

2009

Computational Modeling of the Helical Axis in the Human Knee--Detecting Changes in the Transverse Plane

Emily Carney
Virginia Commonwealth University

Follow this and additional works at: <http://scholarscompass.vcu.edu/etd>

 Part of the [Biomedical Engineering and Bioengineering Commons](#)

© The Author

Downloaded from

<http://scholarscompass.vcu.edu/etd/3>

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

Department of Biomedical Engineering
Virginia Commonwealth University

This is to certify that the thesis prepared by Emily Joyce Carney entitled
COMPUTATIONAL MODELING OF THE HELICAL AXIS IN THE HUMAN
KNEE—DETECTING CHANGES IN THE TRANSVERSE PLANE has been approved
by her committee as satisfactory completion of the thesis or dissertation requirement for
the degree of Master of Science in Biomedical Engineering.

Gerald E. Miller, Ph.D., Director and Chair Biomedical Engineering, School of Engineering

Peter E. Pidcoe, PT, DPT, Ph.D., Associate Professor, School of Allied Health Professions

Jennifer S. Wayne, Ph.D., Professor, School of Engineering

Rosalyn S. Hobson, Associate Dean Graduate Studies, School of Engineering

Russell D. Jamison, Ph.D., Dean, School of Engineering

Dr. F. Douglas Boudinot, Dean of the School of Graduate Studies

July 7, 2009

© Emily Joyce Carney 2009

All Rights Reserved

COMPUTATIONAL MODELING OF THE HELICAL AXIS IN THE HUMAN
KNEE—DETECTING CHANGES IN THE TRANSVERSE PLANE

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

by

EMILY JOYCE CARNEY
B.S. in Biomedical Engineering, Louisiana Tech University, May 2006

Director: DR. GERALD E. MILLER
CHAIR, DEPARTMENT OF BIOMEDICAL ENGINEERING

Director: DR. PETER E. PIDCOE
ASSOCIATE PROFESSOR, DEPARTMENT OF PHYSICAL THERAPY

Virginia Commonwealth University
Richmond, Virginia
August 2009

Acknowledgement

I would to thank my Advisor, Dr. Peter Pidcoe, for all his help and guidance throughout my research. Dr. Pidcoe has been a great source of inspiration, suggestions, and assistance through my study at Virginia Commonwealth University.

I also would like to extend my appreciation to my committee members, Dr. Gerald Miller and Dr. Jennifer Wayne, for guidance and support. I would also like to extend a note of appreciation to Bruce Larsen, a programmer with MotionMonitor™, for his assistance in programming. I would also like to thank Cynthia Wright for her help in collecting the Vicon data.

Finally, thanks to the faculty and the staff of the Biomedical Engineering Department of Virginia Commonwealth University for giving me the opportunity to study and pursue my Master of Science Degree at Virginia Commonwealth University. I would like to extend appreciation to Norma Briggs and Joyce Wilkins for their willingness to assist me in my studies.

Most important I would like to thank my family and friends for their support, encouragement, and understanding especially during the difficult times.

Table of Contents

	Page
Acknowledgements	ii
List of Tables.....	v
List of Figures	vi
List of Abbreviations.....	viii
Chapter	
1 Introduction	1
1.1 Anatomy of the Knee.....	3
1.2 Meniscal Tears.....	5
1.3 Knee Movement and Gait.....	7
1.4 Helical Axis	8
1.5 Specific Aims.....	12
2 Methods.....	13
2.1 Multiplanar Rigid Body Model	17
Data collected using MotionMonitor™ kinematic system.....	20
Data collected using Vicon kinematic system	21
2.2 Mathematical (Triangle) Model.....	22
2.3 Cylinder Model.....	24
3 Results	27

3.1	Multiplanar Rigid Body Model	27
	Data collected using MotionMonitor™ kinematic system.....	27
	Data collected using Vicon kinematic system	37
3.2	Mathematical (Triangle) Model.....	39
3.3	Cylinder Model.....	46
4	Discussion	50
4.1	Multiplanar Rigid Body Model	50
4.2	Mathematical (Triangle) Model.....	54
4.3	Cylinder Model.....	57
4.4	Future Work.....	58
	References	60
	Appendices	64
A	MATLAB programs for rigid body MotionMonitor™ model.....	64
B	MATLAB programs for rigid body Vicon model.....	84
C	Data for the mathematical model	102
D	MATLAB programs for mathematical (triangle) model.....	129
E	MATLAB programs for cylinder model.....	140
F	Table of Woltring cylinder experiment and cylinder model.....	155

List of Tables

	Page
Table 3-1: ICC Values.....	28
Table 3-2: Average intersection for flexion-extension on the left	33
Table 3-3: Average intersection for internal-external rotation on the left	33
Table 3-4: Average intersection for flexion-extension on the right	33
Table 3-5: Average intersection for internal-external rotation on the right	33
Table 3-6: Average percent difference in distance for MotionMonitor™ and Vicon.....	39
Table 3-7: Mean and Standard Deviation of Mathematical Data.....	40
Table 3-8: Mean and Standard Deviation of $z=3$ at different centers.	41
Table 3-9: Noise Characterization and Helical Axis Error of Mathematical Data.	45
Table 3-10: Percent change in triangle leg length with increased noise.	46
Table 3-11: ANOVA between days.	47
Table 3-12: ANOVA between trials.....	48

List of Figures

	Page
Figure 1-1: Structures of the knee	3
Figure 1-2: Superior view of the knee.....	5
Figure 1-3: Types of meniscal tears	7
Figure 1-4: Components of the helical axis	11
Figure 2-1: Components of the helical axis.	16
Figure 2-2: Photo of rigid body model.....	18
Figure 2-3: View of marble indentions	19
Figure 3-1: Flexion and extension rigid body model results.....	29
Figure 3-2: Rotation rigid body model results	30
Figure 3-3: Sample calculated helical axis.....	31
Figure 3-4: Screen capture MotionMonitor™ helical axis	32
Figure 3-5: Average intersection location on rigid body model	34
Figure 3-6: Intersection of helical axis.....	35
Figure 3-7: Temporal plot of distances	36
Figure 3-8: Velocity vs error of rigid body model	36
Figure 3-9: Histogram of velocities for a typical trial.....	37
Figure 3-10: Comparison of MotionMonitor™ and Vicon Error	38
Figure 3-11: Mathematical (triangle) model distance results.....	40

Figure 3-12: Mathematical model angle results	42
Figure 3-13: Modeled data and helical axis	42
Figure 3-14: Modeled data and helical axis for first 50°	43
Figure 3-15: Helical axis intersection of the model data	43
Figure 3-16: Histogram of normally distributed noise for modeled data.....	44
Figure 3-17: Modeled data with random noise	45
Figure 3-18: Percent change in leg length at different noise amplitude.....	46
Figure 3-19: Cylinder model distance results	48
Figure 3-20: Cylinder model angle results.....	49

List of Abbreviations

IOR = Instantaneous axis of rotation

OKC = Open kinetic chain

HA = Helical axis

DOF = Degrees of freedom

\hat{n} , \mathbf{n} = Unit vector of helical axis

\hat{s} , \mathbf{s} = Radius vector of helical axis

ϕ = Angle of rotation about the helical axis

t = Translation along the helical axis

P₁ = Position 1

P₂ = Position 2

[R] = Rotation matrix

\hat{v} , \mathbf{v} = Translation vector from position 1 to position 2

IR = Infrared

x, X = X axis

y, Y = Y axis

z, Z = Z axis

r = Radius

f = Frequency

ICC = Intra-class correlation

fe = Flexion-extension

rot = Internal-external rotation

L = Left marble location

R = Right marble location

F = Femur

T = Tibia

ANOVA = Analysis of Variance

Abstract

COMPUTATIONAL MODELING OF THE HELICAL AXIS IN THE HUMAN KNEE— DETECTING CHANGES IN THE TRANSVERSE PLANE

By Emily Joyce Carney, B.S. Biomedical Engineering

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

Virginia Commonwealth University, 2009

Major Director: Gerald E. Miller Ph.D.
Chair, Biomedical Engineering

The purpose of this study was to build a helical axis algorithm capable of detecting changes in the helical axis intersection of the tibia and femur in human knees. This algorithm was designed and tested using three models: a multiplanar rigid body model, a mathematical model, and a cylinder model.

The multiplanar rigid body model closely mimics knee movement without supporting ligamentous structures, thus the movement was unconstrained. In this model kinematic position data were collected using both MotionMonitor™ (MotionMonitor™

Version 7.7, Innovative Sports Training, Chicago, IL) and Vicon Nexus 1.3 software system (Oxford Metrics Group, Oxford, UK). To test the algorithm under more controlled conditions and eliminate kinematic collection errors, a mathematical model was used. The cylinder model was used to obtain constrained movement data using MotionMonitor™.

The results from the two models using a kinematic collection system were inconclusive due to the large errors in the collection process. The mathematical model validated the helical axis algorithm. Statistical analysis of kinematic data showed good correlation between the data but the errors from this data were too large to be useful in a clinical setting. Further research is needed to use the helical axis to model the tibial and femoral rotation on the meniscus.

CHAPTER 1–Introduction

During normal ambulation, the knee joint (tibiofemoral joint) goes through motion in several planes. *Sagittal plane motion*, the most obvious, is described as flexion and extension. It is best observed from the side and has an axis that is oriented in a left-to-right direction. During normal walking, knee movement ranges from 0° of flexion (full extension) as the heel strikes the floor to 60° of flexion as the foot clears the floor and advances to the next heel strike [1]. The axis of rotation in the sagittal plane is not fixed and has been described as an instantaneous axis of rotation (IOR) that migrates in a posterior direction with knee extension [2]. *Transverse plane motion* is movement in a plane that is parallel to the floor. Rotation in this plane is described as internal and external rotation around an axis oriented in a superior-to-inferior direction. The knee goes through this motion during the normal gait cycle but to a lesser degree than flexion and extension. The rotation occurs as a result of joint arthrokinematics. During open kinetic chain (OKC) knee extension, the tibia is moving on a fixed femur. Because of the shapes of each joint surface, the tibia is rolling and gliding in an anterior direction [3, 4]. This motion stops first on the lateral condyle since it has a smaller surface area. The result is lateral tibial rotation (or medial femoral rotation) as the knee extends [2].

Frontal plane motion is the last plane to be considered at the tibiofemoral joint.

Movement in this plane occurs around an anterior-to-posterior axis. In a perfectly aligned knee where the load bearing axis runs down the middle of the leg from the hip through the knee to the ankle, the knee joint has minimal rotation in this direction and redundant supporting ligamentous structures to limit this rotation [5]. When the knee is malaligned, more rotation occurs in the frontal plane than in the perfectly aligned knee. Movement in this plane would be best observed from the front and is defined by varus and valgus rotation.

The tibiofemoral joint is a synovial joint. The weight bearing surfaces of the joint are covered with cartilage. Cartilage is responsible for decreasing the friction between adjacent bones [2]. Cartilage is living tissue but has no blood supply. It receives nutrients and disposes of waste products through cyclic compression and decompression during weight-bearing activities like normal ambulation. These activities can produce points of high stress on cartilage and have the potential to damage this tissue. Additional structures exist to reduce this point of stress. These structures are the medial and lateral meniscus [6]. They act to increase the contact surface area between adjacent surfaces; therefore, they decrease the associated pressure. Meniscal defects or tears change that pressure distribution and potentiate injury to the underlying cartilage [6].

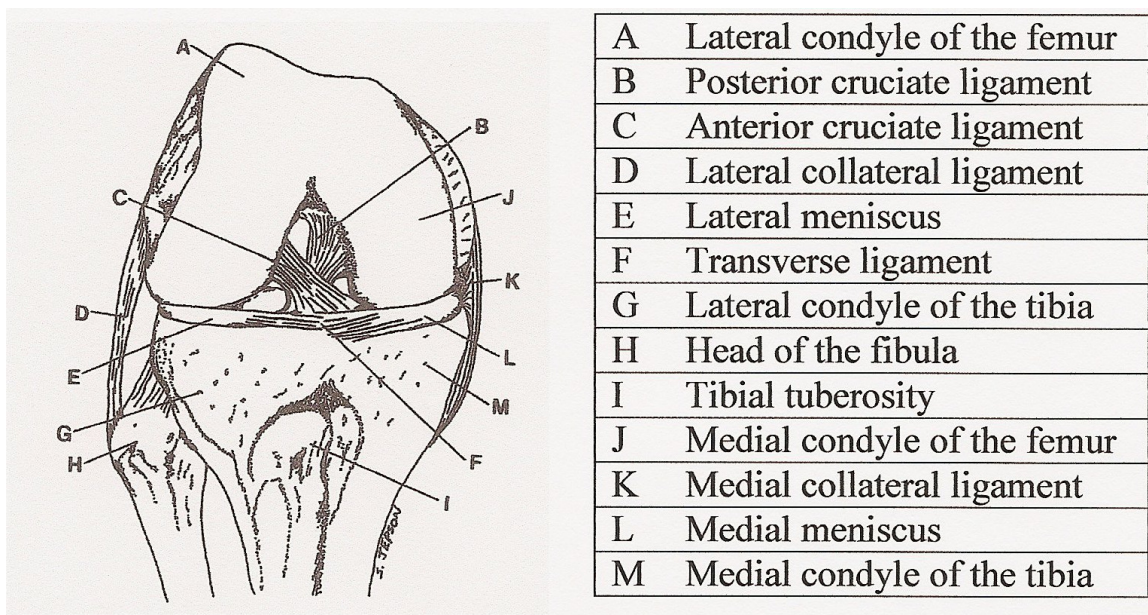
Recent research suggests that meniscal damage most often occurs in the medial meniscus, near the posterior lateral margin [7]. Some physicians question whether or not the transverse plane axis of rotation is migrating to this region and is contributing to meniscal deterioration. To that end, this research attempts to build a helical axis (HA)

algorithm capable of detecting changes in the axis of rotation in the transverse plane within the tibial plateau during gait. If successful, this may lead to not only a better understanding of the mechanism of this injury but also to potential therapeutic interventions.

1.1 Anatomy of the Knee

The knee is the largest and most complex joint of the body [8]. It is a modified hinge joint that is composed of four bones—femur, tibia, patella, and fibula [9]. Passive elements (or soft tissue) are designed to minimize and control non-sagittal plane movements. These tissues include the menisci, ligaments (anterior cruciate ligament, posterior cruciate ligament, medial collateral ligament, and lateral collateral ligament), tendons, and cartilage [6]. These structures are shown in Figure 1-1.

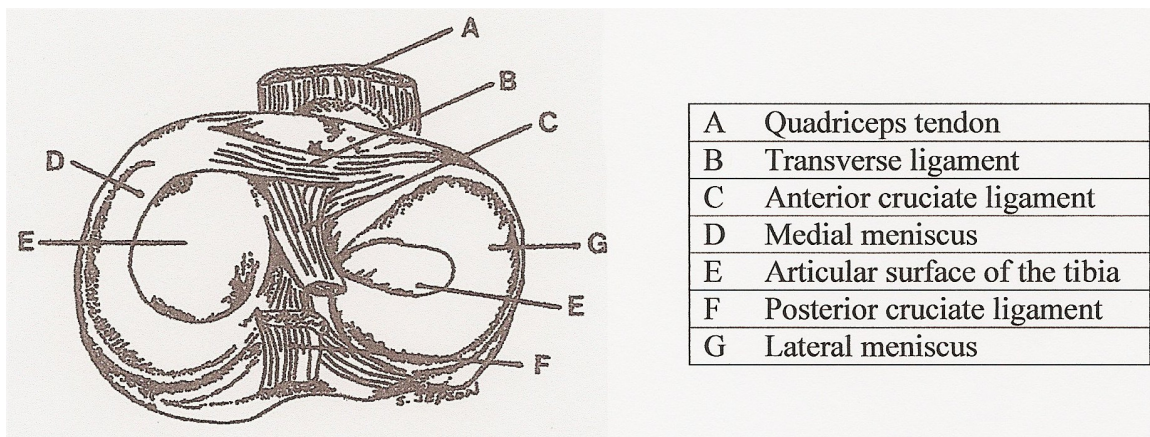
Figure 1-1: Structures of the Knee [2].



The meniscus contributes to optimal knee function and knee joint stabilization. They help stabilize the knee joint through load sharing, stress distribution, anteroposterior stability, articular cartilage nutrition, and joint congruence [3]. The menisci absorb about 50% of the axial load on the joint and distribute the remaining pressure more equally across the underlying cartilage. Thus, the meniscus functions to protect the articular cartilage from excessive load [7]. One of the main functions of the meniscus is to provide shock absorption and distribution of the force of the femur over a larger area of the tibia [3].

The menisci are two semi-lunar, wedge shaped fibro-cartilaginous structures attached to the proximal end of the tibia on the medial and lateral condyle surfaces. The anterior horn of the medial meniscus is attached to the tibial plateau in front of the anterior cruciate ligament while the anterior horn of the lateral meniscus is attached to the tibia behind the anterior cruciate ligament. The posterior horn of the medial meniscus is anchored between the attachments of the lateral meniscus and the posterior cruciate ligaments. The posterior horn of the lateral meniscus is attached in front of the posterior end of the medial meniscus [10]. The outer border of the medial and lateral meniscus is attached to the tibia via coronary ligament while the medial meniscus is also attached to the medial lateral collateral ligament [10]. The inner edge of the meniscus is paper thin lying freely on the floor of the condylar surfaces of the tibia [4]. The medial meniscus has a larger curvature than the lateral meniscus. The average width of the medial meniscus is 10 millimeters while the average width of the lateral meniscus is about 12 millimeters [6]. Figure 1-2 illustrates the position of the meniscus on the knee.

Figure 1-2: Superior View of the Knee [2].



The menisci are vascularized only in its peripheral zones, and the lateral meniscus has better vascular supply than the medial meniscus. The menisci are divided into three histological zones: (1) cartilaginous avascular zone which contains the inner $\frac{3}{4}$ of meniscus; (2) fibrous zone which contains capillaries; and (3) parameniscal zone consisting of loose connective tissue [6]. The medial meniscus is more likely to tear than the lateral meniscus due to less mobility of the medial meniscus because it is also attached to the medial collateral ligament.

1.2 Meniscal Tears

Damage to the meniscus generally occurs when the cartilage is under an abnormal load like excessive rotation of the femur on a fixed tibia in the transverse plane [4]. The medial meniscus tears more frequently than the lateral meniscus primarily due to the larger curvature of the medial meniscus. Tears occur when there is a combination of flexion-rotation or extension-rotation motion [3]. Posterior root attachment of the medial meniscus is the most likely of all attachment sites to experience a tear because this site absorbs most of the force applied to the medial meniscus [7].

The initial tear to a meniscus usually leads to partial detachment from the joint capsule. The joint capsule contains a white fibrous tissue outer layer and an inner layer that is the synovial membrane. Typically, the initial symptoms of a meniscal tear are masked by injuries to the ligaments. If a meniscal tear is not given adequate time to heal, it will gradually become enlarged from continued stresses such as abnormal loads and excessive rotation in the transverse plane [6].

The meniscus is initially torn by being drawn into the joint space where tension force fixes it between the femur and tibia condyles. Sudden extension of the joint creates more tension forces between the fixed meniscus and the joint capsule. A tear occurs when the allowable stress of the meniscus is exceeded. If the tear is small and given time to heal, the meniscus will repair itself.

Several classifications of meniscal tears are longitudinal (bucket handle) tears, radial tears, horizontal tears, vertical tears, and oblique tears. Figure 1-3 illustrates four classifications of tears [11]. The classification of meniscal tears is based on the direction of the fibers torn and the location of the tear. Longitudinal tears, the most common tears, occur parallel to the circumferential fibers. Bucket handle tears, a type of longitudinal tears, are long vertical tears. Radial tears frequently occur in the middle third of the lateral meniscus and are directed outward along any axis from the inner edge [12]. Horizontal tears extend parallel to the transaxial plane on the tibial plateau while vertical tears extend perpendicular to the transaxial plane. Oblique tears extend in neither the radial nor circumferential directions and may commonly be the result of the anterior portion of a bucket handle tear [3]. Meniscal tears are generally the result of partial tears

which have increased in size due to repeated damage. Because the lateral meniscus has only loose connections to joint capsule, no direct connection to lateral collateral ligament and more mobility, they are subjected to less tension forces; therefore, they have fewer lateral tears than the medial meniscus [6].

Figure 1-3: Types of Meniscal Tears [11].



1.3 Knee Movement and Gait

The gait cycle consists of a stance phase and a swing phase. In the stance phase the knee determines the leg's stability, and during the swing phase the flexibility of the knee is the main determinant of the leg's ability to advance. The stance phase consists of about 60% of the normal gait cycle while the swing phase is about 40% of the normal gait cycle [1].

A majority of the motion in the knee is in the sagittal plane with very little motion in the transverse and frontal planes. The motion in sagittal plane is used in the progression of stance phase and limb advancement in swing phase with the average range of motion being 0-60 degrees. This range of motion varies due to the subject's walking speed and style. Motion in the transverse plane occurs as the body moves from behind to ahead of the supporting limb with an average transverse rotation of about eight degrees.

The magnitude and the direction of the transverse rotation change throughout the gait cycle. Motion in the frontal plane assists vertical balance over a single support limb with an average frontal rotation of eight degrees. During stance the frontal plane motion is abducting and during swing the knee returns to a more neutral position by adducting [1].

During passive flexion and extension motion the knee moves in a classic pattern described as the screw home mechanism. In this pattern, as the knee is flexed, the tibia rotates medially and as the knee is straightened, this rotation is reversed [13]. The last few degrees of motion in the transverse plane lock the knee in extension; this is the screw home mechanism. The knee is unlocked by the femur rotating laterally on the tibia for flexion to occur [2]. The screw home mechanism involves both translation and rotation. The screw home mechanism is a feature of the knee as a result of the medial femoral condyle having a longer articular surface than the lateral femoral condyle on the tibial plateau. The continual rotation and loading and unloading involved in the screw home mechanism puts the meniscus under stress that can cause it to tear.

1.4 Helical Axis

For a kinematic analysis of joints, a series of finite steps are used to model movement of one bone relative to another. There are three principle methods used to describe the knee joint motion: Euler angles, Grood and Suntay joint coordinate system, and helical axis (screw axis) [14]. Although the Euler angles are well suited for clinical evaluation of joints with 3 degrees of freedom (DOF) such as the ball and socket hip joint, they are less suited for use with a joint with 6 DOF such as the knee [9]. However,

for ease of modeling the knee is frequently modeled as 3 DOF, due to minimal movement in the frontal plane, thus the movement in this plane is not a factor.

Euler angles do not accurately describe the translation of the knee joint [9]. Limitations of the Euler angles include motion that is described around a single point that corresponds to the center of rotation of a joint. When two axes are aligned, then the Euler angle representation loses one DOF [15]. In addition, Euler angles are dependent on the specific rotation sequence and precise identification of the directions of the relative coordinate system applied to the bones making up the joint. Grood and Suntay method has an advantage over Euler angles because the joint displacements within the system are independent of the order in which the translation and rotation occur, much like HA. However, the Grood and Suntay coordinate system is heavily dependent on the landmark locations such as the tibial condyles and intercondylar notch. As a result it is best suited for cadaveric models [9].

Unlike the Euler angles the helical axis is independent of the order that the translation and rotation occurs. The HA is independent of the coordinate system being used since the rotation about and translation along the axis does not change with a changing coordinate system. Since the HA can be described relative to anatomical landmarks but is not affected by the imprecise location of these landmarks, the reference axis is less important [14, 16]. HA is probably a better means of comparing the motion of different individuals than the use of an anatomical system because anatomical systems are affected by both skin movement and the landmark location, while HA are only

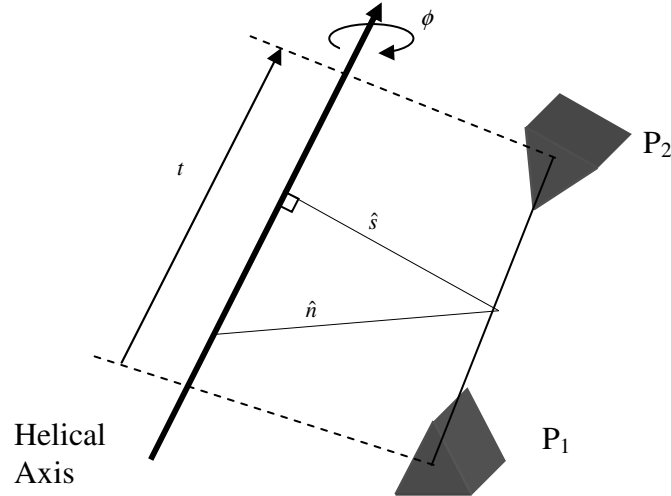
affected by skin movement [17]. The affect from skin movement is reduced further when using the mean HA [16].

Some commercially available software computes the helical axis, but most of these programs anchor the HA throughout the motion at the center of the joint. Thus the HA shown may not be a true representation of the instantaneous HA.

Helical axis is used to represent three dimensional motions. These motions are described by a rotation about an axis and a translation along an axis. Due to difficulties finding the instantaneous HA, it is simulated by determining a finite HA in small steps. Most publications use finite HA to approximate the instantaneous HA [18]. The finite HA is composed of eight parameters: the unit vector, $\hat{n} = \langle n_x, n_y, n_z \rangle$; the radius vector, $\hat{s} = \langle s_x, s_y, s_z \rangle$; the angle of rotation about the HA, ϕ ; and the translation along the HA, t .

The line connecting the endpoints of the unit vector and radius vector represent the helical axis. The unit vector indicates the orientation of the HA. The radius vector is gives the position of the HA. The endpoint of the radius vector is defined in Spoor [19] as a point along the HA closest to the origin of the reference coordinate system and perpendicular to the HA, but the endpoint of the radius vector can be any point along the HA. Figure 1-4 illustrates the different components of the HA.

Figure 1-4: Components of the helical axis. (\hat{n} : unit vector; \hat{s} : radius vector; ϕ : angle of rotation about the helical axis; t : translation along the helical axis; P_1 and P_2 : position 1 and position 2)



The finite helical axis is calculated as the object is moved from position 1 to position 2. A common problem in the calculations of finite HA is the closer together these points are, the closer the model is to the actual path of movement; however, as the points get closer together, the error in the calculated rotation matrix becomes greater. This impacts all the parameters associated with the HA [9]. Increased error is due to two things: (1) in small increments, the errors are a larger percentage of the total increment and (2) as the step size decreases, so does the angle of rotation, at very small angles of rotation, the HA is no longer unique.

There are several limitations of the helical axis method. The HA is undefined when there is no angular rotation, i.e. pure translation components, which occurs during the normal gait cycle at the beginning and end of both stance and swing phases [20]. Angles of rotation less than 3° [20] to 5° [21] are interpreted as no rotation. Another limitation is its sensitivity to the size of translation and angle of rotation [21]. Since the

HA is continually changing position during motion, it is difficult to know its position at any given moment [22]. Since it is changing position, the mean HA is commonly used to represent the HA. Using this mean HA has the advantage of being dependent on the motion and not affected by the variability associated with precise identification of anatomical landmarks [16].

The Euler angles are used more than helical axis in clinical settings because their description of motion is more closely connected to the clinical description of motion [9]. Since the Grood and Suntay joint coordinate system is heavily dependent of landmark location, they are best used on cadaver models. The HA which gives a full description of the 6 DOF is used by engineers in models of the knee and in the design of prostheses [9].

1.5 Specific aims

The specific aims of this study are to: (1) attempt to test the helical axis algorithm using rigid body model to capture multiplanar motion data around multiple fixed axes, (2) attempt to test helical axis algorithm using modeled data, and (3) attempt to test the helical axis algorithm using a rigid body in a constrained planar movement pattern. It is hoped that characterizing the function of this method in a series of controlled experiments will help determine its usefulness in a clinical venue.

CHAPTER 2–Methods

The helical axis algorithm used in this research was developed based on the equations presented by Spoor [19]. The calculations were performed using MATLAB (Version 7.4.0.287 (R2007a)). Orienting the HA from one position to another position in a rigid body required the computation of four components using a 3x3 orthogonal

rotation matrix $\left([R] = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \right)$ and a translation vector (\hat{v}) [19]. Determining

the HA required three or more points, three of which have to be non-collinear points per sampling interval to compute the orthogonal rotation matrix and the translation vector [19]. The equations used in the HA algorithm are presented below.

1. The angle of rotation about the HA (ϕ) can only be between 0 and π . It is

calculated using either equation 1 or 2. If $\sin \phi \leq \frac{1}{2}\sqrt{2}$ equation 1 is used to

calculate ϕ ,

$$\sin \phi = \frac{1}{2} \sqrt{(R_{32} - R_{23})^2 + (R_{13} - R_{31})^2 + (R_{21} - R_{12})^2}. \quad (1)$$

If $\sin \phi > \frac{1}{2}\sqrt{2}$, then the equation 2 is used to find ϕ . The arcsine has real values for the interval $[-1,1]$, which can be reduced to $[0,1]$ because of the symmetry of the arcsine function. The range $[0, 1]$ can be further reduced by the identity

$$\sin^{-1}(x) = -\sin^{-1}(\sqrt{1-x^2}) + \frac{\pi}{2} \text{ which reduces the interval to } \left[0, \frac{\sqrt{2}}{2}\right], \text{ therefore}$$

equation 2 is used to find ϕ when $\sin \phi > \frac{1}{2}\sqrt{2}$ [23].

$$\cos \phi = \frac{1}{2}(R_{11} + R_{22} + R_{33} - 1). \quad (2)$$

2. The endpoint of the unit vector (\mathbf{n}) is one point on the HA and indicates the orientation of the HA. If $\sin \phi \neq 0$ or if $\phi \leq \frac{3}{4}\pi$, then the unit vector is calculated using equation 3:

$$\sin(\phi)\hat{n} = \frac{1}{2} \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix}. \quad (3)$$

Due to the fact that division by zero is undefined, if $\sin \phi = 0$ or if $\phi > \frac{3}{4}\pi$, then

equation 4 is used to calculate the unit vector

$$\frac{1}{2}([\mathbf{R}] + [\mathbf{R}]^T) = \cos \phi \mathbf{I} + (1 - \cos \phi)\hat{n}\hat{n}^T. \quad (4)$$

The process of solving equation 4, involves resolving a square root which can be either negative or positive. The correct sign on the unit vector is determined by

satisfying equation 3, no matter which equation is used to solve for the unit vector.

3. The endpoint of the radius vector (\hat{s}) is a second point on the HA and is calculated to be closest to the origin of the reference coordinate system and perpendicular to the HA in Spoor [19]. It gives the position of the HA. Equation 5 is used to find the radius vector

$$\hat{s} = -\frac{1}{2}\hat{n}\times(\hat{n}\times\hat{v}) + \frac{\sin\phi}{2(1-\cos\phi)}\hat{n}\times\hat{v}. \quad (5)$$

4. The translation along the HA (t) is calculated using equation 6:

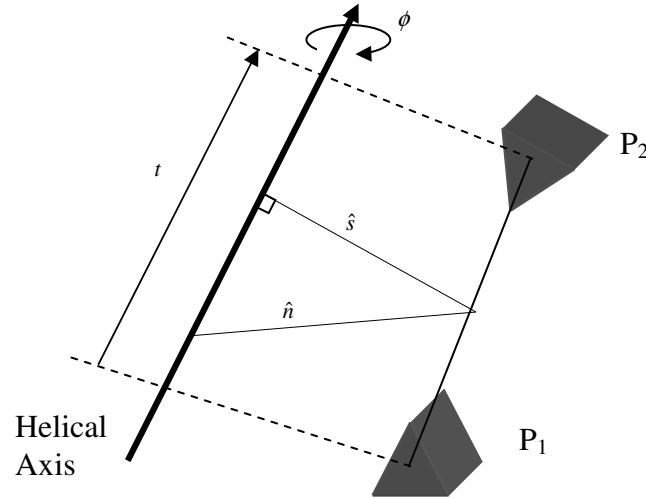
$$t = \hat{n}^T \hat{v}. \quad (6)$$

These equations are only valid when $\phi \neq 0$ because the HA is undefined in the absence of rotation [9,14,19,20,24]. In the case when $\phi = 0$ and $\hat{v} \neq 0$, where the motion is pure translation, non-unique values for the unit vector, radius vector and translation along the HA are defined by equation 7:

$$t = \sqrt{\hat{v}^T \hat{v}}; \quad \hat{n} = \frac{1}{t}\hat{v}; \quad \hat{s} = [0,0,0]. \quad (7)$$

When $\phi = 0$ and $\hat{v} = 0$, there is no motion. This case is not of interest, and these data are ignored in the algorithm. These components of the HA are shown in Figure 2-1.

Figure 2-1: Components of the helical axis (\hat{n} : unit vector; \hat{s} : radius vector; ϕ : angle of rotation about the helical axis; t : translation along the helical axis; P_1 and P_2 : position 1 and position 2)



The intent of this research was to develop a methodology that would be able to determine the transverse plane axis of rotation in the tibiofemoral joint during clinically relevant movement (i.e. ambulation, stair climbing, turning/cutting, etc). Experiments were designed to constrain the helical axis motion so that computed estimates could be compared to known planar intersections. Collected data would be used to compute the reliability and validity of each approach, characterizing the benefits and deficits of each model. Three models were used in this effort and are described below.

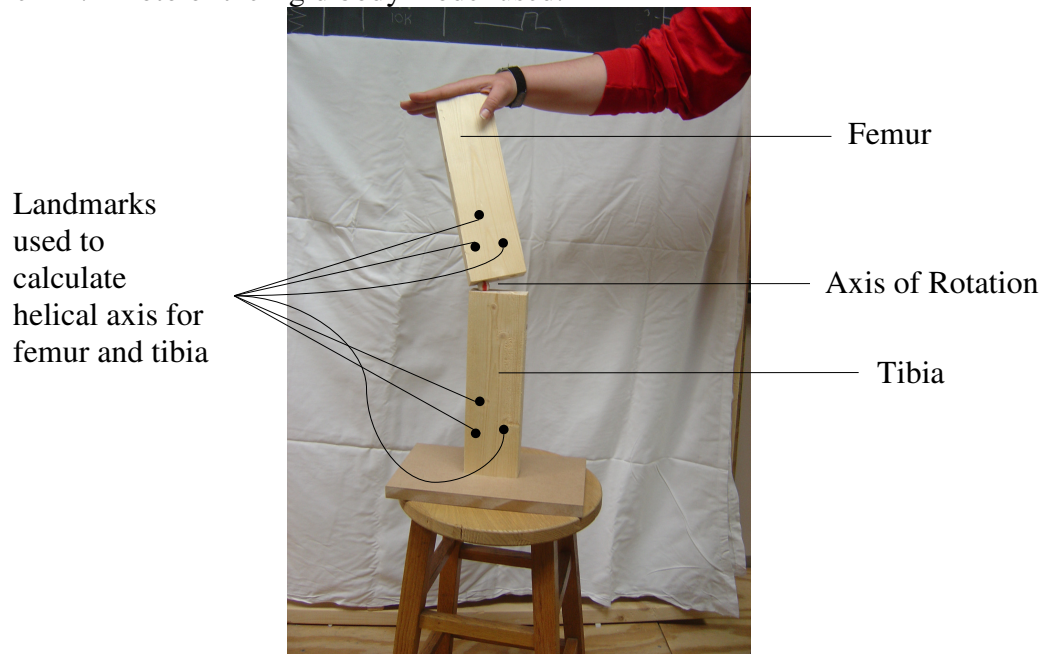
There are many sources of error in an algorithm of this nature. These include numerical computational errors, algorithm errors, or input data errors. The three models described below each attempt to constrain errors in a different way. The multiplanar rigid body, sometimes referred to as the wooden model was the model that was closest to mimicking the movement of the knee because it provided an exaggerated flexion-

extension movement and internal-external rotation movement. This combination of movement was not seen in the other two models. It used a rigid body assumption where movements were completely controlled by the experimenter. The mathematical (triangle) model eliminated errors associated with kinematic measurements systems because it did not rely on collected data. The cylinder model was a constrained rigid body that provided position data. The wooden model most closely resembled the intended use for this algorithm, but errors in the kinematic data made it difficult to thoroughly test the algorithm using this model. The triangle model eliminated errors associated with data collection. Since there were fewer moving parts in the cylinder than the wooden model, errors induced by the experimenter were minimized. These different models and data sets provided the mechanism for testing the HA algorithm.

2.1 Multiplanar Rigid Body Model

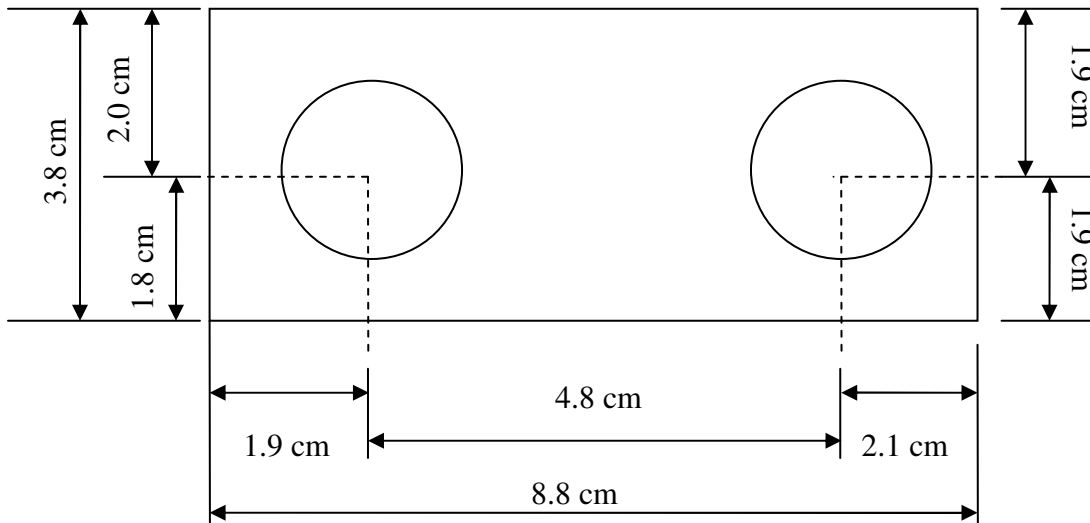
In the rigid body model two pieces of wood were used to model knee movement. The tibia was represented by one piece of wood (3.8 x 8.8 x 27.9 cm) mounted to a fixed base. The femur was represented by an unconstrained second piece of wood of similar dimensions not mounted to a base. The joint was represented by a marble which was the axis of rotation for this model. Figure 2-2 is a picture of the wooden rigid body model that the experiments were run on.

Figure 2-2: Photo of the rigid body model used.



In this model, the femur was considered unconstrained since no ligamentous structures were included. This allowed the femur to be easily manipulated by the experimenter. One end of each piece of wood had two semispherical depressions representing the medial and lateral condyles and medial and lateral meniscus. This allowed a marble to be held between the two pieces of wood to constrain the axis of movement and essentially act as a joint. Figure 2-3 shows the top view of the shank.

Figure 2-3: Top view of the shank of the wood model to show locations for the marble.



This design provided 3 degrees of freedom (DOF) movement between the two rigid bodies. The tibiofemoral joint is normally described as having 2 DOF with primary movement in flexion and extension (the sagittal plane) [9]. The model provided movement in the sagittal plane in a range of from about 25° extension to about 100° flexion. Internal-external rotational movement in the transverse plane also occurs in normal knee movement, typically during the last 10° of knee extension. This rotational movement is due to the differences in the medial and lateral condylar surfaces. The model provided movement in the transverse plane in a range of 0° external rotation to about 85° external rotation.

During data collection, the experimenter was responsible for moving the knee around the marble axis in a multiplanar pattern. Two distinct movement patterns were used: (1) flexion-extension (sagittal plane) and (2) internal and external rotation (transverse plane). Each pattern had to include components of the other to avoid

instability in the helical axis algorithm because when there is no angular rotation, the HA is non-unique and then not a suitable model. Care was taken to maintain contact between the wood and marble at all times. This provided the most robust data set for the HA algorithm to use. Data were collected with the marble in a position representing the medial condyle and in a position representing the lateral condyle. These points were separated by a distance of 4.8 cm. Data were collected using two different kinematic collection systems and differences are reported in the results section of this paper. The specific methods of data collections are described below.

Data collection using the MotionMonitor™ kinematic system

The MotionMonitor™ (MotionMonitor™ Version 7.7, Innovative Sports Training, Chicago, IL) system used in this data collection was configured with Ascension Technologies hardware (Ascension Technology, Burlington VT). This is an electromagnetic system that tracks sensors using field strength as a measure of position. It is capable of collecting 6 DOF data from a single sensor at a sampling rate of 100 Hz. A single sensor was attached to the approximate center of each rigid body. As part of the set up procedure, landmarks were defined relative to each fixed sensor so that the relative coordinate locations could be exported and used in the HA calculation. Three landmarks were digitized in the approximate shape of a triangle, near the joint representing the femur and near the base representing the tibia. The triangle shape was used to ensure that the non-collinear landmark constraint was maintained. In addition to these landmarks, a landmark was digitized at the approximate center of each indentation on the tibia. This was done to allow the comparison of the location of the HA relative to the physical axis

of rotation. The four corners of the proximal end of the tibia were digitized to define a plane representing the tibial plateau. The purpose of this was to define the joint space, and this plane was used to determine where the HA intersected the desired plane. In internal-external rotation, the HA intersection was calculated at the tibial plateau (transverse plane), while in flexion-extension movement the desired plane was perpendicular to the tibial plateau, so the plane was in the sagittal plane containing the axis of rotation (center of the marble). Data were collected for 5 seconds. These experiments were run 9 times for each movement with the marbles in the medial condyle location and then in the lateral condyle location. Using these data, the HA was computed and the intersection of that axis with the defined plane was determined (MATLAB program HA1.m in Appendix A).

MotionMonitor™ does not simultaneously collect X, Y, Z data. For a 100 Hz data collection rate, the transmitting antenna is actually operating at a rate of 400 Hz. It has to get through an X, Y, Z and wait state within 1/100 of a second. This could result in errors in reporting the actual location of the sensor. To compensate for this, the data was fit with a cubic spline. The X, Y, and Z data were re-sampled to ensure that the data for all three coordinate locations came from the same point in time. This was done by a MATLAB program HA1_spline.m in Appendix A.

Data collection using the Vicon kinematic system

The Vicon Nexus 1.3 software system (Oxford Metrics Group, Oxford, UK) is an optical tracker that uses reflective markers and infrared (IR) cameras to collect position data. When a single marker is captured by 2 or more cameras, then a 3-dimensional

location of that marker can be determined. Reflective markers were placed on the eight corners of the wood block representing the tibia and eight corners of the wood block representing the femur. Three additional markers were placed in a triangular shape toward the distal end of the femur and toward the proximal end of the tibia to ensure the markers were non-collinear. The marble did not contain a marker because it would not be seen by the cameras. The marble location was calculated in the program using the position of the data from the four corners representing the tibial plateau. Once the set up was complete, position data were captured during the two movements at each of the marble locations. Each experiment was run once at a sampling frequency of 100 Hz. The middle 10 seconds of data from each trial were used to compute the HA and the intersection of that axis with the defined tibia plateau during internal-external rotation and the intersection of the HA with a sagittal plane containing the marble location for flexion-extension (MATLAB program HA2.m in Appendix B).

2.2 Mathematical (Triangle) Model

The performance of the multiplanar rigid body model was less than expected. This performance may have been due to computational errors, algorithm errors, or measurement errors in the kinematic data. In an effort to determine the source of the error, movement data were modeled by mathematically rotating an equilateral triangle inscribed in a circle. A triangle was selected because the three vertices would define 3 non-collinear points required for the HA calculations.

Each triangle vertex was tracked, and X, Y, Z coordinate points were computed. The X and Y coordinate data were created by following each vertex as the triangle

rotated around a central axis. The X and Y movement of a vertex was 120° out of phase with each neighbor, resulting in a slight irregular motion as the triangle was rotated. The Z coordinate was created using a sine wave function to simulate vertical movement at several different frequencies. The data in the X direction were computed using equation 8:

$$x = r \sin(\theta) \quad (8)$$

where r is the radius of the circle. θ are degree measures 0 to 360° in one degree increments used to obtain data points in one revolution. The data in the Y direction were computed using, equation 9:

$$y = r \cos(\theta). \quad (9)$$

The data in the X and Y coordinate direction were created using four different circle radii (0.5, 1, 1.5, and 2 cm). The data in the Z direction were computed using equation 10:

$$z = A \sin(f\theta) \quad (10)$$

where f was one of five different frequencies (2π , 4π , 6π , 8π , and 10π). A is the amplitude of the sine wave, which was limited to 0.1 cm to maintain the integrity of the triangle as it wobbled around the central axis. This amplitude was also consistent with the small movements expected in a real knee [1]. After the XY data for one point was computed the angle of change in the Z direction was calculated. This was used to adjust the coordinates and to compensate for the movement in the Z direction. Once the data were computed for a complete revolution of one vertex, the other two vertices were the same data phase shifted $\pm 120^\circ$. A sample set of these data (with the calculated components of the HA) can be found in Appendix C.

These data were used to compute the finite HA and to determine where and at what angle the HA intersected the plane defined by the triangle (MATLAB program HA3.m in Appendix D). It was assumed that the center of the triangle would be where the HA intersected the plane and the HA would intersect the plane at a right angle. To simulate errors associated with kinematic data collect systems, normally distributed noise was introduced into the model. The effects of this noise were then quantified in subsequent HA calculations.

2.3 Cylinder Model

A third experiment was devised based on a rigid body model described in Woltring [18]. In this model, a cylinder was rolled on a flat surface while collecting positional data from six isotropically distributed landmarks. This was a physical model much like the first experiment, but with some imposed motion constraints. When a cylinder was rolled on a plane, the instantaneous HA should be tangent to the cylinder, i.e. parallel to the plane on which it was being rolled. This model was intended to provide a more robust application of rigid body data collection.

The rigid body model consisted of a 10.2 cm diameter by 35.6 cm long piece of PVC pipe. Data were captured with the MotionMonitor™ at a frequency of 100 Hz during one revolution of the cylinder. The time it took for one revolution varied between approximately 4 and 8 seconds. The cylinder moved 0.85° per step and an average angular velocity of 85° per second. The variation in time was caused by the operator's reaction time and was not a factor in the data collection. For the analysis of the HA calculations the middle four seconds of data were used.

Data were collected from 2 sensors. One sensor was attached to the interior of the PVC pipe while a second sensor was attached to the plane on which the pipe was being rolled. The sensor attached to the interior of the PVC pipe was used as a referenced point for the six landmarks required used in calculating the HA. The six landmarks were points on a simulated sphere inside the pipe. These six landmarks were evenly distributed on the surface of the sphere and were used to collect the position data required by the HA algorithm. A seventh landmark was located at the center of the sphere. It also represented the center of the cylinder and was used to determine the distance from which the HA intersected an imaginary plane. The imaginary plane was defined as perpendicular to the surface on which the cylinder was being rolled, tangent to the path of motion, and contained the point defining the center of the sphere. The second sensor was attached to the table and was used to define the plane on which the cylinder was rolled. To define this plane, three landmarks distributed in a triangular pattern were digitized.

These data were used to calculate the finite HA (MATLAB program, HA4.m in Appendix E). Then the finite HA was compared to the instantaneous HA. The instantaneous HA would be expected to be parallel to the plane on which the cylinder was rolled. Thus, the closer the finite HA was to being parallel to the plane, the closer the approximation was to the instantaneous HA. In this program, a distance from where the HA intersected a plane running through the cylinder, perpendicular to the plane the cylinder is being rolled on, and the center of the sphere was also computed. MotionMonitor™ does not simultaneously collect X, Y, Z data which could lead to errors in reporting the actual location of the sensor. To compensate for this, the data was fit

with a cubic spline. The X, Y, and Z data were re-sampled to ensure that all three coordinate locations came from the same point in time. This was done by a MATLAB program HA4_spline.m in Appendix E.

