

5-7-2011

The Use of Motion Analysis Technology as an Alternative Means of Assessing Spinal Deformity in Patients with Adolescent Idiopathic Scoliosis

Matthew J. Solomito

University of Connecticut - Storrs, msolomito@ccmckids.org

Recommended Citation

Solomito, Matthew J., "The Use of Motion Analysis Technology as an Alternative Means of Assessing Spinal Deformity in Patients with Adolescent Idiopathic Scoliosis" (2011). *Master's Theses*. 74.

https://opencommons.uconn.edu/gs_theses/74

This work is brought to you for free and open access by the University of Connecticut Graduate School at OpenCommons@UConn. It has been accepted for inclusion in Master's Theses by an authorized administrator of OpenCommons@UConn. For more information, please contact opencommons@uconn.edu.

The Use of Motion Analysis Technology as an Alternative Means
of Assessing Spinal Deformity in Patients with Adolescent
Idiopathic Scoliosis

Matthew John Solomito

B.S., Western New England College, 2007

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

at the

University of Connecticut

2011


APPROVAL PAGE

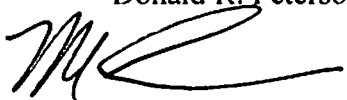
Master of Science Thesis

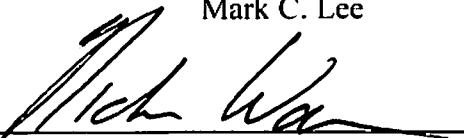
The Use of Motion Analysis Technology as an Alternative Means
of Assessing Spinal Deformity in Patients with Adolescent
Idiopathic Scoliosis

Presented by

Matthew John Solomito, B.S.

Major Advisor 
Donald R. Peterson

Associate Advisor 
Mark C. Lee

Associate Advisor 
Nicholas D. Warren

University of Connecticut

2011

Table of Contents

1.0 Introduction	1
1.1 Background	2
1.1.0 Anatomy.....	2
1.1.1 Scoliosis.....	5
1.1.2 Radiation Exposure.....	11
1.1.3 Motion Analysis.....	12
1.2 Pilot Work	19
2.0 Methods	21
2.1 Marker Configuration.....	22
2.2 X-ray Frame.....	24
2.3 Subjects.....	26
2.4 Subject testing.....	27
2.5 Matlab.....	30
2.6 Static Calculations.....	32
2.7 Dynamic Calculations.....	42
2.8 Filtering.....	49
2.9 Validation.....	51
3.0 Results	53
3.1 Subject Demographics.....	53
3.2 Validation.....	53
3.3 Static Results.....	57
3.4 Dynamic Results.....	61
4.0 Discussion	72
4.1 Model Considerations.....	72
4.2 Validation.....	73
4.3 Static Analysis.....	75
4.4 Dynamic Analysis.....	79
4.5 Clinical Considerations.....	82
4.6 Limitations.....	85
4.7 Future Work.....	86
5.0 Conclusion	87
6.0 Reference	89
7.0 Appendix A	91

List of Figures

Figure 1: Illustration of vertebra	2
Figure 2: Oblique view of spine	3
Figure 3: Sagittal vertical alignment.....	8
Figure 4: Coronal vertical alignment	9
Figure 5: Cobb angle.....	10
Figure 6: Marker configuration by Frigo.....	15
Figure 7: Marker configuration by Engsberg	16
Figure 8: Marker configuration by Cerveri.....	17
Figure 9: Marker configuration by Masso and Gorton	19
Figure 10: Marker set in this work.....	22
Figure 11: X-ray positioning frame	24
Figure 12: Schematics for X-ray frame	25
Figure 13: Global coordinate system	28
Figure 14: Illustration of bending tasks	29
Figure 15: Program flow chart.....	30
Figure 16: Angle of kyphosis.....	34
Figure 17: Proximal angle.....	36
Figure 18: Primary pelvic axis.....	43
Figure 19: Temporary axis.....	44
Figure 20: Z-axis of pelvis.....	44
Figure 21: Y-axis of pelvis	45
Figure 22: Primary mid-thorax axis.....	46
Figure 23: Primary upper thorax axis	47
Figure 24: Comparison of filter cutoff frequencies	50
Figure 25: Comparison of filter order.....	51
Figure 26: Validation test with manual goniometer	52
Figure 27: Comparison of Matlab and Vicon output-test subject.....	56
Figure 28: Comparison of Matlab and Vicon output-control subject.....	56
Figure 29: Kinematic profiles of anatomic motion in lateral bending.....	62
Figure 30: Kinematic profiles of spine motion in lateral bending.....	64
Figure 31: Kinematic profiles of anatomic motion in axial rotation	65
Figure 32: Kinematic profiles of spine motion in axial rotation.....	67
Figure 33: Kinematic profiles of anatomic motion in flexion task.....	69
Figure 34: Kinematic profiles of spine motion in flexion task	71
Figure 35: Comparison of Center of Pressure for control and test groups	75
Figure 36: Graphical representation of spinal curves	83
Figure 37: Graphical representation of static analysis results	83

List of Tables

Table 1: Pilot work asymmetry results	19
Table 2: Pilot work center of pressure results.....	20
Table 3: Pilot work static analysis results	20
Table 4: Subject demographics	53
Table 5: Comparison of known angles to calculated angles.....	54
Table 6: Comparison of proximal angle radiographic and calculated values	54
Table 7: Results of precision analysis.....	55
Table 8: Postural alignment results.....	57
Table 9: Static asymmetry results	58
Table 10: Correlation between radiographic and calculated measures.....	59
Table 11: Comparison of static measures with and without the X-ray frame.....	60
Table 12: Correlation between radiographic and calculated measures without frame	60
Table 13: Comparison of average range of anatomic motion in lateral bending	62
Table 14: Comparison of average range of spine motion in lateral bending	63
Table 15: Comparison of average range of anatomic motion in axial rotation.....	65
Table 16: Comparison of average range of spine motion in axial rotation.....	66
Table 17: Comparison of average range of anatomic motion in flexion task	68
Table 18: Comparison of average range of spine motion in flexion task	70

Abstract

Adolescent Idiopathic Scoliosis (AIS) is a three dimensional deformity of the spine that causes abnormal curvatures in the coronal and sagittal planes and abnormal vertebral rotations. AIS affects one to three percent of the population aged 10 to 16. The gold standard for diagnosing and tracking the progression of the deformity is through the use of repeated radiography and a patient can receive upwards of fifteen radiographic series over the course of treatment, which exposes this population to an increased risk of radiation-dependent complications.

This project proposes an alternative means of diagnosing scoliosis in an effort to reduce the patient's exposure to ionizing radiation. This alternative means uses motion analysis equipment to produce the measures commonly derived from radiographs and provide additional information about the rotational deformity and kinematic profiles of the spine. This thesis presents the necessary calculations and the marker set required to make the motion capture technology a viable alternative to radiographs.

Using a simplified marker set of 10 retro-reflective markers, it is possible to reproduce six radiographic measures: the sagittal vertical alignment, the coronal vertical alignment, the proximal angle, the distal angle, the angle of lordosis, and the angle of kyphosis. The additional use of force plates makes it is possible to determine asymmetries in the frontal and transverse plane and the balance of the spine for each subject. The kinematic profiles of the spine can also be determined using Euler rotation sequences.

The results of this work successfully showed that there is a high correlation between the radiographic measures and those calculated using motion analysis

technology. When applied clinically, the results from a small set of AIS patients indicated that patients with scoliosis tend to have their balance shifted towards the right when compared to those with radiographically normal spines. These results also indicate that patients with scoliosis have more rotational asymmetry than the control group. The kinematic profiles of the spine indicate that there are subtle differences between the scoliosis group and control group, which can be used to help describe the underlying spinal and muscular deformities of patients with AIS. This work provides a viable means of reducing the number of radiographs this patient population is subjected to over the course of their non-operative treatment. It also provides a strong foundation for future work on applying motion capture technology to advance scoliosis research.

1.0 Introduction

Adolescent idiopathic scoliosis is a three dimensional deformity of the spine that affects children between the ages of 10 and 16. Currently the gold standard for diagnosing and tracking the progression of this deformity is through the use of radiographs. There are two drawbacks to the use of radiographs, the first is that radiographs are two dimensional in nature; while they are capable of providing a clear understanding of the coronal and sagittal plane deformities of the spine it is difficult to ascertain the rotational deformity. The second drawback and, of greater concern, is the risk of radiation exposure to this patient population; although X-ray technology has improved over the past thirty years, there is still a risk of exposure to ionizing radiation and therefore, a risk of these patients developing radiation dependent complications.

The goal of this project is to create a method of defining the spinal deformity seen in adolescent idiopathic scoliosis without the need for ionizing radiation. The focus of this project is to determine if three dimensional motion analysis technology could be used to accomplish this goal. A marker set for the motion analysis equipment was designed to describe the deformities of the spine in all three anatomic planes. The motion capture information was compared against the current gold standard to prove that the motion analysis equipment can provide similar data to the radiographs and provide additional data about the rotational deformity of the spine. A Matlab program was designed to read the data produced by the motion capture system, and calculate the required angles to describe the spinal deformity caused by scoliosis. The Matlab program and motion analysis equipment were used to provide dynamic data about the spinal range of motion for this patient population to better describe the spinal deformity.

The motion analysis data and corresponding Matlab program will provide physicians with an alternative means of tracking and diagnosing adolescent idiopathic scoliosis, in an effort to reduce the number of radiographs that this patient population requires over the course of their treatment.

1.1 Background

1.1.0 Anatomy

The spine is a complex series of joints made up of vertebra and articulating surfaces, intervertebral discs that create a flexible column. The spine is constructed of 33 vertebrae; each vertebra has two distinct parts, the anterior portion (i.e., the body) and the posterior portion (i.e., the arch). The body is a solid segment of bone, and the arch is formed by two pedicles, two lamina, and seven processes illustrated in Figure 1a and 1b [1].

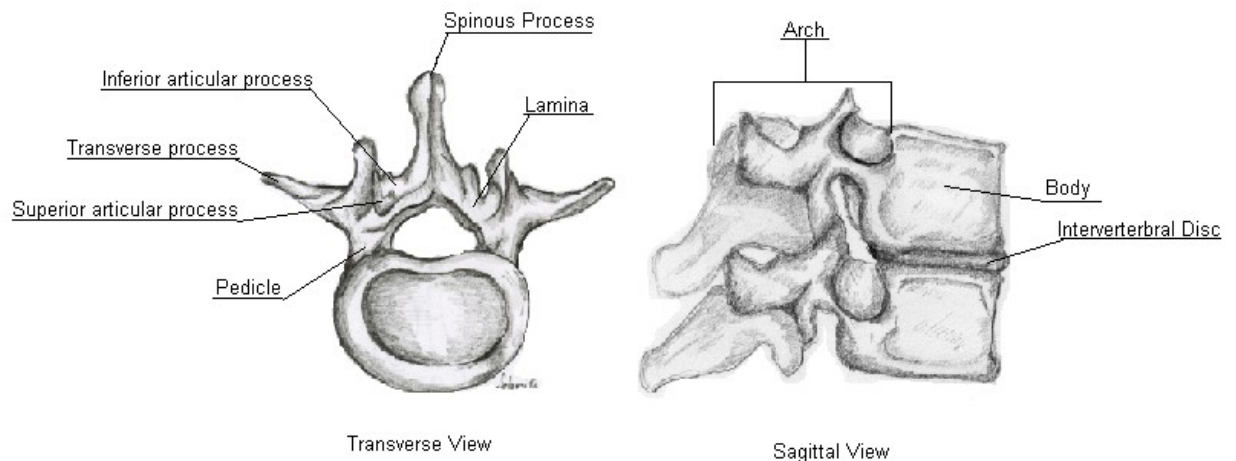


Figure 1: A sagittal (a) and transverse (b) view of a vertebra

The body is the largest portion of the vertebra, and makes up the structural aspect of the spine. Each vertebra is separated by an intervertebral disc, an articulating surface, which

allows for movement throughout the spine. The pedicles protrude from the body and are evenly spaced on either side of the body [1]. The lamina are two bony plates that fuse in the back creating a canal for the spinal cord to pass through. The processes project from both the pedicles and lamina, and serve as the attachment sites for the muscles and the ligaments of the trunk [1]. The spine is separated into five distinct regions as depicted in Figure 2.

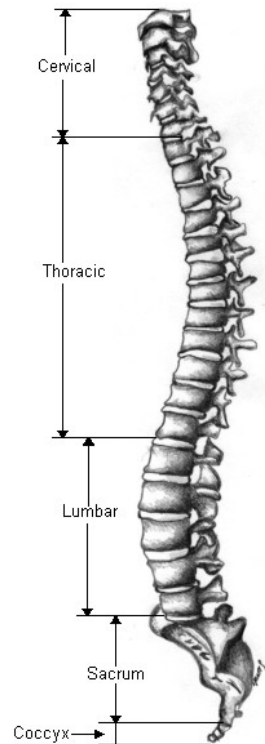


Figure 2: An oblique view of the spine

The upper most region of the spine is called the cervical spine, and consists of seven vertebrae; these seven vertebrae tend to be the smallest vertebra of the spine. The next 12 vertebra make up the thoracic portion of the spine; these vertebrae are slightly larger than those found in the cervical region, and have more angled spinous processes [1]. The lumbar region of the spine is made up of five vertebrae and are the largest of the non-

fused vertebra found in the spine. The sacrum and coccyx are the last two regions of the spine and are constructed of fused vertebra; the sacrum contains five vertebrae and the coccyx region contains the remaining four vertebrae [1]. While this is the common distribution of vertebrae throughout the five regions of the spine, it is not uncommon in some pathological conditions to find a sixth lumbar vertebra and only 11 thoracic vertebrae [1].

The spine provides the body with structure, supports the skull, and allows for the motion of the trunk. The spine on average is two feet, three inches in length, when measured along the anterior aspect of the spine [1]. The sagittal aspect of the spine has two main curves; the upper most curve is the thoracic kyphosis, which is an anteriorly concave curve that measuring between 20° and 40° , and the kyphosis begins at the second thoracic vertebra and terminates at the twelfth thoracic vertebra with the apex of the curve occurring around the seventh thoracic vertebrae [1, 2]. The second curve, the lower of the two sagittal curves of the spine, is the lumbar lordosis; this is an anteriorly convex curve measuring between 40° and 60° ; the lordosis begins at the twelfth thoracic vertebra and terminates at the lumbar-sacral joint, and the apex of the curve occurs around the third lumbar vertebra [1, 2]. The coronal aspect of the spine has a slight curvature to the right; this curvature in a healthy spine is less than 10° , and is caused by the muscle attachments along the spine [1, 2].

The muscles that move the spine are extremely complex and have multiple origins and insertions along the spine; this causes the muscles to overlap providing stability to the spine [3]. The erector spinae is the largest of the muscles attached to the spine and creates a bulge on either side of the spine. The erector plays an important role in allowing

the spine to flex, laterally lean, and rotate [3]. The multifidus aids in the extension of the spine as well as lateral leaning, and the rotatores aid in the extension of the spine and spine rotation [3].

1.1.1 Scoliosis

Scoliosis is an abnormal curvature of the spine that can cause back pain and in some severe cases it can also cause cardio-pulmonary complications. There are three main classifications of scoliosis: congenital scoliosis, neuromuscular scoliosis, and idiopathic scoliosis [2]. Neuromuscular Scoliosis is commonly seen in patients with a neuromuscular disorder such as cerebral palsy or spina bifida. Congenital Scoliosis is a scoliosis that occurs due to a malformation of a vertebra or rib at birth [2]. Idiopathic Scoliosis is a scoliosis that has no known cause [4]. The scope of this project is narrowed to adolescent idiopathic scoliosis.

Adolescent idiopathic scoliosis (AIS) is a structural deformity of the spine that occurs in all three planes. AIS affects one to three percent of children between the ages of 10 and 16 [4]. The cause of Adolescent idiopathic scoliosis is unknown and can present with symptoms such as back pain, asymmetry in the muscles of the back, and uneven shoulders and pelvis in otherwise healthy children around the age of puberty [2]. Many early studies showed a poor outlook for those diagnosed with this deformity, stating that there were severe cardio-pulmonary complications caused by the abnormal curves of the spine; however, recent studies have shown that vital capacity is slightly reduced with curves of 50° or greater because of decreased thoracic space. It is not until curves of 100° or greater are there signs of cardio-pulmonary complications [4].

While there is no known cause for Idiopathic Scoliosis, there are many theories on why this disorder occurs. These theories include genetic factors, growth anomalies, connective tissue disorders, and biomechanical problems [2]. The theories for a genetic factor revolve around the theory that there seems to be a hereditary link; a patient has a higher risk of developing Idiopathic Scoliosis if a parent or sibling has already been diagnosed with the disorder. There is a 73% chance that both identical twins will have Idiopathic Scoliosis if one twin has already been diagnosed [2]. Some researchers also believe that Idiopathic Scoliosis is an X-linked genetic disorder since females are four times more likely to have Idiopathic Scoliosis, and ten times more likely for their curves to progress than males [2, 4]. There has been no conclusive gene or genetic anomaly found to cause AIS. Some researchers theorize that Adolescent idiopathic scoliosis is caused from connective tissue disorders that arise during times of rapid growth. This theory has come about since AIS occurs around the time of puberty, commonly associated with rapid growth; during these growth spurts the connective tissue, muscles, and ligaments place greater stress on the spine causing the abnormal curvatures [2]. However, the theory that unusual strains on the muscles attached to the spine is caused by rather than the cause of AIS [2]. The biomechanical theory is that spine stability relies on the vertebral bodies, intervertebral discs, and muscles and ligaments attached to the ribs and spine. A defect in any one of these structures could cause an imbalance of the distributed forces along the spine; these unbalanced forces could cause abnormal curves and rotations found with AIS [2]. Another unknown factor in Adolescent idiopathic scoliosis is why some curves progress while others do not. Curves progress at various rates, and there are factors that help to determine if a curve may progress. These factors

include maturity level and curve magnitude [4]. Patients with more growth remaining have a greater chance of having their curves progress. One common indication of curve progression is the Risser sign, a grade of calcification of the iliac apophysis [5]. Risser grade 0 or 1 indicates that the apophysis is still open and the patient has about two years of growth remaining, while a Risser grade 5 indicates complete ossification of the iliac apophysis and the patient has little to no growth remaining [5]. A study by Lonestein and Carlson showed that a 20° curve was 22% more likely to progress if the Risser grade was 0 or 1 rather than, if the Risser grade was 4 or 5 [2]. Another study showed that a patient with a 40° curve had a 100 percent chance of curve progression if the patient was between 10 and 12, whereas a patient with the same curve who was 16 or older, had a 70 percent chance of curve progression [2]. The magnitude of the curve also factors into the possibility of curve progression; curves that are 30° or less by the time of maturity have a small chance of progressing [2]. Curves between 30 and 50 degrees will progress about 10 to 15 degrees over a patient's lifetime if the curve is left untreated, and curves that are over 50° will progress about one degree a year [2].

The diagnosis of adolescent idiopathic scoliosis is one of exclusion rather than inclusion, where all possible causes for the abnormal curvature of the spine must be excluded prior to making the diagnosis of idiopathic scoliosis. Most commonly patients notice an asymmetry in their posture, one shoulder or hip is higher than the other, or they are experiencing some back pain [4]. A physician tests for abnormal curvature of the spine by the Adam's forward bending test in which the patient bends forward, and the physician can determine if there is an asymmetry in the scapula and abnormal curves in the spine [2, 4]. The final diagnosis for scoliosis must be done using an initial

radiographic series which consists of a standing lateral and posterior-anterior (PA) radiograph and side bending radiographs. On the standing PA radiograph, the curves must measure 10° or greater to have a conclusive diagnosis of scoliosis [2,4,6].

Radiographic measures are the most common ways of diagnosing and tracking the progression of scoliosis. There are five measures taken from the standing PA and lateral radiographs these are: the sagittal vertical alignment (SVA), the coronal vertical alignment (CVA), Cobb angles on the coronal view, and the measures of kyphosis and lordosis from the sagittal view. The sagittal vertical alignment is taken from the lateral radiograph and describes the global alignment of the spine in the sagittal plane. The SVA is measured by drawing a vertical line down from the center of C7 and drawing a second line up from the posterior superior corner of the first sacral vertebra [7]. The horizontal distance between these two lines is the SVA as shown in Figure 3. A normal SVA is around 0 or slightly negative indicating that C7 is slightly behind the first sacral vertebra [7].



Figure 3: Sagittal vertical alignment

The coronal vertical alignment is very similar to the sagittal vertical alignment. The CVA is taken from the PA radiograph and describes the global alignment of the spine in the coronal plane. The CVA is measured by drawing a vertical line down from the center of the C7 vertebral body and up from the center of the first sacral vertebra [8]. The horizontal distance between these two lines is the CVA as shown in Figure 4. The normal CVA is 0 and a negative shift indicates C7 is to the left of the first sacral vertebra [8].



Figure 4: Coronal vertical alignment

The Cobb angles are measured from the PA radiograph and measure the amount of curvature of the spine in the coronal plane. The Cobb angle is measured by first finding the vertebra that is the apex of the curve, this is commonly the most displaced and rotated vertebra with the least tilt. The two end vertebrae are then found; these are usually the least rotated and displaced but have the greatest tilt [8]. A line is then drawn along the superior aspect of the upper end vertebra at the same angle as the vertebral body's tilt angle and a second line is drawn along the inferior aspect of the lower end vertebra in the

same manner. Two lines are drawn at 90° angles to these two lines, and the Cobb angle is measured as the interior angle created by the intersection of these two lines as shown in Figure 5 [8].

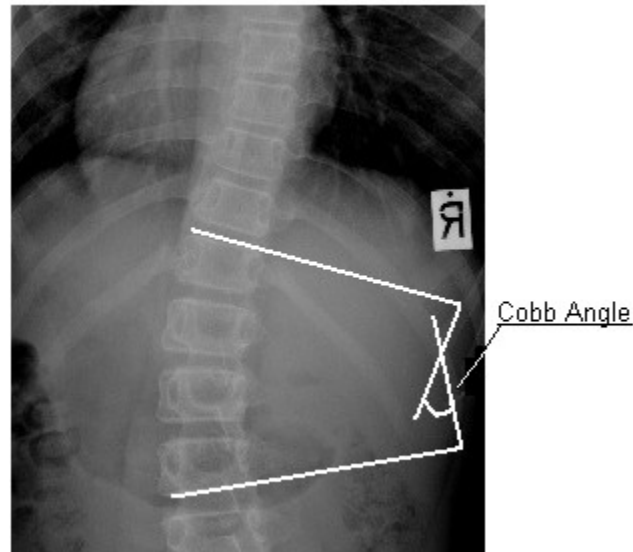


Figure 5: Cobb angle

The angles for lordosis and kyphosis are measured in the exact same way as a Cobb angle, the only difference is these angles are measured on the lateral radiograph.

There are two methods of treating scoliosis. The first method is a non-surgical means of slowing curve progression. This option often includes the use of braces in which the patient wears the brace for 16 to 23 hours a day. Bracing is usually considered for patients whose curves are between 25° and 40° and the patients have growth remaining [9]. The other option for treating scoliosis is by surgical means; this option often requires a patient's spine to be fused along the structural curves using rods, hooks, screws, and/or wires. Depending on the severity of the curve and curve location an anterior or a posterior approach can be performed [9]. Although surgical options are advised for curves 50° and greater, only about 10 percent of patients diagnosed with AIS

require surgical intervention [4, 6]. The surgical option will correct the scoliotic curves, and will level the asymmetries found in the hips and shoulders; however, the tradeoff is a decrease in spinal range of motion.

1.1.2 Radiation exposure

The most reliable means for diagnosing scoliosis and tracking curve progression is the use of radiographs to assess the spine in both the coronal and sagittal planes [6]. Researchers and medical personnel are concerned about the risk of radiation exposure to these adolescent patients due to the number of radiographs that a scoliosis patient receives over the course of their treatment [9]. Their concern comes from the fact that children and adolescents are more sensitive to radiation than adults [9]. Over the course of treatment a patient can receive upwards of fifteen radiographic evaluations that consist of a PA and lateral radiograph [6, 10, 11].

As a result of these concerns, many researchers have looked into trying to determine the risks of radiation exposure for these patients, specifically looking at the risk of radiation induced cancers and genetic risks [6, 10]. In one study 13 females with adolescent idiopathic scoliosis were followed through the course of treatment. In this study, surface radiation exposures were measured and these values were compared to the American Cancer Society's radiation exposure guidelines. The results of this study showed that scoliosis patients had a 7.5 percent greater risk for lung cancer, 110 percent greater risk of breast cancer and a 1 percent increased risk of other types of cancers [10]. As a result of this study the coronal radiograph was changed from an anterior-posterior view to a posterior-anterior view to reduce the radiation exposure to the breast tissue. This change reduced the risk of breast cancer twelve fold for these patients [11]. Another

study looked into the risk of radiographs to scoliosis patients. This study was performed about 30 years after the first study and after X-ray technology had improved, reducing the risk of radiation exposure for the patient. In this study 93 female AIS patients were followed over the course of their treatment and the surface radiation was measured and the radiation absorption of specific organs was also calculated [6]. The results of this study showed that AIS patients had a 4.2 percent greater risk of radiation induced breast cancer than the normal population and a 2.1 percent increased risk of breast cancer [6]. The results also showed that the scoliosis population has a one percent greater risk of developing leukemia than the normal population [6]. Studies have also shown that scoliosis patients have a three percent greater risk of birth defects in their children; the most common birth complications included premature births and low birth weights [6, 12].

1.1.3 Motion Analysis

There are a number of motion analysis techniques used to measure the kinematics of the human body. These techniques include goniometric measurements, the use of accelerometers, film digitization, and optoelectronic motion analysis. The scope of this work has been limited to optoelectronic motion analysis.

Optoelectronic motion capture systems track the three dimensional movements of a person in space using markers affixed to various bony landmarks [13]. These markers can be either passive, retro-reflective markers that reflect infrared light back to motion cameras to track the movement of various body segments or active markers. Active markers are LED diodes placed on the patient and instead of reflecting light back to the

motion cameras; these active markers produce the light that is tracked by the motion cameras [13].

Motion analysis systems are used in a variety of disciplines including medical science and sports biomechanics [14]. In medical science most of the research revolves around clinical gait analysis in which markers are placed on a patient and the patient is asked to walk through a calibrated area. The markers along with the motion camera system are capable of tracking the positions of the various markers, and computer programs will create body segments from which the kinematic description of human gait can be determined. In sports biomechanics, the same systems are used with more complex marker sets, and are used to track the motion of various sports activities, such as baseball pitching, jumping tasks, and spinning sports such as figure skating and gymnastics, in an effort to improve performance and reduce injury [14].

The calculations that are commonly used in optoelectronic motion analysis were published by Kadaba et al. [15]. In Kadaba's work he describes in detail the methods of determining segment rotations based on a surface marker configuration. Motion analysis models assume that each body segment is a rigid body whose relative rotations occur around a single point at the center of the joint [15]. Using a global coordinate system based on the configuration of a motion laboratory and local coordinate system defined by a given marker configuration it is possible to describe the motion of each joint using Euler rotation sequences [15]. Customarily the global coordinate system is defined as the x-axis point parallel to the walkway in a motion laboratory, the z-axis pointing in the vertical direction and the y-axis being perpendicular to both. The primary axes for the local coordinate systems are commonly defined as the axis in which the initial motion

would rotate around [15]. To accurately describe the relative rotations of the body segments and joints, it is essential that the coordinate systems be accurately defined.

While motion analysis can provide a great deal of information about the motion of the human body in three dimensional space there are a number of drawbacks. The first is that skin motion artifact caused by the movement of the skin and underlying musculature can introduce noise into the data [14]. Another drawback is the location of the markers that are used to track the motion. These markers must be placed on specific bony landmarks, and deviations in these positions can create inaccurate data; since more than one clinician will often place markers, intra and inter-clinician placement of markers may introduce artifact as well [14].

A number of researchers have begun to use motion analysis techniques and design marker sets to describe spinal ranges of motion and the standing posture of those affected by scoliosis. The marker sets used to define the motion segments must be clinically feasible, in that they can easily be used in a clinical setting using the minimum number of markers required, while providing enough information to correctly describe the motion in question. Many of these researchers looked into how spinal morphology affected gait. Frigo et al. studied the motion pattern of the trunk and shoulders during level gait, using a marker system specifically designed to look at the movement of spinal segments and calculate various static measures [16]. Frigo's marker set as shown in Figure 6 used 15mm diameter markers placed on C7, T6, and L3 vertebra, also on the right and left posterior superior iliac spine, and right and left acromions [16]. Markers were also placed an equal distance between C7 and T6, and two additional markers were placed equal distance between T6 and L3. These additional markers were not placed on

specific bony landmarks but served to aid the graphic representation of the spine. Using this marker set Frigo et al. calculated the angles of kyphosis, lordosis, and coronal plane angles to represent Cobb angle measurements, as well as shoulder tilt [16]. In the work by Frigo et al. work 18 female patients between the ages of 10 and 18 were asked to walk through a collection space to measure the motion of the spine during normal gait. Frigo concluded that spine angles in a normal population showed a deviation of 5°.

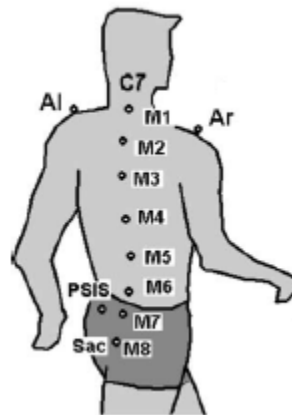


Figure 6: The marker configuration by Frigo [16]

Engsberg et al. focused on the spinal range of motion pre and post spinal fusion [17]. Engsberg's marker set, shown in Figure 7, combined a clinical gait marker set along with additional markers placed along the spine; however, his work focused only on the ten retro-reflective markers placed on the right and left acromions, right and left anterior superior iliac spines, C7, T7, T10, L4, and the first sacral vertebra [17]. Using this marker set Engsberg et al. was able to measure the global motion of the spine as well as measuring the motion in specific regions of the spine.

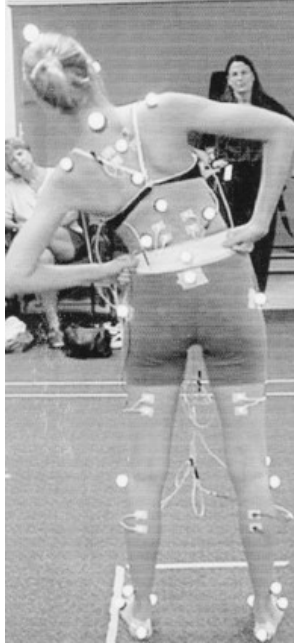


Figure 7: The marker configuration by Engsberg [17]

In Engsberg's study 30 patients with scoliosis had a motion analysis two months prior to a spinal fusion to correct their scoliosis. These 30 patients were then tested again one year and two years after their spinal fusion to determine the loss in spinal motion caused by spine fusion [17]. Each of the patients went through a series of bending exercises which included spine flexion and extension, lateral bending to both the right and left side, and axial rotation of the spine. Motion data was available for only 24 of the 30 patients due to marker obstruction [17]. The results of Engsberg's work showed a significant reduction in lateral motion of the spine as well as a reduction in spine flexion, however he noted there was little change in the range of motion in the segments that were not fused [17].

Cerveri et al. designed a marker set to measure the rotational aspects of the vertebra in the lumbar region of the spine [18]. To accomplish this Cerveri et al. used 28, 2mm diameter markers placed on the spinous process, and transverse processes of T11

through S1 and the iliac crests as seen in Figure 8 [18]. Using this marker configuration Cerveri was able to describe the motion of the lumbar vertebra.



Figure 8: The marker configuration by Cerveri [18]

Cerveri found that while his marker set was capable of describing the general motion of the spine, it was far more difficult to determine the intervertebral motion due to skin movement artifact [18]. Cerveri also found that rotational motion is often coupled with some lateral bending and subsequent marker configurations and model must be capable of taking this potentially confounding motion into account.

Masso and Gorton used a marker set to calculate the static posture of patients with scoliosis [19]. The marker set, shown in Figure 9, placed a marker on the right and left acromion, the right and left anterior superior iliac spine, the ninth rib on the left and right side, C7, the first sacral vertebra, the right and left thighs, knees, ankles, and foot [19]. Using this marker configuration Masso and Gorton calculated the standing posture of scoliosis patients. These measures included the patient's center of pressure, their sagittal vertical alignment, their coronal vertical alignment and the coronal and transverse asymmetries of the shoulders, pelvis, and hips [19].



Figure 9: The marker configuration by Masso and Gorton [19]

Skalli et al. investigated how the pelvis compensates for the loss of spine motion caused by posterior spinal fusion in patients with Idiopathic Scoliosis [20]. In Skalli's work 30 patients undergoing spinal fusion for idiopathic scoliosis had a comprehensive motion analysis pre and post operatively. A control group made up of 32 patients with radiographically normal spines was also tested to serve as a reference [20]. The marker configuration was similar to that of Masso and Gorton, markers were placed on the right and left acromion, right and left anterior superior iliac spines, as well as on C7, S1, and the first vertebra below the instrumentation. The patients in both the control and test groups underwent both static and dynamic analysis. The dynamic analysis included flexion and extension of the spine, lateral bending, and axial rotation [20]. The results of Skalli's work showed significant differences in the pre-operative spine range of motion in the flexion task between the test and control group. Skalli also noted that the pelvis helps to compensate for reduced spinal motion in patients with idiopathic scoliosis both prior to and after posterior spinal fusion [20].

1.2 Pilot Work

Pilot work on this topic included designing a marker set to allow for static calculations that would help to describe the spinal and asymmetric deformities in each of the anatomic planes. A Matlab program was created to calculate the spine angles in the coronal and sagittal planes, to calculate the center of pressure to describe the patient's posture, and to calculate the asymmetries of the shoulders, thorax, and pelvis. The program was tested using a normal subject to validate the program.

The results of this trial were calculated in Matlab and output into Excel. The asymmetry calculations, those that describe the coronal plane and transverse asymmetry of the shoulders, mid-thorax, and pelvis were checked visually. The results for the one trial calculated by Matlab shows that all measurements are comparable to those visually assessed as seen in Table 1.

Table 1: Asymmetry results for the Coronal and Rotational plane for the shoulders, pelvis, and mid-thorax of the preliminary work.

Anatomic Level		Transverse Plane	Frontal Plane
Shoulder	Mean	0	-1
	Max	1	-1
	Min	-2	-2
Mid-Thorax	Mean	0	2
	Max	2	3
	Min	-1	2
Pelvis	Mean	2	0
	Max	4	0
	Min	1	0

The results for the center of pressure calculations were also calculated in Matlab and output to Excel seen in Table 2. These measurements were compared to a known value for the center of pressure. A person with a normal spine should have a center of pressure value close to 50 percent. This indicates that the center of pressure is running directly between the right and left markers placed at the shoulders, mid-thorax, and pelvis. A value less than 50 percent would indicate that the center of pressure line is skewed to the right.

Table 2: Center of Pressure values

Anatomic Level		COP
Shoulder	Mean	46.9
	Max	54.0
	Min	44.0
Thorax	Mean	45.3
	Max	51.5
	Min	42.2
Pelvis	Mean	45.6
	Max	52.7
	Min	42.9

Matlab was used to calculate the CVA, SVA, angles of kyphosis and lordosis, and proximal and distal angles. These measures were then checked against MRI values taken from the subject's MRI that was on file seen in Table 3.

Table 3: MRI measures compared to Matlab calculations

Measure	Calculated	Radiograph	Difference
SVA (mm)	63	41.1	21.9
CVA (mm)	2	1.1	0.9
Kyphosis (°)	5	19	-14
Lordosis (°)	21	23	-2
Proximal Angle (°)	4	3.3	0.7
Distal Angle (°)	4	2.3	1.7

The results of this pilot work showed some of the differences between the radiographic measures and calculated measures. These differences could be caused because the subject's MRI was taken in a supine position whereas most scoliosis films are taken from a standing position; therefore, some of the radiographic measures may be slightly different than what would commonly be seen on a spinal radiograph. The radiographic measures are taken directly from the spine, whereas the Matlab calculations are dependent upon the position of the markers placed on the surface of the skin. Some of the measures may be different as a result of the thickness of the soft tissue covering the spine. The results of this pilot work were taken into account in the current work, in an effort to increase the accuracy of the Matlab calculations.

2.0 Methods

This work was approved by the Connecticut Children's Medical Center's IRB (09-049) and performed at the Center for Motion Analysis at the Connecticut Children's Medical Center in Farmington, Connecticut. Prior to the evaluation at the Center for Motion Analysis the protocol was explained to the subjects and their guardians, and both the subjects and guardians signed a consent form.

The materials available at the Center for Motion Analysis includes a Vicon 512 motion capture system (Vicon, Los Angeles, CA). This motion capture system utilizes 12 infrared motion cameras that are capable of accurately measuring planar distances between two 14mm markers to within one millimeter, and can accurately measure the angle between three markers to within one degree. The motion system collects at 120 Hz and has a calibrated collection space that is 5 by 1.5 by 2.5m. The Center for Motion

Analysis also has three, six-axis AMTI force platforms (AMTI, Watertown, MA), 500mm by 500mm, analog force platforms that have low pass Butterworth filters with a 160Hz cutoff frequency. The force platforms have a force resolution of 0.18N along the x and y axes and 0.72N along the z axis, and a moment resolution of 0.72 Nm along the x axis, and 0.07Nm along the y and z axes.

2.1 Marker Configuration

The marker configuration used in this work combines elements of the marker sets proposed by Frigo et al. [15] and Masso and Gorton [19]. The current marker set (Figure 10) utilizes 20, 14mm retro-reflective markers placed on specific bony landmarks. The marker set was slightly modified from the pilot work on this topic in order to make the marker set more repeatable from subject to subject. The original marker set placed the markers more laterally on the rib cage. However, after some consideration it was determined that abnormal rotation of the rib cage, caused by the scoliosis, could cause laterally placed markers to become obscured during the evaluation. This led to moving the markers to their current location which is more medial.



