2013

A Robotic Neuro-Musculoskeletal Simulator for Spine Research

Robb W. Colbrunn
Cleveland State University

A ROBOTIC NEURO-MUSCULOSKELETAL SIMULATOR FOR SPINE RESEARCH

ROBB W. COLBRUNN

Bachelor of Science in Mechanical Engineering

Grove City College

May 1996

Master of Science in Mechanical Engineering

Case Western Reserve University

May 2000

submitted in partial fulfillment of requirement for the degree

DOCTOR OF ENGINEERING IN APPLIED BIOMEDICAL ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

APRIL 2013

This dissertation has been approved

for the Department of CHEMICAL AND BIOMEDICAL ENGINEERING

and the College of Graduate Studies by

_____

Dissertation Chairperson, Robert McLain M.D.

_____

Department & Date

_____

Antonie van den Bogert Ph.D.

_____

Department & Date

_____

Lars Gilbertson Ph.D.

_____

Department & Date

_____

George Chatzimavroudis Ph.D.

_____

Department & Date

_____

Dan Simon Ph.D.

_____

Department & Date

_____

Michael Hammonds Ph.D.

_____

Department & Date

ACKNOWLEDGEMENTS

I am deeply indebted to the many folks who made this dissertation possible. Chronologically is probably the best way to explain it. Ton van den Bogert was the first to plant the idea in my head back in 2005 that going back to school to pursue a doctorate would be a worthwhile venture, though I did not take the plunge until the fall of 2008. Along the way, his guidance and insights as my supervisor at the Cleveland Clinic served as a very influential part of my development as a scientist/engineer. Many of his ideas still reside at the core of the software, systems, and mission of the Cleveland Clinic BioRobotics Lab. One small example is that the look-up-table that drives the compensatory robot motion presented in this work was an idea Ton had provided.

He was joined by Peter Cavanagh, BME department chair at the time, in being a strong influence for the direction of the type of work we pursued. Shortly after I arrived at the Clinic, Peter laid out a vision for the BioRobotics Lab to be the world leader for how this type of testing is done. Though we have not arrived, we are certainly working in this direction. I am extremely grateful for his work ethic, leadership, and influence as a role model.

Lars Gilbertson was the man with the idea and vision for the creation of the Neuro-Musculoskeletal Simulator. As I was looking for projects, this was an idea that matched my strengths, and I even recall Ton commenting that this idea had very exciting implications. Lars' calm and patient demeanor, coupled with his four dimensional thinking and genuine supportiveness, was a wonderful environment for me to be

mentored and incubate many of the spine biomechanical lessons learned in this work. His servant leadership style is apparent, and it is a blessing to have been given this opportunity to work with him. I should also mention that much of the financial support for this work came from the Cleveland Clinic Spine Research Laboratory that Lars was the director of at the time. Additional funding came from the Cleveland Clinic Stanley Zielony Spinal Surgery Research and Education Fund and through grants I was able to write in support of my training. I received a Cleveland Clinic Research Program Committee grant (CCF RPC Grant: 2012-1003) and a Cleveland State Doctoral Dissertation Research Expense Award (DDREAFP Grant: 508081307-0001).

As Ton and Lars were called on to other career ventures, Rob McLain agreed to serve as my advisor. Even before then, I had the opportunity to work with him as he was the first spine surgeon to fully engage in the development and use of the robotic Spine Musculoskeletal Simulator in the Spine Research Lab. He has seen the potential for the uses and development of these technologies and is very willing to offer clinical insight to keep us engineers pointed in the right direction. He is encouraging, supportive, and passionately interested in these topics, and I look forward to continued developments and clinical insights that will come.

My hardest course at Cleveland State was Optimal Controls taught by Dan Simon. His high standards and high expectations are admirable and something I aspire to. This was one of the primary reasons I sought his presence on this committee. As iron sharpens iron, so one person sharpens another.

Once I decided to pursue Lars' idea for the Neuro-Musculoskeletal Simulator, I enrolled in a neuroscience course taught by Mike Hammonds. Dr. Hammonds, among other strengths, is an educator. He has a real passion for helping people learn, and since I have the habit of wanting to understand everything, he was a great mentor to develop my understanding in neuroscience.

George Chatzimavroudis, in addition to serving on my committee, offered me wise council for realistic expectations in both the beginning and ending of my time at Cleveland State. He has also helped me navigate the proper channels for the University. This guidance was much appreciated.

I have also been fortunate to have the guidance and advice of several, what I would consider, non-official members of my committee. These are folks who have been sounding posts, collaborators, mentors, and colleagues: Kathe Derwin, Ahmet Erdemir, and Jason Halloran. I am always learning something new from them, and their influence on this dissertation, whether they know it or not, is tangible.

Two others comprise a pair of humble, insightful, and dedicated driving forces behind the completion of this work: Prasath Mageswaran and Tara Bonner. This would not have been possible without their hard work, long hours, and dedication to helping me complete it. Literally. Not possible. Software development, specimen testing, brainstorming, telling me why my ideas were lousy, and doing anything else that needed

done while still not losing their cheery attitudes was such a huge benefit. I am proud to call them my colleagues.

The Spine Research Laboratory was instrumental in making this possible. First, under the leadership of Lars Gilbertson, then under Adam Bartsch, their support is greatly appreciated. The support of Brian Perse and Ed Benzel is also gratefully acknowledged.

I would also like to acknowledge the folks in the Cleveland Clinic Research Core Services and Medical Device Solutions groups including Karl West and Jeanne Ineman for giving me the unflinching support and flexibility to pursue these goals.

I appreciate Sam Butler's support with making sure the statistics were sound.

I would like to acknowledge several individuals, organizations, and journals who have granted me permission to reprint their images in this dissertation.

- University of Maryland Medical Center: Figure 4 ©2007
- Medical Multimedia Group LLC, www.eOrthopod.com: Figure 4 ©2002
- Vertical Health LLC: Figure 5 ©2010
- Mayo Clinic: Figure 7c ©2011
- William Ledoux Ph.D., VA Center of Excellence for Limb Loss Prevention and Prosthetic Engineering: Seattle, WA: Figure 8a ©2009

- Savio L-Y. Woo Ph.D., Musculoskeletal Research Center Director: Department of BioEngineering: Swanson School of Engineering: University of Pittsburgh: Figure 8b ©2010
- Richard Debski Ph.D, Orthopaedic Robotics Laboratory Co-Director: Departments of Orthopaedic Surgery and Bioengineering: Swanson School of Engineering: University of Pittsburgh: Figure 8c ©2004
- Journal of Biomechanics: Figures 9 ©2000, 11g ©2002, 11i ©2010:
- Spine: Figures 10 ©2006, 11a ©1997, 11b ©2008, 11f ©2007
- Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine: Figure 11c ©2007
- Operative Techniques in Orthopaedics: Figure 11d ©2000
- MTS Systems Corporation: Figure 11e ©2006
- European Spine Journal: Figure 11h ©1994
- John Mooradd, Tekscan Inc. Figure 48a ©2013

I am an engineer and we are not known for our prowess with the English language. I would like to thank Joni Colbrunn, George Churm, and Sarah Bianchi with their help with proofreading and editing this document.