Once again it should be noted that in this type of experiment, there are many potential sources of error. These include computational errors, algorithm errors, and kinematic measurement errors. Each of the previously described models attempts to constrain these errors differently. The multiplanar rigid body model was the closest to mimicking the real movement of the knee. The mathematical (triangle) model eliminated errors associated with kinematic measurement systems because it did not rely on collected data. The cylinder model was a more constrained rigid body model whose data were again measured via kinematic system.

CHAPTER 3–Results

Prior to evaluating the data, the HA mean and standard deviation were calculated using all the data in each trial. Data outliers, defined as values outside two standard deviations from the mean, were removed. This was done to decrease the impact of errors in the data collection process and instrumentation noise. A new mean and standard deviation were calculated from the remaining values. These data were used in the analysis below.

3.1 Multiplanar Rigid Body Model

Data collected using MotionMonitor™ kinematic system

The rigid body model was manipulated in movements of internal-external rotation (transverse plane) and flexion-extension (sagittal plane) by pivoting the femoral component on a fixed tibia using a marble to represent the joint axis. The HA was computed from the resulting kinematic data. The intersection of the HA with the movement plane was determined for a movement plane that bisected the known axis of rotation (the center of the marble). The difference between the average HA intersection and the known axis of rotation was defined as the error. These average HA coordinate locations and the average error were computed across 36 trials and an intra-class correlation (ICC) was calculated (SPSS-release 17.0.1) for both the femur and the tibia.

The overall ICC value when using the femoral data to compute the HA was 0.990, indicating a very high correlation between these two points. Comparison of these coordinates and errors for the 36 trials using the tibial data to compute the HA yielded an ICC value of 0.893. Table 3-1 has a breakdown of the ICC values for the two different movements at the two different marble locations, using both the femur and tibia reference points to perform the calculations.

Table 3-1: ICC values for the different movements and marble locations for both femur and tibia of the average helical axis intersection and error.

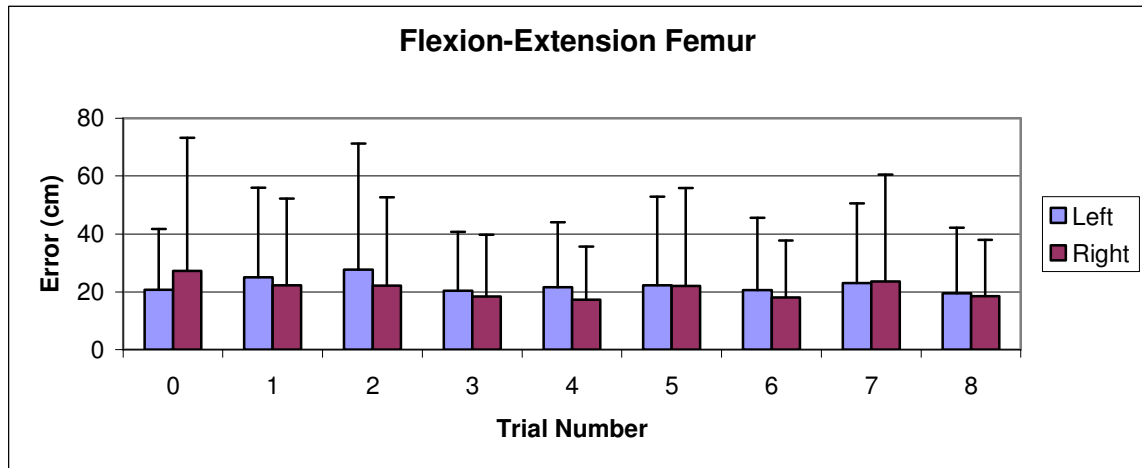
	Femur	Tibia
Overall	0.990	0.893
Flexion-Extension Left	0.997	0.771
Rotation Left	0.999	0.969
Flexion-Extension Right	0.993	0.947
Rotation Right	0.997	0.957

In addition to correlating the intersection of the HA and error of the femur data, an ICC was run correlating the average intersection of the HA and the axis of rotation (marble location). From this an ICC value of 0.997 when the movement was at the left indentation for both movement, and an ICC value of 0.999 when the axis of rotation was at the right indentation for both movements. This ICC showed a good correlation between the average HA intersection and the marble location.

To evaluate the effect of temporal skewing as a result of the MotionMonitor™ data collection method, the distances between the femoral landmarks were examined both before and after the spline adjustment. This was done for each movement and marble location. Prior to data adjustment, there was an average length change of 0.17%. This did not change following the cubic spline temporal correction of the data.

The average error for all the trials and movements of the femur was 18.48 ± 20.09 cm. The average error for all trials and movements of the tibia was much larger (135.68 ± 236.69 cm) since the tibia was stationary during the movement. This was an expected outcome of the helical axis calculation methodology. For this reason only the femur data is plotted. For ease of comparing the flexion-extension pattern and rotation pattern of the femur, the same scale was used for each plot. Flexion-extension pattern data is illustrated in Figure 3-1.

Figure 3-1: Average error of the helical axis of the femur intersection with the tibial plateau from each marble position using flexion-extension pattern



The rotation pattern of the femur and tibia at each marble position are shown in Figure 3-2.

Figure 3-2: Average error of the helical axis of the femur intersection of the tibial plateau from each marble position using internal-external rotation pattern

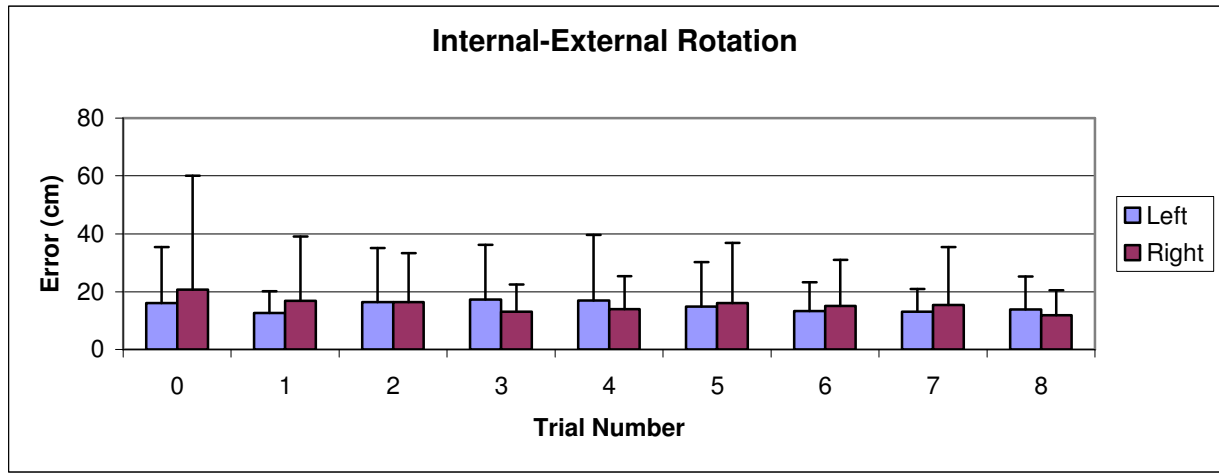


Figure 3-3 shows a sample helical axis represented by the endpoints of the unit vector and radius vector on the rigid body model. This sample HA was taken from a flexion-extension movement on the left marble location. Figure 3-4 is a similarly oriented screen capture from MotionMonitor™ showing a HA from a flexion-extension movement on the left marble location.

Figure 3-3: Sample calculated helical axis on rigid body model for flexion-extension movement on the left marble

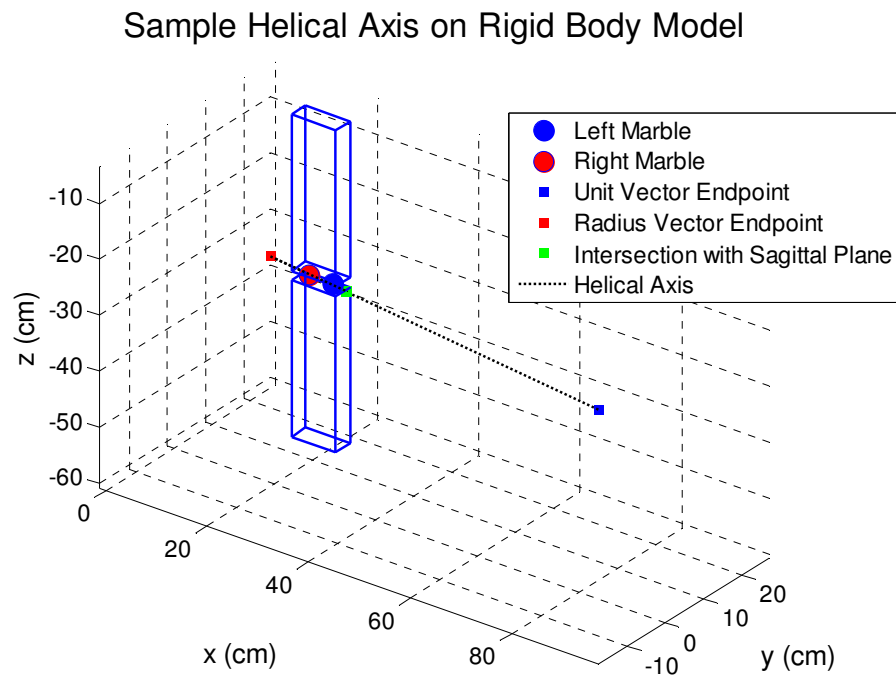


Figure 3-4 is a screen capture from MotionMonitor™ illustrating the helical axis. This screen capture is from a flexion-extension movement with the marble on the left side.

Figure 3-4: Screen capture from MotionMonitor™ showing the helical axis for flexion-extension movement on the left marble.

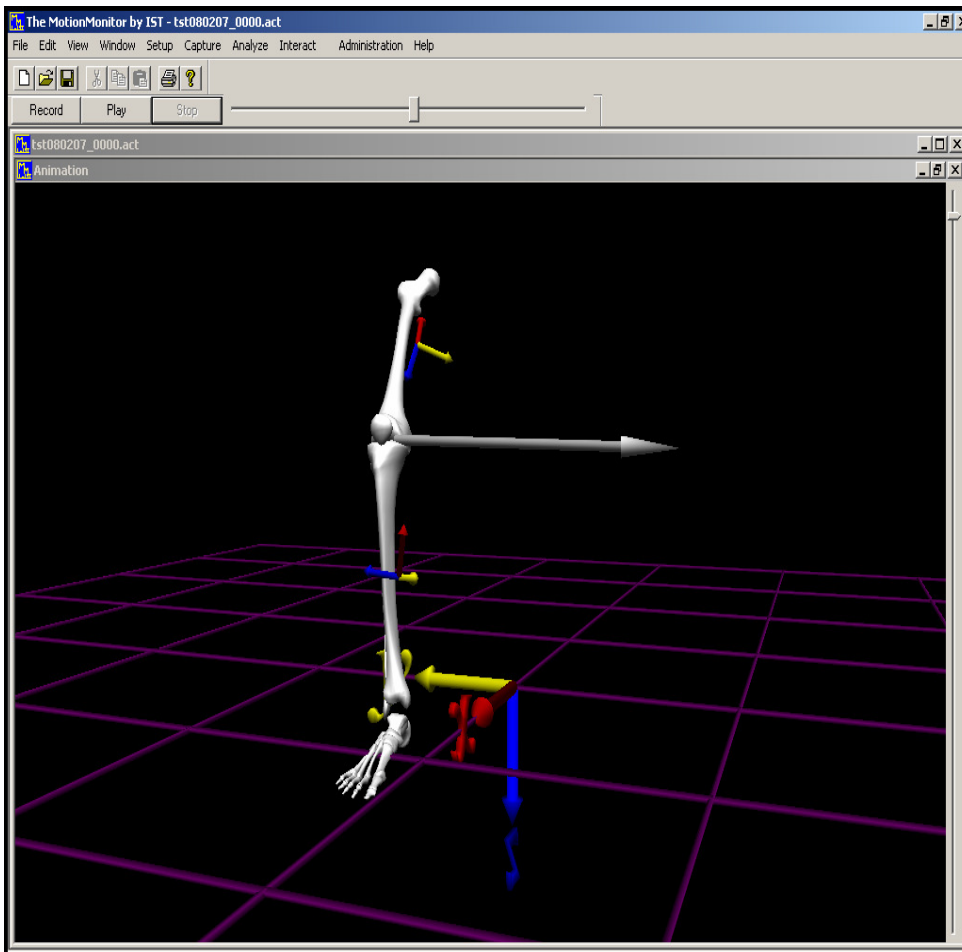


Figure 3-5 shows the average intersection point for each movement and marble location on a graphical figure representing the femur and tibia of the rigid body model and the location of the marbles. Table 3-2 is the average intersection location and average marble location for flexion extension on the left. This is illustrated as the blue square on Figure 3-5. Table 3-3 is the average intersection location and average marble location for the rotation on the left. In Figure 3-5 this point is represented by the red square. Table 3-4 is the average intersection location and average marble location for the flexion-

extension on the right. This is shown in Figure 3-5 as the green square. Table 3-5 is the average intersection location and average marble location for the internal-external rotation on the right and is shown in Figure 3-5 as the yellow square.

Table 3-2: Average intersection location for flexion-extension on the left with the average helical axis error.

	X (cm)	Y (cm)	Z (cm)	Average Intersection Error
Intersection Location	19.33	14.73	-33.01	
Marble Location	19.33	16.15	-32.09	
Absolute Difference	0	1.42	0.92	2.93 cm

Table 3-3: Average intersection location for internal-external rotation on the left with the average helical axis error.

	X (cm)	Y (cm)	Z (cm)	Average Intersection Error
Intersection Location	19.99	16.09	-32.09	
Marble Location	19.33	16.15	-32.09	
Absolute Difference	0.66	0.06	0.00	0.66 cm

Table 3-4: Average intersection location for flexion-extension on the right with the average helical axis error.

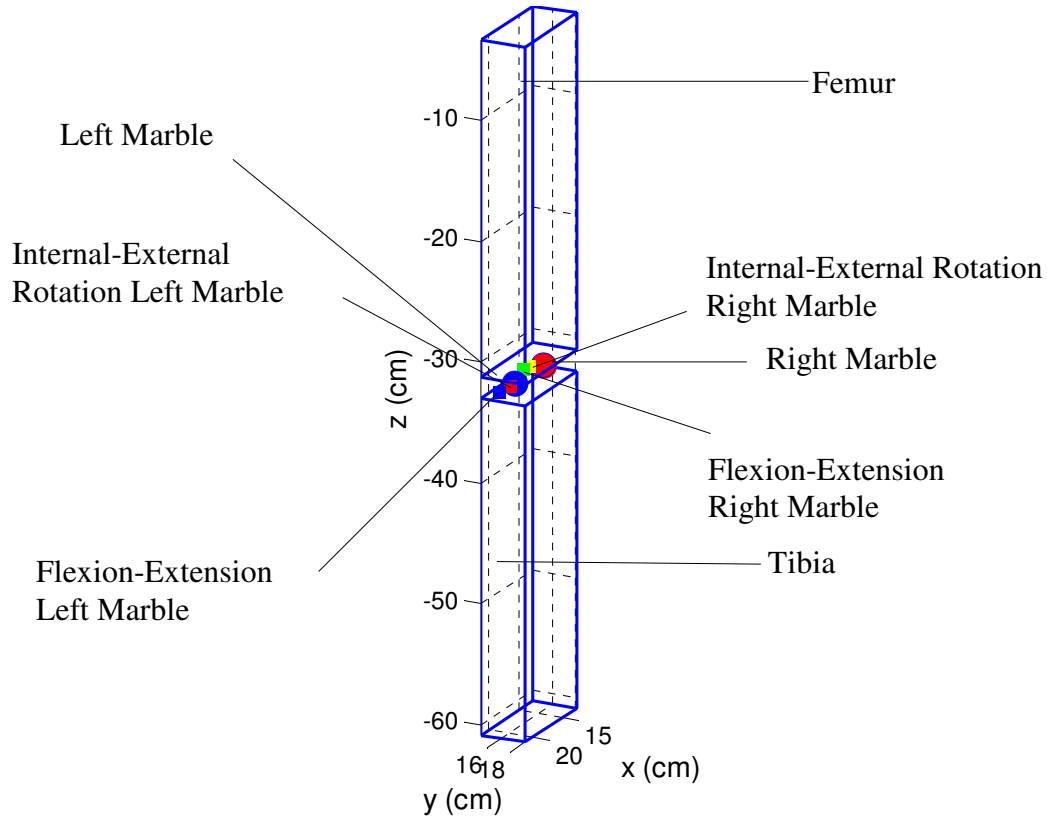
	X (cm)	Y (cm)	Z (cm)	Average Intersection Error
Intersection Location	14.52	14.46	-32.79	
Marble Location	14.52	16.15	-32.09	
Difference	0	1.69	0.70	1.83 cm

Table 3-5: Average intersection location for internal-external rotation on the right with the average helical axis error.

	X (cm)	Y (cm)	Z (cm)	Average Intersection Error
Intersection Location	15.51	15.15	-32.09	
Marble Location	14.52	16.15	-32.09	
Difference	0.99	1	0	1.41 cm

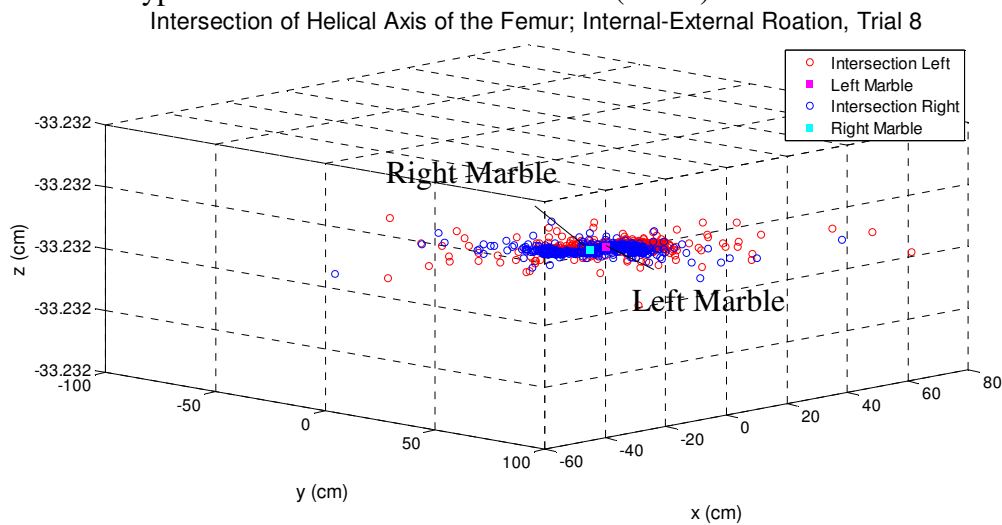
Figure 3-5: Average intersection location on rigid body model from movement of the femur.

Average Intersection Location on the Rigid Body Model



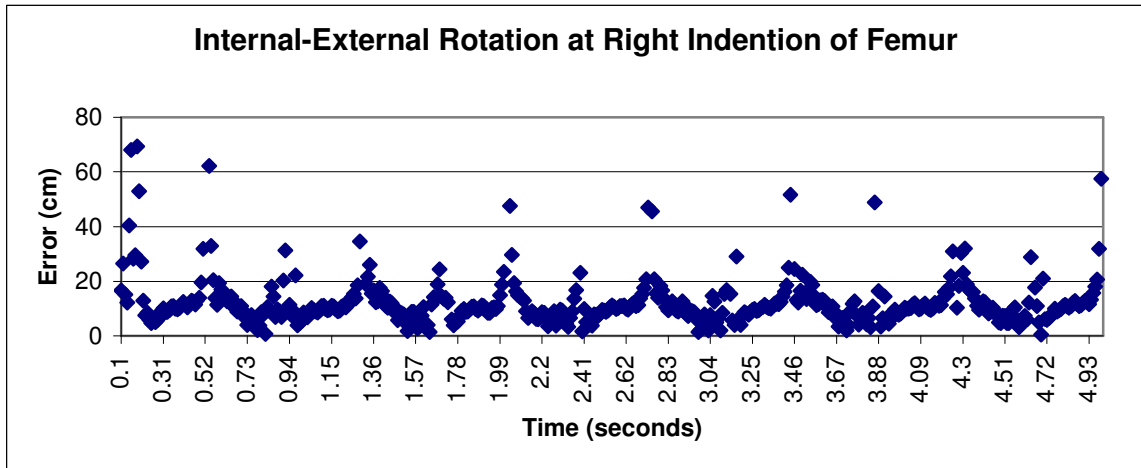
To help illustrate the differences between the intersections of the helical axis with the tibial plateau when the marble was in the left indentation as compared to the right indentation, one typical rotational trial from each indentation was taken, and the intersections of the helical axis and the marble locations were plotted in Figure 3-6. In Figure 3-6 the values on the Z axis are all listed as the same number, because the variance in these points are equivalent to 10^{-9} cm.

Figure 3-6: Plot of the intersection of the helical axis on the tibial plateau and the marble location for a typical internal-external rotation trial (trial 8)



The following plot, Figure 3-7, illustrates the temporal nature of the 3-D error magnitude between the calculated helical axis and the right marble joint center as the femur was manipulated in an internal-external rotation pattern. These data were computed from the “intersection right” and “right marble” data shown in Figure 3-6. Figure 3-7 depicts 7 repeated femur movements. Note the majority of the errors range from 0 to 20 cm.

Figure 3-7: Temporal plot of the errors for typical trial (trial 8)



The cyclical variation in the error plot suggests velocity dependency. In an attempt to characterize the HA error, the error of the movement was plotted against the velocity and no correlation was seen. A sample of one of these plots is shown in Figure 3-8.

Figure 3-8: Velocity vs error scatter plot for a typical trial (trial 8 of rotation right indentation of femur)

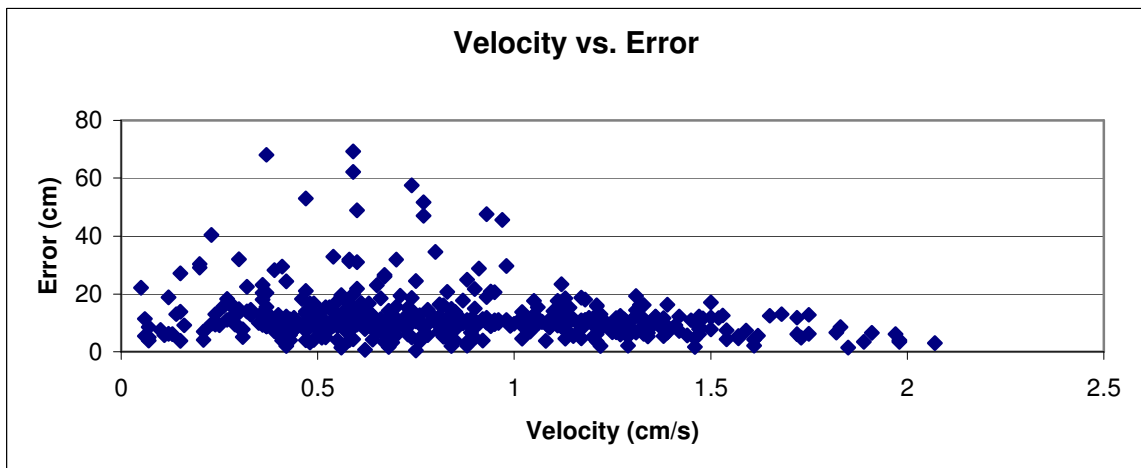
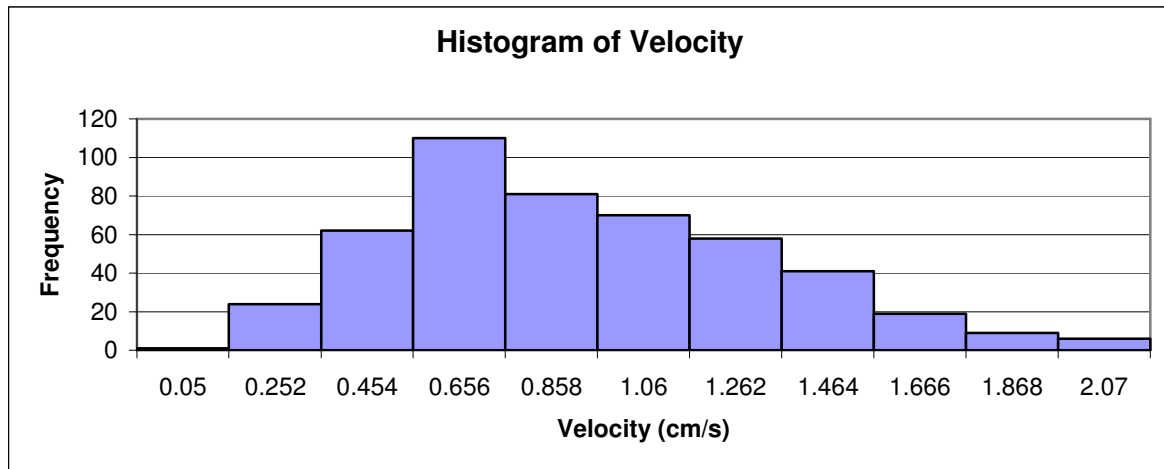


Figure 3-9 is a histogram of the velocity for the movements from a typical internal-external rotation. These results illustrate the variance in the velocity of movement is in a right skewed pattern.

Figure 3-9: Histogram of velocities for a typical internal-external rotational trial (trial 8 on the right indentation).



Data collected using the Vicon kinematic system

The Vicon kinematic data collection system was also used to collect data from the rigid body model. This is an optical system that samples position data at 100 Hz from passive reflective markers attached to the model. Optical systems are often used as the gold standard in kinematic data collection. A comparison of these data to data collected using the MotionMonitor™ was made to determine if kinematic errors could be reduced. The rationale of this approach was based on known data slewing in electromagnetic tracking systems. The MotionMonitor™ does not collect XYZ data simultaneously. The resulting positional errors could influence the HA calculation. To assess these differences, the average error of the HA intersection and marble location for the MotionMonitor™ trials were compared to the average error for the Vicon trial for the

same motion and marble location in Figure 3-10. Note that the units of Figure 3-10 for the error are in meters secondary to large HA errors in the Vicon data. To aid with visual presentation the error bars were removed in Figure 3-10b. The MotionMonitor™ results were consistently better than the Vicon results. These results were unexpected.

Figure 3-10: Comparison of MotionMonitor™ Data and Vicon Data (fe: Flexion-Extension Motion; rot: Rotation Motion; L: Marble Left; R: Marble Right; F: Femur; T: Tibia)

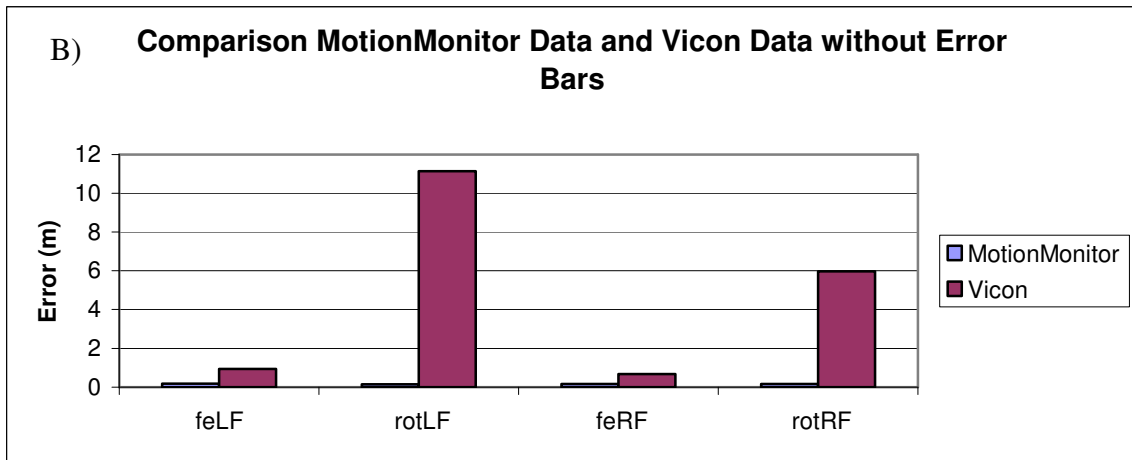
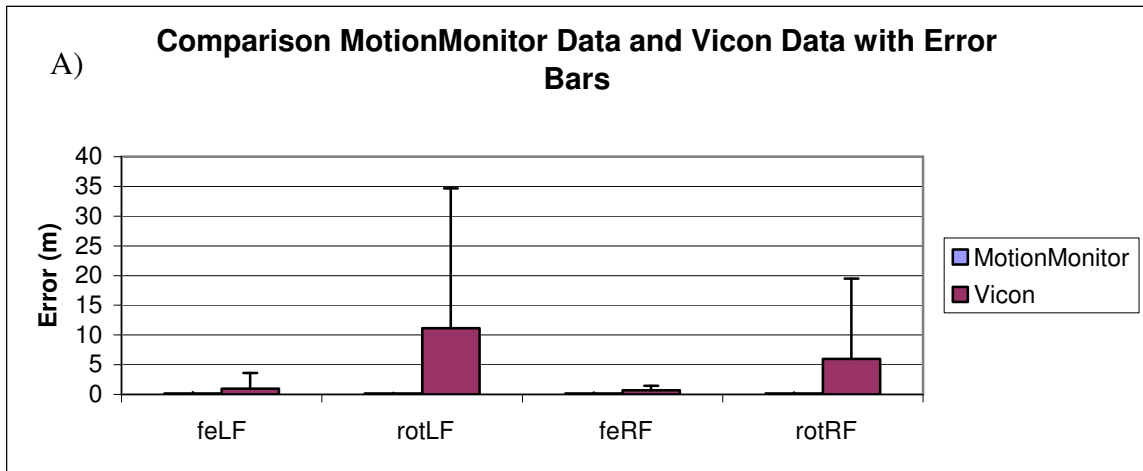


Table 3-6 compares the average percent difference of the distance between the points used in the HA calculation for both MotionMonitor™ and Vicon for both marble locations and both movements. The overall average percent difference for

MotionMonitor™ is 0.17% while the overall average percent difference for Vicon is 0.36%. From this table the percent difference with the Vicon data is about double that of MotionMonitor™ for the different marble location and different movements. This could explain in part why the MotionMonitor™ data gives much lower error values than Vicon.

Table 3-6: Average percent difference in the distance between points used to calculate the helical axis for MotionMonitor™ and Vicon data

	MotionMonitor™ Average Percent Difference	Vicon Average Percent Difference
Flexion-Extension Left	0.09%	0.21%
Internal-External Rotation Left	0.22%	0.51%
Flexion-Extension Right	0.31%	0.10%
Internal-External Rotation Right	0.40%	0.25%
Overall	0.17%	0.36%

3.2 Mathematical (Triangle) Model

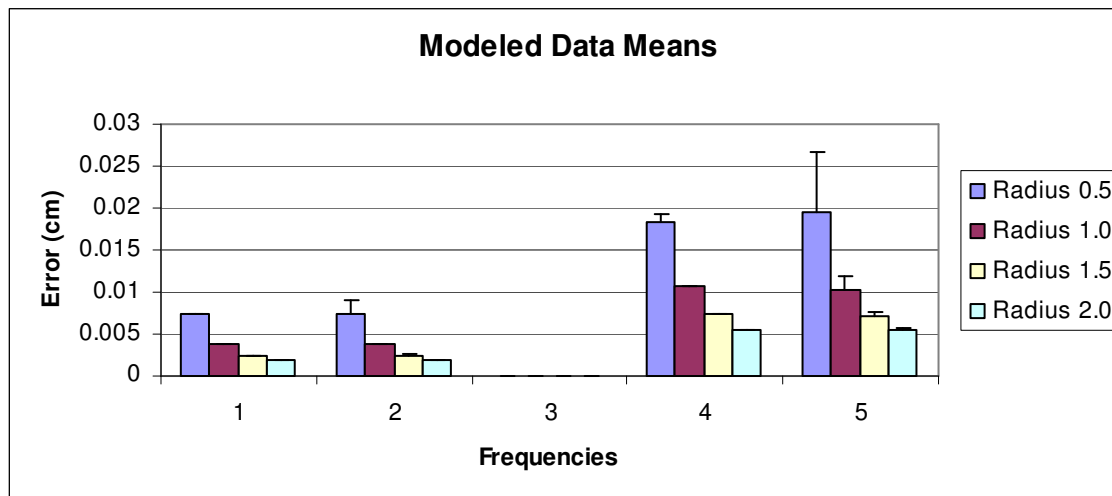
In the mathematical model, X, Y, Z data were created by rotation and oscillating an equilateral triangle in space around a central axis. Each vertex of the triangle followed a circular path on the XY plane defined by sine and cosine functions. The three vertices moved independently in the Z (vertical) direction by manipulating an additional sine function. The location of each vertex was projected onto the plane defined by their vertices and the triangle center for every degree of triangle motion during one full rotation. The result was 360 points of three dimensional data for each triangle vertex. The rigid body assumption was tested by computing and maintaining the geometric relationships of the vertices and center of the triangle. Both the size of the triangle and the frequency of vertical oscillations were manipulated. From the 3D movement of these three points, the orientation of the helical axis was calculated. The larger radius and lower frequency combination would be expected to produce smaller HA errors because

these combinations resulted in the greatest distance between position one and position two used to calculate the HA. According to Bull, the greater the distance between the positions being used to calculate the HA, the less errors in the calculations [9]. The distances between the HA and the center of the triangle were computed and reported in Table 3-7 and Figure 3-11 as means and standard deviation. A full presentation of the data for one trial with a radius of 1 cm and a Z value of 2 (4π) is found in Appendix C.

Table 3-7: Intersection error mean and standard deviation (cm) for each Z value and each radii combination

Z Value	R=0.5	R=1	R=1.5	R=2
1 (2π)	0.0074±0.0000	0.0037±0.0000	0.0025±0.0000	0.0019±0.0000
2 (4π)	0.0075±0.0016	0.0037±0.0002	0.0025±0.0001	0.0019±0.0000
3 (6π)	0.0000±0.0000	0.0000±0.0000	0.0000±0.0000	0.0000±0.0000
4 (8π)	0.0183±0.0011	0.0106±0.0002	0.0073±0.0001	0.0055±0.0000
5 (10π)	0.0195±0.0071	0.0103±0.0015	0.0071±0.0005	0.0055±0.0002

Figure 3-11: Intersection error mean and standard deviation for all triangle data



The sine wave frequency of 3 (or 6π) resulted in all corners of the triangle moving up and down at the same time. This gave the best results or the least error when comparing the computed helical axis to the known axis of rotation. This was an artificial

result however since this movement produced no difference in the Z position between the three vertices. Under this circumstance, the HA did not move. It was expected that the HA would have slight variations throughout the movement of the triangle. From Equation 7 in the Methods section, it is known that when there is no rotation about the HA, the calculated HA is non-unique and the radius vector of the HA is 0, ($\hat{s} = [0,0,0]$). In the calculations for the HA in the modeled data the triangle was centered at the origin (0, 0, 0). If the HA being calculated was a non-unique HA, the error would be 0 when the triangle is centered at the origin, because the HA would pass directly through this center point. To test whether the small error associated with a Z frequency of 3 is real or due to non-unique HA the origin of the triangle was changed but the radius of the triangle was kept constant with a value of 1 cm. These results are in Table 3-8.

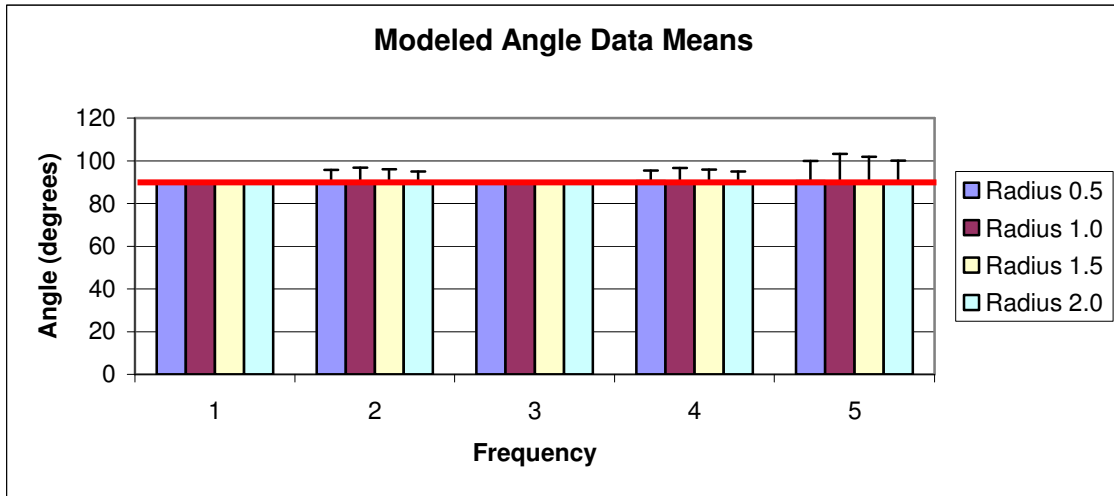
Table 3-8: Mean error for a radius of 1 cm and frequency of 3 at different origin

Origin	Mean ± Standard Deviation (cm)
(0,0,0)	0.0000±0.0000
(1,0,0)	0.0636±0.0306
(0,1,0)	0.0636±0.0306
(0,0,1)	0.0000±0.0000
(1,1,1)	2.4123±0.0346
(-1,-1,-1)	2.4123±0.0356
(-2,2,2)	8.1859±0.0180

In the mathematical model, another variable of interest was the angle the HA made with the plane of the triangle. Ideally, this angle would be 90° regardless of the orientation of the triangle as it moves through space. Figure 3-12 presents these averaged angle data with the bold red line indicating 90°. Also note that the error bars decline

slightly with larger radius triangles. This is because in the larger radius triangles, the distance between position one and position two is greater. This greater distance increases the accuracy of the HA calculations.

Figure 3-12: Relative helical axis angle mean and standard deviation.



The data from a trial with a radius of 1.0 cm and z value of $2(4\pi)$ was used to plot the triangle as it rotated around the circle and the HA intersecting it is seen in Figure 3-13.

Figure 3-13: Triangle and Helical Axis for one typical trial.

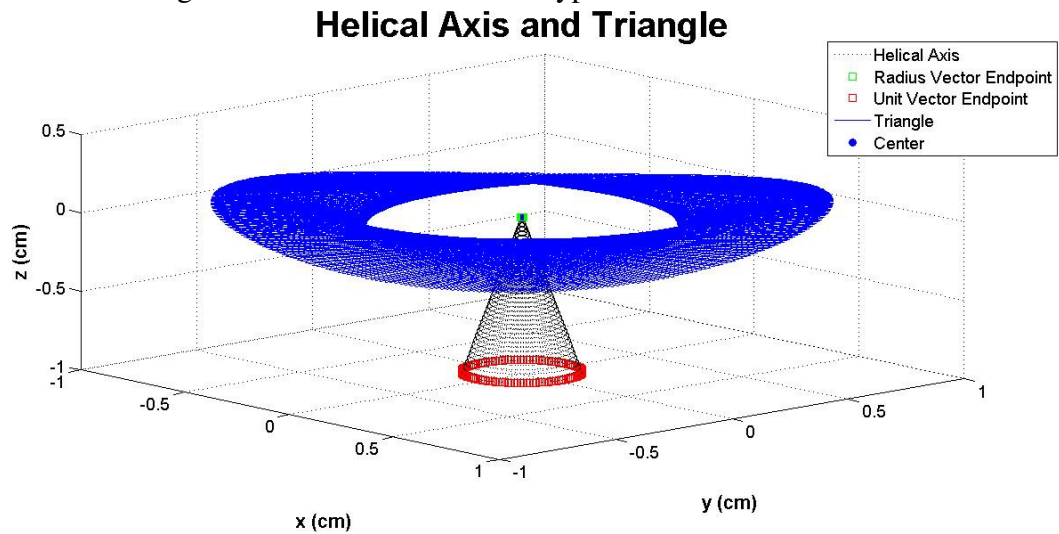


Figure 3-14 shows the first 50° of movement seen in Figure 3-13 to better visualize the triangle. To visualize the error in this trial, Figure 3-15 shows the center of the triangle and the HA intersection. Note that the units in this plot are 10^{-3} cm. The difference between the HA intersection and the known axis of rotation (center of the triangle) are really quite small.

Figure 3-14: First 50° of movements from Figure 3-13.

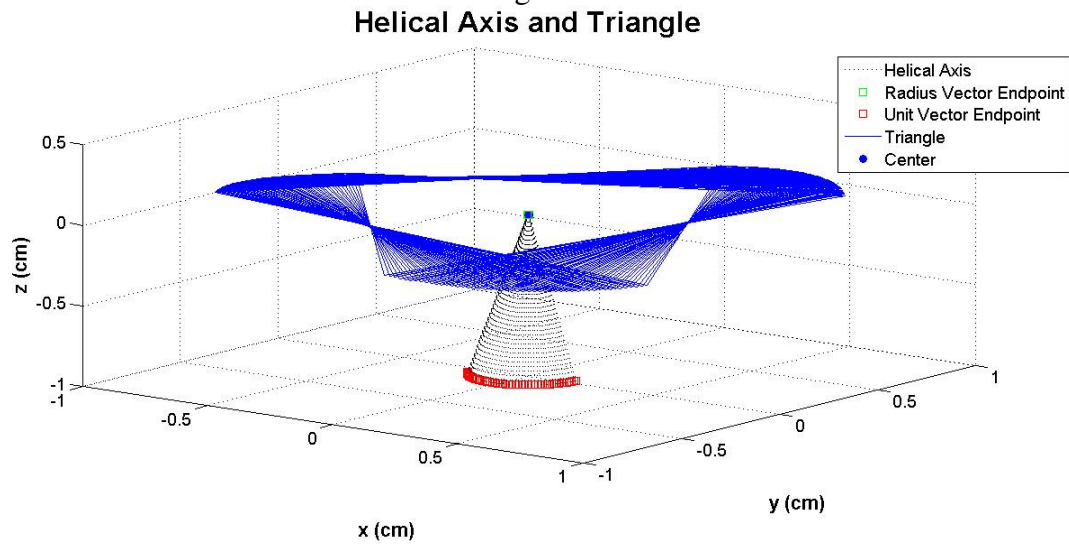
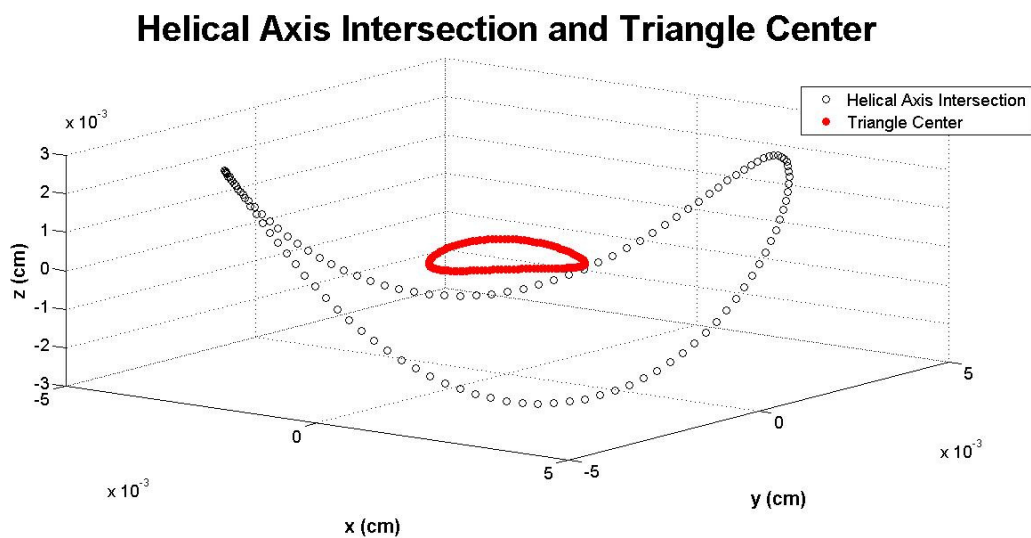


Figure 3-15: Plot of the Helical Axis Intersection and the Triangle Center for a typical trial.



To further test the numerical model, randomly distributed noise was added to assess the impact on helical axis error. This “noise” was created using the random number generator function in Microsoft Excel. A normal distribution was selected with a mean of 0 and standard deviation of 1. Figure 3-16 is a histogram of the noise distribution from a typical trial with noise amplitude of 1 cm.

Figure 3-16: Histogram of noise using a mean of 0 and standard deviation of 1.

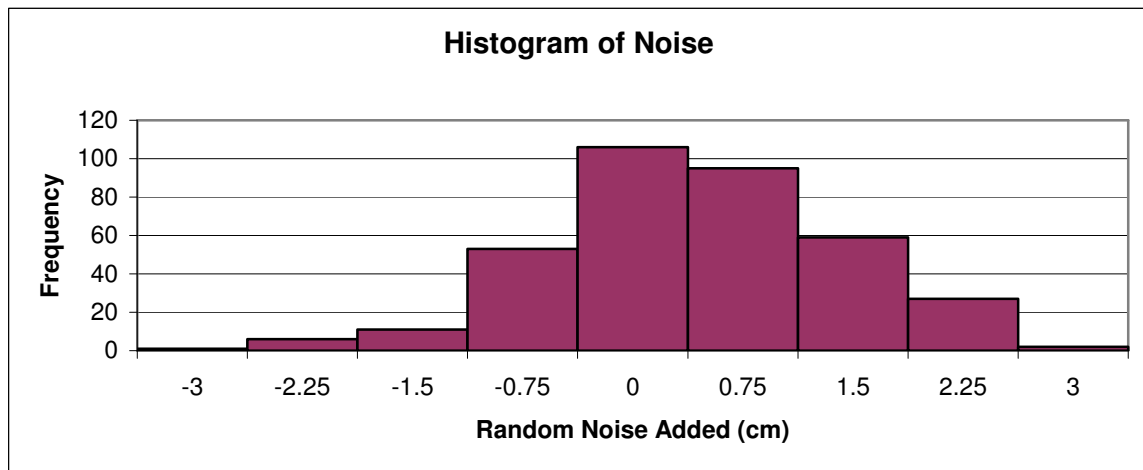
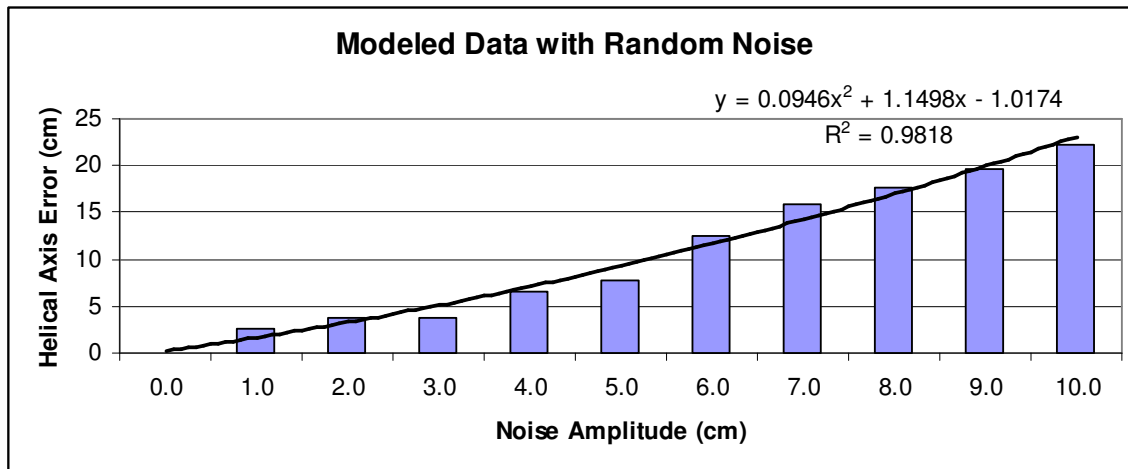


Table 3-9 has the HA error with and without noise. These results are also illustrated in Figure 3-17. Note the plot is fitted with a second order polynomial curve with an R-squared value of 0.9818.

