechanical Testing

The results of the mechanical test, shown in Table 4.1, indicate that there was an error associated with multiple large rotations. When there were one or fewer rotations of any amount or there were two rotations of 30 degrees, the average difference was within 1 degree of the uncertainty estimate. The values of the rotations appear to have influenced the average difference more than the number of rotations, which was evident where three 30 degree rotations had a lower average difference than two 60 degree rotations. The mechanical test results showed that there is a limitation to the method; however, the errors appeared to be lower than the errors associated with the Euler method calculated using a similar test by Coates (2007).

5.3 Human Testing

One of the goals of this modified approach was to allow for free movement of the trunk with respect to the global coordinate system without effecting the shoulder calculations. The hip flexion test was conducted in order to evaluate the method's success, where results showed no apparent pattern to the small discrepancy in subject

shoulder posture as the hip flexion changes. After analyzing the visual recording, the small discrepancy appeared to be due to the subject's inability to maintain a shoulder posture while moving, which would indicate that the hip flexion did not appear to have an effect on shoulder posture calculations. The average results were within the estimated uncertainty.

The shoulder flexion test with varying rotation and abduction indicated that the shoulder flexion was calculated to be accurate within one degree of the uncertainty for all combinations of abduction and lateral rotation. This difference can likely be attributed to the human error limitations of the test.

The shoulder medial and lateral rotation test showed that the medial/lateral rotation of the upper arm could not be accurately calculated beyond 60 degrees. Through visual observation, this error appeared to be a product of the offsetting upper arm marker moving with the skin and muscles of the triceps and not the bone below. This error showed limitations in the calculation method.

The forearm rotation range of motion test showed that the calculated results were within 4 degrees of the measured results. The wrist flexion/extension and ulnar/radial deviation tests showed that the wrist calculation method can be accurate to within 4 degrees of the range of motion. These differences were within 1 degree or less of the uncertainty. Some of these differences may have been due to uncertainty in range of motion measurement.

5.4 Application

For the shoulder postures, the OEMC and PATH results were within 10 percent of each other for the manual and powered devices. For the wrist flexion and extension, the results were within 10 percent of each other for the manual and powered devices. For the wrist ulnar and radial deviation, the results were within 5 percent for the manual device and 33.9 percent different for the powered device. Given the accuracy found in the OEMC wrist calculation method, the PATH method's subjective measurement technique may be the cause in the great discrepancy. It is very difficult for an observer to identify an anatomical rotation as greater, or less, than 15 degrees. This inaccuracy could lead to misevaluation of the biomechanical risk associated with this task. In the awkward posture, the percent of time at risk and the intensity of the posture increased for the wrist and shoulder. This was to be expected and showed that the OEMC calculation method was able to capture this.

This method of interpreting OEMC data appeared to be capable of analyzing upper extremity posture for biomechanical risk during hand tool use, can be applied to the use of other tools, and can be applied in conjunction with other biomedical instrumentation. The simplified marker configuration allows for space on the subjects upper extremities in order to use other instrumentation such as surface electromyography (EMG) and force sensors. This allows for a more comprehensive analysis of biomechanical risk factors.

By applying the RULA scoring techniques to the calculated postures, the occupational biomechanical risk can be evaluated at a more objective and precise level.

This can be a powerful tool in the field of occupational health where RULA is used for evaluation. RULA's subjectivity and lack of accuracy is a limiting factor in its capability to properly analyze the biomechanical risk associated with occupational tasks. Using OEMC methods in conjunction with other biomechanical measurements provides a much more accurate and precise method to the RULA threshold scoring technique.

5.5 Future Investigation

The results of this study suggested that the vector planar projection method can be a viable method for calculating some upper extremity postures including shoulder abduction/adduction, flexion/extension, forearm rotation, wrist flexion/extension and wrist ulnar/radial deviation. Alternatively, results indicated that the medial/lateral rotation cannot be calculated accurately using this marker configuration. The study also suggested that there are errors associated with this method that occur when there are multiple rotations that approach 60 degrees. One direction for future investigation is to improve the accuracy of the method. A change in marker placement to more accurately represent body segment planes and planes to project onto should improve the calculations.

A novel method for dealing with the errors that occur when there are multiple large rotations needs to be investigated. The error occurs due to vector projection inaccuracies that occur as the size of the projection decreases. A correcting factor should be investigated based on the angle errors and vector projections. It may be found that vector projection inaccuracies occur on a consistent scale. If this is found to be the case, a correcting factor could be implemented in order to reduce these errors.

A major limitation of comparing joint angles was that the neutral postures and ranges of motion are different for all subjects. Different neutral postures and ranges of motion would mean that not all subjects incur comparable biomechanical risk, so using thresholds to compare biomechanical risk based off of the same angles is limited. The effect of different neutral angles and ranges of motion on biomechanical risks needs to be investigated, in order to create a method for quantifying this risk. EMG, which measures muscle activity, utilizes the percent of maximum voluntary contraction (MVC) to represent the muscle activity because muscle activity capabilities are different for all subjects. A similar technique should be developed for assessing postures because tendon, ligament and nerve pressure may be more influenced by the percent of range of motion of a subject than by just the angle.

6 Conclusion

By reviewing the literature on joint angle representation and body segment orientation and movement calculation, it became apparent that there is a need for further development of a method for calculating upper extremity posture. The advantages of the planar projection method are that it yields angles with anatomical meaning, its calculations are simple and not convoluted, it does not have Gimbal lock issues, and it is more accurate at, and beyond, 30 degrees of rotation than the other methods. The method is limited in its accuracy when the vectors being projected are rotated away from the planes they are to be projected onto and when there are large rotations in multiple directions.

The main goal of this study was to modify the planar projection method, in order to calculate upper extremity posture and to reduce the error of the angles. The first modification was the method by which anatomical planes were defined. In this study, the anatomical planes were defined by, first, using three markers to define orthogonal axes on the chest and then recording a neutral capture to determine how to adjust these orthogonal axes to account for the slope of the chest and accurately represent the anatomical planes. These planes were used for projecting vectors of the upper arm and calculating shoulder joint rotations and allowed for free movement of the subject without disturbing the projections. This eliminated the need to calculate the angle errors due to rotation of the reference vector with respect to the global plane as Peterson (1999) introduced. This was tested and its success is reported.

The overall viability of the method and its accuracy is evaluated using mechanical and human testing. The wrist and forearm calculations were found to be accurate while the shoulder calculations were found to be inaccurate when there are multiple rotations approaching 60 degrees. The human and static tests showed that angles are inaccurate when there are multiple 60 degree rotations. The calculation of shoulder medial and lateral rotations does not appear to be feasible due to the marker setup limitations.

The methods were applied to data recorded for laparoscopic surgeons during laparoscopic surgical tasks. A preliminary study analyzed similar tasks using the PATH method, which yielded the percent of time the subject is beyond a threshold. Threshold calculations were made for the OEMC posture data and the two were compared. This comparison found that they matched up very closely for shoulder postures, but not for wrist postures. This suggests that subjective methods were not capable of quantifying biomechanical risk associated with posture with the same precision as a quantitative method. A novel method for quantifying biomechanical risk was presented where calculated postural data was used to calculate the average RULA scores for the different tasks. The RULA scores for the surgical tasks were calculated to be about 3, which is representative of a hand tool task. The distribution of RULA scores between the manual and powered devices agreed with the results of thresholding technique.

Ultimately, this study has made some successful alterations to a known method to increase accuracy and reduce the calculated error. In addition, the method was applied to data of hand tool use and is compared to results of a subjective study of the same tasks. Additionally, a novel method for quantifying biomechanical risk was introduced.

References

Buchholz B, Paquet V, Punnett L, Lee D, Moir S. PATH: A work sampling-based approach to ergonomic job analysis for construction and other non-repetitive work. Applied Ergonomics 1996. 27.3: 177-187.

Chao EYS. "Justification of triaxial goniometer for the measurement of joint rotation." Journal of Biomechanics. 13 (1980): 989-1006.

Chen CT. Linear System Theory and Design. Third Edition. Oxford University Press. New York, NY. 1999.

Cheng PL, Nicol AC, Paul JP. "Determination of axial rotation angles of limb segments – a new method." Journal of Biomechanics. 33 (2000): 837-843.

Coates JO. "Using a simplified marker configuration to determine shoulder orientation." M.S. Thesis. University of Connecticut, 2007.

Dura JV, Forner A, Garcia AC, Ferrandis R, Brizuela G. "Joint co-ordinate system and attitude vector: influence in the interpretation of movements."