Figure 10: The marker set in this work

The upper thorax was described using three markers. These markers were placed on the right and left clavicles at one to two inches lateral from the sternal notch and one marker was placed on the spinous process of C7. The locations were chosen for their easily palpable landmarks and since C7 is a radiographic landmark used to calculate the SVA and CVA measurements. The mid-thoracic plane is created by placing markers on the 10th rib on the right and left side, superior and medial of the floating ribs, and by placing one marker on the spinous process of T7. These markers were chosen for the mid-thoracic plane, since the base of the rib cage is easily located, even if it is rotated due to spinal deformity. The T7 vertebra was chosen, since this is usually the apex of the thoracic kyphosis. Finally, the pelvic plane is constructed using markers placed on the right and left anterior superior iliac spines (ASIS) and the first sacral vertebra at an equal distance between the two posterior superior iliac spines. The marker locations for the pelvis were chosen because they are easily located in a clinical setting and have been used in numerous marker sets that are based on the Helen Hays marker configuration. An additional marker was placed on the spinous process of L3 (i.e., apex of the lumbar lordosis). Marker wands were also placed on the right and left thighs and shanks. Markers were also placed on the right and left lateral knee, right and left lateral malleoli, and the right and left foot. The lower extremity markers were not used in the current work, but were included for possible future work, which will consider the motion of the hips or knees in patients with scoliosis.

Using this marker configuration, it was possible to measure the pelvic tilt, obliquity, and axial rotation to determine the pelvic asymmetry and the asymmetries of the thorax and shoulder. These markers also allowed for the calculation of the sagittal

vertical alignment, coronal vertical alignment, and angles of kyphosis and lordosis. The marker set was designed to produce measures similar to Cobb angle measures using the upper and lower Cobb angle. Finally, the positional data this marker set and data from the force platforms allowed for the calculation of a subject's center of pressure while standing in a relaxed static posture.

2.2 X-ray frame

In an effort to standardize the standing posture of the subjects during the lateral radiograph, a positioning X-ray frame was designed and constructed to place subjects in a repeatable position as shown in figure 11. The subject stood on the base of the frame, and the adjustable bar was moved up to the zyphoid process. The subject was then asked to stand with their elbows placed on the bar.

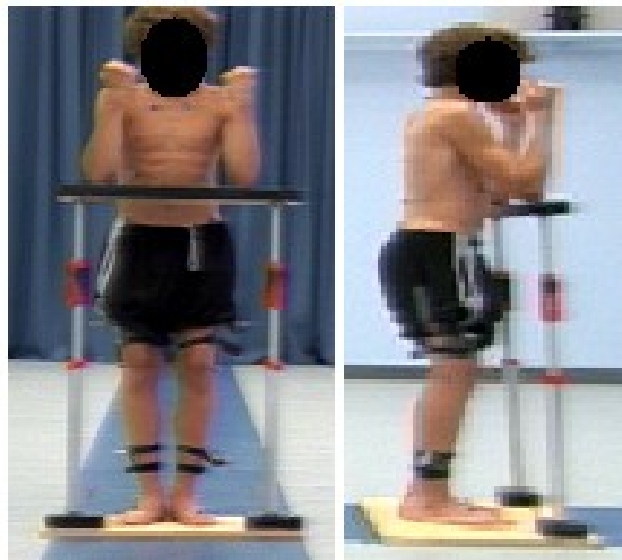


Figure 11: X-ray positioning frame

The x-ray frame was designed with four criteria. The height of the bar had to be adjustable, since the population this frame was used on adolescents of varying heights. It was assumed that the height of the zyphoid process varied from to 2ft to 5ft, and

expandable painter's poles were used to allow for adjustability. Fifteen additional 1/4th inch diameter holes were drilled into the painter's poles to ensure that the height adjustments could be made in two inch increments. The additional holes were made at the same level on both poles to ensure that the cross bar would remain level. Since the subject was instructed to place their elbows on the bar at shoulder width, the frame was designed to be 24 inches wide to account for width variation. Figure 12 shows the schematic diagram for the frame.

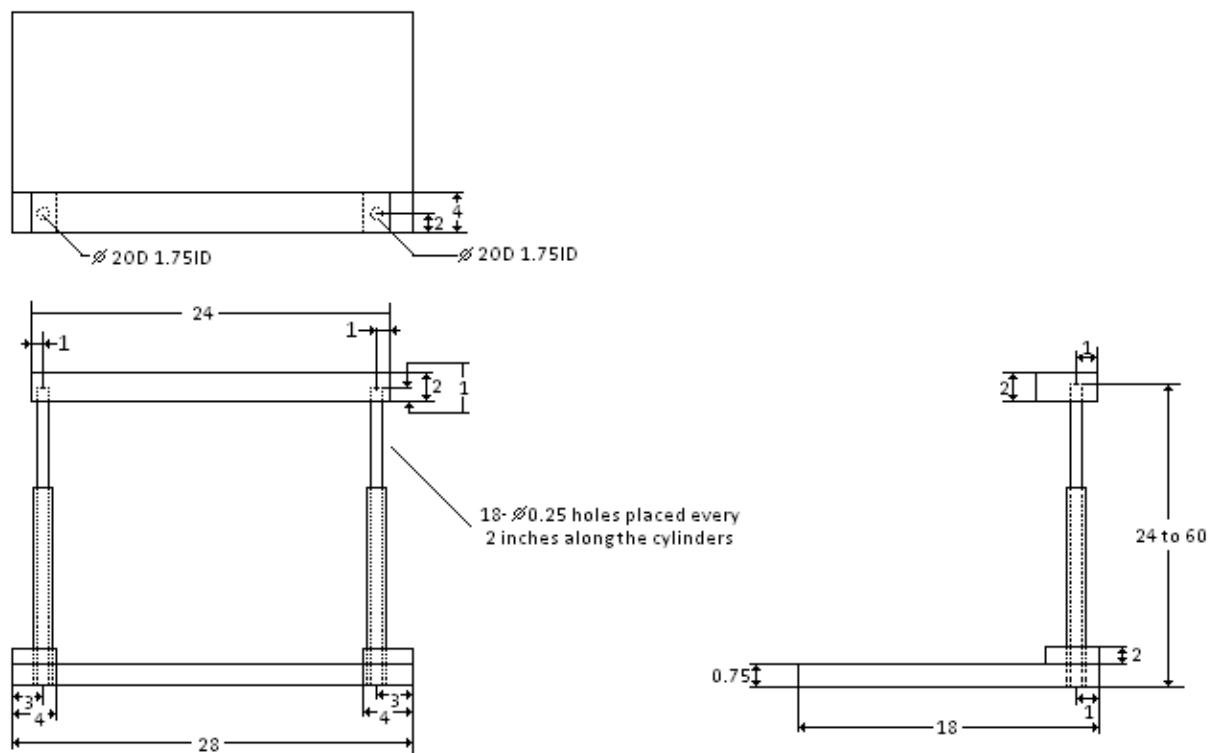


Figure 12: Schematics for the x-ray frame (all units in inches)

The frame was used for all patients receiving lateral full spine radiographs, regardless if they were participants. To ensure that the frame was both durable and stable the adjustable painter's poles were constructed out of aluminum painter's poles and the base and crossbar were constructed out of 3/4th inch medium density fiberboard (MDF)

and a pressure treated 2x4 wood plank, respectively. All of the joints were secured using two inch decking screws and epoxy. The laminated MDF for the base was chosen for its finish. The finish is smooth and free of splinters, since subjects are x-rayed barefoot for spine radiographs. The laminate also allowed for the surface to be easily cleaned. The 2x4 crossbar supports the weight of the subject's arms during the x-ray and for safety, the bar was carefully sanded painted using an acrylic paint to seal the wood.

The frame could be positioned quickly and easily so as not to add time to the radiography protocol. More specifically the painter's poles allowed for quick adjustment with the radiology technician pushing in on the handles and pulling up or down to set the height. The adjustment procedure takes no more than 30 seconds.

Finally a total budget of \$150 was provided for frame construction and a total of four frames had to be constructed for use at the various Connecticut Children's Medical Center's satellites that commonly perform scoliosis radiographs. The final total cost for the four frames was \$116, with the greatest cost coming from the painter's poles.

Two designs were prototyped prior to choosing the frame used in this work. The first frame was constructed out of PVC pipe; however, the design was rejected because it was difficult to position, was very unstable, and often fell apart during use. The second prototype was very similar to the final design, but used telescoping monopoles instead of the painter's poles. The monopoles allowed for greater adjustability but were unable to support varying arm weight.

2.3 Subjects

A total of ten subjects were used in this work: five with a normal spine and five with a diagnosis of Adolescent Idiopathic Scoliosis (AIS). The subjects were all seen at one of

Connecticut Children's Medical Center's Orthopaedic clinics for a scoliosis evaluation, which included an initial series of diagnostic radiography. Inclusion criteria for this study required a normal spine for the control group and a diagnosis of AIS, with the main structural curve measuring at least 10° on the posterior-anterior radiograph for the study group. Subjects were excluded from this study if they did not have full length coronal and sagittal plane spinal radiographs. Subjects were also excluded if they had a body mass index of 30 or greater, which could affect the placement of the markers and the palpation for the necessary bony landmarks due to large amounts of adipose tissue. A significant amount of adipose tissue will produce a skin motion artifact that will influence motion data results. Finally, subjects were also excluded if they had a secondary diagnosis, such as a neuromuscular disorder.

2.4 Subject testing

Prior to participation in the motion analysis, all subjects received a full series of scoliosis radiography, which included a posterior-anterior, PA, radiograph and a lateral radiographs. The subject was asked to stand on a force plate, facing in the positive x-direction within the lab space (Figure 13).

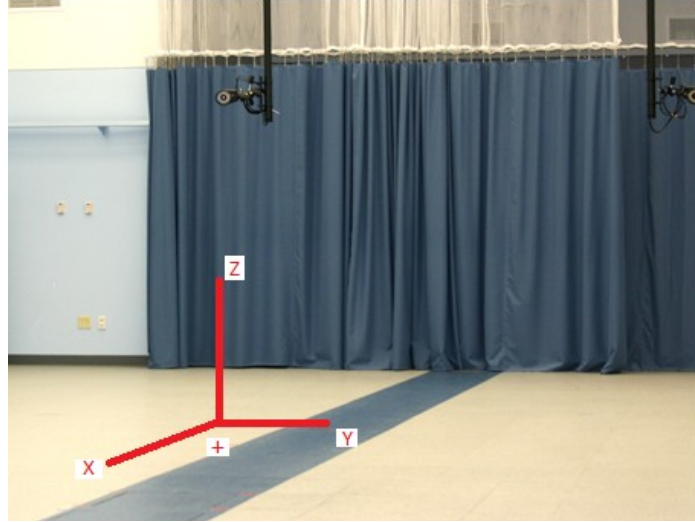


Figure 13: Global coordinate system

The static portion of the data collection was initiated and the subject was asked to step forward onto a force plate and remain standing the force plate in a relaxed posture for 10 seconds. Once the data collection was completed, the markers were identified using Vicon Workstation software (Vicon, Los Angeles, CA) and the first five seconds of the trial were removed to reduce the motion of the subjects settling into position. This was repeated three times. The subject was then asked to stand in the X-ray frame and the device was set up using the established protocol. Data was collected for 10 seconds and trials with the X-ray frame were repeated three times. The first two seconds of each trial were removed to reduce system artifact that could influence the static calculations.

The dynamic portion incorporated a series of maximal bending tasks. The first bending task required the subject to laterally bend a lateral bend to the left is illustrated in figure 14a, the second task required the subject to axially rotate and a rotation to the left is depicted in figure 14b, and the final task required the subject to perform a trunk flexion and extension and example of trunk extension is shown in figure 14c.

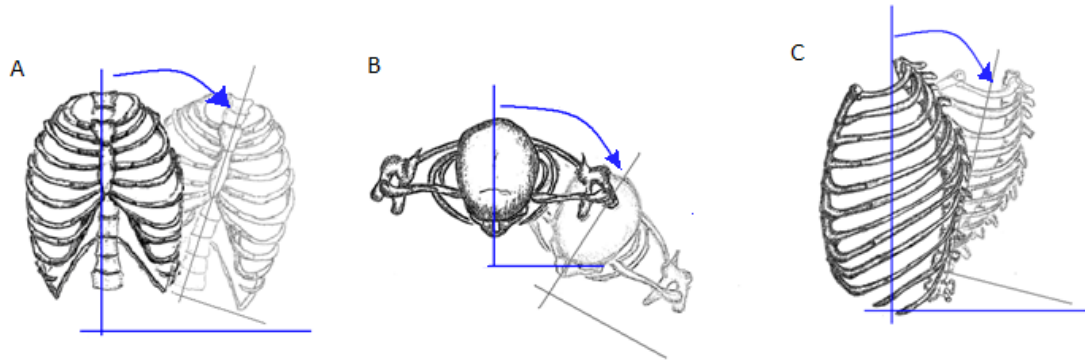


Figure 14: Illustrations of the bending tasks.

Each of the motion tasks were repeated three times. Prior to performing any of the bending tasks the subjects were given instructions on how to perform the task, and the task was demonstrated. During the collection, the patients were instructed on which side to bend toward, in an effort to standardize the direction and timing of the tasks. The subject's were also instructed to perform the bending task to the best of their ability. All of the motion data was saved to a C3D file using the Vicon workstation software and imported into a custom written Matlab (Mathworks, Natick, MA) program, where the calculations were performed. The static portion of the test allowed for the calculations of frontal and transverse plane asymmetries and for the center of pressure of the subject. The static portion of the test also allowed for the SVA, CVA, proximal and distal angles, and angles of kyphosis and lordosis to be calculated. The dynamic portion of the test allowed for the calculations of the global spine range of motion in all three planes, as well as the calculations for spine range of motion for the upper spine, the seventh cervical vertebra to the seventh thoracic vertebra, the lower portion of the spine, and the seventh thoracic vertebra to the first sacral vertebra.

2.5 Matlab

The Matlab program, used in this work, which can be seen in Appendix A, relies on a structured architecture and the equations in the following sections. The structured architecture in Matlab is a means of organizing and storing multiple data arrays within an organized hierarchy. The program consists of nine subroutines and three function callouts, as depicted in Figure 15.

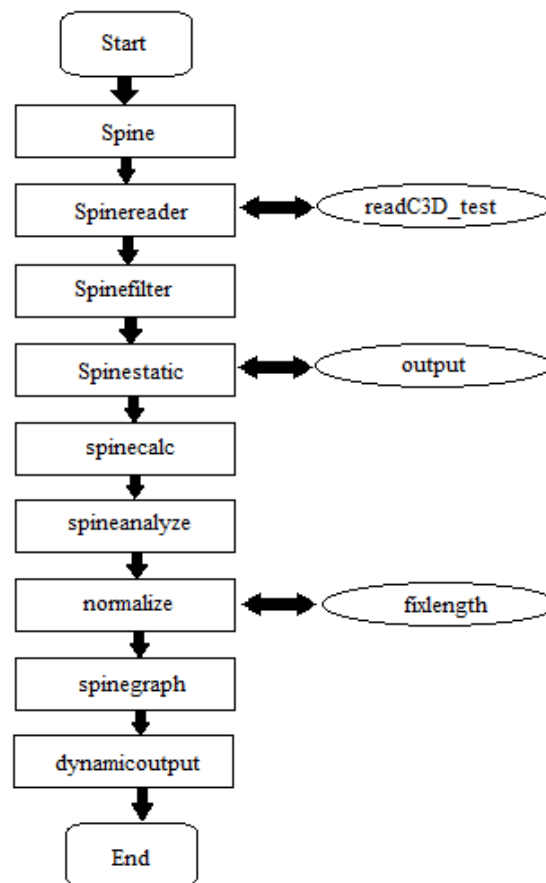


Figure 15: Program flow chart

There are two additional subroutines that are not depicted in Figure 15. These two subroutines contain code to assemble variables from multiple subjects to create a group mean and are run after the end of the main program.

The program called *spine* serves as the main program and launches the program in the Matlab environment. It asks for user input to find where the motion files are located on the computer. *Spinereader* is a subroutine that reads the C3D file output by the motion analysis software and relies on the function *readC3D_test* written by Jaap Haarlar [21]. Once the function is called, the program reads the C3D file into the Matlab environment and the *spinereader* code organizes the motion data into a series of structures and substructures to be called in the later subroutines. *Spinereader* also calculates the centers of each force platform and ground reaction forces required for the center of pressure calculations described in the following sections. Once the data is structured, the remaining subroutines manipulate the data to produce the desired outputs. *Spinefilter* is a subroutine that uses a fourth order Butterworth filter to remove noise within the motion capture system and to reduce skin motion artifact. It is important to note that the coding within *spinefilter* requires the use of the signal analysis and filtering toolboxes Matlab. *Spinestatic* is the subroutine responsible for calculating all of the variables required for the static analysis portion of this work and the relevant equations are discussed in detail in the following section. After the static variables have been calculated, the program calls the function output and exports all of the static data to an Excel worksheet.

The dynamic calculations are performed after the static analysis. All of dynamic calculations are embedded in the *spinecalc code*, including the creation of the motion segments and the calculation of Euler rotation sequences. The equations in this code are also described in the following sections. The next subroutine, *spineanalyze*, reads the dynamic variables and separates the bending tasks using the event times written into the C3D file and calculates the endpoints of the motion and the maximum and minimum

angle values. The dynamic variables are then saved into the structure and the program diverges into two paths. The first path exports the dynamic data calculated in *spineanalyze* to an Excel worksheet using the subroutine *dynamicoutput*, while the other path sets up the dynamic variables for plotting. To accurately compare the kinematic profiles for each of the bending motions, the plots were created to have the same time scaling and the subroutine *normalize* takes the frame data and down samples the data to fewer than 100 total frames. The subroutine then calls the *fixlength* function and increases the time scaling to 100 and performs a fourth order Butterworth filtering operation to remove noise introduced during the down and up sampling. Once the time scale has been normalized, the kinematic data is plotted with the *spinegraph* code and the motion of each segment in the coronal, sagittal, and transverse planes is provided.

2.6 Static Calculations

The calculations used to describe the static and postural characteristics of both the normal and pathological spine were based on the global coordinates of the Center for Motion Analysis lab. The sagittal vertical alignment given in Equation 1 was calculated by subtracting the average x-axis position of the sacral marker from the average x-axis position of the C7 marker. This equation also included a correction factor to remove the distance of the marker from the skin.

$$SVA = (C7_x - M_{rad}) - (S1_x - M_{rad}) \quad (1)$$

Where $C7_x$ is the average x-axis marker position within the global coordinated system for the C7 vertebra, $S1_x$ is the average x-axis marker position within the global coordinate

system for the S1 vertebra, and M_{rad} is the radius of the markers. If C7 is in front of the first sacral vertebra then the SVA is positive.

The coronal vertical alignment (CVA) which is given in Equation 2 was calculated by subtracting the average y-axis position of the sacral marker from the average y-axis position of the C7 marker.

$$CVA = C7_y - S1_y \quad (2)$$

Where $C7_y$ is the average y-axis marker position within the global coordinated system for the C7 vertebra, $S1_y$ is the average y-axis marker position within the global coordinate system for the S1 vertebra. If C7 is to the right of the first sacral vertebra then the CVA is positive.

The angle of kyphosis was measured in the sagittal plane using the positional data from the C7, T7, and L3 markers in Equations 3a, 3b, 3c, 3d and 3e. The average positions of these markers were ultimately used and were calculated used with the x-z coordinates for each of the markers within the global coordinate system.

$$distCT = \sqrt{(C7_x - T7_x)^2 + (C7_z - T7_z)^2} \quad (3a)$$

$$\theta_1 = \sin^{-1} \left(\frac{C7_x - T7_x}{distCT} \right) \quad (3b)$$

$$distTL = \sqrt{(T7_x - L3_x)^2 + (T7_z - L3_z)^2} \quad (3c)$$

$$\theta_2 = \sin^{-1} \left(\frac{T7_x - L3_x}{distTL} \right) \quad (3d)$$

$$Kyphosis = \theta_1 + \theta_2 \quad (3e)$$

Where $C7_x$, $T7_x$, and $L3_x$, are the average x-axis positions for the C7, T7, and L3 vertebrae respectively and $C7_z$, $T7_z$, and $L3_z$ are the average z-axis positions for the C7,

T7, and L3 vertebrae, respectively. Also $distCT$ is the distance between the C7 and T7 markers, $distTL$ is the distance between the T7 and L3 markers, θ_1 is the angle between the C7 and T7 markers, and θ_2 is the angle between the T7 and L3 markers. The angle was broken into two separate angles and the resulting angles were added to find the angle of kyphosis as shown in Figure 16.

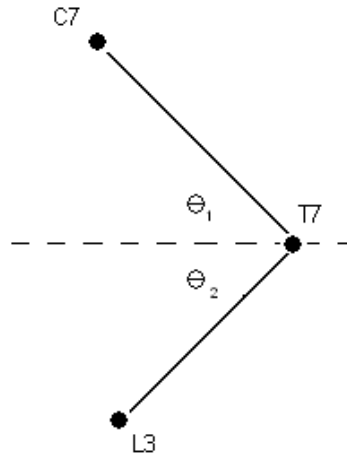


Figure 16: The angles used to determine the angle of kyphosis

The angle of lordosis calculated using Equations 4a, 4b, 4c, 4d, and 4e was measured using the positional data from the T7, L3, and S1 markers, and was calculated in the same manner as the angle of kyphosis.

$$distTL = \sqrt{(T7_x - L3_x)^2 + (T7_z - L3_z)^2} \quad (4a)$$

$$\theta_1 = \sin^{-1} \left(\frac{T7_x - L3_x}{distTL} \right) \quad (4b)$$

$$distLS = \sqrt{(L3_x - S1_x)^2 + (L3_z - S1_z)^2} \quad (4c)$$

$$\theta_2 = \sin^{-1} \left(\frac{L3_x - S1_x}{distLS} \right) \quad (4d)$$

$$Lordosis = \theta_1 + \theta_2 \quad (4e)$$

In Equations 4a, 4b, 4c, 4d, and 4e $T7x$, $L3x$, and $S1x$, are the average x-axis positions for the T7, L3, and S1 vertebrae respectively and $T7z$, $L3z$, and $S1z$ are the average z-axis positions for the T7, L3, and S1 vertebrae, respectively. Also $distTL$ is the distance between the T7 and L3 markers, $distLS$ is the distance between the L3 and S1 markers, θ_1 is the angle between the T7 and L3 markers, and θ_2 is the angle between the L3 and S1 markers.

The proximal estimation for the Cobb angle calculated using Equations 5a, 5b, 5c, 5d, and 5e was measured in the coronal plane using the C7, T7, and L3 markers. Similar to the angle of kyphosis calculation, the average positions of these markers were used and were calculated using the y-z coordinates for each of the markers within the global coordinate system.

$$distCT = \sqrt{(C7_y - T7_y)^2 + (C7_z - T7_z)^2} \quad (5a)$$

$$\theta_1 = \sin^{-1} \left(\frac{C7_y - T7_y}{distCT} \right) \quad (5b)$$

$$distTL = \sqrt{(T7_y - L3_y)^2 + (T7_z - L3_z)^2} \quad (5c)$$

$$\theta_2 = \sin^{-1} \left(\frac{T7_y - L3_y}{distTL} \right) \quad (5d)$$

$$Proximal = \theta_1 + \theta_2 \quad (5e)$$

In Equations 5a, 5b, 5c, 5d, and 5e, $C7y$, $T7y$, and $L3y$, are the average y-axis positions for the C7, T7, and L3 vertebrae respectively and $C7z$, $T7z$, and $L3z$ are the average z-axis positions for the C7, T7, and L3 vertebrae, respectively. Also $distCT$ is the distance between the C7 and T7 markers, $distTL$ is the distance between the T7 and L3 markers, θ_1 is the angle between the C7 and T7 markers, and θ_2 is the angle between the T7 and L3

markers. The angle was broken into two separate angles and the resulting angles were added to find the proximal angle as shown in Figure 17.

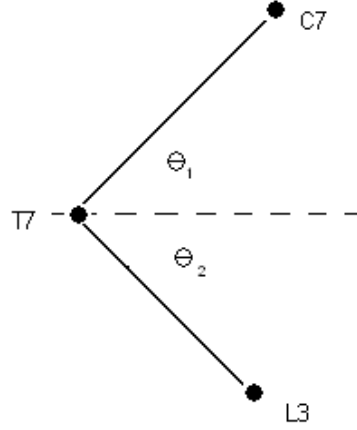


Figure 17: The angles used to determine the estimated proximal angle.

The distal estimation for the Cobb angle was calculated using Equations 6a, 6b, 6c, 6d, and 6e and measured in the coronal plane using the T7, L3, and S1 markers, and was calculated using the same method as the proximal estimation for the Cobb angle.

$$distTL = \sqrt{(T7_y - L3_y)^2 + (T7_z - L3_z)^2} \quad (6a)$$

$$\theta_1 = \sin^{-1} \left(\frac{T7_y - L3_y}{distTL} \right) \quad (6b)$$

$$distLS = \sqrt{(L3_y - S1_y)^2 + (L3_z - S1_z)^2} \quad (6c)$$

$$\theta_2 = \sin^{-1} \left(\frac{L3_y - S1_y}{distLS} \right) \quad (6d)$$

$$Distal = \theta_1 + \theta_2 \quad (6e)$$

In Equations 6a, 6b, 6c, 6d, and 6e, $T7_y$, $L3_y$, and $S1_y$, are the average y-axis positions for the T7, L3, and S1 vertebrae respectively and $T7_z$, $L3_z$, and $S1_z$ are the average z-axis positions for the T7, L3, and S1 vertebrae, respectively. Also $distTL$ is the distance

between the T7 and L3 markers, $distLS$ is the distance between the L3 and S1 markers, θ_1 is the angle between the T7 and L3 markers, and θ_2 is the angle between the L3 and S1 markers.

The rotational and frontal asymmetries for the shoulder and pelvis were calculated using the global coordinates of the markers. The frontal plane asymmetry for the shoulders was calculated using y-z coordinates of the two clavicle markers as given in Equations 7a and 7b and the degree of asymmetry was determined using the distance and angle and, based on these two markers.

$$distCLA = \sqrt{(RCLA_y - LCLA_y)^2 + (RCLA_z - LCLA_z)^2} \quad (7a)$$

$$\theta_{sho} = \sin^{-1} \left(\frac{RCLA_y - LCLA_y}{distCLA} \right) \quad (7b)$$

In Equations 7a and 7b, $RCLA_y$ and $LCLA_y$ are the y-axis positions of the right clavicle and left clavicle markers in global coordinates respectively, and $RCLA_z$ and $LCLA_z$ are the z-axis positions of the right clavicle and left clavicle markers, respectively, in global coordinates. Also $distCla$ is the coronal plane distance between the two clavicle markers, and θ_{sho} is the angle of asymmetry of the shoulder. If the value of θ_{sho} is positive then the right shoulder is higher than the left.

The frontal plane asymmetry for the pelvis was calculated using y-z coordinates of the two anterior superior iliac spine markers given by Equations 8a and 8b. This calculation was identical to the frontal asymmetry of the shoulders.

$$distPel = \sqrt{(RASI_y - LASI_y)^2 + (RASI_z - LASI_z)^2} \quad (8a)$$

$$\theta_{pel} = \sin^{-1} \left(\frac{RASI_y - LASI_y}{distCLA} \right) \quad (8b)$$

In Equations 8a and 8b, $RASI_y$ and $LASI_y$ are the y-axis positions of the right and left pelvic markers, respectively, in global coordinates, and $RASI_z$ and $LASI_z$ are the z-axis positions of the right and left pelvic markers, respectively, in global coordinates. Also $distPel$ is the coronal plane distance between the two pelvic markers, and θ_{pel} is the angle of asymmetry of the shoulder. If the value of θ_{pel} is positive then the right side of the pelvis is higher than the left.

The rotational plane asymmetry for the shoulders was calculated using x-y coordinates of the two clavicle markers and is given by Equations 9a and 9b. The distance between the two markers was found and then the angle was calculated to determine the rotational asymmetry.

$$distCLA = \sqrt{(RCLA_x - LCLA_x)^2 + (RCLA_y - LCLA_y)^2} \quad (9a)$$

$$\theta_{sho} = \sin^{-1} \left(\frac{RCLA_x - LCLA_x}{distCLA} \right) \quad (9b)$$

In Equations 9a and 9b, $RCLA_x$ is the x-axis position of the right clavicle marker in global coordinates, $RCLA_y$ is the y-axis position of the right clavicle marker in global coordinates, $LCLA_x$ is the x-axis position of the left clavicle marker in global coordinates, $LCLA_y$ is the y-axis position of the left clavicle marker in global coordinates, $distCla$ is the transverse plane distance between the two clavicle markers, and θ_{sho} is the angle of asymmetry of the shoulder. If the value of θ_{sho} is positive then the right shoulder is in front of the left.

The rotational plane asymmetry for the pelvis was calculated using x-y coordinates of the two anterior superior iliac spine markers given by Equations 10a and 10b. This calculation was similar to the rotational asymmetry of the shoulders.

$$distPel = \sqrt{(RASI_x - LASI_x)^2 + (RASI_y - LASI_y)^2} \quad (10a)$$

$$\theta_{pel} = \sin^{-1} \left(\frac{RASI_x - LASI_x}{distCLA} \right) \quad (10b)$$

In Equations 10a and 10b $RASI_x$ and $LASI_x$ are the x-axis position of the right pelvic and left pelvic markers, respectively, in global coordinates, $RASI_y$ and $LASI_y$ are the y-axis positions of the right and left pelvic markers, respectively, in global coordinates, $distPel$ is the transverse plane distance between the two pelvic markers, and θ_{pel} is the angle of asymmetry of the shoulder. If the value of θ_{pel} is positive then the right side of the pelvis is in front of the left.

The center of pressure calculation was dependent on the position of the force vector measured using the force platform. The first portion of this calculation required that the center of the plate be known. The force plate corners were known within the calibrated space at the Center for Motion Analysis and every morning a calibration frame is used to ensure the positions of the corners are accurate. These corners are essential to find the x and y coordinates for the center of the plate, calculated using Equations 11a and 11b, the z coordinate is assumed to be zero.

$$C_x = \frac{Ax_1 + (Ax_4 - Ax_1)}{2} + \frac{Ax_2 + (Ax_2 - Ax_3)}{2} \quad (11a)$$

$$C_y = \frac{Ay_1 + (Ay_4 - Ay_1)}{2} + \frac{Ay_2 + (Ay_2 - Ay_3)}{2} \quad (11b)$$

In Equation 11a, C_x is the x coordinate of the force plate in millimeters, and in Equation 11b, C_y is the y coordinate of the force plate in millimeters. In Equations 11a and 11b Ax_n is the x-coordinate of the corner, Ay_n is the y-coordinate of the corner, and n is the number of the corner.

The forces and moments on the plate were calculated from the raw data gathered by the force platform, using the values for the analog offset and analog scaling factors contained in the C3D file given by Equation 12a and 12b.

$$GRf = (F - Ao) \cdot As \quad (12a)$$

$$GRm = (M - Ao) \cdot As \quad (12b)$$

In Equations 12a and 12b, GRf is the ground reaction force, GRm is the ground reaction moment, F is the force in a particular direction, M is the moment in a particular direction, Ao is the analog offset, and As is the analog scaling factor.

The center of pressure on the plate was calculated in x and y coordinates in millimeters using the calculated ground reaction forces and moments given by Equations 13a and 13b.

$$Cop_x = \frac{GRm_x}{GRf_x} + Cf_x \quad (13a)$$

$$Cop_y = \frac{GRm_y}{GRf_y} + Cf_y \quad (13b)$$

In Equations 13a and 13b Cop_x and Cop_y are the x and y coordinates of the center of pressure, respectively. Also GRM_x is the ground reaction moment in the x-direction, GRM_y is the ground reaction moment in the y-direction, GRF_x is the ground reaction force in the x-direction, GRF_y is the ground reaction force in the y-direction, and Cf_x and Cf_y are the x and y coordinates for the center of the force plates, respectively. The center of pressure in the z direction is zero.

The center of pressure was used to determine where the center of pressure line bisected the vector created by the markers placed on the patient's shoulder, mid-thoracic

region, and pelvis. To accurately calculate the intersection of the center of pressure vector and the vector between the markers the patient's asymmetry needed to be accounted for and is given by Equation 14.

$$Corr = R_n - \left(\tan \left(\frac{R_n - L_n}{dist_n} \right) \right) \quad (14)$$

In Equation 14, R_n is the coordinate of the right marker, L_n is the coordinate of the left marker, and $dist_n$ is the asymmetric distance calculated using frontal plane asymmetry equations.

The center of pressure calculation was completed by determining how far the center of pressure was from the right side marker at the pelvis, mid-thorax, and shoulders given by Equations 15a and 15b. This calculation determined a percentage of distance between the two markers for that segment, where the right side was set as zero percent and the left was set as 100%.

$$distcop = \sqrt{(Rmarker_y - COP_y)^2 + (Rmarker_z - corr)^2} \quad (15a)$$

$$COPline = \left(\frac{dist_n}{distcop} \right) \cdot 100 \quad (15b)$$

In Equations 15a and 15b, $Rmarker_y$ is the y coordinate of the right side marker for that segment, $Rmarker_z$ is the z coordinate of the right side marker for that segment, COP_y is the y coordinate of the center of pressure, and $corr$ is the correction value determined in Equation 14. The normal value for the center of pressure would be 50 percent which is halfway between the two markers.

2.7 Dynamic Calculations:

The calculations used to describe the dynamic motion of both the normal and pathological spines were based on the work by Davis [22] and Kadabba [15]. The focus of the dynamics work was on the movement of the pelvis, thorax, and spine. Therefore, three anatomical planes were created at the pelvis, mid-thorax, and upper thorax and these planes allowed for the creation of three motion segments: the entire spine, lower spine, and upper spine.

The anatomic planes must be constructed using a minimum of three markers. The pelvic plane was constructed using the markers on the left and right ASIS and the marker on the S1 vertebra. The mid-thorax plane was created using the markers placed on the base of the rib cage, and the marker on the T7 vertebra. Finally, the upper thoracic plane was constructed by the markers placed on the right and left clavicle and the C7 vertebra. Each of the anatomic planes has its own axis to determine the movement of the plane within the global coordinate system. In this work the pelvic plane is considered the primary plane; since it is assumed that the motion of the spine will originate at the pelvis.

The pelvic axes were based on the position of the pelvis within the global coordinate system of the motion laboratory. The primary axis, the y-axis, of the pelvis is depicted in Figure 18.

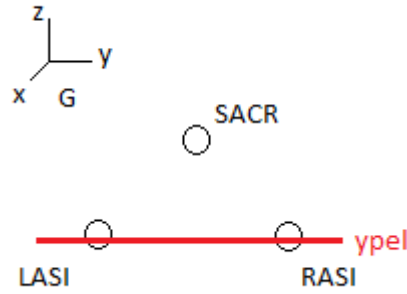


Figure 18: The primary pelvic axis (in red)

The primary axis, the unit vector created by the two ASIS markers is given by Equation 16.

$$ypel = \left(\frac{LASI - RASI}{norm(LASI - RASI)} \right) \quad (16)$$

In Equation 16, $ypel$ is the y-axis of the pelvis, $LASI$ and $RASI$ are the global coordinates of the markers placed on the left and right ASIS respectively. The Euclidean norm of the vector is taken to determine the magnitude of the vector required to create the unit vector. A temporary axis was created between the center point of the primary vector and the S1 marker given by Equation 17a and 17b.

$$fpel = \left(\frac{LASI + RASI}{2} \right) \quad (17a)$$

$$Tvector = \left(\frac{fpel - SACR}{norm(fpel - SACR)} \right) \quad (17b)$$

In Equations 17a and 17b, $fpel$ is the midpoint of the vector between the two ASIS markers, $LASI$ and $RASI$ are the global positions of the left and right ASIS markers, and $SACR$ is the global position of the S1 vertebra. $Tvector$, shown in Figure 19, is the

temporary vector created between midpoint of the primary vector and the marker on the S1 vertebra.

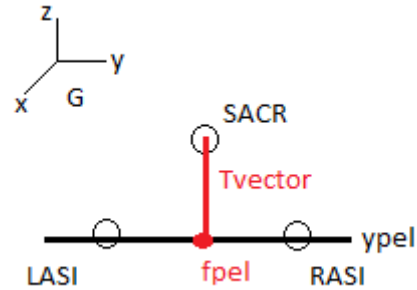


Figure 19: Temporary axis (depicted in red)

The z-vector of the pelvis was created by crossing the temporary axis, *Tvector*, with the y-axis of the pelvis given by Equation 18.

$$zpel = Tvector \times ypel \quad (18)$$

In Equation 18, *zpel* is the z-axis of the pelvis, depicted in Figure 20, and *Tvector* and *ypel* are as described above.

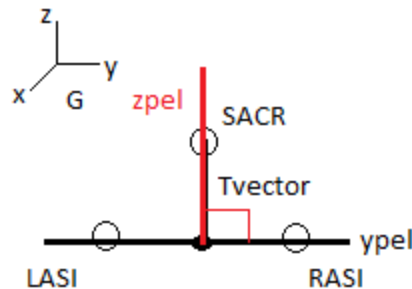


Figure 20: The z-axis, depicted in red, is perpendicular to the y-axis of the pelvis.

The x-axis of the pelvis was constructed by crossing the z-axis of the pelvis with the y-axis of the pelvis given by Equation 19.

$$xpel = zpel \times ypel \quad (19)$$

In Equation 19, x_{pel} is the unit vector describing the x-axis of the pelvis and illustrated in Figure 21, and z_{pel} and y_{pel} are the unit vectors describing the z and y axes of the pelvis, respectively.