Table 3-9: Helical axis error (cm) with and without the normally distributed noise added. The Noise was increased with each trial. The HA error data with noise increased non-linearly as noise levels increased.

Trial	Helical Axis Error without Noise (cm)	Noise (cm)			Helical Axis Error with Noise (cm)
	Mean ± StDev	Mean ± StDev	Max	Min	Mean ± StDev
control	0.002±0.000				0.002±0.000
1	0.002±0.0000	0.070±1.016	3.456	-2.922	2.668±3.571
2	0.002±0.0000	0.140±2.033	6.912	-5.844	3.850±6.871
3	0.002±0.0000	0.211±3.049	10.367	-8.766	3.772±5.554
4	0.002±0.0000	0.282±4.065	13.823	-11.688	6.517±12.579
5	0.002±0.0000	0.352±5.082	17.279	-14.609	7.683±15.931
6	0.002±0.0000	0.423±6.098	20.735	-17.531	12.442±33.337
7	0.002±0.0000	0.493±7.114	24.191	-20.453	15.961±55.934
8	0.002±0.0000	0.564±8.130	27.646	-23.375	17.694±45.360
9	0.002±0.0000	0.634±9.147	31.102	-26.297	19.679±50.033
10	0.002±0.0000	0.705±10.163	34.558	-29.219	22.298±55.117

Figure 3-17: Modeled data helical axis error with random noise at increasing amplitudes.



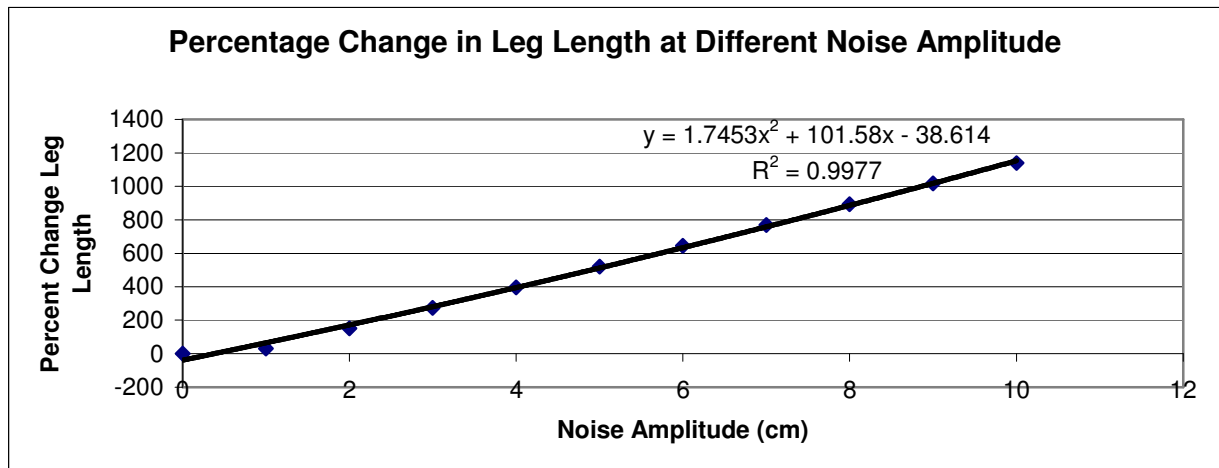
One final way to characterize the effects of noise is to examine the change in lengths of the triangle sides. These are presented as a percent change in Table 3-10. The triangle distortion increases dramatically with an increase in noise amplitude.

Table 3-10: Percent change in triangle leg lengths with increasing noise amplitude.

Amplitude (cm)	Percent Change
control	0
1	31.12
2	151.34
3	273.96
4	397.21
5	520.74
6	644.40
7	768.14
8	891.93
9	1015.75
10	1139.60

Figure 3-18 is a graphical representation of Table 3-9, with a second order polynomial trendline added with the equation.

Figure 3-18: Percentage change in leg length at different noise amplitude



3.3 Cylinder Model

The cylinder model was used to provide more controlled movement during kinematic data collection than the rigid body model described earlier. It was expected that these data would provide a more robust calculation of helical axis location. In this model, the recorded landmarks were from 6 geometrically equidistant points on a sphere

centered in the cylinder as the cylinder was rolled on a plane. The HA intersection was calculated as the axis passed through an imaginary plane centered in the sphere and perpendicular to the surface on which the cylinder was rolled, and tangential to the direction of movement. The distance between this point and the center of the sphere (intersection error) was determined. The angle of the HA with this imaginary plane was also determined (angular error).

Data were collected 3 times each day on 3 different days. These data were used to run an analysis of variance (ANOVA) assuming equal variance with a significant level of 0.05. The results between the days are presented in Table 3-11 and the results between the trials are presented in Table 3-12. It was expected that between-day comparison would show larger differences than between-trial data. This was not the case. This could be due to the fact that larger sample size (greater than 1000) tends to be more statistically similar [25]. In these ANOVAs, the day-to-day comparison has a sample size of 1200, while in the trial-to-trial comparison the sample size is only 400.

Table 3-11: ANOVA between the days shows the largest significant difference between data collection on days 1 and 2.

	Significance
Day 1 – Day 2	0.982
Day 1 – Day 3	0.470
Day 2 – Day 3	0.366

Table 3-12: ANOVA between trials. The majority of the differences between the trials are not significant.

Day	Trial	Significance
1	1-2	0.000
1	1-3	0.105
1	2-3	0.008
2	1-2	0.002
2	1-3	0.012
2	2-3	0.829
3	1-2	0.289
3	1-3	0.083
3	2-3	0.796

Figure 3-19 depicts mean and standard deviation the of intersection error.

Figure 3-19: Intersection error mean and standard deviation (the linear distance from where the helical axis intersected the center of the cylinder to the center of the sphere).

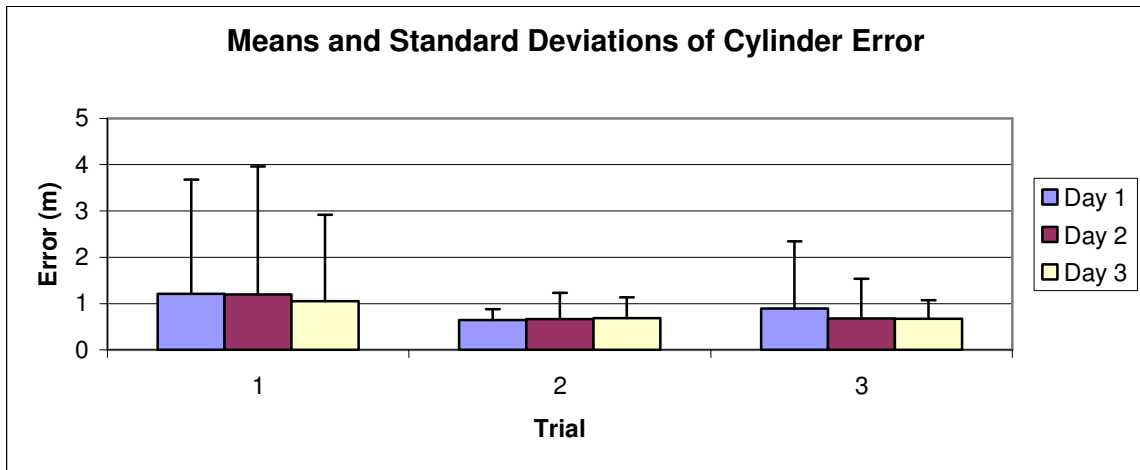
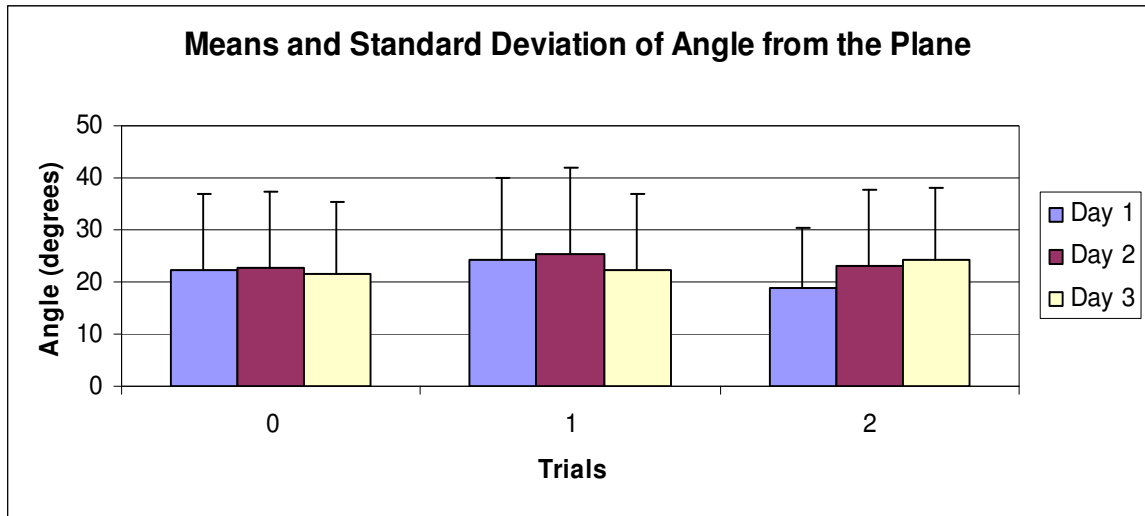


Figure 3-20 illustrates the mean and standard deviations for the angular error between the helical axis angle and a transverse cylinder plane. The angle of intersection for the instantaneous HA is 0°, thus the closer this angle of intersection is to 0° the closer the finite HA is to the instantaneous HA.

Figure 3-20: Mean and standard deviation for the angle the helical axis makes with the plane on which it is being rolled.



CHAPTER 4–Discussion

The purpose of this study was to design and test a helical axis algorithm and to see if it was plausible to use HA to determine where the tibial and femoral rotation intersects the meniscus during normal walking. The HA algorithm was tested through the use of three models: a multiplanar rigid body model, a mathematical (triangle) model, and a cylinder model. The HA method was selected since it allows non-anatomical referenced data points to be used to compute the HA orientation making it easily implemented in clinical data collection scenarios.

4.1 Multiplanar Rigid Body Model

The multiplanar rigid body model was used first to model the knee using two pieces of wood. One represented the femur and the other the tibia. The joint was simulated by a marble fixed in either a medial or lateral location. This provided a fixed axis of rotation during manipulation of the model that could be used as a reference in helical axis comparisons. During data collection, the tibia was held fixed and femur was moved in a multiplanar direction.

When calculating the helical axis for the femur, there was a high correlation (ICC 0.990) between the average HA location and the location of the actual axis of movement (center of the marble) across all 36 trials. When calculating the HA for the tibia, there

was a lower correlation (ICC 0.893) between these locations across all 36 trials. These correlations were performed on individual (n=9) and combined (n=36) data sets. The consistency of the correlation in four individual data sets suggests reliability in these results. The overall high correlation suggests a consistent representation of the HA. The lower overall correlation of the tibia data were expected since there was limited movement of the tibia when compared to the femur resulting in the computation of non-unique HA. The second ICC that was calculated, showed a high correlation between the average helical axis intersection and the axis of rotation (marble location), indicating that the helical axis intersection was near the marble location.

When comparing the change in the distance of the points used to determine the helical axis, there was no difference in the distance for the data from MotionMonitor™ and the data fitted with a cubic spline. Both of these sets of data had an average percent difference of 0.17%. Because there is no difference in this percent difference, the fact that MotionMonitor™ doesn't simultaneously collect the X, Y, and Z coordinates, the difference doesn't affect the position data. From this, it can be assumed that the larger errors of the instantaneous HA are not due to slewed data.

Using the equations in Figure 3-17 and Figure 3-18, one would expect with a percentage change of 0.17%, the average error would be 0.74 cm. This was not the case, for this marble location and movement; the average HA error was 1.41 cm. A plausible explanation for the larger than expected errors could be the velocity of the rigid body model was not constant while the velocity of the triangle was held constant throughout the movement. Although when comparing the velocities with the magnitude of the error,

there was no correlation, this wider range of errors could be due to the fact that the velocities are changing in the rigid body model and the velocity is constant in the mathematical modeled data.

Although the instantaneous helical axis location may be a more precise measure, the average HA is often used to represent the actual axis of rotation throughout the movement. According to Lenz et al. the instantaneous HA changes throughout the movement so the average HA is commonly used [16]. Using this average HA is advantageous because it is dependent of the motion and not affected by the variability associated with identifying anatomical landmarks.

There was a large transverse plane error when comparing the HA planar intercept with the center of the marble. The vertical errors (along Z direction) were minimized since the plane was defined as going through the center of the marble. The standard deviation for these femoral data was on the average of 20 cm. This results in an inability to differentiate between the actual axis of rotation and the HA intersection since the difference between left and right marble (axis) location was 4.8 cm. The large standard deviations in this model suggest limited usefulness in tracking movements in a human knee; however, a more refined model may help to determine the source of these errors. The sagittal plane movements (flexion-extension) had slightly larger errors than transverse plane movements (internal-external rotation). When viewing the errors temporally, the majority of the HA intersections were within 20 cm of each other.

To test the sources of these errors, kinematic data were collected using an optical system (Vicon) in addition to an electromagnetic system (MotionMonitor™).

MotionMonitor™ does not collect X, Y, Z data simultaneously. For a 100 Hz data collection rate, the transmitting antenna is actually operating at a rate of 400 Hz. It has to get through an X, Y, Z, and wait state within 1/100 of a second. This can result in errors in reporting the actual location of the sensor. Optical kinematic systems do not have this problem. In an effort to determine if this was a source of some of the HA calculation error, the experiments were repeated using an optical kinematic tracking system. These types of systems are often used as the gold standard. The Vicon system used in this second data collection has a linear spatial resolution of 1.04 mm and an angular resolution of 0.57 degrees. HA errors did not improve with the use of this system and can be explained in part by the average percentage change in the distance between the points used in the HA calculation. When using MotionMonitor™ the average percent difference in the distance between the points used to calculate the HA as 0.17% while the average percent difference in distance between the points when using Vicon was 0.36%.

One source of the error associated with the helical axis calculation for both MotionMonitor™ and Vicon data could be the number of landmarks used. According to Challis, as the number of landmarks was increased from the minimum needed to calculate the HA, three non-collinear landmarks, to eight landmarks, the size of the error was significantly reduced [26]. Increasing the number of landmarks from 8 to 36, in steps of one, resulted in a slight overall improvement, but there were fluctuations in the error size. Increasing the number of landmarks did not always increase the accuracy. Besides the number of landmarks, the distribution of the landmarks is also an important factor in the accuracy of the calculations [26]. The reason three landmarks were used instead of more

for this experiment protocol is due to the fact that the version of MotionMonitor™ used only allowed one sensor to be attached to each body segment. The landmarks are in reference to that sensor, increasing the number of landmarks would not necessarily decrease the error. For the purpose of consistency the same procedure was used in the Vicon experiment.

The only way to test the impact of instrumentation errors is to calculate the helical axis from mathematically modeled data. This would affectively eliminate any errors associated with data collection systems.

4.2 Mathematical (Triangle) Model

The multiplanar rigid body model showed large absolute errors between the helical axis intersection and the actual mechanical axis of rotation. Although the correlation between these locations was quite high, there was a large variance in the HA locations. It was not clear if instrumentation errors were contributing to these errors. In the mathematical model, a rigid body (with a known axis of rotation) was mathematically rotated in space. By using mathematically modeled data the algorithm used to compute the HA could be tested. Since the mathematical modeled data gave accurate results the errors in the rigid body model were due to computational or kinematic measurement errors.

The mathematically modeled data provide a method of testing the algorithm by allowing full control of both the location of the axis of rotation and the axis orientation. The mathematically modeled data showed smaller absolute errors, with the largest being 0.02 cm in intersection errors. These results suggest a robust algorithm that the

elimination of kinematic errors would improve the performance of the helical axis algorithm. It was assumed that the smaller errors would be associated with larger radii and smaller frequencies because in these data sets the distance between position 1 to position 2 used to calculate the HA is the greatest. It should be noted that as the radii increased the standard deviations decreased slightly. The decrease in standard deviations could be attributed to larger distances between the points used to calculate the HA. As has been noted by Bull, the closer together the points are, the more inaccurate the results; thus, there is less variation when the points are spaced further apart [9].

Some authors suggest filtering the kinematic data as a method of reducing these errors [20]. This would not work in a real-time data collection scenario, but it might still have clinical efficacy by providing patients with delayed feedback on their performance.

For the data used, the triangle acted like a rigid body without added noise due to the small amplitude in the Z direction. Increasing the amplitude in the Z direction distorts the shape of the triangle, but using a three dimensional solid body such as a tetrahedron could potentially reduce the errors further [17].

It was also expected that as the Z frequency increased, the errors would also increase because there were shorter steps between the data. According to Bull, as the steps between data tend toward zero, the closer the finite helical axis approaches the instantaneous HA, but at the same time the errors in the position data used to calculate the HA render the calculations more inaccurate [9]. A Z value of 3 (6π) for this example show an error of 0.000 cm and standard deviation of 0.000 cm when the triangle is centered at 0,0,0. When the center of the triangle is moved this error increases. The

reason for this increase is at this Z frequency the HA being computed is non-unique. In this situation the radius vector of the HA is [0,0,0]. Since the radius vector corresponds to the center of the triangle when centered at the origin, this intersection is not a true representation.

There is often noise associated with kinematic data collection. To illustrate this noise, a random number generator was used to create normally distributed random noise. This noise was added to the raw data. These data with noise were then used in the helical axis algorithm. The results showed that increased the amplitude of the noise from 0.0 to 10.0 cm in increments of 1.0 cm increased the errors in the HA intersection calculation. A second order polynomial was fitted to the data with an R-squared value of 0.9818.

The helical axis calculated from the mathematically modeled data was better than the data from the multiplanar rigid model because these data did not include errors associated with kinematic collection systems. Results of the modeled data show that the HA algorithm was calculating the HA as was expected which validates the HA algorithm. From this it is known that the errors from the rigid body model were not due to algorithm errors, thus the errors are either computational errors or kinematic collection errors.

4.3 Cylinder Model

The combination of the results from the two previous models suggests that the majority of the helical axis calculation error result from the kinematic data collection process. This final model was an effort to constrain the motion of a real world model (thereby improving the kinematics) and provide a more controlled HA location and orientation.

Kinematic data collection error heavily influenced these results in a negative manner. Since this was a more constrained kinematic experiment, the opportunity to test the reliability of these results was taken. These data showed that between the days there were no significant differences in data collected on multiple days. The results of the ANOVA between the trials on the same day were inconclusive. For day three there was no significant difference between the trials, but for the other two days this is not the case. From this, to conclude that between the trials are not significantly different, more tests would need to be run.

Since the experiment protocol was similar to the protocol in the Woltring article, the results were compared between the two experiments [18]. The Woltring article used this format and presented the standard deviations from the angle that the helical axis made with the plane. Woltring's experiment yielded a standard deviation of about 2.65° [18], while the standard deviations from the above experiment were about 12° . This could be due to the fact that Woltring's experiment collected data at 128 equidistant steps, so the angular rotation was about 2.8° [18], and this experiment had a step size of about 0.85° and has been noted in Bull that increasing the step size leads to decrease in error [9]. A comparison of Woltring's standard deviation and the standard deviations from this experiment can be found in Appendix F.

The multiplanar rigid body model most closely resembled the intended application of this algorithm. Unfortunately, it had the largest HA error because the errors associated with the kinematic collection systems. This model was also the most dependent on the experimenter. The mathematical model removed the errors associated

with kinematic data collection. The results of these modeled data validated the HA algorithm and illustrated the negative impact of kinematic noise. The cylinder model was used to see if kinematic data collection errors could be reduced by constraining the movement of the model. Kinematic errors were still too large to utilize the HA method. These results suggest that the HA method is feasible, but is heavily influenced by errors in the kinematic data.

4.4 Future Work

In Spoor [17], verification of Woltring's [22] error propagation formulas, there were five conditions that must be met to use these formulas to limit the error. These five initial conditions are: (1) at least 3 non-collinear points on a rigid body; (2) these points must be isotropically distributed to imply absence of preferential direction or planes in the marker distribution; (3) inaccuracies in the marker coordinates be unbiased, uncorrelated, and equal for all markers and all directions; (4) the helical axis calculation be statistically optimal, such as the method in Spoor [19]; and (5) errors must be relatively small [17]. Applying these principles to the data collection may reduce the errors in the calculations, making this method more optimal in the use of modeling the tibial and femoral rotation on the meniscus.

In order to reduce the magnitudes of the errors associated with the helical axis algorithm, smoothing splines could be used to filter the marker position prior to the calculations. The use of splines have been used by other investigators and seen to reduce the errors by up to two orders of magnitude [20]. As previously stated, studies have shown that increasing the number of landmarks to a degree decreases the errors in the

calculations [26]. All the systems currently used to determine IOR (Euler angles, Grood and Suntay joint coordinate system, and helical axis) have problems. A problem with HA as previously stated is that at small rotation such as micromotion between two bodies, HA are highly sensitive to marker position errors [27]. A common problem with Euler angles is it is dependent on the rotation sequence, and Grood and Suntay coordinate system is extremely sensitive to landmark location. It could be possible to integrate HA and Euler angles by using Euler angles to form the rotation matrix when the angular rotation is so small that it is considered as having no rotation [20,21]. In order to eliminate the errors from the experimenter in the multiplanar rigid body motion, it might be possible to use a robot to more precisely move the femur and the tibia to represent a more normal gait cycle.

Literature Cited

Literature Cited

1. Perry J. Gait Analysis: Normal and Pathological functions. Thorofare, NJ: SLACK Incorporated; 1992
2. Lipper LS. Clinical Kinesiology for Physical Therapist Assistants. Philadelphia, PA: F.A. Davis Company; 2000.
3. Schepesis AA, ed. , et al. Sports Medicine. Philadelphia, PA: Lippincott Williams & Wilkins, 2005.
4. Behnke RS. Kinetic Anatomy. New York: Human Kinetics, 2001.
5. Prentice WE and Onate JA. Rehabilitation of Knee Injuries. In Prentice WE, ed. Rehabilitation Techniques for Sports Medicine and Athletic Training. Boston, MA: McGraw Hill; 2004:526-578.
6. Ricklin P. Meniscus lesions practical problems of clinical diagnosis, arthrography and therapy. Stuttgart: G.Thieme., 1971.
7. Jones AO, Houang MTW, Low RS, Wood DG. Medial meniscus posterior root attachment injury and degeneration: MRI findings. Australasian Radiology. 2006;50:306-313.
8. Kaufman KR and An KN. Joint-articulating surface motion. In Bronzino JD, ed. The Biomedical Engineering Handbook. Boca Raton, FL: CRC P LLC;2000: 20-1-20-37.
9. Bull AMJ and Amis AA. Knee joint motion: description and measurements. Proceedings Institution of Mechanical Engineers Part H. 1998;212: 357-72.
10. Arnoczky SP. Gross and vascular anatomy of the meniscus and its role in meniscal healing, regeneration, and remodeling. In Mow VC, Arnoczky SP, and Jackson DW, eds. Knee Meniscus: Basic and Clinical Foundations. New York: Raven Press, Ltd; 1992:1-14.

11. Brindle T, Nyland J, Johnson DL. The Meniscus: Review of Basic Principles with Applications to Surgery and Rehabilitation. *Journal of Athletic Training*. 2001;36(2):160-169.
12. DiCarlo EF. Pathology of the Meniscus. In Mow VC, Arnoczky SP, and Jackson DW, eds. *Knee Meniscus: Basic and Clinical Foundations*. New York: Raven Press, Ltd; 1992:117-130.
13. Moglo KE and Shirzai-Adl A. Cruciate coupling and screw-home mechanism in passive knee joint during extension-flexion. *Journal of Biomechanics*. 2005;38:1075-1083.
14. Blankevoort LR, Huiskes R, and de Lange A. Helical Axes of Passive Knee Joint Motions. *Journal of Biomechanics*. 1990;28:1291-1229.
15. O'Reilly OM. The Dual Euler Basis: Constraints, Potentials, and Lagrange's Equations in Rigid Body Dynamics. *Journal of Applied Mechanics*. 2007;74:256-258.
16. Lenz NM, Mane A, Maletsky LP, Morton NA. The Effects of Femoral Fixed Body Coordinate System Definition on Knee Kinematic Description. *Journal of Biomechanical Engineering*. 2008;103:021014-1-021014-7.
17. Spoor CW. Explanation, verification and application of helical-axis error propagation formulas. *Human Movement Science*. 1984;3:96-117.
18. Woltring HJ and Huiskes R. A Statistically Motivated Approach to Instantaneous Helical Axis Estimation from Noise, Sampled Landmark Coordinates. In Winter DA, et al. eds. *Biomechanics IX-B*. Champaign, Illinois: Human Kinetics Publishers; 1985:274-279.
19. Spoor CW and Veldpaus FE. Rigid Body Motion Calculated from Spatial Coordinates of Markers. *Journal of Biomechanics*. 1980;12:391-393.
20. Johnson TS, Andriacchi TP, and Erdman AG. Sensitivity of finite helical axis parameters to temporally varying realistic motion utilizing an idealized knee model. *Proceedings of the Institution of Mechanical Engineers Part H: Journal of Engineering in Medicine*. 2004;218:89-100.
21. Duck TR, Ferreira LM, King GJM, Johnson JAS. Assessment of screw displacement axis accuracy and repeatability for joint kinematic description using an electromagnetic tracking device. *Journal of Biomechanics*. 2004;37:163-167.

22. Woltring HJ, Huiskes R, de Lange A. Finite centroid and helical axis estimation from noisy landmark measurements in the study of human joint kinematics. *Journal of Biomechanics*. 1985;18:379-389.
23. Abramowitz M and Stegun IA eds. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover Publications, Inc. 1970.
24. Aissaoui R, Mecheri HD, and Guise JA. Validation of Four Major Algorithms for Estimating the Instantaneous Helical Axis with Miniature Triaxial Gyroscope. *Proceedings of the 26th Annual International Conference of the IEEE EMBS*. 2004:2442-2445.
25. Gamst G, Meyers LS, and Guarino AJ. *Analysis of Variance Designs: A Conceptual and Computational Approach with SPSS and SAS*. New York: Cambridge University Press. 2008
26. Challis JH. A Procedure for Determining Rigid Body Transformation Parameters. *Journal of Biomechanics*. 1995;28:733-737.
27. Yuan X, Ryd L, Blankevoort L. Error propagation for relative motion determined from marker positions. *Journal of Biomechanics*. 1997;30:989-992.

APPENDIX A

Programs used to determine the helical axis and helical axis intersection using data collected with MotionMonitor™ kinematic collection system.

```
% HA1.m
% Helical Axis from Motion Monitor Data

% These calculations are taken from Spoons 1980
clear
close all
clf

file_num=input('What file number is this? ');
mar_loc=input('Which location is the marble in? ', 's');
mov=input('What is the movement being performed? ', 's');
filename=['Sheet' num2str(file_num)];
row_num=file_num+2;
data=xlsread('MM Splined Data', filename, 'a1:ak500');

flag1=0;%Graphing the calculating helical axis
flag2=0;%Graphing the intersection
flag3=0;%Graphing individual helical axis components
fig=1;

frame=data(:,1);

%-----
%Landmark Data to Use
%-----
txl0=data(:,2); tyl0=data(:,3); tzl0=data(:,4);
txl1=data(:,5); tyl1=data(:,6); tzl1=data(:,7);
txl2=data(:,8); tyl2=data(:,9); tzl2=data(:,10);
%X,Y,Z individual data
tx=[txl0 txl1 txl2];%All the X landmark data
ty=[tyl0 tyl1 tyl2];%All the Y landmark data
tz=[tzl0 tzl1 tzl2];%All the Z landmark data
%Average of the X,Y,Z landmark data for each point in time
tavgx=mean(tx,2);
tavgy=mean(ty,2);
tavgz=mean(tz,2);
taverage=[tavgx tavgy tavgz];
%X,Y,Z Data of the Landmark
```

```

t10=[tx10 ty10 tz10];
t11=[tx11 ty11 tz11];
t12=[tx12 ty12 tz12];
%-----
%Landmark Data to Use
%-----
sx10=data(:,11); sy10=data(:,12); sz10=data(:,13);
sx11=data(:,14); sy11=data(:,15); sz11=data(:,16);
sx12=data(:,17); sy12=data(:,18); sz12=data(:,19);
%X,Y,Z individual Data
sx=[sx10 sx11 sx12];%All the X landmark data
sy=[sy10 sy11 sy12];%All the Y landmark data
sz=[sz10 sz11 sz12];%All the Z landmark data
%Average of the X,Y,Z landmark data for each point in time
savgx=mean(sx,2);
savgy=mean(sy,2);
savgz=mean(sz,2);
saverage=[savgx savgy savgz];
%X,Y,Z Data of the Landmark
s10=[sx10 sy10 sz10];
s11=[sx11 sy11 sz11];
s12=[sx12 sy12 sz12];
%-----
%Points to determine the plane
%-----
p0x=data(:,20); p0y=data(:,21); p0z=data(:,22);
p1x=data(:,23); p1y=data(:,24); p1z=data(:,25);
p2x=data(:,26); p2y=data(:,27); p2z=data(:,28);
p3x=data(:,29); p3y=data(:,30); p3z=data(:,31);
%-----
%Marble Location
%-----
Lx=data(:,32); Ly=data(:,33); Lz=data(:,34);
Rx=data(:,35); Ry=data(:,36); Rz=data(:,37);
%Frequency
freq=100;
frate=1/freq;
%
I=eye(3);
len=length(frame);
%Counters for both shank and thigh
c=1;
tem=1;
tcount=1;
td=1;
sem=1;
scount=1;
sd=1;
for ii=1:len-1
    a=ii;
    p=ii+1;
    %-----
    %Ensure that the landmark points aren't collinear
    %-----

```

```

%Thigh
tacol=iscollinear(tl0(a,:), tl1(a,:), tl2(a,:));
tpcol=iscollinear(tl0(p,:), tl1(p,:), tl2(p,:));
if tacol==1
    msg=['*** tacol (' num2str(a) ') the points are collinear'];
    disp(msg);
    continue
elseif tpcol==1
    msg=['*** tpcol (' num2str(p) ') the points are collinear'];
    disp(msg);
    continue
end
%Shank
sacol=iscollinear(sl0(a,:), sl1(a,:), sl2(a,:));
spcol=iscollinear(sl0(p,:), sl1(p,:), sl2(p,:));
if sacol==1
    msg=['*** sacol (' num2str(a) ') the points are collinear'];
    disp(msg);
    continue
elseif spcol==1
    msg=['*** spcol (' num2str(p) ') the points are collinear'];
    disp(msg);
    continue
end
%-----
%Make a 3x3 Matrix that will be manipulated to become the rotation
%matrix used in the rest of the calculations.  These steps make up
%equation (5) in Spoors paper
%-----
%Thigh Rotation Matrix
tm0=tl0(p,:)'*tl0(a,:);
tm1=tl1(p,:)'*tl1(a,:);
tm2=tl2(p,:)'*tl2(a,:);
tsumm=(tm0+tm1+tm2)/3;
tmavg=taverage(p,:)'*taverage(a,:);
tM=tsumm-tmavg;
%Function sommer comes from the subroutine MATPP3 by H.J. Sommer.
[tR, tS1]=sommer(tM);
RMV('tR',tR,a);
%Shank Rotation Matrix
sm0=sl0(p,:)'*sl0(a,:);
sm1=sl1(p,:)'*sl1(a,:);
sm2=sl2(p,:)'*sl2(a,:);
ssumm=(sm0+sm1+sm2)/3;
smavg=saverage(p,:)'*saverage(a,:);
sM=ssumm-smavg;
%Function sommer comes from the subroutine MATPP3 by H.J. Sommer.
[sR, sS1]=sommer(sM);
RMV('sR',sR,a);
%-----
%Translation Vector from equation (10) in Spoors
%-----
%Thigh
tv=taverage(p,:)'-tR*taverage(a,:)';

```



```

%Shank
sv=saverage(p,:)'-sR*saverage(a,:)';
%-----
%Solving for angle of rotation about the helical axis (phi)
%-----
%Thigh
%Spoors Eq 32
tsinphi=0.5*sqrt((tR(3,2)-tR(2,3))^2+(tR(1,3)-tR(3,1))^2+...
    (tR(2,1)-tR(1,2))^2);
%Spoors Eq 34
tcosphi=0.5*(trace(tR)-1);
%If statements from Spoors; different methods are preferred a
%different times because the arcsine function
if tsinphi<=1/2*sqrt(2)
    tphi=asin(tsinphi);
    if tcosphi<0
        tphi=pi-tphi;
    end
else
    tphi=acos(tcosphi);
end
%Shank
%Spoors Eq 32
ssinphi=0.5*sqrt((sR(3,2)-sR(2,3))^2+(sR(1,3)-sR(3,1))^2+...
    (sR(2,1)-sR(1,2))^2);
%Spoors Eq 34
scosphi=0.5*(trace(sR)-1);
%If statements from Spoors; different methods are preferred a
%different because the arcsine function
if ssinphi<=1/2*sqrt(2)
    sphi=asin(ssinphi);
    if scosphi<0
        sphi=pi-sphi;
    end
else
    sphi=acos(scosphi);
end
%-----
%Solving for Unit vector (n)
%-----
%Thigh
if tphi<=3/4*pi && tphi~=0
    %Spoors 31
    tn=(.5*[tR(3,2)-tR(2,3); tR(1,3)-tR(3,1); tR(2,1)-tR(1,2)])/...
        sin(tphi);
else
    %Spoors 35 and 36
    tb=0.5*(tR+tR')-cos(tphi)*I;
    tblen=[sqrt(tb(:,1)'*tb(:,1));
        sqrt(tb(:,2)'*tb(:,2));
        sqrt(tb(:,3)'*tb(:,3))];
    [tBlen tIn]=max(tblen);
    tn=tb(:,tIn)/tBlen;
    if sin(tphi)<0

```

```

        tn=-tn;
    end
end
%Shank
if sphi<=3/4*pi && sphi~=0
    %Spoors 31
    sn=(.5*[sR(3,2)-sR(2,3); sR(1,3)-sR(3,1); sR(2,1)-sR(1,2)])/...
        sin(sphi);
else
    %Spoors 35 and 36
    sb=0.5*(sR+sR')-cos(sphi)*I;
    sbLen=[sqrt(sb(:,1)'*sb(:,1));
           sqrt(sb(:,2)'*sb(:,2));
           sqrt(sb(:,3)'*sb(:,3))];
    [sBlen sIn]=max(sbLen);
    sn=sb(:,sIn)/sBlen;
    if sin(sphi)<0
        sn=-sn;
    end
end
end
%-----
%Solving for Translation Component (t)
%-----
%Thigh
%Spoors 37
tt=tn'*tv;
%Shank
%Spoors 37
st=sn'*sv;
%-----
%Solving for Radius Vector (s)
%-----
%Thigh
tdenom=1-cos(tphi); %denominator portion of radius vector
%If it equals 0 making it a really small number so it will give a
%actual value and not NaN
if tdenom==0
    tdenom=0.0000001;
    tcount=tcount+1;
end
%Spoors 38
ts=-.5*cross(tn,cross(tn,tv))+sin(tphi)/(2*(tdenom))*cross(tn,tv);
%Shank
sdenom=1-cos(sphi); %denominator portion of radius vector
%If it equals 0 making it a really small number so it will give a
%actual value and not NaN
if sdenom==0
    sdenom=0.0000001;
    scount=scount+1;
end
%Spoors 38
ss=-.5*cross(sn,cross(sn,sv))+sin(sphi)/(2*(sdenom))*cross(sn,sv);
%-----
%If Phi==0 no defined helical axis and therefore n,s, and t aren't

```

```

%unique, these equations are used to give non-unique helical axis
%values
%-----
%Thigh
if tphi==0 && tv~= [0;0;0]
    tt1=tv'*tv;
    tt=sqrt(tt1);
    tn=(1./tt)*tv;
    ts=[0; 0; 0];
    tem=tem+1;
end
%Shank
if sphl==0 && sv~= [0;0;0]
    st1=tv'*tv;
    st=sqrt(st1);
    sn=(1./st)*sv;
    ss=[0; 0; 0];
    sem=sem+1;
end
%-----
%Saving Data
%-----
%Thigh
tN(c,1)=tn(1,1); tN(c,2)=tn(2,1); tN(c,3)=tn(3,1);
tPHI(c,1)=tphi;
tS(c,1)=ts(1,1); tS(c,2)=ts(2,1); tS(c,3)=ts(3,1);
tT(c,1)=tt;
tV(c,1)=tv(1,1); tV(c,2)=tv(2,1); tV(c,3)=tv(3,1);
%Shank
sN(c,1)=sn(1,1); sN(c,2)=sn(2,1); sN(c,3)=sn(3,1);
sPHI(c,1)=sphl;
sS(c,1)=ss(1,1); sS(c,2)=ss(2,1); sS(c,3)=ss(3,1);
sT(c,1)=st;
sV(c,1)=sv(1,1); sV(c,2)=sv(2,1); sV(c,3)=sv(3,1);
%-----
%Checks, ensure the unit vectors are unit vectors and the unit
%vector and radius vector are orthogonal
%-----
%Thigh
tnpn=tn'*tn;
tnps=tn'*ts;
tUnit(c,1)=tnpn;
tOrth(c,1)=tnps;
%Shank
snpn=sn'*sn;
snps=sn'*ss;
sUnit(c,1)=snpn;
sOrth(c,1)=snps;
%-----
%Used in the graph of the helical axis
%-----
if flag1==1
    %Thigh
    tNS=vertcat(tn',ts');

```

```

%Shank
sNS=vertcat(sn',ss');
figure(fig)
hold on
plot3(tNS(:,1), tNS(:,2), tNS(:,3), 'b')
scatter3(ts(1), ts(2), ts(3), 'g')
scatter3(tn(1), tn(2), tn(3), 'r')
plot3(sNS(:,1), sNS(:,2), sNS(:,3), ':k')
scatter3(ss(1), ss(2), ss(3), 'sg')
scatter3(sn(1), sn(2), sn(3), 'sr')
grid on
xlabel('x')
ylabel('y')
zlabel('z')
view(37,28)
end
if mov=='RT'
%-----
%Plane to see where the HA intersects is the tibial plateau
%To get this plane take three of the four corners of the
%top of the shank
%-----
pal0x=(p0x(a)+p0x(p))/2; pal0y=(p0y(a)+p0y(p))/2;
pal0z=(p0z(a)+p0z(p))/2;
pal1x=(p1x(a)+p1x(p))/2; pal1y=(p1y(a)+p1y(p))/2;
pal1z=(p1z(a)+p1z(p))/2;
pal2x=(p2x(a)+p2x(p))/2; pal2y=(p2y(a)+p2y(p))/2;
pal2z=(p2z(a)+p2z(p))/2;
pal0=[pal0x pal0y pal0z]; pal1=[pal1x pal1y pal1z];
pal2=[pal2x pal2y pal2z];
pA10(c,:)=pal0; pA11(c,:)=pal1; pA12(c,:)=pal2;
P1=pal1-pal0;
P2=pal2-pal0;
P3=cross(P1,P2);
%-----
%Check for HA being parallel to plane want to see where
%intersects
%-----
%Thigh
tVect=tn-ts;
tp3dot(c,1)=dot(P3,tVect);
if tp3dot==0
    msg=['**** ' ii ' Vector is parallel to the plane'];
    disp(msg);
    continue
end
%Shank
sVect=sn-ss;
sp3dot(c,1)=dot(P3,sVect);
if sp3dot==0
    msg=['**** ' ii ' Vector is parallel to the plane'];
    disp(msg);
    continue
end
end

```

```

%-----
%Where the intersection occurs in the plane of interest
%-----
%Thigh
tThigh(ii)=(P3(1)*(pal0x-tn(1))+P3(2)*(pal0y-tn(2))+P3(3)*...
    (pal0z-tn(3)))/(P3(1)*tVect(1)+P3(2)*tVect(2)+...
    P3(3)*tVect(3));
ttx(ii)=tn(1)+tVect(1)*tThigh(ii);
tty(ii)=tn(2)+tVect(2)*tThigh(ii);
ttz(ii)=tn(3)+tVect(3)*tThigh(ii);
ttv=[ttx(ii) tty(ii) ttz(ii)];
%Shank
sThigh(ii)=(P3(1)*(pal0x-sn(1))+P3(2)*(pal0y-sn(2))+P3(3)*...
    (pal0z-sn(3)))/(P3(1)*sVect(1)+P3(2)*sVect(2)+...
    P3(3)*sVect(3));
stx(ii)=sn(1)+sVect(1)*sThigh(ii);
sty(ii)=sn(2)+sVect(2)*sThigh(ii);
stz(ii)=sn(3)+sVect(3)*sThigh(ii);
stv=[stx(ii) sty(ii) stz(ii)];
elseif mov=='FE'
%-----
%Plane to see where the HA intersects is the tibial plateau
%To get this plane take three of the four corners of the
%top of the shank
%-----
pal0x=(p0x(a)+p0x(p))/2; pal0y=(p0y(a)+p0y(p))/2;
pal0z=(p0z(a)+p0z(p))/2;
pal1x=(p1x(a)+p1x(p))/2; pal1y=(p1y(a)+p1y(p))/2;
pal1z=(p1z(a)+p1z(p))/2;
pal2x=(p2x(a)+p2x(p))/2; pal2y=(p2y(a)+p2y(p))/2;
pal2z=(p2z(a)+p2z(p))/2;
if mar_loc=='L'
    aLx(a,1)=(Lx(a)+Lx(p))/2;
    aLy(a,1)=(Ly(a)+Ly(p))/2;
    aLz(a,1)=(Lz(a)+Lz(p))/2;
elseif mar_loc=='R'
    aLx(a,1)=(Rx(a)+Rx(p))/2;
    aLy(a,1)=(Ry(a)+Ry(p))/2;
    aLz(a,1)=(Rz(a)+Rz(p))/2;
end
pal0=[pal0x pal0y pal0z]; pal1=[pal1x pal1y pal1z];
pal2=[pal2x pal2y pal2z];
pA10(c,:)=pal0; pA11(c,:)=pal1; pA12(c,:)=pal2;
p1=pal1-pal0;
p2=pal2-pal0;
p3=cross(p1,p2);
A=p3(1); B=p3(2); C=p3(3);
asb=-1; asc=1; asa=(-B*asb-C*asc)/A;
a1=asa; b1=asb; c1=asc; P3=[a1 b1 c1];
d1=-a1*aLx(a)-b1*aLy(a)-c1*aLz(a);
Plane1=[A B C]; Plane2=[a1 b1 c1];
perp_check(c,1)=dot(Plane1,Plane2);

```

```

%-----
%Check for HA being parallel to plane want to see where
%intersects
%-----
%Thigh
tVect=tn-ts;
tp3dot(c,1)=dot(P3,tVect);
if tp3dot==0
    msg=['**** ' ii ' Vector is parallel to the plane'];
    disp(msg);
    continue
end
%Shank
sVect=sn-ss;
sp3dot(c,1)=dot(P3,sVect);
if sp3dot==0
    msg=['**** ' ii ' Vector is parallel to the plane'];
    disp(msg);
    continue
end

%-----
%Where the intersection occurs in the plane of interest
%-----
%Thigh
tul=(a1*tn(1)+b1*tn(2)+c1*tn(3)+d1)/(a1*(tn(1)-ts(1))+...
    b1*(tn(2)-ts(2))+c1*(tn(3)-ts(3)));
tP_int=tn+tul*(ts-tn);
tPInt(c,1)=tP_int(1); tPInt(c,2)=tP_int(2);
tPInt(c,3)=tP_int(3);
ttx(ii)=tP_int(1);
tty(ii)=tP_int(2);
ttz(ii)=tP_int(3);
ttv=[ttx(ii) tty(ii) ttz(ii)];
%Shank
sul=(a1*sn(1)+b1*sn(2)+c1*sn(3)+d1)/(a1*(sn(1)-ss(1))+...
    b1*(sn(2)-ss(2))+c1*(sn(3)-ss(3)));
sP_int=sn+sul*(ss-sn);
sPInt(c,1)=sP_int(1); sPInt(c,2)=sP_int(2);
sPInt(c,3)=sP_int(3);
stx(ii)=sP_int(1);
sty(ii)=sP_int(2);
stz(ii)=sP_int(3);
stv=[stx(ii) sty(ii) stz(ii)];
end
%-----

if flag2==1
    figure(2)
    hold on
    scatter3(ttx(ii), tty(ii),ttz(ii), 'b')
    scatter3(stx(ii), sty(ii),stz(ii), 'sr')
    if mar_loc=='L'

```

```

        scatter3(Lx(ii), Ly(ii), Lz(ii), 'FaceColor', 'k')
elseif mar_loc=='R'
        scatter3(Rx(ii), Ry(ii), Rz(ii), 'FaceColor', 'k')
end
grid on
xlabel('x')
ylabel('y')
zlabel('z')
if ii==len-1
    fig=fig+2;
end
view(37,28)
end
if mar_loc=='L'
    aLx(a,1)=(Lx(a)+Lx(p))/2;
    aLy(a,1)=(Ly(a)+Ly(p))/2;
    aLz(a,1)=(Lz(a)+Lz(p))/2;
    Tmar(c,1)=sqrt((ttx(ii)-aLx(a,1))^2+(tty(ii)-aLy(a,1))^2+...
        (ttz(ii)-aLz(a,1))^2);
    Smar(c,1)=sqrt((stx(ii)-aLx(a,1))^2+(sty(ii)-aLy(a,1))^2+...
        (stz(ii)-aLz(a,1))^2);
elseif mar_loc=='R'
    aLx(a,1)=(Rx(a)+Rx(p))/2;
    aLy(a,1)=(Ry(a)+Ry(p))/2;
    aLz(a,1)=(Rz(a)+Rz(p))/2;
    Tmar(c,1)=sqrt((ttx(ii)-aLx(a,1))^2+(tty(ii)-aLy(a,1))^2+...
        (ttz(ii)-aLz(a,1))^2);
    Smar(c,1)=sqrt((stx(ii)-aLx(a,1))^2+(sty(ii)-aLy(a,1))^2+...
        (stz(ii)-aLz(a,1))^2);
end
%-----
%Length of triangle sides used to calculate the helical axis to
%ensure that the sides are consistent throughout the movement
%-----
ts1=sqrt((txl0(a)-txl1(a))^2+(tyl0(a)-tyl1(a))^2+...
    (tzl0(a)-tzl1(a))^2);
ts2=sqrt((txl1(a)-txl2(a))^2+(tyl1(a)-tyl2(a))^2+...
    (tzl1(a)-tzl2(a))^2);
ts3=sqrt((txl2(a)-txl0(a))^2+(tyl2(a)-tyl0(a))^2+...
    (tzl2(a)-tzl0(a))^2);
tside(c,1)=ts1; tside(c,2)=ts2; tside(c,3)=ts3;
ss1=sqrt((sxl0(a)-sxl1(a))^2+(syl0(a)-syl1(a))^2+...
    (szl0(a)-szl1(a))^2);
ss2=sqrt((sxl1(a)-sxl2(a))^2+(syl1(a)-syl2(a))^2+...
    (szl1(a)-szl2(a))^2);
ss3=sqrt((sxl2(a)-sxl0(a))^2+(syl2(a)-syl0(a))^2+...
    (szl2(a)-szl0(a))^2);
sside(c,1)=ss1; sside(c,2)=ss2; sside(c,3)=ss3;
c=c+1;
end
tPHI_deg=tPHI*180/pi;
sPHI_deg=sPHI*180/pi;
if flag3==1
    figure(fig)

```

```

hold on
plot(tPHI_deg, 'r')
plot(sPHI_deg, 'b')
title('Phi')
legend('Thigh', 'Shank')
fig=fig+1;
%
figure(fig)
hold on
scatter3(tS(:,1), tS(:,2), tS(:,3), 'r')
scatter3(sS(:,1), sS(:,2), sS(:,3), 'b')
xlabel('x')
ylabel('y')
zlabel('z')
grid on
title('Radius Vector')
legend('Thigh', 'Shank')
fig=fig+1;

figure(fig)
hold on
scatter3(tN(:,1), tN(:,2), tN(:,3), 'r')
scatter3(sN(:,1), sN(:,2), sN(:,3), 'b')
xlabel('x')
ylabel('y')
zlabel('z')
grid on
title('Unit Vector')
legend('Thigh', 'Shank')
fig=fig+1;

figure(fig)
hold on
plot(tT, 'r')
plot(sT, 'b')
title('Translation Component')
legend('Thigh', 'Shank')
fig=fig+1;
%
figure(fig)
hold on
scatter3(tV(:,1), tV(:,2), tV(:,3), 'r')
scatter3(sV(:,1), sV(:,2), sV(:,3), 'b')
xlabel('x')
ylabel('y')
zlabel('z')
title('Translation Vector')
legend('Thigh', 'Shank')
grid on
fig=fig+1;
end

Tmar_disp=[mean(Tmar) min(Tmar) max(Tmar)];

```



```

Smar_disp=[mean(Smar) min(Smar) max(Smar)];
Sheet_name=['Sheet' num2str(file_num)];
alen=1:length(Tmar);
title1={'Frame' 'Tmar' 'ttx' 'tty' 'ttz' 'Smar' 'stx' 'sty' ...
        'stz' 'mLx' 'mLy' 'mLz'};
sav_data1=[alen' Tmar ttx' tty' ttz' Smar stx' sty' stz' aLx aLy aLz];
xlswrite('RBM_int Data', title1, Sheet_name, 'a1')
xlswrite('RBM_int Data', sav_data1, Sheet_name, 'a2')
title2={'Tmar' 'Tside1' 'Tside2' 'Tside3' 'Smar' 'Sside1' 'Sside2'...
        'Sside3'};
sav_data2=[Tmar tside Smar sside];
xlswrite('RBM_Tri', title2, Sheet_name, 'a1')
xlswrite('RBM_Tri', sav_data2, Sheet_name, 'a2')
temj=1;
Tmar_out=0;
for iii=1:length(Tmar)
    if Tmar(iii)>mean(Tmar)+std(Tmar)
        Tmar_out(iii,1)=iii;
    else
        Tmar(temj,1)=Tmar(iii);
        Tside(temj,:)=tside(iii,:);
        Ttx(temj,1)=ttx(iii);
        Tty(temj,1)=tty(iii);
        Ttz(temj,1)=ttz(iii);
        tLx(temj,1)=aLx(iii);
        tLy(temj,1)=aLy(iii);
        tLz(temj,1)=aLz(iii);
        Tmar_inc(temj,1)=iii;
        temj=temj+1;
    end
end
semj=1;
Smar_out=0;
for iii=1:length(Smar)
    if Smar(iii)>mean(Smar)+std(Smar)
        Smar_out(iii,1)=iii;
    else
        Smar(semj,1)=Smar(iii);
        Sside(semj,:)=sside(iii,:);
        Stx(semj,1)=stx(iii);
        Sty(semj,1)=sty(iii);
        Stz(semj,1)=stz(iii);
        sLx(semj,1)=aLx(iii);
        sLy(semj,1)=aLy(iii);
        sLz(semj,1)=aLz(iii);
        Smar_inc(semj,1)=iii;
        semj=semj+1;
    end
end
[mean(TMar) std(TMar)]
tlen=1:length(TMar);
slen=1:length(SMar);
title3={'T_Frame' 'tFrames_Inc' 'Tmar' 'ttx' 'tty' 'ttz' 'tLx' 'tLy'...
        'tLz' 'T_Outliers' 'S_Frame' 'sFrames_Inc' 'Smar' 'stx' 'sty' ...