I would especially like to thank my kids Boyd, Lilia, Gabe, Troy, and Adalai for their patience, understanding, and sacrifice as Dad spent many hours away from home trying to finish "school." I am humbled and so fortunate to get to be their father.

I am even more fortunate to have had the support of my loving wife, Joni Colbrunn. She worked significantly harder than I did through these past 5 years ($p < 0.001$). While I was off at class sitting down in a chair listening to someone talk, she was tending to the needs of a house and 5 kids, including young twins. Later on, while I was spending long nights in the lab sitting in a chair programming and listening to music on the radio, she was getting the same rambunctious kids to do their homework and off to all the events of their busy social calendars. In the immortal words of Tara, "Joni is a saint!" There is not enough room here to describe how grateful I am. I am unable to describe how humbled and lucky I am to have married such a loving, supportive, and wonderful woman.

Lastly, I want to thank God for life and the ability we have to study it.

A ROBOTIC NEURO-MUSCULOSKELETAL SIMULATOR FOR SPINE RESEARCH

ROBB W. COLBRUNN

ABSTRACT

An influential conceptual framework advanced by Panjabi represents the living spine as a complex neuromusculoskeletal system whose biomechanical functioning is rather finely dependent upon the interactions among and between three principal subsystems: the passive musculoskeletal subsystem (osteoligamentous spine plus passive mechanical contributions of the muscles), the active musculoskeletal subsystem (muscles and tendons), and the neural and feedback subsystem (neural control centers and feedback elements such as mechanoreceptors located in the soft tissues) [1]. The interplay between subsystems readily encourages "thought experiments" of how pathologic changes in one subsystem might influence another—for example, prompting one to speculate how painful arthritic changes in the facet joints might affect the neuromuscular control of spinal movement.

To answer clinical questions regarding the interplay between these subsystems the proper experimental tools and techniques are required. Traditional spine biomechanical experiments are able to provide comprehensive characterization of the structural properties of the osteoligamentous spine. However, these technologies do not incorporate a simulated neural feedback from neural elements, such as mechanoreceptors and nociceptors, into the control loop. Doing so enables the study of how this feedback—including pain-related— alters spinal loading and motion patterns. The first

such development of this technology was successfully completed in this study and constitutes a Neuro-Musculoskeletal Simulator. A Neuro-Musculoskeletal Simulator has the potential to reduce the gap between bench and bedside by creating a new paradigm in estimating the outcome of spine pathologies or surgeries. The traditional paradigm is unable to estimate pain and is also unable to determine how the treatment, combined with the natural pain avoidance of the patient, would transfer the load to other structures and potentially increase the risk for other problems.

The novel Neuro-Musculoskeletal Simulator described in this work has demonstrated, through simulation and cadaveric experimentation, that it is able to incorporate data from external sensors (e.g. force, motion tracking) to modulate spine biomechanical responses. In addition, the Neuro-Musculoskeletal Simulator exhibited the ability to use an estimated nociceptive response in unilateral facet arthritis to elucidate statistically significant compensatory kinetic and kinematic changes. These changes included a 37% increase in spine shear force, and an 18% increase in applied spine torque.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The spine is a complex system which can be broken down into three subsystems of function: Passive (bone, ligament, and intervertebral discs), Active (muscles), and Control (neural components) (Figure 1). These subsystems work together to provide mobility, protect the spinal cord, and bear loads. Our understanding of the interaction of these three subsystems is limited at best. Pioneers in the field, such as Panjabi, have developed methods to apply pure moments to a single axis of the spine to elucidate the mechanical properties of the spine. The application of those concepts continues to be applied with custom loading frames [2-13], custom robotics systems [14-22], and the adaptation of commercial robotic technology [23-33]. With these systems and pure moment testing, spinal biomechanics variables such as the neutral zone and range of motion can be determined. While all these systems continue to provide benefit, the current testing methodologies are based on the principle of using the robot or loading frame to perform as a multi-axis material testing system. To reduce the gap between

bench and bedside, a new paradigm in estimating the outcome of spinal pathologies or surgeries was required.

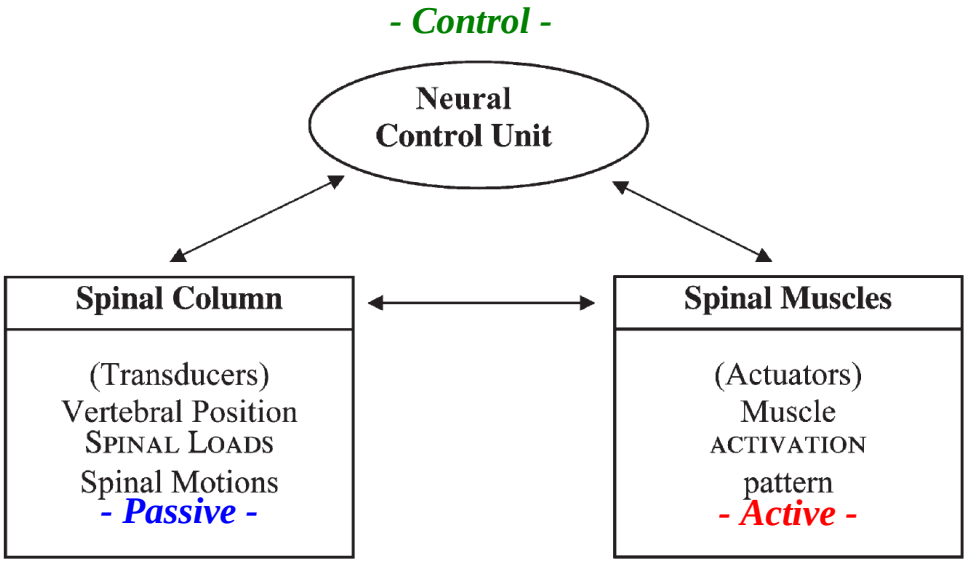


Figure 1: Panjabi's Spinal Stability Model

Elucidation of the load-motion characteristics of the passive, osteoligamentous spine has been the subject of numerous *in vitro* biomechanical experiments and typically involves the use of a testing system to apply loads (i.e., forces and/or moments) to a specimen while measuring resulting deformations (vertebral translations and rotations). Until recently, a technical limitation of these kinds of *in vitro* experiments has been that it is not possible to apply large compressive forces to a lengthy specimen without it buckling, whereas in-vivo the osteoligamentous spine routinely withstands large forces without buckling. This highlights the importance of the muscles in providing mechanical stability. Patwardhan [11] has advanced the concept and technology of a "follower load" which enables load-motion characteristics of the osteoligamentous spine to be studied under large compressive loads; supplementary investigations are underway to determine

whether the “follower load” accurately reflects in-vivo muscle forces and neuromuscular control strategies.