Hedge, Alan. "RULA Employee Assessment Worksheet". Cornell University, November 2000.

Lee G and Park AE. "Development of a more robust tool for postural stability analysis of laparoscopic surgeons." Surgical Endoscopy 22 (2008): 1087-1092.

Lee G, Sutton E, Clanton T, Park A. "Higher physical workload risks with NOTES versus laparoscopy: a quantitative ergonomic assessment." Surgical Endoscopy. (2010)

Maths – Projections of lines on planes. 2010. September 2011. <http://www.euclideanspace.com/maths/geometry/elements/plane/lineOnPlane/index.htm>

Moon, Francis. Applied Dynamics: with Applications to Multibody and Mechatronic Systems. John Wiley & Sons, Inc. New York, NY. 1998.

NaturalPoint Tracking Tools Users Manual: Getting Started. Version 2.0.

Person JG, Hodgson AJ, Nagy AG. "Automated high-frequency posture sampling for ergonomic assessment of laparoscopic surgery." Surgical Endoscopy. 15(2001): 997-1003.

Peterson, DR. "A Method for Quantifying the Biodynamics of Abnormal Distal Upper Extremity Function: Application to Computer Keyboard Typing." Ph.D. Dissertation. University of Connecticut, 1999.

Peterson DR, Bronzino JD. Biomechanics Principles and Applications. CRC Press. Boca Raton, FL. 2008.

Peterson DR, Cherniack MG. "Repetitive impacts from manual hammering: Physiological effects on the hand-arm system." Canadian Acoustics. 12(2001).

Peterson DR, Cherniack MG. "Vibration, grip force, muscle activity, and upper extremity movement of a manual hammering task." Proc. Intl Conf. Hand-Arm Vibration. Nancy, France, 2001.

Tupling SJ and Pierrynowski MR. "Use of cardan angles to locate rigid bodies in three-dimensional space." Medical and Biological Engineering and Computing. 26(1987): 527-532.

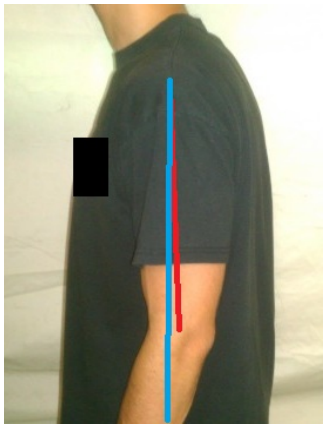
White AA, Panjabi MM, Brand RA. "A system for defining position and motion of the human body parts." Medical and Biomedical Engineering. (1975): 261-265.

Woltring HJ. "3-D Attitude representation of human joints: A standardization Proposal." Journal of Biomechanics. 27.12(1994): 1399-1414.

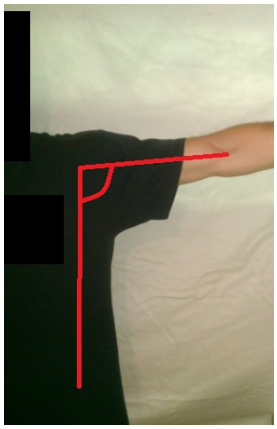
APPENDIX A: Joint Rotations

Shoulder

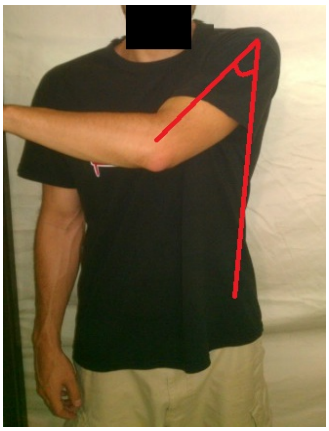
Neutral



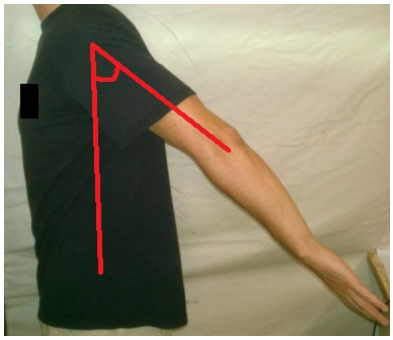
Abduction



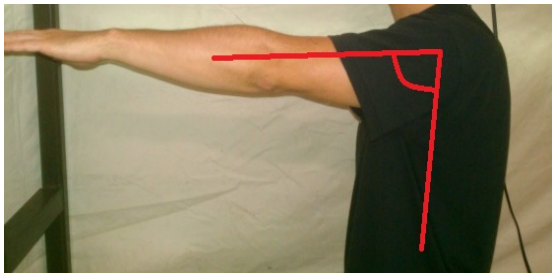
Adduction



Extension

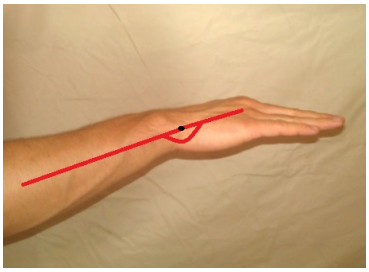


Flexion



Wrist

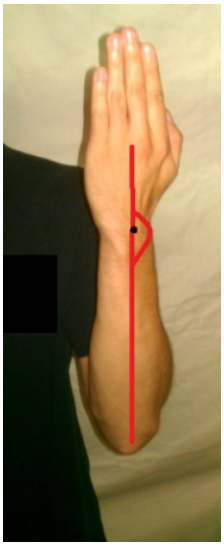
Neutral



Extension

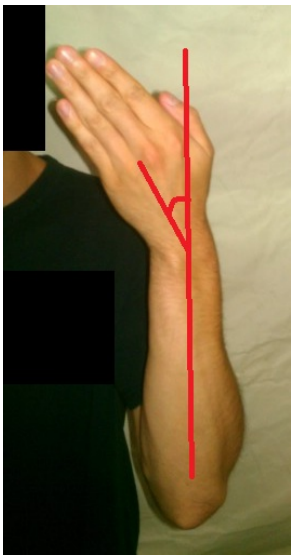
Flexion

Neutral



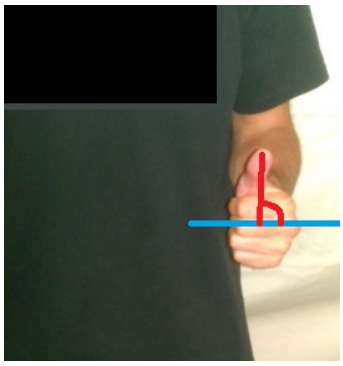
Radial Deviation

Ulnar Deviation

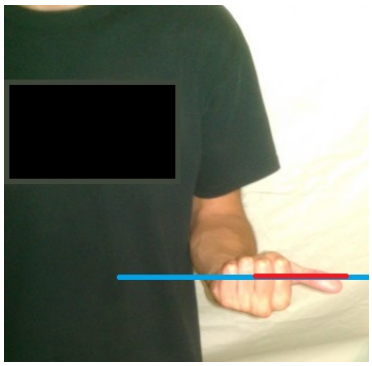


Forearm

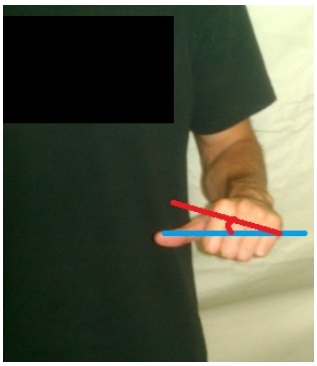
Neutral



Supination



Pronation



APPENDIX B: MATLAB code for neutral posture angle of chest slope for the torso orthogonal axes

Appendix B: MATLAB code for finding the slope of the chest from the neutral capture