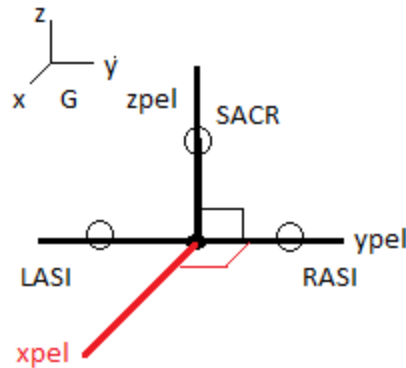


Figure 21: The x-axis of the pelvis (depicted in red)

The axes for the upper and mid thorax planes were constructed in a similar manner to the pelvic coordinate system. The major difference between these two planes was the primary axis. Since the majority of the motion through the two thoracic planes was transferred along the vertical axis, the primary axis for these planes was the z-axis. Due to the similarity in methods to construct the axes for the thoracic planes and those of the pelvic plane, only the differences will be discussed.

The primary axis for the mid-thoracic plane was based on the vertical axis connecting the mid-thoracic plane to the pelvis. Since the motion of the mid-thorax was transferred up from the pelvis along the spine. Therefore, the primary axis, the z-axis, of the mid-thorax was constructed by the vector connecting the center point of the pelvic plane and the center point of the mid-thoracic plane and is given by Equations 20a, 20b, and 20c.

$$cpel = fpel + \left(\frac{SACR}{2}\right) \quad (20a)$$

$$cmthor = \left(\frac{LRIB+RRIB}{2}\right) + \left(\frac{T7}{2}\right) \quad (20b)$$

$$zmthor = \left(\frac{cmthor-cpel}{norm(cmthor-cpel)}\right) \quad (20c)$$

In Equations 20a, 20b, and 20c, $fpel$ is the center point of the primary axis of the pelvis, $LRIB$ and $RRIB$ are the global coordinates of the left and right rib markers, $T7$ is the global coordinates of the T7 marker, $cpel$ is the center point of the pelvic plane in global coordinates, $cmthor$ is the center point of the mid-thoracic plane in global coordinates, and $zmthor$ is the unit vector describing the z-axis of the mid-thoracic plane, and shown in Figure 22.

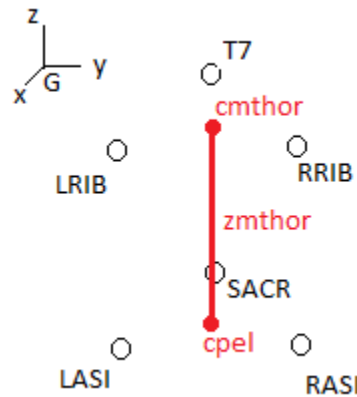


Figure 22: The primary axis for the mid-thorax, the z-axis (depicted in red)

The x-axis was created by crossing the z-axis and the temporary axis, the vector between the right and left rib markers. The y-axis was created by crossing the x and z axes of the mid-thoracic plane.

The upper thoracic plane was constructed in an identical fashion to the mid-thoracic plane. The main difference between this plane and the mid-thoracic plane was that the primary axis, the z-axis, was based on the vector connecting the center point of the mid-thoracic plane and the center point of the upper thoracic plane. Again, this was done so the primary axis is along the same axis as the transferred motion from the lower planes given by Equations 21a, and 21b and depicted in Figure 23.

$$cth\text{or} = \left(\frac{LCLA + RCLA}{2} \right) + \left(\frac{C7}{2} \right) \quad (21a)$$

$$z\text{thor} = \left(\frac{cth\text{or} - cm\text{thor}}{\text{norm}(cth\text{or} - cm\text{thor})} \right) \quad (21b)$$

In Equation 21a and 21b, *LCLA* and *RCLA* are the global coordinates of the left and right clavicle markers, *C7* is the global coordinate of the C7 marker, *cmthor* is the center point of the mid-thoracic plane in global coordinates, *cthor* is the center point of the upper thoracic plane in global coordinates, and *zthor* is the unit vector describing the z-axis of the upper thoracic plane.

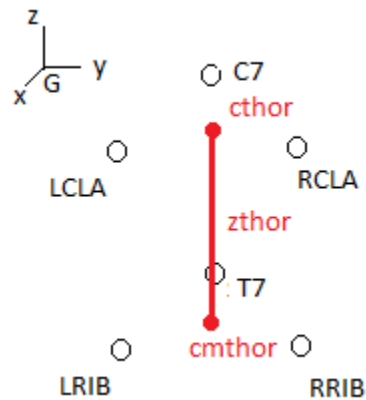


Figure 23: The primary axis for the upper thorax, the z-axis (depicted in red)

The x-axis is created by crossing the z-axis and the temporary axis, the vector between the right and left clavicle markers. The y-axis was created by crossing the x and z axes of the upper thoracic plane.

Using these axes, it was possible to calculate Euler angles that describe the three dimensional motion of each anatomic plane. The rotation sequence used in this work was an YXZ rotation to describe the motion in the sagittal plane, followed by the coronal plane, and then the transverse plane. This sequence was chosen because it was a common rotational sequence used in gait analysis and was described by Kaddaba [15]. The rotation matrix given by Equation 22 can be used to calculate the Euler angles for each plane.

$$\begin{bmatrix} IL \\ JL \\ KL \end{bmatrix} = \begin{bmatrix} C1C3 + S1S2S3 & C2S3 & -S1C3 + C1S2S3 \\ -C1S3 + S1S2C3 & C2C3 & S1S3 + C1S2C3 \\ S1C2 & -S2 & C1C2 \end{bmatrix} \begin{bmatrix} Ig \\ Jg \\ Kg \end{bmatrix} \quad (22)$$

In Equation 22, Ig , Jg , and Kg are the unit vectors of the global coordinates of the lab in the x, y, and z direction respectively, IL , JL , and KL are the unit vectors of the local coordinates of each of the motion planes in the x, y, and z directions respectively, C is cosine, S is sine, and the subscripts of the cosine and sine term relate to the rotational sequence where 1 is the sagittal plane angle, 2 is the coronal plane angle, and 3 is the transverse plane angle. Solving for the unknown angles in Equation 22 provides the equations to describe the motion of the planes' in three dimensional space, give in Equations 23a, 23b and 23c.

$$\theta_2 = \sin^{-1}(-KL * JG) \quad (23a)$$

$$\theta_1 = \sin^{-1} \left(\frac{KL * Ig}{\cos \theta_2} \right) \quad (23b)$$

$$\theta_3 = \sin^{-1} \left(\frac{IL * Jg}{\cos \theta_2} \right) \quad (23c)$$

In Equations 23a, 23b, and 23c, θ_2 is the coronal plane angle, KL is the z axis of the local coordinate system, Jg is the global y axis, θ_1 is the sagittal plane angle, Ig is the global x axis, θ_3 is the transverse plane angle, and IL is the x axis of the local coordinate system. The arcsine function can produce two possible answers for each angle; therefore, the atan2 function was used when programming to ensure that the angles were calculated in the correct quadrant.

2.8 Filtering:

The raw data imported into Matlab needed to be filtered to remove system noise and movement artifact prior to performing the necessary calculations. There are multiple means to filter motion data, but in this work, two methods were attempted to identify the best filtering method. Initially a moving average filter was tested but the results attenuated the motion data and it was determined to be ineffective. A low-pass Butterworth filter was implemented next because human movement normally occurs at low frequency and, in this work; the motions of the subjects were extremely slow. The cutoff frequency and order of the filter were selected based on sequential testing where raw motion data for the L3 marker was read into Matlab and filtered with a low-pass fifth order Butterworth filter while varying the cutoff frequencies from 5 to 25Hz as shown in Figure 24.

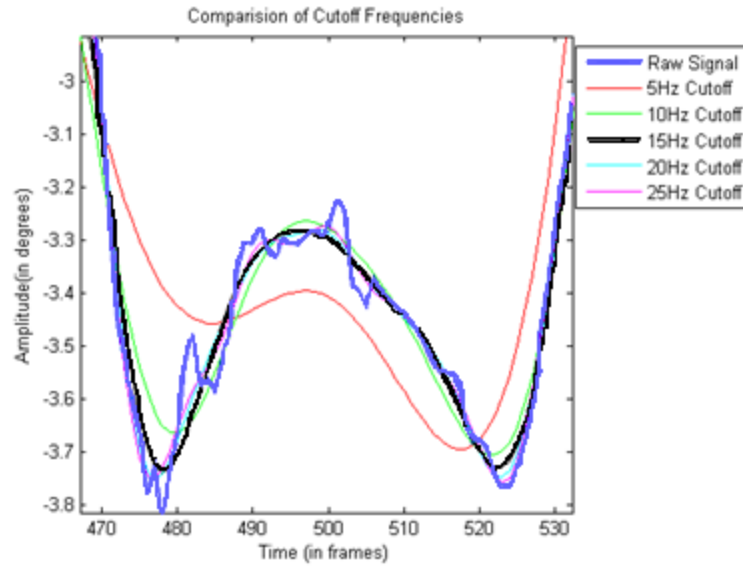


Figure 24: Comparison cutoff frequencies used in the Butterworth filter on a sample of motion data

The results of the this testing showed that cutoff frequencies of 5Hz and 10Hz over-attenuated the raw data, frequencies of 20Hz and 25Hz under-attenuated the data, and the cutoff frequency of 15Hz was sufficient.

The order of the Butterworth filter was chosen in a manner similar to the cutoff frequency, where the same motion data from the L3 marker was read into Matlab again and the filtered with a low-pass Butterworth filter with a cutoff frequency of 15Hz, while the order of the filter varied from 1st order to 5th order as shown in Figure 25.

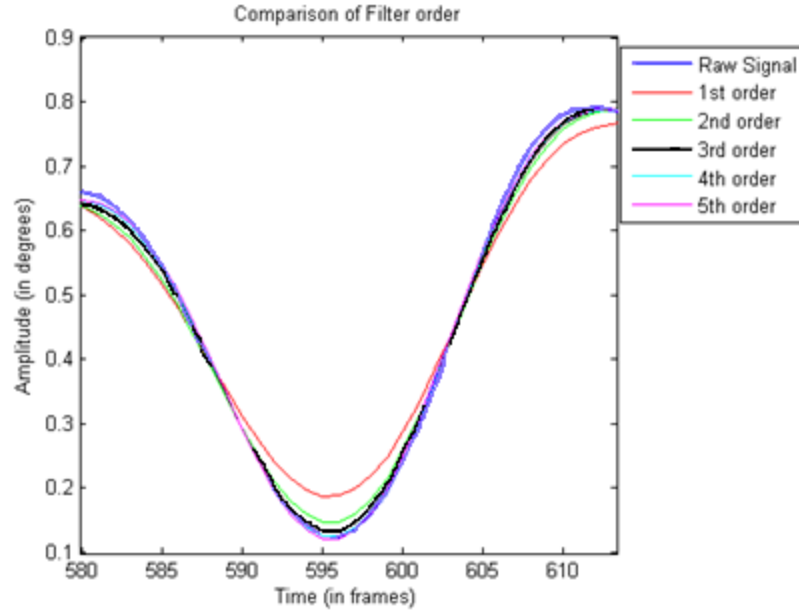


Figure 25: Comparison orders used in the Butterworth filter on a sample of motion data

In a similar manner to the cutoff frequencies the results of the order testing showed that 1st, 2nd, and 3rd order filters over-attenuated the data, but showed minor differences between the 4th and 5th order filters. Upon further examination, the 4th order filter provided slightly better results and was selected

2.9 Validation:

The marker set and Matlab model used in this work were new and required validation to prove the accuracy of the results obtained by this model. Two methods were employed in order to validate the static portion of the model. The first method involved testing a series of known angles using a manual goniometer and three markers as seen in Figure 26. The goniometer was set to 5°, 10°, 15°, 20°, 45°, and 60°, and placed in the measurement volume for a static data collection interval of five seconds each. The calculated angles were then compared to the known angle.

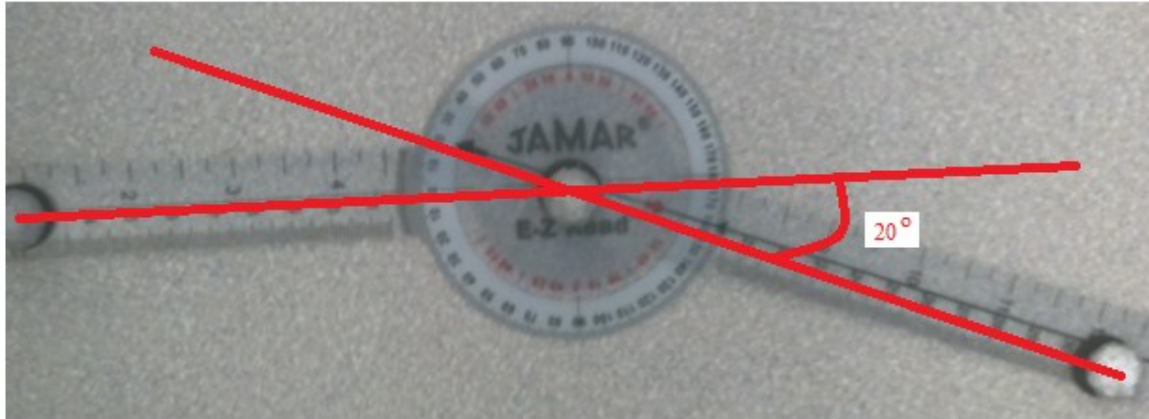


Figure 26: Validation test using a manual goniometer

The second portion of the static model validation was to compare the results obtained from the proximal and distal angles, as well as the SVA and CVA, to the radiographic measures obtained from each subject. Model accuracy was validated by determining the difference between the radiographic and the calculated measures. To be considered accurate the model had to be within 5° of the radiographic measurements, especially since radiographic measures commonly have $\pm 5^{\circ}$ intra-observer reliability. The precision of the model was determined using the standard deviations for the three static trials for the same four variables. The model was considered precise if the standard deviation in the calculated measures was $\pm 5^{\circ}$ or less.

The dynamic portion of the model also required validation. Since the dynamic model used in this work was based on the model by Kadaba et al. [15], the results of two subjects, one normal and one AIS subject, were compared with the VICON Workstation software. The VICON software is a proven and validated model that is based on the work of Kadaba et al., and the model was considered valid if the results of the Matlab output were within 2° of the VICON output.

3.0 Results:

3.1 Subject Demographics

A total of 10 subjects were used in this study: five subjects in the control group and five patients in the study group as seen in Table 4, the average age of the both the control and scoliosis group was 15 ± 1 years and, the control group had a major Cobb angle of $2 \pm 2^\circ$ while the scoliosis group had a major Cobb angle of $41 \pm 13^\circ$. The curves of the scoliosis group were described using a modified Lenke classification, where there were three Lenke 1 curves, one Lenke 2 curve, and one Lenke 3 curve.

Table 4: Subject demographics

	Control Group	Scoliosis Group
Age (years)	15 ± 1 (14-15)	15 ± 1 (13-16)
Gender: Male	3	4
Female	2	1
Major Cobb Angle (°)	2 ± 2 (0 -4)	41 ± 13 (25-57)

3.2 Validation

The results of the validation work done for the static portion of this model were based on the calculated output and the direct measures from the radiographs and The statistical analysis for the validation was performed in Microsoft Excel (Microsoft, Redmond , WA). The comparison of the model to the known angles was determined by taking the difference between the Matlab value and the known values as seen in Table 5.

Table 5: Comparison of the known angle versus the angles calculated in Matlab.

Known Angle (°)	Calculated Angle (°)	Difference (°)
5	2	3
10	14	-4
15	17	-2
20	23	-2
45	46	-1
60	58	2

The results show that there was an average difference of 2.3° between the known angles and the angles calculated using the Matlab code, which was better than the $\pm 5^\circ$ accuracy rate accepted in radiographic measures.

The accuracy of the model was determined by comparing the radiographic measures to those calculated using Matlab. Once again, these measures were determined by taking the difference between the radiographic and calculated values as shown in Table 6.

Table 6: Comparison of the proximal angle (C7-T7-L3) determined from the radiograph and calculated in Matlab

Subject	Radiographic Angle (°)	Calculated Angle (°)	Difference (°)
1	14	15	-1
2	4	10	-6
3	2	8	-6
4	2	1	1
5	8	12	-4
6	1	5	-4
7	11	17	-6
8	10	12	-2
9	3	3	0
10	4	5	-1
Average	9 ± 5	6 ± 5	-3 ± 2

The result of the accuracy testing showed that the static calculations were within the common error of radiographic measures. It is also important to note that the model tends to underestimate the Cobb angle measure as shown by the average difference of $-3 \pm 2^\circ$.

The precision of the model was determined by calculating the standard deviation between the three static trials for the SVA, CVA, distal angle, and proximal angle using Excel as shown in Table 7.

Table 7: Standard deviations of the three static trials for each subject

Subject	SVA (mm)	CVA (mm)	Proximal Angle (°)	Distal Angle (°)
1	4	1	1	1
2	2	1	0	0
3	6	3	1	1
4	2	4	1	1
5	8	3	1	0
6	2	1	1	6
7	5	0	0	0
8	3	3	0	0
9	1	3	1	1
10	3	3	1	1
Average	4	2	1	1

The results of the precision testing showed that the repeatability of the Matlab model on average was $\pm 1^\circ$ with a precision of $\pm 4\text{mm}$ in terms of distance measures.

When comparing the dynamic results output from the Matlab model, which was used in this work, and the Vicon model, which was based on a given validated model, the results showed that the difference between the two models was less than 1° (i.e., 0 to 0.8°). This level of accuracy held true for both scoliosis and normal subjects in each of the three planes of motion. Since the VICON model does not calculate spine angles or the

mid-thorax angles, only the upper thorax and pelvis angles were compared as shown in Figure 27a-f and Figure 28a-f.

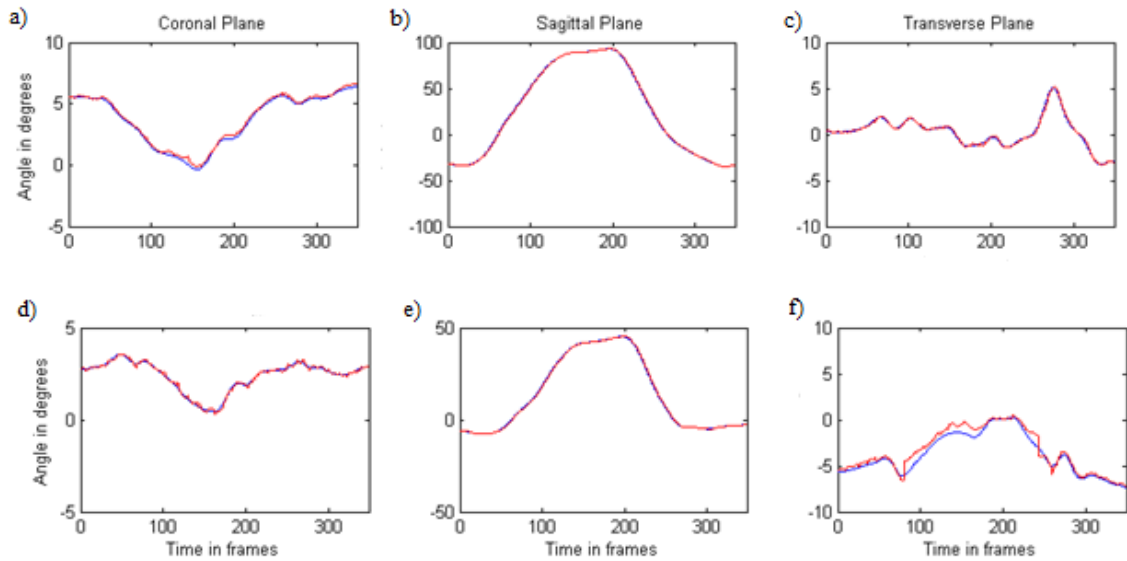


Figure 27: Comparison of Vicon output to the Matlab model for a subject with scoliosis (Red solid line is Vicon data, blue solid line is Matlab data)

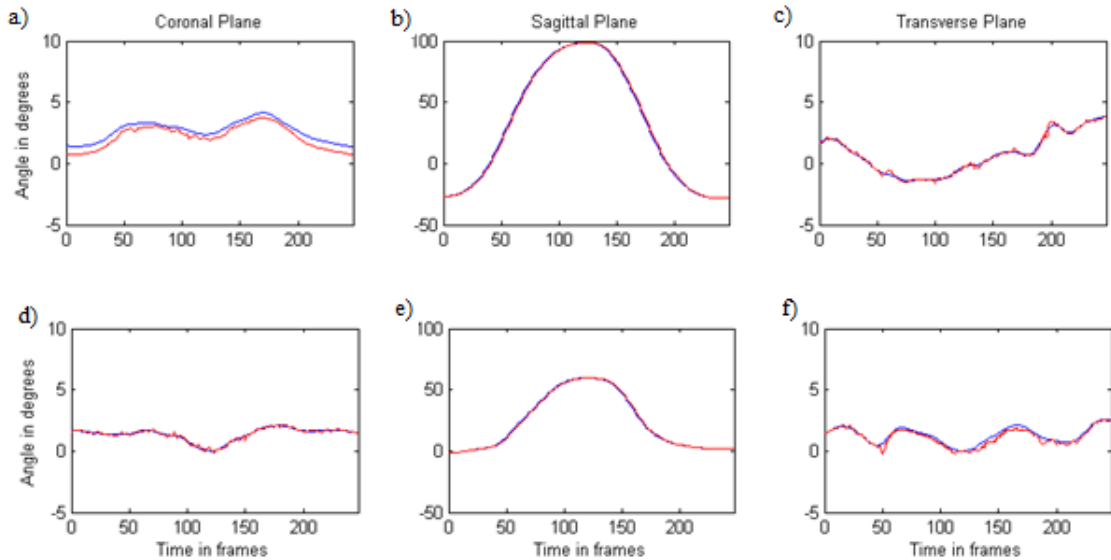


Figure 28: Comparison of Vicon output to the Matlab model for a control subject (Red solid line is Vicon data, blue solid line is the Matlab data)

3.3 Static Results

The postural alignment calculations were calculated using Matlab and the center of pressure data was derived using the force plates during the relaxed standing static trials. A single factor ANOVA, with an alpha of 0.05, was calculated in Excel to determine if there was a statistically significant difference between the scoliosis group and control group. The average postural alignment for the study group was based on a total of 15 postural trials (i.e., three trials per patient) and was calculated at the shoulders, base of the rib cage, and at the pelvis as seen in Table 8.

Table 8: Postural alignment for the scoliosis and control group

	Scoliosis Group	Control Group	P
Shoulders (%)	40 ± 8	44 ± 4	0.35
Rib Cage (%)	47 ± 3	49 ± 2	0.15
Pelvis (%)	51 ± 6	50 ± 5	0.92

Postural alignment values below 50% indicate that the center of pressure vector passed through the subject's body closer to the right side markers, while values greater than 50% indicate that the center of pressure vector passed closer to the left side markers. A value of 50% indicated the center of pressure was aligned directly between the two markers. Based on the postural alignment calculation, there were minor differences between the two study groups; however, the center of pressure for the scoliosis group was shifted slightly more to the right than the control group.

Asymmetry values were also calculated for 15 postural trials for the two study groups. The asymmetry values measure to what extent the shoulders, base of the rib cage, and pelvis were skewed in the transverse and coronal plane. In the transverse plane, the

values calculated were based on how much the right marker was in front (internally rotated) or behind (externally rotated) the left marker and a positive value indicated that the right marker was in front of the left. In the coronal plane, the values calculated described how much the vector was tilted upward or downward, where a positive value indicated that the marker on the right was tilted up compared to the left marker. A single factor ANOVA, with an alpha value of 0.05, was calculated using Excel to determine if there was a statistically significant difference between the scoliosis group and the control group as seen in Table 9.

Table 9: Comparison of frontal and transverse plane asymmetry at the shoulders, rib cage, and pelvis

		Scoliosis Group	Control Group	P
Frontal Plane Asymmetry (°)	Shoulders	-1 ± 4	0 ± 4	0.70
	Rib Cage	1 ± 4	1 ± 5	0.99
	Pelvis	1 ± 1	0 ± 2	0.26
Transverse Plane Asymmetry (°)	Shoulders	1 ± 2	3 ± 2	0.32
	Rib Cage	-2 ± 4	2 ± 2	0.11
	Pelvis	-4 ± 3	1 ± 3	0.03

The frontal plane asymmetry shows minor differences between the scoliosis and control groups. Similar results were also seen when looking at the rotational profiles of the subjects, with the exception of the pelvis. Subjects diagnosed with scoliosis tended to have a more externally rotated pelvis indicating that the left side of the pelvis was rotated outward compared to the right side. This difference was found to be statistically significant (p=0.03).

During the static collection the values were calculated based on the marker placement, in order to replicate those measures commonly derived from spinal

radiographs. The proximal and distal angles (i.e., the angles created using C7, T7 and L3 markers for the proximal angle and T7, L3, and S1 markers for the distal angle) were measured off the radiographs using the same landmarks used to place the retro-reflective markers. The radiographic data and calculated data were then correlated using a Pearson's correlation coefficient. This method was also used for the kyphosis and lordosis angles, and for the SVA and CVA measurements as shown in Table 10, where a value of 0.80 was considered to be highly correlated.

Table 10: Correlation between radiographic measures and calculated values

	Radiographic Values	Calculated Values	R
SVA (mm)	28 ± 26	31 ± 27	0.92
CVA (mm)	6 ± 18	7 ± 18	0.86
Proximal Angle (°)	6 ± 5	9 ± 5	0.85
Distal Angle (°)	10 ± 8	12 ± 10	0.93
Kyphosis (°)	30 ± 11	3 ± 8	0.30
Lordosis (°)	42 ± 17	4 ± 12	0.12

The results showed that the coronal plane profiles, the proximal angle, distal angle, and coronal vertical alignment are highly correlated to the radiographic measures. The sagittal plane profile did not correlate as well, while the sagittal vertical alignment was observed to be highly correlated ($r=0.92$). The kyphosis and lordosis angles do not correlate with the radiographic measures indicating that the Matlab model was not able to accurately capture these measures.

An additional set of data was also collected to determine how much the X-ray frame affected the data, if the frame helped to standardize the radiographic data, and if the data collected during the motion capture was accurate. The effectiveness of the frame

was determined by calculating the standard deviations between the static measures for both the frame data and the relaxed standing data in Excel as shown in Table 11.

Table 11: Comparison of data collected with and without positioning frame

	Radiographic Frame	Relaxed Standing	Radiographic Values
SVA (mm)	31 ± 27	26 ± 19	28 ± 26
CVA (mm)	7 ± 18	4 ± 19	6 ± 18
Proximal Angle (°)	9 ± 5	10 ± 7	6 ± 5
Distal Angle (°)	12 ± 10	15 ± 14	10 ± 8
Kyphosis (°)	3 ± 8	1 ± 7	30 ± 11
Lordosis (°)	4 ± 12	1 ± 12	42 ± 17

The measurements made during the relaxed standing trials including the SVA, CVA, kyphosis, lordosis, proximal angle, and distal angle, were also correlated to the radiographic data as seen in Table 12, where a correlation coefficient of 0.80 was considered highly correlated.

Table 12: Correlation between radiographic measures and values collected without the X-ray frame.

	Radiographic Values	Relaxed Standing	r
SVA (mm)	28 ± 26	26 ± 19	0.53
CVA (mm)	6 ± 18	4 ± 19	0.77
Proximal Angle (°)	6 ± 5	10 ± 7	0.83
Distal Angle (°)	10 ± 8	15 ± 14	0.65
Kyphosis (°)	30 ± 11	1 ± 7	0.05
Lordosis (°)	42 ± 17	1 ± 12	0.02

The method that had the highest correlation coefficients was considered to be the most effective method at reproducing the radiographic data. Tables 11 and 12, show that the relaxed standing profiles did not correlate with radiographic measures as well as the frame trials.

3.4 Dynamic Results

The dynamic results were based on the three bending motions that were performed by the study subjects. For each bending motion the rotation angles were calculated in each of the three motion planes (i.e., coronal, sagittal, and transverse), for the anatomic planes (i.e., upper thorax, mid-thorax, and pelvis), and for the three segments (i.e., the entire spine, upper spine (C7 to T7), and the lower spine (T7 to S1)). These angles were calculated and plotted using Matlab for each subject. The average for each of the subjects was calculated and plotted in Matlab to demonstrate the differences in motion between the scoliosis and control group. The time scale was normalized to the full range of motion to ensure that the motions could be accurately compared from one patient to another. The average motion plots were based on two trials for each of the subjects which yielded a total of 10 trials for each of the two study groups. In the case of the flexion and extension bending task, the scoliosis group mean consisted of only four subjects. One subject was removed from the scoliosis group because this subject had a large body mass index (BMI), and markers were obscured during the flexion portion of the bending tasks. The range of motion and maximum endpoints for each of the trials for each plane and segment were calculated using Matlab. The data for the scoliosis and control groups were then compared to determine if a statistically significant difference between the two groups was present using a single factor ANOVA with an alpha of 0.05 in Excel.

The lateral bending task showed similar results between the control and scoliosis groups when looking at the anatomic plane angles of the upper thorax, mid-thorax, and pelvis as seen in Table 13 and in Figures 29a-i. In Figure 29 the first column is the

coronal plane and positive values indicate a lean to the left, the second column is the sagittal plane and positive values indicate flexion, and the third column is the transverse plane and positive values indicate internal rotation.

Table 13: Comparison of average range of motion for each of the anatomic planes in the lateral bending task.

	Upper Thorax Angles (°)			Mid-Thorax Angles (°)			Pelvis Angles (°)		
	Coronal Plane	Sagittal Plane	Transverse Plane	Coronal Plane	Sagittal Plane	Transverse Plane	Coronal Plane	Sagittal Plane	Transverse Plane
Scoliosis Group	66 ± 12	12 ± 6	18 ± 6	41 ± 11	10 ± 7	30 ± 9	13 ± 11	6 ± 2	28 ± 16
Control Group	59 ± 15	10 ± 5	24 ± 12	40 ± 4	8 ± 3	19 ± 20	11 ± 2	3 ± 2	19 ± 21
P	0.44	0.59	0.34	0.9	0.35	0.29	0.57	0.08	0.53

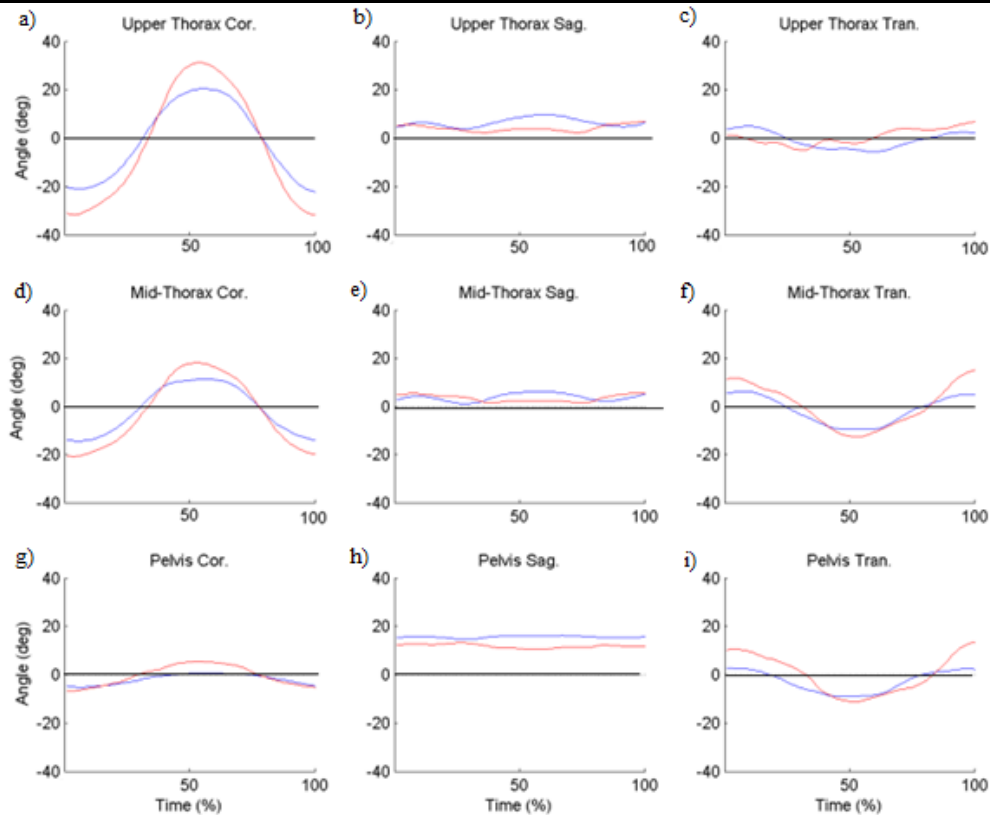


Figure 29: Comparison of the anatomic plane ranges of motion for the lateral bending task (Red line is the scoliosis group, blue line is the control group)

The scoliosis group had a greater range of motion in the primary motion plane (i.e., the coronal plane) in each of the three anatomic planes. The largest difference in the primary motion plane was the total range of motion for the upper thorax ($66 \pm 12^\circ$ scoliosis group, $59 \pm 15^\circ$ control group); however, this difference was not significant. There was an 11° difference in the range of motion at the mid-thorax in the transverse plane between the scoliosis group and the control group. Also of note, was a 3° difference in the pelvic sagittal plane that was approaching significance ($p = 0.08$), and a 9° difference in the transverse plane; however, this difference was not significant ($p = 0.53$).

The dynamic motion for the spinal segments, the upper spine, lower spine, and total spine, produced similar results to the anatomic motion planes during the lateral bending task as seen in Table 14 and Figure 30a-i. In Figure 30 the first column is the coronal plane and positive values indicate a lean to the left, the second column is the sagittal plane and positive values indicate flexion, and the third column is the transverse plane and positive values indicate internal rotation.

Table 14: Comparison of average range of motion for each of the spinal segments during the lateral bending task

	Upper Spine Angles ($^\circ$)			Lower Spine Angles ($^\circ$)			Total Spine Angles ($^\circ$)		
	Coronal Plane	Sagittal Plane	Transverse Plane	Coronal Plane	Sagittal Plane	Transverse Plane	Coronal Plane	Sagittal Plane	Transverse Plane
Scoliosis Group	26 ± 10	7 ± 4	25 ± 12	28 ± 8	7 ± 3	11 ± 5	53 ± 5	11 ± 5	26 ± 19
Control Group	19 ± 12	4 ± 3	21 ± 13	31 ± 5	5 ± 2	13 ± 5	50 ± 16	7 ± 4	33 ± 13
P	0.33	0.34	0.61	0.33	0.23	0.36	0.36	0.20	0.48

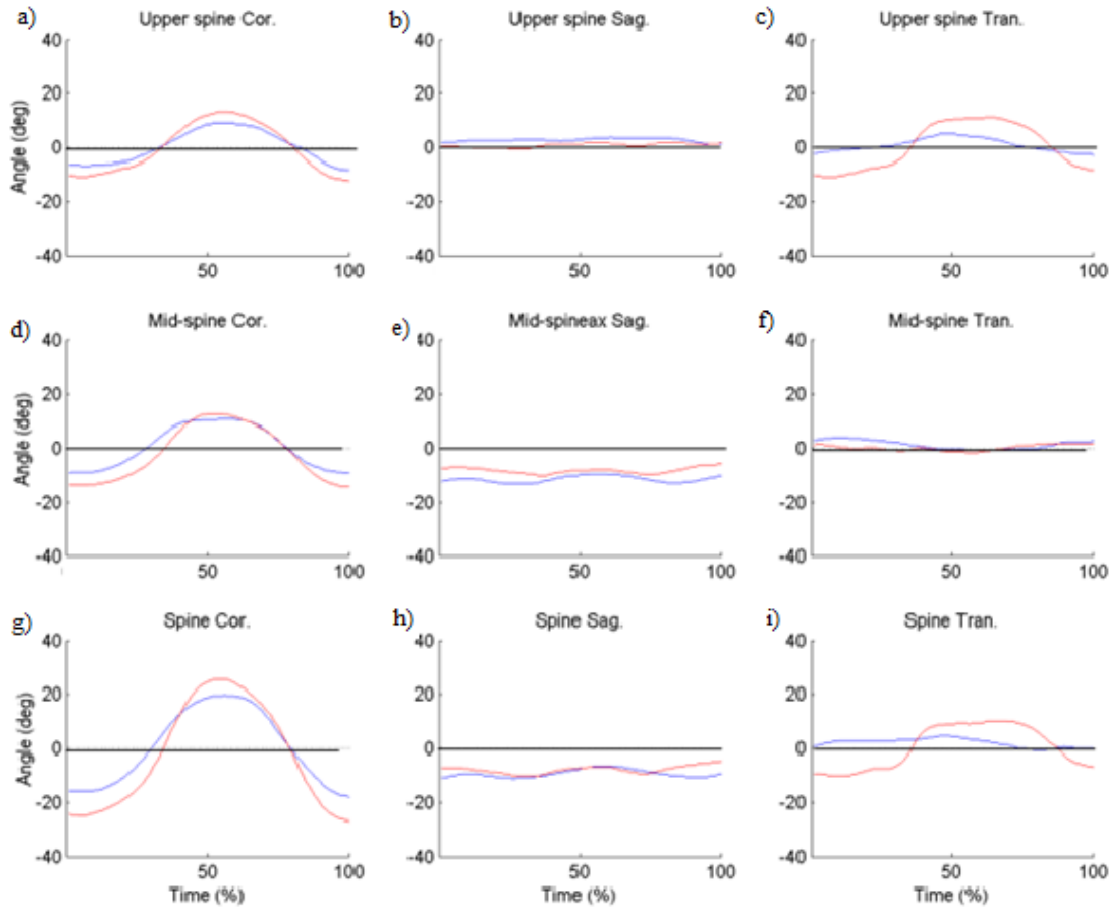


Figure 30: Comparison of the spinal segment ranges of motion for the lateral bending task (Red line is the scoliosis group, blue line is the control group)

The results of the spinal segments are similar to those of the anatomic motion planes for the lateral bending task. The scoliosis group had a greater range of motion in the primary motion plane, with the exception of the lower spine ($28 \pm 8^\circ$ scoliosis group, $31 \pm 5^\circ$ control group, $p=0.33$). The spinal segments showed similar ranges of motion between the two groups in the secondary motion planes (i.e., sagittal and transverse planes).