While the above body of experimental work resulted in a comprehensive characterization of the structural properties of the osteoligamentous spine and the technology is evolving to the point where the effects of muscle forces and some aspects of neuromuscular control can be simulated, current technology had yet to incorporate neural elements, such as mechanoreceptors and nociceptors. This would enable the study of how neural feedback—including pain-related— alters spinal loading and motion patterns. The first such development of this technology was successfully completed and constitutes a Neuro-Musculoskeletal Simulator (NMS). The NMS has the potential to reduce the gap between bench and bedside by creating a new paradigm in estimating the outcome of spine pathologies or surgeries. The traditional paradigm is unable to estimate pain. In addition, it is also unable to determine how the treatment, combined with the natural pain avoidance of the patient, would transfer the load to other structures and potentially increase the risk for other problems. The NMS was developed to perform testing on cadaveric spines and use estimates of nociceptive responses to predict corresponding changes in kinematics and kinetics. In addition, other sensors were used to monitor loading at “pain free” locations to see what new and potentially noxious loading occurred as a result.

The hypotheses going into this study were

(1) a neuro-musculoskeletal simulator can be developed to perform motion and loading simulations on cadaveric spines, and

(2) this simulator can be effectively used to estimate a nociceptive response and predict a modulated motion that results in alternative, potentially harmful, loading conditions to the spine.

For testing these hypotheses, we proposed the following Specific Aims:

Aim 1: To develop a robotic neuro-musculoskeletal simulator for cadaveric spine testing which incorporates data from external sensors (e.g. pressure, force, motion tracking) to modulate spinal motion and loading patterns.

Aim 2: To demonstrate the ability of the neuro-musculoskeletal simulator to use an estimated nociceptive response in unilateral facet arthritis to modulate the motion pattern and elucidate subsequent injurious loading conditions that could lead to contralateral arthritis.

Testing hypothesis (1) was accomplished through the demonstration of multi-axis loading of cadaveric spines. The system accepted inputs of kinetic and kinematic trajectories and applied them to the specimen in a controlled manner. The modification/creation and architecture of the Universal Musculoskeletal Simulator (UMS) software outlined in chapter III serve as the explanation of the system to achieve this goal. Hypothesis (2) considered a generic system design, but required it to be tested with a specific clinical question. In this case, cervical facet joint arthritis was selected

with the intent to see if, and how, the neural feedback resulted in any changes in kinetics or kinematics. Also of interest was to see if these neuro-muscular control schemes have the potential to predict the onset of other pathologies. The assumption was made that facet joint force is an analog for pain in a facet joint with osteoarthritis. Figure 2 shows how a sensor was placed in the spine to measure facet contact force.

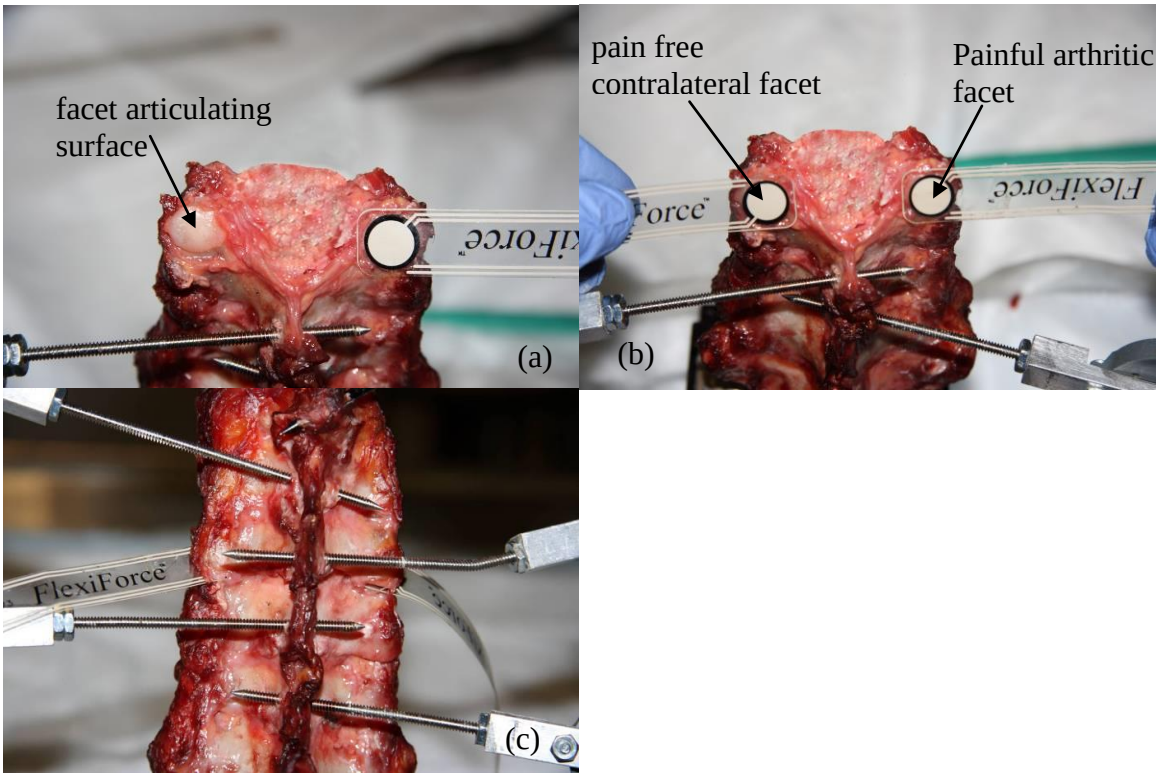


Figure 2: (a) Single force sensor over the C5 articulating surface of a facet joint. (b) "Painful" and "Pain Free" contralateral sensors. (c) Sensors installed at C4-C5 level of an intact cervical spine.

Chapter IV provides a detailed review of the techniques employed to use the sensor output to drive pain-modulated motion. These motions represent potential compensatory kinetic and kinematic solutions to achieve high level task objectives (e.g. turn your head and look in a certain direction) while compensating for pain. Control system creation, surrogate model development, and simulations of the control system are

included. Chapter V and VI include the methods and results for the cadaveric experimentation. Finally, Chapter VII presents the implications, limitations, and future directions for this new technology.

CHAPTER II

BACKGROUND

Section 2.01 Anatomy of the Passive Osteoligamentous Spine

When discussing spine pain and motion, it is worth providing a brief overview of the spine anatomy for reference. There are 5 sections to the spine shown in Figure 3a: Cervical, Thoracic, Lumbar, Sacrum, and Coccyx. The three major sections, Cervical, Thoracic, and Lumbar, are groups of vertebrae consisting of the vertebral body, pedicles, and posterior arch. Load is transferred from vertebra to vertebra via the intervertebral discs, ligaments, and facet joints (Fig. 3b,c).