Define subj as subject number

m-file code:

```
function [theta] = NeutralAngle(subj)

NeutralMarkerData=xlsread(['C:\Users\Tarek Tantawy\Desktop\Final Marker
Data\Neutral\', subj, 'N.xlsx']);
timec=[1];       %defining column ids
shoulder=[3:5];
tricep=[7:9];
le=[11:13];
rclav=[31:33];
stern=[35:37];
lclav=[39:41];

%------- Vector Definitions -------

%--- Left Clavicle to Right Clavicle ---
lclav2rclav=NeutralMarkerData(:,lclav)-NeutralMarkerData(:,rclav);
Rlclav2rclav(:,1)=sqrt(abs(lclav2rclav(:,1)).^2 + abs(lclav2rclav(:,2)).^2 +
abs(lclav2rclav(:,3)).^2);
elclav2rclav(:,1)=lclav2rclav(:,1)./Rlclav2rclav(:,1);
elclav2rclav(:,2)=lclav2rclav(:,2)./Rlclav2rclav(:,1);
elclav2rclav(:,3)=lclav2rclav(:,3)./Rlclav2rclav(:,1);

%--- Sternum to Right Clavicle---
stern2rclav=NeutralMarkerData(:,stern)-NeutralMarkerData(:,rclav);
Rstern2rclav(:,1)=sqrt(abs(stern2rclav(:,1)).^2 + abs(stern2rclav(:,2)).^2 +
abs(stern2rclav(:,3)).^2);
estern2rclav(:,1)=stern2rclav(:,1)./Rstern2rclav(:,1);
estern2rclav(:,2)=stern2rclav(:,2)./Rstern2rclav(:,1);
estern2rclav(:,3)=stern2rclav(:,3)./Rstern2rclav(:,1);

%--- Torso Axes Calculations---
Torso(:,1:3)=elclav2rclav(:,1:3); %x
TY(:,1:3)=cross(Torso(:,1:3),estern2rclav(:,1:3)); %y
RTY(:,1)=sqrt(abs(TY(:,1)).^2 + abs(TY(:,2)).^2 + abs(TY(:,3)).^2);
Torso(:,4)=TY(:,1)./RTY(:,1);
Torso(:,5)=TY(:,2)./RTY(:,1);
Torso(:,6)=TY(:,3)./RTY(:,1);
Torso(:,7:9)=cross(Torso(:,1:3), Torso(:,4:6)); %z

%--- Project Torso onto  global ZY plane ---
```

```
[R,C]=size(Torso);
y(1:R,1)=0;
y(1:R,2)=1;
y(1:R,3)=0;

x(1:R,1)=1;
x(1:R,2)=0;
x(1:R,3)=0;

YZprojection(:,1:3)= cross(cross(x(:,1:3),Torso(:,7:9)),x(:,1:3));
RYZprojection(:,1)=sqrt(abs(YZprojection(:,1)).^2 + abs(YZprojection(:,2)).^2 +
abs(YZprojection(:,3)).^2);
YZprojection(:,1)=YZprojection(:,1)./RYZprojection(:,1);
YZprojection(:,2)=YZprojection(:,2)./RYZprojection(:,1);
YZprojection(:,3)=YZprojection(:,3)./RYZprojection(:,1);

%--- Find Angle Between Projection and Global y-axis ---
for r=1:R
   th(r,1) = rad2deg(acos(dot(y(r,1:3),YZprojection(r,1:3))));
end

theta= mean(th);

end
```

APPENDIXC: MATLAB Code for calculating shoulder posture

Appendix C: MATLAB code for calculating shoulder posture where function 'ShoulderProjection1' is shown in Appendix D.

m-file code:

```
clear
clc
count=0;

for s=2:11
subject = num2str(s);
files=dir(['C:\Users\Tarek Tantawy\Desktop\Final Marker Data\Subject ' subject ]);

%--- Neutral Calculations ---
theta=NeutralAngle([subject]);
theta=deg2rad(theta);
Rotation=[1 0 0; 0 cos(theta) sin(theta); 0 -sin(theta) cos(theta)];

for t=3:length(files)       %for loop to go through all files

%-- empty matricies
time=[];
XZprojection=[];
RXZprojection=[];
YZprojection=[];
RYZprojection=[];
XYprojection=[];
RXYprojection=[];

x=[];
y=[];
z=[];

datatowrite=[];

if strcmp(files(t).name(end-4:end),'.xlsx')

count=count+1
tic
clf

[upperarm,torso] = ShoulderProjection1(['Subject ' subject '\' files(t).name]);
[R,C]=size(upperarm);
time(:,1)=100 .* (upperarm(:,1)-upperarm(1,1))./ (upperarm(R,1)-upperarm(1,1));


%--- Shoulder Calculations ---
```

```
%--- Rotate about theta ---

for r=1:R
torso(r,2:4)=transpose( Rotation * transpose(torso(r,2:4)));
torso(r,5:7)=transpose( Rotation * transpose(torso(r,5:7)));
torso(r,8:10)=transpose( Rotation * transpose(torso(r,8:10)));
end

%-z- xz- Plane Projection -
XZprojection(:,1:3)= cross(torso(:,5:7),cross(upperarm(:,8:10),torso(:,5:7)));
RXZprojection(:,1)=sqrt(abs(XZprojection(:,1)).^2 + abs(XZprojection(:,2)).^2 +
abs(XZprojection(:,3)).^2);
XZprojection(:,1)=XZprojection(:,1)./RXZprojection(:,1);
XZprojection(:,2)=XZprojection(:,2)./RXZprojection(:,1);
XZprojection(:,3)=XZprojection(:,3)./RXZprojection(:,1);

%-z- yz- Plane Projection -
YZprojection(:,1:3)= cross(torso(:,2:4),cross(upperarm(:,8:10),torso(:,2:4)));
RYZprojection(:,1)=sqrt(abs(YZprojection(:,1)).^2 + abs(YZprojection(:,2)).^2 +
abs(YZprojection(:,3)).^2);
YZprojection(:,1)=YZprojection(:,1)./RYZprojection(:,1);
YZprojection(:,2)=YZprojection(:,2)./RYZprojection(:,1);
YZprojection(:,3)=YZprojection(:,3)./RYZprojection(:,1);

%-x- xy- Plane Projection -
XYprojection(:,1:3)= cross(torso(:,8:10),cross(upperarm(:,2:4),torso(:,8:10)));
RXYprojection(:,1)=sqrt(abs(XYprojection(:,1)).^2 + abs(XYprojection(:,2)).^2 +
abs(XYprojection(:,3)).^2);
XYprojection(:,1)=XYprojection(:,1)./RXYprojection(:,1);
XYprojection(:,2)=XYprojection(:,2)./RXYprojection(:,1);
XYprojection(:,3)=XYprojection(:,3)./RXYprojection(:,1);

for r=1:R

%New Base
Btorso(1:3,1)=torso(r,2:4);
Btorso(1:3,2)=torso(r,5:7);
Btorso(1:3,3)=torso(r,8:10);

%Old Base
Bnew=[1 0 0; 0 1 0; 0 0 1];

P = Bnew*inv(Btorso);

YZnew=P*transpose(YZprojection(r,1:3));
```

```
XZnew=P*transpose(XZprojection(r,1:3));
XYnew=P*transpose(XYprojection(r,1:3));

%--- x- ---

if YZnew(2) <= 0
x(r,1)=rad2deg(acos(dot(YZprojection(r,1:3),torso(r,8:10))));
else
x(r,1)=-rad2deg(acos(dot(YZprojection(r,1:3),torso(r,8:10))));
end

%--- y- ---

if XZnew(1) >= 0
y(r,1)=rad2deg(acos(dot(XZprojection(r,1:3),torso(r,8:10))));
else
y(r,1)=-rad2deg(acos(dot(XZprojection(r,1:3),torso(r,8:10))));
end

%--- z- ---

if XYnew(2) >= 0
z(r,1)=rad2deg(acos(dot(XYprojection(r,1:3),torso(r,2:4))));
else
z(r,1)=-rad2deg(acos(dot(XYprojection(r,1:3),torso(r,2:4))));
end
end

datatowrite=[time smooth(x) smooth(y) smooth(z)];

subplot(3,1,1);
plot(datatowrite(:,1),datatowrite(:,2), 'r');
title({files(t).name(1:end-5);'Shoulder Flexion/Extension'},'fontsize', 18, 'fontweight', 'b');
grid on;
axis([datatowrite(1,1) datatowrite(R,1) -inf inf]);
xlabel('Time (seconds)', 'fontsize', 14, 'fontweight', 'b');
ylabel({'Shoulder Extension (positive)'; 'and Flexion (negative)'}, 'fontsize', 14,
'fontweight', 'b');

subplot(3,1,2);
plot(datatowrite(:,1),datatowrite(:,3), 'b');
title('Shoulder Abduction/Adduction','fontsize', 18, 'fontweight', 'b');
grid on;
axis([datatowrite(1,1) datatowrite(R,1) -inf inf]);
xlabel('Time (seconds)', 'fontsize', 14, 'fontweight', 'b');
```

```
ylabel({'Shoulder Abduction (positive)'; 'and Adduction (negative)'}, 'fontsize', 14,
'fontweight', 'b');

subplot(3,1,3);
plot(datatowrite(:,1),datatowrite(:,4), 'g');
title('Shoulder Medial/Lateral Rotation','fontsize', 18, 'fontweight', 'b');
grid on;
axis([datatowrite(1,1) datatowrite(R,1) -inf inf]);
xlabel('Time (seconds)', 'fontsize', 14, 'fontweight', 'b');
ylabel({'Shoulder Medial Rotation (Positive)';'and Lateral Rotation (negative)'},
'fontsize', 14, 'fontweight', 'b');

saveas(gcf, ['C:\Users\Tarek Tantawy\Desktop\Shoulder 4-14\Subject ' subject '\'
files(t).name(1:end-5) '.fig']);
xlswrite(['C:\Users\Tarek Tantawy\Desktop\Shoulder 4-14\Subject ' subject '\'
files(t).name], datatowrite);

end
toc
end
end
```