```

```

    'stz' 'sLx' 'sLy' 'sLz' 'S_Outliers'}];
sav_data3=[tlen' Tmar_inc TMar Ttx Tty Ttz tLx tLy tLz];
sav_data3_1=Tmar_out;
sav_data3_2=[slen' Smar_inc SMar Stx Sty Stz sLx sLy sLz];
sav_data3_3=Smar_out;
xlswrite('RBM_int Outliers', title3, Sheet_name, 'a1')
xlswrite('RBM_int Outliers', sav_data3, Sheet_name, 'a2')
xlswrite('RBM_int Outliers', sav_data3_1, Sheet_name, 'j2')
xlswrite('RBM_int Outliers', sav_data3_2, Sheet_name, 'k2')
xlswrite('RBM_int Outliers', sav_data3_3, Sheet_name, 't2')
title4={'T_Frame' 'Tmar' 'Tsidel' 'Tside2' 'Tside3' 'T_Outliers' ...
    'S_Frame' 'Smar' 'Ssidel' 'Sside2' 'Sside3' 'S_Outliers'}];
sav_data4=[tlen' TMar Tside];
sav_data4_1=Tmar_out;
sav_data4_2=[slen' SMar Sside];
sav_data4_3=Smar_out;
xlswrite('RBM_Tri Outliers', title4, Sheet_name, 'a1')
xlswrite('RBM_Tri Outliers', sav_data4, Sheet_name, 'a2')
xlswrite('RBM_Tri Outliers', sav_data4_1, Sheet_name, 'f2')
xlswrite('RBM_Tri Outliers', sav_data4_2, Sheet_name, 'g2')
xlswrite('RBM_Tri Outliers', sav_data4_3, Sheet_name, 'l2')
Row_num=['A' num2str(row_num)];
title5={'Average TMar' 'Average Ttx' 'Average Tty' 'Average Ttz' ...
    'std TMar' 'std Ttx' 'std Tty' 'std Ttz'}];
Tavg_ICC=[mean(TMar) mean(Ttx) mean(Tty) mean(Ttz) std(TMar) ...
    std(Ttx) std(Tty) std(Ttz)];
xlswrite('RBM_Average', title5, 'Femur Int', 'a1')
xlswrite('RBM_Average', Tavg_ICC, 'Femur Int', Row_num)
title6={'Average SMar' 'Average Stx' 'Average Sty' 'Average Stz'...
    'std TMar' 'std Ttx' 'std Tty' 'std Ttz'}];
Savg_ICC=[mean(SMar) mean(Stx) mean(Sty) mean(Stz) std(SMar) ...
    std(Stx) std(Sty) std(Stz)];
xlswrite('RBM_Average', title6, 'Tibia Int', 'a1')
xlswrite('RBM_Average', Savg_ICC, 'Tibia Int', Row_num)

```

```

%iscollinear.m
function[outdata]=iscollinear(a, b, c)

%Three points are collinear all three lie on a straight line. So the sum
%of the distance between points a and b and the distance between points
%b and c equals the sum between points a and c.

ab=sqrt((a(1)-b(1))^2+(a(2)-b(2))^2+(a(3)-b(3))^2);
bc=sqrt((b(1)-c(1))^2+(b(2)-c(2))^2+(b(3)-c(3))^2);
ac=sqrt((a(1)-c(1))^2+(a(2)-c(2))^2+(a(3)-c(3))^2);
if ab+bc==ac||bc+ac==bc||ac+bc==ab
    outdata=1;
else
    outdata=0;
end

```

```

%RMV.m
%-----
%Rotation matrix validation (RMV) -- Compute the sqrt of the sum of the
%squared values in each row and column. Return the product of all these
%values multiplied together. The answer should be '1' for a true
%rotation matrix.
%-----
function[outdata] = RMV(name,indata,indx)

[m,n]=size(indata);

%compute column values first
cnt=1;
for i=1:m
    sum = 0;
    for j=1:n
        sum = sum + indata(i,j)^2;
    end
    result(cnt) = sqrt(sum);
    cnt = cnt+1;
end

%compute row values next
for j=1:m
    sum = 0;
    for i=1:n
        sum = sum + indata(i,j)^2;
    end
    result(cnt) = sqrt(sum);
    cnt = cnt+1;
end

%check all results
outdata = 1;
cnt=cnt-1;
for i=1:cnt
    outdata = outdata * result(i);
end

if cast(outdata,'single') ~= 1
    msg=['*** ' name '(' num2str(indx) ') is not a rotation matrix.'];
    disp(msg);
end

```

```

%summer.m
function [R, S]=summer(G)

%Positive polar decomposition of 3x3 matrix G=R*S
%Forcing positive orthonormal rotation matrix

%Inputs
%G=3x3 general matrix

%Outputs
%R=3x3 positive orthonormal matrix
%S=3x3 symmetric matrix

EPS=1.0e-5;

%Cofactors and Determinant of g
COG(1,1)=G(2,2)*G(3,3)-G(2,3)*G(3,2);
COG(2,1)=G(1,3)*G(3,2)-G(1,2)*G(3,3);
COG(3,1)=G(1,2)*G(2,3)-G(1,3)*G(2,2);
COG(1,2)=G(2,3)*G(3,1)-G(2,1)*G(3,3);
COG(2,2)=G(1,1)*G(3,3)-G(1,3)*G(3,1);
COG(3,2)=G(1,3)*G(2,1)-G(1,1)*G(2,3);
COG(1,3)=G(2,1)*G(3,2)-G(2,2)*G(3,1);
COG(2,3)=G(1,2)*G(3,1)-G(1,1)*G(3,2);
COG(3,3)=G(1,1)*G(2,2)-G(1,2)*G(2,1);
G3=G(1,1)*COG(1,1)+G(2,1)*COG(2,1)+G(3,1)*COG(3,1);

P=G'*G;
%Adjoint of P
ADP(1,1)=P(2,2)*P(3,3)-P(2,3)*P(3,2);
ADP(2,2)=P(1,1)*P(3,3)-P(1,3)*P(3,1);
ADP(3,3)=P(1,1)*P(2,2)-P(1,2)*P(2,1);

%G invariants
G1SQ=P(1,1)+P(2,2)+P(3,3);
G1=sqrt(G1SQ);
G2SQ=ADP(1,1)+ADP(2,2)+ADP(3,3);
G2=sqrt(G2SQ);

%Initialize Iteration
H1=G2/G1SQ;
H2=G3*G1/G2SQ;
X=1.0;
Y=1.0;
DX=1.0;
DY=1.0;

%Iteration Loop
while abs(DX/X)>EPS || abs(DY/Y)>EPS
    DEN=2.0*(X*Y-H1*H2);
    RES1=1.0-X*X+2.0*H1*Y;
    RES2=1.0-Y*Y+2.0*H2*X;

```

```

DX=(Y*RES1+H1*RES2)/DEN;
DY=(H2*RES1+X*RES2)/DEN;
X=X+DX;
Y=Y+DY;
end

%Beta Invariants
BETA1=X*G1;
BETA2=Y*G2;

%Invert(G'*G+BETA2*I)
P(1,1)=P(1,1)+BETA2;
P(2,2)=P(2,2)+BETA2;
P(3,3)=P(3,3)+BETA2;
PBI(1,1)=P(2,2)*P(3,3)-P(2,3)*P(3,2);
PBI(1,2)=P(1,3)*P(3,2)-P(1,2)*P(3,3);
PBI(1,3)=P(1,2)*P(2,3)-P(1,3)*P(2,2);
PBI(2,1)=P(2,3)*P(3,1)-P(2,1)*P(3,3);
PBI(2,2)=P(1,1)*P(3,3)-P(1,3)*P(3,1);
PBI(2,3)=P(1,3)*P(2,1)-P(1,1)*P(2,3);
PBI(3,1)=P(2,1)*P(3,2)-P(2,2)*P(3,1);
PBI(3,2)=P(1,2)*P(3,1)-P(1,1)*P(3,2);
PBI(3,3)=P(1,1)*P(2,2)-P(1,2)*P(2,1);
DETPBI=P(1,1)*PBI(1,1)+P(2,1)*PBI(1,2)+P(3,1)*PBI(1,3);

for i=1:3

R(i,1)=((COG(i,1)+BETA1*G(i,1))*PBI(1,1)+(COG(i,2)+BETA1*G(i,2))*PBI(2,1)
)+(COG(i,3)+BETA1*G(i,3))*PBI(3,1))/DETPBI;

R(i,2)=((COG(i,1)+BETA1*G(i,1))*PBI(1,2)+(COG(i,2)+BETA1*G(i,2))*PBI(2,2)
)+(COG(i,3)+BETA1*G(i,3))*PBI(3,2))/DETPBI;

R(i,3)=((COG(i,1)+BETA1*G(i,1))*PBI(1,3)+(COG(i,2)+BETA1*G(i,2))*PBI(2,3)
)+(COG(i,3)+BETA1*G(i,3))*PBI(3,3))/DETPBI;
end
%
for i=1:3
S(i,1)=R(1,i)*G(1,1)+R(2,i)*G(2,1)+R(3,i)*G(3,1);
S(i,2)=R(1,i)*G(1,2)+R(2,i)*G(2,2)+R(3,i)*G(3,2);
S(i,3)=R(1,i)*G(1,3)+R(2,i)*G(2,3)+R(3,i)*G(3,3);
end

%
```

```

% HA1_spline.m
% Spline the position data used to calculate the helical axis

clear
close all
clf

file_num=input('What file number is this? ');
file_name=['C:\Users\Emily\Documents\Thesis\102108\102508_'...
    num2str(file_num)];
%mar_loc=input('Which location is the marble in? ', 's');
%filename=input('What is the trial filename with extension? ', 's');
data=xlsread(file_name, 'Sheet1', 'a10:aq509');

flag1=0;%Graphing the calculating helical axis
flag2=1;%Graphing the intersection
flag3=0;%Graphing individual helical axis components
fig=1;

frame=data(:,1);

%-----
%Landmark Data to Use
%-----
tX=data(:,2); tY=data(:,3); tZ=data(:,4);
txl0=data(:,5); tyl0=data(:,6); tzl0=data(:,7);
txl1=data(:,8); tyl1=data(:,9); tzl1=data(:,10);
txl2=data(:,11); tyl2=data(:,12); tzl2=data(:,13);
%X,Y,Z individual data
tx=[txl0 txl1 txl2];%All the X landmark data
ty=[tyl0 tyl1 tyl2];%All the Y landmark data
tz=[tzl0 tzl1 tzl2];%All the Z landmark data
%Average of the X,Y,Z landmark data for each point in time
tavgx=mean(tx,2);
tavgy=mean(ty,2);
tavgz=mean(tz,2);
taverage=[tavgx tavgy tavgz];
%X,Y,Z Data of the Landmark
t10=[txl0 tyl0 tzl0];
t11=[txl1 tyl1 tzl1];
t12=[txl2 tyl2 tzl2];
%-----
%Landmark Data to Use
%-----
sx10=data(:,17); sy10=data(:,18); sz10=data(:,19);
sx11=data(:,20); sy11=data(:,21); sz11=data(:,22);
sx12=data(:,23); sy12=data(:,24); sz12=data(:,25);
%X,Y,Z individual Data
sx=[sx10 sx11 sx12];%All the X landmark data
sy=[sy10 sy11 sy12];%All the Y landmark data
sz=[sz10 sz11 sz12];%All the Z landmark data

```

```

%Average of the X,Y,Z landmark data for each point in time
savgx=mean(sx,2);
savgy=mean(sy,2);
savgz=mean(sz,2);
saverage=[savgx savgy savgz];
%X,Y,Z Data of the Landmark
s10=[sx10 sy10 sz10];
s11=[sx11 sy11 sz11];
s12=[sx12 sy12 sz12];
%-----
%Points to determine the plane
%-----
p0x=data(:,26); p0y=data(:,27); p0z=data(:,28);
p1x=data(:,29); p1y=data(:,30); p1z=data(:,31);
p2x=data(:,32); p2y=data(:,33); p2z=data(:,34);
p3x=data(:,35); p3y=data(:,36); p3z=data(:,37);
%-----
%Marble Location
%-----
Lx=data(:,38); Ly=data(:,39); Lz=data(:,40);
Rx=data(:,41); Ry=data(:,42); Rz=data(:,43);
%Frequency
freq=100;
frate=1/freq;
%
I=eye(3);
len=length(frame);
%Counters for both shank and thigh
c=1;
tem=1;
tcount=1;
td=1;
sem=1;
scount=1;
sd=1;
x=1:len;y=1:len;z=1:len;
x1=1.5:1:len+1; y1=1.25:1:len+1; z1=1:1:len;
ppx0t=spline(x,tx10,x1);
ppy0t=spline(y,ty10,y1);
ppz0t=spline(z,tz10,z1);
ppx1t=spline(x,tx11,x1);
ppy1t=spline(y,ty11,y1);
ppz1t=spline(z,tz11,z1);
ppx2t=spline(x,tx12,x1);
ppy2t=spline(y,ty12,y1);
ppz2t=spline(z,tz12,z1);
ppx0s=spline(x,sx10,x1);
ppy0s=spline(y,sy10,y1);
ppz0s=spline(z,sz10,z1);
ppx1s=spline(x,sx11,x1);
ppy1s=spline(y,sy11,y1);
ppz1s=spline(z,sz11,z1);
ppx2s=spline(x,sx12,x1);
ppy2s=spline(y,sy12,y1);

```



```

ppz2s=spline(z, szl2, z1);
ppx0=spline(x, p0x, x1);
ppy0=spline(y, p0y, y1);
ppz0=spline(z, p0z, z1);
ppx1=spline(x, p1x, x1);
ppy1=spline(y, p1y, y1);
ppz1=spline(z, p1z, z1);
ppx2=spline(x, p2x, x1);
ppy2=spline(y, p2y, y1);
ppz2=spline(z, p2z, z1);
ppx3=spline(x, p3x, x1);
ppy3=spline(y, p3y, y1);
ppz3=spline(z, p3z, z1);
pplx=spline(x, Lx, x1);
pply=spline(y, Ly, y1);
pplz=spline(z, Lz, z1);
pprx=spline(x, Rx, x1);
ppry=spline(y, Ry, y1);
pprz=spline(z, Rz, z1);
pTX=spline(x, tX, x1);
pTY=spline(y, tY, y1);
pTZ=spline(z, tZ, z1);

save_data=[ppx0t' ppy0t' ppz0t' ppx1t' ppy1t' ppz1t' ppx2t' ppy2t' ...
           ppz2t' ppx0s' ppy0s' ppz0s' ppx1s' ppy1s' ppz1s' ppx2s' ppy2s' ...
           ppz2s' ppx0' ppy0' ppz0' ppx1' ppy1' ppz1' ppx2' ppy2' ppz2' ...
           ppx3' ppy3' ppz3' pplx' pply' pplz' pprx' ppry' pprz'];

Sheet_name=['Sheet' num2str(file_num)];
xlswrite('MM Splined Data', save_data, Sheet_name, 'a1')

```

APPENDIX B

Programs used to determine helical axis and helical axis intersection using data collected
with Vicon kinematic collection system

```
% HA2.m
% Helical Axis from Vicon Data
% These calculations are taken from Spoors 1980, "Rigid Body Motion
% Calculated from Spatial Co-ordinates of Markers"

clear
close all
clf

file_num=input('What file number is this? ');
mar_loc=input('Which location is the marble in? ', 's');
mov=input('What is the movement? ', 's');
filename=['C:\Users\Emily\Documents\Thesis\102808\Oct280' ...
    num2str(file_num)];
data=xlsread(filename, 'Sheet1', 'a6:af2877');

flag1=0;%Graphing the calculating helical axis
flag2=0;%Graphing the intersection of HA and marble location
flag3=0;%Graphing calculated and MM components of HA
fig=1;

frame=data(:,1);

%-----
%Landmark Data to Use
%-----
txl0=data(:,12); tyl0=data(:,13); tzl0=data(:,14);
txl1=data(:,15); tyl1=data(:,16); tzl1=data(:,17);
txl2=data(:,18); tyl2=data(:,19); tzl2=data(:,20);
%X,Y,Z individual data
tx=[txl0 txl1 txl2];%All the X landmark data
ty=[tyl0 tyl1 tyl2];%All the Y landmark data
tz=[tzl0 tzl1 tzl2];%All the Z landmark data
%Average of the X,Y,Z landmark data for each point in time
tavgx=mean(tx,2);
tavgy=mean(ty,2);
tavgz=mean(tz,2);
```

```

taverage=[tavgx tavgy tavgz];
%X,Y,Z Data of the Landmark
t10=[tx10 ty10 tz10];
t11=[tx11 ty11 tz11];
t12=[tx12 ty12 tz12];
%-----
%Landmark Data to Use
%-----
sx10=data(:,3); sy10=data(:,4); sz10=data(:,5);
sx11=data(:,6); sy11=data(:,7); sz11=data(:,8);
sx12=data(:,9); sy12=data(:,10); sz12=data(:,11);
%X,Y,Z individual Data
sx=[sx10 sx11 sx12];%All the X landmark data
sy=[sy10 sy11 sy12];%All the Y landmark data
sz=[sz10 sz11 sz12];%All the Z landmark data
%Average of the X,Y,Z landmark data for each point in time
savgx=mean(sx,2);
savgy=mean(sy,2);
savgz=mean(sz,2);
saverage=[savgx savgy savgz];
%X,Y,Z Data of the Landmark
s10=[sx10 sy10 sz10];
s11=[sx11 sy11 sz11];
s12=[sx12 sy12 sz12];
%-----
%Points to determine the plane
%-----
p0x=data(:,21); p0y=data(:,22); p0z=data(:,23);
p1x=data(:,24); p1y=data(:,25); p1z=data(:,26);
p2x=data(:,27); p2y=data(:,28); p2z=data(:,29);
p3x=data(:,30); p3y=data(:,31); p3z=data(:,32);
%Frequency
freq=100;
frate=1/freq;

I=eye(3);
len=length(frame);
%Counters for both shank and thigh
c=1;
tem=1;
tcount=1;
td=1;
sem=1;
scount=1;
sd=1;
for ii=1:len-1
    a=ii;
    p=ii+1;
    %-----
    %Ensure that the landmark points aren't collinear
    %-----
    %Thigh
    tacol=iscollinear(t10(a,:), t11(a,:), t12(a,:));

```

```

tpcol=iscollinear(tl0(p,:), tl1(p,:), tl2(p,:));
if tacol==1
    msg=['*** tacol (' num2str(a) ') the points are collinear'];
    disp(msg);
    continue
elseif tpcol==1
    msg=['*** tpcol (' num2str(p) ') the points are collinear'];
    disp(msg);
    continue
end
%Shank
sacol=iscollinear(sl0(a,:), sl1(a,:), sl2(a,:));
spcol=iscollinear(sl0(p,:), sl1(p,:), sl2(p,:));
if sacol==1
    msg=['*** sacol (' num2str(a) ') the points are collinear'];
    disp(msg);
    continue
elseif spcol==1
    msg=['*** spcol (' num2str(p) ') is not a rotation matrix.'];
    disp(msg);
    continue
end
%-----
%Make a 3x3 Matrix that will be manipulated to become the rotation
%matrix used in the rest of the calculations.  These steps make up
%equation (5) in Spoors paper
%-----
%Thigh Rotation Matrix
tm0=tl0(p,:)'*tl0(a,:);
tm1=tl1(p,:)'*tl1(a,:);
tm2=tl2(p,:)'*tl2(a,:);
tsumm=(tm0+tm1+tm2)/3;
tmavg=taverage(p,:)'*taverage(a,:);
tM=tsumm-tmavg;
%Function sommer comes from the subroutine MATPP3 by H.J. Sommer.
[tR, tS1]=sommer(tM);
RMV('tR',tR,a);
%Shank Rotation Matrix
sm0=sl0(p,:)'*sl0(a,:);
sm1=sl1(p,:)'*sl1(a,:);
sm2=sl2(p,:)'*sl2(a,:);
ssumm=(sm0+sm1+sm2)/3;
smavg=saverage(p,:)'*saverage(a,:);
sM=ssumm-smavg;
%Function sommer comes from the subroutine MATPP3 by H.J. Sommer.
[sR, sS1]=sommer(sM);
RMV('sR',sR,a);
%-----
%Translation Vector from equation (10) in Spoors
%-----
%Thigh
tv=taverage(p,:)'-tR*taverage(a,:)';
%Shank
sv=saverage(p,:)'-sR*saverage(a,:)';

```

```

%-----
%Solving for angle of rotation about the helical axis (phi)
%-----
%Thigh
%Spoors Eq 32
tsinphi=0.5*sqrt((tR(3,2)-tR(2,3))^2+(tR(1,3)-tR(3,1))^2+...
    (tR(2,1)-tR(1,2))^2);
%Spoors Eq 34
tcosphi=0.5*(trace(tR)-1);
%If statements from Spoors; different methods are preferred a
%different times because the arcsine function
if tsinphi<=1/2*sqrt(2)
    tphi=asin(tsinphi);
    if tcosphi<0
        tphi=pi-tphi;
    end
else
    tphi=acos(tcosphi);
end
%Shank
%Spoors Eq 32
ssinphi=0.5*sqrt((sR(3,2)-sR(2,3))^2+(sR(1,3)-sR(3,1))^2+...
    (sR(2,1)-sR(1,2))^2);
%Spoors Eq 34
scosphi=0.5*(trace(sR)-1);
%If statements from Spoors; different methods are preferred a
%different times because the arcsine function
if ssinphi<=1/2*sqrt(2)
    sphi=asin(ssinphi);
    if scosphi<0
        sphi=pi-sphi;
    end
else
    sphi=acos(scosp);
end
%-----
%Solving for Unit vector (n)
%-----
%Thigh
if tphi<=3/4*pi && tphi~=0
    %Spoors 31
    tn=(.5*[tR(3,2)-tR(2,3); tR(1,3)-tR(3,1); tR(2,1)-tR(1,2)])/...
        sin(tphi);
else
    %Spoors 35 and 36
    tb=0.5*(tR+tR')-cos(tphi)*I;
    tblen=[sqrt(tb(:,1)'*tb(:,1));
        sqrt(tb(:,2)'*tb(:,2));
        sqrt(tb(:,3)'*tb(:,3))];
    [tBlen tIn]=max(tblen);
    tn=tb(:,tIn)/tBlen;
    if sin(tphi)<0
        tn=-tn;
    end
end

```

```

end
%Shank
if sphi<=3/4*pi && sphi~=0
    %Spoors 31
    sn=(.5*[sR(3,2)-sR(2,3); sR(1,3)-sR(3,1); sR(2,1)-sR(1,2)])/...
        sin(sphi);
else
    %Spoors 35 and 36
    sb=0.5*(sR+sR')-cos(sphi)*I;
    sblen=[sqrt(sb(:,1)'*sb(:,1));
           sqrt(sb(:,2)'*sb(:,2));
           sqrt(sb(:,3)'*sb(:,3))];
    [sBlen sIn]=max(sblen);
    sn=sb(:,sIn)/sBlen;
    if sin(sphi)<0
        sn=-sn;
    end
end
end
%-----
%Solving for Translation Component (t)
%-----
%Thigh
%Spoors 37
tt=tn'*tv;
%Shank
%Spoors 37
st=sn'*sv;
%-----
%Solving for Radius Vector (s)
%-----
%Thigh
tdenom=1-cos(tphi); %denominator portion of radius vector
%If it equals 0 making it a really small number so it will give a
%actual value and not NaN
if tdenom==0
    tdenom=0.0000001;
    tcount=tcount+1;
end
%Spoors 38
ts=-.5*cross(tn,cross(tn,tv))+sin(tphi)/(2*(tdenom))*cross(tn,tv);
%Shank
sdenom=1-cos(sphi); %denominator portion of radius vector
%If it equals 0 making it a really small number so it will give a
%actual value and not NaN
if sdenom==0
    sdenom=0.0000001;
    scount=scount+1;
end
%Spoors 38
ss=-.5*cross(sn,cross(sn,sv))+sin(sphi)/(2*(sdenom))*cross(sn,sv);
%-----
%If Phi==0 no defined helical axis and therefore n,s, and t aren't
%unique, these equations are used to give non-unique helical axis
%values

```

```

%-----
%Thigh
if tphi==0 && tv~= [0;0;0]
    tt1=tv'*tv;
    tt=sqrt(tt1);
    tn=(1./tt)*tv;
    ts=[0; 0; 0];
    tem=tem+1;
end
%Shank
if sphl==0 && sv~= [0;0;0]
    st1=sv'*sv;
    st=sqrt(st1);
    sn=(1./st)*sv;
    ss=[0; 0; 0];
    sem=sem+1;
end
%-----
%Saving Data
%-----
%Thigh
tN(c,1)=tn(1,1); tN(c,2)=tn(2,1); tN(c,3)=tn(3,1);
tPHI(c,1)=tphi;
tS(c,1)=ts(1,1); tS(c,2)=ts(2,1); tS(c,3)=ts(3,1);
tT(c,1)=tt;
tV(c,1)=tv(1,1); tV(c,2)=tv(2,1); tV(c,3)=tv(3,1);
%Shank
sN(c,1)=sn(1,1); sN(c,2)=sn(2,1); sN(c,3)=sn(3,1);
sPHI(c,1)=sphl;
sS(c,1)=ss(1,1); sS(c,2)=ss(2,1); sS(c,3)=ss(3,1);
sT(c,1)=st;
sV(c,1)=sv(1,1); sV(c,2)=sv(2,1); sV(c,3)=sv(3,1);
%-----
%Checks, ensure the unit vectors are unit vectors and the unit
%vector and radius vector are orthogonal
%-----
%Thigh
tnpn=tn'*tn;
tnps=tn'*ts;
tUnit(c,1)=tnpn;
tOrth(c,1)=tnps;
%Shank
snpn=sn'*sn;
snps=sn'*ss;
sUnit(c,1)=snpn;
sOrth(c,1)=snps;
%-----
%Used in the graph of the helical axis
%-----
if flag1==1
    %Thigh
    tNS=vertcat(tn',ts');
    %Shank
    sNS=vertcat(sn',ss');

```

```

figure(fig)
hold on
plot3(tNS(:,1), tNS(:,2), tNS(:,3), 'b')
scatter3(ts(1), ts(2), ts(3), 'g')
scatter3(tn(1), tn(2), tn(3), 'r')
plot3(sNS(:,1), sNS(:,2), sNS(:,3), ':k')
scatter3(ss(1), ss(2), ss(3), 'sg')
scatter3(sn(1), sn(2), sn(3), 'sr')
grid on
xlabel('x')
ylabel('y')
zlabel('z')
view(37,28)
end

if mov=='RT'
%-----
%Plane to see where the HA intersects is the tibial plateau
%To get this plane take three of the four corners of the
%top of the shank
%-----
pal0x=(p0x(a)+p0x(p))/2; pal0y=(p0y(a)+p0y(p))/2;
pal0z=(p0z(a)+p0z(p))/2;
pal1x=(p1x(a)+p1x(p))/2; pal1y=(p1y(a)+p1y(p))/2;
pal1z=(p1z(a)+p1z(p))/2;
pal2x=(p2x(a)+p2x(p))/2; pal2y=(p2y(a)+p2y(p))/2;
pal2z=(p2z(a)+p2z(p))/2;
pal3x=(p3x(a)+p3x(p))/2; pal3y=(p3y(a)+p3y(p))/2;
pal3z=(p3z(a)+p3z(p))/2;
pal0=[pal0x pal0y pal0z]; pal1=[pal1x pal1y pal1z];
pal2=[pal2x pal2y pal2z];
pA10(c,:)=pal0; pA11(c,:)=pal1; pA12(c,:)=pal2;
P1=pal1-pal0;
P2=pal2-pal0;
P3=cross(P1,P2);
%-----
%Check for HA being parallel to plane want to see where
%intersects
%-----
%Thigh
tVect=tn-ts;
tp3dot(c,1)=dot(P3,tVect);
if tp3dot==0
    msg=['**** ' ii ' Vector is parallel to the plane'];
    disp(msg);
    continue
end
%Shank
sVect=sn-ss;
sp3dot(c,1)=dot(P3,sVect);
if sp3dot==0
    msg=['**** ' ii ' Vector is parallel to the plane'];
    disp(msg);
end

```



```

        continue
    end

%-----
%Where the intersection occurs in the plane of interest
%-----
%Thigh
tThigh(ii)=(P3(1)*(pal0x-tn(1))+P3(2)*(pal0y-tn(2))+P3(3)*...
            (pal0z-tn(3)))/(P3(1)*tVect(1)+P3(2)*tVect(2)+...
            P3(3)*tVect(3));
ttx(ii)=tn(1)+tVect(1)*tThigh(ii);
tty(ii)=tn(2)+tVect(2)*tThigh(ii);
ttz(ii)=tn(3)+tVect(3)*tThigh(ii);
ttv=[ttx(ii) tty(ii) ttz(ii)];
%Shank
sThigh(ii)=(P3(1)*(pal0x-sn(1))+P3(2)*(pal0y-sn(2))+P3(3)*...
            (pal0z-sn(3)))/(P3(1)*sVect(1)+P3(2)*sVect(2)+...
            P3(3)*sVect(3));
stx(ii)=sn(1)+sVect(1)*sThigh(ii);
sty(ii)=sn(2)+sVect(2)*sThigh(ii);
stz(ii)=sn(3)+sVect(3)*sThigh(ii);
stv=[stx(ii) sty(ii) stz(ii)];
elseif mov=='FE'
%-----
%Plane to see where the HA intersects is the tibial plateau
%To get this plane take three of the four corners of the
%top of the shank
%-----
pal0x=(p0x(a)+p0x(p))/2; pal0y=(p0y(a)+p0y(p))/2;
pal0z=(p0z(a)+p0z(p))/2;
pal1x=(p1x(a)+p1x(p))/2; pal1y=(p1y(a)+p1y(p))/2;
pal1z=(p1z(a)+p1z(p))/2;
pal2x=(p2x(a)+p2x(p))/2; pal2y=(p2y(a)+p2y(p))/2;
pal2z=(p2z(a)+p2z(p))/2;
pal3x=(p3x(a)+p3x(p))/2; pal3y=(p3y(a)+p3y(p))/2;
pal3z=(p3z(a)+p3z(p))/2;

if mar_loc=='L'
    aLx(c,1)=pal0x+19;
    aLy(c,1)=pal0y-18;
    aLz(c,1)=pal0z+8.5;
elseif mar_loc=='R'
    aLx(c,1)=pal1x-21;
    aLy(c,1)=pal1y-19;
    aLz(c,1)=pal0z+8.5;
end
pal0=[pal0x pal0y pal0z]; pal1=[pal1x pal1y pal1z];
pal2=[pal2x pal2y pal2z];
pA10(c,:)=pal0; pA11(c,:)=pal1; pA12(c,:)=pal2;
p1=pal1-pal0;
p2=pal2-pal0;
p3=cross(p1,p2);
A=p3(1); B=p3(2); C=p3(3);

```

```

asb=-1; asc=1; asa=(-B*asb-C*asc)/A;

a1=asa; b1=asb; c1=asc; P3=[a1 b1 c1];
d1=-a1*aLx(a)-b1*aLy(a)-c1*aLz(a);
Plane1=[A B C]; Plane2=[a1 b1 c1];
perp_check(c,1)=dot(Plane1,Plane2);
%-----
%Check for HA being parallel to plane want to see where
%intersects
%-----
%Thigh
tVect=tn-ts;
tp3dot(c,1)=dot(P3,tVect);
if tp3dot==0
    msg=['**** ' ii ' Vector is parallel to the plane'];
    disp(msg);
    continue
end
%Shank
sVect=sn-ss;
sp3dot(c,1)=dot(P3,sVect);
if sp3dot==0
    msg=['**** ' ii ' Vector is parallel to the plane'];
    disp(msg);
    continue
end

%-----
%Where the intersection occurs in the plane of interest
%-----
%Thigh
tul=(a1*tn(1)+b1*tn(2)+c1*tn(3)+d1)/(a1*(tn(1)-ts(1))+...
    b1*(tn(2)-ts(2))+c1*(tn(3)-ts(3)));
tP_int=tn+tul*(ts-tn);
tPInt(c,1)=tP_int(1); tPInt(c,2)=tP_int(2);
tPInt(c,3)=tP_int(3);
ttx(ii)=tP_int(1);
tty(ii)=tP_int(2);
ttz(ii)=tP_int(3);
ttv=[ttx(ii) tty(ii) ttz(ii)];
%Shank
sul=(a1*sn(1)+b1*sn(2)+c1*sn(3)+d1)/(a1*(sn(1)-ss(1))+...
    b1*(sn(2)-ss(2))+c1*(sn(3)-ss(3)));
sP_int=sn+sul*(ss-sn);
sPInt(c,1)=sP_int(1); sPInt(c,2)=sP_int(2);
sPInt(c,3)=sP_int(3);
stx(ii)=sP_int(1);
sty(ii)=sP_int(2);
stz(ii)=sP_int(3);
stv=[stx(ii) sty(ii) stz(ii)];
end
%-----
%Calculate Marble Location

```

```

%-----
if mar_loc=='L'
    Mx(c,1)=pal0x+19;
    My(c,1)=pal0y-18;
    Mz(c,1)=pal0z+8.5;
    Tmar(c,1)=sqrt((ttx(ii)-Mx(c,1))^2+(tty(ii)-My(c,1))^2+...
        (ttz(ii)-Mz(c,1))^2);
    Smar(c,1)=sqrt((stx(ii)-Mx(c,1))^2+(sty(ii)-My(c,1))^2+...
        (stz(ii)-Mz(c,1))^2);
elseif mar_loc=='R'
    Mx(c,1)=pallx-21;
    My(c,1)=pally-19;
    Mz(c,1)=pal0z+8.5;
    Tmar(c,1)=sqrt((ttx(ii)-Mx(c,1))^2+(tty(ii)-My(c,1))^2+...
        (ttz(ii)-Mz(c,1))^2);
    Smar(c,1)=sqrt((stx(ii)-Mx(c,1))^2+(sty(ii)-My(c,1))^2+...
        (stz(ii)-Mz(c,1))^2);
end
if flag2==1
    figure(2)
    hold on
    scatter3(ttx(ii), tty(ii),ttz(ii), 'b')
    scatter3(stx(ii), sty(ii),stz(ii), 'sr')
    scatter3(Mx(c),My(c),Mz(c), 'FaceColor', 'k')
    grid on
    xlabel('x')
    ylabel('y')
    zlabel('z')
    view(37,28)
    if ii==len-1
        fig=fig+2;
    end
end
%-----
%Length of triangle sides used to calculate the helical axis to
%ensure that the sides are consistent throughout the movement
%-----
ts1=sqrt((txl0(a)-txl1(a))^2+(tyl0(a)-tyl1(a))^2+...
    (tzl0(a)-tzl1(a))^2);
ts2=sqrt((txl1(a)-txl2(a))^2+(tyl1(a)-tyl2(a))^2+...
    (tzl1(a)-tzl2(a))^2);
ts3=sqrt((txl2(a)-txl0(a))^2+(tyl2(a)-tyl0(a))^2+...
    (tzl2(a)-tzl0(a))^2);
tside(c,1)=ts1; tside(c,2)=ts2; tside(c,3)=ts3;
ss1=sqrt((sxl0(a)-sxl1(a))^2+(syl0(a)-syl1(a))^2+...
    (szl0(a)-szl1(a))^2);
ss2=sqrt((sxl1(a)-sxl2(a))^2+(syl1(a)-syl2(a))^2+...
    (szl1(a)-szl2(a))^2);
ss3=sqrt((sxl2(a)-sxl0(a))^2+(syl2(a)-syl0(a))^2+...
    (szl2(a)-szl0(a))^2);
sside(c,1)=ss1; sside(c,2)=ss2; sside(c,3)=ss3;

c=c+1;

```

```

end
tPHI_deg=tPHI*180/pi;
sPHI_deg=sPHI*180/pi;
if flag3==1
    figure(fig)
    hold on
    plot(tPHI_deg, 'r')
    plot(sPHI_deg, 'b')
    title('Phi')
    legend('Thigh', 'Shank')
    fig=fig+1;
    %
    figure(fig)
    hold on
    scatter3(tS(:,1), tS(:,2), tS(:,3), 'r')
    scatter3(sS(:,1), sS(:,2), sS(:,3), 'b')
    xlabel('x')
    ylabel('y')
    zlabel('z')
    grid on
    title('Radius Vector')
    legend('Thigh', 'Shank')
    fig=fig+1;

    figure(fig)
    hold on
    scatter3(tN(:,1), tN(:,2), tN(:,3), 'r')
    scatter3(sN(:,1), sN(:,2), sN(:,3), 'b')
    xlabel('x')
    ylabel('y')
    zlabel('z')
    grid on
    title('Unit Vector')
    legend('Thigh', 'Shank')
    fig=fig+1;

    figure(fig)
    hold on
    plot(tT, 'r')
    plot(sT, 'b')
    title('Translation Component')
    legend('Thigh', 'Shank')
    fig=fig+1;
    %
    figure(fig)
    hold on
    scatter3(tV(:,1), tV(:,2), tV(:,3), 'r')
    scatter3(sV(:,1), sV(:,2), sV(:,3), 'b')
    xlabel('x')
    ylabel('y')
    zlabel('z')
    title('Translation Vector')
    legend('Thigh', 'Shank')

```

```

        grid on
        fig=fig+1;
end

Sheet_name=['Sheet' num2str(file_num)];
alen=1:length(Tmar);
title1={'Frame' 'Tmar' 'ttx' 'tty' 'ttz' 'Smar' 'stx' 'sty' 'stz' ...
        'mLx' 'mLy' 'mLz'};
sav_data1=[alen' Tmar ttx' tty' ttz' Smar stx' sty' stz' Mx My Mz];
xlswrite('vRBM_int Data', title1, Sheet_name, 'a1')
xlswrite('vRBM_int Data', sav_data1, Sheet_name, 'a2')
title2={'Tmar' 'Tside1' 'Tside2' 'Tside3' 'Smar' 'Sside1' 'Sside2' ...
        'Sside3'};
sav_data2=[Tmar tside Smar sside];
xlswrite('vRBM_Tri', title2, Sheet_name, 'a1')
xlswrite('vRBM_Tri', sav_data2, Sheet_name, 'a2')
temj=1;
Tmar_out=0;
temj=1;
Tmar_out=0;
for iii=1:length(Tmar)
    if Tmar(iii)>mean(Tmar)+std(Tmar)
        Tmar_out(iii,1)=iii;
    else
        Tmar(temj,1)=Tmar(iii);
        Tside(temj,:)=tside(iii,:);
        Ttx(temj,1)=ttx(iii);
        Tty(temj,1)=tty(iii);
        Ttz(temj,1)=ttz(iii);
        tLx(temj,1)=Mx(iii);
        tLy(temj,1)=My(iii);
        tLz(temj,1)=Mz(iii);
        Tmar_inc(temj,1)=iii;
        temj=temj+1;
    end
end
semj=1;
Smar_out=0;
for iii=1:length(Smar)
    if Smar(iii)>mean(Smar)+std(Smar)
        Smar_out(iii,1)=iii;
    else
        Smar(semj,1)=Smar(iii);
        Sside(semj,:)=sside(iii,:);
        Stx(semj,1)=stx(iii);
        Sty(semj,1)=sty(iii);
        Stz(semj,1)=stz(iii);
        sLx(semj,1)=Mx(iii);
        sLy(semj,1)=My(iii);
        sLz(semj,1)=Mz(iii);
        Smar_inc(semj,1)=iii;
        semj=semj+1;
    end
end

```

```

end
TLen=length(TMar);
TLen2=(TLen-1000)/2;
if isa(TLen2, 'integer')==1
    count1=1;
    for m=TLen2:TLen2+1000
        TMar_Inc(count1,1)=Tmar_inc(m);
        TMar(count1,1)=TMar(m);
        TTx(count1,1)=Ttx(m);
        TTy(count1,1)=Tty(m);
        TTz(count1,1)=Ttz(m);
        TLx(count1,1)=tLx(m);
        TLy(count1,1)=tLy(m);
        TLz(count1,1)=tLz(m);
        count1=count1+1;
    end
else
    count2=1;
    for mm=TLen2-0.5:TLen2+1000.5
        TMar_Inc(count2,1)=Tmar_inc(mm);
        TMar(count2,1)=TMar(mm);
        TTx(count2,1)=Ttx(mm);
        TTy(count2,1)=Tty(mm);
        TTz(count2,1)=Ttz(mm);
        TLx(count2,1)=tLx(mm);
        TLy(count2,1)=tLy(mm);
        TLz(count2,1)=tLz(mm);
        count2=count2+1;
    end
end
SLen=length(TMar);
SLen2=(SLen-1000)/2;
if isa(SLen2, 'integer')==1
    count3=1;
    for mmm=SLen2:SLen2+1000
        SMAR_Inc(count3)=Smar_inc(mmm);
        SMAR(count3)=SMAR(mmm);
        STx(count3)=Stx(mmm);
        STy(count3)=Sty(mmm);
        STz(count3)=Stz(mmm);
        SLx(count3)=sLx(mmm);
        SLy(count3)=sLy(mmm);
        SLz(count3)=sLz(mmm);
        count3=count3+1;
    end
else
    count4=1;
    for mmmm=SLen2-0.5:SLen2+1000.5
        SMAR_Inc(count4,1)=Smar_inc(mmmm);
        SMAR(count4,1)=SMAR(mmmm);
        STx(count4,1)=Stx(mmmm);
        STy(count4,1)=Sty(mmmm);
        STz(count4,1)=Stz(mmmm);
        SLx(count4,1)=sLx(mmmm);

```

```

        SLy(count4,1)=sLy(mmmm);
        SLz(count4,1)=sLz(mmmm);
        count4=count4+1;;
    end
end

[mean(TMAr) std(TMAr)]
tlen=1:length(TMAr);
slen=1:length(SMAr);
title3={'T_Frame' 'tFrames_Inc' 'Tmar' 'ttx' 'tty' 'ttz' 'tLx' 'tLy'...
        'tLz' 'T_Outliers' 'S_Frame' 'sFrames_Inc' 'Smar' 'stx' 'sty'...
        'stz' 'sLx' 'sLy' 'sLz' 'S_Outliers'};
sav_data3=[tlen' TMAr_Inc TMAr TTx TTy TTz TLx TLy TLz];
sav_data3_1=Tmar_out;
sav_data3_2=[slen' SMAr_Inc SMAr STx STy STz SLx SLy SLz];
sav_data3_3=Smar_out;
xlswrite('vRBM_int Outliers', title3, Sheet_name, 'a1')
xlswrite('vRBM_int Outliers', sav_data3, Sheet_name, 'a2')
xlswrite('vRBM_int Outliers', sav_data3_1, Sheet_name, 'j2')
xlswrite('vRBM_int Outliers', sav_data3_2, Sheet_name, 'k2')
xlswrite('vRBM_int Outliers', sav_data3_3, Sheet_name, 't2')
title4={'Tmar' 'Tside1' 'Tside2' 'Tside3' 'T_Outliers' 'S_Frame'...
        'Smar' 'Sside1' 'Sside2' 'Sside3' 'S_Outliers'};
sav_data4=[Tmar Tside];
sav_data4_1=Tmar_out;
sav_data4_2=[Smar Sside];
sav_data4_3=Smar_out;
xlswrite('vRBM_Tri Outliers', title4, Sheet_name, 'a1')
xlswrite('vRBM_Tri Outliers', sav_data4, Sheet_name, 'a2')
xlswrite('vRBM_Tri Outliers', sav_data4_1, Sheet_name, 'f2')
xlswrite('vRBM_Tri Outliers', sav_data4_2, Sheet_name, 'g2')
xlswrite('vRBM_Tri Outliers', sav_data4_3, Sheet_name, 'l2')
Row_num=['A' num2str(file_num)];
title5={'Average TMar' 'Average Ttx' 'Average Tty' 'Average Ttz' ...
        'std TMar' 'std Ttx' 'std Tty' 'std Ttz'};
Tavg_ICC=[mean(TMar) mean(Ttx) mean(Tty) mean(Ttz) std(TMar) std(Ttx)...
        std(Tty) std(Ttz)];
xlswrite('vRBM_Average', title5, 'Femur Int', 'a1')
xlswrite('vRBM_Average', Tavg_ICC, 'Femur Int', Row_num)
title6={'Average SMAr' 'Average Stx' 'Average Sty' 'Average Stz' ...
        'std TMar' 'std Ttx' 'std Tty' 'std Ttz'};
Savg_ICC=[mean(SMar) mean(Stx) mean(Sty) mean(Stz) std(SMar) std(Stx)...
        std(Sty) std(Stz)];
xlswrite('vRBM_Average', title6, 'Tibia Int', 'a1')
xlswrite('vRBM_Average', Savg_ICC, 'Tibia Int', Row_num)