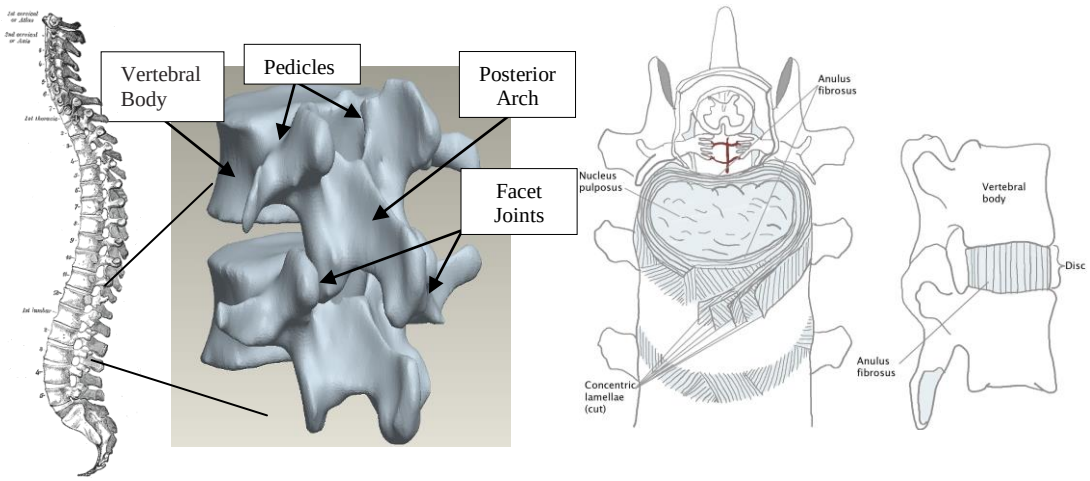


Figure 3: (a) Overview of Spinal Sections [34], (b) Vertebrae and (c) Disc Anatomy

The intervertebral disc has a periphery called the Anulus Fibrosus which is a laminated matrix of fibrous tissue and fibrocartilage. In the center of this circumferential band is the Nucleus Pulposus which is a network of collagenous fibers in a mucoprotein gel rich in polysaccharide. The gelatinous substance serves to hydrostatically distribute the load between adjacent vertebral bodies.

Each adjacent pair of vertebrae has an interlocking set of synovial joints called facet joints. They are located on the dorsal part of the spine with their cartilage covered joint surfaces surrounded by a capsule. The facet joints assist the disc in allowing certain relative motions of the adjacent vertebra while restricting others.

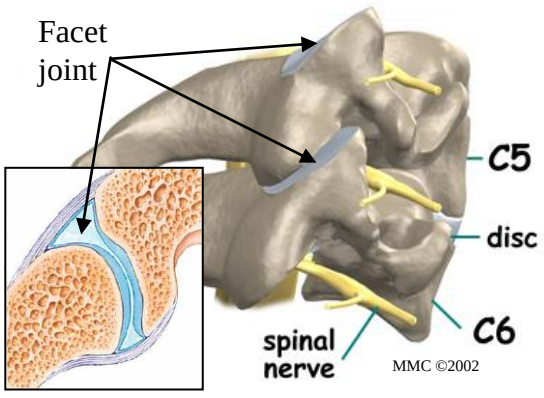


Figure 4: Cervical Facet Joints

The osseous spine is woven together with a series of ligaments that provide flexibility, passive stability, and prevent excessive movement. Ligaments such as the posterior and anterior longitudinal ligaments span the entire length of the spine and attach at each bone. Other ligaments, such as the facet capsulary ligament, span short distances to impart localized stability.

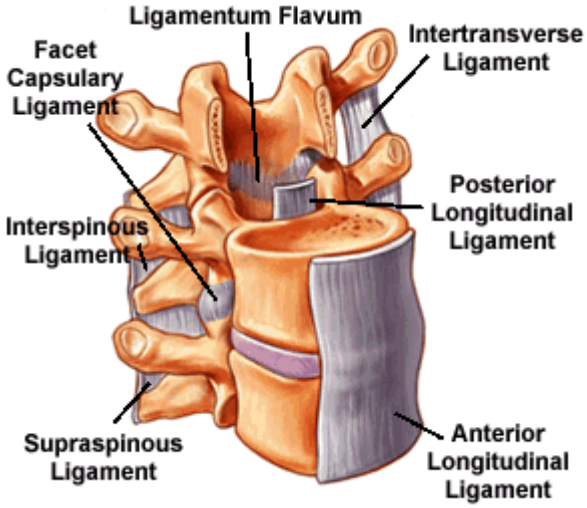


Figure 5: Ligaments of the spine

Ligaments are not only structural in purpose, but they also serve as sensors. Embedded within the ligaments are small mechanoreceptor nerve cells which provide ligament stretch information that is ultimately used in proprioceptive and nociceptive feedback to assist in muscle control.

Section 2.02 Anatomy of the Spinal Muscles and Tendons

The spinal muscular system is complex. In just the cervical spine alone, there are over 18 muscles or muscle groups to assist with head and neck movement as well as support and stabilize the cervical spine [35]. Through a system of contraction, co-contraction, and balancing against gravity, the muscles can impart this functionality. Each muscle is connected to the osteoligamentous spine via a tendon and is innervated and controlled in a way that provides conscious control of motion and subconscious dynamic stabilization.

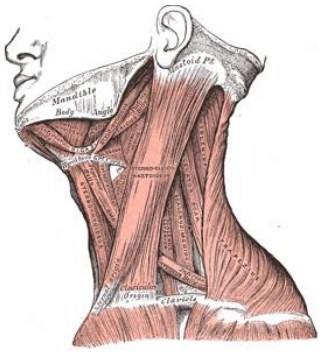


Figure 6: Muscles of the neck. Lateral view [34]

Table I: Muscles of the Cervical Spine

CERVICAL MUSCLES	FUNCTION
Sternocleidomastoid	Extends & rotates head, flexes vertebral column
Scalenus	Flexes & rotates neck
Spinalis Cervicis	Extends & rotates head
Spinalis Capitis	Extends & rotates head
Semispinalis Cervicis	Extends & rotates vertebral column
Semispinalis Capitis	Rotates head & pulls backward
Splenius Cervicis	Extends vertebral column
Longus Colli Cervicis	Flexes cervical vertebrae
Longus Capitus	Flexes head
Rectus Capitus Anterior	Flexes head
Rectus Capitus Lateralis	Bends head laterally
Iliocostalis Cervicis	Extends cervical vertebrae
Longissimus Cervicis	Extends cervical vertebrae
Longissimus Capitis	Rotates head & pulls backward
Rectus Capitis Posterior Major	Extends & rotates head
Rectus Capitis Posterior Minor	Extends head
Obliquus Capitis Inferior	Rotates atlas
Obliquus Capitis Superior	Extends & bends head laterally

Section 2.03 Neural Anatomy & Neuromuscular Control of Spinal Movement

In addition to the structural purposes of the osteoligamentous spine, another vital function is to protect the neural structures that pass through its core. The spinal cord descends from the brain and branches bilaterally between each vertebra via dorsal (sensory information in) and ventral (motor control out) roots. Some of these branches in the cervical region serve as conduits for sensory information and motor control of the

upper extremities. There are also branches that don't travel as far which serve the same purpose for head and neck motion.