The rotational task was analyzed in a similar fashion to the lateral bending task. The plane and segment motion between the two study groups were compared as shown in Table 15 and Figure 31a-i. In Figure 31 the first column is the coronal plane and positive

values indicate a lean to the left, the second column is the sagittal plane and positive values indicate flexion, and the third column is the transverse plane and positive values indicate internal rotation.

Table 15: Comparison of average range of motion for each of the anatomic planes in the axial rotation task

	Upper Thorax Angles (°)			Mid-Thorax Angles (°)			Pelvis Angles (°)		
	Coronal Plane	Sagittal Plane	Transverse Plane	Coronal Plane	Sagittal Plane	Transverse Plane	Coronal Plane	Sagittal Plane	Transverse Plane
Scoliosis Group	18 ± 13	14 ± 9	220 ± 24	20 ± 9	12 ± 3	158 ± 26	26 ± 17	8 ± 3	128 ± 20
Control Group	11 ± 12	7 ± 4	211 ± 57	11 ± 6	16 ± 2	139 ± 19	16 ± 10	5 ± 2	116 ± 38
P	0.39	0.18	0.66	0.08	0.19	0.21	0.07	0.06	0.52

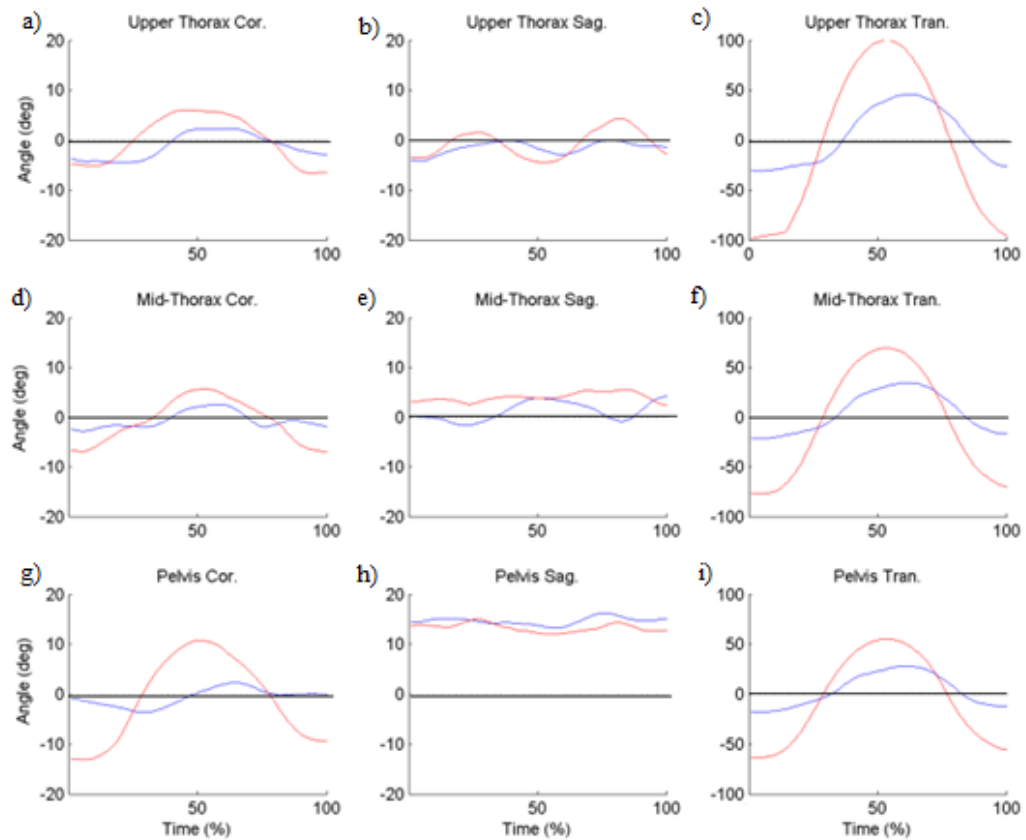


Figure 31: Comparison of the anatomic plane ranges of motion for the axial rotation task (Red line is the scoliosis group, blue line is the control group)

Similar to the results of the lateral bending tasks the total range of motion in the primary motion plane (i.e., the transverse plane) were greater in the scoliosis group. The largest difference in the total range of motion was the transverse plane of the mid-thorax that had a difference of 19° (158 ± 26° scoliosis group, and 139 ± 19°); however, this was not statistically significant (p=0.21). The scoliosis group had more compensatory motion, there was a 3° difference in the sagittal plane motion of the pelvis (p=0.06) and a 9° difference in the coronal plane motion of the mid-thorax (p=0.08). The endpoints of the axial rotation task for the scoliosis group were less symmetric in the anatomic planes than the control group.

The axial rotation profiles for the spinal segments were determined and seen in Table 16 and Figure 32a-i. In Figure 32 the first column is the coronal plane and positive values indicate a lean to the left, the second column is the sagittal plane and positive values indicate flexion, and the third column is the transverse plane and positive values indicate internal rotation.

Table 16: Comparison of average range of motion of the spinal segments in the axial rotation task

	Upper Spine Angles (°)			Lower Spine Angles (°)			Total Spine Angles (°)		
	Coronal Plane	Sagittal Plane	Transverse Plane	Coronal Plane	Sagittal Plane	Transverse Plane	Coronal Plane	Sagittal Plane	Transverse Plane
Scoliosis Group	11 ± 5	13 ± 8	84 ± 52	24 ± 12	12 ± 3	31 ± 12	18 ± 10	16 ± 10	111 ± 14
Control Group	9 ± 4	11 ± 4	76 ± 35	23 ± 5	10 ± 4	41 ± 13	12 ± 10	6 ± 4	107 ± 38
P	0.59	0.55	0.57	0.94	0.56	0.24	0.45	0.07	0.87

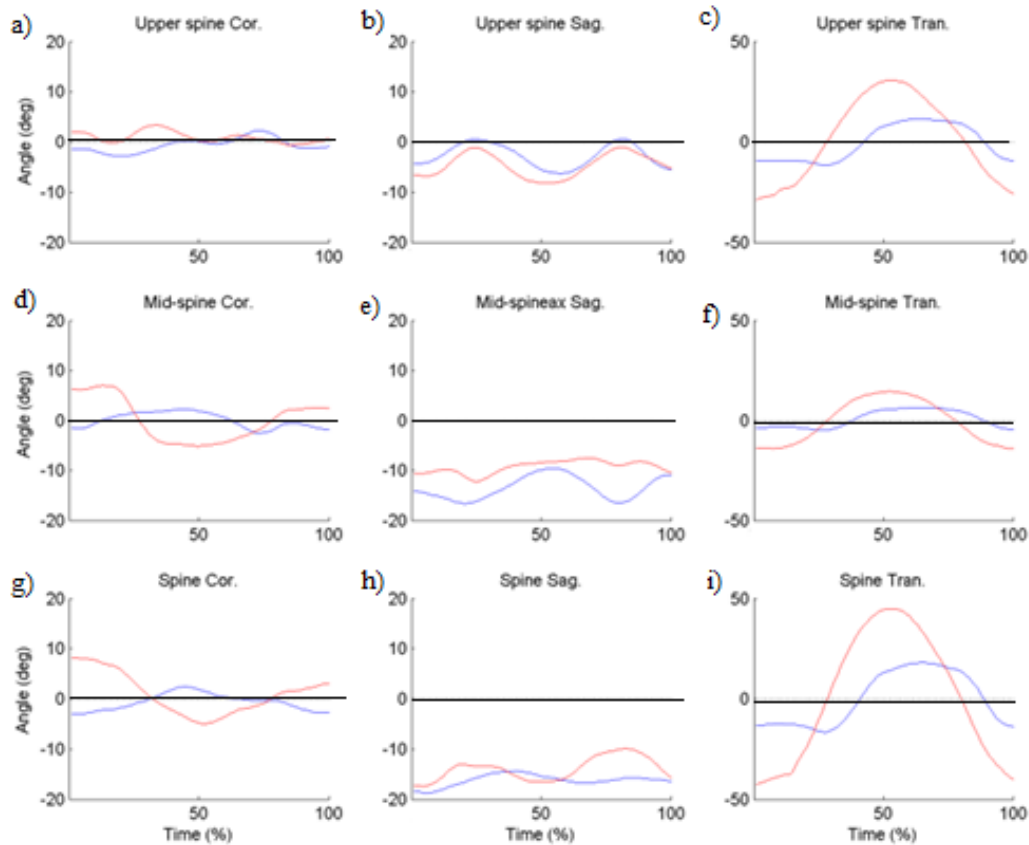


Figure 32: Comparison of the spinal segment ranges of motion for the axial rotation task. (Red line is the scoliosis group, blue line is the control group)

The spinal segment motion profiles showed similar motion profiles between the two study groups. There was greater total spine rotation in the scoliosis group; however, this difference was not statistically significant ($p=0.87$). Compensatory motion was noted in the coronal plane total spine motion profiles for the scoliosis group as the subjects rotated to the left they also laterally leaned to the right; in comparison the control group leaned into the rotation. The scoliosis group also had a 10° greater range of motion in the sagittal plane of the pelvis ($16 \pm 10^\circ$ scoliosis group and $6 \pm 4^\circ$ control group), this difference was approaching a level of statistical significance. The end points of the motion in the transverse plane showed differences between the two study groups, especially in the

lower spine. The motion endpoints of the lower spine indicate the control group rotated to a greater extent both internally ($-15 \pm 7^\circ$ scoliosis group, $-22 \pm 13^\circ$ normal group, $p=0.33$) and externally ($31 \pm 12^\circ$ scoliosis group, $42 \pm 14^\circ$ normal group, $p=0.33$); although these differences were not statistically significant, it does indicate there was a reduction in lower spine motion in the scoliosis group. Also of note was the spine rotation in the normal group was more symmetric ($53 \pm 15^\circ$ internal and $-53 \pm 23^\circ$ external) than the scoliosis group ($66 \pm 14^\circ$ internal and $-45 \pm 6^\circ$ external). This also indicated that the spinal deformity seen in the scoliosis group affects the axial rotation of the spine.

The flexion and extension bending task was analyzed in the same manner as the previous bending tasks. The motion profiles and range of motions for the anatomic motions planes were analyzed to determine differences between the two study groups as seen in Table 17 and Figure 33a-i. In Figure 33 the first column is the coronal plane and positive values indicate a lean to the left, the second column is the sagittal plane and positive values indicate flexion, and the third column is the transverse plane and positive values indicate internal rotation.

Table 17: Comparison of average range of motion for each of the anatomic planes in the flexion/extension task.

	Upper Thorax Angles (°)			Mid-Thorax Angles (°)			Pelvis Angles (°)		
	Coronal Plane	Sagittal Plane	Transverse Plane	Coronal Plane	Sagittal Plane	Transverse Plane	Coronal Plane	Sagittal Plane	Transverse Plane
Scoliosis Group	5 ± 1	123 ± 18	7 ± 1	8 ± 8	107 ± 18	7 ± 1	4 ± 2	60 ± 14	9 ± 1
Control Group	5 ± 3	138 ± 10	7 ± 3	8 ± 5	107 ± 6	5 ± 1	5 ± 4	60 ± 7	5 ± 2
P	0.97	0.17	0.79	0.96	0.91	0.09	0.81	0.98	0.01

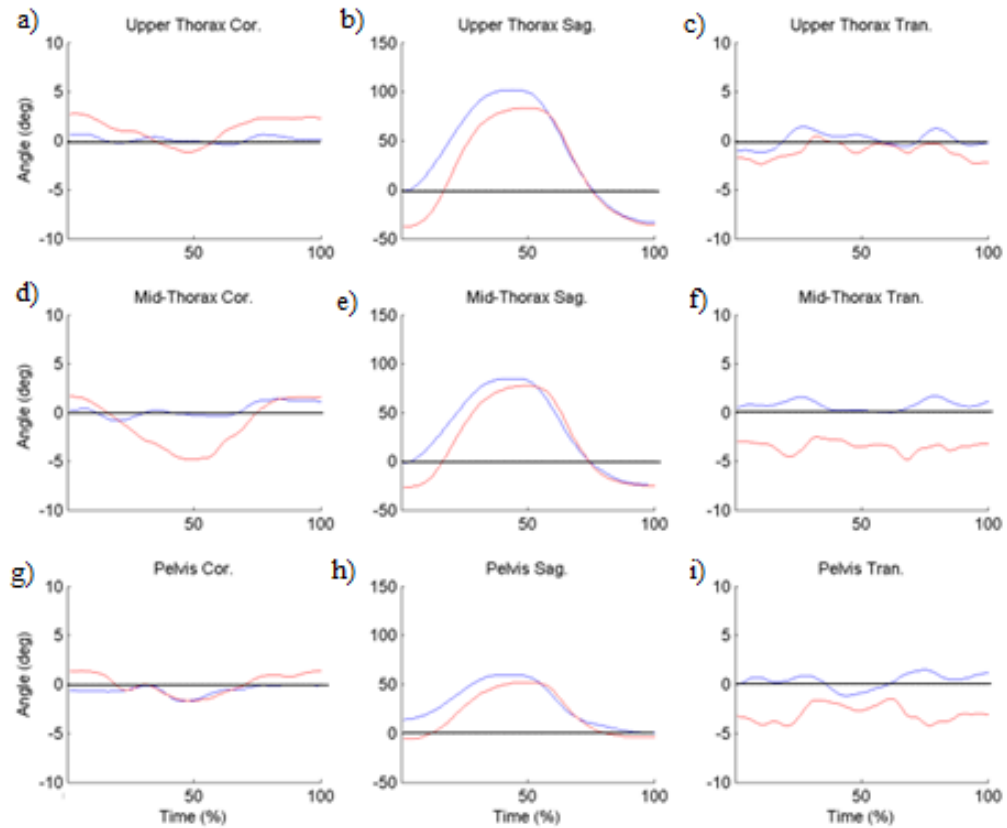


Figure 33: Comparison of the anatomic plane ranges of motion for the flexion bending task (Red line is the scoliosis group, blue line is the control group)

There were minor differences between the motion profiles of the flexion and extension task for two study groups. Unlike the previous bending tasks, this task showed that the range of motion in the primary plane between the two groups was very similar, with the exception of the upper thorax of the control group which had a 15° greater range of motion than the scoliosis group. The motion in transverse plane for the mid-thoracic and pelvic planes of the scoliosis group had a statistically significant greater range of motion $7 \pm 1^\circ$ scoliosis group and $5 \pm 1^\circ$ ($p=0.09$) normal group, $9 \pm 1^\circ$ scoliosis group and $5 \pm 2^\circ$ ($p=0.01$) respectively.

The spinal segment's motion profile and range of motion were analyzed as seen in Table 18 and Figure 34a-i. In Figure 34 the first column is the coronal plane and positive values indicate a lean to the left, the second column is the sagittal plane and positive values indicate flexion, and the third column is the transverse plane and positive values indicate internal rotation.

Table 18: Comparison of average range of motion for each of the spinal segments in the flexion/extension task.

	Upper Spine Angles (°)			Lower Spine Angles (°)			Total Spine Angles (°)		
	Coronal Plane	Sagittal Plane	Transverse Plane	Coronal Plane	Sagittal Plane	Transverse Plane	Coronal Plane	Sagittal Plane	Transverse Plane
Scoliosis Group	4 ± 3	28 ± 11	5 ± 1	6 ± 1	47 ± 6	5 ± 3	5 ± 1	67 ± 19	8 ± 2
Control Group	4 ± 3	38 ± 23	5 ± 3	5 ± 3	55 ± 5	5 ± 1	3 ± 1	80 ± 6	8 ± 1
P	0.96	0.48	0.95	0.40	0.08	0.58	0.01	0.20	0.71

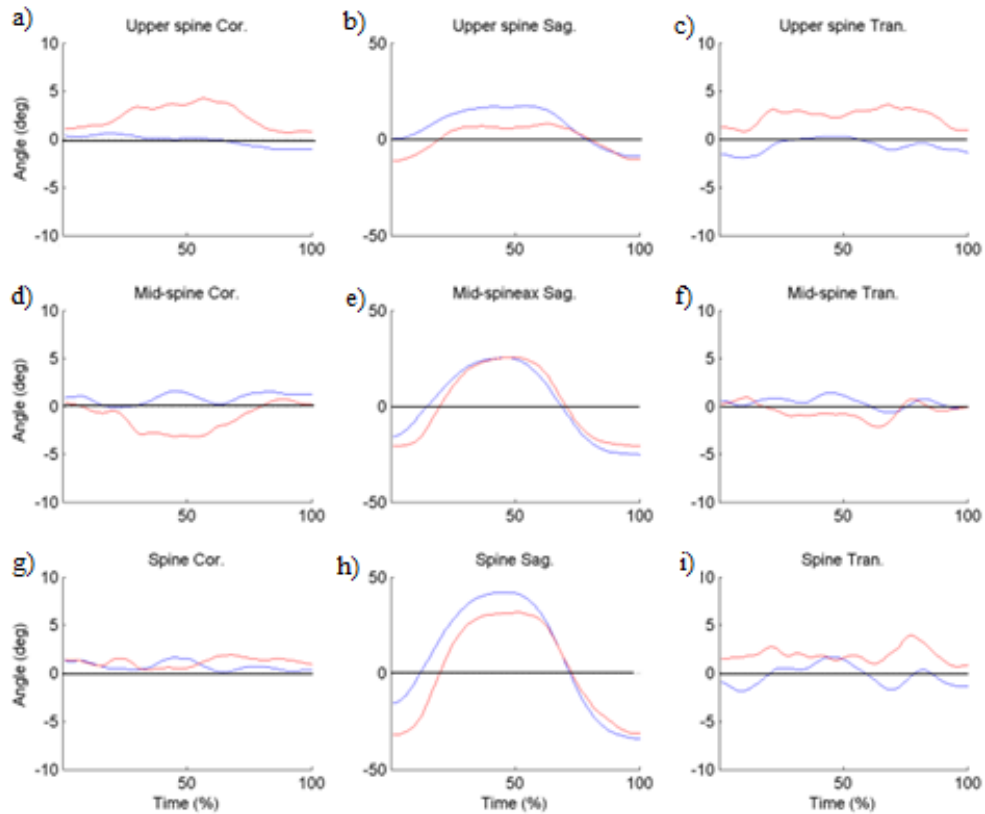


Figure 34: Comparison of the spinal segment ranges of motion for the flexion task (Red line is the scoliosis group, blue line is the control group)

The motion profiles of the spinal segments showed that the scoliosis group tended to lean laterally to the left and internally rotate to a greater extent, especially in the lower spine. The control group had a greater total range of motion in both the upper and lower spine in the primary motion plane; however, this difference is not statistically significant ($p=0.08$). The motion profiles for the spine showed similar extension between the two groups but flexion, especially in the upper spine, was reduced in the scoliosis group ($13 \pm 6^\circ$ scoliosis group, 22 ± 12 normal group, $p=0.22$). Although this was not a statistically significant difference, it does indicate the spinal deformity reduced spinal motion in the sagittal plane.

4.0 Discussion

4.1 Model Considerations

The dynamic model used in this work is based on the works by Davis [22] and Kadaba [15] and relies heavily on Euler rotations. Euler rotation sequences are a means of determining the orientation and position of a rigid body within a defined space. There are multiple Euler rotation sequences that describe the order of rotation about each of the axes of the rigid body. If an xyz rotation is performed, the segment is first rotated about the x-axis, then about the newly positioned y-axis, and finally about the newly positioned z-axis. Each of the Euler sequences correctly describes the movement of the segment but each produces a slightly different result; therefore, it is essential to choose the correct sequence to best describe the movement of the anatomic segments. In this work the sequence used is an yxz rotation. This sequence has been used in numerous gait applications and is used by both Kadaba and Davis. The yxz sequence provides the most clinically accurate representation of joint motion.

A drawback to using Euler sequences is a possibility that extreme rotations about a single axis could produce artifact motion along the other axes or lock an axis so that it cannot rotate. This drawback was not an issue in this work, since the bending tasks were designed to isolate motion along a single axis to prevent these issues.

The model is dependent on subject-mounted retro-reflective markers that are subject to motion artifact caused by the movement of adipose and muscle tissue below the skin. Filtering is commonly used to reduce the skin motion artifact and, in this work a Butterworth filter was employed. The Butterworth filter is a proven filter capable of reducing the effects of the motion artifact; however, subjects with a high BMI require a

greater level of filtering (i.e., using lower cutoff frequencies) which can attenuate the motion profiles. The testing methodology proposed in this work requires patients to have a BMI below 29 to ensure accurate data.

4.2 Validation

The validation of this model was essential to prove that the method proposed in this work to diagnose and track the progression of adolescent idiopathic scoliosis was accurate. The validations were designed to demonstrate that the model could produce and reproduce the desired measurements in both the static calculations and dynamic analysis.

The first validation test was designed to ensure that the Matlab code correctly calculate angles based on the markers' positional data. A manual goniometer was used to create known angles with three retro-reflective markers. The results were considered accurate if the calculated angle were within 5° of the known angle. This benchmark was chosen since the reliability of intra-observer measures derived from radiographs is 5° . The results showed the calculated angles had a maximum error of 4° and a mean error of 2.3° . Therefore, the model is capable of reproducing known angles with less error than radiographs. It was noted that the accuracy of the model improved as the angles increased to 15° and greater, this could be caused by the error inherent to both the motion analysis system and manual goniometer affecting the accuracy of the small angles.

Replicating the measures of known angles was important to demonstrate that the model could accurately calculate the angles from the motion capture data. It was also important to demonstrate that the motion analysis system could accurately produce Cobb angle measures based on retro-reflective marker data; therefore the proximal angle data

was examined as a representative angle. The results showed that angles calculated by the Matlab and marker model were similar to those measured from the radiographs using the same spinal landmarks. Furthermore, the difference between the calculated angles and radiographic measures was $3 \pm 2^\circ$ indicating that the model was as accurate as the radiographic intra-observer reliability. The results showed that the precision, of the model was within one degree, once again the model showed that it was more precise than radiographic intra-observer reliability. This indicates that the methods of calculating angles based on retro-reflective markers were valid, and could be used to assess spinal deformity in patients with idiopathic scoliosis.

The dynamic portion of this model was based on the calculations described by Kadaba [15]; however, programming the model in Matlab required minor modifications to ensure the code functioned appropriately. The most apparent change was the use of `atan2` function in place of the inverse sine function described by Kadaba. To validate the model a direct comparison between the Vicon kinematic output and the Matlab kinematic output was used to validate the dynamic portion of this work. A 2° difference between the Matlab output and Vicon output was allowed to account for variations in filtering methods; the Vicon model filters the motion data using a Woltering filter, while the model in this paper used a Butterworth filter. Based on the results of this validation testing, the dynamic modeling used in this can accurately assess the kinematics of subjects with normal spines and pathological spines.

4.3 Static Analysis

The static application of this work describes the spinal curvature of a patient. This was accomplished by calculating a postural alignment based on the subject's center of pressure (COP), where the postural alignment was directly related to the shifting of the subject's weight on the force plate. Based on these results, there are differences in the postural alignment between the two study groups as seen in Figure 35a and 35b.

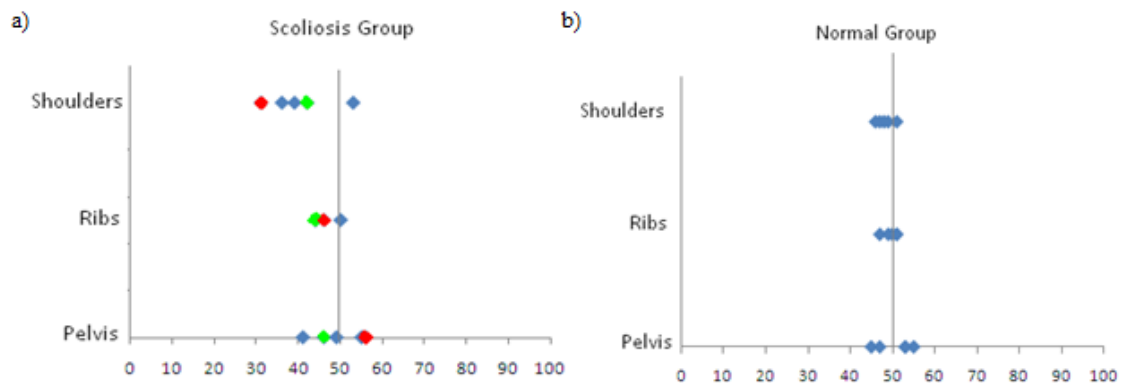


Figure 35: Center of pressure for the scoliosis group and control group

The center of pressure vector passed through the center of the pelvis for both the scoliosis and control group. As the vector traveled superior from the pelvis, the center of pressure shifted to the right for both groups; however, the vector was shifted more to the right in the subjects with scoliosis. This postural shift was related to the curvature of the spine. Common adolescent idiopathic scoliosis patterns are right major curves with up to two compensatory left curves. Since the major curve is to the right, (i.e., the COP shifted to the right) the subject shifts their weight to the right. In those subjects that had more balanced curves, (e.g., the subject highlighted in green in Figure 35a), the right and left curves were near equal in magnitude. The subject in Figure 35 highlighted in green had a 46° left high thoracic curve and a 48° right main thoracic curve and a more centered

COP was observed as was seen in the control group. When the curves were more unbalanced, (e.g., the subject highlighted in red in Figure 35a) the COP shifts in the direction of the larger curve. In this case, the high left thoracic curve of 18° does not compensate for the 46° right main thoracic curve shifting the vector to the right at the shoulders. In the singular case where a subject's major curve was a thoracolumbar left curve, the COP vector was shifted to the left at the pelvis, mid-thorax, and shoulder. These results indicate that the postural alignment could be used to determine curve direction and curve balance. As the curve magnitude increases and the curves become less balanced, it is likely that that center of pressure will shift towards the curve with the greatest magnitude. Based on the results of the postural alignment calculations, this measure could be used to track curve progression in patients with adolescent idiopathic scoliosis, although further investigation is required prior to making a definitive conclusions. This investigation would require a prospective longitudinal study to monitor changes in the COP as curves progressed.

The frontal and transverse plane asymmetries calculated in this work provide a quantitative description of shoulder and pelvis asymmetry in the coronal and transverse planes, which are visually assessed by physicians during clinical visits. The results indicated the scoliosis group had slightly more asymmetry in the frontal plane than the control group, especially at the pelvis. The results also showed that the scoliosis group had greater variation in frontal plane asymmetry than the control group. The transverse plane analysis showed difference in postural symmetry between the two groups with the scoliosis subjects having an external pelvic and mid-thoracic rotation. The external rotation at the pelvis could be caused by the rotational deformity of the spine and

subsequent underlying muscle deformity retracting the right side of the pelvis. Those subjects with more severe curves tended to have more rotational asymmetry. A possible source of error in the asymmetry calculations was the retro-reflective marker placement. Minor deviations in marker placement could cause the calculations to falsely indicate the amount of postural asymmetry. Careful marker placement is essential to reducing marker artifact as it still remains a source of confounding error. The information provided from these calculations should be checked through visual assessment of the subject to ensure that these measures accurately describe the postural asymmetry.

The ability of the motion capture system to accurately assess the spinal deformity of the subjects was essential to proving that motion analysis methods could reduce or replace radiographs for patients with AIS. The proximal and distal angles were chosen to represent Cobb angle measures and determine if these measures could be replicated with the motion analysis technology. The SVA and CVA are common measures used to determine spinal balance and were chosen to ensure that distance calculations could also be replicated. Finally, the measures of kyphosis and lordosis were also chosen to determine how accurately the model could replicate the spine's sagittal profile.

The results of the radiographic correlations showed that there were high correlations between those values derived from radiographs and those calculated based on the static model and motion capture data. The SVA and CVA showed highly correlated values, 0.92 and 0.86 respectively, with those measured from the radiographs, and the correlations were similar for both the scoliosis and control groups. Based on the results presented in this work it is possible to replicate the SVA and CVA values and describe the spinal symmetry using the motion analysis technology. The proximal and

distal angles also showed high correlations between the radiographic measures and the measures calculated based on the motion data. It is important to note that the values presented in this work were based on comparing the radiographic angles measures based on the angles measured with a T7 apex and end vertebra of C7 and L3 and with an apex of L3 and end vertebra of T7 and S1, and not the actual Cobb angles that are used to describe the coronal plane deformity of the spine in patients with scoliosis. The correlation values decrease substantially when comparing the actual Cobb angles to the calculated proximal angle ($r=0.37$) and distal angle ($r=0.45$), which is caused by a limitation of the current model. The current model does not have enough markers on the subject's spine to adequately capture all of the spinal curves; however, the correlation values in the control group remained high indicating that with additional markers a more accurate representation of the Cobb angles can be produced in the scoliosis group. The marker set and the model was unable to accurately replicate the sagittal profile measures as evident by the low correlation values, which was due to the muscle and skin bulk that covers the spine. In one case the (i.e., subject 10), was very slender with minimal muscle bulk, and the correlation value increased to 0.63, while not a highly correlated value this result suggests that this error is originating from skin bulk. Further work is required to determine if there is a method to account for subject specific muscle thickness to improve the accuracy of this measure.

The results of the relaxed standing values correlated well with the radiographic value but were less than those with the X-ray frame. This indicates that the frame was capable of standardizing the position between radiographs and the motion capture.

The results of the static analysis portion of this work demonstrates that motion analysis technology provides a suitable means of reducing the need for repeated radiographs. Further work is required to modify the marker model to increase the accuracy of describing the true Cobb angles and additional work is also required to determine how sensitive this model is at determining curve progression over time.

4.4 Dynamic Analysis

The ability of the motion capture system to accurately replicate the static measures obtained from radiographs is essential to proving that the motion capture methods could be a viable alternative to x-rays but motion analysis is capable of providing more information about the dynamic range of motion of the spine. Information about losses in the spinal range of motion caused by the scoliosis and compensatory motion can be as important in diagnosing and describing the spinal deformity as the static measures.

Each of the bending tasks were designed to isolate motion to a specific plane. The lateral bending isolated coronal plane motion, the axial rotation isolated transverse plane motion, and the flexion/extension task isolated sagittal plane motion. While the differences in the primary motion plane between the two groups is important, it is also important to understand the compensation in the other two planes to describe the spinal pathology.

The results of the coronal plane bending task showed similar spinal and anatomic segment ranges of motion between the two study groups; the only exception was the upper spine motion in the transverse plane. The results indicated more rotational

movement of the spine and slightly greater overall range of motion in the scoliosis group; indicating that the scoliosis group axially rotates to perform the lateral bending motion.

The results of the transverse plane bending task showed differences in both the anatomic segments and the spinal segments. The primary motion (i.e., the transverse plane motion) was similar for both groups and the overall motion of the scoliosis group was slightly greater than the control group. The compensatory motion was very different between the two groups. Results showed the scoliosis group had a greater amount of compensatory motion in the sagittal and coronal planes. The anatomic segment motion profiles in the scoliosis group showed that subjects with scoliosis tilt their pelvis up in the direction of rotation indicating that the spinal deformity and underlying muscle deformity pull upward on the pelvis; therefore, the subjects with scoliosis had to modify their pelvic obliquity to achieve the same rotational range of motion as the control group. A similar pattern could also be seen in sagittal profile of the upper thorax. The scoliosis group tended to flex slightly more than the control group suggesting the presence of a means of compensating for the underlying muscle deformity. By increasing their thoracic flexion, they reduce the strain on the muscles while performing the axial rotation task. A number of compensatory motions were trending towards statistical significance, especially the pelvic coronal plane with a p-value of 0.07 and a 10° difference in motion. Although the sagittal plane angle was approaching statistical significance, the difference in motion was only 3° and should not be considered clinically significant.

The spinal ranges of motion for the axial rotation showed similar results to those of the anatomic planes. The coronal plane motion between the two study groups was similar and the measured differences that were found were not statistically significant.

The motion pattern for the scoliosis group was opposite of the control group. As the spine rotates internally and the spine also laterally leans to the right, while the control group leaned to the left. This behavior was most obvious in the lower spine. This motion pattern indicates that the scoliosis subjects lean in an effort to counter balance the deformity of the spine. The motion endpoints for the spinal rotation in the control group is more symmetric, with an internal and external rotation of 53° , while the scoliosis group had 66° of internal rotation and 45° of external rotation. This rotation asymmetry indicates that both the transverse and coronal plane deformity, caused by the scoliosis, restricts the subject's ability to externally rotate. The underlying musculature restricts external rotation because the muscles are strained and, the subject rotates internally, the muscles relax allowing for greater rotation.

The flexion/extension task also showed minor differences between the two study groups in both the anatomic and spinal segments. The flexion/extension task was the only task in which the control group had a greater range of motion than the scoliosis group suggesting that the rotational and coronal plane spinal deformities reduce sagittal plane motion. The greatest difference in motion between the two study groups was found in the upper thorax sagittal plane motion, which had a 15° difference between the two groups and approached a level of significance. There were statistically significant differences observed in the transverse plane motion of the pelvis and a near statistically significant difference in the transverse plane at the mid-thorax; however, with differences of 2° and 4° respectively, these differences cannot be considered clinically relevant. While the spinal segment ranges of motion showed similar results to the anatomic planes for the flexion/extension task, the greatest difference in the spinal segments was observed to be

in the lower spine and in sagittal plane of motion. The scoliosis group had an 8° reduction in lower spine motion and this difference was trending toward statistical significance.

4.5 Clinical considerations

The results showed that motion analysis technology can provide a suitable alternative to radiographs in the diagnosis and of adolescent idiopathic scoliosis. For methods described in this work to become common clinical practice, physicians must be comfortable with the data and data calculations. Radiographs can be quickly read and visually represent the spinal deformity but, the data provided by the methodology described in this paper provides more information than radiographs, especially when describing the rotational aspect of the spinal deformity and the underlying muscular deformity. This methodology requires the interpretation of a large amount of numeric data and future work on this protocol would require the information to be presented in a more graphical way, to simplify the interpretation process. These graphical depictions could include simple plots illustrating the computed radiographic measures in a format similar to radiographs. An example is shown in Figure 36a and 36b.

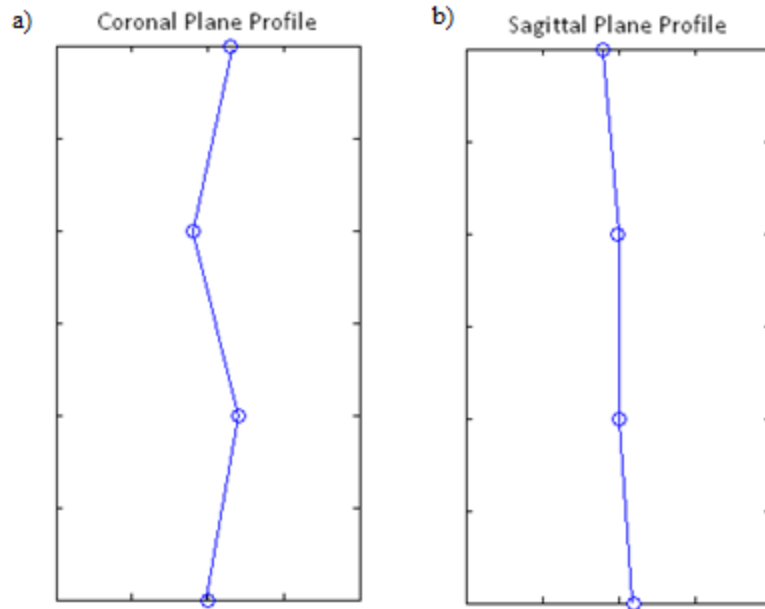


Figure 36: Graphical representation of the Cobb angles (a) and angles of kyphosis and lordosis (b)

The center of pressure and asymmetry data could be overlaid over a static image to illustrate the information to making it easier to understand as seen in Figure 37a-c.

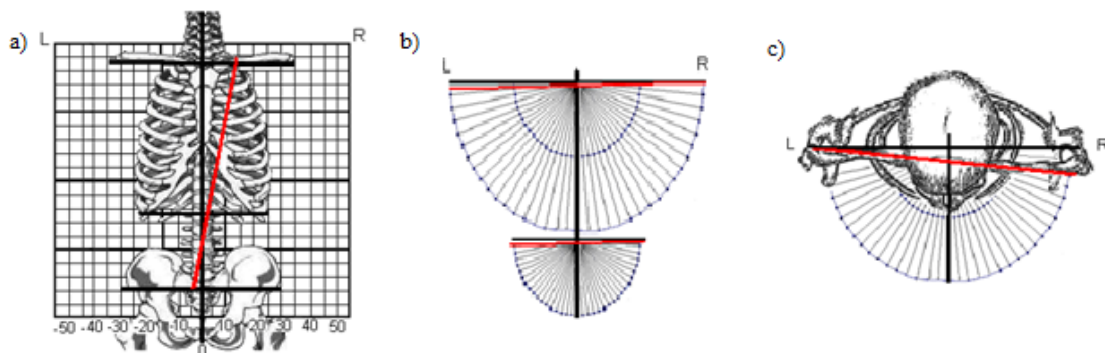


Figure 37: Illustrations to depicting the center of pressure (a), frontal plane asymmetry (b), and rotational asymmetry (c).