APPENDIX D: MATLAB Code for calculating vectors for shoulder posture calculation

Appendix D: MATLAB code for calculating vectors for shoulder posture calculations.

m-file code:

```
function [Upperarm, Torso] = Postures1(filepath)
%clear
%clc

%files=dir('C:\Users\Tarek Tantawy\Desktop\Final Marker Data');

timec=[1];      %defining column ids
shoulder=[3:5];
tricep=[7:9];
le=[11:13];
rclav=[31:33];
stern=[35:37];
lclav=[39:41];

MarkerData=xlsread(['C:\Users\Tarek Tantawy\Desktop\Final Marker Data\', filepath]);

time=MarkerData(:,timec);
L=length(time);
%------- Vector Definitions -------

%--- Left Clavicle to Right Clavicle ---
lclav2rclav=MarkerData(:,lclav)-MarkerData(:,rclav);
Rlclav2rclav(:,1)=sqrt(abs(lclav2rclav(:,1)).^2 + abs(lclav2rclav(:,2)).^2 +
abs(lclav2rclav(:,3)).^2);
elclav2rclav(:,1)=lclav2rclav(:,1)./Rlclav2rclav(:,1);
elclav2rclav(:,2)=lclav2rclav(:,2)./Rlclav2rclav(:,1);
elclav2rclav(:,3)=lclav2rclav(:,3)./Rlclav2rclav(:,1);

%--- Sternum to Right Clavicle---
stern2rclav=MarkerData(:,stern)-MarkerData(:,rclav);
Rstern2rclav(:,1)=sqrt(abs(stern2rclav(:,1)).^2 + abs(stern2rclav(:,2)).^2 +
abs(stern2rclav(:,3)).^2);
estern2rclav(:,1)=stern2rclav(:,1)./Rstern2rclav(:,1);
estern2rclav(:,2)=stern2rclav(:,2)./Rstern2rclav(:,1);
estern2rclav(:,3)=stern2rclav(:,3)./Rstern2rclav(:,1);

%--- Torso Axes Calculations---
Torso(:,1:3)=elclav2rclav(:,1:3); %x
TY(:,1:3)=cross(Torso(:,1:3),estern2rclav(:,1:3)); %y
RTY(:,1)=sqrt(abs(TY(:,1)).^2 + abs(TY(:,2)).^2 + abs(TY(:,3)).^2);
Torso(:,4)=TY(:,1)./RTY(:,1);
Torso(:,5)=TY(:,2)./RTY(:,1);
```

```
Torso(:,6)=TY(:,3)./RTY(:,1);
Torso(:,7:9)=cross(Torso(:,1:3), Torso(:,4:6)); %z
Torso=[time Torso];

%--- Shoulder to Lateral Epicondyle---
sho2le=MarkerData(:,shoulder)-MarkerData(:,le);
Rsho2le(:,1)=sqrt(abs(sho2le(:,1)).^2 + abs(sho2le(:,2)).^2 + abs(sho2le(:,3)).^2);
esho2le(:,1)=sho2le(:,1)./Rsho2le(:,1);
esho2le(:,2)=sho2le(:,2)./Rsho2le(:,1);
esho2le(:,3)=sho2le(:,3)./Rsho2le(:,1);

%--- Tricep to Lateral Epicondyle---
tri2le=MarkerData(:,tricep)-MarkerData(:,le);
Rtri2le(:,1)=sqrt(abs(tri2le(:,1)).^2 + abs(tri2le(:,2)).^2 + abs(tri2le(:,3)).^2);
etri2le(:,1)=tri2le(:,1)./Rtri2le(:,1);
etri2le(:,2)=tri2le(:,2)./Rtri2le(:,1);
etri2le(:,3)=tri2le(:,3)./Rtri2le(:,1);

%--- Upper Arm Axes Calculations---
Upperarm(:,7:9)=esho2le(:,1:3); %z
UX(:,1:3)=cross(etri2le,Upperarm(:,7:9)); %x
RUX(:,1)=sqrt(abs(UX(:,1)).^2 + abs(UX(:,2)).^2 + abs(UX(:,3)).^2);
Upperarm(:,1)=UX(:,1)./RUX(:,1);
Upperarm(:,2)=UX(:,2)./RUX(:,1);
Upperarm(:,3)=UX(:,3)./RUX(:,1);
Upperarm(:,4:6)=cross(Upperarm(:,7:9),Upperarm(:,1:3)); %y
Upperarm=[time Upperarm];

end
```

APPENDIX E: MATLAB Code for calculating wrist and forearm posture

Appendix E: MATLAB code for calculating wrist and elbow postures where function 'Postures1' is shown in Appendix F.