From the sensory perspective, there are sensor cells embedded in the vertebral discs, ligaments, facet capsules, tendons, and muscles. The sensor cells of interest for the purpose of this study are mechanoreceptors. These types of sensor cells fire action potentials when they are mechanically distorted in some manner. They can be used to sense touch, pressure, stretch, vibration, and pain. These signals typically ascend to the central nervous system (CNS) via the dorsal columns/medial lemniscus, anterolateral system/spinothalamic tracts, and spinocerebellar tracts. Nociceptors are the class of sensory cells that communicate painful stimuli. These can take the form of chemoreceptors, thermoreceptors, and mechanoreceptors. It is the nociceptive type of mechanoreceptor that we are most interested in for this study and they are most commonly free nerve endings. Ligaments, intervertebral discs, subchondral bone, nerve roots, muscles, and facet joints have all been shown to contain free nerve endings [36-37]. Information from a field of mechanoreceptors can be localized information from each receptor or combined as a network and aggregated by the CNS to provide a more global picture of the response. When this occurs with the other modalities of mechanoreceptors (stretch, pressure, etc.), the CNS gains the ability to sense the position, location, orientation, and movement of the body and its parts. This global sense is considered proprioceptive feedback.

From the motor control perspective, these neurons synapse on individual motor units for each muscle involved. If we assume an average of about 100 motor units per muscle and 18 muscles controlling the head and cervical spine, that yields a total of 1800 degrees of freedom (DOF) for the control inputs to the system. These 1800 possible different inputs can be applied to the osteoligamentous system that has at most 36 degrees of freedom (6-DOF per vertebral pair) for possible motions. Clearly this is an indeterminate system, and this quick analysis suggests that human movement is very complex and cannot be adequately described in this short paragraph or fully replicated by the NMS. Many researchers have spent significant amounts of time and energy trying to develop motor control models of voluntary motion [38-39]. However, the general principle in these models is that motion is intended, sensory data is read (from many inputs including vision, etc.), these signals are combined and processed by the CNS (including spinal cord reflexes), and then the signals are sent as muscle activations. These muscle activations create motion and loading and thus change the output of the sensors involved. The cycle continues until the desired outcome of the intended motion is met. This same principle of intended motion being modulated by multiple sensor inputs is designed into the control architecture of the NMS.

Section 2.04 Pathology

The osteoligamentous, muscular, and neural systems all work together for their intended purposes. However, if one or more of these systems becomes degraded (aging, degeneration, trauma, surgery, etc.), the whole system will suffer. For example, degenerative disc disease can lead to a reduced disc height which places pressure on the

dorsal and/or ventral roots which can lead to numbness or pain. The degeneration process itself can be a vicious cycle due to the fact that mechanoreceptors in the disc are more prevalent in degenerated discs than in pain free patients with scoliosis [40]. Brisby [36] considers that even though disc degeneration is commonly considered a cause of low-back pain, the actual pain source may be from any spinal tissue that contains free nerve endings. Chronic or traumatic injury to the ligaments can also lead to inflammation and back pain. A strained ligament can place pressure on the nerves and cause numbness. Loss of ligamentous stability may require more, or different, work by the muscles to compensate. These different loading conditions can lead to other secondary complications. Fusion surgeries and osteoporosis are a few other common conditions that can also change the statics of the system in such a way that different loading conditions can potentially initiate complications in another system.

Figure 7: Examples of spinal pathologies (a) whiplash injury (b) damage to the passive structures of the spine (c) osteoporosis causing loss in vertebra height as well as creating new and undesirable loading conditions.

Section 2.05 Neuro-musculoskeletal Modeling

Neuro-musculoskeletal modeling is an *in silico* technique of developing computational models for under-constrained neuro-musculoskeletal systems and applying various motor control algorithms in order to elucidate underlying mechanisms for

processing and motor control. These tools can be very beneficial in studying the mechanisms of motor learning as well as pathologic conditions of the neuro-musculoskeletal system caused by injury (stroke, peripheral nerve or spinal cord injury, and amputation) [41-43]. These tools also serve as platforms for research into how sensory feedback can affect selections of adaptive motor strategies. The NMS is the first system to show that these types of investigations can also occur in an *in vitro* setting . The speed and flexibility of the *in silico* testing to control variables and iterate is not something that the NMS is able to replicate. However, it provides a platform by which some of these models can be validated. It is important to understand that the question is not whether one should choose *in silico* modeling or *in vitro* testing. Each tool provides insights that the other cannot. Where there is overlap, both should be used for validation purposes.

Section 2.06 Pain-Modulated Motion

Intuitively and experientially, we know that pain-modulated motion is an *in vivo* phenomenon. Simply observe the gait of someone with a knee or hip injury, and the modified kinematics, which serve to guard the painful area, cannot be missed. Researchers have spent time and energy to quantify kinematic and kinetic changes due to back and neck pain [44-47]. Typically, their results focus on measures of speed and range of motion of specific activities. In addition, Shum noticed that not only is the maximum range of motion reduced with back pain, but a significant increase in the moment acting through the range occurred [48]. In his work on the effect of experimental low back pain on neuromuscular control of the trunk, Dubois concluded

that the kinematic data suggested that chronic low back pain patients adopted a different movement strategy than healthy controls [47]. It is these types of observations that serve as the impetuous for neuro-musculoskeletal simulation.

Section 2.07 Measuring "Pain"

The most common sources of back pain include intervertebral discs, the facet joints, and the sacroiliac joints [49]. In the first iteration of the NMS, thin film force sensors in the facet joints were used to provide feedback to the controller during a simulated motion of the spine. Biologically, nociceptors can provide the pain stimuli that would modify this motion. This is why it is important to understand where these nociceptors exist in the facet joint and if force measurement is an adequate analog for *in vitro* cervical facet joint pain estimation. McLain [50-52] and others [53-54] have investigated the types and locations of the mechanoreceptors that act as nociceptors in the facet joint capsule, synovium, and the loose areolar tissue of the facet joint capsule. However, a thin film force sensor in the facet joint is going to measure the force seen by the cartilage on the articulating surfaces rather than on the capsule. Nociceptors are not usually found in articulating cartilage, though Szadek potentially found some in sacroiliac joint cartilage [55]. So, for the purpose of this study, we will assume we are starting with facet osteoarthritis which would likely have nociceptive responses related to force applied to the joint articulating surface. This assumption is based on the idea that the cartilage would be degraded to the point where joint contact would be made with the subchondral bone rather than the articulating cartilage. We assume that any sleeping nociceptors in the subchondral bone would be activated due to the inflammation caused

by joint loading without articular cartilage. We then make the assumption that increases in joint force would be proportional to increases in the nociceptive response.