A second consideration is that this work was performed in a specially designed motion analysis laboratory. Many institutions cannot afford the space or equipment required to design such a lab and, if this methodology was to become more common place in scoliosis clinics it would be impractical to send patients for a special time –

intensive analyses. Although the methodology presented in this work utilized 12 motion cameras, much of the static analysis could be performed using three well-positioned cameras and a single force platform. The dynamic analysis would require at least six cameras to ensure that the markers could be tracked through the entire range of motion without losing markers due to marker occlusion. Reducing the number of cameras would allow a “miniature lab” to be established in a room the size of a room used to house an X-ray machine. Minor modifications to the Matlab code presented in this work would allow for it to be integrated into the motion capture software as a plugin, so that results can be available for analysis in the same amount of time as radiographs.

Finally, the cost of this test is also of concern in a clinical setting. The current billable cost for a scoliosis radiographic series (i.e., PA and lateral radiograph) at Connecticut Children’s Medical Center is \$125. While there is currently no billable cost for the methodology presented in this work it, the estimated cost for the motion analysis is \$225, assuming an average salary including benefits for a full time technician, a thirty minute test, and including a 100% mark up to cover the cost of the equipment and overhead. Although the estimated cost for motion analysis is nearly double the current cost of radiographs, it provides more information about the spinal deformity than current radiographs, and there is no associated health risks associated with the motion analysis technology. It is also important to note that a supine MRI can give an understanding of the coronal and rotational deformity of the spine and information about the underlying musculature; however, the MRI would not be able to provide pressure data or an accurate representation of the spinal balance especially since the subject is non- weight bearing.

The cost of such an MRI, performed at Connecticut Children’s Medical Center, such , not

including the cost of a radiologist to read the image, and the accounting for any mark up in price for over head is \$350; therefore, the motion analysis method is a safer and more cost effective option.

4.6 Limitations

Many of the values presented in the results section of this work were not statistically significant. The reason for this is that this work has a limited sample size of five subjects in each study group. With a larger sample size, it is possible that many of the values that have a p-value of 0.20 or lower will become significant. Future work on this topic should include at least 10 subjects in the normal group and 20 subjects in the scoliosis group. The larger sample for the scoliosis group would allow for a greater variation of spinal curvature thus providing results for a more generalized idiopathic scoliosis population.

Another limitation of this work is the marker set. Although it is capable of reproducing radiographic measures, it does not have the detail required to truly describe the coronal plane deformity of a scoliotic spine. Future work on this topic should include the development of a more complex marker model that would have four additional markers: two above the T7 marker and two below the T7 marker. These additional markers should allow for a more accurate description of the Cobb angles.

Finally, another weakness of this work is its reliance on marker placement. Minor deviations in marker placement could cause significant changes in both the static analysis and kinematic calculations. Furthermore, the data is filtered to reduce skin motion artifact and it is essential that care is taken to ensure proper marker placement.

4.7 Future work

Future work on this topic would involve further study of the rotational information provided by the motion analysis technology. Currently surgeons do not have a clear understanding of the rotational aspect of the spinal deformity until they have begun the surgery and are able to visualize the spine. The correction of the rotational deformity is then planned and performed during the actual surgery. Further information about the rotational aspect of the deformity would make it possible for surgeons to create better preoperative strategy and to possibly reduce surgery time. Future work on this topic would include correlating the transverse asymmetry and dynamic motion asymmetry in the transverse plane motion, with the amount spinal de-rotation performed during the surgery.

Another area of future work on this topic would be to determine the sensitivity of the model to minor changes in the scoliosis as the curves progress. The data presented in this work showed that the model can accurately calculate the curvature of a normal spine and a spine with scoliosis; however, the results do not indicate if the model could assess minor changes in the curves as the scoliosis progresses. Future studies would include following a group of subjects over the course of their treatment to determine how accurately the model could track the progression of the scoliosis over time and how much of an increase in spinal curvature would be required to indicate a difference. The results of this longitudinal study would also help to determine how many radiographs could be replaced by the motion analysis.

Finally, the dynamic motion analysis described in this work was designed to standardize the timing and the speed of the motions. Future work would include

modifying these tasks to allow the subject to perform the task at a self selected speed. A self-selected speed would allow for an understanding of the angular velocities accelerations, and decelerations of the subject's motion and ultimately the spine. The data provided by this analysis could indicate if the velocity and acceleration profiles could be used to explain why a patient's curve progresses or provide additional information about the spinal deformity.

5.0 Conclusion

This work presents a viable means of describing the spinal deformity in a patient population diagnosed with adolescent idiopathic scoliosis. The static analysis showed that high correlations between radiographic measurements and motion analysis calculations can be achieved with a simple marker configuration and a series of trigonometric equations. Using this static analysis methodology, it is possible to describe the spinal balance and provide a means of quantitatively describing a subjective clinical measure to describe postural asymmetry. Furthermore, the static analysis can replicate some of the radiographic measures with greater precision than the current gold standard for diagnosing and tracking the progression of scoliosis.

The kinematic analysis can provide information on the spine's range of motion. Differences seen in the motion profiles between the control and scoliosis groups indicate subtle differences in the total range of motion in the degree of symmetry in side-to-side motion. This information could describe the underlying muscular deformities caused by the scoliosis. It could also be used to describe the rotational deformity of the spine inherent to scoliosis but is difficult to determine using standard plain radiographs.

This work represents the foundation for future work on using motion analysis technology on patients with adolescent idiopathic scoliosis in order to minimize their exposure to repeated spinal radiographs; thereby reducing the risks inherent to cumulative doses of ionizing radiation. While the method presented in this work for diagnosing adolescent idiopathic scoliosis may not be able to completely alleviate the need for radiographs, it may provide a viable means of reducing the total number of radiographs that patients receive by up to fifty percent.

6.0 References

- [1] H Grey. Grey's Anatomy 15th edition. Barnes and Noble; New York: 1995.
- [2] N Miller. "Cause and Natural History of Adolescent Idiopathic Scoliosis." *Orthopedic Clinics of North America*. Vol. 30 issue 3, pages 343-352, July 1999
- [3] Tortora and Grabwski. Principles of Anatomy and Physiology 10th edition. John Wiley and Sons; New Jersey: 2003.
- [4] S Weinstein, LA Dolan, JC Chang, A Danielsson, and J Morcuende. "Adolescent Idiopathic Scoliosis." *Lancet*. Vol. 371, pages 1527-1537, 2008.
- [5] DG Little and M Sussman. "The Risser Sign: A Critical Analysis." *Journal of Pediatric Orthopaedics*. Vol.14 issue 5, pages 569-575, 1994.
- [6] CM Bone and G Hsieh. "The Risk of Carcinogenesis from Radiographs to Pediatric Orthopaedic Patients." *Journal of Pediatric Orthopaedics*. Vol. 20 issue 2, pages 251-254, 2000.
- [7] M Marks, C Stanford, A Mahar, and P Newton. "Standing Lateral Radiographic Positions does not Represent Customary Standing Balance." *Spine*. Vol. 28 issue 11, pages 1176-1182, 2003.
- [8] C William, J Jorg, an T Grossman. Examination and Diagnosis of Musculoskeletal Disorders: Clinical Examination and Imaging Modalities. Georg Thieme Verlag; New York: 2001.
- [9] S Donaldson, D Stephens, A Howard, B Alman, U Narayana, and J Wright. "Surgical Decision Making in Adolescent Idiopathic Scoliosis." *Spine*. Vol. 32 issue 4, pages 1526-1532, 2007.
- [10] CL Nash, EC Gregg, RH Brown, and K Pillai. "Risks of Exposure to X-rays in Patients Undergoing Long Term Treatment for Scoliosis." *Journal of Bone and Joint Surgery*. Vol. 61, pages 371-374, 1979.
- [11] JE Gray, AD Hoffman, and HA Peterson. "Reduction of Radiation Exposure during Radiography for Scoliosis." *Journal of Bone and Joint Surgery*. Vol. 65, pages 5-12, 1983.
- [12] MS Goldberg, NE Mayo, and AR Levy. "Adverse reproductive outcomes among women exposed to low levels of ionizing radiation from diagnostic radiography for adolescent idiopathic scoliosis." *Epidemiology*. Vol. 9, issue 3, pages 271-278, 1998.
- [13] DB Chaffin, GBJ Andersson, and BJ Martin. Occupational Biomechanics 4th edition. John Wiely and Sons; New Jersey: 2006.
- [14] DR Peterson, and JD Bronzio. Biomechanics Principles and Applications. CRC press; New York: 2008.
- [15] MP Kadaba, HK Ramakrishnan, and ME Wootten. "Measurement of Lower Extremity Kinematics During Level Walking." *Journal of Orthopaedic Research*. Vol. 8 issue 3, pages 383-392, 1990.
- [16] C Frigo, R Carabalona, MD Mura, and S Negrini. "The Upper Body Segmental Movements During Walking by Young Females." *Clinical Biomechanics*. Vol. 18, pages 419-425, 2003.
- [17] JR Engsberg, L Lenke, A Reintenbach, K Hullander, K Bridwell, and K Blanke. "Prospective Evaluation of Trunk Range of Motion in Adolescents with Idiopathic

Scoliosis Undergoing Spinal Fusion Surgery.” *Spine*. Vol. 27 issue 12, pages 1346-1354, 2002.

[18] P Cerveri, A Pedotti, and G Ferrigno. “Non-Invasive Approach Towards the In-vivo Estimation of 3D Inter-vertebral Movements: Methods and Preliminary Results.”

Medical Engineering and Physics. Vol. 26, pages 841-853, 2004.

[19] PD Masso and GE Gorton. “Quantifying Changes in Standing Body Segment Alignment Following Spinal Instrumentation and Fusion in Idiopathic Scoliosis Using an Optoelectronic Measurement System.” *Spine*. Vol. 25, issue 4, pages 457-462, 2000.

[20] W Skalli, RD Zeller, L Miladi, G Bourcereau, M Savidan, F Lavaste, and J Duboussier. “Importance of Pelvic Compensation in Posture and Motion After Posterior Spinal Fusion Using CD Instrumentation for Idiopathic Scoliosis.” *Spine*. Vol. 31 issue 12 pages E359-E366, 2006.

[21] J Harlaar, A Morris. *ReadC3D.m*

<http://www.cs.cmu.edu/~akalamda/kdc/project/code/readC3D.m> April 2002.

[22] RB Davis, S Ounpuu, D Tyburski, and JR Gage. “A Gait Analysis Data Collection and Reduction Technique.” *Human Movement Science*. Vol. 10, issue 5, pages 575-585, 1991.

7.0 Appendix A

This appendix contains the Matlab code used in this work.

Spine

```
% 7-20-09~MS
%This code is the launcher for the spine program that will calculate both the static and dynamic portions of
the test. %The static portion will calculate angles, COP and standing symmetry, while the dynamic will
evaluate spinal %motion.
curr=cd;
clear all;
clear all;
% Reading C3D files:
cpath = input('Enter the location of the patient: ','s');
patnum = input('Enter the Hospital number for the patient: ','s');
evalnum= input('Enter the evaluation number for the test: ','s');
condition=input('Enter the condition name: ','s');
    spinereader;
    spinefilter;
%Calculating static measurements:
load([cpath,patnum,'\Evaluation ',evalnum,'\condition,'\patnum,'_reader.mat']);
trialnum(1)=input('Input the first standing trial: ');
trialnum(2)=input('Input the second standing trial: ');
trialnum(3)=input('Input the third standing trial: ');
trialnum(4)=input('Input the first frame trial: ');
trialnum(5)=input('Input the second frame trial: ');
trialnum(6)=input('Input the third frame trial: ');
for i=1:length(trialnum)
    framerange {trialnum(i)}=(1:size(S(trialnum(i)).C7,1));
end
    spinestatic;
    output;
%Calculating dynamic ROM measurements:
trialn(1)=input('Input the lateral bend trial: ');
trialn(2)=input('Input the rotational trial: ');
trialn(3)=input('Input the flexion trial: ');
for i=1:length(trialn)
    framerange {trialn(i)}=(1:size(S(trialn(i)).C7,1));
end
    spinecalc;
    spineanalyze;
    spinegraph;
    save([cpath,patnum,'\Evaluation ',evalnum,'\condition,'\patnum,'_final.mat']);
    dynamicoutput;
```

Spinereader

```
% This code will read the C3D file into the Matlab environment.
file=dir([cpath,patnum,'\Evaluation ',evalnum,'\condition','*.c3d']);
% Sorting the trials like we do with pitching-single digits first then double digit trials
xxx=1;
for zzz=1:length(file)
    temp=file(zzz).name;
    ind1=findstr(temp,'')+1; ind2=findstr(temp,'.')->1;
    if ind1==ind2
```

```

        files(xxx)=file(zzz); xxx=xxx+1; end, end
for zzz=1:length(file)
    temp=file(zzz).name;
    ind1=findstr(temp,'')+1; ind2=findstr(temp,'.')->1;
    if ind1~=ind2
        files(xxx)=file(zzz); xxx=xxx+1;
    end
end
file=files;
for zzz=2:length(file)% read each file
    file_name=file(zzz).name;
    % extract the trial number from the filename
    ind1=findstr(file_name,'')+1; ind2=findstr(file_name,'.')->1;
    if ind1~=ind2
        trialnumber=str2num(file_name((ind1):(ind2)));
    else
        trialnumber=str2num(file_name(ind1));
    end
    % Read C3D file with a code by Alan Morris and later revised by Jaap Harlaar found on the C3D
    website.

[Markers,VideoFrameRate,AnalogSignals,AnalogFrameRate,Event,ParameterGroup,CameraInfo,Residual
Error]= ... readC3D_test([cpath,patnum,'\Evaluation ',evalnum,'\',condition,'\',file_name]);
samrate = VideoFrameRate;
bbutt=1;
abutt=1;
[bbutt,abutt]=butter(2,10/(samrate/2));
% From the parameter block of the c3d file reading the order and data
% stored in the c3d file.
for i=1:length(ParameterGroup),
    if strcmp('TRIAL',ParameterGroup(i).name,'exact')
        trialGroupID=i;
    end
end
% recording the parameter ID for the parameter
for i=1:length(ParameterGroup(trialGroupID).Parameter),
    if strcmp('ACTUAL_START_FIELD',ParameterGroup(trialGroupID).Parameter(i).name,'exact')
        trialParameterStartID=i;
        S(trialnumber).START=ParameterGroup(trialGroupID).Parameter(trialParameterStartID).data(1);
    end
    if strcmp('ACTUAL_END_FIELD',ParameterGroup(trialGroupID).Parameter(i).name,'exact')
        S(trialnumber).END=ParameterGroup(trialGroupID).Parameter(i).data(1);
    end
end
for i=1:length(ParameterGroup),
    if strcmp('POINT',ParameterGroup(i).name,'exact')
        pointGroupID=i;
    end
end
% recording the parameter ID for the parameter POINT.LABELS
for i=1:length(ParameterGroup(pointGroupID).Parameter),
    if strcmp('LABELS',ParameterGroup(pointGroupID).Parameter(i).name,'exact')
        pointParameterLabelsID=i;
    end
end
for i=1:length(ParameterGroup),

```

```

    if strmatch('ANALOG',ParameterGroup(i).name,'exact')
        analogGroupID=i;
    end
end
%The Event group allows for markers to be placed in on the VICON side
%to specify points of maximum movement.
for i=1:length(ParameterGroup),
    if strmatch('EVENT',ParameterGroup(i).name,'exact')
        eventGroupID=i;
    end
end
for i=1:length(ParameterGroup(eventGroupID).Parameter),
    if strmatch('USED',ParameterGroup(eventGroupID).Parameter(i).name,'exact')
        S(zzz).numevent=ParameterGroup(eventGroupID).Parameter(i).data(1);
    end
    if strmatch('TIMES',ParameterGroup(eventGroupID).Parameter(i).name,'exact')
        S(zzz).eventtime=(ParameterGroup(eventGroupID).Parameter(i).data);
    end
end
S(zzz).eventside=char(S(zzz).eventside);
[S(zzz).eventtime_sort,ind]=sortrows(S(zzz).eventtime,2);
for i=1:length(ParameterGroup),
    if strmatch('FORCE_PLATFORM',ParameterGroup(i).name,'exact')
        forceplatformGroupID=i;
    end
end
% recording the parameter ID for the parameter ANALOG.USED to check if the force plate data was
recorded or %not.
for i=1:length(ParameterGroup(analogGroupID).Parameter),
    if strmatch('USED',ParameterGroup(analogGroupID).Parameter(i).name,'exact')
        if ParameterGroup(analogGroupID).Parameter(i).data>0
            forcepresent=1;
        else
            forcepresent=0;
        end
    end
end
if forcepresent==1
    % recording the parameter ID for the parameter ANALOG.LABELS
    for i=1:length(ParameterGroup(analogGroupID).Parameter),
        if strmatch('LABELS',ParameterGroup(analogGroupID).Parameter(i).name,'exact')
            analogParameterLabelsID=i;
        end
        if strmatch('USED',ParameterGroup(analogGroupID).Parameter(i).name,'exact')
            S(trialnumber).Nanalogchannels=ParameterGroup(analogGroupID).Parameter(i).data;
        end
        if strmatch('GAIN',ParameterGroup(analogGroupID).Parameter(i).name,'exact')
            S(trialnumber).analog_gain=ParameterGroup(analogGroupID).Parameter(i).data;
        end
        if strmatch('SCALE',ParameterGroup(analogGroupID).Parameter(i).name,'exact')
            S(trialnumber).analog_scale=ParameterGroup(analogGroupID).Parameter(i).data;
        end
        if strmatch('OFFSET',ParameterGroup(analogGroupID).Parameter(i).name,'exact')
            S(trialnumber).analog_offset=ParameterGroup(analogGroupID).Parameter(i).data;
        end
    end
end

```

```

end
for i=1:length(ParameterGroup(forceplatformGroupID).Parameter),
    if strcmp('CORNERS',ParameterGroup(forceplatformGroupID).Parameter(i).name,'exact')
        S(trialnumber).forceplate_corners=ParameterGroup(forceplatformGroupID).Parameter(i).data;
    end
    if strcmp('ORIGIN',ParameterGroup(forceplatformGroupID).Parameter(i).name,'exact')
        S(trialnumber).forceplate_origin=ParameterGroup(forceplatformGroupID).Parameter(i).data;
    end
end
for i=1:length(ParameterGroup(pointGroupID).Parameter(pointParameterLabelsID).data)
    % reading the marker coordinates and calculated bone markers.
    % ===== Desired kinematic variables =====
    if strcmp('C7',ParameterGroup(pointGroupID).Parameter(pointParameterLabelsID).data(i),'exact')
        S(trialnumber).C7= Markers(:,i,:); i_C7=i; end
    if
    strcmp('LCLA',ParameterGroup(pointGroupID).Parameter(pointParameterLabelsID).data(i),'exact')
        S(trialnumber).LCLA= Markers(:,i,:); i_LCLA=i;end
    if
    strcmp('RCLA',ParameterGroup(pointGroupID).Parameter(pointParameterLabelsID).data(i),'exact')
        S(trialnumber).RCLA= Markers(:,i,:); i_RCLA=i;end
    if strcmp('T7',ParameterGroup(pointGroupID).Parameter(pointParameterLabelsID).data(i),'exact')
        S(trialnumber).T7= Markers(:,i,:); i_T7=i; end
    if strcmp('LRIB',ParameterGroup(pointGroupID).Parameter(pointParameterLabelsID).data(i),'exact')
        S(trialnumber).LRIB= Markers(:,i,:); i_LR10=i; end
    if
    strcmp('RRIB',ParameterGroup(pointGroupID).Parameter(pointParameterLabelsID).data(i),'exact')
        S(trialnumber).RRIB= Markers(:,i,:); i_RR10=i; end
    if strcmp('L3',ParameterGroup(pointGroupID).Parameter(pointParameterLabelsID).data(i),'exact')
        S(trialnumber).L3= Markers(:,i,:); i_T12=i; end
    if
    strcmp('SACR',ParameterGroup(pointGroupID).Parameter(pointParameterLabelsID).data(i),'exact')
        S(trialnumber).SACR= Markers(:,i,:); i_SACR=i; end
    if strcmp('LASI',ParameterGroup(pointGroupID).Parameter(pointParameterLabelsID).data(i),'exact')
        S(trialnumber).LASI= Markers(:,i,:); i_LASI=i; end
    if strcmp('RASI',ParameterGroup(pointGroupID).Parameter(pointParameterLabelsID).data(i),'exact')
        S(trialnumber).RASI= Markers(:,i,:); i_RASI=i; end
    end % end of reading the Markers information
    if forcepresent==1
        for i=1:length(ParameterGroup(analogGroupID).Parameter(analogParameterLabelsID).data)
            % ===== Desired kinetic variables =====
            if
            strcmp('Fx1',ParameterGroup(analogGroupID).Parameter(analogParameterLabelsID).data(i),'exact')
                S(trialnumber).Fx1= AnalogSignals(:,i,:); i_Fx1=i; end
            if
            strcmp('Fy1',ParameterGroup(analogGroupID).Parameter(analogParameterLabelsID).data(i),'exact')
                S(trialnumber).Fy1= AnalogSignals(:,i,:); i_Fy1=i; end
            if
            strcmp('Fz1',ParameterGroup(analogGroupID).Parameter(analogParameterLabelsID).data(i),'exact')
                S(trialnumber).Fz1= AnalogSignals(:,i,:); i_Fz1=i; end
            if
            strcmp('Mx1',ParameterGroup(analogGroupID).Parameter(analogParameterLabelsID).data(i),'exact')
                S(trialnumber).Mx1= AnalogSignals(:,i,:); i_Mx1=i; end
            if
            strcmp('My1',ParameterGroup(analogGroupID).Parameter(analogParameterLabelsID).data(i),'exact')
                S(trialnumber).My1= AnalogSignals(:,i,:); i_My1=i; end

```

```

        if
strmatch('Mz1',ParameterGroup(analogGroupID).Parameter(analogParameterLabelsID).data(i),'exact')
        S(trialnumber).Mz1= AnalogSignals(:,i,:); i_Mz1=i; end
        if
strmatch('Fx2',ParameterGroup(analogGroupID).Parameter(analogParameterLabelsID).data(i),'exact')
        S(trialnumber).Fx2= AnalogSignals(:,i,:); i_Fx2=i; end
        if
strmatch('Fy2',ParameterGroup(analogGroupID).Parameter(analogParameterLabelsID).data(i),'exact')
        S(trialnumber).Fy2= AnalogSignals(:,i,:); i_Fy2=i; end
        if
strmatch('Fz2',ParameterGroup(analogGroupID).Parameter(analogParameterLabelsID).data(i),'exact')
        S(trialnumber).Fz2= AnalogSignals(:,i,:); i_Fz2=i; end
        if
strmatch('Mx2',ParameterGroup(analogGroupID).Parameter(analogParameterLabelsID).data(i),'exact')
        S(trialnumber).Mx2= AnalogSignals(:,i,:); i_Mx2=i; end
        if
strmatch('My2',ParameterGroup(analogGroupID).Parameter(analogParameterLabelsID).data(i),'exact')
        S(trialnumber).My2= AnalogSignals(:,i,:); i_My2=i; end
        if
strmatch('Mz2',ParameterGroup(analogGroupID).Parameter(analogParameterLabelsID).data(i),'exact')
        S(trialnumber).Mz2= AnalogSignals(:,i,:); i_Mz2=i; end
end % end of reading the analog data.
% ***** CALCULATING THE COP *****
% The center of the force plates at CCMC, calculated from the force plate
% corners.
S(trialnumber).forceplate_corners(:,1) = ...
[-141.5000 -141.2000 -650.0000 -649.7000; 486.8000 22.8000 21.6000 485.6000 ; 0 0
0 0];

S(trialnumber).forceplate_corners(:,2) =...
[464.0000 0 0 464.3000; 0 0 509.0000 509.0000; 0 0 0 0];

center_fp1(1,1) = ((S(trialnumber).forceplate_corners(1,1,1) +
(S(trialnumber).forceplate_corners(1,4,1)-S(trialnumber).forceplate_corners(1,1,1))/2) + ...
(S(trialnumber).forceplate_corners(1,2,1) + (S(trialnumber).forceplate_corners(1,3,1)-
S(trialnumber).forceplate_corners(1,2,1))/2))/2;
center_fp1(1,2) = ((S(trialnumber).forceplate_corners(2,2,1) +
(S(trialnumber).forceplate_corners(2,1,1)-S(trialnumber).forceplate_corners(2,2,1))/2) + ...
(S(trialnumber).forceplate_corners(2,3,1) + (S(trialnumber).forceplate_corners(2,4,1)-
S(trialnumber).forceplate_corners(2,3,1))/2))/2;
center_fp1(1,3) = ((S(trialnumber).forceplate_corners(3,1,1) +
(S(trialnumber).forceplate_corners(3,4,1)-S(trialnumber).forceplate_corners(3,1,1))/2) + ...
(S(trialnumber).forceplate_corners(3,2,1) + (S(trialnumber).forceplate_corners(3,3,1)-
S(trialnumber).forceplate_corners(3,2,1))/2))/2;

center_fp2(1,1) = ((S(trialnumber).forceplate_corners(1,1,2) +
(S(trialnumber).forceplate_corners(1,4,2)-S(trialnumber).forceplate_corners(1,1,2))/2) + ...
(S(trialnumber).forceplate_corners(1,2,2) + (S(trialnumber).forceplate_corners(1,3,2)-
S(trialnumber).forceplate_corners(1,2,2))/2))/2;
center_fp2(1,2) = ((S(trialnumber).forceplate_corners(2,2,2) +
(S(trialnumber).forceplate_corners(2,1,2)-S(trialnumber).forceplate_corners(2,2,2))/2) + ...
(S(trialnumber).forceplate_corners(2,3,2) + (S(trialnumber).forceplate_corners(2,4,2)-
S(trialnumber).forceplate_corners(2,3,2))/2))/2;

```

```

center_fp2(1,3) = ((S(trialnumber).forceplate_corners(3,1,2) +
(S(trialnumber).forceplate_corners(3,4,2)-S(trialnumber).forceplate_corners(3,1,2))/2) + ...
(S(trialnumber).forceplate_corners(3,2,2) + (S(trialnumber).forceplate_corners(3,3,2)-
S(trialnumber).forceplate_corners(3,2,2))/2))/2;

% Determining the ground reaction force locations.
if exist('i_LGroundReactionForce')
    for i=1:length(S(trialnumber).LGroundReactionMoment)
        if S(trialnumber).LGroundReactionForce(i,3)>0.05
            S(trialnumber).LCOP(i,1)=center_fp1(1)-
S(trialnumber).LGroundReactionMoment(i,2)/S(trialnumber).LGroundReactionForce(i,3);

S(trialnumber).LCOP(i,2)=center_fp1(2)+S(trialnumber).LGroundReactionMoment(i,1)/S(trialnumber).LGroundReactionForce(i,3);
            S(trialnumber).LCOP(i,3)=center_fp1(3);
        else
            S(trialnumber).LCOP(i,1)=center_fp1(1);
            S(trialnumber).LCOP(i,2)=center_fp1(2);
            S(trialnumber).LCOP(i,3)=center_fp1(3);
        end
    end
end
if exist('i_RGroundReactionForce')
    for i=1:length(S(trialnumber).RGroundReactionMoment)
        if S(trialnumber).RGroundReactionForce(i,3)>0.05
            S(trialnumber).RCOP(i,1)=center_fp2(1)-
S(trialnumber).RGroundReactionMoment(i,2)/S(trialnumber).RGroundReactionForce(i,3);

S(trialnumber).RCOP(i,2)=center_fp2(2)+S(trialnumber).RGroundReactionMoment(i,1)/S(trialnumber).RGroundReactionForce(i,3);
            S(trialnumber).RCOP(i,3)=center_fp2(3);
        else
            S(trialnumber).RCOP(i,1)=center_fp2(1);
            S(trialnumber).RCOP(i,2)=center_fp2(2);
            S(trialnumber).RCOP(i,3)=center_fp2(3);
        end
    end
end
% Computing the COP locations from the raw analog data requires the ground reaction forces and
moments. %To calculate these values taking the raw data and correcting using the scale and offsets.
S(trialnumber).GRF_fp1(:,1)=-S(trialnumber).Fy1 -
S(trialnumber).analog_offset(1))*S(trialnumber).analog_scale(1);
S(trialnumber).GRF_fp1(:,2)=-S(trialnumber).Fx1 -
S(trialnumber).analog_offset(2))*S(trialnumber).analog_scale(2);
S(trialnumber).GRF_fp1(:,3)=(S(trialnumber).Fz1 -
S(trialnumber).analog_offset(3))*S(trialnumber).analog_scale(3);
S(trialnumber).GRM_fp1(:,1)=-S(trialnumber).My1 -
S(trialnumber).analog_offset(4))*S(trialnumber).analog_scale(4);
S(trialnumber).GRM_fp1(:,2)=-S(trialnumber).Mx1 -
S(trialnumber).analog_offset(5))*S(trialnumber).analog_scale(5);
S(trialnumber).GRM_fp1(:,3)=(S(trialnumber).Mz1 -
S(trialnumber).analog_offset(6))*S(trialnumber).analog_scale(6);
S(trialnumber).GRF_fp2(:,1)=-S(trialnumber).Fx2 -
S(trialnumber).analog_offset(7))*S(trialnumber).analog_scale(7);

```



```

S(trialnumber).GRF_fp2(:,2)=(S(trialnumber).Fy2 -
S(trialnumber).analog_offset(8))*S(trialnumber).analog_scale(8);
S(trialnumber).GRF_fp2(:,3)=(S(trialnumber).Fz2 -
S(trialnumber).analog_offset(9))*S(trialnumber).analog_scale(9);
S(trialnumber).GRM_fp2(:,1)=(S(trialnumber).Mx2 -
S(trialnumber).analog_offset(10))*S(trialnumber).analog_scale(10);
S(trialnumber).GRM_fp2(:,2)=(S(trialnumber).My2 -
S(trialnumber).analog_offset(11))*S(trialnumber).analog_scale(11);
S(trialnumber).GRM_fp2(:,3)=(S(trialnumber).Mz2 -
S(trialnumber).analog_offset(12))*S(trialnumber).analog_scale(12);
% since the sample rates are different for video and force plates this portion of the program reduces the
ground reaction forces and moments to the camera rate. To do this I am using the fixlength program
which will reduce the analog data to match with the video data. this will also filter the analog data using a
moving average with a window of 5.
S(trialnumber).GRF_fp1=fixlength(1:length(S(trialnumber).GRF_fp1),S(trialnumber).GRF_fp1,1:length(S(
trialnumber).GRF_fp1)/(AnalogFrameRate/VideoFrameRate),3,bbutt,abutt);
S(trialnumber).GRM_fp1=fixlength(1:length(S(trialnumber).GRM_fp1),S(trialnumber).GRM_fp1,1:length
(S(trialnumber).GRM_fp1)/(AnalogFrameRate/VideoFrameRate),3,bbutt,abutt);
S(trialnumber).GRF_fp2=fixlength(1:length(S(trialnumber).GRF_fp2),S(trialnumber).GRF_fp2,1:length(S(
trialnumber).GRF_fp2)/(AnalogFrameRate/VideoFrameRate),3,bbutt,abutt);
S(trialnumber).GRM_fp2=fixlength(1:length(S(trialnumber).GRM_fp2),S(trialnumber).GRM_fp2,1:length
(S(trialnumber).GRM_fp2)/(AnalogFrameRate/VideoFrameRate),3,bbutt,abutt);
for i=1:length(S(trialnumber).GRF_fp1)
if S(trialnumber).GRF_fp1(i,3)>5
S(trialnumber).COP1(i,1)=
(S(trialnumber).GRM_fp1(i,2)./S(trialnumber).GRF_fp1(i,3))+center_fp1(1);
S(trialnumber).COP1(i,2)=(S(trialnumber).GRM_fp1(i,1)./S(trialnumber).GRF_fp1(i,3))+center_fp1(2);
S(trialnumber).COP1(i,3)=center_fp1(3);
else
S(trialnumber).COP1(i,1)=center_fp1(1);
S(trialnumber).COP1(i,2)=center_fp1(2);
S(trialnumber).COP1(i,3)=center_fp1(3);
end
end
for i=1:length(S(trialnumber).GRF_fp2)
if S(trialnumber).GRF_fp2(i,3)>5
S(trialnumber).COP2(i,1)=
(S(trialnumber).GRM_fp2(i,2)./S(trialnumber).GRF_fp2(i,3))+center_fp2(1);
S(trialnumber).COP2(i,2)=(S(trialnumber).GRM_fp2(i,1)./S(trialnumber).GRF_fp2(i,3))+center_fp2(2);
S(trialnumber).COP2(i,3)=center_fp2(3);
else
S(trialnumber).COP2(i,1)=center_fp2(1);
S(trialnumber).COP2(i,2)=center_fp2(2);
S(trialnumber).COP2(i,3)=center_fp2(3);
end
end
% end of COP computation
end % end of force present.
clear temp
end % end of the loop to read all the trials for one subject
save([cpath,patnum,'Evaluation',evalnum,'\',condition,'\',patnum,'_reader.mat']);

```

Readtest_c3d:

```
function
[Markers,VideoFrameRate,AnalogSignals,AnalogFrameRate,Event,ParameterGroup,CameraInfo,Residual
Error]=readc3d_test(FullFileName)
% GetC3D: Getting 3D coordinate/analog data from a C3D file
%
% Input: FullFileName - file (including path) to be read
%
% Output:
% Markers      3D-marker data [Nmarkers x NvideoFrames x Ndim(=3)]
% VideoFrameRate  Frames/sec
% AnalogSignals  Analog signals [Nsignals x N analogSamples ]
% AnalogFrameRate  Samples/sec
% Event      Event(Nevents).time ..value ..name
% ParameterGroup  ParameterGroup(Ngroups).Parameters(Nparameters).data ..etc.
% CameraInfo      MarkerRelated CameraInfo [Nmarkers x NvideoFrames]
% ResidualError    MarkerRelated ErrorInfo [Nmarkers x NvideoFrames]

% AUTHOR(S) AND VERSION-HISTORY
% Ver. 1.0 Creation (Alan Morris, Toronto, October 1998) [originally named "getc3d.m"]
% Ver. 2.0 Revision (Jaap Harlaar, Amsterdam, april 2002)
Markers=[];
VideoFrameRate=0;
AnalogSignals=[];
AnalogFrameRate=0;
Event=[];
ParameterGroup=[];
CameraInfo=[];
ResidualError=[];
ind=findstr(FullFileName,'\');
if ind>0, FileName=FullFileName(ind(length(ind))+1:length(FullFileName)); else
FileName=FullFileName; end
fid=fopen(FullFileName,'r','n'); % native format (PC-intel)
if fid==-1,
    h=errorldg(['File: ',FileName,' could not be opened'],'application error');
    uiwait(h)
    return
end
NrecordFirstParameterblock=fread(fid,1,'int8'); % Reading record number of parameter section
key=fread(fid,1,'int8'); % key = 80;
if key~=80,
    h=errorldg(['File: ',FileName,' does not comply to the C3D format'],'application error');
    uiwait(h)
    fclose(fid)
    return
end
fseek(fid,512*(NrecordFirstParameterblock-1)+3,'bof'); % jump to proctype - field
proctype=fread(fid,1,'int8')-83; % proctype: 1(INTEL-PC); 2(DEC-VAX); 3(MIPS-
SUN/SGI)
if proctype==2,
    fclose(fid);
    fid=fopen(FullFileName,'r','d'); % DEC VAX D floating point and VAX ordering
end
%NrecordFirstParameterblock=fread(fid,1,'int8'); % Reading record number of parameter section
```

```

%key1=fread(fid,1,'int8');           % key = 80;
fseek(fid,2,'bof');
Nmarkers=fread(fid,1,'int16');      %number of markers
NanalogSamplesPerVideoFrame=fread(fid,1,'int16'); %number of analog channels x #analog frames
per video frame
StartFrame=fread(fid,1,'int16');    %# of first video frame
EndFrame=fread(fid,1,'int16');      %# of last video frame
MaxInterpolationGap=fread(fid,1,'int16'); %maximum interpolation gap allowed (in frame)
Scale=fread(fid,1,'float32');        %floating-point scale factor to convert 3D-integers to ref system
units
NrecordDataBlock=fread(fid,1,'int16'); %starting record number for 3D point and analog data
NanalogFramesPerVideoFrame=fread(fid,1,'int16');
if NanalogFramesPerVideoFrame > 0,
    NanalogChannels=NanalogSamplesPerVideoFrame/NanalogFramesPerVideoFrame;
else
    NanalogChannels=0;
end
VideoFrameRate=fread(fid,1,'float32');
AnalogFrameRate=VideoFrameRate*NanalogFramesPerVideoFrame;
fseek(fid,298,'bof');
EventIndicator=fread(fid,1,'int16');
if EventIndicator==12345,
    Nevents=fread(fid,1,'int16');
    fseek(fid,2,'cof'); % skip one position/2 bytes
    if Nevents>0,
        for i=1:Nevents,
            Event(i).time=fread(fid,1,'float');
        end
        fseek(fid,188*2,'bof');
        for i=1:Nevents,
            Event(i).value=fread(fid,1,'int8');
        end
        fseek(fid,198*2,'bof');
        for i=1:Nevents,
            Event(i).name=cellstr(char(fread(fid,4,'char')));
        end
    end
end
fseek(fid,512*(NrecordFirstParameterblock-1),'bof');
dat1=fread(fid,1,'int8');
key2=fread(fid,1,'int8');           % key = 80;
NparameterRecords=fread(fid,1,'int8');
proctype=fread(fid,1,'int8')-83;    % proctype: 1(INTEL-PC); 2(DEC-VAX); 3(MIPS-SUN/SGI)
Ncharacters=fread(fid,1,'int8');    % characters in group/parameter name
GroupNumber=fread(fid,1,'int8');    % id number -ve=group / +ve=parameter
PointGroupID=0;                    % initializing the POINT group id for the add-on manipulation!!!
while Ncharacters > 0 % The end of the parameter record is indicated by <0 characters for group/parameter
name
    if GroupNumber<0 % Group data
        GroupNumber=abs(GroupNumber);
        GroupName=fread(fid,[1,Ncharacters],'char');
        ParameterGroup(GroupNumber).name=cellstr(char(GroupName)); %group name
        offset=fread(fid,1,'int16'); %offset in bytes
        deschars=fread(fid,1,'int8'); %description characters
        GroupDescription=fread(fid,[1,deschars],'char');
    end
end