m-file code:

```
clear
clc
count=0;

for s=2:11

subject=num2str(s);
files=dir(['C:\Users\Tarek Tantawy\Desktop\Final Marker Data\Subject ' subject]);

for t=3:length(files)       %for loop to go through all files
clf
%--empty matrices
time=[];
XZprojection=[];
RXZprojection=[];
YZprojection=[];
RYZprojection=[];
XYprojection=[];
RXYprojection=[];
UFXYprojection=[];
RUFXYprojection=[];

elbowr=[];
flexext=[];
deviation=[];
elbow=[];
datatowrite=[];

if strcmp(files(t).name(end-4:end),'.xlsx')

tic
count=count+1

[upperarm, forearm, torso, hand, tool] = Postures1(['Subject ' subject '\' files(t).name]);
[R,C]=size(upperarm);
time(:,1)=100 .* (upperarm(:,1)-upperarm(1,1))./ (upperarm(R,1)-upperarm(1,1));

%--- Wrist Calculations ---

%-z- xz- Plane Projection -
XZprojection(:,1:3)= cross(forearm(:,5:7),cross(hand(:,8:10),forearm(:,5:7)));
```

```
RXZprojection(:,1)=sqrt(abs(XZprojection(:,1)).^2 + abs(XZprojection(:,2)).^2 +
abs(XZprojection(:,3)).^2);
XZprojection(:,1)=XZprojection(:,1)./RXZprojection(:,1);
XZprojection(:,2)=XZprojection(:,2)./RXZprojection(:,1);
XZprojection(:,3)=XZprojection(:,3)./RXZprojection(:,1);

%-z- yz- Plane Projection -
YZprojection(:,1:3)= cross(forearm(:,2:4), cross(hand(:,8:10),forearm(:,2:4)));
RYZprojection(:,1)=sqrt(abs(YZprojection(:,1)).^2 + abs(YZprojection(:,2)).^2 +
abs(YZprojection(:,3)).^2);
YZprojection(:,1)=YZprojection(:,1)./RYZprojection(:,1);
YZprojection(:,2)=YZprojection(:,2)./RYZprojection(:,1);
YZprojection(:,3)=YZprojection(:,3)./RYZprojection(:,1);

%-x- xy- Plane Projection -
XYprojection(:,1:3)= cross(forearm(:,8:10),cross(hand(:,2:4),forearm(:,8:10)));
RXYprojection(:,1)=sqrt(abs(XYprojection(:,1)).^2 + abs(XYprojection(:,2)).^2 +
abs(XYprojection(:,3)).^2);
XYprojection(:,1)=XYprojection(:,1)./RXYprojection(:,1);
XYprojection(:,2)=XYprojection(:,2)./RXYprojection(:,1);
XYprojection(:,3)=XYprojection(:,3)./RXYprojection(:,1);

%--- Forearm Rotation Projection ---

%- Upperarm x- Projected to Forearm xy- -
UFXYprojection(:,1:3)= cross(forearm(:,8:10), cross(upperarm(:,2:4),forearm(:,8:10)));
RUFXYprojection(:,1)=sqrt(abs(UFXYprojection(:,1)).^2 + abs(UFXYprojection(:,2)).^2
+ abs(UFXYprojection(:,3)).^2);
UFXYprojection(:,1)=UFXYprojection(:,1)./RUFXYprojection(:,1);
UFXYprojection(:,2)=UFXYprojection(:,2)./RUFXYprojection(:,1);
UFXYprojection(:,3)=UFXYprojection(:,3)./RUFXYprojection(:,1);


for r=1:R

   %Base in old
   Bforearm(1:3,1)=forearm(r,2:4);
   Bforearm(1:3,2)=forearm(r,5:7);
   Bforearm(1:3,3)=forearm(r,8:10);

   %Base in new
   Bnew=[1 0 0; 0 1 0; 0 0 1];

   P = Bnew*inv(Bforearm);

   %--- Wrist Flexion/Extension ---
```

```
    %Angle sign check
    XZnew=P*transpose(XZprojection(r,1:3));
    YZnew=P*transpose(YZprojection(r,1:3));
    UFXYnew=P*transpose(UFXYprojection(r,1:3));

    if XZnew(1) >= 0
       flexext(r,1)=rad2deg(acos(dot(XZprojection(r,1:3),forearm(r,8:10))));
    else
       flexext(r,1)=-rad2deg(acos(dot(XZprojection(r,1:3),forearm(r,8:10))));
    end

    %--- Wrist Ulnar/Radial Deviation

    if YZnew(2) <= 0
       deviation(r,1)=rad2deg(acos(dot(YZprojection(r,1:3),forearm(r,8:10))));
    else
       deviation(r,1)=-rad2deg(acos(dot(YZprojection(r,1:3),forearm(r,8:10))));
    end

    %--- Elbow Rotation ---

    if UFXYnew(2)<=0
       elbowr(r,1)=rad2deg(acos(dot(UFXYprojection(r,1:3),forearm(r,2:4))));
    else
       elbowr(r,1)=-rad2deg(acos(dot(UFXYprojection(r,1:3),forearm(r,2:4))));
    end

    %--- Elbow Flexion/Extension

    elbow(r,1)=180-rad2deg(acos(dot(forearm(r,8:10), upperarm(r,8:10))));
end


datatowrite = [time smooth(flexext) smooth(deviation) smooth(elbow) smooth(elbowr)];

subplot(2,1,1);
plot(datatowrite(:,1),datatowrite(:,2), 'r');
hold on
title({files(t).name(1:end-5);'Wrist Flexion/Extension'},'fontsize', 18, 'fontweight', 'b');
grid on;
axis([datatowrite(1,1) datatowrite(R,1) -inf inf]);
xlabel('Time (% Task Completion)', 'fontsize', 11, 'fontweight', 'b');
ylabel({'Wrist Flexion (negative)'; 'and Extension (positive)'}, 'fontsize', 11, 'fontweight',
'b');
```

```
subplot(2,1,2);
plot(datatowrite(:,1),datatowrite(:,3), 'b');
hold on
title('Wrist Ulnar/Radial Deviation','fontsize', 18, 'fontweight', 'b');
grid on;
axis([datatowrite(1,1) datatowrite(R,1) -inf inf]);
xlabel('Time (% Task Completion)', 'fontsize', 11, 'fontweight', 'b');
ylabel({'Wrist Radial (negative) and'; 'Ulnar (positive) Deviation'}, 'fontsize', 11,
'fontweight', 'b');

saveas(gcf, ['C:\Users\Tarek Tantawy\Desktop\Postures 4-14\Subject ' subject '\'
files(t).name(1:end-5) ' Wrist.fig']);

clf

subplot(2,1,1);
plot(datatowrite(:,1),datatowrite(:,4), 'b');
hold on
title({files(t).name(1:end-5);'Elbow Flexion/Extension'},'fontsize', 18, 'fontweight', 'b');
grid on;
axis([datatowrite(1,1) datatowrite(R,1) -inf inf]);
xlabel('Time (% Task Completion)', 'fontsize', 11, 'fontweight', 'b');
ylabel('Elbow Flexion/Extension', 'fontsize', 11, 'fontweight', 'b');

subplot(2,1,2);
plot(datatowrite(:,1),datatowrite(:,5), 'b');
hold on
title('Elbow Rotation','fontsize', 18, 'fontweight', 'b');
grid on;
axis([datatowrite(1,1) datatowrite(R,1) -inf inf]);
xlabel('Time (% Task Completion)', 'fontsize', 11, 'fontweight', 'b');
ylabel('Elbow Rotation', 'fontsize', 11, 'fontweight', 'b');

saveas(gcf, ['C:\Users\Tarek Tantawy\Desktop\Postures 4-14\Subject ' subject '\'
files(t).name(1:end-5) ' Elbow.fig']);
xlswrite(['C:\Users\Tarek Tantawy\Desktop\Postures 4-14\Subject ' subject '\'
files(t).name(1:end-5) '.xlsx'], datatowrite);

end
toc
end
end
```

APPENDIX F: MATLAB Code for calculating vectors for wrist posture calculation

Appendix F: MATLAB code for calculating vectors for wrist posture calculations.