```

```

%iscollinear.m
function[outdata]=iscollinear(a, b, c)

%Three points are collinear all three lie on a straight line. So the sum
%of the distance between points a and b and the distance between points
%b and c equals the sum between points a and c.

ab=sqrt((a(1)-b(1))^2+(a(2)-b(2))^2+(a(3)-b(3))^2);
bc=sqrt((b(1)-c(1))^2+(b(2)-c(2))^2+(b(3)-c(3))^2);
ac=sqrt((a(1)-c(1))^2+(a(2)-c(2))^2+(a(3)-c(3))^2);
if ab+bc==ac||bc+ac==bc||ac+bc==ab
    outdata=1;
else
    outdata=0;
end

```



```

%RMV.m
%-----
%Rotation matrix validation (RMV) -- Compute the sqrt of the sum of the
%squared values in each row and column. Return the product of all these
%values multiplied together. The answer should be '1' for a true
%rotation matrix.
%-----
function[outdata] = RMV(name,indata,indx)

[m,n]=size(indata);

%compute column values first
cnt=1;
for i=1:m
    sum = 0;
    for j=1:n
        sum = sum + indata(i,j)^2;
    end
    result(cnt) = sqrt(sum);
    cnt = cnt+1;
end

%compute row values next
for j=1:m
    sum = 0;
    for i=1:n
        sum = sum + indata(i,j)^2;
    end
    result(cnt) = sqrt(sum);
    cnt = cnt+1;
end

%check all results
outdata = 1;
cnt=cnt-1;
for i=1:cnt
    outdata = outdata * result(i);
end

if cast(outdata,'single') ~= 1
    msg=['*** ' name '(' num2str(indx) ') is not a rotation matrix.'];
    disp(msg);
end

```

```

%summer.m
function [R, S]=summer(G)

%Positive polar decomposition of 3x3 matrix G=R*S
%Forcing positive orthonormal rotation matrix

%Inputs
%G=3x3 general matrix

%Outputs
%R=3x3 positive orthonormal matrix
%S=3x3 symmetric matrix

EPS=1.0e-5;

%Cofactors and Determinant of g
COG(1,1)=G(2,2)*G(3,3)-G(2,3)*G(3,2);
COG(2,1)=G(1,3)*G(3,2)-G(1,2)*G(3,3);
COG(3,1)=G(1,2)*G(2,3)-G(1,3)*G(2,2);
COG(1,2)=G(2,3)*G(3,1)-G(2,1)*G(3,3);
COG(2,2)=G(1,1)*G(3,3)-G(1,3)*G(3,1);
COG(3,2)=G(1,3)*G(2,1)-G(1,1)*G(2,3);
COG(1,3)=G(2,1)*G(3,2)-G(2,2)*G(3,1);
COG(2,3)=G(1,2)*G(3,1)-G(1,1)*G(3,2);
COG(3,3)=G(1,1)*G(2,2)-G(1,2)*G(2,1);
G3=G(1,1)*COG(1,1)+G(2,1)*COG(2,1)+G(3,1)*COG(3,1);

P=G'*G;
%Adjoint of P
ADP(1,1)=P(2,2)*P(3,3)-P(2,3)*P(3,2);
ADP(2,2)=P(1,1)*P(3,3)-P(1,3)*P(3,1);
ADP(3,3)=P(1,1)*P(2,2)-P(1,2)*P(2,1);

%G invariants
G1SQ=P(1,1)+P(2,2)+P(3,3);
G1=sqrt(G1SQ);
G2SQ=ADP(1,1)+ADP(2,2)+ADP(3,3);
G2=sqrt(G2SQ);

%Initialize Iteration
H1=G2/G1SQ;
H2=G3*G1/G2SQ;
X=1.0;
Y=1.0;
DX=1.0;
DY=1.0;

%Iteration Loop
while abs(DX/X)>EPS || abs(DY/Y)>EPS
    DEN=2.0*(X*Y-H1*H2);
    RES1=1.0-X*X+2.0*H1*Y;
    RES2=1.0-Y*Y+2.0*H2*X;

```

```

DX=(Y*RES1+H1*RES2)/DEN;
DY=(H2*RES1+X*RES2)/DEN;
X=X+DX;
Y=Y+DY;
end

%Beta Invariants
BETA1=X*G1;
BETA2=Y*G2;

%Invert(G'*G+BETA2*I)
P(1,1)=P(1,1)+BETA2;
P(2,2)=P(2,2)+BETA2;
P(3,3)=P(3,3)+BETA2;
PBI(1,1)=P(2,2)*P(3,3)-P(2,3)*P(3,2);
PBI(1,2)=P(1,3)*P(3,2)-P(1,2)*P(3,3);
PBI(1,3)=P(1,2)*P(2,3)-P(1,3)*P(2,2);
PBI(2,1)=P(2,3)*P(3,1)-P(2,1)*P(3,3);
PBI(2,2)=P(1,1)*P(3,3)-P(1,3)*P(3,1);
PBI(2,3)=P(1,3)*P(2,1)-P(1,1)*P(2,3);
PBI(3,1)=P(2,1)*P(3,2)-P(2,2)*P(3,1);
PBI(3,2)=P(1,2)*P(3,1)-P(1,1)*P(3,2);
PBI(3,3)=P(1,1)*P(2,2)-P(1,2)*P(2,1);
DETPBI=P(1,1)*PBI(1,1)+P(2,1)*PBI(1,2)+P(3,1)*PBI(1,3);

for i=1:3

R(i,1)=((COG(i,1)+BETA1*G(i,1))*PBI(1,1)+(COG(i,2)+BETA1*G(i,2))*PBI(2,1)
)+(COG(i,3)+BETA1*G(i,3))*PBI(3,1))/DETPBI;

R(i,2)=((COG(i,1)+BETA1*G(i,1))*PBI(1,2)+(COG(i,2)+BETA1*G(i,2))*PBI(2,2)
)+(COG(i,3)+BETA1*G(i,3))*PBI(3,2))/DETPBI;

R(i,3)=((COG(i,1)+BETA1*G(i,1))*PBI(1,3)+(COG(i,2)+BETA1*G(i,2))*PBI(2,3)
)+(COG(i,3)+BETA1*G(i,3))*PBI(3,3))/DETPBI;
end
%
for i=1:3
S(i,1)=R(1,i)*G(1,1)+R(2,i)*G(2,1)+R(3,i)*G(3,1);
S(i,2)=R(1,i)*G(1,2)+R(2,i)*G(2,2)+R(3,i)*G(3,2);
S(i,3)=R(1,i)*G(1,3)+R(2,i)*G(2,3)+R(3,i)*G(3,3);
end

%
```

APPENDIX C

A typical trial of the Modeled data used to calculate the helical axis.

Raw Data														
1				2				3				Triangle Center		
x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x (cm)	y (cm)	z (cm)
0	1	0		0.8628	-0.4981	-0.0863		-0.8628	-0.4981	0.0863		0.0000	0.00127	0.00000
0.0175	0.9998	0.0035		0.8538	-0.513	-0.088		-0.8715	-0.4831	0.0845		-0.00007	0.00123	0.00000
0.0349	0.9994	0.007		0.8446	-0.5278	-0.0895		-0.8799	-0.4679	0.0826		-0.00013	0.00123	0.00003
0.0523	0.9986	0.0105		0.8352	-0.5424	-0.091		-0.8881	-0.4525	0.0806		-0.00020	0.00123	0.00003
0.0697	0.9975	0.0139		0.8255	-0.5568	-0.0923		-0.896	-0.437	0.0786		-0.00027	0.00123	0.00007
0.0871	0.996	0.0174		0.8156	-0.5711	-0.0936		-0.9037	-0.4214	0.0764		-0.00033	0.00117	0.00007
0.1045	0.9943	0.0208		0.8054	-0.5851	-0.0947		-0.911	-0.4056	0.0741		-0.00037	0.00120	0.00007
0.1218	0.9923	0.0242		0.795	-0.5991	-0.0957		-0.9181	-0.3897	0.0717		-0.00043	0.00117	0.00007
0.1391	0.9899	0.0276		0.7843	-0.6128	-0.0966		-0.925	-0.3737	0.0693		-0.00053	0.00113	0.00010
0.1564	0.9872	0.0309		0.7735	-0.6263	-0.0974		-0.9315	-0.3576	0.0668		-0.00053	0.00110	0.00010
0.1735	0.9842	0.0342		0.7624	-0.6397	-0.098		-0.9378	-0.3413	0.0641		-0.00063	0.00107	0.00010
0.1907	0.9809	0.0374		0.751	-0.6529	-0.0985		-0.9437	-0.325	0.0614		-0.00067	0.00100	0.00010
0.2077	0.9773	0.0406		0.7395	-0.6658	-0.099		-0.9494	-0.3085	0.0587		-0.00073	0.00100	0.00010
0.2247	0.9734	0.0438		0.7277	-0.6786	-0.0993		-0.9548	-0.2919	0.0558		-0.00080	0.00097	0.00010
0.2417	0.9692	0.0469		0.7158	-0.6912	-0.0994		-0.9599	-0.2753	0.0529		-0.00080	0.00090	0.00013
0.2585	0.9647	0.0499		0.7036	-0.7036	-0.0995		-0.9647	-0.2585	0.0499		-0.00087	0.00087	0.00010
0.2753	0.9599	0.0529		0.6912	-0.7158	-0.0994		-0.9692	-0.2417	0.0469		-0.00090	0.00080	0.00013
0.2919	0.9548	0.0558		0.6786	-0.7277	-0.0993		-0.9734	-0.2247	0.0438		-0.00097	0.00080	0.00010
0.3085	0.9494	0.0587		0.6658	-0.7395	-0.099		-0.9773	-0.2077	0.0406		-0.00100	0.00073	0.00010
0.325	0.9437	0.0614		0.6529	-0.751	-0.0985		-0.9809	-0.1907	0.0374		-0.00100	0.00067	0.00010
0.3413	0.9378	0.0641		0.6397	-0.7624	-0.098		-0.9842	-0.1735	0.0342		-0.00107	0.00063	0.00010
0.3576	0.9315	0.0668		0.6263	-0.7735	-0.0974		-0.9872	-0.1564	0.0309		-0.00110	0.00053	0.00010
0.3737	0.925	0.0693		0.6128	-0.7843	-0.0966		-0.9899	-0.1391	0.0276		-0.00113	0.00053	0.00010
0.3897	0.9181	0.0717		0.5991	-0.795	-0.0957		-0.9923	-0.1218	0.0242		-0.00117	0.00043	0.00007
0.4056	0.911	0.0741		0.5851	-0.8054	-0.0947		-0.9943	-0.1045	0.0208		-0.00120	0.00037	0.00007
0.4214	0.9037	0.0764		0.5711	-0.8156	-0.0936		-0.996	-0.0871	0.0174		-0.00117	0.00033	0.00007
0.437	0.896	0.0786		0.5568	-0.8255	-0.0923		-0.9975	-0.0697	0.0139		-0.00123	0.00027	0.00007

Raw Data														
1				2				3				Triangle Center		
x' (cm)	y' (cm)	z' (cm)	x' (cm)	y' (cm)	z' (cm)	x' (cm)	y' (cm)	z' (cm)	x' (cm)	y' (cm)	z' (cm)	x (cm)	y (cm)	z (cm)
0.4525	0.8881	0.0806	0.5424	-0.8352	-0.091	-0.9986	-0.0523	0.0105	-0.00123	0.00020	0.00003			
0.4679	0.8799	0.0826	0.5278	-0.8446	-0.0895	-0.9994	-0.0349	0.007	-0.00123	0.00013	0.00003			
0.4831	0.8715	0.0845	0.513	-0.8538	-0.088	-0.9998	-0.0175	0.0035	-0.00123	0.00007	0.00000			
0.4981	0.8628	0.0863	0.4981	-0.8628	-0.0863	-1	0	0	-0.00127	0.00000	0.00000			
0.513	0.8538	0.088	0.4831	-0.8715	-0.0845	-0.9998	0.0175	-0.0035	-0.00123	-0.00007	0.00000			
0.5278	0.8446	0.0895	0.4679	-0.8799	-0.0826	-0.9994	0.0349	-0.007	-0.00123	-0.00013	-0.00003			
0.5424	0.8352	0.091	0.4525	-0.8881	-0.0806	-0.9986	0.0523	-0.0105	-0.00123	-0.00020	-0.00003			
0.5568	0.8255	0.0923	0.437	-0.896	-0.0786	-0.9975	0.0697	-0.0139	-0.00123	-0.00027	-0.00007			
0.5711	0.8156	0.0936	0.4214	-0.9037	-0.0764	-0.996	0.0871	-0.0174	-0.00117	-0.00033	-0.00007			
0.5851	0.8054	0.0947	0.4056	-0.911	-0.0741	-0.9943	0.1045	-0.0208	-0.00120	-0.00037	-0.00007			
0.5991	0.795	0.0957	0.3897	-0.9181	-0.0717	-0.9923	0.1218	-0.0242	-0.00117	-0.00043	-0.00007			
0.6128	0.7843	0.0966	0.3737	-0.925	-0.0693	-0.9899	0.1391	-0.0276	-0.00113	-0.00053	-0.00010			
0.6263	0.7735	0.0974	0.3576	-0.9315	-0.0668	-0.9872	0.1564	-0.0309	-0.00110	-0.00053	-0.00010			
0.6397	0.7624	0.098	0.3413	-0.9378	-0.0641	-0.9842	0.1735	-0.0342	-0.00107	-0.00063	-0.00010			
0.6529	0.751	0.0985	0.325	-0.9437	-0.0614	-0.9809	0.1907	-0.0374	-0.00100	-0.00067	-0.00010			
0.6658	0.7395	0.099	0.3085	-0.9494	-0.0587	-0.9773	0.2077	-0.0406	-0.00100	-0.00073	-0.00010			
0.6786	0.7277	0.0993	0.2919	-0.9548	-0.0558	-0.9734	0.2247	-0.0438	-0.00097	-0.00080	-0.00010			
0.6912	0.7158	0.0994	0.2753	-0.9599	-0.0529	-0.9692	0.2417	-0.0469	-0.00090	-0.00080	-0.00013			
0.7036	0.7036	0.0995	0.2585	-0.9647	-0.0499	-0.9647	0.2585	-0.0499	-0.00087	-0.00087	-0.00010			
0.7158	0.6912	0.0994	0.2417	-0.9692	-0.0469	-0.9599	0.2753	-0.0529	-0.00080	-0.00090	-0.00013			
0.7277	0.6786	0.0993	0.2247	-0.9734	-0.0438	-0.9548	0.2919	-0.0558	-0.00080	-0.00097	-0.00010			
0.7395	0.6658	0.099	0.2077	-0.9773	-0.0406	-0.9494	0.3085	-0.0587	-0.00073	-0.00100	-0.00010			
0.751	0.6529	0.0985	0.1907	-0.9809	-0.0374	-0.9437	0.325	-0.0614	-0.00067	-0.00100	-0.00010			
0.7624	0.6397	0.098	0.1735	-0.9842	-0.0342	-0.9378	0.3413	-0.0641	-0.00063	-0.00107	-0.00010			
0.7735	0.6263	0.0974	0.1564	-0.9872	-0.0309	-0.9315	0.3576	-0.0668	-0.00053	-0.00110	-0.00010			
0.7843	0.6128	0.0966	0.1391	-0.9899	-0.0276	-0.925	0.3737	-0.0693	-0.00053	-0.00113	-0.00010			
0.795	0.5991	0.0957	0.1218	-0.9923	-0.0242	-0.9181	0.3897	-0.0717	-0.00043	-0.00117	-0.00007			

Raw Data														
1				2				3				Triangle Center		
x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x (cm)	y (cm)	z (cm)
0.8054	0.5851	0.0947		0.1045	-0.9943	-0.0208		-0.911	0.4056	-0.0741		-0.00037	-0.00120	-0.00007
0.8156	0.5711	0.0936		0.0871	-0.996	-0.0174		-0.9037	0.4214	-0.0764		-0.00033	-0.00117	-0.00007
0.8255	0.5568	0.0923		0.0697	-0.9975	-0.0139		-0.896	0.437	-0.0786		-0.00027	-0.00123	-0.00007
0.8352	0.5424	0.091		0.0523	-0.9986	-0.0105		-0.8881	0.4525	-0.0806		-0.00020	-0.00123	-0.00003
0.8446	0.5278	0.0895		0.0349	-0.9994	-0.007		-0.8799	0.4679	-0.0826		-0.00013	-0.00123	-0.00003
0.8538	0.513	0.088		0.0175	-0.9998	-0.0035		-0.8715	0.4831	-0.0845		-0.00007	-0.00123	0.00000
0.8628	0.4981	0.0863		0	-1	0		-0.8628	0.4981	-0.0863		0.00000	-0.00127	0.00000
0.8715	0.4831	0.0845		-0.0175	-0.9998	0.0035		-0.8538	0.513	-0.088		0.00007	-0.00123	0.00000
0.8799	0.4679	0.0826		-0.0349	-0.9994	0.007		-0.8446	0.5278	-0.0895		0.00013	-0.00123	0.00003
0.8881	0.4525	0.0806		-0.0523	-0.9986	0.0105		-0.8352	0.5424	-0.091		0.00020	-0.00123	0.00003
0.896	0.437	0.0786		-0.0697	-0.9975	0.0139		-0.8255	0.5568	-0.0923		0.00027	-0.00123	0.00007
0.9037	0.4214	0.0764		-0.0871	-0.996	0.0174		-0.8156	0.5711	-0.0936		0.00033	-0.00117	0.00007
0.911	0.4056	0.0741		-0.1045	-0.9943	0.0208		-0.8054	0.5851	-0.0947		0.00037	-0.00120	0.00007
0.9181	0.3897	0.0717		-0.1218	-0.9923	0.0242		-0.795	0.5991	-0.0957		0.00043	-0.00117	0.00007
0.925	0.3737	0.0693		-0.1391	-0.9899	0.0276		-0.7843	0.6128	-0.0966		0.00053	-0.00113	0.00010
0.9315	0.3576	0.0668		-0.1564	-0.9872	0.0309		-0.7735	0.6263	-0.0974		0.00053	-0.00110	0.00010
0.9378	0.3413	0.0641		-0.1735	-0.9842	0.0342		-0.7624	0.6397	-0.098		0.00063	-0.00107	0.00010
0.9437	0.325	0.0614		-0.1907	-0.9809	0.0374		-0.751	0.6529	-0.0985		0.00067	-0.00100	0.00010
0.9494	0.3085	0.0587		-0.2077	-0.9773	0.0406		-0.7395	0.6658	-0.099		0.00073	-0.00100	0.00010
0.9548	0.2919	0.0558		-0.2247	-0.9734	0.0438		-0.7277	0.6786	-0.0993		0.00080	-0.00097	0.00010
0.9599	0.2753	0.0529		-0.2417	-0.9692	0.0469		-0.7158	0.6912	-0.0994		0.00080	-0.00090	0.00013
0.9647	0.2585	0.0499		-0.2585	-0.9647	0.0499		-0.7036	0.7036	-0.0995		0.00087	-0.00087	0.00010
0.9692	0.2417	0.0469		-0.2753	-0.9599	0.0529		-0.6912	0.7158	-0.0994		0.00090	-0.00080	0.00013
0.9734	0.2247	0.0438		-0.2919	-0.9548	0.0558		-0.6786	0.7277	-0.0993		0.00097	-0.00080	0.00010
0.9773	0.2077	0.0406		-0.3085	-0.9494	0.0587		-0.6658	0.7395	-0.099		0.00100	-0.00073	0.00010
0.9809	0.1907	0.0374		-0.325	-0.9437	0.0614		-0.6529	0.751	-0.0985		0.00100	-0.00067	0.00010
0.9842	0.1735	0.0342		-0.3413	-0.9378	0.0641		-0.6397	0.7624	-0.098		0.00107	-0.00063	0.00010

Raw Data														
1				2				3				Triangle Center		
x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x (cm)	y (cm)	z (cm)
0.9872	0.1564	0.0309		-0.3576	-0.9315	0.0668		-0.6263	0.7735	-0.0974		0.00110	-0.00053	0.00010
0.9899	0.1391	0.0276		-0.3737	-0.925	0.0693		-0.6128	0.7843	-0.0966		0.00113	-0.00053	0.00010
0.9923	0.1218	0.0242		-0.3897	-0.9181	0.0717		-0.5991	0.795	-0.0957		0.00117	-0.00043	0.00007
0.9943	0.1045	0.0208		-0.4056	-0.911	0.0741		-0.5851	0.8054	-0.0947		0.00120	-0.00037	0.00007
0.996	0.0871	0.0174		-0.4214	-0.9037	0.0764		-0.5711	0.8156	-0.0936		0.00117	-0.00033	0.00007
0.9975	0.0697	0.0139		-0.437	-0.896	0.0786		-0.5568	0.8255	-0.0923		0.00123	-0.00027	0.00007
0.9986	0.0523	0.0105		-0.4525	-0.8881	0.0806		-0.5424	0.8352	-0.091		0.00123	-0.00020	0.00003
0.9994	0.0349	0.007		-0.4679	-0.8799	0.0826		-0.5278	0.8446	-0.0895		0.00123	-0.00013	0.00003
0.9998	0.0175	0.0035		-0.4831	-0.8715	0.0845		-0.513	0.8538	-0.088		0.00123	-0.00007	0.00000
1	0	0		-0.4981	-0.8628	0.0863		-0.4981	0.8628	-0.0863		0.00127	0.00000	0.00000
0.9998	-0.0175	-0.0035		-0.513	-0.8538	0.088		-0.4831	0.8715	-0.0845		0.00123	0.00007	0.00000
0.9994	-0.0349	-0.007		-0.5278	-0.8446	0.0895		-0.4679	0.8799	-0.0826		0.00123	0.00013	-0.00003
0.9986	-0.0523	-0.0105		-0.5424	-0.8352	0.091		-0.4525	0.8881	-0.0806		0.00123	0.00020	-0.00003
0.9975	-0.0697	-0.0139		-0.5568	-0.8255	0.0923		-0.437	0.896	-0.0786		0.00123	0.00027	-0.00007
0.996	-0.0871	-0.0174		-0.5711	-0.8156	0.0936		-0.4214	0.9037	-0.0764		0.00117	0.00033	-0.00007
0.9943	-0.1045	-0.0208		-0.5851	-0.8054	0.0947		-0.4056	0.911	-0.0741		0.00120	0.00037	-0.00007
0.9923	-0.1218	-0.0242		-0.5991	-0.795	0.0957		-0.3897	0.9181	-0.0717		0.00117	0.00043	-0.00007
0.9899	-0.1391	-0.0276		-0.6128	-0.7843	0.0966		-0.3737	0.925	-0.0693		0.00113	0.00053	-0.00010
0.9872	-0.1564	-0.0309		-0.6263	-0.7735	0.0974		-0.3576	0.9315	-0.0668		0.00110	0.00053	-0.00010
0.9842	-0.1735	-0.0342		-0.6397	-0.7624	0.098		-0.3413	0.9378	-0.0641		0.00107	0.00063	-0.00010
0.9809	-0.1907	-0.0374		-0.6529	-0.751	0.0985		-0.325	0.9437	-0.0614		0.00100	0.00067	-0.00010
0.9773	-0.2077	-0.0406		-0.6658	-0.7395	0.099		-0.3085	0.9494	-0.0587		0.00100	0.00073	-0.00010
0.9734	-0.2247	-0.0438		-0.6786	-0.7277	0.0993		-0.2919	0.9548	-0.0558		0.00097	0.00080	-0.00010
0.9692	-0.2417	-0.0469		-0.6912	-0.7158	0.0994		-0.2753	0.9599	-0.0529		0.00090	0.00080	-0.00013
0.9647	-0.2585	-0.0499		-0.7036	-0.7036	0.0995		-0.2585	0.9647	-0.0499		0.00087	0.00087	-0.00010
0.9599	-0.2753	-0.0529		-0.7158	-0.6912	0.0994		-0.2417	0.9692	-0.0469		0.00080	0.00090	-0.00013
0.9548	-0.2919	-0.0558		-0.7277	-0.6786	0.0993		-0.2247	0.9734	-0.0438		0.00080	0.00097	-0.00010

Raw Data														
1				2				3				Triangle Center		
x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x (cm)	y (cm)	z (cm)
0.9494	-0.3085	-0.0587		-0.7395	-0.6658	0.099		-0.2077	0.9773	-0.0406		0.00073	0.00100	-0.00010
0.9437	-0.325	-0.0614		-0.751	-0.6529	0.0985		-0.1907	0.9809	-0.0374		0.00067	0.00100	-0.00010
0.9378	-0.3413	-0.0641		-0.7624	-0.6397	0.098		-0.1735	0.9842	-0.0342		0.00063	0.00107	-0.00010
0.9315	-0.3576	-0.0668		-0.7735	-0.6263	0.0974		-0.1564	0.9872	-0.0309		0.00053	0.00110	-0.00010
0.925	-0.3737	-0.0693		-0.7843	-0.6128	0.0966		-0.1391	0.9899	-0.0276		0.00053	0.00113	-0.00010
0.9181	-0.3897	-0.0717		-0.795	-0.5991	0.0957		-0.1218	0.9923	-0.0242		0.00043	0.00117	-0.00007
0.911	-0.4056	-0.0741		-0.8054	-0.5851	0.0947		-0.1045	0.9943	-0.0208		0.00037	0.00120	-0.00007
0.9037	-0.4214	-0.0764		-0.8156	-0.5711	0.0936		-0.0871	0.996	-0.0174		0.00033	0.00117	-0.00007
0.896	-0.437	-0.0786		-0.8255	-0.5568	0.0923		-0.0697	0.9975	-0.0139		0.00027	0.00123	-0.00007
0.8881	-0.4525	-0.0806		-0.8352	-0.5424	0.091		-0.0523	0.9986	-0.0105		0.00020	0.00123	-0.00003
0.8799	-0.4679	-0.0826		-0.8446	-0.5278	0.0895		-0.0349	0.9994	-0.007		0.00013	0.00123	-0.00003
0.8715	-0.4831	-0.0845		-0.8538	-0.513	0.088		-0.0175	0.9998	-0.0035		0.00007	0.00123	0.00000
0.8628	-0.4981	-0.0863		-0.8628	-0.4981	0.0863		0	1	0		0.00000	0.00127	0.00000
0.8538	-0.513	-0.088		-0.8715	-0.4831	0.0845		0.0175	0.9998	0.0035		-0.00007	0.00123	0.00000
0.8446	-0.5278	-0.0895		-0.8799	-0.4679	0.0826		0.0349	0.9994	0.007		-0.00013	0.00123	0.00003
0.8352	-0.5424	-0.091		-0.8881	-0.4525	0.0806		0.0523	0.9986	0.0105		-0.00020	0.00123	0.00003
0.8255	-0.5568	-0.0923		-0.896	-0.437	0.0786		0.0697	0.9975	0.0139		-0.00027	0.00123	0.00007
0.8156	-0.5711	-0.0936		-0.9037	-0.4214	0.0764		0.0871	0.996	0.0174		-0.00033	0.00117	0.00007
0.8054	-0.5851	-0.0947		-0.911	-0.4056	0.0741		0.1045	0.9943	0.0208		-0.00037	0.00120	0.00007
0.795	-0.5991	-0.0957		-0.9181	-0.3897	0.0717		0.1218	0.9923	0.0242		-0.00043	0.00117	0.00007
0.7843	-0.6128	-0.0966		-0.925	-0.3737	0.0693		0.1391	0.9899	0.0276		-0.00053	0.00113	0.00010
0.7735	-0.6263	-0.0974		-0.9315	-0.3576	0.0668		0.1564	0.9872	0.0309		-0.00053	0.00110	0.00010
0.7624	-0.6397	-0.098		-0.9378	-0.3413	0.0641		0.1735	0.9842	0.0342		-0.00063	0.00107	0.00010
0.751	-0.6529	-0.0985		-0.9437	-0.325	0.0614		0.1907	0.9809	0.0374		-0.00067	0.00100	0.00010
0.7395	-0.6658	-0.099		-0.9494	-0.3085	0.0587		0.2077	0.9773	0.0406		-0.00073	0.00100	0.00010
0.7277	-0.6786	-0.0993		-0.9548	-0.2919	0.0558		0.2247	0.9734	0.0438		-0.00080	0.00097	0.00010
0.7158	-0.6912	-0.0994		-0.9599	-0.2753	0.0529		0.2417	0.9692	0.0469		-0.00080	0.00090	0.00013

Raw Data														
1				2				3				Triangle Center		
x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x (cm)	y (cm)	z (cm)
0.7036	-0.7036	-0.0995		-0.9647	-0.2585	0.0499		0.2585	0.9647	0.0499		-0.00087	0.00087	0.00010
0.6912	-0.7158	-0.0994		-0.9692	-0.2417	0.0469		0.2753	0.9599	0.0529		-0.00090	0.00080	0.00013
0.6786	-0.7277	-0.0993		-0.9734	-0.2247	0.0438		0.2919	0.9548	0.0558		-0.00097	0.00080	0.00010
0.6658	-0.7395	-0.099		-0.9773	-0.2077	0.0406		0.3085	0.9494	0.0587		-0.00100	0.00073	0.00010
0.6529	-0.751	-0.0985		-0.9809	-0.1907	0.0374		0.325	0.9437	0.0614		-0.00100	0.00067	0.00010
0.6397	-0.7624	-0.098		-0.9842	-0.1735	0.0342		0.3413	0.9378	0.0641		-0.00107	0.00063	0.00010
0.6263	-0.7735	-0.0974		-0.9872	-0.1564	0.0309		0.3576	0.9315	0.0668		-0.00110	0.00053	0.00010
0.6128	-0.7843	-0.0966		-0.9899	-0.1391	0.0276		0.3737	0.925	0.0693		-0.00113	0.00053	0.00010
0.5991	-0.795	-0.0957		-0.9923	-0.1218	0.0242		0.3897	0.9181	0.0717		-0.00117	0.00043	0.00007
0.5851	-0.8054	-0.0947		-0.9943	-0.1045	0.0208		0.4056	0.911	0.0741		-0.00120	0.00037	0.00007
0.5711	-0.8156	-0.0936		-0.996	-0.0871	0.0174		0.4214	0.9037	0.0764		-0.00117	0.00033	0.00007
0.5568	-0.8255	-0.0923		-0.9975	-0.0697	0.0139		0.437	0.896	0.0786		-0.00123	0.00027	0.00007
0.5424	-0.8352	-0.091		-0.9986	-0.0523	0.0105		0.4525	0.8881	0.0806		-0.00123	0.00020	0.00003
0.5278	-0.8446	-0.0895		-0.9994	-0.0349	0.007		0.4679	0.8799	0.0826		-0.00123	0.00013	0.00003
0.513	-0.8538	-0.088		-0.9998	-0.0175	0.0035		0.4831	0.8715	0.0845		-0.00123	0.00007	0.00000
0.4981	-0.8628	-0.0863		-1	0	0		0.4981	0.8628	0.0863		-0.00127	0.00000	0.00000
0.4831	-0.8715	-0.0845		-0.9998	0.0175	-0.0035		0.513	0.8538	0.088		-0.00123	-0.00007	0.00000
0.4679	-0.8799	-0.0826		-0.9994	0.0349	-0.007		0.5278	0.8446	0.0895		-0.00123	-0.00013	-0.00003
0.4525	-0.8881	-0.0806		-0.9986	0.0523	-0.0105		0.5424	0.8352	0.091		-0.00123	-0.00020	-0.00003
0.437	-0.896	-0.0786		-0.9975	0.0697	-0.0139		0.5568	0.8255	0.0923		-0.00123	-0.00027	-0.00007
0.4214	-0.9037	-0.0764		-0.996	0.0871	-0.0174		0.5711	0.8156	0.0936		-0.00117	-0.00033	-0.00007
0.4056	-0.911	-0.0741		-0.9943	0.1045	-0.0208		0.5851	0.8054	0.0947		-0.00120	-0.00037	-0.00007
0.3897	-0.9181	-0.0717		-0.9923	0.1218	-0.0242		0.5991	0.795	0.0957		-0.00117	-0.00043	-0.00007
0.3737	-0.925	-0.0693		-0.9899	0.1391	-0.0276		0.6128	0.7843	0.0966		-0.00113	-0.00053	-0.00010
0.3576	-0.9315	-0.0668		-0.9872	0.1564	-0.0309		0.6263	0.7735	0.0974		-0.00110	-0.00053	-0.00010
0.3413	-0.9378	-0.0641		-0.9842	0.1735	-0.0342		0.6397	0.7624	0.098		-0.00107	-0.00063	-0.00010
0.325	-0.9437	-0.0614		-0.9809	0.1907	-0.0374		0.6529	0.751	0.0985		-0.00100	-0.00067	-0.00010

Raw Data														
1				2				3				Triangle Center		
x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x (cm)	y (cm)	z (cm)
0.3085	-0.9494	-0.0587		-0.9773	0.2077	-0.0406		0.6658	0.7395	0.099		-0.00100	-0.00073	-0.00010
0.2919	-0.9548	-0.0558		-0.9734	0.2247	-0.0438		0.6786	0.7277	0.0993		-0.00097	-0.00080	-0.00010
0.2753	-0.9599	-0.0529		-0.9692	0.2417	-0.0469		0.6912	0.7158	0.0994		-0.00090	-0.00080	-0.00013
0.2585	-0.9647	-0.0499		-0.9647	0.2585	-0.0499		0.7036	0.7036	0.0995		-0.00087	-0.00087	-0.00010
0.2417	-0.9692	-0.0469		-0.9599	0.2753	-0.0529		0.7158	0.6912	0.0994		-0.00080	-0.00090	-0.00013
0.2247	-0.9734	-0.0438		-0.9548	0.2919	-0.0558		0.7277	0.6786	0.0993		-0.00080	-0.00097	-0.00010
0.2077	-0.9773	-0.0406		-0.9494	0.3085	-0.0587		0.7395	0.6658	0.099		-0.00073	-0.00100	-0.00010
0.1907	-0.9809	-0.0374		-0.9437	0.325	-0.0614		0.751	0.6529	0.0985		-0.00067	-0.00100	-0.00010
0.1735	-0.9842	-0.0342		-0.9378	0.3413	-0.0641		0.7624	0.6397	0.098		-0.00063	-0.00107	-0.00010
0.1564	-0.9872	-0.0309		-0.9315	0.3576	-0.0668		0.7735	0.6263	0.0974		-0.00053	-0.00110	-0.00010
0.1391	-0.9899	-0.0276		-0.925	0.3737	-0.0693		0.7843	0.6128	0.0966		-0.00053	-0.00113	-0.00010
0.1218	-0.9923	-0.0242		-0.9181	0.3897	-0.0717		0.795	0.5991	0.0957		-0.00043	-0.00117	-0.00007
0.1045	-0.9943	-0.0208		-0.911	0.4056	-0.0741		0.8054	0.5851	0.0947		-0.00037	-0.00120	-0.00007
0.0871	-0.996	-0.0174		-0.9037	0.4214	-0.0764		0.8156	0.5711	0.0936		-0.00033	-0.00117	-0.00007
0.0697	-0.9975	-0.0139		-0.896	0.437	-0.0786		0.8255	0.5568	0.0923		-0.00027	-0.00123	-0.00007
0.0523	-0.9986	-0.0105		-0.8881	0.4525	-0.0806		0.8352	0.5424	0.091		-0.00020	-0.00123	-0.00003
0.0349	-0.9994	-0.007		-0.8799	0.4679	-0.0826		0.8446	0.5278	0.0895		-0.00013	-0.00123	-0.00003
0.0175	-0.9998	-0.0035		-0.8715	0.4831	-0.0845		0.8538	0.513	0.088		-0.00007	-0.00123	0.00000
0	-1	0		-0.8628	0.4981	-0.0863		0.8628	0.4981	0.0863		0.00000	-0.00127	0.00000
-0.0175	-0.9998	0.0035		-0.8538	0.513	-0.088		0.8715	0.4831	0.0845		0.00007	-0.00123	0.00000
-0.0349	-0.9994	0.007		-0.8446	0.5278	-0.0895		0.8799	0.4679	0.0826		0.00013	-0.00123	0.00003
-0.0523	-0.9986	0.0105		-0.8352	0.5424	-0.091		0.8881	0.4525	0.0806		0.00020	-0.00123	0.00003
-0.0697	-0.9975	0.0139		-0.8255	0.5568	-0.0923		0.896	0.437	0.0786		0.00027	-0.00123	0.00007
-0.0871	-0.996	0.0174		-0.8156	0.5711	-0.0936		0.9037	0.4214	0.0764		0.00033	-0.00117	0.00007
-0.1045	-0.9943	0.0208		-0.8054	0.5851	-0.0947		0.911	0.4056	0.0741		0.00037	-0.00120	0.00007
-0.1218	-0.9923	0.0242		-0.795	0.5991	-0.0957		0.9181	0.3897	0.0717		0.00043	-0.00117	0.00007
-0.1391	-0.9899	0.0276		-0.7843	0.6128	-0.0966		0.925	0.3737	0.0693		0.00053	-0.00113	0.00010

Raw Data														
1				2				3				Triangle Center		
x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x (cm)	y (cm)	z (cm)
-0.1564	-0.9872	0.0309		-0.7735	0.6263	-0.0974		0.9315	0.3576	0.0668		0.00053	-0.00110	0.00010
-0.1735	-0.9842	0.0342		-0.7624	0.6397	-0.098		0.9378	0.3413	0.0641		0.00063	-0.00107	0.00010
-0.1907	-0.9809	0.0374		-0.751	0.6529	-0.0985		0.9437	0.325	0.0614		0.00067	-0.00100	0.00010
-0.2077	-0.9773	0.0406		-0.7395	0.6658	-0.099		0.9494	0.3085	0.0587		0.00073	-0.00100	0.00010
-0.2247	-0.9734	0.0438		-0.7277	0.6786	-0.0993		0.9548	0.2919	0.0558		0.00080	-0.00097	0.00010
-0.2417	-0.9692	0.0469		-0.7158	0.6912	-0.0994		0.9599	0.2753	0.0529		0.00080	-0.00090	0.00013
-0.2585	-0.9647	0.0499		-0.7036	0.7036	-0.0995		0.9647	0.2585	0.0499		0.00087	-0.00087	0.00010
-0.2753	-0.9599	0.0529		-0.6912	0.7158	-0.0994		0.9692	0.2417	0.0469		0.00090	-0.00080	0.00013
-0.2919	-0.9548	0.0558		-0.6786	0.7277	-0.0993		0.9734	0.2247	0.0438		0.00097	-0.00080	0.00010
-0.3085	-0.9494	0.0587		-0.6658	0.7395	-0.099		0.9773	0.2077	0.0406		0.00100	-0.00073	0.00010
-0.325	-0.9437	0.0614		-0.6529	0.751	-0.0985		0.9809	0.1907	0.0374		0.00100	-0.00067	0.00010
-0.3413	-0.9378	0.0641		-0.6397	0.7624	-0.098		0.9842	0.1735	0.0342		0.00107	-0.00063	0.00010
-0.3576	-0.9315	0.0668		-0.6263	0.7735	-0.0974		0.9872	0.1564	0.0309		0.00110	-0.00053	0.00010
-0.3737	-0.925	0.0693		-0.6128	0.7843	-0.0966		0.9899	0.1391	0.0276		0.00113	-0.00053	0.00010
-0.3897	-0.9181	0.0717		-0.5991	0.795	-0.0957		0.9923	0.1218	0.0242		0.00117	-0.00043	0.00007
-0.4056	-0.911	0.0741		-0.5851	0.8054	-0.0947		0.9943	0.1045	0.0208		0.00120	-0.00037	0.00007
-0.4214	-0.9037	0.0764		-0.5711	0.8156	-0.0936		0.996	0.0871	0.0174		0.00117	-0.00033	0.00007
-0.437	-0.896	0.0786		-0.5568	0.8255	-0.0923		0.9975	0.0697	0.0139		0.00123	-0.00027	0.00007
-0.4525	-0.8881	0.0806		-0.5424	0.8352	-0.091		0.9986	0.0523	0.0105		0.00123	-0.00020	0.00003
-0.4679	-0.8799	0.0826		-0.5278	0.8446	-0.0895		0.9994	0.0349	0.007		0.00123	-0.00013	0.00003
-0.4831	-0.8715	0.0845		-0.513	0.8538	-0.088		0.9998	0.0175	0.0035		0.00123	-0.00007	0.00000
-0.4981	-0.8628	0.0863		-0.4981	0.8628	-0.0863		1	0	0		0.00127	0.00000	0.00000
-0.513	-0.8538	0.088		-0.4831	0.8715	-0.0845		0.9998	-0.0175	-0.0035		0.00123	0.00007	0.00000
-0.5278	-0.8446	0.0895		-0.4679	0.8799	-0.0826		0.9994	-0.0349	-0.007		0.00123	0.00013	-0.00003
-0.5424	-0.8352	0.091		-0.4525	0.8881	-0.0806		0.9986	-0.0523	-0.0105		0.00123	0.00020	-0.00003
-0.5568	-0.8255	0.0923		-0.437	0.896	-0.0786		0.9975	-0.0697	-0.0139		0.00123	0.00027	-0.00007
-0.5711	-0.8156	0.0936		-0.4214	0.9037	-0.0764		0.996	-0.0871	-0.0174		0.00117	0.00033	-0.00007

Raw Data														
1				2				3				Triangle Center		
x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x (cm)	y (cm)	z (cm)
-0.5851	-0.8054	0.0947		-0.4056	0.911	-0.0741		0.9943	-0.1045	-0.0208		0.00120	0.00037	-0.00007
-0.5991	-0.795	0.0957		-0.3897	0.9181	-0.0717		0.9923	-0.1218	-0.0242		0.00117	0.00043	-0.00007
-0.6128	-0.7843	0.0966		-0.3737	0.925	-0.0693		0.9899	-0.1391	-0.0276		0.00113	0.00053	-0.00010
-0.6263	-0.7735	0.0974		-0.3576	0.9315	-0.0668		0.9872	-0.1564	-0.0309		0.00110	0.00053	-0.00010
-0.6397	-0.7624	0.098		-0.3413	0.9378	-0.0641		0.9842	-0.1735	-0.0342		0.00107	0.00063	-0.00010
-0.6529	-0.751	0.0985		-0.325	0.9437	-0.0614		0.9809	-0.1907	-0.0374		0.00100	0.00067	-0.00010
-0.6658	-0.7395	0.099		-0.3085	0.9494	-0.0587		0.9773	-0.2077	-0.0406		0.00100	0.00073	-0.00010
-0.6786	-0.7277	0.0993		-0.2919	0.9548	-0.0558		0.9734	-0.2247	-0.0438		0.00097	0.00080	-0.00010
-0.6912	-0.7158	0.0994		-0.2753	0.9599	-0.0529		0.9692	-0.2417	-0.0469		0.00090	0.00080	-0.00013
-0.7036	-0.7036	0.0995		-0.2585	0.9647	-0.0499		0.9647	-0.2585	-0.0499		0.00087	0.00087	-0.00010
-0.7158	-0.6912	0.0994		-0.2417	0.9692	-0.0469		0.9599	-0.2753	-0.0529		0.00080	0.00090	-0.00013
-0.7277	-0.6786	0.0993		-0.2247	0.9734	-0.0438		0.9548	-0.2919	-0.0558		0.00080	0.00097	-0.00010
-0.7395	-0.6658	0.099		-0.2077	0.9773	-0.0406		0.9494	-0.3085	-0.0587		0.00073	0.00100	-0.00010
-0.751	-0.6529	0.0985		-0.1907	0.9809	-0.0374		0.9437	-0.325	-0.0614		0.00067	0.00100	-0.00010
-0.7624	-0.6397	0.098		-0.1735	0.9842	-0.0342		0.9378	-0.3413	-0.0641		0.00063	0.00107	-0.00010
-0.7735	-0.6263	0.0974		-0.1564	0.9872	-0.0309		0.9315	-0.3576	-0.0668		0.00053	0.00110	-0.00010
-0.7843	-0.6128	0.0966		-0.1391	0.9899	-0.0276		0.925	-0.3737	-0.0693		0.00053	0.00113	-0.00010
-0.795	-0.5991	0.0957		-0.1218	0.9923	-0.0242		0.9181	-0.3897	-0.0717		0.00043	0.00117	-0.00007
-0.8054	-0.5851	0.0947		-0.1045	0.9943	-0.0208		0.911	-0.4056	-0.0741		0.00037	0.00120	-0.00007
-0.8156	-0.5711	0.0936		-0.0871	0.996	-0.0174		0.9037	-0.4214	-0.0764		0.00033	0.00117	-0.00007
-0.8255	-0.5568	0.0923		-0.0697	0.9975	-0.0139		0.896	-0.437	-0.0786		0.00027	0.00123	-0.00007
-0.8352	-0.5424	0.091		-0.0523	0.9986	-0.0105		0.8881	-0.4525	-0.0806		0.00020	0.00123	-0.00003
-0.8446	-0.5278	0.0895		-0.0349	0.9994	-0.007		0.8799	-0.4679	-0.0826		0.00013	0.00123	-0.00003
-0.8538	-0.513	0.088		-0.0175	0.9998	-0.0035		0.8715	-0.4831	-0.0845		0.00007	0.00123	0.00000
-0.8628	-0.4981	0.0863		0	1	0		0.8628	-0.4981	-0.0863		0.00000	0.00127	0.00000
-0.8715	-0.4831	0.0845		0.0175	0.9998	0.0035		0.8538	-0.513	-0.088		-0.00007	0.00123	0.00000
-0.8799	-0.4679	0.0826		0.0349	0.9994	0.007		0.8446	-0.5278	-0.0895		-0.00013	0.00123	0.00003