Simply because a nociceptor is firing action potentials does not mean that pain is perceived by the CNS. In the ascending somatosensory pathways, these signals can be modulated by the periaqueductal grey, thalamus, and other structures to change the perceived amount of pain. For the purpose of this study, we will assume that whatever modulation is provided by these structures is minimal and consistent. In this way, we can compare results of pain-modulated motion, with varying levels of pain, to elucidate characteristics such as linearity of the response without confounding the results. However, it is important to note that this assumption should not be overlooked when applying results of these studies to *in vivo* applications.

If we assume that force is an adequate analog for pain, the next thing to determine is the correct sensor type. Several researchers have investigated the question of facet joint force/pressure to understand the impact of pathologies [56], trauma [57], and surgeries [58], as well as to validate finite element models [59]. The most common type of measurement technique is a thin film force sensor [60-62]. Others have used pressure sensitive film [63], strain gages bonded to the articulating surface [64-66], an array of individual sensor elements which are composed of two beryllium–copper surfaces with a strain gauge between them [67], and cylindrical pressure sensors with a sensing membrane at their tip [68]. The thin film force sensor, pressure sensitive paper, and strain gage array require some amount of resection of the capsule and have been shown to

increase the estimate of the contact area [69]. The bonded strain gage technique conforms to the surface but requires full transection. The cylindrical pressure sensors are passed through the bone of the facet joint and, as a result, leave the capsule intact. The reason some have been led to use this technique is that a partially resected capsule may potentially bias pressure measurements by changing loading patterns [70]. In addition, the fully transected configuration produces a hyper-mobile joint that can produce contact in non-physiologic locations [68]. The cylindrical pressure sensor is not without limitations as well. The sensing tip is small in comparison to the load bearing surface, and the performance in complex geometries and loading scenarios is not well established. In addition, the installation and use is not trivial. For the purpose of this study, we are not interested in what the actual force is. We are simply interested in using this sensor as a surrogate for a nociceptor. We are willing to accept the limitations in measurement bias and contact area in return for a relatively straightforward and proven technique.

A force sensor may not be able to be considered an adequate analog for pain measurement due to the fact that it would be difficult to estimate the locations and loading conditions that ultimately cause pain with a relatively large thin film sensor (compared to the smaller neural structures). In essence, it is attempting to model a complex microbiological system with constitutive relations and theories based on the application of continuum mechanics. Certainly, the nociceptive structures don't fit their continuum mechanics models in terms of the actual physics. However, Humphrey [71] argues that the real strength of these models is in their predictive nature. As an example, the total joint force will not represent the force found on the free nerve ending, but it

certainly can be an estimator of the potential for pain. When the force, as measured by the sensor, increases with mechanical loading from a robot, then the likelihood that stresses will rise near those free nerve endings also increases. A rise in stress in those structures can indeed lead to pain.

Section 2.08 Musculoskeletal Simulators

Musculoskeletal Simulators (MS) provide the ability to study underlying bone and soft tissue interactions. Performing these types of studies *in vivo* presents technical and ethical challenges that can be overcome with the use of *in vitro* cadaveric studies. MS possess the ability to apply kinematic and kinetic changes to a joint while estimating responses to those conditions. This can be used for studying intrinsic joint behavior as well as outcomes of surgical interventions [72-73]. The motion changes for a given loading condition have been studied [74-76]. In addition, sensors can be added to the joint to study joint pressure [77] or other independent variables. MS have been used to study joints such as knee [72, 75, 78], hip [79-80], shoulder [81-82], foot/ankle [77, 83], hand [84], and spine [85][2-33].

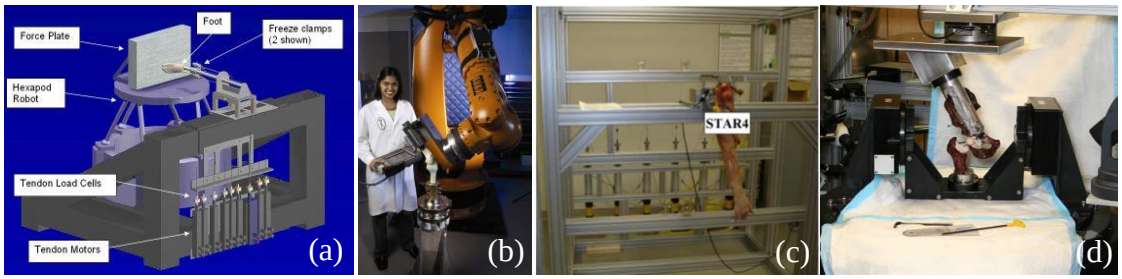


Figure 8: (a) Seattle VA Robotic Gait Simulator (b) Pitt Knee Robot (c) Pitt Shoulder Testing Apparatus (d) Cleveland Clinic Hip Simulator

For testing of some joints, dynamic or static muscle forces can be added and applied as a joint-level load. In other cases, such as foot and ankle, the osteoligamentous

interactions are much more complex and require the addition of individual tendon actuators to simulate the activities of individual muscles [76-77, 83].

Section 2.09 Spine Musculoskeletal Simulators

Panjabi, Patwardhan, Goel, and others have developed methods to apply pure moments to a single axis of the spine to elucidate the mechanical properties of the spine before and after treatment. Figure 9 is an example of these types of loading frames. One technique to biomechanically investigate these questions is to hold one vertebra stationary and manipulate the adjacent level. A bone-disc-bone pair of adjacent levels would be considered a Functional Spinal Unit (FSU). Through kinetic and kinematic manipulation, the FSU passive properties, such as range of motion and stiffness, can be elucidated. Multiple FSUs can be tested simultaneously by leaving them intact and testing a few levels or entire sections (Cervical, Thoracic, or Lumbar). With these technologies and the pure moment testing technique, the stiffness and range of motion can be extracted from the curves for both the neutral and elastic zones.

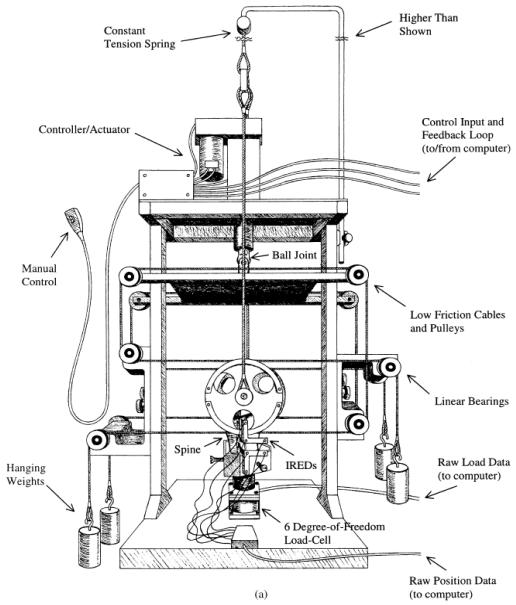


Figure 9: Loading apparatus for the application of continuous pure moment loads to multi-segment spine specimens. This apparatus allows continuous cycling of the spine between specified flexion and extension (or right and left lateral bending) maximum load endpoints. [4]

Since the loading is performed via deadweights, the data points will be discrete and the curves will be interpolated (Fig 10). It is also difficult in manual application of loads to measure hysteresis-like effects that appear with loading and unloading applications of the loads.