m-file code:

```
function [Upperarm, Forearm, Torso, Hand, Tool] = Postures1(filepath)
%clear
%clc

%files=dir('C:\Users\Tarek Tantawy\Desktop\Final Marker Data');

timec=[1];      %defining column ids
shoulder=[3:5];
tricep=[7:9];
le=[11:13];
us=[15:17];
rs=[19:21];
fmcp=[23:25];
smcp=[27:29];
rclav=[31:33];
stern=[35:37];
lclav=[39:41];
toolrear=[55:57];
toolcorner=[59:61];
toolwing=[63:65];
MarkerData=xlsread(['C:\Users\Tarek Tantawy\Desktop\Final Marker Data\' filepath]);

time=MarkerData(:,timec);

%------- Vector Definitions -------

%--- Shoulder to Lateral Epicondyle---
sho2le=MarkerData(:,shoulder)-MarkerData(:,le);
Rsho2le(:,1)=sqrt(abs(sho2le(:,1)).^2 + abs(sho2le(:,2)).^2 + abs(sho2le(:,3)).^2);
esho2le(:,1)=sho2le(:,1)./Rsho2le(:,1);
esho2le(:,2)=sho2le(:,2)./Rsho2le(:,1);
esho2le(:,3)=sho2le(:,3)./Rsho2le(:,1);

%--- Tricep to Lateral Epicondyle---
tri2le=MarkerData(:,tricep)-MarkerData(:,le);
Rtri2le(:,1)=sqrt(abs(tri2le(:,1)).^2 + abs(tri2le(:,2)).^2 + abs(tri2le(:,3)).^2);
etri2le(:,1)=tri2le(:,1)./Rtri2le(:,1);
etri2le(:,2)=tri2le(:,2)./Rtri2le(:,1);
etri2le(:,3)=tri2le(:,3)./Rtri2le(:,1);

%--- Styloid MP---
styloidmp(:,1:3)=.5*(MarkerData(:,us)+MarkerData(:,rs));
```

```
%--- MCP MP---
mcpmp(:,1:3)=.5*(MarkerData(:,fmcp)+MarkerData(:,smcp));

%--- Styloid MP to MCP MP---
smp2mcpmp=styloidmp(:,1:3)-mcpmp(:,1:3);
Rsmp2mcpmp(:,1)=sqrt(abs(smp2mcpmp(:,1)).^2 + abs(smp2mcpmp(:,2)).^2 +
abs(smp2mcpmp(:,3)).^2);
esmp2mcpmp(:,1)=smp2mcpmp(:,1)./Rsmp2mcpmp(:,1);
esmp2mcpmp(:,2)=smp2mcpmp(:,2)./Rsmp2mcpmp(:,1);
esmp2mcpmp(:,3)=smp2mcpmp(:,3)./Rsmp2mcpmp(:,1);

%--- Fifth MCP to MCP MP---
fmcp2mcpmp=MarkerData(:,fmcp)-mcpmp(:,1:3);
Rfmcp2mcpmp(:,1)=sqrt(abs(fmcp2mcpmp(:,1)).^2 + abs(fmcp2mcpmp(:,2)).^2 +
abs(fmcp2mcpmp(:,3)).^2);
efmcp2mcpmp(:,1)=fmcp2mcpmp(:,1)./Rfmcp2mcpmp(:,1);
efmcp2mcpmp(:,2)=fmcp2mcpmp(:,2)./Rfmcp2mcpmp(:,1);
efmcp2mcpmp(:,3)=fmcp2mcpmp(:,3)./Rfmcp2mcpmp(:,1);

%--- Hand Axes Calculations---
Hand(:,7:9)=esmp2mcpmp(:,1:3); %z
FX(:,1:3)=cross(Hand(:,7:9),efmcp2mcpmp); %x
RFX(:,1)=sqrt(abs(FX(:,1)).^2 + abs(FX(:,2)).^2 + abs(FX(:,3)).^2);
Hand(:,1)=FX(:,1)./RFX(:,1);
Hand(:,2)=FX(:,2)./RFX(:,1);
Hand(:,3)=FX(:,3)./RFX(:,1);
Hand(:,4:6)=cross(Hand(:,7:9),Hand(:,1:3)); %y
Hand=[time Hand];

%--- Lateral Epicondyle to Styloid MP---
le2smp=MarkerData(:,le)-styloidmp(:,1:3);
Rle2smp(:,1)=sqrt(abs(le2smp(:,1)).^2 + abs(le2smp(:,2)).^2 + abs(le2smp(:,3)).^2);
ele2smp(:,1)=le2smp(:,1)./Rle2smp(:,1);
ele2smp(:,2)=le2smp(:,2)./Rle2smp(:,1);
ele2smp(:,3)=le2smp(:,3)./Rle2smp(:,1);

%--- Styloid MP to MCP MP---
smp2mcpmp=styloidmp(:,1:3)-mcpmp(:,1:3);
Rsmp2mcpmp(:,1)=sqrt(abs(smp2mcpmp(:,1)).^2 + abs(smp2mcpmp(:,2)).^2 +
abs(smp2mcpmp(:,3)).^2);
esmp2mcpmp(:,1)=smp2mcpmp(:,1)./Rsmp2mcpmp(:,1);
esmp2mcpmp(:,2)=smp2mcpmp(:,2)./Rsmp2mcpmp(:,1);
esmp2mcpmp(:,3)=smp2mcpmp(:,3)./Rsmp2mcpmp(:,1);

%--- Ulnar Styloid to Styloid MP---
us2smp=MarkerData(:,us)-styloidmp(:,1:3);
```

```
Rus2smp(:,1)=sqrt(abs(us2smp(:,1)).^2 + abs(us2smp(:,2)).^2 + abs(us2smp(:,3)).^2);
eus2smp(:,1)=us2smp(:,1)./Rus2smp(:,1);
eus2smp(:,2)=us2smp(:,2)./Rus2smp(:,1);
eus2smp(:,3)=us2smp(:,3)./Rus2smp(:,1);

%--- Forearm Axes Calculations---
Forearm(:,7:9)=ele2smp(:,1:3); %z
FX(:,1:3)=cross(eus2smp,Forearm(:,7:9)); %x
RFX(:,1)=sqrt(abs(FX(:,1)).^2 + abs(FX(:,2)).^2 + abs(FX(:,3)).^2);
Forearm(:,1)=FX(:,1)./RFX(:,1);
Forearm(:,2)=FX(:,2)./RFX(:,1);
Forearm(:,3)=FX(:,3)./RFX(:,1);
Forearm(:,4:6)=cross(Forearm(:,7:9),Forearm(:,1:3)); %y
Forearm=[time Forearm];

%--- Upper Arm Axes Calculations---
Upperarm(:,7:9)=esho2le(:,1:3); %z
UX(:,1:3)=cross(etri2le,Upperarm(:,7:9)); %x
RUX(:,1)=sqrt(abs(UX(:,1)).^2 + abs(UX(:,2)).^2 + abs(UX(:,3)).^2);
Upperarm(:,1)=UX(:,1)./RUX(:,1);
Upperarm(:,2)=UX(:,2)./RUX(:,1);
Upperarm(:,3)=UX(:,3)./RUX(:,1);
Upperarm(:,4:6)=cross(Upperarm(:,7:9),Upperarm(:,1:3)); %y
Upperarm=[time Upperarm];

%--- Left Clavicle to Right Clavicle ---
lclav2rclav=MarkerData(:,lclav)-MarkerData(:,rclav);
Rlclav2rclav(:,1)=sqrt(abs(lclav2rclav(:,1)).^2 + abs(lclav2rclav(:,2)).^2 +
abs(lclav2rclav(:,3)).^2);
elclav2rclav(:,1)=lclav2rclav(:,1)./Rlclav2rclav(:,1);
elclav2rclav(:,2)=lclav2rclav(:,2)./Rlclav2rclav(:,1);
elclav2rclav(:,3)=lclav2rclav(:,3)./Rlclav2rclav(:,1);

%--- Sternum to Right Clavicle---
stern2rclav=MarkerData(:,stern)-MarkerData(:,rclav);
Rstern2rclav(:,1)=sqrt(abs(stern2rclav(:,1)).^2 + abs(stern2rclav(:,2)).^2 +
abs(stern2rclav(:,3)).^2);
estern2rclav(:,1)=stern2rclav(:,1)./Rstern2rclav(:,1);
estern2rclav(:,2)=stern2rclav(:,2)./Rstern2rclav(:,1);
estern2rclav(:,3)=stern2rclav(:,3)./Rstern2rclav(:,1);

%--- Torso Axes Calculations---
Torso(:,1:3)=elclav2rclav(:,1:3); %x
TY(:,1:3)=cross(Torso(:,1:3),estern2rclav(:,1:3)); %y
RTY(:,1)=sqrt(abs(TY(:,1)).^2 + abs(TY(:,2)).^2 + abs(TY(:,3)).^2);
Torso(:,4)=TY(:,1)./RTY(:,1);
```

```
Torso(:,5)=TY(:,2)./RTY(:,1);
Torso(:,6)=TY(:,3)./RTY(:,1);
Torso(:,7:9)=cross(Torso(:,1:3), Torso(:,4:6)); %z
Torso=[time Torso];

%--- Tool Rear to Corner---
re2co=MarkerData(:,toolrear)-MarkerData(:,toolcorner);
Rre2co(:,1)=sqrt(abs(re2co(:,1)).^2 + abs(re2co(:,2)).^2 + abs(re2co(:,3)).^2);
ere2co(:,1)=re2co(:,1)./Rre2co(:,1);
ere2co(:,2)=re2co(:,2)./Rre2co(:,1);
ere2co(:,3)=re2co(:,3)./Rre2co(:,1);

%--- Tool Corner to Wing---
co2wing=MarkerData(:,toolcorner)-MarkerData(:,toolwing);
Rco2wing(:,1)=sqrt(abs(co2wing(:,1)).^2 + abs(co2wing(:,2)).^2 +
abs(co2wing(:,3)).^2);
eco2wing(:,1)=co2wing(:,1)./Rco2wing(:,1);
eco2wing(:,2)=co2wing(:,2)./Rco2wing(:,1);
eco2wing(:,3)=co2wing(:,3)./Rco2wing(:,1);

%--- Tool Axes Calculations---
Tool(:,7:9)=ere2co(:,1:3); %z
ToolY(:,1:3)=cross(eco2wing,Tool(:,7:9)); %y
RToolY(:,1)=sqrt(abs(ToolY(:,1)).^2 + abs(ToolY(:,2)).^2 + abs(ToolY(:,3)).^2);
Tool(:,4)=ToolY(:,1)./RToolY(:,1);
Tool(:,5)=ToolY(:,2)./RToolY(:,1);
Tool(:,6)=ToolY(:,3)./RToolY(:,1);
Tool(:,1:3)=cross(Tool(:,4:6),Tool(:,7:9)); %x
Tool=[time Tool];

end
```

APPENDIX G: MATLAB Code for threshold calculations of shoulder postures

Appendix G: MATLAB code for making threshold calculations from shoulder posture calculations.