Raw Data														
1				2				3				Triangle Center		
x' (cm)	y' (cm)	z' (cm)	x' (cm)	y' (cm)	z' (cm)	x' (cm)	y' (cm)	z' (cm)	x' (cm)	y' (cm)	z' (cm)	x (cm)	y (cm)	z (cm)
-0.8881	-0.4525	0.0806	0.0523	0.9986	0.0105	0.8352	-0.5424	-0.091	-0.00020	0.00123	0.00003			
-0.896	-0.437	0.0786	0.0697	0.9975	0.0139	0.8255	-0.5568	-0.0923	-0.00027	0.00123	0.00007			
-0.9037	-0.4214	0.0764	0.0871	0.996	0.0174	0.8156	-0.5711	-0.0936	-0.00033	0.00117	0.00007			
-0.911	-0.4056	0.0741	0.1045	0.9943	0.0208	0.8054	-0.5851	-0.0947	-0.00037	0.00120	0.00007			
-0.9181	-0.3897	0.0717	0.1218	0.9923	0.0242	0.795	-0.5991	-0.0957	-0.00043	0.00117	0.00007			
-0.925	-0.3737	0.0693	0.1391	0.9899	0.0276	0.7843	-0.6128	-0.0966	-0.00053	0.00113	0.00010			
-0.9315	-0.3576	0.0668	0.1564	0.9872	0.0309	0.7735	-0.6263	-0.0974	-0.00053	0.00110	0.00010			
-0.9378	-0.3413	0.0641	0.1735	0.9842	0.0342	0.7624	-0.6397	-0.098	-0.00063	0.00107	0.00010			
-0.9437	-0.325	0.0614	0.1907	0.9809	0.0374	0.751	-0.6529	-0.0985	-0.00067	0.00100	0.00010			
-0.9494	-0.3085	0.0587	0.2077	0.9773	0.0406	0.7395	-0.6658	-0.099	-0.00073	0.00100	0.00010			
-0.9548	-0.2919	0.0558	0.2247	0.9734	0.0438	0.7277	-0.6786	-0.0993	-0.00080	0.00097	0.00010			
-0.9599	-0.2753	0.0529	0.2417	0.9692	0.0469	0.7158	-0.6912	-0.0994	-0.00080	0.00090	0.00013			
-0.9647	-0.2585	0.0499	0.2585	0.9647	0.0499	0.7036	-0.7036	-0.0995	-0.00087	0.00087	0.00010			
-0.9692	-0.2417	0.0469	0.2753	0.9599	0.0529	0.6912	-0.7158	-0.0994	-0.00090	0.00080	0.00013			
-0.9734	-0.2247	0.0438	0.2919	0.9548	0.0558	0.6786	-0.7277	-0.0993	-0.00097	0.00080	0.00010			
-0.9773	-0.2077	0.0406	0.3085	0.9494	0.0587	0.6658	-0.7395	-0.099	-0.00100	0.00073	0.00010			
-0.9809	-0.1907	0.0374	0.325	0.9437	0.0614	0.6529	-0.751	-0.0985	-0.00100	0.00067	0.00010			
-0.9842	-0.1735	0.0342	0.3413	0.9378	0.0641	0.6397	-0.7624	-0.098	-0.00107	0.00063	0.00010			
-0.9872	-0.1564	0.0309	0.3576	0.9315	0.0668	0.6263	-0.7735	-0.0974	-0.00110	0.00053	0.00010			
-0.9899	-0.1391	0.0276	0.3737	0.925	0.0693	0.6128	-0.7843	-0.0966	-0.00113	0.00053	0.00010			
-0.9923	-0.1218	0.0242	0.3897	0.9181	0.0717	0.5991	-0.795	-0.0957	-0.00117	0.00043	0.00007			
-0.9943	-0.1045	0.0208	0.4056	0.911	0.0741	0.5851	-0.8054	-0.0947	-0.00120	0.00037	0.00007			
-0.996	-0.0871	0.0174	0.4214	0.9037	0.0764	0.5711	-0.8156	-0.0936	-0.00117	0.00033	0.00007			
-0.9975	-0.0697	0.0139	0.437	0.896	0.0786	0.5568	-0.8255	-0.0923	-0.00123	0.00027	0.00007			
-0.9986	-0.0523	0.0105	0.4525	0.8881	0.0806	0.5424	-0.8352	-0.091	-0.00123	0.00020	0.00003			
-0.9994	-0.0349	0.007	0.4679	0.8799	0.0826	0.5278	-0.8446	-0.0895	-0.00123	0.00013	0.00003			
-0.9998	-0.0175	0.0035	0.4831	0.8715	0.0845	0.513	-0.8538	-0.088	-0.00123	0.00007	0.00000			

Raw Data														
1				2				3				Triangle Center		
x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x (cm)	y (cm)	z (cm)
-1	0	0		0.4981	0.8628	0.0863		0.4981	-0.8628	-0.0863		-0.00127	0.00000	0.00000
-0.9998	0.0175	-0.0035		0.513	0.8538	0.088		0.4831	-0.8715	-0.0845		-0.00123	-0.00007	0.00000
-0.9994	0.0349	-0.007		0.5278	0.8446	0.0895		0.4679	-0.8799	-0.0826		-0.00123	-0.00013	-0.00003
-0.9986	0.0523	-0.0105		0.5424	0.8352	0.091		0.4525	-0.8881	-0.0806		-0.00123	-0.00020	-0.00003
-0.9975	0.0697	-0.0139		0.5568	0.8255	0.0923		0.437	-0.896	-0.0786		-0.00123	-0.00027	-0.00007
-0.996	0.0871	-0.0174		0.5711	0.8156	0.0936		0.4214	-0.9037	-0.0764		-0.00117	-0.00033	-0.00007
-0.9943	0.1045	-0.0208		0.5851	0.8054	0.0947		0.4056	-0.911	-0.0741		-0.00120	-0.00037	-0.00007
-0.9923	0.1218	-0.0242		0.5991	0.795	0.0957		0.3897	-0.9181	-0.0717		-0.00117	-0.00043	-0.00007
-0.9899	0.1391	-0.0276		0.6128	0.7843	0.0966		0.3737	-0.925	-0.0693		-0.00113	-0.00053	-0.00010
-0.9872	0.1564	-0.0309		0.6263	0.7735	0.0974		0.3576	-0.9315	-0.0668		-0.00110	-0.00053	-0.00010
-0.9842	0.1735	-0.0342		0.6397	0.7624	0.098		0.3413	-0.9378	-0.0641		-0.00107	-0.00063	-0.00010
-0.9809	0.1907	-0.0374		0.6529	0.751	0.0985		0.325	-0.9437	-0.0614		-0.00100	-0.00067	-0.00010
-0.9773	0.2077	-0.0406		0.6658	0.7395	0.099		0.3085	-0.9494	-0.0587		-0.00100	-0.00073	-0.00010
-0.9734	0.2247	-0.0438		0.6786	0.7277	0.0993		0.2919	-0.9548	-0.0558		-0.00097	-0.00080	-0.00010
-0.9692	0.2417	-0.0469		0.6912	0.7158	0.0994		0.2753	-0.9599	-0.0529		-0.00090	-0.00080	-0.00013
-0.9647	0.2585	-0.0499		0.7036	0.7036	0.0995		0.2585	-0.9647	-0.0499		-0.00087	-0.00087	-0.00010
-0.9599	0.2753	-0.0529		0.7158	0.6912	0.0994		0.2417	-0.9692	-0.0469		-0.00080	-0.00090	-0.00013
-0.9548	0.2919	-0.0558		0.7277	0.6786	0.0993		0.2247	-0.9734	-0.0438		-0.00080	-0.00097	-0.00010
-0.9494	0.3085	-0.0587		0.7395	0.6658	0.099		0.2077	-0.9773	-0.0406		-0.00073	-0.00100	-0.00010
-0.9437	0.325	-0.0614		0.751	0.6529	0.0985		0.1907	-0.9809	-0.0374		-0.00067	-0.00100	-0.00010
-0.9378	0.3413	-0.0641		0.7624	0.6397	0.098		0.1735	-0.9842	-0.0342		-0.00063	-0.00107	-0.00010
-0.9315	0.3576	-0.0668		0.7735	0.6263	0.0974		0.1564	-0.9872	-0.0309		-0.00053	-0.00110	-0.00010
-0.925	0.3737	-0.0693		0.7843	0.6128	0.0966		0.1391	-0.9899	-0.0276		-0.00053	-0.00113	-0.00010
-0.9181	0.3897	-0.0717		0.795	0.5991	0.0957		0.1218	-0.9923	-0.0242		-0.00043	-0.00117	-0.00007
-0.911	0.4056	-0.0741		0.8054	0.5851	0.0947		0.1045	-0.9943	-0.0208		-0.00037	-0.00120	-0.00007
-0.9037	0.4214	-0.0764		0.8156	0.5711	0.0936		0.0871	-0.996	-0.0174		-0.00033	-0.00117	-0.00007
-0.896	0.437	-0.0786		0.8255	0.5568	0.0923		0.0697	-0.9975	-0.0139		-0.00027	-0.00123	-0.00007

Raw Data														
1				2				3				Triangle Center		
x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x (cm)	y (cm)	z (cm)
-0.8881	0.4525	-0.0806		0.8352	0.5424	0.091		0.0523	-0.9986	-0.0105		-0.0020	-0.00123	-0.00003
-0.8799	0.4679	-0.0826		0.8446	0.5278	0.0895		0.0349	-0.9994	-0.007		-0.00013	-0.00123	-0.00003
-0.8715	0.4831	-0.0845		0.8538	0.513	0.088		0.0175	-0.9998	-0.0035		-0.00007	-0.00123	0.00000
-0.8628	0.4981	-0.0863		0.8628	0.4981	0.0863		0	-1	0		0.00000	-0.00127	0.00000
-0.8538	0.513	-0.088		0.8715	0.4831	0.0845		-0.0175	-0.9998	0.0035		0.00007	-0.00123	0.00000
-0.8446	0.5278	-0.0895		0.8799	0.4679	0.0826		-0.0349	-0.9994	0.007		0.00013	-0.00123	0.00003
-0.8352	0.5424	-0.091		0.8881	0.4525	0.0806		-0.0523	-0.9986	0.0105		0.00020	-0.00123	0.00003
-0.8255	0.5568	-0.0923		0.896	0.437	0.0786		-0.0697	-0.9975	0.0139		0.00027	-0.00123	0.00007
-0.8156	0.5711	-0.0936		0.9037	0.4214	0.0764		-0.0871	-0.996	0.0174		0.00033	-0.00117	0.00007
-0.8054	0.5851	-0.0947		0.911	0.4056	0.0741		-0.1045	-0.9943	0.0208		0.00037	-0.00120	0.00007
-0.795	0.5991	-0.0957		0.9181	0.3897	0.0717		-0.1218	-0.9923	0.0242		0.00043	-0.00117	0.00007
-0.7843	0.6128	-0.0966		0.925	0.3737	0.0693		-0.1391	-0.9899	0.0276		0.00053	-0.00113	0.00010
-0.7735	0.6263	-0.0974		0.9315	0.3576	0.0668		-0.1564	-0.9872	0.0309		0.00053	-0.00110	0.00010
-0.7624	0.6397	-0.098		0.9378	0.3413	0.0641		-0.1735	-0.9842	0.0342		0.00063	-0.00107	0.00010
-0.751	0.6529	-0.0985		0.9437	0.325	0.0614		-0.1907	-0.9809	0.0374		0.00067	-0.00100	0.00010
-0.7395	0.6658	-0.099		0.9494	0.3085	0.0587		-0.2077	-0.9773	0.0406		0.00073	-0.00100	0.00010
-0.7277	0.6786	-0.0993		0.9548	0.2919	0.0558		-0.2247	-0.9734	0.0438		0.00080	-0.00097	0.00010
-0.7158	0.6912	-0.0994		0.9599	0.2753	0.0529		-0.2417	-0.9692	0.0469		0.00080	-0.00090	0.00013
-0.7036	0.7036	-0.0995		0.9647	0.2585	0.0499		-0.2585	-0.9647	0.0499		0.00087	-0.00087	0.00010
-0.6912	0.7158	-0.0994		0.9692	0.2417	0.0469		-0.2753	-0.9599	0.0529		0.00090	-0.00080	0.00013
-0.6786	0.7277	-0.0993		0.9734	0.2247	0.0438		-0.2919	-0.9548	0.0558		0.00097	-0.00080	0.00010
-0.6658	0.7395	-0.099		0.9773	0.2077	0.0406		-0.3085	-0.9494	0.0587		0.00100	-0.00073	0.00010
-0.6529	0.751	-0.0985		0.9809	0.1907	0.0374		-0.325	-0.9437	0.0614		0.00100	-0.00067	0.00010
-0.6397	0.7624	-0.098		0.9842	0.1735	0.0342		-0.3413	-0.9378	0.0641		0.00107	-0.00063	0.00010
-0.6263	0.7735	-0.0974		0.9872	0.1564	0.0309		-0.3576	-0.9315	0.0668		0.00110	-0.00053	0.00010
-0.6128	0.7843	-0.0966		0.9899	0.1391	0.0276		-0.3737	-0.925	0.0693		0.00113	-0.00053	0.00010
-0.5991	0.795	-0.0957		0.9923	0.1218	0.0242		-0.3897	-0.9181	0.0717		0.00117	-0.00043	0.00007

Raw Data											
1			2			3			Triangle Center		
x' (cm)	y' (cm)	z' (cm)	x' (cm)	y' (cm)	z' (cm)	x' (cm)	y' (cm)	z' (cm)	x (cm)	y (cm)	z (cm)
-0.5851	0.8054	-0.0947	0.9943	0.1045	0.0208	-0.4056	-0.911	0.0741	0.00120	-0.00037	0.00007
-0.5711	0.8156	-0.0936	0.996	0.0871	0.0174	-0.4214	-0.9037	0.0764	0.00117	-0.00033	0.00007
-0.5568	0.8255	-0.0923	0.9975	0.0697	0.0139	-0.437	-0.896	0.0786	0.00123	-0.00027	0.00007
-0.5424	0.8352	-0.091	0.9986	0.0523	0.0105	-0.4525	-0.8881	0.0806	0.00123	-0.00020	0.00003
-0.5278	0.8446	-0.0895	0.9994	0.0349	0.007	-0.4679	-0.8799	0.0826	0.00123	-0.00013	0.00003
-0.513	0.8538	-0.088	0.9998	0.0175	0.0035	-0.4831	-0.8715	0.0845	0.00123	-0.00007	0.00000
-0.4981	0.8628	-0.0863	1	0	0	-0.4981	-0.8628	0.0863	0.00127	0.00000	0.00000
-0.4831	0.8715	-0.0845	0.9998	-0.0175	-0.0035	-0.513	-0.8538	0.088	0.00123	0.00007	0.00000
-0.4679	0.8799	-0.0826	0.9994	-0.0349	-0.007	-0.5278	-0.8446	0.0895	0.00123	0.00013	-0.00003
-0.4525	0.8881	-0.0806	0.9986	-0.0523	-0.0105	-0.5424	-0.8352	0.091	0.00123	0.00020	-0.00003
-0.437	0.896	-0.0786	0.9975	-0.0697	-0.0139	-0.5568	-0.8255	0.0923	0.00123	0.00027	-0.00007
-0.4214	0.9037	-0.0764	0.996	-0.0871	-0.0174	-0.5711	-0.8156	0.0936	0.00117	0.00033	-0.00007
-0.4056	0.911	-0.0741	0.9943	-0.1045	-0.0208	-0.5851	-0.8054	0.0947	0.00120	0.00037	-0.00007
-0.3897	0.9181	-0.0717	0.9923	-0.1218	-0.0242	-0.5991	-0.795	0.0957	0.00117	0.00043	-0.00007
-0.3737	0.925	-0.0693	0.9899	-0.1391	-0.0276	-0.6128	-0.7843	0.0966	0.00113	0.00053	-0.00010
-0.3576	0.9315	-0.0668	0.9872	-0.1564	-0.0309	-0.6263	-0.7735	0.0974	0.00110	0.00053	-0.00010
-0.3413	0.9378	-0.0641	0.9842	-0.1735	-0.0342	-0.6397	-0.7624	0.098	0.00107	0.00063	-0.00010
-0.325	0.9437	-0.0614	0.9809	-0.1907	-0.0374	-0.6529	-0.751	0.0985	0.00100	0.00067	-0.00010
-0.3085	0.9494	-0.0587	0.9773	-0.2077	-0.0406	-0.6658	-0.7395	0.099	0.00100	0.00073	-0.00010
-0.2919	0.9548	-0.0558	0.9734	-0.2247	-0.0438	-0.6786	-0.7277	0.0993	0.00097	0.00080	-0.00010
-0.2753	0.9599	-0.0529	0.9692	-0.2417	-0.0469	-0.6912	-0.7158	0.0994	0.00090	0.00080	-0.00013
-0.2585	0.9647	-0.0499	0.9647	-0.2585	-0.0499	-0.7036	-0.7036	0.0995	0.00087	0.00087	-0.00010
-0.2417	0.9692	-0.0469	0.9599	-0.2753	-0.0529	-0.7158	-0.6912	0.0994	0.00080	0.00090	-0.00013
-0.2247	0.9734	-0.0438	0.9548	-0.2919	-0.0558	-0.7277	-0.6786	0.0993	0.00080	0.00097	-0.00010
-0.2077	0.9773	-0.0406	0.9494	-0.3085	-0.0587	-0.7395	-0.6658	0.099	0.00073	0.00100	-0.00010
-0.1907	0.9809	-0.0374	0.9437	-0.325	-0.0614	-0.751	-0.6529	0.0985	0.00067	0.00100	-0.00010
-0.1735	0.9842	-0.0342	0.9378	-0.3413	-0.0641	-0.7624	-0.6397	0.098	0.00063	0.00107	-0.00010

Raw Data														
1				2				3				Triangle Center		
x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x' (cm)	y' (cm)	z' (cm)		x (cm)	y (cm)	z (cm)
-0.1564	0.9872	-0.0309		0.9315	-0.3576	-0.0668		-0.7735	-0.6263	0.0974		0.00053	0.00110	-0.00010
-0.1391	0.9899	-0.0276		0.925	-0.3737	-0.0693		-0.7843	-0.6128	0.0966		0.00053	0.00113	-0.00010
-0.1218	0.9923	-0.0242		0.9181	-0.3897	-0.0717		-0.795	-0.5991	0.0957		0.00043	0.00117	-0.00007
-0.1045	0.9943	-0.0208		0.911	-0.4056	-0.0741		-0.8054	-0.5851	0.0947		0.00037	0.00120	-0.00007
-0.0871	0.996	-0.0174		0.9037	-0.4214	-0.0764		-0.8156	-0.5711	0.0936		0.00033	0.00117	-0.00007
-0.0697	0.9975	-0.0139		0.896	-0.437	-0.0786		-0.8255	-0.5568	0.0923		0.00027	0.00123	-0.00007
-0.0523	0.9986	-0.0105		0.8881	-0.4525	-0.0806		-0.8352	-0.5424	0.091		0.00020	0.00123	-0.00003
-0.0349	0.9994	-0.007		0.8799	-0.4679	-0.0826		-0.8446	-0.5278	0.0895		0.00013	0.00123	-0.00003
-0.0175	0.9998	-0.0035		0.8715	-0.4831	-0.0845		-0.8538	-0.513	0.088		0.00007	0.00123	0.00000

Helical Axis Components													
Unit Vector			Radius Vector				Phi (deg)		Helical Axis Intersection				Error
X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	Phi (deg)	T (cm)	X (cm)	Y (cm)	Z (cm)		(cm)	
0.1933	-0.0051	-0.9811	-0.0001	0.0046	0	1.0293	0	-0.0001	0.0046	-0.0001	0.0001	0.0034	
0.1927	-0.0152	-0.9811	-0.0004	0.0046	-0.0001	1.0293	0	-0.0003	0.0046	-0.0004	0.0004	0.0034	
0.1917	-0.0252	-0.9811	-0.0006	0.0046	-0.0002	1.0293	0	-0.0006	0.0046	-0.0006	0.0006	0.0034	
0.1901	-0.0352	-0.9811	-0.0009	0.0046	-0.0003	1.0293	0	-0.0008	0.0045	-0.0008	0.0008	0.0035	
0.188	-0.0451	-0.9811	-0.0011	0.0045	-0.0004	1.0293	0	-0.001	0.0045	-0.0011	0.0011	0.0035	
0.1854	-0.0549	-0.9811	-0.0014	0.0045	-0.0005	1.0293	0	-0.0012	0.0044	-0.0013	0.0013	0.0036	
0.1822	-0.0645	-0.9811	-0.0016	0.0044	-0.0006	1.0293	0	-0.0014	0.0043	-0.0015	0.0015	0.0037	
0.1786	-0.074	-0.9811	-0.0018	0.0043	-0.0007	1.0293	0	-0.0017	0.0042	-0.0017	0.0017	0.0037	
0.1745	-0.0832	-0.9811	-0.0021	0.0042	-0.0007	1.0293	0	-0.0019	0.0041	-0.0018	0.0018	0.0038	
0.1699	-0.0922	-0.9811	-0.0023	0.0041	-0.0008	1.0293	0	-0.0021	0.004	-0.002	0.002	0.0039	
0.1648	-0.101	-0.9811	-0.0025	0.004	-0.0008	1.0293	0	-0.0023	0.0039	-0.0021	0.0021	0.0039	
0.1593	-0.1095	-0.9811	-0.0027	0.0039	-0.0009	1.0293	0	-0.0025	0.0037	-0.0022	0.0022	0.004	
0.1534	-0.1177	-0.9811	-0.0029	0.0037	-0.0009	1.0293	0	-0.0027	0.0036	-0.0023	0.0023	0.004	
0.147	-0.1256	-0.9811	-0.0031	0.0036	-0.0009	1.0293	0	-0.0029	0.0034	-0.0023	0.0023	0.004	
0.1402	-0.1331	-0.9811	-0.0033	0.0034	-0.0009	1.0293	0	-0.0031	0.0032	-0.0024	0.0024	0.0041	
0.1331	-0.1402	-0.9811	-0.0034	0.0033	-0.0009	1.0293	0	-0.0032	0.0031	-0.0024	0.0024	0.0041	
0.1256	-0.147	-0.9811	-0.0036	0.0031	-0.0009	1.0293	0	-0.0034	0.0029	-0.0023	0.0023	0.004	
0.1177	-0.1534	-0.9811	-0.0037	0.0029	-0.0009	1.0293	0	-0.0036	0.0027	-0.0023	0.0023	0.004	
0.1095	-0.1593	-0.9811	-0.0039	0.0027	-0.0009	1.0293	0	-0.0037	0.0025	-0.0022	0.0022	0.004	
0.101	-0.1648	-0.9811	-0.004	0.0025	-0.0008	1.0293	0	-0.0039	0.0023	-0.0021	0.0021	0.0039	
0.0922	-0.1699	-0.9811	-0.0041	0.0023	-0.0008	1.0293	0	-0.004	0.0021	-0.002	0.002	0.0039	
0.0832	-0.1745	-0.9811	-0.0042	0.0021	-0.0007	1.0293	0	-0.0041	0.0019	-0.0018	0.0018	0.0038	
0.074	-0.1786	-0.9811	-0.0043	0.0018	-0.0007	1.0293	0	-0.0042	0.0017	-0.0017	0.0017	0.0037	
0.0645	-0.1822	-0.9811	-0.0044	0.0016	-0.0006	1.0293	0	-0.0043	0.0014	-0.0015	0.0015	0.0037	
0.0549	-0.1854	-0.9811	-0.0045	0.0014	-0.0005	1.0293	0	-0.0044	0.0012	-0.0013	0.0013	0.0036	
0.0451	-0.188	-0.9811	-0.0045	0.0011	-0.0004	1.0293	0	-0.0045	0.001	-0.0011	0.0011	0.0035	
0.0352	-0.1901	-0.9811	-0.0046	0.0009	-0.0003	1.0293	0	-0.0045	0.0008	-0.0009	0.0009	0.0035	
0.0252	-0.1917	-0.9811	-0.0046	0.0006	-0.0002	1.0293	0	-0.0046	0.0006	-0.0006	0.0006	0.0034	
0.0152	-0.1927	-0.9811	-0.0046	0.0004	-0.0001	1.0293	0	-0.0046	0.0003	-0.0004	0.0004	0.0034	

Helical Axis Components													Helical Axis Intersection			Error	
Unit Vector				Radius Vector				Phi (deg)	T (cm)	X (cm)			Y (cm)	Z (cm)	(cm)		
X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	Phi (deg)	T (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	(cm)
0.0051	-0.1933	-0.9811	-0.0046	0.0001	0	1.0293	0	0	1.0293	0	-0.0046	0.0001	-0.0001	-0.0001	0.0001	0.0034	
-0.0051	-0.1933	-0.9811	-0.0046	-0.0001	0	1.0293	0	0	1.0293	0	-0.0046	-0.0001	0.0001	0.0001	0.0001	0.0034	
-0.0152	-0.1927	-0.9811	-0.0046	-0.0004	0.0001	1.0293	0	0	1.0293	0	-0.0046	-0.0003	0.0004	0.0004	0.0004	0.0034	
-0.0252	-0.1917	-0.9811	-0.0046	-0.0006	0.0002	1.0293	0	0	1.0293	0	-0.0046	-0.0006	0.0006	0.0006	0.0006	0.0034	
-0.0352	-0.1901	-0.9811	-0.0046	-0.0009	0.0003	1.0293	0	0	1.0293	0	-0.0045	-0.0008	0.0008	0.0008	0.0008	0.0035	
-0.0451	-0.188	-0.9811	-0.0045	-0.0011	0.0004	1.0293	0	0	1.0293	0	-0.0045	-0.001	0.0011	0.0011	0.0011	0.0035	
-0.0549	-0.1854	-0.9811	-0.0045	-0.0014	0.0005	1.0293	0	0	1.0293	0	-0.0044	-0.0012	0.0012	0.0012	0.0012	0.0036	
-0.0645	-0.1822	-0.9811	-0.0044	-0.0016	0.0006	1.0293	0	0	1.0293	0	-0.0043	-0.0014	0.0014	0.0014	0.0014	0.0037	
-0.074	-0.1786	-0.9811	-0.0043	-0.0018	0.0007	1.0293	0	0	1.0293	0	-0.0042	-0.0017	0.0017	0.0017	0.0017	0.0037	
-0.0832	-0.1745	-0.9811	-0.0042	-0.0021	0.0007	1.0293	0	0	1.0293	0	-0.0041	-0.0019	0.0019	0.0018	0.0018	0.0038	
-0.0922	-0.1699	-0.9811	-0.0041	-0.0023	0.0008	1.0293	0	0	1.0293	0	-0.004	-0.0021	0.0021	0.002	0.002	0.0039	
-0.101	-0.1648	-0.9811	-0.004	-0.0025	0.0008	1.0293	0	0	1.0293	0	-0.0039	-0.0023	0.0023	0.0021	0.0021	0.0039	
-0.1095	-0.1593	-0.9811	-0.0039	-0.0027	0.0009	1.0293	0	0	1.0293	0	-0.0037	-0.0025	0.0025	0.0022	0.0022	0.004	
-0.1177	-0.1534	-0.9811	-0.0037	-0.0029	0.0009	1.0293	0	0	1.0293	0	-0.0036	-0.0027	0.0027	0.0023	0.0023	0.004	
-0.1256	-0.147	-0.9811	-0.0036	-0.0031	0.0009	1.0293	0	0	1.0293	0	-0.0034	-0.0029	0.0029	0.0023	0.0023	0.0041	
-0.1331	-0.1402	-0.9811	-0.0034	-0.0033	0.0009	1.0293	0	0	1.0293	0	-0.0032	-0.0031	0.0031	0.0024	0.0024	0.0041	
-0.1402	-0.1331	-0.9811	-0.0033	-0.0034	0.0009	1.0293	0	0	1.0293	0	-0.0031	-0.0032	0.0032	0.0024	0.0024	0.0041	
-0.147	-0.1256	-0.9811	-0.0031	-0.0036	0.0009	1.0293	0	0	1.0293	0	-0.0029	-0.0034	0.0034	0.0023	0.0023	0.0041	
-0.1534	-0.1177	-0.9811	-0.0029	-0.0037	0.0009	1.0293	0	0	1.0293	0	-0.0027	-0.0036	0.0036	0.0023	0.0023	0.004	
-0.1593	-0.1095	-0.9811	-0.0027	-0.0039	0.0009	1.0293	0	0	1.0293	0	-0.0025	-0.0037	0.0037	0.0022	0.0022	0.004	
-0.1648	-0.101	-0.9811	-0.0025	-0.004	0.0008	1.0293	0	0	1.0293	0	-0.0023	-0.0039	0.0039	0.0021	0.0021	0.0039	
-0.1699	-0.0922	-0.9811	-0.0023	-0.0041	0.0008	1.0293	0	0	1.0293	0	-0.0021	-0.004	0.004	0.002	0.002	0.0039	
-0.1745	-0.0832	-0.9811	-0.0021	-0.0042	0.0007	1.0293	0	0	1.0293	0	-0.0019	-0.0041	0.0041	0.0018	0.0018	0.0038	
-0.1786	-0.074	-0.9811	-0.0018	-0.0043	0.0007	1.0293	0	0	1.0293	0	-0.0017	-0.0042	0.0042	0.0017	0.0017	0.0038	
-0.1822	-0.0645	-0.9811	-0.0016	-0.0044	0.0006	1.0293	0	0	1.0293	0	-0.0014	-0.0043	0.0043	0.0015	0.0015	0.0037	
-0.1854	-0.0549	-0.9811	-0.0014	-0.0045	0.0005	1.0293	0	0	1.0293	0	-0.0012	-0.0044	0.0044	0.0013	0.0013	0.0036	
-0.188	-0.0451	-0.9811	-0.0011	-0.0045	0.0004	1.0293	0	0	1.0293	0	-0.001	-0.0045	0.0045	0.0011	0.0011	0.0035	
-0.1901	-0.0352	-0.9811	-0.0009	-0.0046	0.0003	1.0293	0	0	1.0293	0	-0.0008	-0.0045	0.0045	0.0009	0.0009	0.0035	
-0.1917	-0.0252	-0.9811	-0.0006	-0.0046	0.0002	1.0293	0	0	1.0293	0	-0.0006	-0.0046	0.0046	0.0006	0.0006	0.0034	
-0.1927	-0.0152	-0.9811	-0.0004	-0.0046	0.0001	1.0293	0	0	1.0293	0	-0.0003	-0.0046	0.0046	0.0004	0.0004	0.0034	

Helical Axis Components													Helical Axis Intersection			Error
Unit Vector				Radius Vector												
X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	Phi (deg)	T (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	(cm)		
-0.1933	-0.0051	-0.9811	-0.0001	-0.0046	0	1.0293	0	-0.0001	-0.0046	0.0001	0.0001	-0.0046	0.0001	0.0034		
-0.1933	0.0051	-0.9811	0.0001	-0.0046	0	1.0293	0	0.0001	-0.0046	-0.0001	0.0001	-0.0046	-0.0001	0.0034		
-0.1927	0.0152	-0.9811	0.0004	-0.0046	-0.0001	1.0293	0	0.0003	-0.0046	-0.0004	0.0003	-0.0046	-0.0004	0.0034		
-0.1917	0.0252	-0.9811	0.0006	-0.0046	-0.0002	1.0293	0	0.0006	-0.0046	-0.0006	0.0006	-0.0046	-0.0006	0.0034		
-0.1901	0.0352	-0.9811	0.0009	-0.0046	-0.0003	1.0293	0	0.0008	-0.0045	-0.0008	0.0008	-0.0045	-0.0008	0.0035		
-0.188	0.0451	-0.9811	0.0011	-0.0045	-0.0004	1.0293	0	0.001	-0.0045	-0.001	0.001	-0.0045	-0.001	0.0035		
-0.1854	0.0549	-0.9811	0.0014	-0.0045	-0.0005	1.0293	0	0.0012	-0.0044	-0.0012	0.0012	-0.0044	-0.0012	0.0036		
-0.1822	0.0645	-0.9811	0.0016	-0.0044	-0.0006	1.0293	0	0.0014	-0.0043	-0.0014	0.0014	-0.0043	-0.0014	0.0037		
-0.1786	0.074	-0.9811	0.0018	-0.0043	-0.0007	1.0293	0	0.0017	-0.0042	-0.0017	0.0017	-0.0042	-0.0017	0.0037		
-0.1745	0.0832	-0.9811	0.0021	-0.0042	-0.0007	1.0293	0	0.0019	-0.0041	-0.0019	0.0019	-0.0041	-0.0019	0.0038		
-0.1699	0.0922	-0.9811	0.0023	-0.0041	-0.0008	1.0293	0	0.0021	-0.004	-0.0021	0.0021	-0.004	-0.0021	0.0039		
-0.1648	0.101	-0.9811	0.0025	-0.004	-0.0008	1.0293	0	0.0023	-0.0039	-0.0023	0.0023	-0.0039	-0.0023	0.0039		
-0.1593	0.1095	-0.9811	0.0027	-0.0039	-0.0009	1.0293	0	0.0025	-0.0037	-0.0025	0.0025	-0.0037	-0.0025	0.004		
-0.1534	0.1177	-0.9811	0.0029	-0.0037	-0.0009	1.0293	0	0.0027	-0.0036	-0.0027	0.0027	-0.0036	-0.0027	0.004		
-0.147	0.1256	-0.9811	0.0031	-0.0036	-0.0009	1.0293	0	0.0029	-0.0034	-0.0029	0.0029	-0.0034	-0.0029	0.004		
-0.1402	0.1331	-0.9811	0.0033	-0.0034	-0.0009	1.0293	0	0.0031	-0.0032	-0.0031	0.0031	-0.0032	-0.0031	0.0041		
-0.1331	0.1402	-0.9811	0.0034	-0.0033	-0.0009	1.0293	0	0.0032	-0.0031	-0.0032	0.0032	-0.0031	-0.0032	0.0041		
-0.1256	0.147	-0.9811	0.0036	-0.0031	-0.0009	1.0293	0	0.0034	-0.0029	-0.0034	0.0034	-0.0029	-0.0034	0.004		
-0.1177	0.1534	-0.9811	0.0037	-0.0029	-0.0009	1.0293	0	0.0036	-0.0027	-0.0036	0.0036	-0.0027	-0.0036	0.004		
-0.1095	0.1593	-0.9811	0.0039	-0.0027	-0.0009	1.0293	0	0.0037	-0.0025	-0.0037	0.0037	-0.0025	-0.0037	0.004		
-0.101	0.1648	-0.9811	0.004	-0.0025	-0.0008	1.0293	0	0.0039	-0.0023	-0.0039	0.0039	-0.0023	-0.0039	0.0039		
-0.0922	0.1699	-0.9811	0.0041	-0.0023	-0.0008	1.0293	0	0.004	-0.0021	-0.004	0.004	-0.0021	-0.004	0.0039		
-0.0832	0.1745	-0.9811	0.0042	-0.0021	-0.0007	1.0293	0	0.0041	-0.0019	-0.0041	0.0041	-0.0019	-0.0041	0.0038		
-0.074	0.1786	-0.9811	0.0043	-0.0018	-0.0007	1.0293	0	0.0042	-0.0017	-0.0042	0.0042	-0.0017	-0.0042	0.0037		
-0.0645	0.1822	-0.9811	0.0044	-0.0016	-0.0006	1.0293	0	0.0043	-0.0014	-0.0043	0.0043	-0.0014	-0.0043	0.0037		
-0.0549	0.1854	-0.9811	0.0045	-0.0014	-0.0005	1.0293	0	0.0044	-0.0012	-0.0044	0.0044	-0.0012	-0.0044	0.0036		
-0.0451	0.188	-0.9811	0.0045	-0.0011	-0.0004	1.0293	0	0.0045	-0.001	-0.0045	0.0045	-0.001	-0.0045	0.0035		
-0.0352	0.1901	-0.9811	0.0046	-0.0009	-0.0003	1.0293	0	0.0045	-0.0008	-0.0045	0.0045	-0.0008	-0.0045	0.0035		
-0.0252	0.1917	-0.9811	0.0046	-0.0006	-0.0002	1.0293	0	0.0046	-0.0006	-0.0046	0.0046	-0.0006	-0.0046	0.0034		
-0.0152	0.1927	-0.9811	0.0046	-0.0004	-0.0001	1.0293	0	0.0046	-0.0003	-0.0046	0.0046	-0.0003	-0.0046	0.0034		

Helical Axis Components													Helical Axis Intersection			Error
Unit Vector				Radius Vector												
X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	Phi (deg)	T (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	(cm)		
-0.0051	0.1933	-0.9811	0.0046	-0.0001	0	1.0293	0	0.0046	-0.0001	-0.0001	0.0046	-0.0001	-0.0001	0.0034		
0.0051	0.1933	-0.9811	0.0046	0.0001	0	1.0293	0	0.0046	0.0001	0.0001	0.0046	0.0001	0.0001	0.0034		
0.0152	0.1927	-0.9811	0.0046	0.0004	0.0001	1.0293	0	0.0046	0.0003	0.0004	0.0046	0.0003	0.0004	0.0034		
0.0252	0.1917	-0.9811	0.0046	0.0006	0.0002	1.0293	0	0.0046	0.0006	0.0006	0.0046	0.0006	0.0006	0.0034		
0.0352	0.1901	-0.9811	0.0046	0.0009	0.0003	1.0293	0	0.0045	0.0008	0.0008	0.0045	0.0008	0.0008	0.0035		
0.0451	0.188	-0.9811	0.0045	0.0011	0.0004	1.0293	0	0.0045	0.001	0.001	0.0045	0.001	0.001	0.0035		
0.0549	0.1854	-0.9811	0.0045	0.0014	0.0005	1.0293	0	0.0044	0.0012	0.0012	0.0044	0.0012	0.0012	0.0036		
0.0645	0.1822	-0.9811	0.0044	0.0016	0.0006	1.0293	0	0.0043	0.0014	0.0014	0.0043	0.0014	0.0014	0.0037		
0.074	0.1786	-0.9811	0.0043	0.0018	0.0007	1.0293	0	0.0042	0.0017	0.0017	0.0042	0.0017	0.0017	0.0037		
0.0832	0.1745	-0.9811	0.0042	0.0021	0.0007	1.0293	0	0.0041	0.0019	0.0019	0.0041	0.0019	0.0018	0.0038		
0.0922	0.1699	-0.9811	0.0041	0.0023	0.0008	1.0293	0	0.004	0.0021	0.0021	0.004	0.0021	0.002	0.0039		
0.101	0.1648	-0.9811	0.004	0.0025	0.0008	1.0293	0	0.0039	0.0023	0.0023	0.0039	0.0023	0.0021	0.0039		
0.1095	0.1593	-0.9811	0.0039	0.0027	0.0009	1.0293	0	0.0037	0.0025	0.0025	0.0037	0.0025	0.0022	0.004		
0.1177	0.1534	-0.9811	0.0037	0.0029	0.0009	1.0293	0	0.0036	0.0027	0.0027	0.0036	0.0027	0.0023	0.004		
0.1256	0.147	-0.9811	0.0036	0.0031	0.0009	1.0293	0	0.0034	0.0029	0.0029	0.0034	0.0029	0.0023	0.0041		
0.1331	0.1402	-0.9811	0.0034	0.0033	0.0009	1.0293	0	0.0032	0.0031	0.0031	0.0032	0.0031	0.0024	0.0041		
0.1402	0.1331	-0.9811	0.0033	0.0034	0.0009	1.0293	0	0.0031	0.0032	0.0032	0.0031	0.0032	0.0024	0.0041		
0.147	0.1256	-0.9811	0.0031	0.0036	0.0009	1.0293	0	0.0029	0.0034	0.0034	0.0029	0.0034	0.0023	0.0041		
0.1534	0.1177	-0.9811	0.0029	0.0037	0.0009	1.0293	0	0.0027	0.0036	0.0036	0.0027	0.0036	0.0023	0.004		
0.1593	0.1095	-0.9811	0.0027	0.0039	0.0009	1.0293	0	0.0025	0.0037	0.0037	0.0025	0.0037	0.0022	0.004		
0.1648	0.101	-0.9811	0.0025	0.004	0.0008	1.0293	0	0.0023	0.0039	0.0039	0.0023	0.0039	0.0021	0.0039		
0.1699	0.0922	-0.9811	0.0023	0.0041	0.0008	1.0293	0	0.0021	0.004	0.004	0.0021	0.004	0.002	0.0039		
0.1745	0.0832	-0.9811	0.0021	0.0042	0.0007	1.0293	0	0.0019	0.0041	0.0041	0.0019	0.0041	0.0018	0.0038		
0.1786	0.074	-0.9811	0.0018	0.0043	0.0007	1.0293	0	0.0017	0.0042	0.0042	0.0017	0.0042	0.0017	0.0038		
0.1822	0.0645	-0.9811	0.0016	0.0044	0.0006	1.0293	0	0.0014	0.0043	0.0043	0.0014	0.0043	0.0015	0.0037		
0.1854	0.0549	-0.9811	0.0014	0.0045	0.0005	1.0293	0	0.0012	0.0044	0.0044	0.0012	0.0044	0.0013	0.0036		
0.188	0.0451	-0.9811	0.0011	0.0045	0.0004	1.0293	0	0.001	0.0045	0.0045	0.001	0.0045	0.0011	0.0035		
0.1901	0.0352	-0.9811	0.0009	0.0046	0.0003	1.0293	0	0.0008	0.0045	0.0045	0.0008	0.0045	0.0009	0.0035		
0.1917	0.0252	-0.9811	0.0006	0.0046	0.0002	1.0293	0	0.0006	0.0046	0.0046	0.0006	0.0046	0.0006	0.0034		
0.1927	0.0152	-0.9811	0.0004	0.0046	0.0001	1.0293	0	0.0003	0.0046	0.0046	0.0003	0.0046	0.0004	0.0034		

Helical Axis Components													Helical Axis Intersection			Error
Unit Vector				Radius Vector				Phi (deg)	T (cm)	X (cm)	Y (cm)	Z (cm)	Error			
X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	Z (cm)	Phi (deg)	T (cm)	X (cm)	Y (cm)	Z (cm)	Error				
0.1933	0.0051	-0.9811	0.0001	0.0046	0	0	1.0293	0	0.0001	0.0046	0.0001	0.0034				
0.1933	-0.0051	-0.9811	-0.0001	0.0046	0	0	1.0293	0	-0.0001	0.0046	-0.0001	0.0034				
0.1927	-0.0152	-0.9811	-0.0004	0.0046	-0.0001	0	1.0293	0	-0.0003	0.0046	-0.0004	0.0034				
0.1917	-0.0252	-0.9811	-0.0006	0.0046	-0.0002	0	1.0293	0	-0.0006	0.0046	-0.0006	0.0034				
0.1901	-0.0352	-0.9811	-0.0009	0.0046	-0.0003	0	1.0293	0	-0.0008	0.0045	-0.0008	0.0035				
0.188	-0.0451	-0.9811	-0.0011	0.0045	-0.0004	0	1.0293	0	-0.001	0.0045	-0.0011	0.0035				
0.1854	-0.0549	-0.9811	-0.0014	0.0045	-0.0005	0	1.0293	0	-0.0012	0.0044	-0.0013	0.0036				
0.1822	-0.0645	-0.9811	-0.0016	0.0044	-0.0006	0	1.0293	0	-0.0014	0.0043	-0.0015	0.0037				
0.1786	-0.074	-0.9811	-0.0018	0.0043	-0.0007	0	1.0293	0	-0.0017	0.0042	-0.0017	0.0037				
0.1745	-0.0832	-0.9811	-0.0021	0.0042	-0.0007	0	1.0293	0	-0.0019	0.0041	-0.0018	0.0038				
0.1699	-0.0922	-0.9811	-0.0023	0.0041	-0.0008	0	1.0293	0	-0.0021	0.004	-0.002	0.0039				
0.1648	-0.101	-0.9811	-0.0025	0.004	-0.0008	0	1.0293	0	-0.0023	0.0039	-0.0021	0.0039				
0.1593	-0.1095	-0.9811	-0.0027	0.0039	-0.0009	0	1.0293	0	-0.0025	0.0037	-0.0022	0.004				
0.1534	-0.1177	-0.9811	-0.0029	0.0037	-0.0009	0	1.0293	0	-0.0027	0.0036	-0.0023	0.004				
0.147	-0.1256	-0.9811	-0.0031	0.0036	-0.0009	0	1.0293	0	-0.0029	0.0034	-0.0023	0.004				
0.1402	-0.1331	-0.9811	-0.0033	0.0034	-0.0009	0	1.0293	0	-0.0031	0.0032	-0.0024	0.0041				
0.1331	-0.1402	-0.9811	-0.0034	0.0033	-0.0009	0	1.0293	0	-0.0032	0.0031	-0.0024	0.0041				
0.1256	-0.147	-0.9811	-0.0036	0.0031	-0.0009	0	1.0293	0	-0.0034	0.0029	-0.0023	0.004				
0.1177	-0.1534	-0.9811	-0.0037	0.0029	-0.0009	0	1.0293	0	-0.0036	0.0027	-0.0023	0.004				
0.1095	-0.1593	-0.9811	-0.0039	0.0027	-0.0009	0	1.0293	0	-0.0037	0.0025	-0.0022	0.004				
0.101	-0.1648	-0.9811	-0.004	0.0025	-0.0008	0	1.0293	0	-0.0039	0.0023	-0.0021	0.0039				
0.0922	-0.1699	-0.9811	-0.0041	0.0023	-0.0008	0	1.0293	0	-0.004	0.0021	-0.002	0.0039				
0.0832	-0.1745	-0.9811	-0.0042	0.0021	-0.0007	0	1.0293	0	-0.0041	0.0019	-0.0018	0.0038				
0.074	-0.1786	-0.9811	-0.0043	0.0018	-0.0007	0	1.0293	0	-0.0042	0.0017	-0.0017	0.0037				
0.0645	-0.1822	-0.9811	-0.0044	0.0016	-0.0006	0	1.0293	0	-0.0043	0.0014	-0.0015	0.0037				
0.0549	-0.1854	-0.9811	-0.0045	0.0014	-0.0005	0	1.0293	0	-0.0044	0.0012	-0.0013	0.0036				
0.0451	-0.188	-0.9811	-0.0045	0.0011	-0.0004	0	1.0293	0	-0.0045	0.001	-0.0011	0.0035				
0.0352	-0.1901	-0.9811	-0.0046	0.0009	-0.0003	0	1.0293	0	-0.0045	0.0008	-0.0009	0.0035				
0.0252	-0.1917	-0.9811	-0.0046	0.0006	-0.0002	0	1.0293	0	-0.0046	0.0006	-0.0006	0.0034				
0.0152	-0.1927	-0.9811	-0.0046	0.0004	-0.0001	0	1.0293	0	-0.0046	0.0003	-0.0004	0.0034				