```

```

ParameterGroup(GroupNumber).description=cellstr(char(GroupDescription)); %group description
ParameterNumberIndex(GroupNumber)=0;
fseek(fid,offset-3-deschars,'cof');
else % parameter data
clear dimension;
ParameterNumberIndex(GroupNumber)=ParameterNumberIndex(GroupNumber)+1;
ParameterNumber=ParameterNumberIndex(GroupNumber); % index all parameters within a
group
ParameterName=fread(fid,[1,Ncharacters],'char'); % name of parameter
% read parameter name
if size(ParameterName)>0

ParameterGroup(GroupNumber).Parameter(ParameterNumber).name=cellstr(char(ParameterName));
%save parameter name
end
% read offset
offset=fread(fid,1,'int16'); %offset of parameters in bytes
filepos=ftell(fid); %present file position
nextrec=filepos+offset(1)-2; %position of beginning of next record
% read type
type=fread(fid,1,'int8'); % type of data: -1=char/1=byte/2=integer*2/4=real*4
ParameterGroup(GroupNumber).Parameter(ParameterNumber).datatype=type;
for i=1:length(ParameterGroup),
if strcmp('POINT',ParameterGroup(i).name,'exact')
PointGroupID=i;
end
end
% read number of dimensions
dimnum=fread(fid,1,'int8');
if dimnum==0
datalength=abs(type); %length of data record
else
mult=1;
for j=1:dimnum
% checking and manipulating the second dimension for POINT.LABELS
if GroupNumber==PointGroupID & j>1 &
strcmp('LABELS',ParameterGroup(GroupNumber).Parameter(ParameterNumber).name,'exact')
dimension(j)=fread(fid,1,'int8');
if dimension(j)~= PointUsedData
dimension(j)=PointUsedData;
end
% checking and manipulating the second dimension for POINT.DESCRPTIONS
elseif GroupNumber==PointGroupID & j>1 &
strcmp('DESCRIPTIONS',ParameterGroup(GroupNumber).Parameter(ParameterNumber).name,'exact')
dimension(j)=fread(fid,1,'int8');
if dimension(j)~= PointUsedData
dimension(j)=PointUsedData;
end
% no manipulation
else
dimension(j)=fread(fid,1,'int8');
end
mult=mult*dimension(j);
ParameterGroup(GroupNumber).Parameter(ParameterNumber).dim(j)=dimension(j); %save
parameter dimension data

```

```

        end
        datalength=abs(type)*mult;           %length of data record for multi-dimensional array
    end
    if type==-1 %datatype=='char'
        wordlength=dimension(1); %length of character word
        if dimnum==2 & datalength>0 %& parameter(idnumber,index,2).dim>0
            for j=1:dimension(2)
                data=fread(fid,[1,wordlength],'char'); %character word data record for 2-D array
                ParameterGroup(GroupNumber).Parameter(ParameterNumber).data(j)=cellstr(char(data));
            end
        elseif dimnum==1 & datalength>0
            data=fread(fid,[1,wordlength],'char'); %numerical data record of 1-D array
            ParameterGroup(GroupNumber).Parameter(ParameterNumber).data=cellstr(char(data));
        end
        elseif type==1 %1-byte for boolean
            Nparameters=datalength/abs(type);
            data=fread(fid,Nparameters,'int8');
            ParameterGroup(GroupNumber).Parameter(ParameterNumber).data=data;
        elseif type==2 & datalength>0 %integer
            Nparameters=datalength/abs(type);
            data=fread(fid,Nparameters,'int16');
            if dimnum>1
                ParameterGroup(GroupNumber).Parameter(ParameterNumber).data=reshape(data,dimension);
            elseif GroupNumber==PointGroupID
                if strcmp('USED',ParameterGroup(PointGroupID).Parameter(ParameterNumber).name,'exact')
                    ParameterGroup(GroupNumber).Parameter(ParameterNumber).data=data;
                    PointUsedData=data;
                end
            else
                ParameterGroup(GroupNumber).Parameter(ParameterNumber).data=data;
            end

        elseif type==4 & datalength>0
            Nparameters=datalength/abs(type);
            data=fread(fid,Nparameters,'float');
            if dimnum>1
                ParameterGroup(GroupNumber).Parameter(ParameterNumber).data=reshape(data,dimension);
            else
                ParameterGroup(GroupNumber).Parameter(ParameterNumber).data=data;
            end
        else
            % error
        end
        deschars=fread(fid,1,'int8'); %description characters
        if deschars>0
            description=fread(fid,[1,deschars],'char');

            ParameterGroup(GroupNumber).Parameter(ParameterNumber).description=cellstr(char(description));
        end
        %moving ahead to next record
        fseek(fid,nextrec,'bof');
    end
    % check group/parameter characters and idnumber to see if more records present
    Ncharacters=fread(fid,1,'int8'); % characters in next group/parameter name

```

```

    GroupNumber=fread(fid,1,'int8');          % id number -ve=group / +ve=parameter
end
% Get the coordinate and analog data
fseek(fid,(NrecordDataBlock-1)*512,'bof');
h = waitbar(0,['FileName',' is loading...']);
NvideoFrames=EndFrame - StartFrame + 1;
if Scale < 0
    for i=1:NvideoFrames
        for j=1:Nmarkers
            Markers(i,j,1:3)=fread(fid,3,'float32');
            a=fix(fread(fid,1,'float32'));
            highbyte=fix(a/256);
            lowbyte=a-highbyte*256;
            CameraInfo(i,j)=highbyte;
            ResidualError(i,j)=lowbyte*abs(Scale);
        end
        waitbar(i/NvideoFrames)
        for j=1:NanalogFramesPerVideoFrame,
            AnalogSignals(j+NanalogFramesPerVideoFrame*(i-1),1:NanalogChannels)=...
                fread(fid,NanalogChannels,'float32');
        end
    end
else
    for i=1:NvideoFrames
        for j=1:Nmarkers
            Markers(i,j,1:3)=fread(fid,3,'int16')*.Scale;
            ResidualError(i,j)=fread(fid,1,'int8');
            CameraInfo(i,j)=fread(fid,1,'int8');
        end
        waitbar(i/NvideoFrames)
        for j=1:NanalogFramesPerVideoFrame,
            AnalogSignals(j+NanalogFramesPerVideoFrame*(i-1),1:NanalogChannels)=...
                fread(fid,NanalogChannels,'int16');
        end
    end
end
close(h) % waitbar
fclose(fid);
return

```

Spinefilter

```

%11-1-10
%MS
%Filter program:
%This program is used to filter the required data for the spine program. It is based on a 4th-order low pass
%butterworth filter, with a cutofffrequency of 15Hz.
[a,b]=butter(4,.083,'low');
for xxx=2:trialnumber
    S(xxx).C7=filtfilt(a,b, S(xxx).C7);
    S(xxx).T7=filtfilt(a,b, S(xxx).T7);
    S(xxx).L3=filtfilt(a,b, S(xxx).L3);
    S(xxx).SACR=filtfilt(a,b, S(xxx).SACR);
    S(xxx).LCLA=filtfilt(a,b, S(xxx).LCLA);
    S(xxx).RCLA=filtfilt(a,b, S(xxx).RCLA);
end

```

```

S(xxx).LRIB=filtfilt(a,b, S(xxx).LRIB);
S(xxx).RRIB=filtfilt(a,b, S(xxx).RRIB);
S(xxx).LASI=filtfilt(a,b, S(xxx).LASI);
S(xxx).RASI=filtfilt(a,b, S(xxx).RASI);
end
save([cpath,patnum,'Evaluation ',evalnum,'\',condition,'\',patnum,'_reader.mat']);

```

Spinestatic

```
%8-31-09 MS
```

%This program calculates the spine angles and symmetry measures. The spine angles are calculated using the %markers placed along the spine. The proximal angle and angle of kyhopsis are calculated using the C7, T7 and L3 %markers. The distal and lordosis angles are calculated using the T7, L3, and Sacrum markers. SVA and CVA %calculate the spinal balance. The symmetry angles calculated to determine anatomical rotations and tilts.

%The static calculations for the spine program.

```
for xxx=1:length(trialnum)
```

```
%preparing for static spine calculations
```

```

S(trialnum(xxx)).C7mx=mean(S(trialnum(xxx)).C7(300:600,1));
S(trialnum(xxx)).C7my=mean(S(trialnum(xxx)).C7(300:600,2));
S(trialnum(xxx)).C7mz=mean(S(trialnum(xxx)).C7(300:600,3));
S(trialnum(xxx)).T7mx=mean(S(trialnum(xxx)).T7(300:600,1));
S(trialnum(xxx)).T7my=mean(S(trialnum(xxx)).T7(300:600,2));
S(trialnum(xxx)).T7mz=mean(S(trialnum(xxx)).T7(300:600,3));
S(trialnum(xxx)).L3mx=mean(S(trialnum(xxx)).L3(300:600,1));
S(trialnum(xxx)).L3my=mean(S(trialnum(xxx)).L3(300:600,2));
S(trialnum(xxx)).L3mz=mean(S(trialnum(xxx)).L3(300:600,3));
S(trialnum(xxx)).S1mx=mean(S(trialnum(xxx)).SACR(300:600,1));
S(trialnum(xxx)).S1my=mean(S(trialnum(xxx)).SACR(300:600,2));
S(trialnum(xxx)).S1mz=mean(S(trialnum(xxx)).SACR(300:600,3));

```

```
%Static Spine Calculations
```

```
%Calculation 1: SVA and CVA
```

```
%SVA
```

```
S(trialnum(xxx)).SVA=(S(trialnum(xxx)).C7mx-14)-(S(trialnum(xxx)).S1mx-14);
```

```
%CVA
```

```
S(trialnum(xxx)).CVA=S(trialnum(xxx)).C7my-S(trialnum(xxx)).S1my;
```

```
%Calculation 2: Thoracic kyphosis and lumbar lordosis
```

```
%kyphosis
```

```

S(trialnum(xxx)).distct=sqrt((((S(trialnum(xxx)).C7mx-
S(trialnum(xxx)).T7mx)^2)+((S(trialnum(xxx)).C7mz-S(trialnum(xxx)).T7mz)^2)));

```

```

S(trialnum(xxx)).disttl=sqrt((((S(trialnum(xxx)).T7mx-
S(trialnum(xxx)).L3mx)^2)+((S(trialnum(xxx)).T7mz-S(trialnum(xxx)).L3mz)^2)));

```

```

S(trialnum(xxx)).angk1=asin(((S(trialnum(xxx)).C7mx-
S(trialnum(xxx)).T7mx)/S(trialnum(xxx)).distct));

```

```

S(trialnum(xxx)).angk2=asin(((S(trialnum(xxx)).T7mx-
S(trialnum(xxx)).L3mx)/S(trialnum(xxx)).disttl));

```

```
S(trialnum(xxx)).kyphosis=(S(trialnum(xxx)).angk1*(180/pi)+(S(trialnum(xxx)).angk2*(180/pi));
```

```
%lordosis
```

```

S(trialnum(xxx)).distlt=sqrt((((S(trialnum(xxx)).T7mx-
S(trialnum(xxx)).L3mx)^2)+((S(trialnum(xxx)).T7mz-S(trialnum(xxx)).L3mz)^2)));

```

```

S(trialnum(xxx)).distls=sqrt((((S(trialnum(xxx)).L3mx-
S(trialnum(xxx)).S1mx)^2)+((S(trialnum(xxx)).L3mz-S(trialnum(xxx)).S1mz)^2)));

```

```
S(trialnum(xxx)).angl1=asin(((S(trialnum(xxx)).T7mx-S(trialnum(xxx)).L3mx)/S(trialnum(xxx)).distlt));
```

```

S(trialnum(xxx)).angl2=asin(((S(trialnum(xxx)).L3mx-
S(trialnum(xxx)).S1mx)/S(trialnum(xxx)).distls));

```

```

S(trialnum(xxx)).lordosis=(S(trialnum(xxx)).angl1*(180/pi)+(S(trialnum(xxx)).angl2*(180/pi));
%Calculation 3: Proximal and Distal Angles
%Proximal
S(trialnum(xxx)).distpct=sqrt(((S(trialnum(xxx)).C7mz-
S(trialnum(xxx)).T7mz)^2)+((S(trialnum(xxx)).C7my-S(trialnum(xxx)).T7my)^2));
S(trialnum(xxx)).distptl=sqrt(((S(trialnum(xxx)).T7mz-
S(trialnum(xxx)).L3mz)^2)+((S(trialnum(xxx)).T7my-S(trialnum(xxx)).L3my)^2));
S(trialnum(xxx)).angp1=asin(((S(trialnum(xxx)).C7mz-
S(trialnum(xxx)).T7mz)/S(trialnum(xxx)).distpct));
S(trialnum(xxx)).angp2=asin(((S(trialnum(xxx)).T7mz-
S(trialnum(xxx)).L3mz)/S(trialnum(xxx)).distptl));
S(trialnum(xxx)).proximal=180-
((S(trialnum(xxx)).angp1*(180/pi)+(S(trialnum(xxx)).angp2*(180/pi)));
%Distal
S(trialnum(xxx)).distdlt=sqrt(((S(trialnum(xxx)).T7mz-
S(trialnum(xxx)).L3mz)^2)+((S(trialnum(xxx)).T7my-S(trialnum(xxx)).L3my)^2));
S(trialnum(xxx)).distdls=sqrt(((S(trialnum(xxx)).L3mz-
S(trialnum(xxx)).S1mz)^2)+((S(trialnum(xxx)).L3my-S(trialnum(xxx)).S1my)^2));
S(trialnum(xxx)).angd1=asin(((S(trialnum(xxx)).T7mz-
S(trialnum(xxx)).L3mz)/S(trialnum(xxx)).distdlt));
S(trialnum(xxx)).angd2=asin(((S(trialnum(xxx)).L3mz-
S(trialnum(xxx)).S1mz)/S(trialnum(xxx)).distdls));
S(trialnum(xxx)).distal=180-((S(trialnum(xxx)).angd1*(180/pi)+(S(trialnum(xxx)).angd2*(180/pi)));
%end of static spine calculations
% Calculations for COP Measures: All measures are based from the right side
%Shoulder Calculations
S(trialnum(xxx)).sdist=sqrt(((S(trialnum(xxx)).RCLA(300:600,2)-
S(trialnum(xxx)).LCLA(300:600,2)).^2) + ((S(trialnum(xxx)).RCLA(300:600,3)-
S(trialnum(xxx)).LCLA(300:600,3)).^2));
S(trialnum(xxx)).angs=asin((S(trialnum(xxx)).RCLA(300:600,3)-
S(trialnum(xxx)).LCLA(300:600,3))/S(trialnum(xxx)).sdist);
S(trialnum(xxx)).corrections=S(trialnum(xxx)).RCLA(300:600,3)-(tan(S(trialnum(xxx)).angs).*(-
S(trialnum(xxx)).RCLA(300:600,2)+S(trialnum(xxx)).COP2(300:600,2)));
S(trialnum(xxx)).copdists=sqrt(((S(trialnum(xxx)).RCLA(300:600,2)-
S(trialnum(xxx)).COP2(300:600,2)).^2) + ((S(trialnum(xxx)).RCLA(300:600,3)-
S(trialnum(xxx)).corrections).^2));
S(trialnum(xxx)).rsratio=(S(trialnum(xxx)).copdists./S(trialnum(xxx)).sdist).*100;
%Mid-Thorax Calculations
S(trialnum(xxx)).tdist=sqrt(((S(trialnum(xxx)).RRIB(300:600,2)-S(trialnum(xxx)).LRIB(300:600,2)).^2)
+ ((S(trialnum(xxx)).RRIB(300:600,3)-S(trialnum(xxx)).LRIB(300:600,3)).^2));
S(trialnum(xxx)).angt=asin((S(trialnum(xxx)).RRIB(300:600,3)-
S(trialnum(xxx)).LRIB(300:600,3))/S(trialnum(xxx)).tdist);
S(trialnum(xxx)).correctiont=S(trialnum(xxx)).RRIB(300:600,3)-(tan(S(trialnum(xxx)).angt).*(-
S(trialnum(xxx)).RRIB(300:600,2)+S(trialnum(xxx)).COP2(300:600,2)));
S(trialnum(xxx)).copdistt=sqrt(((S(trialnum(xxx)).RRIB(300:600,2)-
S(trialnum(xxx)).COP2(300:600,2)).^2) + ((S(trialnum(xxx)).RRIB(300:600,3)-
S(trialnum(xxx)).correctiont).^2));
S(trialnum(xxx)).rtratio=(S(trialnum(xxx)).copdistt./S(trialnum(xxx)).tdist).*100;
%Pelvis Calculations
S(trialnum(xxx)).pdist=sqrt(((S(trialnum(xxx)).RASI(300:600,2)-S(trialnum(xxx)).LASI(300:600,2)).^2)
+ ((S(trialnum(xxx)).RASI(300:600,3)-S(trialnum(xxx)).LASI(300:600,3)).^2));
S(trialnum(xxx)).angp=asin((S(trialnum(xxx)).RASI(300:600,3)-
S(trialnum(xxx)).LASI(300:600,3))/S(trialnum(xxx)).pdist);
S(trialnum(xxx)).correctionp=S(trialnum(xxx)).RASI(300:600,3)-(tan(S(trialnum(xxx)).angp).*(-
S(trialnum(xxx)).RASI(300:600,2)+S(trialnum(xxx)).COP2(300:600,2)));

```



```

S(trialnum(xxx)).copdistp=sqrt(((S(trialnum(xxx)).RASI(300:600,2)-
S(trialnum(xxx)).COP2(300:600,2)).^2) + ((S(trialnum(xxx)).RASI(300:600,3)-
S(trialnum(xxx)).correctionp).^2));
S(trialnum(xxx)).rpratio=(S(trialnum(xxx)).copdistp./S(trialnum(xxx)).pdist). * 100;
%end of COP calculations

% Calculations for Symmetry Measures
%Frontal Symmetry: With respect to Global Coordinates done for the
%Shoulders, Mid-Thorax, and Pelvis.
S(trialnum(xxx)).Shouldercor=S(trialnum(xxx)).angs.*(180/pi);
S(trialnum(xxx)).Thoraxcor=S(trialnum(xxx)).angt.*(180/pi);
S(trialnum(xxx)).Pelviscor=S(trialnum(xxx)).angp.*(180/pi);
%Transverse Symmetry: With respect to Global Coordinates- done for the
%Shoulders, Mid-Thorax, and Pelvis
S(trialnum(xxx)).distts=sqrt(((S(trialnum(xxx)).RCLA(300:600,1)-
S(trialnum(xxx)).LCLA(300:600,1)).^2) + ((S(trialnum(xxx)).RCLA(300:600,2)-
S(trialnum(xxx)).LCLA(300:600,2)).^2));
S(trialnum(xxx)).angts=asin((S(trialnum(xxx)).RCLA(300:600,1)-
S(trialnum(xxx)).LCLA(300:600,1))./S(trialnum(xxx)).distts);
S(trialnum(xxx)).Shoulderrot=S(trialnum(xxx)).angts.*(180/pi);
S(trialnum(xxx)).disttt=sqrt(((S(trialnum(xxx)).RRIB(300:600,1)-
S(trialnum(xxx)).LRIB(300:600,1)).^2) + ((S(trialnum(xxx)).RRIB(300:600,2)-
S(trialnum(xxx)).LRIB(300:600,2)).^2));
S(trialnum(xxx)).angtt=asin((S(trialnum(xxx)).RRIB(300:600,1)-
S(trialnum(xxx)).LRIB(300:600,1))./S(trialnum(xxx)).disttt);
S(trialnum(xxx)).Thoraxrot=S(trialnum(xxx)).angtt.*(180/pi);
S(trialnum(xxx)).disttp=sqrt(((S(trialnum(xxx)).RASI(300:600,1)-
S(trialnum(xxx)).LASI(300:600,1)).^2) + ((S(trialnum(xxx)).RASI(300:600,2)-
S(trialnum(xxx)).LASI(300:600,2)).^2));
S(trialnum(xxx)).angtp=asin((S(trialnum(xxx)).RASI(300:600,1)-
S(trialnum(xxx)).LASI(300:600,1))./S(trialnum(xxx)).disttp);
S(trialnum(xxx)).Pelvisrot=S(trialnum(xxx)).angtp.*(180/pi);
%end of symmetry calculations
end
save([cpath,patnum,'\Evaluation ',evalnum,'\',patnum,'_staticspine.mat']);

```

Output

```

%10-30-09 ~MS
%Excel output program for the spine program
%load([cpath,patnum,'\Evaluation ',evalnum,'\',patnum,'_staticspine.mat']);
close all;
vind=0;
%Output COP measures
varlist{vind+1,1}='RELAXED STANDING';
varlist{vind+1,1}='COP ratios'; vind=vind+1;
varlist{vind+1,1}='Anatomic Level'; varlist{vind+1,3}='Trial 1'; varlist{vind+1,4}='Trial 2';
varlist{vind+1,5}='Trial 3'; vind=vind+1;
varlist{vind+1,1}=' R.Shoulder'; vind=vind+1;
varlist{vind+1,2}='Mean'; varlist{vind+1,3}=num2str(mean(S(1,2).rsratio));
varlist{vind+1,4}=num2str(mean(S(1,3).rsratio)); varlist{vind+1,5}=num2str(mean(S(1,4).rsratio));
vind=vind+1;
varlist{vind+1,2}='Max'; varlist{vind+1,3}=num2str(max(S(1,2).rsratio));
varlist{vind+1,4}=num2str(max(S(1,3).rsratio)); varlist{vind+1,5}=num2str(max(S(1,4).rsratio));
vind=vind+1;

```

```

varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str(min(S(1,2).rsratio));
varlist{vind+1,4}=num2str(min(S(1,3).rsratio));  varlist{vind+1,5}=num2str(min(S(1,4).rsratio));
vind=vind+1;
varlist{vind+1,1}=' R.Thorax';  vind=vind+1;
varlist{vind+1,2}='Mean';      varlist{vind+1,3}=num2str(mean(S(1,2).rratio));
varlist{vind+1,4}=num2str(mean(S(1,3).rratio));  varlist{vind+1,5}=num2str(mean(S(1,4).rratio));
vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str(max(S(1,2).rratio));
varlist{vind+1,4}=num2str(max(S(1,3).rratio));  varlist{vind+1,5}=num2str(max(S(1,4).rratio));
vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str(min(S(1,2).rratio));
varlist{vind+1,4}=num2str(min(S(1,3).rratio));  varlist{vind+1,5}=num2str(min(S(1,4).rratio));
vind=vind+1;
varlist{vind+1,1}=' R.Pelvis';  vind=vind+1;
varlist{vind+1,2}='Mean';      varlist{vind+1,3}=num2str(mean(S(1,2).rpratio));
varlist{vind+1,4}=num2str(mean(S(1,3).rpratio));  varlist{vind+1,5}=num2str(mean(S(1,4).rpratio));
vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str(max(S(1,2).rpratio));
varlist{vind+1,4}=num2str(max(S(1,3).rpratio));  varlist{vind+1,5}=num2str(max(S(1,4).rpratio));
vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str(min(S(1,2).rpratio));
varlist{vind+1,4}=num2str(min(S(1,3).rpratio));  varlist{vind+1,5}=num2str(min(S(1,4).rpratio));
vind=vind+1;
vind=vind+1;
%Output Frontal Plane Angles
varlist{vind+1,1}='RELAXED STANDING AND FRAME';
varlist{vind+1,1}='Frontal Plane Angles'; vind=vind+1;
varlist{vind+1,1}='Anatomic Level'; varlist{vind+1,3}='Trial 1 Standing';
varlist{vind+1,4}='Trial 2 Standing';      varlist{vind+1,5}='Trial 3 Standing';
varlist{vind+1,6}='Trial 1 Frame';      varlist{vind+1,7}='Trial 2 Frame';
varlist{vind+1,8}='Trial 3 Frame'; vind=vind+1;
varlist{vind+1,1}=' F.Shoulder';  vind=vind+1;
varlist{vind+1,2}='Mean';      varlist{vind+1,3}=num2str(mean(S(1,2).Shouldercor));
varlist{vind+1,4}=num2str(mean(S(1,3).Shouldercor));
varlist{vind+1,5}=num2str(mean(S(1,4).Shouldercor));
varlist{vind+1,6}=num2str(mean(S(1,5).Shouldercor));
varlist{vind+1,7}=num2str(mean(S(1,6).Shouldercor));
varlist{vind+1,8}=num2str(mean(S(1,7).Shouldercor)); vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str(max(S(1,2).Shouldercor));
varlist{vind+1,4}=num2str(max(S(1,3).Shouldercor));
varlist{vind+1,5}=num2str(max(S(1,4).Shouldercor));
varlist{vind+1,6}=num2str(max(S(1,5).Shouldercor));
varlist{vind+1,7}=num2str(max(S(1,6).Shouldercor));
varlist{vind+1,8}=num2str(max(S(1,7).Shouldercor)); vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str(min(S(1,2).Shouldercor));
varlist{vind+1,4}=num2str(min(S(1,3).Shouldercor));
varlist{vind+1,5}=num2str(min(S(1,4).Shouldercor));
varlist{vind+1,6}=num2str(min(S(1,5).Shouldercor));
varlist{vind+1,7}=num2str(min(S(1,6).Shouldercor));
varlist{vind+1,8}=num2str(min(S(1,7).Shouldercor)); vind=vind+1;
varlist{vind+1,1}=' F.Thorax';  vind=vind+1;
varlist{vind+1,2}='Mean';      varlist{vind+1,3}=num2str(mean(S(1,2).Thoraxcor));
varlist{vind+1,4}=num2str(mean(S(1,3).Thoraxcor));
varlist{vind+1,5}=num2str(mean(S(1,4).Thoraxcor));
varlist{vind+1,6}=num2str(mean(S(1,5).Thoraxcor));

```

```

varlist{vind+1,7}=num2str(mean(S(1,6).Thoraxcor));
varlist{vind+1,8}=num2str(mean(S(1,7).Thoraxcor)); vind=vind+1;
varlist{vind+1,2}='Max'; varlist{vind+1,3}=num2str(max(S(1,2).Thoraxcor));
varlist{vind+1,4}=num2str(max(S(1,3).Thoraxcor));
varlist{vind+1,5}=num2str(max(S(1,4).Thoraxcor));
varlist{vind+1,6}=num2str(max(S(1,5).Thoraxcor));
varlist{vind+1,7}=num2str(max(S(1,6).Thoraxcor));
varlist{vind+1,8}=num2str(max(S(1,7).Thoraxcor)); vind=vind+1;
varlist{vind+1,2}='Min'; varlist{vind+1,3}=num2str(min(S(1,2).Thoraxcor));
varlist{vind+1,4}=num2str(min(S(1,3).Thoraxcor));
varlist{vind+1,5}=num2str(min(S(1,4).Thoraxcor));
varlist{vind+1,6}=num2str(min(S(1,5).Thoraxcor)); varlist{vind+1,7}=num2str(min(S(1,6).Thoraxcor));
varlist{vind+1,8}=num2str(min(S(1,7).Thoraxcor)); vind=vind+1;
varlist{vind+1,1}=' F.Pelvis'; vind=vind+1;
varlist{vind+1,2}='Mean'; varlist{vind+1,3}=num2str(mean(S(1,2).Pelviscor));
varlist{vind+1,4}=num2str(mean(S(1,3).Pelviscor));
varlist{vind+1,5}=num2str(mean(S(1,4).Pelviscor));
varlist{vind+1,6}=num2str(mean(S(1,5).Pelviscor));
varlist{vind+1,7}=num2str(mean(S(1,6).Pelviscor));
varlist{vind+1,8}=num2str(mean(S(1,7).Pelviscor)); vind=vind+1;
varlist{vind+1,2}='Max'; varlist{vind+1,3}=num2str(max(S(1,2).Pelviscor));
varlist{vind+1,4}=num2str(max(S(1,3).Pelviscor));
varlist{vind+1,5}=num2str(max(S(1,4).Pelviscor)); varlist{vind+1,6}=num2str(max(S(1,5).Pelviscor));
varlist{vind+1,7}=num2str(max(S(1,6).Pelviscor)); varlist{vind+1,8}=num2str(max(S(1,7).Pelviscor));
vind=vind+1;
varlist{vind+1,2}='Min'; varlist{vind+1,3}=num2str(min(S(1,2).Pelviscor));
varlist{vind+1,4}=num2str(min(S(1,3).Pelviscor));
varlist{vind+1,5}=num2str(min(S(1,4).Pelviscor)); varlist{vind+1,6}=num2str(min(S(1,5).Pelviscor));
varlist{vind+1,7}=num2str(min(S(1,6).Pelviscor)); varlist{vind+1,8}=num2str(min(S(1,7).Pelviscor));
vind=vind+1;
vind=vind+1;
% Output Transverse Plane Angles
varlist{vind+1,1}='RELAXED STANDING AND FRAME';
varlist{vind+1,1}='Transverse Plane Angles'; vind=vind+1;
varlist{vind+1,1}='Anatomic Level'; varlist{vind+1,3}='Trial 1 Standing';
varlist{vind+1,4}='Trial 2 Standing'; varlist{vind+1,5}='Trial 3 Standing';
varlist{vind+1,6}='Trial 1 Frame'; varlist{vind+1,7}='Trial 2 Frame';
varlist{vind+1,8}='Trial 3 Frame'; vind=vind+1;
varlist{vind+1,1}=' T.Shoulder'; vind=vind+1;
varlist{vind+1,2}='Mean'; varlist{vind+1,3}=num2str(mean(S(1,2).Shoulderrot));
varlist{vind+1,4}=num2str(mean(S(1,3).Shoulderrot));
varlist{vind+1,5}=num2str(mean(S(1,4).Shoulderrot));
varlist{vind+1,6}=num2str(mean(S(1,5).Shoulderrot));
varlist{vind+1,7}=num2str(mean(S(1,6).Shoulderrot));
varlist{vind+1,8}=num2str(mean(S(1,7).Shoulderrot)); vind=vind+1;
varlist{vind+1,2}='Max'; varlist{vind+1,3}=num2str(max(S(1,2).Shoulderrot));
varlist{vind+1,4}=num2str(max(S(1,3).Shoulderrot));
varlist{vind+1,5}=num2str(max(S(1,4).Shoulderrot));
varlist{vind+1,6}=num2str(max(S(1,5).Shoulderrot));
varlist{vind+1,7}=num2str(max(S(1,6).Shoulderrot));
varlist{vind+1,8}=num2str(max(S(1,7).Shoulderrot)); vind=vind+1;
varlist{vind+1,2}='Min'; varlist{vind+1,3}=num2str(min(S(1,2).Shoulderrot));
varlist{vind+1,4}=num2str(min(S(1,3).Shoulderrot));
varlist{vind+1,5}=num2str(min(S(1,4).Shoulderrot));
varlist{vind+1,6}=num2str(min(S(1,5).Shoulderrot));

```

```

varlist{vind+1,7}=num2str(min(S(1,6).Shoulderrot));
varlist{vind+1,8}=num2str(min(S(1,7).Shoulderrot)); vind=vind+1;
varlist{vind+1,1}=' T.Thorax'; vind=vind+1;
varlist{vind+1,2}='Mean'; varlist{vind+1,3}=num2str(mean(S(1,2).Thoraxrot));
varlist{vind+1,4}=num2str(mean(S(1,3).Thoraxrot));
varlist{vind+1,5}=num2str(mean(S(1,4).Thoraxrot));
varlist{vind+1,6}=num2str(mean(S(1,5).Thoraxrot));
varlist{vind+1,7}=num2str(mean(S(1,6).Thoraxrot));
varlist{vind+1,8}=num2str(mean(S(1,7).Thoraxrot)); vind=vind+1;
varlist{vind+1,2}='Max'; varlist{vind+1,3}=num2str(max(S(1,2).Thoraxrot));
varlist{vind+1,4}=num2str(max(S(1,3).Thoraxrot));
varlist{vind+1,5}=num2str(max(S(1,4).Thoraxrot));
varlist{vind+1,6}=num2str(max(S(1,5).Thoraxrot)); varlist{vind+1,7}=num2str(max(S(1,6).Thoraxrot));
varlist{vind+1,8}=num2str(max(S(1,7).Thoraxrot)); vind=vind+1;
varlist{vind+1,2}='Min'; varlist{vind+1,3}=num2str(min(S(1,2).Thoraxrot));
varlist{vind+1,4}=num2str(min(S(1,3).Thoraxrot));
varlist{vind+1,5}=num2str(min(S(1,4).Thoraxrot));
varlist{vind+1,6}=num2str(min(S(1,5).Thoraxrot)); varlist{vind+1,7}=num2str(min(S(1,6).Thoraxrot));
varlist{vind+1,8}=num2str(min(S(1,7).Thoraxrot)); vind=vind+1;
varlist{vind+1,1}=' T.Pelvis'; vind=vind+1;
varlist{vind+1,2}='Mean'; varlist{vind+1,3}=num2str(mean(S(1,2).Pelvisrot));
varlist{vind+1,4}=num2str(mean(S(1,3).Pelvisrot));
varlist{vind+1,5}=num2str(mean(S(1,4).Pelvisrot));
varlist{vind+1,6}=num2str(mean(S(1,5).Pelvisrot)); varlist{vind+1,7}=num2str(mean(S(1,6).Pelvisrot));
varlist{vind+1,8}=num2str(mean(S(1,7).Pelvisrot)); vind=vind+1;
varlist{vind+1,2}='Max'; varlist{vind+1,3}=num2str(max(S(1,2).Pelvisrot));
varlist{vind+1,4}=num2str(max(S(1,3).Pelvisrot));
varlist{vind+1,5}=num2str(max(S(1,4).Pelvisrot)); varlist{vind+1,6}=num2str(max(S(1,5).Pelvisrot));
varlist{vind+1,7}=num2str(max(S(1,6).Pelvisrot));
varlist{vind+1,8}=num2str(max(S(1,7).Pelvisrot)); vind=vind+1;
varlist{vind+1,2}='Min'; varlist{vind+1,3}=num2str(min(S(1,2).Pelvisrot));
varlist{vind+1,4}=num2str(min(S(1,3).Pelvisrot)); varlist{vind+1,5}=num2str(min(S(1,4).Pelvisrot));
varlist{vind+1,6}=num2str(min(S(1,5).Pelvisrot)); varlist{vind+1,7}=num2str(min(S(1,6).Pelvisrot));
varlist{vind+1,8}=num2str(min(S(1,7).Pelvisrot)); vind=vind+1;
vind=vind+1;
%Output Static Spine Measures
varlist{vind+1,1}='RELAXED STANDING AND FRAME';
varlist{vind+1,1}='Static spine measures'; vind=vind+1;
varlist{vind+1,1}='Measure'; varlist{vind+1,2}='Trial 1 Standing'; varlist{vind+1,3}='Trial 2
Standing'; varlist{vind+1,4}='Trial 3 Standing'; varlist{vind+1,5}='Trial 1 Frame';
varlist{vind+1,6}='Trial 2 Frame'; varlist{vind+1,7}='Trial 3 Frame'; vind=vind+1;
varlist{vind+1,1}='SVA'; varlist{vind+1,2}=num2str(S(1,2).SVA);
varlist{vind+1,3}=num2str(S(1,3).SVA); varlist{vind+1,4}=num2str(S(1,4).SVA);
varlist{vind+1,5}=num2str(S(1,5).SVA); varlist{vind+1,6}=num2str(S(1,6).SVA);
varlist{vind+1,7}=num2str(S(1,7).SVA); vind=vind+1;
varlist{vind+1,1}='CVA'; varlist{vind+1,2}=num2str(S(1,2).CVA);
varlist{vind+1,3}=num2str(S(1,3).CVA); varlist{vind+1,4}=num2str(S(1,4).CVA);
varlist{vind+1,5}=num2str(S(1,5).CVA); varlist{vind+1,6}=num2str(S(1,6).CVA);
varlist{vind+1,7}=num2str(S(1,7).CVA); vind=vind+1;
varlist{vind+1,1}='Kyphosis'; varlist{vind+1,2}=num2str(S(1,2).kyphosis);
varlist{vind+1,3}=num2str(S(1,3).kyphosis); varlist{vind+1,4}=num2str(S(1,4).kyphosis);
varlist{vind+1,5}=num2str(S(1,5).kyphosis); varlist{vind+1,6}=num2str(S(1,6).kyphosis);
varlist{vind+1,7}=num2str(S(1,7).kyphosis); vind=vind+1;
varlist{vind+1,1}='Lordosis'; varlist{vind+1,2}=num2str(S(1,2).lordosis);
varlist{vind+1,3}=num2str(S(1,3).lordosis); varlist{vind+1,4}=num2str(S(1,4).lordosis);

```

```

varlist{vind+1,5}=num2str(S(1,5).lordosis); varlist{vind+1,6}=num2str(S(1,6).lordosis);
varlist{vind+1,7}=num2str(S(1,7).lordosis); vind=vind+1;
varlist{vind+1,1}='Proximal Angle'; varlist{vind+1,2}=num2str(S(1,2).proximal);
varlist{vind+1,3}=num2str(S(1,3).proximal); varlist{vind+1,4}=num2str(S(1,4).proximal);
varlist{vind+1,5}=num2str(S(1,5).proximal); varlist{vind+1,6}=num2str(S(1,6).proximal);
varlist{vind+1,7}=num2str(S(1,7).proximal); vind=vind+1;
varlist{vind+1,1}='Distal Angle'; varlist{vind+1,2}=num2str(S(1,2).distal);
varlist{vind+1,3}=num2str(S(1,3).distal); varlist{vind+1,4}=num2str(S(1,4).distal);
varlist{vind+1,5}=num2str(S(1,5).distal); varlist{vind+1,6}=num2str(S(1,6).distal);
varlist{vind+1,7}=num2str(S(1,7).distal); vind=vind+1;
warning off MATLAB:xlswrite:AddSheet;
xlswrite([cpath,num2str(patnum)],[RelaxedStatic.xlsx'],varlist,[num2str(patnum)],'A3');