m-file code:

```
clear
clc
count=1;
names{1}='Files';
for s=2:11
subject = num2str(s);
files=dir(['C:\Users\Tarek Tantawy\Desktop\Shoulder 2\Subject ' subject]);


for t=3:length(files)       %for loop to go through all files

timex=[];
timey=[];
timez=[];
ShoulderData=[];
x-=[];
y-=[];
z-=[];
Tx=[];
Ty=[];
Tz=[];
indx=[];
indy=[];
indz=[];

if strcmp(files(t).name(end-4:end),'.xlsx')
count=count+1
tic
%Open data
ShoulderData=xlsread(['C:\Users\Tarek Tantawy\Desktop\Shoulder 2\Subject ' subject '\'
files(t).name]);
[R,C]=size(ShoulderData);

%Add/Subtract Thresholds from Rotations

x-(:,1)=abs(ShoulderData(:,2));
y-(:,1)=abs(ShoulderData(:,3));
z-(:,1)=abs(ShoulderData(:,4));

indx = find(x->60);
Tx=x-(indx);
Tx=Tx-60;
```

```
timex=ShoulderData(indx);
Ax=trapz(timex,Tx);
totalareax=trapz(ShoulderData(:,1), x-);
PercentAx=100*Ax/totalareax;
PercentTimex=100*(length(timex)/R);


indy = find(y->60);
Ty=y-(indy);
Ty=Ty-60;
timey=ShoulderData(indy);
Ay=trapz(timey,Ty);
totalareay=trapz(ShoulderData(:,1), y-);
PercentAy=100*Ay/totalareay;
PercentTimey=100*(length(timey)/R);
indz = find(z->60);
Tz=z-(indz);
Tz=Tz-60;
timez=ShoulderData(indz);
Az=trapz(timez,Tz);
totalareaz=trapz(ShoulderData(:,1), z-);
PercentAz=100*Az/totalareaz;
PercentTimez=100*(length(timez)/R);

datatowrite(count, 2:10)=[Ax Ay Az PercentTimex PercentTimey PercentTimez
PercentAx PercentAy PercentAz];
filename=files(t).name(1:end-5);
names{count,1}= filename;
end
toc
end
end

data=num2cell(datatowrite);
data(:,1)=names;
data(1,2)= {'Area Beyond Threshold x-'};
data(1,3)= {'Area Beyond Threshold y-'};
data(1,4)= {'Area Beyond Threshold z-'};
data(1,5)= {'Percent Time x-'};
data(1,6)= {'Percent Time y-'};
data(1,7)= {'Percent Time z-'};
data(1,8)= {'Percent Area x-'};
data(1,9)= {'Percent Area y-'};
data(1,10)= {'Percent Area z-'};

xlswrite('C:\Users\Tarek Tantawy\Desktop\Shoulder Threshold Calculations .xlsx', data);
```

APPENDIX H: MATLAB Code for threshold calculations of wrist postures

Appendix H: MATLAB code for making threshold calculations from wrist posture calculations.

m-file code:

```
clear
clc
count=1;
names{1}='Files';
for s=2:11
subject = num2str(s);
files=dir(['C:\Users\Tarek Tantawy\Desktop\Postures 1\Subject ' subject]);


for t=3:length(files)       %for loop to go through all files
tic
t
timewfe=[];
timewur=[];
timeefe=[];
timeer=[];
PostureData=[];
WFE=[];
WUR=[];
EFE=[];
ER=[];
Twfe=[];
Twur=[];
Tefe=[];
Ter=[];
indwfe=[];
indwur=[];
indefe=[];
inder=[];

if strcmp(files(t).name(end-4:end),'.xlsx')
count=count+1;

%Open data
PostureData=xlsread(['C:\Users\Tarek Tantawy\Desktop\Postures 1\Subject ' subject '\'
files(t).name]);
[R,C]=size(PostureData);

%Add/Subtract Thresholds from Rotations

WFE(:,1)=abs(PostureData(:,2));
WUR(:,1)=abs(PostureData(:,3));
```

```
EFE(:,1)=abs(PostureData(:,4));
ER(:,1)=abs(PostureData(:,5));

indwfe = find(WFE>15);
Twfe=WFE(indwfe);
Twfe=Twfe-15;
timewfe=PostureData(indwfe);
Awfe=trapz(timewfe,Twfe);
totalareawfe=trapz(PostureData(:,1), WFE);
PercentAwfe=100*Awfe/totalareawfe;
PercentTimewfe=100*(length(timewfe)/R);


indwur = find(WUR>15);
Twur=WUR(indwur);
Twur=Twur-15;
timewur=PostureData(indwur);
Awur=trapz(timewur,Twur);
totalareawur=trapz(PostureData(:,1), WUR);
PercentAwur=100*Awur/totalareawur;
PercentTimewur=100*(length(timewur)/R);

indefe = find(60<EFE<120);
Tefe=EFE(indefe);
timeefe=PostureData(indefe);
Aefe=trapz(timeefe,Tefe);
totalareaefe=trapz(PostureData(:,1), EFE);
Aefe=totalareaefe-Aefe;
PercentAefe=100*(totalareaefe-Aefe)/totalareaefe;
PercentTimeefe=100*((R-length(timeefe))/R);

inder = find(ER>45);
Ter=ER(inder);
Ter=Ter-45;
timeer=PostureData(inder);
Aer=trapz(timeer,Ter);
totalareaer=trapz(PostureData(:,1), ER);
PercentAer=100*Aer/totalareaer;
PercentTimeer=100*(length(timeer)/R);

datatowrite(count, 2:13)=[Awfe Awur Aefe Aer PercentTimewfe PercentTimewur
PercentTimeefe PercentTimeer PercentAwfe PercentAwur PercentAefe PercentAer];
filename=files(t).name(1:end-5);
names{count,1}= filename;

end
```

```
toc
end
end

data=num2cell(datatowrite);

data(:,1)=names;
data(1,2)= {'Area Beyond Threshold WFE'};
data(1,3)= {'Area Beyond Threshold WUR'};
data(1,4)= {'Area Beyond Threshold EFE'};
data(1,5)= {'Area Beyond Threshold ER'};
data(1,6)= {'Percent Time WFE'};
data(1,7)= {'Percent Time WUR'};
data(1,8)= {'Percent Time EFE'};
data(1,9)= {'Percent Time ER'};
data(1,10)= {'Percent Area WFE'};
data(1,11)= {'Percent Area WUR'};
data(1,12)= {'Percent Area EFE'};
data(1,13)= {'Percent Area ER'};
xlswrite('C:\Users\Tarek Tantawy\Desktop\Wrist and Elbow Threshold Calculations 3-
26.xlsx', data);
```

APPENDIX I: MATLAB Code for calculating tool orientation

Appendix I: MATLAB code for calculating tool orientation where function 'ToolOrientation1' is shown in Appendix J.