Helical Axis Components													Helical Axis Intersection			Error
Unit Vector				Radius Vector				Phi (deg)	T (cm)							
X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	(cm)	
0.0051	-0.1933	-0.9811	-0.0046	0.0001	0	1.0293	0	0	-0.0046	0.0001	-0.0001	0.0001	0.0001	-0.0001	0.0034	
-0.0051	-0.1933	-0.9811	-0.0046	-0.0001	0	1.0293	0	0	-0.0046	-0.0001	0.0001	-0.0001	-0.0001	0.0001	0.0034	
-0.0152	-0.1927	-0.9811	-0.0046	-0.0004	0.0001	1.0293	0	0	-0.0046	-0.0003	0.0004	0.0004	-0.0003	0.0004	0.0034	
-0.0252	-0.1917	-0.9811	-0.0046	-0.0006	0.0002	1.0293	0	0	-0.0046	-0.0006	0.0006	0.0006	-0.0006	0.0006	0.0034	
-0.0352	-0.1901	-0.9811	-0.0046	-0.0009	0.0003	1.0293	0	0	-0.0045	-0.0008	0.0008	0.0008	-0.0008	0.0008	0.0035	
-0.0451	-0.188	-0.9811	-0.0045	-0.0011	0.0004	1.0293	0	0	-0.0045	-0.001	0.001	0.0011	-0.001	0.0011	0.0035	
-0.0549	-0.1854	-0.9811	-0.0045	-0.0014	0.0005	1.0293	0	0	-0.0044	-0.0012	0.0012	0.0013	-0.0012	0.0013	0.0036	
-0.0645	-0.1822	-0.9811	-0.0044	-0.0016	0.0006	1.0293	0	0	-0.0043	-0.0014	0.0014	0.0015	-0.0014	0.0015	0.0037	
-0.074	-0.1786	-0.9811	-0.0043	-0.0018	0.0007	1.0293	0	0	-0.0042	-0.0017	0.0017	0.0017	-0.0017	0.0017	0.0037	
-0.0832	-0.1745	-0.9811	-0.0042	-0.0021	0.0007	1.0293	0	0	-0.0041	-0.0019	0.0019	0.0018	-0.0019	0.0018	0.0038	
-0.0922	-0.1699	-0.9811	-0.0041	-0.0023	0.0008	1.0293	0	0	-0.004	-0.0021	0.0021	0.002	-0.0021	0.002	0.0039	
-0.101	-0.1648	-0.9811	-0.004	-0.0025	0.0008	1.0293	0	0	-0.0039	-0.0023	0.0023	0.0021	-0.0023	0.0021	0.0039	
-0.1095	-0.1593	-0.9811	-0.0039	-0.0027	0.0009	1.0293	0	0	-0.0037	-0.0025	0.0025	0.0022	-0.0025	0.0022	0.004	
-0.1177	-0.1534	-0.9811	-0.0037	-0.0029	0.0009	1.0293	0	0	-0.0036	-0.0027	0.0027	0.0023	-0.0027	0.0023	0.004	
-0.1256	-0.147	-0.9811	-0.0036	-0.0031	0.0009	1.0293	0	0	-0.0034	-0.0029	0.0029	0.0023	-0.0029	0.0023	0.0041	
-0.1331	-0.1402	-0.9811	-0.0034	-0.0033	0.0009	1.0293	0	0	-0.0032	-0.0031	0.0031	0.0024	-0.0031	0.0024	0.0041	
-0.1402	-0.1331	-0.9811	-0.0033	-0.0034	0.0009	1.0293	0	0	-0.0031	-0.0032	0.0032	0.0024	-0.0032	0.0024	0.0041	
-0.147	-0.1256	-0.9811	-0.0031	-0.0036	0.0009	1.0293	0	0	-0.0029	-0.0034	0.0034	0.0023	-0.0034	0.0023	0.0041	
-0.1534	-0.1177	-0.9811	-0.0029	-0.0037	0.0009	1.0293	0	0	-0.0027	-0.0036	0.0036	0.0023	-0.0036	0.0023	0.004	
-0.1593	-0.1095	-0.9811	-0.0027	-0.0039	0.0009	1.0293	0	0	-0.0025	-0.0037	0.0037	0.0022	-0.0037	0.0022	0.004	
-0.1648	-0.101	-0.9811	-0.0025	-0.004	0.0008	1.0293	0	0	-0.0023	-0.0039	0.0039	0.0021	-0.0039	0.0021	0.0039	
-0.1699	-0.0922	-0.9811	-0.0023	-0.0041	0.0008	1.0293	0	0	-0.0021	-0.004	0.004	0.002	-0.004	0.002	0.0039	
-0.1745	-0.0832	-0.9811	-0.0021	-0.0042	0.0007	1.0293	0	0	-0.0019	-0.0041	0.0041	0.0018	-0.0041	0.0018	0.0038	
-0.1786	-0.074	-0.9811	-0.0018	-0.0043	0.0007	1.0293	0	0	-0.0017	-0.0042	0.0042	0.0017	-0.0042	0.0017	0.0038	
-0.1822	-0.0645	-0.9811	-0.0016	-0.0044	0.0006	1.0293	0	0	-0.0014	-0.0043	0.0043	0.0015	-0.0043	0.0015	0.0037	
-0.1854	-0.0549	-0.9811	-0.0014	-0.0045	0.0005	1.0293	0	0	-0.0012	-0.0044	0.0044	0.0013	-0.0044	0.0013	0.0036	
-0.188	-0.0451	-0.9811	-0.0011	-0.0045	0.0004	1.0293	0	0	-0.001	-0.0045	0.0045	0.0011	-0.0045	0.0011	0.0035	
-0.1901	-0.0352	-0.9811	-0.0009	-0.0046	0.0003	1.0293	0	0	-0.0008	-0.0045	0.0045	0.0009	-0.0045	0.0009	0.0035	
-0.1917	-0.0252	-0.9811	-0.0006	-0.0046	0.0002	1.0293	0	0	-0.0006	-0.0046	0.0046	0.0006	-0.0046	0.0006	0.0034	
-0.1927	-0.0152	-0.9811	-0.0004	-0.0046	0.0001	1.0293	0	0	-0.0004	-0.0046	0.0046	0.0004	-0.0046	0.0004	0.0034	

Helical Axis Components													Helical Axis Intersection			Error								
Unit Vector				Radius Vector				Phi (deg)					T (cm)			X (cm)			Y (cm)			Z (cm)		
X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	Phi (deg)	T (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	Error	
-0.1933	-0.0051	-0.9811	-0.0001	-0.0046	0	1.0293	0	0	1.0293	0	-0.0001	-0.0046	0.0001	0.0001	-0.0046	0.0001	0.0001	-0.0046	0.0001	0.0001	-0.0046	0.0001	0.0034	
-0.1933	0.0051	-0.9811	0.0001	-0.0046	0	1.0293	0	0	1.0293	0	0.0001	-0.0046	-0.0001	0.0001	-0.0046	-0.0001	0.0001	-0.0046	-0.0001	0.0001	-0.0046	-0.0001	0.0034	
-0.1927	0.0152	-0.9811	0.0004	-0.0046	-0.0001	1.0293	0	0	1.0293	0	0.0003	-0.0046	-0.0004	0.0003	-0.0046	-0.0004	0.0003	-0.0046	-0.0004	0.0003	-0.0046	-0.0004	0.0034	
-0.1917	0.0252	-0.9811	0.0006	-0.0046	-0.0002	1.0293	0	0	1.0293	0	0.0006	-0.0046	-0.0006	0.0006	-0.0046	-0.0006	0.0006	-0.0046	-0.0006	0.0006	-0.0046	-0.0006	0.0034	
-0.1901	0.0352	-0.9811	0.0009	-0.0046	-0.0003	1.0293	0	0	1.0293	0	0.0008	-0.0045	-0.0008	0.0008	-0.0045	-0.0008	0.0008	-0.0045	-0.0008	0.0008	-0.0045	-0.0008	0.0035	
-0.188	0.0451	-0.9811	0.0011	-0.0045	-0.0004	1.0293	0	0	1.0293	0	0.0011	-0.0045	-0.0011	0.0011	-0.0045	-0.0011	0.0011	-0.0045	-0.0011	0.0011	-0.0045	-0.0011	0.0035	
-0.1854	0.0549	-0.9811	0.0014	-0.0045	-0.0005	1.0293	0	0	1.0293	0	0.0014	-0.0044	-0.0014	0.0014	-0.0044	-0.0014	0.0014	-0.0044	-0.0014	0.0014	-0.0044	-0.0014	0.0036	
-0.1822	0.0645	-0.9811	0.0016	-0.0044	-0.0006	1.0293	0	0	1.0293	0	0.0016	-0.0044	-0.0016	0.0016	-0.0044	-0.0016	0.0016	-0.0044	-0.0016	0.0016	-0.0044	-0.0016	0.0037	
-0.1786	0.074	-0.9811	0.0018	-0.0043	-0.0007	1.0293	0	0	1.0293	0	0.0018	-0.0043	-0.0018	0.0018	-0.0043	-0.0018	0.0018	-0.0043	-0.0018	0.0018	-0.0043	-0.0018	0.0037	
-0.1745	0.0832	-0.9811	0.0021	-0.0042	-0.0007	1.0293	0	0	1.0293	0	0.0021	-0.0042	-0.0021	0.0021	-0.0042	-0.0021	0.0021	-0.0042	-0.0021	0.0021	-0.0042	-0.0021	0.0038	
-0.1699	0.0922	-0.9811	0.0023	-0.0041	-0.0008	1.0293	0	0	1.0293	0	0.0023	-0.0041	-0.0023	0.0023	-0.0041	-0.0023	0.0023	-0.0041	-0.0023	0.0023	-0.0041	-0.0023	0.0039	
-0.1648	0.101	-0.9811	0.0025	-0.004	-0.0008	1.0293	0	0	1.0293	0	0.0025	-0.004	-0.0025	0.0025	-0.004	-0.0025	0.0025	-0.004	-0.0025	0.0025	-0.004	-0.0025	0.0039	
-0.1593	0.1095	-0.9811	0.0027	-0.0039	-0.0009	1.0293	0	0	1.0293	0	0.0027	-0.0039	-0.0027	0.0027	-0.0039	-0.0027	0.0027	-0.0039	-0.0027	0.0027	-0.0039	-0.0027	0.004	
-0.1534	0.1177	-0.9811	0.0029	-0.0037	-0.0009	1.0293	0	0	1.0293	0	0.0029	-0.0037	-0.0029	0.0029	-0.0037	-0.0029	0.0029	-0.0037	-0.0029	0.0029	-0.0037	-0.0029	0.004	
-0.147	0.1256	-0.9811	0.0031	-0.0036	-0.0009	1.0293	0	0	1.0293	0	0.0031	-0.0036	-0.0031	0.0031	-0.0036	-0.0031	0.0031	-0.0036	-0.0031	0.0031	-0.0036	-0.0031	0.004	
-0.1402	0.1331	-0.9811	0.0033	-0.0034	-0.0009	1.0293	0	0	1.0293	0	0.0033	-0.0034	-0.0033	0.0033	-0.0034	-0.0033	0.0033	-0.0034	-0.0033	0.0033	-0.0034	-0.0033	0.0041	
-0.1331	0.1402	-0.9811	0.0034	-0.0033	-0.0009	1.0293	0	0	1.0293	0	0.0034	-0.0033	-0.0034	0.0034	-0.0033	-0.0034	0.0034	-0.0033	-0.0034	0.0034	-0.0033	-0.0034	0.0041	
-0.1256	0.147	-0.9811	0.0036	-0.0031	-0.0009	1.0293	0	0	1.0293	0	0.0036	-0.0031	-0.0036	0.0036	-0.0031	-0.0036	0.0036	-0.0031	-0.0036	0.0036	-0.0031	-0.0036	0.004	
-0.1177	0.1534	-0.9811	0.0037	-0.0029	-0.0009	1.0293	0	0	1.0293	0	0.0037	-0.0029	-0.0037	0.0037	-0.0029	-0.0037	0.0037	-0.0029	-0.0037	0.0037	-0.0029	-0.0037	0.004	
-0.1095	0.1593	-0.9811	0.0039	-0.0027	-0.0009	1.0293	0	0	1.0293	0	0.0039	-0.0027	-0.0039	0.0039	-0.0027	-0.0039	0.0039	-0.0027	-0.0039	0.0039	-0.0027	-0.0039	0.004	
-0.101	0.1648	-0.9811	0.004	-0.0025	-0.0008	1.0293	0	0	1.0293	0	0.004	-0.0025	-0.004	0.004	-0.0025	-0.004	0.004	-0.0025	-0.004	0.004	-0.0025	-0.004	0.0039	
-0.0922	0.1699	-0.9811	0.0041	-0.0023	-0.0008	1.0293	0	0	1.0293	0	0.0041	-0.0023	-0.0041	0.0041	-0.0023	-0.0041	0.0041	-0.0023	-0.0041	0.0041	-0.0023	-0.0041	0.0039	
-0.0832	0.1745	-0.9811	0.0042	-0.0021	-0.0007	1.0293	0	0	1.0293	0	0.0042	-0.0021	-0.0042	0.0042	-0.0021	-0.0042	0.0042	-0.0021	-0.0042	0.0042	-0.0021	-0.0042	0.0038	
-0.074	0.1786	-0.9811	0.0043	-0.0018	-0.0007	1.0293	0	0	1.0293	0	0.0043	-0.0018	-0.0043	0.0043	-0.0018	-0.0043	0.0043	-0.0018	-0.0043	0.0043	-0.0018	-0.0043	0.0037	
-0.0645	0.1822	-0.9811	0.0044	-0.0016	-0.0006	1.0293	0	0	1.0293	0	0.0044	-0.0016	-0.0044	0.0044	-0.0016	-0.0044	0.0044	-0.0016	-0.0044	0.0044	-0.0016	-0.0044	0.0037	
-0.0549	0.1854	-0.9811	0.0045	-0.0014	-0.0005	1.0293	0	0	1.0293	0	0.0045	-0.0014	-0.0045	0.0045	-0.0014	-0.0045	0.0045	-0.0014	-0.0045	0.0045	-0.0014	-0.0045	0.0036	
-0.0451	0.188	-0.9811	0.0045	-0.0011	-0.0004	1.0293	0	0	1.0293	0	0.0045	-0.0011	-0.0045	0.0045	-0.0011	-0.0045	0.0045	-0.0011	-0.0045	0.0045	-0.0011	-0.0045	0.0035	
-0.0352	0.1901	-0.9811	0.0046	-0.0009	-0.0003	1.0293	0	0	1.0293	0	0.0046	-0.0009	-0.0046	0.0046	-0.0009	-0.0046	0.0046	-0.0009	-0.0046	0.0046	-0.0009	-0.0046	0.0035	
-0.0252	0.1917	-0.9811	0.0046	-0.0006	-0.0002	1.0293	0	0	1.0293	0	0.0046	-0.0006	-0.0046	0.0046	-0.0006	-0.0046	0.0046	-0.0006	-0.0046	0.0046	-0.0006	-0.0046	0.0034	
-0.0152	0.1927	-0.9811	0.0046	-0.0004	-0.0001	1.0293	0	0	1.0293	0	0.0046	-0.0004	-0.0046	0.0046	-0.0004	-0.0046	0.0046	-0.0004	-0.0046	0.0046	-0.0004	-0.0046	0.0034	

Helical Axis Components													Helical Axis Intersection			Error
Unit Vector				Radius Vector												
X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	Phi (deg)	T (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	(cm)		
-0.0051	0.1933	-0.9811	0.0046	-0.0001	0	1.0293	0	0.0046	-0.0001	-0.0001	0.0046	-0.0001	-0.0001	0.0034		
0.0051	0.1933	-0.9811	0.0046	0.0001	0	1.0293	0	0.0046	0.0001	0.0001	0.0046	0.0001	0.0001	0.0034		
0.0152	0.1927	-0.9811	0.0046	0.0004	0.0001	1.0293	0	0.0046	0.0003	0.0004	0.0046	0.0003	0.0004	0.0034		
0.0252	0.1917	-0.9811	0.0046	0.0006	0.0002	1.0293	0	0.0046	0.0006	0.0006	0.0046	0.0006	0.0006	0.0034		
0.0352	0.1901	-0.9811	0.0046	0.0009	0.0003	1.0293	0	0.0045	0.0008	0.0008	0.0045	0.0008	0.0008	0.0035		
0.0451	0.188	-0.9811	0.0045	0.0011	0.0004	1.0293	0	0.0045	0.001	0.001	0.0045	0.001	0.001	0.0035		
0.0549	0.1854	-0.9811	0.0045	0.0014	0.0005	1.0293	0	0.0044	0.0012	0.0012	0.0044	0.0012	0.0012	0.0036		
0.0645	0.1822	-0.9811	0.0044	0.0016	0.0006	1.0293	0	0.0043	0.0014	0.0014	0.0043	0.0014	0.0014	0.0037		
0.074	0.1786	-0.9811	0.0043	0.0018	0.0007	1.0293	0	0.0042	0.0017	0.0017	0.0042	0.0017	0.0017	0.0037		
0.0832	0.1745	-0.9811	0.0042	0.0021	0.0007	1.0293	0	0.0041	0.0019	0.0019	0.0041	0.0019	0.0018	0.0038		
0.0922	0.1699	-0.9811	0.0041	0.0023	0.0008	1.0293	0	0.004	0.0021	0.0021	0.004	0.0021	0.002	0.0039		
0.101	0.1648	-0.9811	0.004	0.0025	0.0008	1.0293	0	0.0039	0.0023	0.0023	0.0039	0.0023	0.0021	0.0039		
0.1095	0.1593	-0.9811	0.0039	0.0027	0.0009	1.0293	0	0.0037	0.0025	0.0025	0.0037	0.0025	0.0022	0.004		
0.1177	0.1534	-0.9811	0.0037	0.0029	0.0009	1.0293	0	0.0036	0.0027	0.0027	0.0036	0.0027	0.0023	0.004		
0.1256	0.147	-0.9811	0.0036	0.0031	0.0009	1.0293	0	0.0034	0.0029	0.0029	0.0034	0.0029	0.0023	0.0041		
0.1331	0.1402	-0.9811	0.0034	0.0033	0.0009	1.0293	0	0.0032	0.0031	0.0031	0.0032	0.0031	0.0024	0.0041		
0.1402	0.1331	-0.9811	0.0033	0.0034	0.0009	1.0293	0	0.0031	0.0032	0.0032	0.0031	0.0032	0.0024	0.0041		
0.147	0.1256	-0.9811	0.0031	0.0036	0.0009	1.0293	0	0.0029	0.0034	0.0034	0.0029	0.0034	0.0023	0.0041		
0.1534	0.1177	-0.9811	0.0029	0.0037	0.0009	1.0293	0	0.0027	0.0036	0.0036	0.0027	0.0036	0.0023	0.004		
0.1593	0.1095	-0.9811	0.0027	0.0039	0.0009	1.0293	0	0.0025	0.0037	0.0037	0.0025	0.0037	0.0022	0.004		
0.1648	0.101	-0.9811	0.0025	0.004	0.0008	1.0293	0	0.0023	0.0039	0.0039	0.0023	0.0039	0.0021	0.0039		
0.1699	0.0922	-0.9811	0.0023	0.0041	0.0008	1.0293	0	0.0021	0.004	0.004	0.0021	0.004	0.002	0.0039		
0.1745	0.0832	-0.9811	0.0021	0.0042	0.0007	1.0293	0	0.0019	0.0041	0.0041	0.0019	0.0041	0.0018	0.0038		
0.1786	0.074	-0.9811	0.0018	0.0043	0.0007	1.0293	0	0.0017	0.0042	0.0042	0.0017	0.0042	0.0017	0.0038		
0.1822	0.0645	-0.9811	0.0016	0.0044	0.0006	1.0293	0	0.0014	0.0043	0.0043	0.0014	0.0043	0.0015	0.0037		
0.1854	0.0549	-0.9811	0.0014	0.0045	0.0005	1.0293	0	0.0012	0.0044	0.0044	0.0012	0.0044	0.0013	0.0036		
0.188	0.0451	-0.9811	0.0011	0.0045	0.0004	1.0293	0	0.001	0.0045	0.0045	0.001	0.0045	0.0011	0.0035		
0.1901	0.0352	-0.9811	0.0009	0.0046	0.0003	1.0293	0	0.0008	0.0045	0.0045	0.0008	0.0045	0.0009	0.0035		
0.1917	0.0252	-0.9811	0.0006	0.0046	0.0002	1.0293	0	0.0006	0.0046	0.0046	0.0006	0.0046	0.0006	0.0034		
0.1927	0.0152	-0.9811	0.0004	0.0046	0.0001	1.0293	0	0.0004	0.0046	0.0046	0.0004	0.0046	0.0004	0.0034		

Helical Axis Components													Helical Axis Intersection			Error
Unit Vector				Radius Vector				Phi (deg)	T (cm)							
X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	(cm)	
0.1933	0.0051	-0.9811	0.0001	0.0046	0	0	0	1.0293	0	0.0001	0.0046	0.0001	0.0046	0.0001	0.0034	
0.1933	-0.0051	-0.9811	-0.0001	0.0046	0	0	0	1.0293	0	-0.0001	0.0046	-0.0001	0.0046	-0.0001	0.0034	
0.1927	-0.0152	-0.9811	-0.0004	0.0046	-0.0001	0	0	1.0293	0	-0.0003	0.0046	-0.0004	0.0046	-0.0004	0.0034	
0.1917	-0.0252	-0.9811	-0.0006	0.0046	-0.0002	0	0	1.0293	0	-0.0006	0.0046	-0.0006	0.0046	-0.0006	0.0034	
0.1901	-0.0352	-0.9811	-0.0009	0.0046	-0.0003	0	0	1.0293	0	-0.0008	0.0045	-0.0008	0.0045	-0.0008	0.0035	
0.188	-0.0451	-0.9811	-0.0011	0.0045	-0.0004	0	0	1.0293	0	-0.001	0.0045	-0.001	0.0045	-0.001	0.0035	
0.1854	-0.0549	-0.9811	-0.0014	0.0045	-0.0005	0	0	1.0293	0	-0.0012	0.0044	-0.0012	0.0044	-0.0012	0.0036	
0.1822	-0.0645	-0.9811	-0.0016	0.0044	-0.0006	0	0	1.0293	0	-0.0014	0.0043	-0.0014	0.0043	-0.0014	0.0037	
0.1786	-0.074	-0.9811	-0.0018	0.0043	-0.0007	0	0	1.0293	0	-0.0017	0.0042	-0.0017	0.0042	-0.0017	0.0037	
0.1745	-0.0832	-0.9811	-0.0021	0.0042	-0.0007	0	0	1.0293	0	-0.0019	0.0041	-0.0019	0.0041	-0.0018	0.0038	
0.1699	-0.0922	-0.9811	-0.0023	0.0041	-0.0008	0	0	1.0293	0	-0.0021	0.004	-0.0021	0.004	-0.002	0.0039	
0.1648	-0.101	-0.9811	-0.0025	0.004	-0.0008	0	0	1.0293	0	-0.0023	0.0039	-0.0023	0.0039	-0.0021	0.0039	
0.1593	-0.1095	-0.9811	-0.0027	0.0039	-0.0009	0	0	1.0293	0	-0.0025	0.0037	-0.0025	0.0037	-0.0022	0.004	
0.1534	-0.1177	-0.9811	-0.0029	0.0037	-0.0009	0	0	1.0293	0	-0.0027	0.0036	-0.0027	0.0036	-0.0023	0.004	
0.147	-0.1256	-0.9811	-0.0031	0.0036	-0.0009	0	0	1.0293	0	-0.0029	0.0034	-0.0029	0.0034	-0.0023	0.004	
0.1402	-0.1331	-0.9811	-0.0033	0.0034	-0.0009	0	0	1.0293	0	-0.0031	0.0032	-0.0031	0.0032	-0.0024	0.0041	
0.1331	-0.1402	-0.9811	-0.0034	0.0033	-0.0009	0	0	1.0293	0	-0.0032	0.0031	-0.0032	0.0031	-0.0024	0.0041	
0.1256	-0.147	-0.9811	-0.0036	0.0031	-0.0009	0	0	1.0293	0	-0.0034	0.0029	-0.0034	0.0029	-0.0023	0.004	
0.1177	-0.1534	-0.9811	-0.0037	0.0029	-0.0009	0	0	1.0293	0	-0.0036	0.0027	-0.0036	0.0027	-0.0023	0.004	
0.1095	-0.1593	-0.9811	-0.0039	0.0027	-0.0009	0	0	1.0293	0	-0.0037	0.0025	-0.0037	0.0025	-0.0022	0.004	
0.101	-0.1648	-0.9811	-0.004	0.0025	-0.0008	0	0	1.0293	0	-0.0039	0.0023	-0.0039	0.0023	-0.0021	0.0039	
0.0922	-0.1699	-0.9811	-0.0041	0.0023	-0.0008	0	0	1.0293	0	-0.004	0.0021	-0.004	0.0021	-0.002	0.0039	
0.0832	-0.1745	-0.9811	-0.0042	0.0021	-0.0007	0	0	1.0293	0	-0.0041	0.0019	-0.0041	0.0019	-0.0018	0.0038	
0.074	-0.1786	-0.9811	-0.0043	0.0018	-0.0007	0	0	1.0293	0	-0.0042	0.0017	-0.0042	0.0017	-0.0017	0.0037	
0.0645	-0.1822	-0.9811	-0.0044	0.0016	-0.0006	0	0	1.0293	0	-0.0043	0.0014	-0.0043	0.0014	-0.0015	0.0037	
0.0549	-0.1854	-0.9811	-0.0045	0.0014	-0.0005	0	0	1.0293	0	-0.0044	0.0012	-0.0044	0.0012	-0.0013	0.0036	
0.0451	-0.188	-0.9811	-0.0045	0.0011	-0.0004	0	0	1.0293	0	-0.0045	0.001	-0.0045	0.001	-0.0011	0.0035	
0.0352	-0.1901	-0.9811	-0.0046	0.0009	-0.0003	0	0	1.0293	0	-0.0045	0.0008	-0.0045	0.0008	-0.0009	0.0035	
0.0252	-0.1917	-0.9811	-0.0046	0.0006	-0.0002	0	0	1.0293	0	-0.0046	0.0006	-0.0046	0.0006	-0.0006	0.0034	
0.0152	-0.1927	-0.9811	-0.0046	0.0004	-0.0001	0	0	1.0293	0	-0.0046	0.0003	-0.0046	0.0003	-0.0004	0.0034	

Helical Axis Components													Helical Axis Intersection			Error
Unit Vector				Radius Vector				Phi (deg)	T (cm)	Helical Axis Intersection			Error			
X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	(cm)				
0.0051	-0.1933	-0.9811	-0.0046	0.0001	0	1.0293	0	-0.0001	0.0001	-0.0001	0.0034					
-0.0051	-0.1933	-0.9811	-0.0046	-0.0001	0	1.0293	0	-0.0046	-0.0001	0.0001	0.0034					
-0.0152	-0.1927	-0.9811	-0.0046	-0.0004	0.0001	1.0293	0	-0.0046	-0.0003	0.0004	0.0034					
-0.0252	-0.1917	-0.9811	-0.0046	-0.0006	0.0002	1.0293	0	-0.0046	-0.0006	0.0006	0.0034					
-0.0352	-0.1901	-0.9811	-0.0046	-0.0009	0.0003	1.0293	0	-0.0045	-0.0008	0.0008	0.0035					
-0.0451	-0.188	-0.9811	-0.0045	-0.0011	0.0004	1.0293	0	-0.0045	-0.001	0.0011	0.0035					
-0.0549	-0.1854	-0.9811	-0.0045	-0.0014	0.0005	1.0293	0	-0.0044	-0.0012	0.0013	0.0036					
-0.0645	-0.1822	-0.9811	-0.0044	-0.0016	0.0006	1.0293	0	-0.0043	-0.0014	0.0015	0.0037					
-0.074	-0.1786	-0.9811	-0.0043	-0.0018	0.0007	1.0293	0	-0.0042	-0.0017	0.0017	0.0037					
-0.0832	-0.1745	-0.9811	-0.0042	-0.0021	0.0007	1.0293	0	-0.0041	-0.0019	0.0018	0.0038					
-0.0922	-0.1699	-0.9811	-0.0041	-0.0023	0.0008	1.0293	0	-0.004	-0.0021	0.002	0.0039					
-0.101	-0.1648	-0.9811	-0.004	-0.0025	0.0008	1.0293	0	-0.0039	-0.0023	0.0021	0.0039					
-0.1095	-0.1593	-0.9811	-0.0039	-0.0027	0.0009	1.0293	0	-0.0037	-0.0025	0.0022	0.004					
-0.1177	-0.1534	-0.9811	-0.0037	-0.0029	0.0009	1.0293	0	-0.0036	-0.0027	0.0023	0.004					
-0.1256	-0.147	-0.9811	-0.0036	-0.0031	0.0009	1.0293	0	-0.0034	-0.0029	0.0023	0.0041					
-0.1331	-0.1402	-0.9811	-0.0034	-0.0033	0.0009	1.0293	0	-0.0032	-0.0031	0.0024	0.0041					
-0.1402	-0.1331	-0.9811	-0.0033	-0.0034	0.0009	1.0293	0	-0.0031	-0.0032	0.0024	0.0041					
-0.147	-0.1256	-0.9811	-0.0031	-0.0036	0.0009	1.0293	0	-0.0029	-0.0034	0.0023	0.0041					
-0.1534	-0.1177	-0.9811	-0.0029	-0.0037	0.0009	1.0293	0	-0.0027	-0.0036	0.0023	0.004					
-0.1593	-0.1095	-0.9811	-0.0027	-0.0039	0.0009	1.0293	0	-0.0025	-0.0037	0.0022	0.004					
-0.1648	-0.101	-0.9811	-0.0025	-0.004	0.0008	1.0293	0	-0.0023	-0.0039	0.0021	0.0039					
-0.1699	-0.0922	-0.9811	-0.0023	-0.0041	0.0008	1.0293	0	-0.0021	-0.004	0.002	0.0039					
-0.1745	-0.0832	-0.9811	-0.0021	-0.0042	0.0007	1.0293	0	-0.0019	-0.0041	0.0018	0.0038					
-0.1786	-0.074	-0.9811	-0.0018	-0.0043	0.0007	1.0293	0	-0.0017	-0.0042	0.0017	0.0038					
-0.1822	-0.0645	-0.9811	-0.0016	-0.0044	0.0006	1.0293	0	-0.0014	-0.0043	0.0015	0.0037					
-0.1854	-0.0549	-0.9811	-0.0014	-0.0045	0.0005	1.0293	0	-0.0012	-0.0044	0.0013	0.0036					
-0.188	-0.0451	-0.9811	-0.0011	-0.0045	0.0004	1.0293	0	-0.001	-0.0045	0.0011	0.0035					
-0.1901	-0.0352	-0.9811	-0.0009	-0.0046	0.0003	1.0293	0	-0.0008	-0.0045	0.0009	0.0035					
-0.1917	-0.0252	-0.9811	-0.0006	-0.0046	0.0002	1.0293	0	-0.0006	-0.0046	0.0006	0.0034					
-0.1927	-0.0152	-0.9811	-0.0004	-0.0046	0.0001	1.0293	0	-0.0003	-0.0046	0.0004	0.0034					

Helical Axis Components													Helical Axis Intersection			Error								
Unit Vector				Radius Vector				Phi (deg)					T (cm)			X (cm)			Y (cm)			Z (cm)		
X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	Phi (deg)	T (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	Error	
-0.1933	-0.0051	-0.9811	-0.0001	-0.0046	0	1.0293	0	0.0001	-0.0046	0.0001	0.0001	-0.0046	0.0001	0.0001	-0.0046	0.0001	0.0001	-0.0046	0.0001	0.0001	-0.0046	0.0001	0.0034	
-0.1933	0.0051	-0.9811	0.0001	-0.0046	0	1.0293	0	0.0001	-0.0046	0.0001	0.0001	-0.0046	0.0001	0.0001	-0.0046	0.0001	0.0001	-0.0046	0.0001	0.0001	-0.0046	0.0001	0.0034	
-0.1927	0.0152	-0.9811	0.0004	-0.0046	-0.0001	1.0293	0	0.0003	-0.0046	-0.0004	0.0003	-0.0046	-0.0004	0.0003	-0.0046	-0.0004	0.0003	-0.0046	-0.0004	0.0003	-0.0046	-0.0004	0.0034	
-0.1917	0.0252	-0.9811	0.0006	-0.0046	-0.0002	1.0293	0	0.0006	-0.0046	-0.0002	1.0293	0	0.0006	-0.0046	-0.0002	1.0293	0	0.0006	-0.0046	-0.0002	1.0293	0	0.0034	
-0.1901	0.0352	-0.9811	0.0009	-0.0046	-0.0003	1.0293	0	0.0009	-0.0046	-0.0003	1.0293	0	0.0008	-0.0045	-0.0008	1.0293	0	0.0008	-0.0045	-0.0008	1.0293	0	0.0035	
-0.188	0.0451	-0.9811	0.0011	-0.0045	-0.0004	1.0293	0	0.0011	-0.0045	-0.0004	1.0293	0	0.001	-0.0045	-0.0011	1.0293	0	0.001	-0.0045	-0.0011	1.0293	0	0.0035	
-0.1854	0.0549	-0.9811	0.0014	-0.0045	-0.0005	1.0293	0	0.0014	-0.0045	-0.0005	1.0293	0	0.0012	-0.0044	-0.0013	1.0293	0	0.0012	-0.0044	-0.0013	1.0293	0	0.0036	
-0.1822	0.0645	-0.9811	0.0016	-0.0044	-0.0006	1.0293	0	0.0016	-0.0044	-0.0006	1.0293	0	0.0014	-0.0043	-0.0015	1.0293	0	0.0014	-0.0043	-0.0015	1.0293	0	0.0037	
-0.1786	0.074	-0.9811	0.0018	-0.0043	-0.0007	1.0293	0	0.0018	-0.0043	-0.0007	1.0293	0	0.0017	-0.0042	-0.0017	1.0293	0	0.0017	-0.0042	-0.0017	1.0293	0	0.0037	
-0.1745	0.0832	-0.9811	0.0021	-0.0042	-0.0007	1.0293	0	0.0021	-0.0042	-0.0007	1.0293	0	0.0019	-0.0041	-0.0018	1.0293	0	0.0019	-0.0041	-0.0018	1.0293	0	0.0038	
-0.1699	0.0922	-0.9811	0.0023	-0.0041	-0.0008	1.0293	0	0.0023	-0.0041	-0.0008	1.0293	0	0.0021	-0.004	-0.002	1.0293	0	0.0021	-0.004	-0.002	1.0293	0	0.0039	
-0.1648	0.101	-0.9811	0.0025	-0.004	-0.0008	1.0293	0	0.0025	-0.004	-0.0008	1.0293	0	0.0023	-0.0039	-0.0021	1.0293	0	0.0023	-0.0039	-0.0021	1.0293	0	0.0039	
-0.1593	0.1095	-0.9811	0.0027	-0.0039	-0.0009	1.0293	0	0.0027	-0.0039	-0.0009	1.0293	0	0.0025	-0.0037	-0.0022	1.0293	0	0.0025	-0.0037	-0.0022	1.0293	0	0.004	
-0.1534	0.1177	-0.9811	0.0029	-0.0037	-0.0009	1.0293	0	0.0029	-0.0037	-0.0009	1.0293	0	0.0027	-0.0036	-0.0023	1.0293	0	0.0027	-0.0036	-0.0023	1.0293	0	0.004	
-0.147	0.1256	-0.9811	0.0031	-0.0036	-0.0009	1.0293	0	0.0031	-0.0036	-0.0009	1.0293	0	0.0029	-0.0034	-0.0023	1.0293	0	0.0029	-0.0034	-0.0023	1.0293	0	0.004	
-0.1402	0.1331	-0.9811	0.0033	-0.0034	-0.0009	1.0293	0	0.0033	-0.0034	-0.0009	1.0293	0	0.0031	-0.0032	-0.0024	1.0293	0	0.0031	-0.0032	-0.0024	1.0293	0	0.0041	
-0.1331	0.1402	-0.9811	0.0034	-0.0033	-0.0009	1.0293	0	0.0034	-0.0033	-0.0009	1.0293	0	0.0032	-0.0031	-0.0024	1.0293	0	0.0032	-0.0031	-0.0024	1.0293	0	0.0041	
-0.1256	0.147	-0.9811	0.0036	-0.0031	-0.0009	1.0293	0	0.0036	-0.0031	-0.0009	1.0293	0	0.0034	-0.0029	-0.0023	1.0293	0	0.0034	-0.0029	-0.0023	1.0293	0	0.004	
-0.1177	0.1534	-0.9811	0.0037	-0.0029	-0.0009	1.0293	0	0.0037	-0.0029	-0.0009	1.0293	0	0.0036	-0.0027	-0.0023	1.0293	0	0.0036	-0.0027	-0.0023	1.0293	0	0.004	
-0.1095	0.1593	-0.9811	0.0039	-0.0027	-0.0009	1.0293	0	0.0039	-0.0027	-0.0009	1.0293	0	0.0037	-0.0025	-0.0022	1.0293	0	0.0037	-0.0025	-0.0022	1.0293	0	0.004	
-0.101	0.1648	-0.9811	0.004	-0.0025	-0.0008	1.0293	0	0.004	-0.0025	-0.0008	1.0293	0	0.0039	-0.0023	-0.0021	1.0293	0	0.0039	-0.0023	-0.0021	1.0293	0	0.0039	
-0.0922	0.1699	-0.9811	0.0041	-0.0023	-0.0008	1.0293	0	0.0041	-0.0023	-0.0008	1.0293	0	0.004	-0.0021	-0.002	1.0293	0	0.004	-0.0021	-0.002	1.0293	0	0.0039	
-0.0832	0.1745	-0.9811	0.0042	-0.0021	-0.0007	1.0293	0	0.0042	-0.0021	-0.0007	1.0293	0	0.0041	-0.0019	-0.0018	1.0293	0	0.0041	-0.0019	-0.0018	1.0293	0	0.0038	
-0.074	0.1786	-0.9811	0.0043	-0.0018	-0.0007	1.0293	0	0.0043	-0.0018	-0.0007	1.0293	0	0.0042	-0.0017	-0.0017	1.0293	0	0.0042	-0.0017	-0.0017	1.0293	0	0.0037	
-0.0645	0.1822	-0.9811	0.0044	-0.0016	-0.0006	1.0293	0	0.0044	-0.0016	-0.0006	1.0293	0	0.0043	-0.0014	-0.0015	1.0293	0	0.0043	-0.0014	-0.0015	1.0293	0	0.0037	
-0.0549	0.1854	-0.9811	0.0045	-0.0014	-0.0005	1.0293	0	0.0045	-0.0014	-0.0005	1.0293	0	0.0044	-0.0012	-0.0013	1.0293	0	0.0044	-0.0012	-0.0013	1.0293	0	0.0036	
-0.0451	0.188	-0.9811	0.0045	-0.0011	-0.0004	1.0293	0	0.0045	-0.0011	-0.0004	1.0293	0	0.0045	-0.001	-0.0011	1.0293	0	0.0045	-0.001	-0.0011	1.0293	0	0.0035	
-0.0352	0.1901	-0.9811	0.0046	-0.0009	-0.0003	1.0293	0	0.0046	-0.0009	-0.0003	1.0293	0	0.0045	-0.0008	-0.0009	1.0293	0	0.0045	-0.0008	-0.0009	1.0293	0	0.0035	
-0.0252	0.1917	-0.9811	0.0046	-0.0006	-0.0002	1.0293	0	0.0046	-0.0006	-0.0002	1.0293	0	0.0046	-0.0006	-0.0006	1.0293	0	0.0046	-0.0006	-0.0006	1.0293	0	0.0034	
-0.0152	0.1927	-0.9811	0.0046	-0.0004	-0.0001	1.0293	0	0.0046	-0.0004	-0.0001	1.0293	0	0.0046	-0.0003	-0.0004	1.0293	0	0.0046	-0.0003	-0.0004	1.0293	0	0.0034	

Helical Axis Components													Helical Axis Intersection			Error
Unit Vector				Radius Vector												
X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	Phi (deg)	T (cm)	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)	(cm)		
-0.0051	0.1933	-0.9811	0.0046	-0.0001	0	1.0293	0	0.0046	-0.0001	-0.0001	0.0046	-0.0001	-0.0001	0.0034		
0.0051	0.1933	-0.9811	0.0046	0.0001	0	1.0293	0	0.0046	0.0001	0.0001	0.0046	0.0001	0.0001	0.0034		
0.0152	0.1927	-0.9811	0.0046	0.0004	0.0001	1.0293	0	0.0046	0.0003	0.0004	0.0046	0.0003	0.0004	0.0034		
0.0252	0.1917	-0.9811	0.0046	0.0006	0.0002	1.0293	0	0.0046	0.0006	0.0006	0.0046	0.0006	0.0006	0.0034		
0.0352	0.1901	-0.9811	0.0046	0.0009	0.0003	1.0293	0	0.0045	0.0008	0.0008	0.0045	0.0008	0.0008	0.0035		
0.0451	0.188	-0.9811	0.0045	0.0011	0.0004	1.0293	0	0.0045	0.001	0.001	0.0045	0.001	0.001	0.0035		
0.0549	0.1854	-0.9811	0.0045	0.0014	0.0005	1.0293	0	0.0044	0.0012	0.0012	0.0044	0.0012	0.0012	0.0036		
0.0645	0.1822	-0.9811	0.0044	0.0016	0.0006	1.0293	0	0.0043	0.0014	0.0014	0.0043	0.0014	0.0014	0.0037		
0.074	0.1786	-0.9811	0.0043	0.0018	0.0007	1.0293	0	0.0042	0.0017	0.0017	0.0042	0.0017	0.0017	0.0037		
0.0832	0.1745	-0.9811	0.0042	0.0021	0.0007	1.0293	0	0.0041	0.0019	0.0019	0.0041	0.0019	0.0018	0.0038		
0.0922	0.1699	-0.9811	0.0041	0.0023	0.0008	1.0293	0	0.004	0.0021	0.0021	0.004	0.0021	0.002	0.0039		
0.101	0.1648	-0.9811	0.004	0.0025	0.0008	1.0293	0	0.0039	0.0023	0.0023	0.0039	0.0023	0.0021	0.0039		
0.1095	0.1593	-0.9811	0.0039	0.0027	0.0009	1.0293	0	0.0037	0.0025	0.0025	0.0037	0.0025	0.0022	0.004		
0.1177	0.1534	-0.9811	0.0037	0.0029	0.0009	1.0293	0	0.0036	0.0027	0.0027	0.0036	0.0027	0.0023	0.004		
0.1256	0.147	-0.9811	0.0036	0.0031	0.0009	1.0293	0	0.0034	0.0029	0.0029	0.0034	0.0029	0.0023	0.0041		
0.1331	0.1402	-0.9811	0.0034	0.0033	0.0009	1.0293	0	0.0032	0.0031	0.0031	0.0032	0.0031	0.0024	0.0041		
0.1402	0.1331	-0.9811	0.0033	0.0034	0.0009	1.0293	0	0.0031	0.0032	0.0032	0.0031	0.0032	0.0024	0.0041		
0.147	0.1256	-0.9811	0.0031	0.0036	0.0009	1.0293	0	0.0029	0.0034	0.0034	0.0029	0.0034	0.0023	0.0041		
0.1534	0.1177	-0.9811	0.0029	0.0037	0.0009	1.0293	0	0.0027	0.0036	0.0036	0.0027	0.0036	0.0023	0.004		
0.1593	0.1095	-0.9811	0.0027	0.0039	0.0009	1.0293	0	0.0025	0.0037	0.0037	0.0025	0.0037	0.0022	0.004		
0.1648	0.101	-0.9811	0.0025	0.004	0.0008	1.0293	0	0.0023	0.0039	0.0039	0.0023	0.0039	0.0021	0.0039		
0.1699	0.0922	-0.9811	0.0023	0.0041	0.0008	1.0293	0	0.0021	0.004	0.004	0.0021	0.004	0.002	0.0039		
0.1745	0.0832	-0.9811	0.0021	0.0042	0.0007	1.0293	0	0.0019	0.0041	0.0041	0.0019	0.0041	0.0018	0.0038		
0.1786	0.074	-0.9811	0.0018	0.0043	0.0007	1.0293	0	0.0017	0.0042	0.0042	0.0017	0.0042	0.0017	0.0038		
0.1822	0.0645	-0.9811	0.0016	0.0044	0.0006	1.0293	0	0.0014	0.0043	0.0043	0.0014	0.0043	0.0015	0.0037		
0.1854	0.0549	-0.9811	0.0014	0.0045	0.0005	1.0293	0	0.0012	0.0044	0.0044	0.0012	0.0044	0.0013	0.0036		
0.188	0.0451	-0.9811	0.0011	0.0045	0.0004	1.0293	0	0.001	0.0045	0.0045	0.001	0.0045	0.0011	0.0035		
0.1901	0.0352	-0.9811	0.0009	0.0046	0.0003	1.0293	0	0.0008	0.0045	0.0045	0.0008	0.0045	0.0009	0.0035		
0.1917	0.0252	-0.9811	0.0006	0.0046	0.0002	1.0293	0	0.0006	0.0046	0.0046	0.0006	0.0046	0.0006	0.0034		
0.1927	0.0152	-0.9811	0.0004	0.0046	0.0001	1.0293	0	0.0003	0.0046	0.0046	0.0003	0.0046	0.0004	0.0034		

APPENDIX D

Program used to determine the helical axis, helical axis intersection, and angle helical axis makes with the plane for mathematical (triangle) model

```
% HA3.m
% Helical Axis from triangle model

% These calculations are taken from Spoors 1980 "Rigid Body Motion
% Calculated from Spatial Co-ordinates of Markers"

clear
close all
clf

file=input('What is the file name? ', 's');
sheet=input('What is the sheet name? ', 's');
data=xlsread(file, sheet, 'c5:k364');

flag1=0;%Graphing the calculating helical axis
flag2=0;%Graphing the intersection
flag3=0;%Graphing individual components of HA
flag4=0;%Make movie of plot
fig=1;

%frame=data(:,1);

%-----
%Landmark Data to Use
%-----
x10=data(:,1); y10=data(:,2); z10=data(:,3);
x11=data(:,4); y11=data(:,5); z11=data(:,6);
x12=data(:,7); y12=data(:,8); z12=data(:,9);

x=[x10 x11 x12];%All the X landmark data
y=[y10 y11 y12];%All the Y landmark data
z=[z10 z11 z12];%All the Z landmark data
%Average of the X,Y,Z landmark data for each point in time
avgx=mean(x,2);
avgy=mean(y,2);
avgz=mean(z,2);
average=[avgx avgy avgz];
```

```

%X,Y,Z Data of the Landmark
l0=[x10 y10 z10];
l1=[x11 y11 z11];
l2=[x12 y12 z12];
I=eye(3);
len=length(x10);
%Counters
c=1;
em=1;
count=1;
d=1;
aviobj=avifile('example6.avi')
for ii=1:len-1
    a=ii;
    p=ii+1;
    acol=iscollinear(l0(a,:), l1(a,:), l2(a,:));
    pcol=iscollinear(l0(p,:), l1(p,:), l2(p,:));
    if acol==1
        msg=['*** acol (' num2str(a) ') the points are collinear'];
        disp(msg);
        continue
    elseif pcol==1
        msg=['*** pcol (' num2str(p) ') the points are collinear'];
        disp(msg);
        continue
    end
    %Make a 3x3 Matrix that will be manipulated to become the rotation
    %matrix used in the rest of the calculations. These steps make up
    %equation (5) in Spoors paper
    m0=l0(p,:)'*l0(a,:);
    m1=l1(p,:)'*l1(a,:);
    m2=l2(p,:)'*l2(a,:);
    summ=(m0+m1+m2)/3;
    m8=average(p,:)'*average(a,:);
    M=summ-m8;
    %Function sommer comes from the subroutine MATPP3 by H.J. Sommer.
    [R, S1]=sommer(M);
    RMV('R',R,a);
    %Translation Vector from equation (10) in Spoors
    v=average(p,:)'-R*average(a,:)';
    %-----
    %Solving for angle of rotation about the helical axis (phi)
    %-----
    %Spoors Eq 32
    sinphi=0.5*sqrt((R(3,2)-R(2,3))^2+(R(1,3)-R(3,1))^2+...
        (R(2,1)-R(1,2))^2);
    %Spoors Eq 34
    cosphi=0.5*(trace(R)-1);
    %If statements from Spoors; different methods are preferred a
    %different times because the arc sine function
    if sinphi<=1/2*sqrt(2)
        phi=asin(sinphi);
        if cosphi<0
            phi=pi-phi;