```

Spinecalc

```

%Spine calculations
%9-10-10
%MS
%This program is set up to define the planes required for the scoliosis project. There are 3 planes of interest,
the %pelvic plane, upper trunk plane, and mid-spine plane. From these values all of the spine angles will be
calculated.
for xxx=1:length(trialn)
%Defining the Pelvic Plane:
%This plane is defined using the SACR, RASI, and LASI markers based on
%the definitions developed by Roy Davis.
%First define the center point of the vector between RASI and LASI
S(trialn(xxx)).FPelvis=(S(trialn(xxx)).RASI+S(trialn(xxx)).LASI)/2;
S(trialn(xxx)).CPelvis=(S(trialn(xxx)).FPelvis+S(trialn(xxx)).SACR)/2;
%Now define the axis
for i=1:size(S(trialn(xxx)).RASI,1)
if norm(S(trialn(xxx)).LASI(i,:)-S(trialn(xxx)).RASI(i,:))~=0
S(trialn(xxx)).yPel(i,:)=(S(trialn(xxx)).LASI(i,:)-
S(trialn(xxx)).RASI(i,:))/norm(S(trialn(xxx)).LASI(i,:)-S(trialn(xxx)).RASI(i,:));
else
S(trialn(xxx)).yPel(i,:)=[0 0 0];
end
%create a temporary axis to create a new vector using the SARC and center point of the front vector, in
order to %create the Z axis
temp(i,:)=S(trialn(xxx)).FPelvis(i,:)-S(trialn(xxx)).SACR(i,:);
if norm(cross(temp(i,:),S(trialn(xxx)).yPel(i,:)))~=0
S(trialn(xxx)).zPel(i,:)=cross(temp(i,:),S(trialn(xxx)).yPel(i,:)) ./
norm(cross(temp(i,:),S(trialn(xxx)).yPel(i,:)));
else
S(trialn(xxx)).zPel(i,:)=[0 0 0];
end
%Finally create the y axis

if norm(cross(S(trialn(xxx)).yPel(i,:),S(trialn(xxx)).zPel(i,:)))~=0
S(trialn(xxx)).xPel(i,:)=cross(S(trialn(xxx)).yPel(i,:),S(trialn(xxx)).zPel(i,:)) ./
norm(cross(S(trialn(xxx)).yPel(i,:),S(trialn(xxx)).zPel(i,:)));
else
S(trialn(xxx)).xPel(i,:)=[0 0 0];
end
end
end

```

```

%Defining the mid-thorax plane This is done in an identical fashion to the pelvis
S(trialn(xxx)).FMThor=(S(trialn(xxx)).RRIB+S(trialn(xxx)).LRIB)./2;
S(trialn(xxx)).CMThor=(S(trialn(xxx)).FMThor+S(trialn(xxx)).T7)./2;
for i=1:size(S(trialn(xxx)).RRIB,1)
    if sqrt(sum(S(trialn(xxx)).CPelvis(i,:)-(S(trialn(xxx)).CMThor(i,:)).^2))~=0
        S(trialn(xxx)).zMTRX(i,:)=(S(trialn(xxx)).CMThor(i,:)-
S(trialn(xxx)).CPelvis(i,:))./norm(S(trialn(xxx)).CMThor(i,:)-S(trialn(xxx)).CPelvis(i,:));
    else
        S(trialn(xxx)).zMTRX(i,:)=[0 0 0];
    end
    temp(i,:)=S(trialn(xxx)).LRIB(i,:)-S(trialn(xxx)).RRIB(i,:);
    if norm(cross(temp(i,:),S(trialn(xxx)).zMTRX(i,:)))~=0
        S(trialn(xxx)).xMTRX(i,:)=cross(temp(i,:),S(trialn(xxx)).zMTRX(i,:)) ./
norm(cross(temp(i,:),S(trialn(xxx)).zMTRX(i,:)));
    else
        S(trialn(xxx)).xMTRX(i,:)=[0 0 0];
    end
    if norm(cross(S(trialn(xxx)).zMTRX(i,:),S(trialn(xxx)).xMTRX(i,:)))~=0
        S(trialn(xxx)).yMTRX(i,:)=cross(S(trialn(xxx)).zMTRX(i,:),S(trialn(xxx)).xMTRX(i,:)) ./
norm(cross(S(trialn(xxx)).zMTRX(i,:),S(trialn(xxx)).xMTRX(i,:)));
    else
        S(trialn(xxx)).yMTRX(i,:)=[0 0 0];
    end
end
%Defining the upper thorax plane This is again done like the previous two planes
S(trialn(xxx)).FUThor=(S(trialn(xxx)).RCLA+S(trialn(xxx)).LCLA)./2;
S(trialn(xxx)).CUThor=(S(trialn(xxx)).FUThor+S(trialn(xxx)).C7)./2;
for i=1:size(S(trialn(xxx)).RCLA,1)
    if sqrt(sum(S(trialn(xxx)).CPelvis(i,:)-(S(trialn(xxx)).CUThor(i,:)).^2))~=0
        S(trialn(xxx)).zUTRX(i,:)=(S(trialn(xxx)).CUThor(i,:)-
S(trialn(xxx)).CPelvis(i,:))./norm(S(trialn(xxx)).CUThor(i,:)-S(trialn(xxx)).CPelvis(i,:));
    else
        S(trialn(xxx)).zUTRX(i,:)=[0 0 0];
    end
    temp(i,:)=S(trialn(xxx)).LCLA(i,:)-S(trialn(xxx)).RCLA(i,:);
    if norm(cross(temp(i,:),S(trialn(xxx)).zUTRX(i,:)))~=0
        S(trialn(xxx)).xUTRX(i,:)=cross(temp(i,:),S(trialn(xxx)).zUTRX(i,:)) ./
norm(cross(temp(i,:),S(trialn(xxx)).zUTRX(i,:)));
    else
        S(trialn(xxx)).xUTRX(i,:)=[0 0 0];
    end
    if norm(cross(S(trialn(xxx)).zUTRX(i,:),S(trialn(xxx)).xUTRX(i,:)))~=0
        S(trialn(xxx)).yUTRX(i,:)=cross(S(trialn(xxx)).zUTRX(i,:),S(trialn(xxx)).xUTRX(i,:)) ./
norm(cross(S(trialn(xxx)).zUTRX(i,:),S(trialn(xxx)).xUTRX(i,:)));
    else
        S(trialn(xxx)).yUTRX(i,:)=[0 0 0];
    end
end
end
save([cpath,patnum,'\Evaluation ',evalnum,'\',patnum,'_axis.mat']);
%Based on the fact that multiple angles can satisfy the solutions for the sagittal and transverse angle since,
the %coronal angle is calculated using a sine componet only, the atan2 function is used to ensure proper
signs.
% Defining the lab coordinates
for i=1:length(S(trialn(xxx)).zPel)
    S(trialn(xxx)).xLab(i,1:3) = [1 0 0];
end

```

```

S(trialn(xxx)).yLab(i,1:3) = [0 1 0];
S(trialn(xxx)).zLab(i,1:3) = [0 0 1];
end
% (Coronal=1, Sagittal=2, Transverse=3)
S(trialn(xxx)).PelvisAngle(:,1)=(asin(dot(S(trialn(xxx)).zPel',S(trialn(xxx)).yLab')))*(180/pi);
S(trialn(xxx)).PelvisAngle(:,2)=(atan2(dot(S(trialn(xxx)).xLab',S(trialn(xxx)).zPel'),dot(S(trialn(xxx)).zLab',S(trialn(xxx)).zPel')))*(180/pi);
S(trialn(xxx)).PelvisAngle(:,3)=(atan2(dot(S(trialn(xxx)).xLab',S(trialn(xxx)).yPel'),dot(S(trialn(xxx)).yLab',S(trialn(xxx)).yPel')))*(180/pi);
S(trialn(xxx)).MThorAngle(:,2)=(atan2(dot(S(trialn(xxx)).zMTRX',S(trialn(xxx)).xLab'),dot(S(trialn(xxx)).zMTRX',S(trialn(xxx)).zLab')))*(180/pi);
S(trialn(xxx)).MThorAngle(:,3)=(atan2(dot(S(trialn(xxx)).xMTRX',S(trialn(xxx)).yLab'),dot(S(trialn(xxx)).yMTRX',S(trialn(xxx)).yLab')))*(180/pi);
S(trialn(xxx)).UThorAngle(:,2)=(atan2(dot(S(trialn(xxx)).zUTRX',S(trialn(xxx)).xLab'),dot(S(trialn(xxx)).zUTRX',S(trialn(xxx)).zLab')))*(180/pi);
S(trialn(xxx)).UThorAngle(:,3)=(atan2(dot(S(trialn(xxx)).xUTRX',S(trialn(xxx)).yLab'),dot(S(trialn(xxx)).yUTRX',S(trialn(xxx)).yLab')))*(180/pi);
%Removes the 180 degree offsets produced by quadrant inversions.
for i=1:length(S(trialn(xxx)).PelvisAngle(:,1))
    if S(trialn(xxx)).PelvisAngle(i,1)<-120
        S(trialn(xxx)).PelvisAngle(i,1)=S(trialn(xxx)).PelvisAngle(i,1)+180;
    elseif S(trialn(xxx)).PelvisAngle(i,1)>120
        S(trialn(xxx)).PelvisAngle(i,1)=S(trialn(xxx)).PelvisAngle(i,1)-180;
    end
    if S(trialn(xxx)).PelvisAngle(i,2)<-120
        S(trialn(xxx)).PelvisAngle(i,2)=S(trialn(xxx)).PelvisAngle(i,2)+180;
    elseif S(trialn(xxx)).PelvisAngle(i,2)>120
        S(trialn(xxx)).PelvisAngle(i,2)=S(trialn(xxx)).PelvisAngle(i,2)-180;
    end
    if S(trialn(xxx)).PelvisAngle(i,3)<-120
        S(trialn(xxx)).PelvisAngle(i,3)=S(trialn(xxx)).PelvisAngle(i,3)+180;
    elseif S(trialn(xxx)).PelvisAngle(i,3)>120
        S(trialn(xxx)).PelvisAngle(i,3)=S(trialn(xxx)).PelvisAngle(i,3)-180;
    end
    if S(trialn(xxx)).MThorAngle(i,1)<-120
        S(trialn(xxx)).MThorAngle(i,1)=S(trialn(xxx)).MThorAngle(i,1)+180;
    elseif S(trialn(xxx)).MThorAngle(i,1)>120
        S(trialn(xxx)).MThorAngle(i,1)=S(trialn(xxx)).MThorAngle(i,1)-180;
    end
    if S(trialn(xxx)).MThorAngle(i,2)<-120
        S(trialn(xxx)).MThorAngle(i,2)=S(trialn(xxx)).MThorAngle(i,2)+180;
    elseif S(trialn(xxx)).MThorAngle(i,2)>120
        S(trialn(xxx)).MThorAngle(i,2)=S(trialn(xxx)).MThorAngle(i,2)-180;
    end
    if S(trialn(xxx)).MThorAngle(i,3)<-120
        S(trialn(xxx)).MThorAngle(i,3)=S(trialn(xxx)).MThorAngle(i,3)+180;
    elseif S(trialn(xxx)).MThorAngle(i,3)>120
        S(trialn(xxx)).MThorAngle(i,3)=S(trialn(xxx)).MThorAngle(i,3)-180;
    end
    if S(trialn(xxx)).UThorAngle(i,1)<-120
        S(trialn(xxx)).UThorAngle(i,1)=S(trialn(xxx)).UThorAngle(i,1)+180;
    elseif S(trialn(xxx)).UThorAngle(i,1)>130
        S(trialn(xxx)).UThorAngle(i,1)=S(trialn(xxx)).UThorAngle(i,1)-180;
    end
end

```

```

if S(trialn(xxx)).UThorAngle(i,2)<=-120
    S(trialn(xxx)).UThorAngle(i,2)=S(trialn(xxx)).UThorAngle(i,2)+180;
elseif S(trialn(xxx)).UThorAngle(i,2)>130
    S(trialn(xxx)).UThorAngle(i,2)=S(trialn(xxx)).UThorAngle(i,2)-180;
end
if S(trialn(xxx)).UThorAngle(i,3)<=-120
    S(trialn(xxx)).UThorAngle(i,3)=S(trialn(xxx)).UThorAngle(i,3)+180;
elseif S(trialn(xxx)).UThorAngle(i,3)>130
    S(trialn(xxx)).UThorAngle(i,3)=S(trialn(xxx)).UThorAngle(i,3)-180;
end
end
end
% Segment Angle calculations. Segment angles are defined as the relative motion between two planes.
%Total Spine (Coronal=1, Sagittal=2, Transverse=3)
S(trialn(xxx)).SpineAngle(:,1)=S(trialn(xxx)).UThorAngle(:,1)-S(trialn(xxx)).PelvisAngle(:,1);
S(trialn(xxx)).SpineAngle(:,2)=S(trialn(xxx)).UThorAngle(:,2)-S(trialn(xxx)).PelvisAngle(:,2);
S(trialn(xxx)).SpineAngle(:,3)=S(trialn(xxx)).UThorAngle(:,3)-S(trialn(xxx)).PelvisAngle(:,3);
%Upper Spine (Coronal=1, Sagittal=2, Transverse=3)
S(trialn(xxx)).USpineAngle(:,1)=S(trialn(xxx)).UThorAngle(:,1)-S(trialn(xxx)).MThorAngle(:,1);
S(trialn(xxx)).USpineAngle(:,2)=S(trialn(xxx)).UThorAngle(:,2)-S(trialn(xxx)).MThorAngle(:,2);
S(trialn(xxx)).USpineAngle(:,3)=S(trialn(xxx)).UThorAngle(:,3)-S(trialn(xxx)).MThorAngle(:,3);
%Lower Spine (Coronal=1, Sagittal=2, Transverse=3)
S(trialn(xxx)).LSpineAngle(:,1)=S(trialn(xxx)).MThorAngle(:,1)-S(trialn(xxx)).PelvisAngle(:,1);
S(trialn(xxx)).LSpineAngle(:,2)=S(trialn(xxx)).MThorAngle(:,2)-S(trialn(xxx)).PelvisAngle(:,2);
S(trialn(xxx)).LSpineAngle(:,3)=S(trialn(xxx)).MThorAngle(:,3)-S(trialn(xxx)).PelvisAngle(:,3);
end
save([cpath,patnum,'\Evaluation ',evalnum,'\',patnum,'_dynamicspine.mat']);

```

Spineanalyze

```

%1-20-11
%MS
%This program will normalize the time for each of the motions so that they can be compared quickly
across a single %subject.This program will also calculate the max and min for each motion and the point at
which the max and min %occur.
% Breaking up the single motion trial into the three motions
for xxx=1:length(trialn)
    S(trialn(xxx)).time=round(S(trialn(xxx)).eventtime_sort*60);

S(trialn(xxx)).UThor1(:,1)=S(trialn(xxx)).UThorAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),1);
S(trialn(xxx)).UThor2(:,1)=S(trialn(xxx)).UThorAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),1);
S(trialn(xxx)).UThor3(:,1)=S(trialn(xxx)).UThorAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),1);
S(trialn(xxx)).UThor1(:,2)=S(trialn(xxx)).UThorAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),2);
S(trialn(xxx)).UThor2(:,2)=S(trialn(xxx)).UThorAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),2);
S(trialn(xxx)).UThor3(:,2)=S(trialn(xxx)).UThorAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),2);
S(trialn(xxx)).UThor1(:,3)=S(trialn(xxx)).UThorAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),3);
S(trialn(xxx)).UThor2(:,3)=S(trialn(xxx)).UThorAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),3);
S(trialn(xxx)).UThor3(:,3)=S(trialn(xxx)).UThorAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),3);

```


S(trialn(xxx)).MThor1(:,1)=S(trialn(xxx)).MThorAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),1);
 ;
 S(trialn(xxx)).MThor2(:,1)=S(trialn(xxx)).MThorAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),1);
 ;
 S(trialn(xxx)).MThor3(:,1)=S(trialn(xxx)).MThorAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),1);
 ;
 S(trialn(xxx)).MThor1(:,2)=S(trialn(xxx)).MThorAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),2);
 ;
 S(trialn(xxx)).MThor2(:,2)=S(trialn(xxx)).MThorAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),2);
 ;
 S(trialn(xxx)).MThor3(:,2)=S(trialn(xxx)).MThorAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),2);
 ;
 S(trialn(xxx)).MThor1(:,3)=S(trialn(xxx)).MThorAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),3);
 ;
 S(trialn(xxx)).MThor2(:,3)=S(trialn(xxx)).MThorAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),3);
 ;
 S(trialn(xxx)).MThor3(:,3)=S(trialn(xxx)).MThorAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),3);
 ;
 S(trialn(xxx)).Pelvis1(:,1)=S(trialn(xxx)).PelvisAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),1);
 S(trialn(xxx)).Pelvis2(:,1)=S(trialn(xxx)).PelvisAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),1);
 S(trialn(xxx)).Pelvis3(:,1)=S(trialn(xxx)).PelvisAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),1);
 S(trialn(xxx)).Pelvis1(:,2)=S(trialn(xxx)).PelvisAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),2);
 S(trialn(xxx)).Pelvis2(:,2)=S(trialn(xxx)).PelvisAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),2);
 S(trialn(xxx)).Pelvis3(:,2)=S(trialn(xxx)).PelvisAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),2);
 S(trialn(xxx)).Pelvis1(:,3)=S(trialn(xxx)).PelvisAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),3);
 S(trialn(xxx)).Pelvis2(:,3)=S(trialn(xxx)).PelvisAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),3);
 S(trialn(xxx)).Pelvis3(:,3)=S(trialn(xxx)).PelvisAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),3);
 S(trialn(xxx)).Spine1(:,1)=S(trialn(xxx)).SpineAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),1);
 S(trialn(xxx)).Spine2(:,1)=S(trialn(xxx)).SpineAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),1);
 S(trialn(xxx)).Spine3(:,1)=S(trialn(xxx)).SpineAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),1);
 S(trialn(xxx)).Spine1(:,2)=S(trialn(xxx)).SpineAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),2);
 S(trialn(xxx)).Spine2(:,2)=S(trialn(xxx)).SpineAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),2);
 S(trialn(xxx)).Spine3(:,2)=S(trialn(xxx)).SpineAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),2);
 S(trialn(xxx)).Spine1(:,3)=S(trialn(xxx)).SpineAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),3);
 S(trialn(xxx)).Spine2(:,3)=S(trialn(xxx)).SpineAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),3);
 S(trialn(xxx)).Spine3(:,3)=S(trialn(xxx)).SpineAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),3);

```

S(trialn(xxx)).USpine1(:,1)=S(trialn(xxx)).USpineAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),1
);
S(trialn(xxx)).USpine2(:,1)=S(trialn(xxx)).USpineAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),1
);
S(trialn(xxx)).USpine3(:,1)=S(trialn(xxx)).USpineAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),1
);
S(trialn(xxx)).USpine1(:,2)=S(trialn(xxx)).USpineAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),2
);
S(trialn(xxx)).USpine2(:,2)=S(trialn(xxx)).USpineAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),2
);
S(trialn(xxx)).USpine3(:,2)=S(trialn(xxx)).USpineAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),2
);
S(trialn(xxx)).USpine1(:,3)=S(trialn(xxx)).USpineAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),3
);
S(trialn(xxx)).USpine2(:,3)=S(trialn(xxx)).USpineAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),3
);
S(trialn(xxx)).USpine3(:,3)=S(trialn(xxx)).USpineAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),3
);
S(trialn(xxx)).LSpine1(:,1)=S(trialn(xxx)).LSpineAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),1)
;
S(trialn(xxx)).LSpine2(:,1)=S(trialn(xxx)).LSpineAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),1)
;
S(trialn(xxx)).LSpine3(:,1)=S(trialn(xxx)).LSpineAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),1)
;
S(trialn(xxx)).LSpine1(:,2)=S(trialn(xxx)).LSpineAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),2)
;
S(trialn(xxx)).LSpine2(:,2)=S(trialn(xxx)).LSpineAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),2)
;
S(trialn(xxx)).LSpine3(:,2)=S(trialn(xxx)).LSpineAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),2)
;
S(trialn(xxx)).LSpine1(:,3)=S(trialn(xxx)).LSpineAngle(S(trialn(xxx)).time(1,2):S(trialn(xxx)).time(2,2),3)
;
S(trialn(xxx)).LSpine2(:,3)=S(trialn(xxx)).LSpineAngle(S(trialn(xxx)).time(2,2):S(trialn(xxx)).time(3,2),3)
;
S(trialn(xxx)).LSpine3(:,3)=S(trialn(xxx)).LSpineAngle(S(trialn(xxx)).time(3,2):S(trialn(xxx)).time(4,2),3)
;
end

```

```

normalize;
%Finding the max and min
for xxx=1:length(trialn)
    S(trialn(xxx)).UTmax1=max(S(trialn(xxx)).UThor1);
S(trialn(xxx)).UTmin1=min(S(trialn(xxx)).UThor1);
    S(trialn(xxx)).UTmax2=max(S(trialn(xxx)).UThor2);
S(trialn(xxx)).UTmin2=min(S(trialn(xxx)).UThor2);
    S(trialn(xxx)).UTmax3=max(S(trialn(xxx)).UThor3);
S(trialn(xxx)).UTmin3=min(S(trialn(xxx)).UThor3);
    S(trialn(xxx)).MTmax1=max(S(trialn(xxx)).MThor1);
S(trialn(xxx)).MTmin1=min(S(trialn(xxx)).MThor1);
    S(trialn(xxx)).MTmax2=max(S(trialn(xxx)).MThor2);
S(trialn(xxx)).MTmin2=min(S(trialn(xxx)).MThor2);
    S(trialn(xxx)).MTmax3=max(S(trialn(xxx)).MThor3);
S(trialn(xxx)).MTmin3=min(S(trialn(xxx)).MThor3);
    S(trialn(xxx)).Pelmax1=max(S(trialn(xxx)).Pelvis1);
S(trialn(xxx)).Pelmin1=min(S(trialn(xxx)).Pelvis1);
    S(trialn(xxx)).Pelmax2=max(S(trialn(xxx)).Pelvis2);
S(trialn(xxx)).Pelmin2=min(S(trialn(xxx)).Pelvis2);
    S(trialn(xxx)).Pelmax3=max(S(trialn(xxx)).Pelvis3);
S(trialn(xxx)).Pelmin3=min(S(trialn(xxx)).Pelvis3);
    S(trialn(xxx)).Smax1=max(S(trialn(xxx)).Spine1); S(trialn(xxx)).Smin1=min(S(trialn(xxx)).Spine1);
S(trialn(xxx)).Smax2=max(S(trialn(xxx)).Spine2); S(trialn(xxx)).Smin2=min(S(trialn(xxx)).Spine2);
    S(trialn(xxx)).Smax3=max(S(trialn(xxx)).Spine3); S(trialn(xxx)).Smin3=min(S(trialn(xxx)).Spine3);
    S(trialn(xxx)).USmax1=max(S(trialn(xxx)).USpine1);
S(trialn(xxx)).USmin1=min(S(trialn(xxx)).USpine1);
    S(trialn(xxx)).USmax2=max(S(trialn(xxx)).USpine2);
S(trialn(xxx)).USmin2=min(S(trialn(xxx)).USpine2);
    S(trialn(xxx)).USmax3=max(S(trialn(xxx)).USpine3);
S(trialn(xxx)).USmin3=min(S(trialn(xxx)).USpine3);
    S(trialn(xxx)).LSmax1=max(S(trialn(xxx)).LSpine1);
S(trialn(xxx)).LSmin1=min(S(trialn(xxx)).LSpine1);
    S(trialn(xxx)).LSmax2=max(S(trialn(xxx)).LSpine2);
S(trialn(xxx)).LSmin2=min(S(trialn(xxx)).LSpine2);
    S(trialn(xxx)).LSmax3=max(S(trialn(xxx)).LSpine3);
S(trialn(xxx)).LSmin3=min(S(trialn(xxx)).LSpine3);
end

```

Normalize

```

%3-3-11
%MS
%Resampling data to produce mean plots for the spine study. Since the data is based on frame data it is
very %difficult to create a mean curve since some data will have 145 points and others 350 points.
Therefore, this %oprogram will normalize the data to create a motion pattern based on a percentage of the
motion cycle, 1% starts %the cycle 100% ends the cycle.
%determining the length of each vector
for xxx=1:length(trialn)
    S(trialn(xxx)).frame1=length(S(trialn(xxx)).UThor1(:,1));
    S(trialn(xxx)).frame2=length(S(trialn(xxx)).UThor2(:,1));
    S(trialn(xxx)).frame3=length(S(trialn(xxx)).UThor3(:,1));
end
%calculate the reduction amount
for xxx=1:length(trialn)
    if S(trialn(xxx)).frame1<=100

```

```

    S(trialn(xxx)).cut1=1;
elseif S(trialn(xxx)).frame1>100 && S(trialn(xxx)).frame1<=200
    S(trialn(xxx)).cut1=2;
elseif S(trialn(xxx)).frame1>200 && S(trialn(xxx)).frame1<=300
    S(trialn(xxx)).cut1=3;
elseif S(trialn(xxx)).frame1>300 && S(trialn(xxx)).frame1<=400
    S(trialn(xxx)).cut1=4;
elseif S(trialn(xxx)).frame1>400 && S(trialn(xxx)).frame1<=500
    S(trialn(xxx)).cut1=5;
end
if S(trialn(xxx)).frame2<=100
    S(trialn(xxx)).cut2=1;
elseif S(trialn(xxx)).frame2>100 && S(trialn(xxx)).frame2<=200
    S(trialn(xxx)).cut2=2;
elseif S(trialn(xxx)).frame2>200 && S(trialn(xxx)).frame2<=300
    S(trialn(xxx)).cut2=3;
elseif S(trialn(xxx)).frame2>300 && S(trialn(xxx)).frame2<=400
    S(trialn(xxx)).cut2=4;
elseif S(trialn(xxx)).frame2>400 && S(trialn(xxx)).frame2<=500
    S(trialn(xxx)).cut2=5;
end
if S(trialn(xxx)).frame3<=100
    S(trialn(xxx)).cut3=1;
elseif S(trialn(xxx)).frame3>100 && S(trialn(xxx)).frame3<=200
    S(trialn(xxx)).cut3=2;
elseif S(trialn(xxx)).frame3>200 && S(trialn(xxx)).frame3<=300
    S(trialn(xxx)).cut3=3;
elseif S(trialn(xxx)).frame3>300 && S(trialn(xxx)).frame3<=400
    S(trialn(xxx)).cut3=4;
elseif S(trialn(xxx)).frame3>400 && S(trialn(xxx)).frame3<=500
    S(trialn(xxx)).cut3=5;
end
end
end

```

%perform the reduction and normalization
downsample will reduce the sample rate without affecting the signal

%quality or signal peaks. Fixlength1 will then normalize the signal to 100%

for xxx=1:length(trialn)

```

    S(trialn(xxx)).dUThor1=downsample(S(trialn(xxx)).UThor1,S(trialn(xxx)).cut1);
    S(trialn(xxx)).dUThor2=downsample(S(trialn(xxx)).UThor2,S(trialn(xxx)).cut2);
    S(trialn(xxx)).dUThor3=downsample(S(trialn(xxx)).UThor3,S(trialn(xxx)).cut3);
    S(trialn(xxx)).dMThor1=downsample(S(trialn(xxx)).MThor1,S(trialn(xxx)).cut1);
    S(trialn(xxx)).dMThor2=downsample(S(trialn(xxx)).MThor2,S(trialn(xxx)).cut2);
    S(trialn(xxx)).dMThor3=downsample(S(trialn(xxx)).MThor3,S(trialn(xxx)).cut3);
    S(trialn(xxx)).dPelvis1=downsample(S(trialn(xxx)).Pelvis1,S(trialn(xxx)).cut1);
    S(trialn(xxx)).dPelvis2=downsample(S(trialn(xxx)).Pelvis2,S(trialn(xxx)).cut2);
    S(trialn(xxx)).dPelvis3=downsample(S(trialn(xxx)).Pelvis3,S(trialn(xxx)).cut3);
    S(trialn(xxx)).dSpine1=downsample(S(trialn(xxx)).Spine1,S(trialn(xxx)).cut1);
    S(trialn(xxx)).dSpine2=downsample(S(trialn(xxx)).Spine2,S(trialn(xxx)).cut2);
    S(trialn(xxx)).dSpine3=downsample(S(trialn(xxx)).Spine3,S(trialn(xxx)).cut3);
    S(trialn(xxx)).dUSpine1=downsample(S(trialn(xxx)).USpine1,S(trialn(xxx)).cut1);
    S(trialn(xxx)).dUSpine2=downsample(S(trialn(xxx)).USpine2,S(trialn(xxx)).cut2);
    S(trialn(xxx)).dUSpine3=downsample(S(trialn(xxx)).USpine3,S(trialn(xxx)).cut3);
    S(trialn(xxx)).dLSpine1=downsample(S(trialn(xxx)).LSpine1,S(trialn(xxx)).cut1);

```

```

S(trialn(xxx)).dLSpine2=downsample(S(trialn(xxx)).LSpine2,S(trialn(xxx)).cut2);
S(trialn(xxx)).dLSpine3=downsample(S(trialn(xxx)).LSpine3,S(trialn(xxx)).cut3);
S(trialn(xxx)).nUThor1(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dUThor1),S(trialn(xxx)).dUThor1(1:length(S(trialn(xxx)).dUThor1(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nUThor2(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dUThor2),S(trialn(xxx)).dUThor2(1:length(S(trialn(xxx)).dUThor2(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nUThor3(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dUThor3),S(trialn(xxx)).dUThor3(1:length(S(trialn(xxx)).dUThor3(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nMThor1(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dMThor1),S(trialn(xxx)).dMThor1(1:length(S(trialn(xxx)).dMThor1(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nMThor2(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dMThor2),S(trialn(xxx)).dMThor2(1:length(S(trialn(xxx)).dMThor2(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nMThor3(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dMThor3),S(trialn(xxx)).dMThor3(1:length(S(trialn(xxx)).dMThor3(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nPelvis1(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dPelvis1),S(trialn(xxx)).dPelvis1(1:length(S(trialn(xxx)).dPelvis1(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nPelvis2(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dPelvis2),S(trialn(xxx)).dPelvis2(1:length(S(trialn(xxx)).dPelvis2(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nPelvis3(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dPelvis3),S(trialn(xxx)).dPelvis3(1:length(S(trialn(xxx)).dPelvis3(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nSpine1(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dSpine1),S(trialn(xxx)).dSpine1(1:length(S(trialn(xxx)).dSpine1(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nSpine2(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dSpine2),S(trialn(xxx)).dSpine2(1:length(S(trialn(xxx)).dSpine2(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nSpine3(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dSpine3),S(trialn(xxx)).dSpine3(1:length(S(trialn(xxx)).dSpine3(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nUSpine1(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dUSpine1),S(trialn(xxx)).dUSpine1(1:length(S(trialn(xxx)).dUSpine1(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nUSpine2(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dUSpine2),S(trialn(xxx)).dUSpine2(1:length(S(trialn(xxx)).dUSpine2(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nUSpine3(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dUSpine3),S(trialn(xxx)).dUSpine3(1:length(S(trialn(xxx)).dUSpine3(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nLSpine1(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dLSpine1),S(trialn(xxx)).dLSpine1(1:length(S(trialn(xxx)).dLSpine1(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nLSpine2(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dLSpine2),S(trialn(xxx)).dLSpine2(1:length(S(trialn(xxx)).dLSpine2(:,1)),:),1:100,2,bbutt,abutt);
S(trialn(xxx)).nLSpine3(1:100,:)=fixlength1(1:length(S(trialn(xxx)).dLSpine3),S(trialn(xxx)).dLSpine3(1:length(S(trialn(xxx)).dLSpine3(:,1)),:),1:100,2,bbutt,abutt);
end

```

Fixlength

```

% code to normalize/change the data range for a nd array. An expansion of mukul's tnorm.
% function[y1]=fixlength1(x,y,x1,mthd,bbutt,abutt)
% x is the non-resampled time window.
% y is the non-resampled data.
% x1 is the resampled time window.
% y1 is the resampled data.
function[y1]=fixlength1(x,y,x1,mthd,bbutt,abutt)
[leny,tmpdmp]=size(y);
leny1=length(x1);
pltfilg=0;
% Use a filter, to smoothen the data.
y1=filtfilt(bbutt,abutt,y1);

```

```

elseif mthd==1,% inbuild command. Gives much smoother data, as matlab uses filtering to smoothen the
data.
    while tmpdmp~=0
        y1(:,tmpdmp) = resample(y(:,tmpdmp),leny1,leny,2);
        tmpdmp=tmpdmp-1;
    end
end
end

```

Spinegraph

1-25-11

%MS

%SpineGraph this is the graphing routine for the spine program. There are a total of 6 figures per subject. Figures 1 %to 3 are plane motions (Pelvis and Thorax) and Figures 4 to 6 are segment motions (Spine and spine sections). On %each plot there are the three motion trials. The graphs follow the standard gait method. 1st column coronal plane, %2nd column sagittal plane, 3rd column transverse plane.