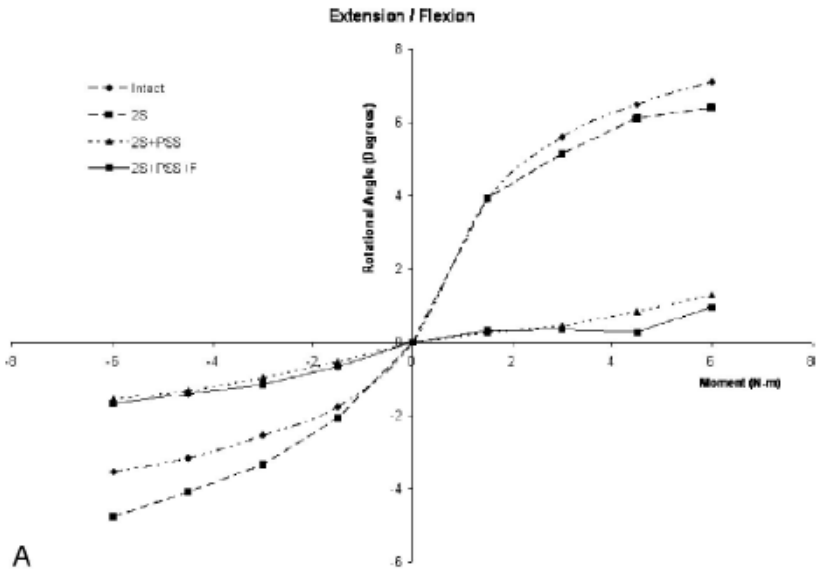


Figure 10: Flexibility testing curve [5]

Researchers including DiAngelo [18], Gilbertson [25] [29-33], Wilke [22], Pearcy [24], Dickey [17, 21], Stokes [20], Cunningham [15], Kawchuck [26], Leibschner [6], Zdeblick [86], and others [16, 23, 28] have taken the loading frame a step further and developed MS for cadaveric spines. These systems include 6 degree of freedom robots and specialized 6-axis force transducers to measure and apply the loads in a controlled manner. Many different control techniques exist to accomplish this goal, but they are fundamentally based on some combination of position control (feedback from the robot) and load control (feedback from the 6-axis load sensor).

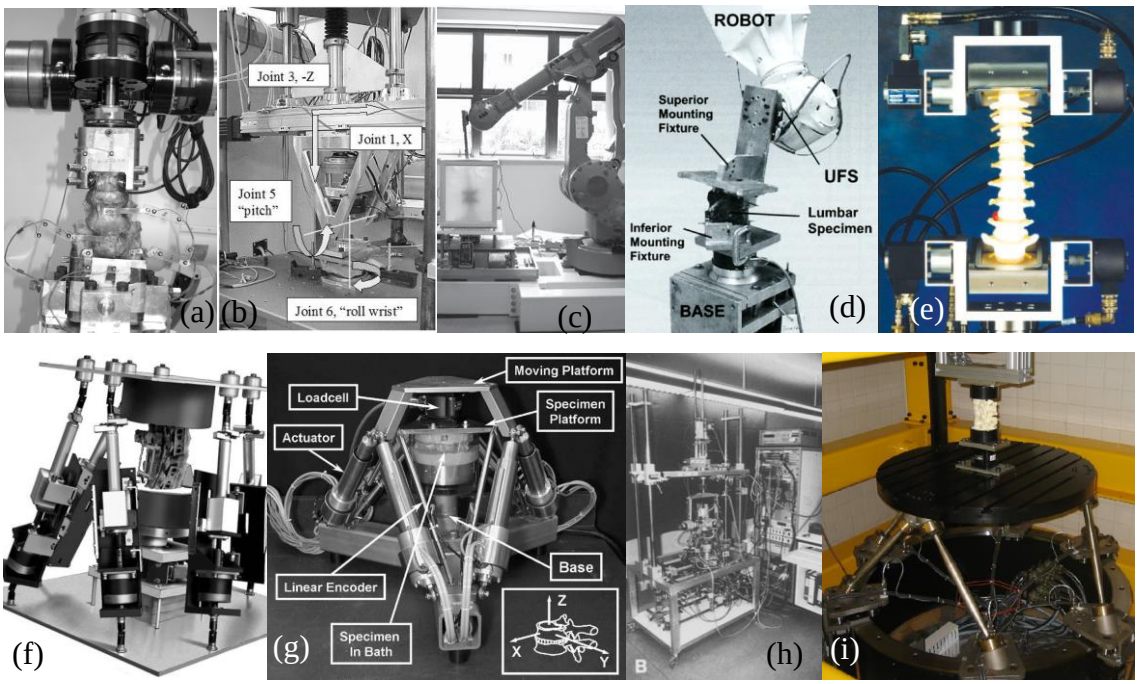


Figure 11: Robotic MS from (a) Cunningham (b) DiAngelo (c) Pearcy (d) Gilbertson (e) MTS/Zdeblick (f) Dickey (g) Stokes (h) Wilke (i) Kawchuck

These MS have been used to perform pure-moment testing similar to the loading frames, and characteristic curves have been generated before and after various surgical interventions to the spine. All off-axis forces and torques are minimized via iterative or real time force/torque control algorithms. Figure 12 contains representative data for a flexion-extension type motion. Note the continuous curves and differentiation of the loading and unloading curves. Also, the stiffness and range of motion variables can be extracted from the plots for both the neutral and elastic zones.

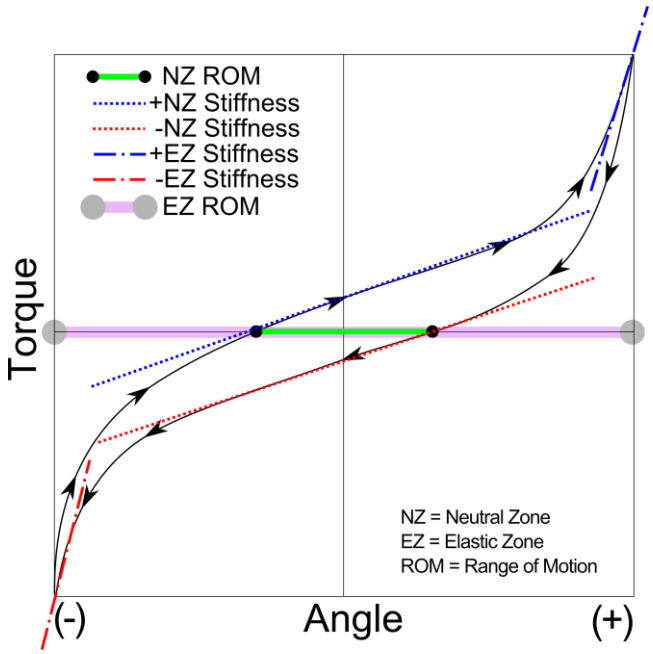


Figure 12: Typical pure moment flexibility curve with continuously changing torque

Table II contains a list of the known multi-axis robotically controlled spinal biomechanics testing systems. Even though these investigators are more prolific in this area than this table may suggest, select manuscripts are referenced for each one.