m-file code:

```
clear
clc
count=0;

for s=2:11

subject=num2str(s);
files=dir(['C:\Users\Tarek Tantawy\Desktop\Final Marker Data\Subject ' subject]);

for t=3:length(files)       %for loop to go through all files
%empty matrices
time=[];
XZprojection=[];
RXZprojection=[];
YZprojection=[];
RYZprojection=[];
XYprojection=[];
RXYprojection=[];

yaw=[];
pitch=[];
roll=[];

datatowrite=[];

if strcmp(files(t).name(end-4:end),'.xlsx')

count=count+1
tic

[hand, tool] = ToolOrientation1(['Subject ' subject '\' files(t).name]);
[R,C]=size(hand);
time(:,1)=100 .* (hand(:,1)-hand(1,1))./ (hand(R,1)-hand(1,1));

%--- Tool Calculations ---

%-Z XZ Plane Projection -
XZprojection(:,1:3)= cross(hand(:,5:7),cross(tool(:,8:10),hand(:,5:7)));
RXZprojection(:,1)=sqrt(abs(XZprojection(:,1)).^2 + abs(XZprojection(:,2)).^2 +
abs(XZprojection(:,3)).^2);
XZprojection(:,1)=XZprojection(:,1)./RXZprojection(:,1);
XZprojection(:,2)=XZprojection(:,2)./RXZprojection(:,1);
```

```
XZprojection(:,3)=XZprojection(:,3)./RXZprojection(:,1);

%-Z YZ Plane Projection -
YZprojection(:,1:3)= cross(hand(:,2:4),cross(tool(:,8:10),hand(:,2:4)));
RYZprojection(:,1)=sqrt(abs(YZprojection(:,1)).^2 + abs(YZprojection(:,2)).^2 +
abs(YZprojection(:,3)).^2);
YZprojection(:,1)=YZprojection(:,1)./RYZprojection(:,1);
YZprojection(:,2)=YZprojection(:,2)./RYZprojection(:,1);
YZprojection(:,3)=YZprojection(:,3)./RYZprojection(:,1);

%-X XY Plane Projection -
XYprojection(:,1:3)= cross(hand(:,8:10),cross(tool(:,2:4),hand(:,8:10)));
RXYprojection(:,1)=sqrt(abs(XYprojection(:,1)).^2 + abs(XYprojection(:,2)).^2 +
abs(XYprojection(:,3)).^2);
XYprojection(:,1)=XYprojection(:,1)./RXYprojection(:,1);
XYprojection(:,2)=XYprojection(:,2)./RXYprojection(:,1);
XYprojection(:,3)=XYprojection(:,3)./RXYprojection(:,1);

for r=1:R

%New Base

Bhand(1:3,1)=hand(r,2:4);
Bhand(1:3,2)=hand(r,5:7);
Bhand(1:3,3)=hand(r,8:10);

%Old Base
Bnew=[1 0 0; 0 1 0; 0 0 1];

P = Bnew*inv(Bhand);

YZnew=P*transpose(YZprojection(r,1:3));
XZnew=P*transpose(XZprojection(r,1:3));
XYnew=P*transpose(XYprojection(r,1:3));

%--- YAW ---
if XZnew(1)>=0
yaw(r,1)=rad2deg(acos(dot(hand(r,8:10),XZprojection(r,1:3))));
else
yaw(r,1)=-rad2deg(acos(dot(hand(r,8:10),XZprojection(r,1:3))));
end

if yaw(r,1)>90
yaw(r,1)=yaw(r,1)-180;
elseif yaw(r,1)<-90
yaw(r,1)=yaw(r,1)+180;
```

```
end

%--- PITCH ---
if YZnew(2)>=0
pitch(r,1)=-rad2deg(acos(dot(hand(r,8:10),YZprojection(r,1:3))));
else
pitch(r,1)=rad2deg(acos(dot(hand(r,8:10),YZprojection(r,1:3))));
end

if pitch(r,1)>90
pitch(r,1)=pitch(r,1)-180;
elseif pitch(r,1)<-90
pitch(r,1)=pitch(r,1)+180;
end

%--- ROLL ---
if XYnew(2)>=0
roll(r,1)=rad2deg(acos(dot(hand(r,2:4), XYprojection(r,1:3))));
else
roll(r,1)=-rad2deg(acos(dot(hand(r,2:4), XYprojection(r,1:3))));
end

if roll(r,1)>90
roll(r,1)=roll(r,1)-180;
elseif roll(r,1)<-90
roll(r,1)=roll(r,1)+180;
end
end

datatowrite = [time smooth(yaw) smooth(pitch) smooth(roll)];

subplot(3,1,1);
plot(datatowrite(:,1),datatowrite(:,2), 'r');
title({files(t).name(1:end-5);'YAW'},'fontsize', 18, 'fontweight', 'b');
grid on;
axis([datatowrite(1,1) datatowrite(R,1) -inf inf]);
xlabel('Time (% Task Completion)', 'fontsize', 11, 'fontweight', 'b');
ylabel('Yaw (Degrees)', 'fontsize', 11, 'fontweight', 'b');

subplot(3,1,2);
plot(datatowrite(:,1),datatowrite(:,3), 'b');
title('PITCH','fontsize', 18, 'fontweight', 'b');
grid on;
axis([datatowrite(1,1) datatowrite(R,1) -inf inf]);
xlabel('Time (% Task Completion)', 'fontsize', 11, 'fontweight', 'b');
ylabel('Pitch (Degrees)', 'fontsize', 11, 'fontweight', 'b');
```

```
subplot(3,1,3);
plot(datatowrite(:,1),datatowrite(:,4), 'b');
title('ROLL','fontsize', 18, 'fontweight', 'b');
grid on;
axis([datatowrite(1,1) datatowrite(R,1) -inf inf]);
xlabel('Time (% Task Completion)', 'fontsize', 11, 'fontweight', 'b');
ylabel('Roll (Degrees)', 'fontsize', 11, 'fontweight', 'b');

saveas(gcf, ['C:\Users\Tarek Tantawy\Desktop\Tool Orientation 4-14\Subject ' subject '\'
files(t).name(1:end-5) '.fig']);
xlswrite(['C:\Users\Tarek Tantawy\Desktop\Tool Orientation 4-14\Subject ' subject '\'
files(t).name(1:end-5) '.xlsx'], datatowrite);

end
toc
end
end
```

APPENDIX J: MATLAB Code for calculating vectors for tool orientation calculation

Appendix J: MATLAB code for calculating vectors for tool orientation calculations.

m-file code:

```
function [Hand, Tool] = Postures1(filepath)
%clear
%clc

%files=dir('C:\Users\Tarek Tantawy\Desktop\Final Marker Data');

timec=[1];      %defining column ids
us=[15:17];
rs=[19:21];
fmcp=[23:25];
smcp=[27:29];
toolrear=[55:57];
toolcorner=[59:61];
toolwing=[63:65];

MarkerData=xlsread(['C:\Users\Tarek Tantawy\Desktop\Final Marker Data\', filepath]);

time=MarkerData(:,timec);

%------- Vector Definitions -------

%--- Styloid MP---
styloidmp(:,1:3)=.5*(MarkerData(:,us)+MarkerData(:,rs));

%--- MCP MP---
mcpmp(:,1:3)=.5*(MarkerData(:,fmcp)+MarkerData(:,smcp));

%--- Styloid MP to MCP MP---
smp2mcpmp=styloidmp(:,1:3)-mcpmp(:,1:3);
Rsmp2mcpmp(:,1)=sqrt(abs(smp2mcpmp(:,1)).^2 + abs(smp2mcpmp(:,2)).^2 +
abs(smp2mcpmp(:,3)).^2);
esmp2mcpmp(:,1)=smp2mcpmp(:,1)./Rsmp2mcpmp(:,1);
esmp2mcpmp(:,2)=smp2mcpmp(:,2)./Rsmp2mcpmp(:,1);
esmp2mcpmp(:,3)=smp2mcpmp(:,3)./Rsmp2mcpmp(:,1);

%--- Fifth MCP to MCP MP---
fmcp2mcpmp=MarkerData(:,fmcp)-mcpmp(:,1:3);
Rfmcp2mcpmp(:,1)=sqrt(abs(fmcp2mcpmp(:,1)).^2 + abs(fmcp2mcpmp(:,2)).^2 +
abs(fmcp2mcpmp(:,3)).^2);
efmcp2mcpmp(:,1)=fmcp2mcpmp(:,1)./Rfmcp2mcpmp(:,1);
efmcp2mcpmp(:,2)=fmcp2mcpmp(:,2)./Rfmcp2mcpmp(:,1);
efmcp2mcpmp(:,3)=fmcp2mcpmp(:,3)./Rfmcp2mcpmp(:,1);
```

```
%--- Hand Axes Calculations---
Hand(:,7:9)=esmp2mcpmp(:,1:3); %z
FX(:,1:3)=cross(Hand(:,7:9),efmcp2mcpmp); %x
RFX(:,1)=sqrt(abs(FX(:,1)).^2 + abs(FX(:,2)).^2 + abs(FX(:,3)).^2);
Hand(:,1)=FX(:,1)./RFX(:,1);
Hand(:,2)=FX(:,2)./RFX(:,1);
Hand(:,3)=FX(:,3)./RFX(:,1);
Hand(:,4:6)=cross(Hand(:,7:9),Hand(:,1:3)); %y
Hand=[time Hand];

%--- Tool Rear to Corner---
re2co=MarkerData(:,toolrear)-MarkerData(:,toolcorner);
Rre2co(:,1)=sqrt(abs(re2co(:,1)).^2 + abs(re2co(:,2)).^2 + abs(re2co(:,3)).^2);
ere2co(:,1)=re2co(:,1)./Rre2co(:,1);
ere2co(:,2)=re2co(:,2)./Rre2co(:,1);
ere2co(:,3)=re2co(:,3)./Rre2co(:,1);

%--- Tool Corner to Wing---
co2wing=MarkerData(:,toolcorner)-MarkerData(:,toolwing);
Rco2wing(:,1)=sqrt(abs(co2wing(:,1)).^2 + abs(co2wing(:,2)).^2 +
abs(co2wing(:,3)).^2);
eco2wing(:,1)=co2wing(:,1)./Rco2wing(:,1);
eco2wing(:,2)=co2wing(:,2)./Rco2wing(:,1);
eco2wing(:,3)=co2wing(:,3)./Rco2wing(:,1);

%--- Tool Axes Calculations---
Tool(:,7:9)=ere2co(:,1:3); %z
ToolY(:,1:3)=cross(eco2wing,Tool(:,7:9)); %y
RToolY(:,1)=sqrt(abs(ToolY(:,1)).^2 + abs(ToolY(:,2)).^2 + abs(ToolY(:,3)).^2);
Tool(:,4)=ToolY(:,1)./RToolY(:,1);
Tool(:,5)=ToolY(:,2)./RToolY(:,1);
Tool(:,6)=ToolY(:,3)./RToolY(:,1);
Tool(:,1:3)=cross(Tool(:,4:6),Tool(:,7:9)); %x
Tool=[time Tool];

end
```

APPENDIX K: RULA scoring sheet

Appendix K: Full RULA scoring sheet (Hedge, 2000).