%Lateral Bending Planar motion

figure (1), ORIENT LANDSCAPE

```

subplot(3,3,1),hold on,plot(S(1,8).nUThor1(:,1),'b');plot(S(1,8).nUThor2(:,1),'r');
axis([0 100 -50 50]); title('Lateral Bend    Upper Thorax Cor. '); ylabel('Angle (deg) Left
+ve');xlabel('Time normalized (%)');
subplot(3,3,2),hold on,plot(S(1,8).nUThor1(:,2),'b');plot(S(1,8).nUThor2(:,2),'r');
axis([0 100 -40 40]); title('Upper Thorax Sag. '); ylabel('Angle (deg) flex +ve');xlabel('Time normalized
(%)' );
subplot(3,3,3),hold on,plot(S(1,8).nUThor1(:,3),'b');plot(S(1,8).nUThor2(:,3),'r');
axis([0 100 -40 40]);title('Upper Thorax Tran. '); ylabel('Angle (deg)int. +ve');xlabel('Time normalized
(%)' );
subplot(3,3,4),hold on,plot(S(1,8).nMThor1(:,1),'b');plot(S(1,8).nMThor2(:,1),'r');
axis([0 100 -50 50]); title('Mid-Thorax Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time normalized (%)');
subplot(3,3,5),hold on,plot(S(1,8).nMThor1(:,2),'b');plot(S(1,8).nMThor2(:,2),'r');
axis([0 100 -40 40]); title('Mid-Thorax Sag. '); ylabel('Angle (deg) +flex');xlabel('Time normalized (%)');
subplot(3,3,6),hold on,plot(S(1,8).nMThor1(:,3),'b');plot(S(1,8).nMThor2(:,3),'r');
axis([0 100 -40 40]);title('Mid-Thorax Tran. '); ylabel('Angle (deg) int +ve');xlabel('Time normalized (%)');
subplot(3,3,7),hold on,plot(S(1,8).nPelvis1(:,1),'b');plot(S(1,8).nPelvis2(:,1),'r');
axis([0 100 -50 50]); title('Pelvis Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time normalized (%)');
subplot(3,3,8),hold on,plot(S(1,8).nPelvis1(:,2),'b');plot(S(1,8).nPelvis2(:,2),'r');
axis([0 100 -40 40]); title('Pelvis Sag. '); ylabel('Angle (deg) flex +ve');xlabel('Time normalized (%)');
subplot(3,3,9),hold on,plot(S(1,8).nPelvis1(:,3),'b');plot(S(1,8).nPelvis2(:,3),'r');
axis([0 100 -40 40]); title('Pelvis Tran. '); ylabel('Angle (deg) int +ve');xlabel('Time normalized (%)');
legend('Trial 1','Trial 2','Location','BestOutside');

```

%Rotation Planar motion

figure (2), ORIENT LANDSCAPE

```

subplot(3,3,1),hold on,plot(S(1,9).nUThor1(:,1),'b');plot(S(1,9).nUThor2(:,1),'r');
axis([0 100 -30 30]); title('Rotation    Upper Thorax Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time
normalized (%)');
subplot(3,3,2),hold on,plot(S(1,9).nUThor1(:,2),'b');plot(S(1,9).nUThor2(:,2),'r');
axis([0 100 -30 30]); title('Upper Thorax Sag. '); ylabel('Angle (deg) flex +ve');xlabel('Time normalized
(%)' );
subplot(3,3,3),hold on,plot(S(1,9).nUThor1(:,3),'b');plot(S(1,9).nUThor2(:,3),'r');
axis([0 100 -200 200]);title('Upper Thorax Tran. '); ylabel('Angle (deg)int. +ve');xlabel('Time normalized
(%)' );
subplot(3,3,4),hold on,plot(S(1,9).nMThor1(:,1),'b');plot(S(1,9).nMThor2(:,1),'r');
axis([0 100 -30 30]); title('Mid-Thorax Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time normalized (%)');

```

```

subplot(3,3,5),hold on,plot(S(1,9).nMThor1(:,2),'b');plot(S(1,9).nMThor2(:,2),'r');
axis([0 100 -30 30]); title('Mid-Thorax Sag. '); ylabel('Angle (deg) +flex');xlabel('Time normalized (%)');
subplot(3,3,6),hold on,plot(S(1,9).nMThor1(:,3),'b');plot(S(1,9).nMThor2(:,3),'r');
axis([0 100 -200 200]);title('Mid-Thorax Tran. '); ylabel('Angle (deg) int +ve');xlabel('Time normalized (%)');
subplot(3,3,7),hold on,plot(S(1,9).nPelvis1(:,1),'b');plot(S(1,9).nPelvis2(:,1),'r');
axis([0 100 -30 30]); title('Pelvis Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time normalized (%)');
subplot(3,3,8),hold on,plot(S(1,9).nPelvis1(:,2),'b');plot(S(1,9).nPelvis2(:,2),'r');
axis([0 100 -30 30]); title('Pelvis Sag. '); ylabel('Angle (deg) flex +ve');xlabel('Time normalized (%)');
subplot(3,3,9),hold on,plot(S(1,9).nPelvis1(:,3),'b');plot(S(1,9).nPelvis2(:,3),'r');
axis([0 100 -200 200]); title('Pelvis Tran. '); ylabel('Angle (deg) int +ve');xlabel('Time normalized (%)');
legend('Trial 1','Trial 2','Location','BestOutside');

```

%Flexion Planar motion

figure (3), ORIENT LANDSCAPE

```

subplot(3,3,1),hold on,plot(S(1,10).nUThor2(:,1),'b');plot(S(1,10).nUThor3(:,1),'r');
axis([0 100 -30 30]); title('Flexion Upper Thorax Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time normalized (%)');
subplot(3,3,2),hold on,plot(S(1,10).nUThor2(:,2),'b');plot(S(1,10).nUThor3(:,2),'r');
axis([0 100 -50 150]); title('Upper Thorax Sag. '); ylabel('Angle (deg) flex +ve');xlabel('Time normalized (%)');
subplot(3,3,3),hold on,plot(S(1,10).nUThor2(:,3),'b');plot(S(1,10).nUThor3(:,3),'r');
axis([0 100 -30 30]);title('Upper Thorax Tran. '); ylabel('Angle (deg)int. +ve');xlabel('Time normalized (%)');
subplot(3,3,4),hold on,plot(S(1,10).nMThor2(:,1),'b');plot(S(1,10).nMThor3(:,1),'r');
axis([0 100 -30 30]); title('Mid-Thorax Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time normalized (%)');
subplot(3,3,5),hold on,plot(S(1,10).nMThor2(:,2),'b');plot(S(1,10).nMThor3(:,2),'r');
axis([0 100 -50 150]); title('Mid-Thorax Sag. '); ylabel('Angle (deg) +flex');xlabel('Time normalized (%)');
subplot(3,3,6),hold on,plot(S(1,10).nMThor2(:,3),'b');plot(S(1,10).nMThor3(:,3),'r');
axis([0 100 -30 30]);title('Mid-Thorax Tran. '); ylabel('Angle (deg) int +ve');xlabel('Time normalized (%)');
subplot(3,3,7),hold on,plot(S(1,10).nPelvis2(:,1),'b');plot(S(1,10).nPelvis3(:,1),'r');
axis([0 100 -30 30]); title('Pelvis Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time normalized (%)');
subplot(3,3,8),hold on,plot(S(1,10).nPelvis2(:,2),'b');plot(S(1,10).nPelvis3(:,2),'r');
axis([0 100 -50 150]); title('Pelvis Sag. '); ylabel('Angle (deg) flex +ve');xlabel('Time normalized (%)');
subplot(3,3,9),hold on,plot(S(1,10).nPelvis2(:,3),'b');plot(S(1,10).nPelvis3(:,3),'r');
axis([0 100 -30 30]); title('Pelvis Tran. '); ylabel('Angle (deg) int +ve');xlabel('Time normalized (%)');
legend('Trial 1','Trial 2','Location','BestOutside');

```

%Lateral Bending Segment Motion

figure (4), ORIENT LANDSCAPE

```

subplot(3,3,1),hold on,plot(S(1,8).nUSpine1(:,1),'b');plot(S(1,8).nUSpine2(:,1),'r');
axis([0 100 -50 50]); title('Lateral Bend Upper Spine Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time normalized (%)');
subplot(3,3,2),hold on,plot(S(1,8).nUSpine1(:,2),'b');plot(S(1,8).nUSpine2(:,2),'r');
axis([0 100 -40 40]); title('Upper Spine Sag. '); ylabel('Angle (deg) Flex +ve');xlabel('Time normalized (%)');
subplot(3,3,3),hold on,plot(S(1,8).nUSpine1(:,3),'b');plot(S(1,8).nUSpine2(:,3),'r');
axis([0 100 -40 40]); title('Upper Spine Tran. '); ylabel('Angle (deg) Int +ve');xlabel('Time normalized (%)');
subplot(3,3,4),hold on,plot(S(1,8).nLSpine1(:,1),'b');plot(S(1,8).nLSpine2(:,1),'r');
axis([0 100 -50 50]); title('Lower Spine Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time normalized (%)');
subplot(3,3,5),hold on,plot(S(1,8).nLSpine1(:,2),'b');plot(S(1,8).nLSpine2(:,2),'r');
axis([0 100 -40 40]); title('Lower Spine Sag. '); ylabel('Angle (deg) Flex +ve');xlabel('Time normalized (%)');

```

```

subplot(3,3,6),hold on,plot(S(1,8).nLSpine1(:,3),'b');plot(S(1,8).nLSpine2(:,3),'r')
axis([0 100 -40 40]); title('Lower Spine Tran. '); ylabel('Angle (deg) Int +ve');xlabel('Time normalized (%)');
subplot(3,3,7),hold on,plot(S(1,8).nSpine1(:,1),'b');plot(S(1,8).nSpine2(:,1),'r');
axis([0 100 -50 50]); title('Total Spine Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time normalized (%)');
subplot(3,3,8),hold on,plot(S(1,8).nSpine1(:,2),'b');plot(S(1,8).nSpine2(:,2),'r');
axis([0 100 -40 40]); title('Total Spine Sag. '); ylabel('Angle (deg) Flex +ve');xlabel('Time normalized (%)');
subplot(3,3,9),hold on,plot(S(1,8).nSpine1(:,3),'b');plot(S(1,8).nSpine2(:,3),'r');
axis([0 100 -40 40]); title('Total Spine Tran. '); ylabel('Angle (deg) Int +ve');xlabel('Time normalized (%)');
legend('Trial 1','Trial 2','Location','BestOutside');

```

%Twisting Segment Motion

figure (5), ORIENT LANDSCAPE

```

subplot(3,3,1),hold on,plot(S(1,9).nUSpine1(:,1),'b');plot(S(1,9).nUSpine2(:,1),'r');
axis([0 100 -40 40]); title('Rotation Bend Upper Spine Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time normalized (%)');
subplot(3,3,2),hold on,plot(S(1,9).nUSpine1(:,2),'b');plot(S(1,9).nUSpine2(:,2),'r')
axis([0 100 -40 40]); title('Upper Spine Sag. '); ylabel('Angle (deg) Flex +ve');xlabel('Time normalized (%)');
subplot(3,3,3),hold on,plot(S(1,9).nUSpine1(:,3),'b');plot(S(1,9).nUSpine2(:,3),'r')
axis([0 100 -150 150]); title('Upper Spine Tran. '); ylabel('Angle (deg) Int +ve');xlabel('Time normalized (%)');
subplot(3,3,4),hold on,plot(S(1,9).nLSpine1(:,1),'b');plot(S(1,9).nLSpine2(:,1),'r')
axis([0 100 -40 40]); title('Lower Spine Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time normalized (%)');
subplot(3,3,5),hold on,plot(S(1,9).nLSpine1(:,2),'b');plot(S(1,9).nLSpine2(:,2),'r')
axis([0 100 -40 40]); title('Lower Spine Sag. '); ylabel('Angle (deg) Flex +ve');xlabel('Time normalized (%)');
subplot(3,3,6),hold on,plot(S(1,9).nLSpine1(:,3),'b');plot(S(1,9).nLSpine2(:,3),'r')
axis([0 100 -150 150]); title('Lower Spine Tran. '); ylabel('Angle (deg) Int +ve');xlabel('Time normalized (%)');
subplot(3,3,7),hold on,plot(S(1,9).nSpine1(:,1),'b');plot(S(1,9).nSpine2(:,1),'r');
axis([0 100 -40 40]); title('Total Spine Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time normalized (%)');
subplot(3,3,8),hold on,plot(S(1,9).nSpine1(:,2),'b');plot(S(1,9).nSpine2(:,2),'r');
axis([0 100 -40 40]); title('Total Spine Sag. '); ylabel('Angle (deg) Flex +ve');xlabel('Time normalized (%)');
subplot(3,3,9),hold on,plot(S(1,9).nSpine1(:,3),'b');plot(S(1,9).nSpine2(:,3),'r');
axis([0 100 -150 150]); title('Total Spine Tran. '); ylabel('Angle (deg) Int +ve');xlabel('Time normalized (%)');
legend('Trial 1','Trial 2','Location','BestOutside');

```

%Flex/Ext Segment Motion

figure (6), ORIENT LANDSCAPE

```

subplot(3,3,1),hold on,plot(S(1,10).nUSpine2(:,1),'b');plot(S(1,10).nUSpine3(:,1),'r');
axis([0 100 -30 30]); title('Flexion Bend Upper Spine Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time normalized (%)');
subplot(3,3,2),hold on,plot(S(1,10).nUSpine2(:,2),'b');plot(S(1,10).nUSpine3(:,2),'r')
axis([0 100 -100 100]); title('Upper Spine Sag. '); ylabel('Angle (deg) Flex +ve');xlabel('Time normalized (%)');
subplot(3,3,3),hold on,plot(S(1,10).nUSpine2(:,3),'b');plot(S(1,10).nUSpine3(:,3),'r')
axis([0 100 -30 30]); title('Upper Spine Tran. '); ylabel('Angle (deg) Int +ve');xlabel('Time normalized (%)');
subplot(3,3,4),hold on,plot(S(1,10).nLSpine2(:,1),'b');plot(S(1,10).nLSpine3(:,1),'r')
axis([0 100 -30 30]); title('Lower Spine Cor. '); ylabel('Angle (deg) Left +ve');xlabel('Time normalized (%)');
subplot(3,3,5),hold on,plot(S(1,10).nLSpine2(:,2),'b');plot(S(1,10).nLSpine3(:,2),'r')

```



```

axis([0 100 -100 100]); title('Lower Spine Sag. '); ylabel('Angle (deg) Flex +ve'); xlabel('Time normalized (%)');
subplot(3,3,6),hold on,plot(S(1,10).nLSpine2(:,3),'b');plot(S(1,10).nLSpine3(:,3),'r')
axis([0 100 -30 30]); title('Lower Spine Tran. '); ylabel('Angle (deg) Int +ve'); xlabel('Time normalized (%)');
subplot(3,3,7),hold on,plot(S(1,10).nSpine2(:,1),'b');plot(S(1,10).nSpine3(:,1),'r');
axis([0 100 -30 30]); title('Total Spine Cor. '); ylabel('Angle (deg) Left +ve'); xlabel('Time normalized (%)');
subplot(3,3,8),hold on,plot(S(1,10).nSpine2(:,2),'b');plot(S(1,10).nSpine3(:,2),'r');
axis([0 100 -100 100]); title('Total Spine Sag. '); ylabel('Angle (deg) Flex +ve'); xlabel('Time normalized (%)');
subplot(3,3,9),hold on,plot(S(1,10).nSpine2(:,3),'b');plot(S(1,10).nSpine3(:,3),'r');
axis([0 100 -30 30]); title('Total Spine Tran. '); ylabel('Angle (deg) Int +ve'); xlabel('Time normalized (%)');
legend('Trial 1','Trial 2','Location','BestOutside');

warning off
for i=1:6
    s1=['print -dtiff -f,num2str(i)']; eval(s1);
end

```

Dynamicoutput

```

%1-31-11
%MS
%Dynamic Output
%This program will output Max and Min data for motion trials. This will also calculate the ROM for each
segment %and plane.
close all;

vind=0;
%Output Max and Min
varlist{vind+1,1}='Lateral Bending Trial 1';
varlist{vind+1,1}='Anatomical Level'; varlist{vind+1,3}='Coronal';
varlist{vind+1,4}='Sagittal'; varlist{vind+1,5}='Transverse'; vind=vind+1;
varlist{vind+1,1}='Upper Thorax'; vind=vind+1;
varlist{vind+1,2}='Max'; varlist{vind+1,3}=num2str((S(1,8).UTmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).UTmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).UTmax1(:,3))); vind=vind+1;
varlist{vind+1,2}='Min'; varlist{vind+1,3}=num2str((S(1,8).UTmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).UTmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).UTmin1(:,3))); vind=vind+1;
varlist{vind+1,2}='ROM'; varlist{vind+1,3}=num2str((S(1,8).UTmax1(:,1)-S(1,8).UTmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).UTmax1(:,2)-S(1,8).UTmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).UTmax1(:,3)-S(1,8).UTmin1(:,3))); vind=vind+1;
varlist{vind+1,1}='Mid Thorax'; vind=vind+1;
varlist{vind+1,2}='Max'; varlist{vind+1,3}=num2str((S(1,8).MTmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).MTmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).MTmax1(:,3))); vind=vind+1;
varlist{vind+1,2}='Min'; varlist{vind+1,3}=num2str((S(1,8).MTmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).MTmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).MTmin1(:,3))); vind=vind+1;
varlist{vind+1,2}='ROM'; varlist{vind+1,3}=num2str((S(1,8).MTmax1(:,1)-S(1,8).MTmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).MTmax1(:,2)-S(1,8).MTmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).MTmax1(:,3)-S(1,8).MTmin1(:,3))); vind=vind+1;
varlist{vind+1,1}='Pelvis'; vind=vind+1;

```

```

varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).Pelmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).Pelmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).Pelmax1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).Pelmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).Pelmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).Pelmin1(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,8).Pelmax1(:,1)-S(1,8).Pelmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).Pelmax1(:,2)-S(1,8).Pelmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).Pelmax1(:,3)-S(1,8).Pelmin1(:,3)));  vind=vind+1;
varlist{vind+1,1}='Upper Spine';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).USmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).USmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).USmax1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).USmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).USmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).USmin1(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,8).USmax1(:,1)-S(1,8).USmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).USmax1(:,2)-S(1,8).USmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).USmax1(:,3)-S(1,8).USmin1(:,3)));  vind=vind+1;
varlist{vind+1,1}='Lower Spine';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).LSmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).LSmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).LSmax1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).LSmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).LSmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).LSmin1(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,8).LSmax1(:,1)-S(1,8).LSmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).LSmax1(:,2)-S(1,8).LSmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).LSmax1(:,3)-S(1,8).LSmin1(:,3)));  vind=vind+1;
varlist{vind+1,1}='Total Spine';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).Smax1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).Smax1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).Smax1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).Smin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).Smin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).Smin1(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,8).Smax1(:,1)-S(1,8).Smin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).Smax1(:,2)-S(1,8).Smin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).Smax1(:,3)-S(1,8).Smin1(:,3)));  vind=vind+1;
varlist{vind+1,1}='Upper Thorax Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).UTAmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).UTAmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).UTAmax1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).UTAmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).UTAmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).UTAmin1(:,3)));      vind=vind+1;
varlist{vind+1,1}='Mid Thorax Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).MTAmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).MTAmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).MTAmax1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).MTAmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).MTAmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).MTAmin1(:,3)));      vind=vind+1;
varlist{vind+1,1}='Pelvis Velocity';  vind=vind+1;

```

```

varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).PelAmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).PelAmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).PelAmax1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).PelAmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).PelAmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).PelAmin1(:,3)));      vind=vind+1;
varlist{vind+1,1}='Upper Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).USAmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).USAmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).USAmax1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).USAmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).USAmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).USAmin1(:,3)));      vind=vind+1;
varlist{vind+1,1}='Lower Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).LSAmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).LSAmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).LSAmax1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).LSAmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).LSAmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).LSAmin1(:,3)));      vind=vind+1;
varlist{vind+1,1}='Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).SAmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).SAmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).SAmax1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).SAmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,8).SAmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,8).SAmin1(:,3)));      vind=vind+1;
vind=vind+1;
varlist{vind+1,1}='Lateral Bending Trial 2';
varlist{vind+1,1}='Anatomical Level'; varlist{vind+1,3}='Coronal';
varlist{vind+1,4}='Sagittal';      varlist{vind+1,5}='Transverse'; vind=vind+1;
varlist{vind+1,1}='Upper Thorax';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).UTmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).UTmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).UTmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).UTmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).UTmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).UTmin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,8).UTmax2(:,1)-S(1,8).UTmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).UTmax2(:,2)-S(1,8).UTmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).UTmax2(:,3)-S(1,8).UTmin2(:,3)));  vind=vind+1;
varlist{vind+1,1}='Mid Thorax';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).MTmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).MTmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).MTmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).MTmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).MTmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).MTmin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,8).MTmax2(:,1)-S(1,8).MTmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).MTmax2(:,2)-S(1,8).MTmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).MTmax2(:,3)-S(1,8).MTmin2(:,3)));  vind=vind+1;
varlist{vind+1,1}='Pelvis';      vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).Pelmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).Pelmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).Pelmax2(:,3)));      vind=vind+1;

```

```

varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).Pelmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).Pelmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).Pelmin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,8).Pelmax2(:,1)-S(1,8).Pelmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).Pelmax2(:,2)-S(1,8).Pelmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).Pelmax2(:,3)-S(1,8).Pelmin2(:,3))); vind=vind+1;
varlist{vind+1,1}='Upper Spine'; vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).USmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).USmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).USmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).USmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).USmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).USmin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,8).USmax2(:,1)-S(1,8).USmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).USmax2(:,2)-S(1,8).USmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).USmax2(:,3)-S(1,8).USmin2(:,3))); vind=vind+1;
varlist{vind+1,1}='Lower Spine'; vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).LSmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).LSmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).LSmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).LSmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).LSmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).LSmin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,8).LSmax2(:,1)-S(1,8).LSmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).LSmax2(:,2)-S(1,8).LSmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).LSmax2(:,3)-S(1,8).LSmin2(:,3))); vind=vind+1;
varlist{vind+1,1}='Total Spine'; vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).Smax2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).Smax2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).Smax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).Smin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).Smin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).Smin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,8).Smax2(:,1)-S(1,8).Smin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).Smax2(:,2)-S(1,8).Smin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).Smax2(:,3)-S(1,8).Smin2(:,3))); vind=vind+1;
varlist{vind+1,1}='Upper Thorax Velocity'; vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).UTAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).UTAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).UTAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).UTAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).UTAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).UTAmin2(:,3)));      vind=vind+1;
varlist{vind+1,1}='Mid Thorax Velocity'; vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).MTAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).MTAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).MTAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).MTAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).MTAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).MTAmin2(:,3)));      vind=vind+1;
varlist{vind+1,1}='Pelvis Velocity'; vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).PelAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).PelAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).PelAmax2(:,3)));      vind=vind+1;

```

```

varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).PelAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).PelAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).PelAmin2(:,3)));      vind=vind+1;
varlist{vind+1,1}='Upper Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).USAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).USAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).USAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).USAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).USAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).USAmin2(:,3)));      vind=vind+1;
varlist{vind+1,1}='Lower Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).LSAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).LSAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).LSAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).LSAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).LSAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).LSAmin2(:,3)));      vind=vind+1;
varlist{vind+1,1}='Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,8).SAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).SAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).SAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,8).SAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,8).SAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,8).SAmin2(:,3)));      vind=vind+1;
vind=vind+1;
varlist{vind+1,1}='Rotation Trial 1';
varlist{vind+1,1}='Anatomical Level'; varlist{vind+1,3}='Coronal';
varlist{vind+1,4}='Sagittal';      varlist{vind+1,5}='Transverse'; vind=vind+1;
varlist{vind+1,1}='Upper Thorax';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).UTmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).UTmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).UTmax1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).UTmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).UTmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).UTmin1(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,9).UTmax1(:,1)-S(1,9).UTmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).UTmax1(:,2)-S(1,9).UTmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).UTmax1(:,3)-S(1,9).UTmin1(:,3)));  vind=vind+1;
varlist{vind+1,1}='Mid Thorax';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).MTmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).MTmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).MTmax1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).MTmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).MTmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).MTmin1(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,9).MTmax1(:,1)-S(1,9).MTmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).MTmax1(:,2)-S(1,9).MTmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).MTmax1(:,3)-S(1,9).MTmin1(:,3)));  vind=vind+1;
varlist{vind+1,1}='Pelvis';      vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).Pelmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).Pelmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).Pelmax1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).Pelmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).Pelmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).Pelmin1(:,3)));      vind=vind+1;

```

```

varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,9).Pelmax1(:,1)-S(1,9).Pelmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).Pelmax1(:,2)-S(1,9).Pelmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).Pelmax1(:,3)-S(1,9).Pelmin1(:,3))); vind=vind+1;
varlist{vind+1,1}='Upper Spine'; vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).USmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).USmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).USmax1(:,3))); vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).USmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).USmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).USmin1(:,3))); vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,9).USmax1(:,1)-S(1,9).USmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).USmax1(:,2)-S(1,9).USmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).USmax1(:,3)-S(1,9).USmin1(:,3))); vind=vind+1;
varlist{vind+1,1}='Lower Spine'; vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).LSmax1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).LSmax1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).LSmax1(:,3))); vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).LSmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).LSmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).LSmin1(:,3))); vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,9).LSmax1(:,1)-S(1,9).LSmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).LSmax1(:,2)-S(1,9).LSmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).LSmax1(:,3)-S(1,9).LSmin1(:,3))); vind=vind+1;
varlist{vind+1,1}='Total Spine'; vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).Smax1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).Smax1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).Smax1(:,3))); vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).Smin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).Smin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).Smin1(:,3))); vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,9).Smax1(:,1)-S(1,9).Smin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).Smax1(:,2)-S(1,9).Smin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).Smax1(:,3)-S(1,9).Smin1(:,3))); vind=vind+1;
varlist{vind+1,1}='Upper Thorax Velocity'; vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).UTAmx1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).UTAmx1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).UTAmx1(:,3))); vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).UTAmn1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).UTAmn1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).UTAmn1(:,3))); vind=vind+1;
varlist{vind+1,1}='Mid Thorax Velocity'; vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).MTAmx1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).MTAmx1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).MTAmx1(:,3))); vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).MTAmn1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).MTAmn1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).MTAmn1(:,3))); vind=vind+1;
varlist{vind+1,1}='Pelvis Velocity'; vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).PelAmx1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).PelAmx1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).PelAmx1(:,3))); vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).PelAmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).PelAmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).PelAmin1(:,3))); vind=vind+1;
varlist{vind+1,1}='Upper Spine Velocity'; vind=vind+1;

```

```

varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).USAmx1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).USAmx1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).USAmx1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).USAmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).USAmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).USAmin1(:,3)));      vind=vind+1;
varlist{vind+1,1}='Lower Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).LSAmx1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).LSAmx1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).LSAmx1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).LSAmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).LSAmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).LSAmin1(:,3)));      vind=vind+1;
varlist{vind+1,1}='Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).SAmx1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).SAmx1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).SAmx1(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).SAmin1(:,1)));
varlist{vind+1,4}=num2str((S(1,9).SAmin1(:,2)));
varlist{vind+1,5}=num2str((S(1,9).SAmin1(:,3)));      vind=vind+1;
vind=vind+1;
varlist{vind+1,1}='Rotation Trial 2';
varlist{vind+1,1}='Anatomical Level'; varlist{vind+1,3}='Coronal';
varlist{vind+1,4}='Sagittal';      varlist{vind+1,5}='Transverse'; vind=vind+1;
varlist{vind+1,1}='Upper Thorax';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).UTmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).UTmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).UTmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).UTmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).UTmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).UTmin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,9).UTmax2(:,1)-S(1,9).UTmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).UTmax2(:,2)-S(1,9).UTmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).UTmax2(:,3)-S(1,9).UTmin2(:,3)));  vind=vind+1;
varlist{vind+1,1}='Mid Thorax';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).MTmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).MTmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).MTmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).MTmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).MTmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).MTmin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,9).MTmax2(:,1)-S(1,9).MTmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).MTmax2(:,2)-S(1,9).MTmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).MTmax2(:,3)-S(1,9).MTmin2(:,3)));  vind=vind+1;
varlist{vind+1,1}='Pelvis';      vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).Pelmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).Pelmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).Pelmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).Pelmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).Pelmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).Pelmin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,9).Pelmax2(:,1)-S(1,9).Pelmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).Pelmax2(:,2)-S(1,9).Pelmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).Pelmax2(:,3)-S(1,9).Pelmin2(:,3)));  vind=vind+1;
varlist{vind+1,1}='Upper Spine';  vind=vind+1;

```

```

varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).USmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).USmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).USmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).USmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).USmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).USmin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,9).USmax2(:,1)-S(1,9).USmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).USmax2(:,2)-S(1,9).USmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).USmax2(:,3)-S(1,9).USmin2(:,3)));  vind=vind+1;
varlist{vind+1,1}='Lower Spine';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).LSmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).LSmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).LSmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).LSmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).LSmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).LSmin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,9).LSmax2(:,1)-S(1,9).LSmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).LSmax2(:,2)-S(1,9).LSmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).LSmax2(:,3)-S(1,9).LSmin2(:,3)));  vind=vind+1;
varlist{vind+1,1}='Total Spine';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).Smax2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).Smax2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).Smax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).Smin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).Smin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).Smin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,9).Smax2(:,1)-S(1,9).Smin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).Smax2(:,2)-S(1,9).Smin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).Smax2(:,3)-S(1,9).Smin2(:,3)));  vind=vind+1;
varlist{vind+1,1}='Upper Thorax Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).UTAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).UTAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).UTAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).UTAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).UTAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).UTAmin2(:,3)));      vind=vind+1;
varlist{vind+1,1}='Mid Thorax Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).MTAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).MTAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).MTAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).MTAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).MTAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).MTAmin2(:,3)));      vind=vind+1;
varlist{vind+1,1}='Pelvis Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).PelAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).PelAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).PelAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).PelAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).PelAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).PelAmin2(:,3)));      vind=vind+1;
varlist{vind+1,1}='Upper Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).USAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).USAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).USAmax2(:,3)));      vind=vind+1;

```



```

varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).USAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).USAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).USAmin2(:,3)));      vind=vind+1;
varlist{vind+1,1}='Lower Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).LSAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).LSAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).LSAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).LSAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).LSAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).LSAmin2(:,3)));      vind=vind+1;
varlist{vind+1,1}='Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,9).SAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).SAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).SAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,9).SAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,9).SAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,9).SAmin2(:,3)));      vind=vind+1;
vind=vind+1;
varlist{vind+1,1}='Flx/Ext Trial 2';
varlist{vind+1,1}='Anatomical Level'; varlist{vind+1,3}='Coronal';
varlist{vind+1,4}='Sagittal';      varlist{vind+1,5}='Transverse'; vind=vind+1;
varlist{vind+1,1}='Upper Thorax';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).UTmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).UTmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).UTmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).UTmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).UTmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).UTmin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,10).UTmax2(:,1)-S(1,10).UTmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).UTmax2(:,2)-S(1,10).UTmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).UTmax2(:,3)-S(1,10).UTmin2(:,3)));  vind=vind+1;
varlist{vind+1,1}='Mid Thorax';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).MTmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).MTmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).MTmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).MTmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).MTmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).MTmin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,10).MTmax2(:,1)-S(1,10).MTmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).MTmax2(:,2)-S(1,10).MTmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).MTmax2(:,3)-S(1,10).MTmin2(:,3)));  vind=vind+1;
varlist{vind+1,1}='Pelvis';      vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).Pelmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).Pelmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).Pelmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).Pelmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).Pelmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).Pelmin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,10).Pelmax2(:,1)-S(1,10).Pelmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).Pelmax2(:,2)-S(1,10).Pelmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).Pelmax2(:,3)-S(1,10).Pelmin2(:,3)));  vind=vind+1;
varlist{vind+1,1}='Upper Spine';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).USmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).USmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).USmax2(:,3)));      vind=vind+1;

```

```

varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).USmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).USmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).USmin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,10).USmax2(:,1)-S(1,10).USmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).USmax2(:,2)-S(1,10).USmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).USmax2(:,3)-S(1,10).USmin2(:,3)));  vind=vind+1;
varlist{vind+1,1}='Lower Spine';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).LSmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).LSmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).LSmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).LSmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).LSmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).LSmin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,10).LSmax2(:,1)-S(1,10).LSmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).LSmax2(:,2)-S(1,10).LSmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).LSmax2(:,3)-S(1,10).LSmin2(:,3)));  vind=vind+1;
varlist{vind+1,1}='Total Spine';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).Smax2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).Smax2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).Smax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).Smin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).Smin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).Smin2(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,10).Smax2(:,1)-S(1,10).Smin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).Smax2(:,2)-S(1,10).Smin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).Smax2(:,3)-S(1,10).Smin2(:,3)));  vind=vind+1;
varlist{vind+1,1}='Upper Thorax Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).UTAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).UTAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).UTAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).UTAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).UTAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).UTAmin2(:,3)));      vind=vind+1;
varlist{vind+1,1}='Mid Thorax Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).MTAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).MTAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).MTAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).MTAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).MTAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).MTAmin2(:,3)));      vind=vind+1;
varlist{vind+1,1}='Pelvis Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).PelAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).PelAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).PelAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).PelAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).PelAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).PelAmin2(:,3)));      vind=vind+1;
varlist{vind+1,1}='Upper Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).USAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).USAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).USAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).USAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).USAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).USAmin2(:,3)));      vind=vind+1;
varlist{vind+1,1}='Lower Spine Velocity';  vind=vind+1;

```

```

varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).LSAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).LSAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).LSAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).LSAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).LSAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).LSAmin2(:,3)));      vind=vind+1;
varlist{vind+1,1}='Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).SAmax2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).SAmax2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).SAmax2(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).SAmin2(:,1)));
varlist{vind+1,4}=num2str((S(1,10).SAmin2(:,2)));
varlist{vind+1,5}=num2str((S(1,10).SAmin2(:,3)));      vind=vind+1;
vind=vind+1;
varlist{vind+1,1}='Flx/Ext Trial 3';
varlist{vind+1,1}='Anatomical Level'; varlist{vind+1,3}='Coronal';
varlist{vind+1,4}='Sagittal';      varlist{vind+1,5}='Transverse'; vind=vind+1;
varlist{vind+1,1}='Upper Thorax';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).UTmax3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).UTmax3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).UTmax3(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).UTmin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).UTmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).UTmin3(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,10).UTmax3(:,1)-S(1,10).UTmin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).UTmax3(:,2)-S(1,10).UTmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).UTmax3(:,3)-S(1,10).UTmin3(:,3)));  vind=vind+1;
varlist{vind+1,1}='Mid Thorax';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).MTmax3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).MTmax3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).MTmax3(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).MTmin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).MTmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).MTmin3(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,10).MTmax3(:,1)-
S(1,10).MTmin3(:,1)));  varlist{vind+1,4}=num2str((S(1,10).MTmax3(:,2)-S(1,10).MTmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).MTmax3(:,3)-S(1,10).MTmin3(:,3)));  vind=vind+1;
varlist{vind+1,1}='Pelvis';      vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).Pelmax3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).Pelmax3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).Pelmax3(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).Pelmin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).Pelmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).Pelmin3(:,3)));      vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,10).Pelmax3(:,1)-S(1,10).Pelmin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).Pelmax3(:,2)-S(1,10).Pelmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).Pelmax3(:,3)-S(1,10).Pelmin3(:,3)));  vind=vind+1;
varlist{vind+1,1}='Upper Spine';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).USmax3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).USmax3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).USmax3(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).USmin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).USmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).USmin3(:,3)));      vind=vind+1;

```

```

varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,10).USmax3(:,1)-S(1,10).USmin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).USmax3(:,2)-S(1,10).USmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).USmax3(:,3)-S(1,10).USmin3(:,3)));  vind=vind+1;
varlist{vind+1,1}='Lower Spine';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).LSmax3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).LSmax3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).LSmax3(:,3)));          vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).LSmin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).LSmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).LSmin3(:,3)));          vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,10).LSmax3(:,1)-S(1,10).LSmin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).LSmax3(:,2)-S(1,10).LSmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).LSmax3(:,3)-S(1,10).LSmin3(:,3)));  vind=vind+1;
varlist{vind+1,1}='Total Spine';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).Smax3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).Smax3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).Smax3(:,3)));          vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).Smin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).Smin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).Smin3(:,3)));          vind=vind+1;
varlist{vind+1,2}='ROM';      varlist{vind+1,3}=num2str((S(1,10).Smax3(:,1)-S(1,10).Smin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).Smax3(:,2)-S(1,10).Smin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).Smax3(:,3)-S(1,10).Smin3(:,3)));  vind=vind+1;
varlist{vind+1,1}='Upper Thorax Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).UTAmax3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).UTAmax3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).UTAmax3(:,3)));          vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).UTAmin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).UTAmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).UTAmin3(:,3)));          vind=vind+1;
varlist{vind+1,1}='Mid Thorax Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).MTAmax3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).MTAmax3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).MTAmax3(:,3)));          vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).MTAmin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).MTAmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).MTAmin3(:,3)));          vind=vind+1;
varlist{vind+1,1}='Pelvis Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).PelAmax3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).PelAmax3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).PelAmax3(:,3)));          vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).PelAmin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).PelAmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).PelAmin3(:,3)));          vind=vind+1;
varlist{vind+1,1}='Upper Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).USAmax3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).USAmax3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).USAmax3(:,3)));          vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).USAmin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).USAmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).USAmin3(:,3)));          vind=vind+1;
varlist{vind+1,1}='Lower Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).LSAmax3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).LSAmax3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).LSAmax3(:,3)));          vind=vind+1;

```

```

varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).LSAmin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).LSAmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).LSAmin3(:,3)));      vind=vind+1;
varlist{vind+1,1}='Spine Velocity';  vind=vind+1;
varlist{vind+1,2}='Max';      varlist{vind+1,3}=num2str((S(1,10).SAmax3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).SAmax3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).SAmax3(:,3)));      vind=vind+1;
varlist{vind+1,2}='Min';      varlist{vind+1,3}=num2str((S(1,10).SAmin3(:,1)));
varlist{vind+1,4}=num2str((S(1,10).SAmin3(:,2)));
varlist{vind+1,5}=num2str((S(1,10).SAmin3(:,3)));      vind=vind+1;
vind=vind+1;

```

```

warning off MATLAB:xlswrite:AddSheet;
xlswrite([cpath,num2str(patnum)],[DynamicSpine.xlsx'],varlist,[num2str(patnum)],[A3]);

```

Assemble

```

dpath = input('Enter the path of the patients folder (ex. "C:\\CCMC\\Little League Data\\lres\\temp\\") :
','s');
ffile=dir([dpath,'*.mat']);
for ffile_index=1:length(ffile)
    load([dpath,ffile(ffile_index).name]);
    patname{ffile_index}=ffile(ffile_index).name(1:(min(findstr(ffile(1).name,'_'))-1));
    patname{ffile_index}
    trials_avg = str2num(input('Enter the trial numbers you want to analyze seperated by spaces: ','s'));
    ntrials_avg = length(trials_avg);
    assemblespine
end
for aaa=1:3
    % Angles
    for j=1:size(MUthor,1)
        AmUthor(j,aaa)=mean(MUthor(j,aaa,:));
        AmMthor(j,aaa)=mean(MMthor(j,aaa,:));
        AmPelvis(j,aaa)=mean(MPelvis(j,aaa,:));
        AmUSpine(j,aaa)=mean(MUSpine(j,aaa,:));
        AmLSpine(j,aaa)=mean(MLSpine(j,aaa,:));
        AmSpine(j,aaa)=mean(MSpine(j,aaa,:));
    end
end

```

Assemblespine

```

for zzz=1:ntrials_avg
    AUthor(1:100,1:3,zzz,ffile_index)=S(trials_avg(zzz)).nUThor2(1:100,1:3);
    AMthor(1:100,1:3,zzz,ffile_index)=S(trials_avg(zzz)).nMThor2(1:100,1:3);
    APelvis(1:100,1:3,zzz,ffile_index)=S(trials_avg(zzz)).nPelvis2(1:100,1:3);
    AUSpine(1:100,1:3,zzz,ffile_index)=S(trials_avg(zzz)).nUSpine2(1:100,1:3);
    ALSpine(1:100,1:3,zzz,ffile_index)=S(trials_avg(zzz)).nLSpine2(1:100,1:3);
    ASpine(1:100,1:3,zzz,ffile_index)=S(trials_avg(zzz)).nSpine2(1:100,1:3);
end
dpath = input('Enter the path of the patients folder (ex. "C:\\CCMC\\Little League Data\\lres\\temp\\") :
','s');
ffile=dir([dpath,'*.mat']);
for ffile_index=1:length(ffile)
    load([dpath,ffile(ffile_index).name]);
    patname{ffile_index}=ffile(ffile_index).name(1:(min(findstr(ffile(1).name,'_'))-1));

```

```

patname{ffile_index}
trials_avg = str2num(input('Enter the trial numbers you want to analyze seperated by spaces: ','s'));
ntrials_avg = length(trials_avg);
assemblespine
end
for aaa=1:3
    % Angles
    for j=1:size(MUthor,1)
        AmUthor(j,aaa)=mean(MUthor(j,aaa,:));
        AmMthor(j,aaa)=mean(MMthor(j,aaa,:));
        APelvis(j,aaa)=mean(MPelvis(j,aaa,:));
        AmUSpine(j,aaa)=mean(MUSpine(j,aaa,:));
        AmLSpine(j,aaa)=mean(MLSpine(j,aaa,:));
        AmSpine(j,aaa)=mean(MSpine(j,aaa,:));
    end
end
for aaa=1:3
    % Angles
    for j=1:size(AUthor,1)
        MUthor(j,aaa,ffile_index)=mean(AUthor(j,aaa,:,ffile_index));
        MMthor(j,aaa,ffile_index)=mean(AMthor(j,aaa,:,ffile_index));
        MPelvis(j,aaa,ffile_index)=mean(APelvis(j,aaa,:,ffile_index));
        MUSpine(j,aaa,ffile_index)=mean(AUSpine(j,aaa,:,ffile_index));
        MLSpine(j,aaa,ffile_index)=mean(ALSpine(j,aaa,:,ffile_index));
        MSpine(j,aaa,ffile_index)=mean(ASpine(j,aaa,:,ffile_index));
    end
end
end

```