```



```

    end
else
    phi=acos(cosphi);
end
%-----
%Solving for Unit vector (n)
%-----
if phi<=3/4*pi && phi~=0
    %Spoors 31
    n=(.5*[R(3,2)-R(2,3); R(1,3)-R(3,1); R(2,1)-R(1,2)])/sin(phi);
else
    %Spoors 35 and 36
    b=0.5*(R+R')-cos(phi)*I;
    blen=[sqrt(b(:,1)'*b(:,1));
          sqrt(b(:,2)'*b(:,2));
          sqrt(b(:,3)'*b(:,3))];
    [Blen In]=max(blen);
    n=b(:,In)/Blen;
    if sin(phi)<0
        n=-n;
    end
end
%-----
%Solving for Translation Component (t)
%-----
%Spoors 37
t=n'*v;
%-----
%Solving for Radius Vector (s)
%-----
denom=1-cos(phi); %denominator portion of radius vector
%If denom equals 0 making it a really small number so it will give a
%actual value and not NaN
if denom==0
    denom=0.0000001;
    count=count+1;
end
%Spoors 38
s=-0.5*cross(n,cross(n,v))+sin(phi)/(2*(denom))*cross(n,v);
%-----
%If Phi==0 no defined helical axis and therefore n,s, and t aren't
%unique, these equations are used to give non-unique helical axis
%values
%-----
if phi==0 && v~= [0;0;0]
    t1=v'*v;
    t=sqrt(t1);
    n=(1./t)*v;
    s=[0; 0; 0];
    em=em+1;
end
%-----
%Save components of helical axis
%-----

```

```

N(c,1)=n(1,1); N(c,2)=n(2,1); N(c,3)=n(3,1);
PHI(c,1)=phi;
S(c,1)=s(1,1); S(c,2)=s(2,1); S(c,3)=s(3,1);
T(c,1)=t;
V(c,1)=v(1,1); V(c,2)=v(2,1); V(c,3)=v(3,1);
%-----
%Checks to ensure the unit vector is a unit vector and ensure unit
%vector and radius vector are orthogonal
%-----
npn=n'*n;
Unit(c,1)=npn;
nps=n'*s;
Orth(c,1)=nps;
%-----
%Used in the graph of the helical axis
%-----
Tri=[x10(a) y10(a) z10(a);
      x11(a) y11(a) z11(a);
      x12(a) y12(a) z12(a);
      x10(a) y10(a) z10(a)];
h=(x10(a)+x11(a)+x12(a))/3;
k=(y10(a)+y11(a)+y12(a))/3;
l=(z10(a)+z11(a)+z12(a))/3;
if flag1==1
    NS=vertcat(n',s');
    figure(1)
    hold on
    plot3(NS(:,1), NS(:,2), NS(:,3), ':k')
    scatter3(s(1), s(2), s(3), 'sg')
    scatter3(n(1), n(2), n(3), 'sr')
    plot3(Tri(:,1), Tri(:,2), Tri(:,3), 'b')
    scatter3(h,k,l, 'FaceColor', 'b')
    grid on
    xlabel('x (cm)', 'fontsize', 16, 'fontweight', 'b')
    ylabel('y (cm)', 'fontsize', 16, 'fontweight', 'b')
    zlabel('z (cm)', 'fontsize', 16, 'fontweight', 'b')
    title('Helical Axis and Triangle', 'fontsize', 24, 'fontweight',
'b')
    legend('Helical Axis', 'Radius Vector Endpoint', 'Unit Vector
Endpoint', 'Triangle', 'Center')
    view(38,27)
    if flag4==1
        F=getframe(gcf);
        aviobj=addframe(aviobj,F);
        Mk(ii)=getframe(gcf);
    end
end
%-----
%Averaging the vertices of the triangle for point a and point p, to
%get a better estimate on the plane that the HA is going through
%-----
al0x=(x10(a)+x10(p))/2; al0y=(y10(a)+y10(p))/2;
al0z=(z10(a)+z10(p))/2;
allx=(x11(a)+x11(p))/2; all1y=(y11(a)+y11(p))/2;

```

```

a11z=(z11(a)+z11(p))/2;
a12x=(x12(a)+x12(p))/2; a12y=(y12(a)+y12(p))/2;
a12z=(z12(a)+z12(p))/2;
a10=[a10x a10y a10z]; a11=[a11x a11y a11z]; a12=[a12x a12y a12z];
A10(c,:)=a10; A11(c,:)=a11; A12(c,:)=a12;
%-----
%Intersection of the helical axis to the plane established by the
%three vertices of the triangle
%-----
tVect=n-s;
P1=l1(a,:)-l0(a,:);
P2=l2(a,:)-l0(a,:);
P3=cross(P1,P2);
p3dot(c,1)=dot(P3,tVect);
if p3dot(c)==0
    msg=['**** ' ii ' Vector is parallel to the plane'];
    disp(msg);
    continue
end
A=P3(1);
B=P3(2);
C=P3(3);
D=A*l0(a,1)+B*l0(a,2)+C*l0(a,3);
tThigh(ii,1)=(A*s(1)+B*s(2)+C*s(3)+D)/(A*(n(1)-s(1))+...
    B*(n(2)-s(2))+C*(n(3)-s(3)));

%-----
%Where the intersection occurs in the plane of the triangle
%-----
tx(ii,1)=s(1)+tVect(1)*tThigh(ii);
ty(ii,1)=s(2)+tVect(2)*tThigh(ii);
tz(ii,1)=s(3)+tVect(3)*tThigh(ii);
tv=[tx(ii) ty(ii) tz(ii)];
Tri=[x10(a) y10(a) z10(a);
    x11(a) y11(a) z11(a);
    x12(a) y12(a) z12(a);
    x10(a) y10(a) z10(a)];
s1=sqrt((x10(a)-x11(a))^2+(y10(a)-y11(a))^2+(z10(a)-z11(a))^2);
s2=sqrt((x11(a)-x12(a))^2+(y11(a)-y12(a))^2+(z11(a)-z12(a))^2);
s3=sqrt((x12(a)-x10(a))^2+(y12(a)-y10(a))^2+(z12(a)-z10(a))^2);
side(c,1)=s1; side(c,2)=s2; side(c,3)=s3;%Length of each side
norm_xy=[P3(1)+tx(ii); P3(2)+ty(ii); P3(3)+tz(ii)];
Norm_xy=[norm_xy'; tx(ii) ty(ii) tz(ii)];
Sxy_ang=acos(dot(norm_xy,s)/(sqrt(s(1)^2+s(2)^2+s(3)^2)*...
    sqrt(norm_xy(1)^2+norm_xy(2)^2+norm_xy(3)^2)));

center(c,1)=h; center(c,2)=k; center(c,3)=l;
PInt(c,1)=tx(ii); PInt(c,2)=ty(ii); PInt(c,3)=tz(ii);
%-----
%Plot of the center of the triangle and the intersection of helical
%axis on the plane
%-----
if flag2==1

```

```

figure(2)
hold on
scatter3(tx(ii), ty(ii),tz(ii), 'k')
scatter3(h,k,l, 'FaceColor','r')
grid on
xlabel('x (cm)', 'fontsize', 16, 'fontweight', 'b')
ylabel('y (cm)', 'fontsize', 16, 'fontweight', 'b')
zlabel('z (cm)', 'fontsize', 16, 'fontweight', 'b')
title('Helical Axis Intersection and Triangle Center', ...
      'fontsize', 30, 'fontweight', 'b')
legend('Helical Axis Intersection', 'Triangle Center')
view(38,27)
if ii==len-1
    fig=fig+2;
end
view(37,28)
end
%-----
%Distance between actual intersection and predicted intersection
%-----
dis(d,1)=sqrt((tx(ii)-h)^2+(ty(ii)-k)^2+(tz(ii)-l)^2);
d=d+1;

c=c+1;
end
PHI_deg=PHI*180/pi;
%-----
%Plots of various individual components of helical axis
%-----
if flag3==1
    y1=PHI_deg;
    x1=1:length(PHI_deg);
    figure(fig)
    hold on
    plot(x1,y1, 'r')
    title('Phi')
    fig=fig+1;
    %
    figure(fig)
    hold on
    scatter3(S(:,1), S(:,2), S(:,3), 'sr')
    xlabel('x')
    ylabel('y')
    zlabel('z')
    grid on
    title('Radius Vector')
    fig=fig+1;

    figure(fig)
    hold on
    scatter3(N(:,1), N(:,2), N(:,3), 'sr')
    xlabel('x')
    ylabel('y')

```

```

zlabel('z')
grid on
title('Unit Vector')
fig=fig+1;

y3=T;
x3=1:length(T);
figure(fig)
hold on
plot(x3,y3, 'r')
title('Translation Component')
fig=fig+1;
%
figure(fig)
hold on
scatter3(V(:,1), V(:,2), V(:,3), 'sr')
xlabel('x')
ylabel('y')
zlabel('z')
title('Translation Vector')
grid on
fig=fig+1;
end
dislen=1:length(dis);
title1={'Frame' 'Dis' 'ttx' 'tty' 'ttz' 'center' 'side1' 'side2' ...
        'side3'};
sav_data1=[dislen' dis PInt center side];
xlswrite('Tri_Data', title1, sheet, 'a1');
xlswrite('Tri_Data', sav_data1, sheet, 'a2');
ej=1;
dis_out=0;
for iii=1:length(dis)
    if dis(iii)>mean(dis)+2*std(dis)
        dis_out(iii,1)=iii;
    else
        disrev(ej,1)=dis(iii);
        PINT(ej,:)=PInt(iii,:);
        SIDE(ej,:)=side(iii,:);
        ej=ej+1;
    end
end
dmmr=[mean(disrev) std(disrev) min(disrev) max(disrev)]
dislen2=1:length(disrev);
title2={'Frame' 'Dis' 'ttx' 'tty' 'ttz' 'center' 'side1' 'side2' ...
        'side3' 'Outliers'};
sav_data2=[dislen2' disrev PINT SIDE];
xlswrite('Tri_Out', title2, sheet, 'a1');
xlswrite('Tri_Out', sav_data2, sheet, 'a2');
xlswrite('Tri_Out', dis_out, sheet, 'j2');

```

```

%iscollinear.m
function[outdata]=iscollinear(a, b, c)

%Three points are collinear all three lie on a straight line. So the sum
%of the distance between points a and b and the distance between points
%b and c equals the sum between points a and c.

ab=sqrt((a(1)-b(1))^2+(a(2)-b(2))^2+(a(3)-b(3))^2);
bc=sqrt((b(1)-c(1))^2+(b(2)-c(2))^2+(b(3)-c(3))^2);
ac=sqrt((a(1)-c(1))^2+(a(2)-c(2))^2+(a(3)-c(3))^2);
if ab+bc==ac||bc+ac==bc||ac+bc==ab
    outdata=1;
else
    outdata=0;
end

```

```

%RMV.m
%-----
%Rotation matrix validation (RMV) -- Compute the sqrt of the sum of the
%squared values in each row and column. Return the product of all these
%values multiplied together. The answer should be '1' for a true
%rotation matrix.
%-----
function[outdata] = RMV(name,indata,indx)

[m,n]=size(indata);

%compute column values first
cnt=1;
for i=1:m
    sum = 0;
    for j=1:n
        sum = sum + indata(i,j)^2;
    end
    result(cnt) = sqrt(sum);
    cnt = cnt+1;
end

%compute row values next
for j=1:m
    sum = 0;
    for i=1:n
        sum = sum + indata(i,j)^2;
    end
    result(cnt) = sqrt(sum);
    cnt = cnt+1;
end

%check all results
outdata = 1;
cnt=cnt-1;
for i=1:cnt
    outdata = outdata * result(i);
end

if cast(outdata,'single') ~= 1
    msg=['*** ' name '(' num2str(indx) ') is not a rotation matrix.'];
    disp(msg);
end

```

```

%summer.m
function [R, S]=summer(G)

%Positive polar decomposition of 3x3 matrix G=R*S
%Forcing positive orthonormal rotation matrix

%Inputs
%G=3x3 general matrix

%Outputs
%R=3x3 positive orthonormal matrix
%S=3x3 symmetric matrix

EPS=1.0e-5;

%Cofactors and Determinant of g
COG(1,1)=G(2,2)*G(3,3)-G(2,3)*G(3,2);
COG(2,1)=G(1,3)*G(3,2)-G(1,2)*G(3,3);
COG(3,1)=G(1,2)*G(2,3)-G(1,3)*G(2,2);
COG(1,2)=G(2,3)*G(3,1)-G(2,1)*G(3,3);
COG(2,2)=G(1,1)*G(3,3)-G(1,3)*G(3,1);
COG(3,2)=G(1,3)*G(2,1)-G(1,1)*G(2,3);
COG(1,3)=G(2,1)*G(3,2)-G(2,2)*G(3,1);
COG(2,3)=G(1,2)*G(3,1)-G(1,1)*G(3,2);
COG(3,3)=G(1,1)*G(2,2)-G(1,2)*G(2,1);
G3=G(1,1)*COG(1,1)+G(2,1)*COG(2,1)+G(3,1)*COG(3,1);

P=G'*G;
%Adjoint of P
ADP(1,1)=P(2,2)*P(3,3)-P(2,3)*P(3,2);
ADP(2,2)=P(1,1)*P(3,3)-P(1,3)*P(3,1);
ADP(3,3)=P(1,1)*P(2,2)-P(1,2)*P(2,1);

%G invariants
G1SQ=P(1,1)+P(2,2)+P(3,3);
G1=sqrt(G1SQ);
G2SQ=ADP(1,1)+ADP(2,2)+ADP(3,3);
G2=sqrt(G2SQ);

%Initialize Iteration
H1=G2/G1SQ;
H2=G3*G1/G2SQ;
X=1.0;
Y=1.0;
DX=1.0;
DY=1.0;

%Iteration Loop
while abs(DX/X)>EPS || abs(DY/Y)>EPS
    DEN=2.0*(X*Y-H1*H2);
    RES1=1.0-X*X+2.0*H1*Y;
    RES2=1.0-Y*Y+2.0*H2*X;

```



```

DX=(Y*RES1+H1*RES2)/DEN;
DY=(H2*RES1+X*RES2)/DEN;
X=X+DX;
Y=Y+DY;
end

%Beta Invariants
BETA1=X*G1;
BETA2=Y*G2;

%Invert(G'*G+BETA2*I)
P(1,1)=P(1,1)+BETA2;
P(2,2)=P(2,2)+BETA2;
P(3,3)=P(3,3)+BETA2;
PBI(1,1)=P(2,2)*P(3,3)-P(2,3)*P(3,2);
PBI(1,2)=P(1,3)*P(3,2)-P(1,2)*P(3,3);
PBI(1,3)=P(1,2)*P(2,3)-P(1,3)*P(2,2);
PBI(2,1)=P(2,3)*P(3,1)-P(2,1)*P(3,3);
PBI(2,2)=P(1,1)*P(3,3)-P(1,3)*P(3,1);
PBI(2,3)=P(1,3)*P(2,1)-P(1,1)*P(2,3);
PBI(3,1)=P(2,1)*P(3,2)-P(2,2)*P(3,1);
PBI(3,2)=P(1,2)*P(3,1)-P(1,1)*P(3,2);
PBI(3,3)=P(1,1)*P(2,2)-P(1,2)*P(2,1);
DETPBI=P(1,1)*PBI(1,1)+P(2,1)*PBI(1,2)+P(3,1)*PBI(1,3);

for i=1:3

R(i,1)=(COG(i,1)+BETA1*G(i,1))*PBI(1,1)+(COG(i,2)+BETA1*G(i,2))*PBI(2,1)
)+(COG(i,3)+BETA1*G(i,3))*PBI(3,1)/DETPBI;

R(i,2)=(COG(i,1)+BETA1*G(i,1))*PBI(1,2)+(COG(i,2)+BETA1*G(i,2))*PBI(2,2)
)+(COG(i,3)+BETA1*G(i,3))*PBI(3,2)/DETPBI;

R(i,3)=(COG(i,1)+BETA1*G(i,1))*PBI(1,3)+(COG(i,2)+BETA1*G(i,2))*PBI(2,3)
)+(COG(i,3)+BETA1*G(i,3))*PBI(3,3)/DETPBI;
end
%
for i=1:3
S(i,1)=R(1,i)*G(1,1)+R(2,i)*G(2,1)+R(3,i)*G(3,1);
S(i,2)=R(1,i)*G(1,2)+R(2,i)*G(2,2)+R(3,i)*G(3,2);
S(i,3)=R(1,i)*G(1,3)+R(2,i)*G(2,3)+R(3,i)*G(3,3);
end

%
```

APPENDIX E

Programs used to determine helical axis, helical axis intersection, and angle helical axis
makes with plane for the cylinder model

```
% HA4.m
% Helical Axis from Motion Monitor data collected from the cylinder

% These calculations are taken from Spoors 1980, "Rigid Body Motion
% Calculated from Spatial Co-ordinates of Markers

clear
close all
clf

file_num=input('What file number is this? ');
sheet_name=['Sheet' num2str(file_num)];
file_name=input('What file name is this? ', 's');
Row_num=file_num+2;
data=xlsread(file_name, sheet_name, 'a1:ae800');

flag1=0;%Graphing the calculating helical axis
flag2=0;%Graphing individual components of HA
fig=1;

frame=data(:,1);

%-----
%Landmark Data to Use
%-----
10x=data(:,2); 10y=data(:,3); 10z=data(:,4);
11x=data(:,5); 11y=data(:,6); 11z=data(:,7);
12x=data(:,8); 12y=data(:,9); 12z=data(:,10);
13x=data(:,11); 13y=data(:,12); 13z=data(:,13);
14x=data(:,14); 14y=data(:,15); 14z=data(:,16);
15x=data(:,17); 15y=data(:,18); 15z=data(:,19);
16x=data(:,20); 16y=data(:,21); 16z=data(:,22);
%X,Y,Z individual data
Lx=[10x 11x 12x 13x 14x 15x 16x];%All the X landmark data
Ly=[10y 11y 12y 13y 14y 15y 16y];%All the Y landmark data
Lz=[10z 11z 12z 13z 14z 15z 16z];%All the Z landmark data
```

```

%Average of the X,Y,Z landmark data for each point in time
avgx=mean(Lx,2);
avgy=mean(Ly,2);
avgz=mean(Lz,2);
avg=[avgx avgy avgz];
%X,Y,Z Data of the Landmark
l0=[l0x l0y l0z];
l1=[l1x l1y l1z];
l2=[l2x l2y l2z];
l3=[l3x l3y l3z];
l4=[l4x l4y l4z];
l5=[l5x l5y l5z];
l6=[l6x l6y l6z];
%Plane
pxl0=data(:,23); pyl0=data(:,24); pzl0=data(:,25);
pxl1=data(:,26); pyl1=data(:,27); pzl1=data(:,28);
pxl2=data(:,29); pyl2=data(:,30); pzl2=data(:,31);
%Frequency
freq=100;
frate=1/freq;
%
I=eye(3);
len=length(frame);
%Counters for both shank and thigh
c=1;
em=1;
td=1;
count=1;
d=1;
for ii=1:len-1
    a=ii;
    p=ii+1;
    %-----
    %Ensure that the landmark points aren't collinear
    %-----
    %
    %Thigh
    acol=iscollinear1(l0(a,:), l1(a,:), l2(a,:), l3(a,:), l4(a,:), ...
        15(a,:), l6(a,:));
    pcol=iscollinear1(l0(p,:), l1(p,:), l2(p,:), l3(p,:), l4(p,:), ...
        15(p,:), l6(p,:));
    if acol==1
        msg=['*** acol (' num2str(a) ') landmarks are collinear.'];
        disp(msg);
        continue
    elseif pcol==1
        msg=['*** pcol (' num2str(p) ') landmarks are collinear.'];
        disp(msg);
        continue
    end
end
%
%-----
%Make a 3x3 Matrix that will be manipulated to become the rotation
%matrix used in the rest of the calculations. These steps make up

```

```

%equation (5) in Spoors paper
%-----
%Rotation Matrix
m0=l0(p,:)'*l0(a,:);
m1=l1(p,:)'*l1(a,:);
m2=l2(p,:)'*l2(a,:);
m3=l3(p,:)'*l3(a,:);
m4=l4(p,:)'*l4(a,:);
m5=l5(p,:)'*l5(a,:);
m6=l6(p,:)'*l6(a,:);
summ=(m0+m1+m2+m3+m4+m5+m6)/7;
mavg=avg(p,:)'*avg(a,:);
M=summ-mavg;
%Function sommer comes from the subroutine MATPP3 by H.J. Sommer.
[R, S1]=sommer(M);
RMV('R',R,a);
%-----
%Translation Vector from equation (10) in Spoors
%-----
v=avg(p,:)'-R*avg(a,:)';
%-----
%Solving for angle of rotation about the helical axis (phi)
%-----
%Spoors Eq 32
sinphi=0.5*sqrt((R(3,2)-R(2,3))^2+(R(1,3)-R(3,1))^2+...
(R(2,1)-R(1,2))^2);
%Spoors Eq 34
cosphi=0.5*(trace(R)-1);
%If statements from Spoors; different methods are preferred a
%different times because the arc sine function
if sinphi<=1/2*sqrt(2)
    phi=asin(sinphi);
    if cosphi<0
        phi=pi-phi;
    end
else
    phi=acos(cosphi);
end
%-----
%Solving for Unit vector (n)
%-----
if phi<=3/4*pi && phi~=0
    %Spoors 31
    n=(.5*[R(3,2)-R(2,3); R(1,3)-R(3,1); R(2,1)-R(1,2)])/sin(phi);
else
    %Spoors 35 and 36
    b=0.5*(R+R')-cos(phi)*I;
    blen=[sqrt(b(:,1)'*b(:,1));
        sqrt(b(:,2)'*b(:,2));
        sqrt(b(:,3)'*b(:,3))];
    [Blen In]=max(blen);
    n=b(:,In)/Blen;
    if sin(phi)<0
        n=-n;
    end
end

```

```

    end
end
%-----
%Solving for Translation Component (t)
%-----
%Spoors 37
t=n'*v;
%-----
%Solving for Radius Vector (s)
%-----
denom=1-cos(phi); %denominator portion of radius vector
%If it equals 0 making it a really small number so it will give a
%actual value and NaN
if denom==0
    denom=0.0000001;
    count=count+1;
end
%Spoors 38
s=-.5*cross(n,cross(n,v))+sin(phi)/(2*(denom))*cross(n,v);
%-----
%If Phi==0 no defined helical axis and therefore n,s, and t aren't
%unique, these equations are used to give non-unique helical axis
%values
%-----
if phi==0 & v~=[0;0;0]
    t1=v'*v;
    t=sqrt(t1);
    n=(1./t)*v;
    s=[0; 0; 0];
    em=em+1;
end
%-----
%Saving Data
%-----
N(c,1)=n(1,1); N(c,2)=n(2,1); N(c,3)=n(3,1);
PHI(c,1)=phi;
S(c,1)=s(1,1); S(c,2)=s(2,1); S(c,3)=s(3,1);
T(c,1)=t;
V(c,1)=v(1,1); V(c,2)=v(2,1); V(c,3)=v(3,1);
%-----
%Checks, ensure the unit vectors are unit vectors and the unit
%vector and radius vector are orthogonal
%-----
%Thigh
npn=n'*n;
nps=n'*s;
Unit(c,1)=npn;
Orth(c,1)=nps;
%-----
%Used in the graph of the helical axis
%-----
NS=vertcat(n',s');
%The graph of the helical axis
%-----

```

```

%The instantaneous helical axis should be parallel to the plane
%which the cylinder is rolled on.
%-----
pal0x=(px10(a)+px10(p))/2; pal0y=(l0(a)+l0(p))/2;
pal0z=(pz10(a)+pz10(p))/2;
pal1x=(px11(a)+px11(p))/2; pal1y=(py11(a)+py11(p))/2;
pal1z=(pz11(a)+pz11(p))/2;
pal2x=(l2(a)+l2(p))/2; pal2y=(l2(a)+l2(p))/2;
pal2z=(l2(a)+l2(p))/2;
pal0=[pal0x pal0y pal0z]; pal1=[pal1x pal1y pal1z];
pal2=[pal2x pal2y pal2z];
pA10(c,:)=pal0; pA11(c,:)=pal1; pA12(c,:)=pal2;
P1=pal1-pal0;
P2=pal2-pal0;
P3=cross(P1,P2);
A=P3(1);
B=P3(2);
C=P3(3);
D=A*pal0(1)+B*pal0(2)+C*pal0(3);
u=(A*n(1)+B*n(2)+C*n(3)+D)/(A*(n(1)-s(1))+B*(n(2)-s(2))+...
    C*(n(3)-s(3)));
P=n+u*(s-n); %Intersection of the HA on the plane
plane=[pal0;pal1;pal2;pal0];
pInt(c,1)=P(1); pInt(c,2)=P(2); pInt(c,3)=P(3);

Vect=n-s;
tp3dot(c,1)=dot(P3,Vect);
if tp3dot==0
    msg=['**** ' ii ' Vector is parallel to the plane'];
    disp(msg);
end
NS1=vertcat(NS,P');
%-----
%Find Location, intersection of the plane at center of cylinder to
%compare with other data
%-----
al0x=(l0x(a)+l0x(p))/2;
al0y=(l0y(a)+l0y(p))/2;
al0z=(l0z(a)+l0z(p))/2;
center=[al0x al0y al0z];
Pal0=[A B C]; Pal1=[pal1x pally pal1z];
Pal2=[pal2x pal2y pal2z];
Plane=[Pal0;pal1;pal2;Pal0];
asb=-1; asc=1; asa=(-B*asb-C*asc)/A;

a1=asa; b1=asb; c1=asc;
d1=-a1*al0x-b1*al0y-c1*al0z;
u1=(a1*n(1)+b1*n(2)+c1*n(3)+d1)/(a1*(n(1)-s(1))+b1*(n(2)-s(2))+...
    c1*(n(3)-s(3)));
ansck=-a1*al0x-b1*al0y-c1*al0z;
if abs(d1-ansck)<10^-4
    output(c,1)=1;
else

```

```

        output(c,1)=d1-ansck;
    end
    P_int=n+u1*(s-n);
    PInt(c,1)=P_int(1); PInt(c,2)=P_int(2); PInt(c,3)=P_int(3);
    PInt_dis(c,1)=sqrt((P_int(1)-a10x)^2+(P_int(2)-a10y)^2+...
        (P_int(3)-a10z)^2);
    Plane1=[A B C]; Plane2=[a1 b1 c1];
    perp_check(c,1)=dot(Plane1,Plane2);
    %-----
    %Find the angle the line makes with the to try and account for
    %the errors

    norm_xy=[P3(1)+P_int(1); P3(2)+P_int(2); P3(3)+P_int(3)];
    Norm_xy=[norm_xy'; P_int(1) P_int(2) P_int(3)];
    Sxy_ang=acos(dot(norm_xy,s)/(sqrt(s(1)^2+s(2)^2+s(3)^2)*...
        sqrt(norm_xy(1)^2+norm_xy(2)^2+norm_xy(3)^2)));
    SXY_Ang(c,1)=(Sxy_ang)*180/pi;
    SXY_Ang90(c,1)=90-(Sxy_ang)*180/pi;
    Nxy_ang=acos(dot(norm_xy,n)/(sqrt(n(1)^2+n(2)^2+n(3)^2)*...
        sqrt(norm_xy(1)^2+norm_xy(2)^2+norm_xy(3)^2)));
    NXY_Ang(c,1)=(Nxy_ang)*180/pi;
    NXY_Ang90(c,1)=90-(Nxy_ang)*180/pi;

    Norm_XY=[A+P(1) B+P(2) C+P(3); P(1) P(2) P(3)];
    NP=vertcat(n',P');
    SP=vertcat(s',P');
    if flag1==1
        figure(fig)
        hold on
        plot3(NS1(:,1), NS1(:,2), NS1(:,3), 'b')
        scatter3(s(1), s(2), s(3), 'g')
        scatter3(n(1), n(2), n(3), 'r')
        grid on
        xlabel('x')
        ylabel('y')
        zlabel('z')
        view(37,28)
        if ii==len-1
            fig=fig+1;
        end
    end
    end
    c=c+1;

end
tPHI_deg=PHI*180/pi;
if flag2==1
    figure(fig)
    hold on
    plot(tPHI_deg, 'r')
    title('Phi')
    fig=fig+1;
    %

```

```

figure(fig)
hold on
scatter3(S(:,1), S(:,2), S(:,3), 'r')
xlabel('x')
ylabel('y')
zlabel('z')
grid on
title('Radius Vector')
fig=fig+1;

figure(fig)
hold on
scatter3(N(:,1), N(:,2), N(:,3), 'r')
xlabel('x')
ylabel('y')
zlabel('z')
grid on
title('Unit Vector')
fig=fig+1;

figure(fig)
hold on
plot(T, 'r')
title('Translation Component')
fig=fig+1;
%
figure(fig)
hold on
scatter3(V(:,1), V(:,2), V(:,3), 'r')
xlabel('x')
ylabel('y')
zlabel('z')
title('Translation Vector')
grid on
fig=fig+1;
end

Sst_dev=std(abs(SXY_Ang));
ans=mean(PInt_dis);
PFrame=1:length(PInt_dis);

title1={'Frame' 'Error' 'X_int' 'Y_int' 'Z_int' 'SXY_Ang' ...
        'SXY_Ang90' 'NXY_Ang' 'NXY_Ang90'};
sav_data=[PFrame' PInt_dis PInt SXY_Ang SXY_Ang90 NXY_Ang NXY_Ang90];
xlswrite('cRBM Data', title1, sheet_name, 'a1')
xlswrite('cRBM Data', sav_data, sheet_name, 'a2')
emj=1;
ej=1;
for i=1:length(PInt_dis)
    if PInt_dis(i)>mean(PInt_dis)+2*std(PInt_dis)
        dis_out(i,1)=i;
    else
        Dis(emj,1)=PInt_dis(i);
    end
end

```



```

        dis_PInt(emj,:) = PInt(i,:);
        SXY(emj,1) = SXY_Ang(i);
        SXY90(emj,1) = SXY_Ang90(i);
        NXY(emj,1) = NXY_Ang(i);
        NXY90(emj,1) = NXY_Ang90(i);
        emj = emj + 1;
    end
end
dislen = length(Dis);
DisLen = (dislen - 400) / 2;
if isa(DisLen, 'integer') == 1
    count1 = 1;
    for m = DisLen : DisLen + 400
        DIS(count1,1) = Dis(m);
        Dis_PInt(count1,:) = dis_PInt(m,:);
        sxy(count1,1) = SXY(m);
        sxy90(count1,1) = SXY90(m);
        nxy(count1,1) = NXY(m);
        nxy90(count1,1) = NXY90(m);
        count1 = count1 + 1;
    end
else
    count2 = 1;
    for mm = ceil(DisLen) : ceil(DisLen + 400.5)
        DIS(count2,1) = Dis(mm);
        Dis_PInt(count2,:) = dis_PInt(mm,:);
        sxy(count2,1) = SXY(mm);
        sxy90(count2,1) = SXY90(mm);
        nxy(count2,1) = NXY(mm);
        nxy90(count2,1) = NXY90(mm);
        count2 = count2 + 1;
    end
end
anglen = 1 : length(DIS);

title1 = {'Frame' 'Error' 'X_int' 'Y_int' 'Z_int' 'SXY_Ang' ...
          'SXY_Ang90' 'NXY_Ang' 'NXY_Ang90' 'Outliers'};
sav_data = [anglen' DIS Dis_PInt sxy sxy90 nxy nxy90];
xlswrite('cRBM Data_out', title1, sheet_name, 'a1')
xlswrite('cRBM Data_out', sav_data, sheet_name, 'a2')
xlswrite('cRBM Data_out', dis_out, sheet_name, 'j2')
title5 = {'Average Error' 'Average SXY' 'Average SXY90' 'Average NXY' ...
          'Average NXY90' 'std Error' 'std SXY' 'std SXY90' 'std NXY' 'std
          NXY90'};
Tavg_ICC = [mean(DIS) mean(sxy) mean(sxy90) mean(nxy) mean(nxy90) ...
            std(DIS) std(sxy) std(sxy90) std(nxy) std(nxy90)];
xlswrite('cRBM_Average', title5, 'Sheet1', 'a1')
row_num = ['a' num2str(Row_num)];
xlswrite('cRBM_Average', Tavg_ICC, 'Sheet1', row_num)

```

```

%iscollinear1.m
function[outdata]=iscollinear1(a0, a1, a2, a3, a4, a5, a6)

%Three points are collinear all three lie on a straight line. So the sum
%of the distance between points a and b and the distance between points
%b and c equals the sum between points a and c.

a=a0; b=a1; c=a2; d=a3; e=a4; f=a5; g=a6;
ab=sqrt((a(1)-b(1))^2+(a(2)-b(2))^2+(a(3)-b(3))^2);
ac=sqrt((a(1)-c(1))^2+(a(2)-c(2))^2+(a(3)-c(3))^2);
ad=sqrt((a(1)-d(1))^2+(a(2)-d(2))^2+(a(3)-d(3))^2);
ae=sqrt((a(1)-e(1))^2+(a(2)-e(2))^2+(a(3)-e(3))^2);
af=sqrt((a(1)-f(1))^2+(a(2)-f(2))^2+(a(3)-f(3))^2);
ag=sqrt((a(1)-g(1))^2+(a(2)-g(2))^2+(a(3)-g(3))^2);
bc=sqrt((b(1)-c(1))^2+(b(2)-c(2))^2+(b(3)-c(3))^2);
bd=sqrt((b(1)-d(1))^2+(b(2)-d(2))^2+(b(3)-d(3))^2);
be=sqrt((b(1)-e(1))^2+(b(2)-e(2))^2+(b(3)-e(3))^2);
bf=sqrt((b(1)-f(1))^2+(b(2)-f(2))^2+(b(3)-f(3))^2);
bg=sqrt((b(1)-g(1))^2+(b(2)-g(2))^2+(b(3)-g(3))^2);
cd=sqrt((c(1)-d(1))^2+(c(2)-d(2))^2+(c(3)-d(3))^2);
ce=sqrt((c(1)-e(1))^2+(c(2)-e(2))^2+(c(3)-e(3))^2);
cf=sqrt((c(1)-f(1))^2+(c(2)-f(2))^2+(c(3)-f(3))^2);
cg=sqrt((c(1)-g(1))^2+(c(2)-g(2))^2+(c(3)-g(3))^2);
de=sqrt((d(1)-e(1))^2+(d(2)-e(2))^2+(d(3)-e(3))^2);
df=sqrt((d(1)-f(1))^2+(d(2)-f(2))^2+(d(3)-f(3))^2);
dg=sqrt((d(1)-g(1))^2+(d(2)-g(2))^2+(d(3)-g(3))^2);
ef=sqrt((e(1)-f(1))^2+(e(2)-f(2))^2+(e(3)-f(3))^2);
eg=sqrt((e(1)-g(1))^2+(e(2)-g(2))^2+(e(3)-g(3))^2);
fg=sqrt((f(1)-g(1))^2+(f(2)-g(2))^2+(f(3)-g(3))^2);
if ab+bc==ac||ab+ac==bc||ac+bc==ab
    outdata=1;
elseif ab+bd==ad||ab+ad==bd||bd+ad==ab
    outdata=1;
elseif ab+be==ae||ab+ae==be||be+ae==ab
    outdata=1;
elseif ab+bf==af||ab+af==bf||bf+af==ab
    outdata=1;
elseif ab+bg==ag||ab+ag==bg||bg+ag==ab
    outdata=1;
elseif ac+cd==ad||ac+ad==cd||cd+ad==ac
    outdata=1;
elseif ac+ce==ae||ac+ae==ce||ce+ae==ac
    outdata=1;
elseif ac+cf==af||ac+af==cf||cf+af==ac
    outdata=1;
elseif ac+cg==ag||ac+ag==cg||cg+ag==ac
    outdata=1;
elseif ad+de==ae||ad+ae==de||de+ae==ad
    outdata=1;
elseif ad+df==af||ad+af==ad||df+af==ad
    outdata=1;

```

```

elseif ad+dg==ag || ad+ag==dg || dg+ag==ad
    outdata=1;
elseif ae+ef==af || ae+af==ef || ef+af==ae
    outdata=1;
elseif ae+eg==ag || ae+ag==eg || eg+ag==ae
    outdata=1;
elseif af+fg==ag || af+ag==fg || fg+ag==af
    outdata=1;
elseif bc+cd==bd || bc+bd==cd || cd+bd==bc
    outdata=1;
elseif bc+ce==be || bc+be==ce || ce+be==bc
    outdata=1;
elseif bc+cf==bf || bc+bf==cf || cf+bf==bc
    outdata=1;
elseif bc+cg==bg || bc+bg==cg || cg+bg==bc
    outdata=1;
elseif bd+de==be || bd+be==de || de+be==bd
    outdata=1;
elseif bd+df==bf || bd+bf==df || df+bf==bd
    outdata=1;
elseif bd+dg==bg || bd+bg==dg || dg+bg==bd
    outdata=1;
elseif be+ef==bf || be+bf==ef || ef+bf==be
    outdata=1;
elseif be+eg==bg || be+bg==eg || eg+bg==be
    outdata=1;
elseif bf+fg==bg || bf+bg==fg || fg+bg==bf
    outdata=1;
elseif cd+de==ce || cd+ce==de || de+ce==cd
    outdata=1;
elseif cd+df==cf || cd+cf==df || df+cf==cd
    outdata=1;
elseif cd+dg==cg || cd+cg==dg || dg+cg==cd
    outdata=1;
elseif ce+ef==cf || ce+cf==ef || ef+cf==ce
    outdata=1;
elseif ce+eg==cg || ce+cg==eg || eg+cg==ce
    outdata=1;
elseif cf+fg==cg || cf+cg==fg || fg+cg==cf
    outdata=1;
elseif de+ef==df || de+df==ef || ef+df==de
    outdata=1;
elseif de+eg==dg || de+dg==eg || eg+dg==de
    outdata=1;
elseif df+fg==dg || df+dg==fg || fg+dg==df
    outdata=1;
elseif ef+fg==eg || ef+eg==fg || fg+eg==ef
    outdata=1;
else
    outdata=0;
end

```

```
%RMV.m
```

```

%-----
%Rotation matrix validation (RMV) -- Compute the sqrt of the sum of the
%squared values in each row and column. Return the product of all these
%values multiplied together. The answer should be '1' for a true
%rotation matrix.
%-----
function[outdata] = RMV(name,indata,indx)

[m,n]=size(indata);

%compute column values first
cnt=1;
for i=1:m
    sum = 0;
    for j=1:n
        sum = sum + indata(i,j)^2;
    end
    result(cnt) = sqrt(sum);
    cnt = cnt+1;
end

%compute row values next
for j=1:m
    sum = 0;
    for i=1:n
        sum = sum + indata(i,j)^2;
    end
    result(cnt) = sqrt(sum);
    cnt = cnt+1;
end

%check all results
outdata = 1;
cnt=cnt-1;
for i=1:cnt
    outdata = outdata * result(i);
end

if cast(outdata,'single') ~= 1
    msg=['*** ' name '(' num2str(indx) ') is not a rotation matrix.'];
    disp(msg);
end

```

```

%summer.m
function [R, S]=summer(G)

%Positive polar decomposition of 3x3 matrix G=R*S
%Forcing positive orthonormal rotation matrix

%Inputs
%G=3x3 general matrix

%Outputs
%R=3x3 positive orthonormal matrix
%S=3x3 symmetric matrix

EPS=1.0e-5;

%Cofactors and Determinant of g
COG(1,1)=G(2,2)*G(3,3)-G(2,3)*G(3,2);
COG(2,1)=G(1,3)*G(3,2)-G(1,2)*G(3,3);
COG(3,1)=G(1,2)*G(2,3)-G(1,3)*G(2,2);
COG(1,2)=G(2,3)*G(3,1)-G(2,1)*G(3,3);
COG(2,2)=G(1,1)*G(3,3)-G(1,3)*G(3,1);
COG(3,2)=G(1,3)*G(2,1)-G(1,1)*G(2,3);
COG(1,3)=G(2,1)*G(3,2)-G(2,2)*G(3,1);
COG(2,3)=G(1,2)*G(3,1)-G(1,1)*G(3,2);
COG(3,3)=G(1,1)*G(2,2)-G(1,2)*G(2,1);
G3=G(1,1)*COG(1,1)+G(2,1)*COG(2,1)+G(3,1)*COG(3,1);

P=G'*G;
%Adjoint of P
ADP(1,1)=P(2,2)*P(3,3)-P(2,3)*P(3,2);
ADP(2,2)=P(1,1)*P(3,3)-P(1,3)*P(3,1);
ADP(3,3)=P(1,1)*P(2,2)-P(1,2)*P(2,1);

%G invariants
G1SQ=P(1,1)+P(2,2)+P(3,3);
G1=sqrt(G1SQ);
G2SQ=ADP(1,1)+ADP(2,2)+ADP(3,3);
G2=sqrt(G2SQ);

%Initialize Iteration
H1=G2/G1SQ;
H2=G3*G1/G2SQ;
X=1.0;
Y=1.0;
DX=1.0;
DY=1.0;

%Iteration Loop
while abs(DX/X)>EPS || abs(DY/Y)>EPS
    DEN=2.0*(X*Y-H1*H2);
    RES1=1.0-X*X+2.0*H1*Y;
    RES2=1.0-Y*Y+2.0*H2*X;

```

```

DX=(Y*RES1+H1*RES2)/DEN;
DY=(H2*RES1+X*RES2)/DEN;
X=X+DX;
Y=Y+DY;
end

%Beta Invariants
BETA1=X*G1;
BETA2=Y*G2;

%Invert(G'*G+BETA2*I)
P(1,1)=P(1,1)+BETA2;
P(2,2)=P(2,2)+BETA2;
P(3,3)=P(3,3)+BETA2;
PBI(1,1)=P(2,2)*P(3,3)-P(2,3)*P(3,2);
PBI(1,2)=P(1,3)*P(3,2)-P(1,2)*P(3,3);
PBI(1,3)=P(1,2)*P(2,3)-P(1,3)*P(2,2);
PBI(2,1)=P(2,3)*P(3,1)-P(2,1)*P(3,3);
PBI(2,2)=P(1,1)*P(3,3)-P(1,3)*P(3,1);
PBI(2,3)=P(1,3)*P(2,1)-P(1,1)*P(2,3);
PBI(3,1)=P(2,1)*P(3,2)-P(2,2)*P(3,1);
PBI(3,2)=P(1,2)*P(3,1)-P(1,1)*P(3,2);
PBI(3,3)=P(1,1)*P(2,2)-P(1,2)*P(2,1);
DETPBI=P(1,1)*PBI(1,1)+P(2,1)*PBI(1,2)+P(3,1)*PBI(1,3);

for i=1:3

R(i,1)=((COG(i,1)+BETA1*G(i,1))*PBI(1,1)+(COG(i,2)+BETA1*G(i,2))*PBI(2,1)
)+(COG(i,3)+BETA1*G(i,3))*PBI(3,1))/DETPBI;

R(i,2)=((COG(i,1)+BETA1*G(i,1))*PBI(1,2)+(COG(i,2)+BETA1*G(i,2))*PBI(2,2)
)+(COG(i,3)+BETA1*G(i,3))*PBI(3,2))/DETPBI;

R(i,3)=((COG(i,1)+BETA1*G(i,1))*PBI(1,3)+(COG(i,2)+BETA1*G(i,2))*PBI(2,3)
)+(COG(i,3)+BETA1*G(i,3))*PBI(3,3))/DETPBI;
end
%
for i=1:3
S(i,1)=R(1,i)*G(1,1)+R(2,i)*G(2,1)+R(3,i)*G(3,1);
S(i,2)=R(1,i)*G(1,2)+R(2,i)*G(2,2)+R(3,i)*G(3,2);
S(i,3)=R(1,i)*G(1,3)+R(2,i)*G(2,3)+R(3,i)*G(3,3);
end

%
```

```

% HA4_spline.m
% Spline the data to be used in the helical axis calculations

clear
close all
clf

file_num=input('What file number is this? ');
filename=input('What is the trial filename with extension? ', 's');
data=xlsread(filename, 'Sheet1', 'a10:ah809');

flag1=0;%Graphing the calculating helical axis
flag2=1;%Graphing the intersection
flag3=0;%Graphing individual helical axis components
fig=1;

frame=data(:,1);

%-----
%Landmark Data to Use
%-----
l0x=data(:,5); l0y=data(:,6); l0z=data(:,7);
l1x=data(:,8); l1y=data(:,9); l1z=data(:,10);
l2x=data(:,11); l2y=data(:,12); l2z=data(:,13);
l3x=data(:,14); l3y=data(:,15); l3z=data(:,16);
l4x=data(:,17); l4y=data(:,18); l4z=data(:,19);
l5x=data(:,20); l5y=data(:,21); l5z=data(:,22);
l6x=data(:,23); l6y=data(:,24); l6z=data(:,25);
pxl0=data(:,26); pyl0=data(:,27); pzl0=data(:,28);
pxl1=data(:,29); pyl1=data(:,30); pzl1=data(:,31);
pxl2=data(:,32); pyl2=data(:,33); pzl2=data(:,34);
%Frequency
freq=100;
frate=1/freq;
%
I=eye(3);
len=length(frame);
%Counters for both shank and thigh
c=1;
tem=1;
tcount=1;
td=1;
sem=1;
scount=1;
sd=1;
x=1:len;y=1:len;z=1:len;
x1=1.5:1:len+1; y1=1.25:1:len+1; z1=1:1:len;
ppx0t=spline(x,l0x,x1);
ppy0t=spline(y,l0y,y1);
ppz0t=spline(z,l0z,z1);
ppx1t=spline(x,l1x,x1);
ppy1t=spline(y,l1y,y1);
ppz1t=spline(z,l1z,z1);

```

```

ppx2t=spline(x,l2x,x1);
ppy2t=spline(y,l2y,y1);
ppz2t=spline(z,l2z,z1);
ppx0s=spline(x,l3x,x1);
ppy0s=spline(y,l3y,y1);
ppz0s=spline(z,l3z,z1);
ppx1s=spline(x,l4x,x1);
ppy1s=spline(y,l4y,y1);
ppz1s=spline(z,l4z,z1);
ppx2s=spline(x,l5x,x1);
ppy2s=spline(y,l5y,y1);
ppz2s=spline(z,l5z,z1);
ppx0=spline(x,l6x,x1);
ppy0=spline(y,l6y,y1);
ppz0=spline(z,l6z,z1);
ppx1=spline(x,px10,x1);
ppy1=spline(y,pyl0,y1);
ppz1=spline(z,pz10,z1);
ppx2=spline(x,px11,x1);
ppy2=spline(y,pyl1,y1);
ppz2=spline(z,pz11,z1);
ppx3=spline(x,px12,x1);
ppy3=spline(y,pyl2,y1);
ppz3=spline(z,pz12,z1);

save_data=[ppx0t' ppy0t' ppz0t' ppx1t' ppy1t' ppz1t' ppx2t' ppy2t' ...
           ppz2t' ppx0s' ppy0s' ppz0s' ppx1s' ppy1s' ppz1s' ppx2s' ppy2s' ...
           ppz2s' ppx0' ppy0' ppz0' ppx1' ppy1' ppz1' ppx2' ppy2' ppz2' ...
           ppx3' ppy3' ppz3'];
Sheet_name=['Sheet' num2str(file_num)];
xlswrite('Cylinder Splined Data', save_data, Sheet_name, 'b1')

```


APPENDIX F

Standard deviation of the angle the cylinder made with the plane being rolled

Experiment Number	Woltring Model	Cylinder Model
1	2.65	22.5
2	2.56	24.1
3	2.56	18.9
4	2.56	22.8
5	2.65	25.5
6	2.65	23.2
7		21.5
8		22.2
9		24.2

VITA

Emily J. Carney was born on February 2, 1983 in Seneca, SC. She graduated from Seneca High School in 2001. Emily received her Bachelors of Science degree in Biomedical Engineering with a concentration in Mechanical Engineering from Louisiana Tech University in Ruston, LA in 2006.