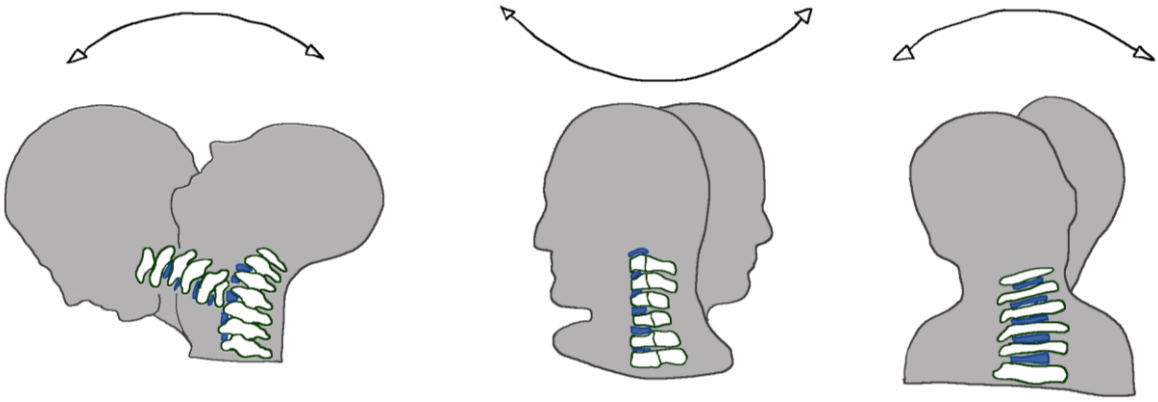
Table II: Robotic testing systems for spinal biomechanics

Investigator	Title	Specimen	Laboratory	Technique	Year
Wilke [22]	A universal spine tester for *in vitro* experiments with muscle force simulation	Human	University of Ulm, Germany	Custom 6-DOF robot 8	1994
Gilbertson [33]	New Methods To Study Lumbar Spine Biomechanics: Delineation Of In Vitro Load-Displacement Characteristics By Using A Robotic/Ufs Testing System With Hybrid Control	Human	MSRC	PUMA	2000
Stokes [20]	Measurement of a spinal motion segment stiffness matrix	Porcine	U of Vermont	Custom Parallel Robot 3	2002
Alberts [23]	Single-unit artificial intervertebral disc	Artificial	Nebraska	Panorobot	2004
Dickey [17]	Biomechanical Role of Lumbar Spine Ligaments in Flexion and Extension: Determination Using a Parallel Linkage Robot and a Porcine Model	Porcine	Guelph	Custom Parallel Robot 2	2004
Liebschner [ORS 2006]	Abstract: Acrylic Vertebroplasty may alter fracture pattern and reduce ultimate load of adjacent vertebrae	Human	Rice	Kuka	2006
DiAngelo [14]	Biomechanical Testing Simulation of a Cadaver Spine Specimen	Human	U of Tennessee	Custom 6-DOF robot 6	2007
Dickey [21]	New methodology for multi-dimensional spinal joint testing with a parallel robot	Porcine	Guelph	Custom Parallel Robot 1	2007
Pearcy [24]	A Robotic Testing Facility for the Measurement of the Mechanics of Spinal Joints	Human	Queensland UT	ABB IRB 4400/60	2007
Ferguson [16]	Minimizing errors during *in vitro* testing of multisegmental spine specimens: Considerations for component selection and kinematic measurement	N/A	AOS research institute	Custom 6-DOF robot 2	2007
Kawchuk [26]	A novel application of velocity-based force control for use in robotic biomechanical testing	Rabbit	U of Alberta	Rotopod	2008
Zdeblick MTS Corp.[86]	A biomechanical comparison evaluating the use of intermediate screws and cross-linkage in lumbar pedicle fixation	Human	U of Wisconsin, MTS	Custom 6-DOF robot 4	1994, 2008
Schulte [28]	The effect of dynamic, semi-rigid implants on the range of motion of lumbar motion segments after decompression	Human	University Hospital Munster	Kuka	2008
Hollis [IFMBE Proceedings 2009]	Abstract: Robotic Biomechanical Testing of Cervical Spine Structures	Human	U of South Alabama	Custom 6-DOF robot 5	2009
Cunningham [87]	Biomechanical Evaluation of a Posterolateral Lumbar Disc Arthroplasty Device An In Vitro Human Cadaveric Model	Human	Orthopaedic Spinal Research Laboratory	Custom 6-DOF robot 7	2010
Mageswaren [88]	Hybrid dynamic stabilization: a biomechanical assessment of adjacent and supraadjacent levels of the lumbar spine	Human	Spine Research Laboratory	Kuka	2012

Note that about half of the investigators built their own robots while the other half chose to start with commercial robotic technology and refine it for the purpose of spine testing. With the exception of the Spine Testing System by MTS, there are no commercial products directly marketed for this purpose. This poses a challenge to researchers as they need to build and validate a system prior to being able to investigate clinical questions. It also provides a challenge for comparison of data between laboratories as variations between equipment and methodologies can produce variation in results [16, 89].

Section 2.10 Cleveland Clinic Spine Musculoskeletal Simulator

The Spine Research Lab at the Cleveland Clinic purchased a Kuka (Augsburg, Germany) KR-16 robot with the goal of using it for spine biomechanical testing. We developed the system to perform pure-moment testing while minimizing off-axis loads using the 6-axis load cell and the real-time force feedback controller. A follower load system has also been developed to simulate compressive loads on the spine. To date, the spine MS has been used on cervical, thoracic, and lumbar studies. Typically, the loading conditions were pure moments in the 3 primary rotational axes (flexion-extension, lateral bending, and axial rotation).



Flexion-Extension Axial Rotation Lateral Bending

Figure 13: Primary axes of rotation in the cervical spine.

Figure 14 contains an entire lumbar spine (T12 to Sacrum) mounted to the Kuka KR-16 robot. The blue circles are the NDI Optotrak (Northern Digital Inc., Ontario, Canada) optical motion tracking sensors. Figure 15 contains representative data for a flexion-extension type motion. One important feature to note is that, unlike Figure 10, Figure 15 shows the continuous curves and differentiation of the loading and unloading curves.

Figure 14: Cleveland Clinic Spine MS

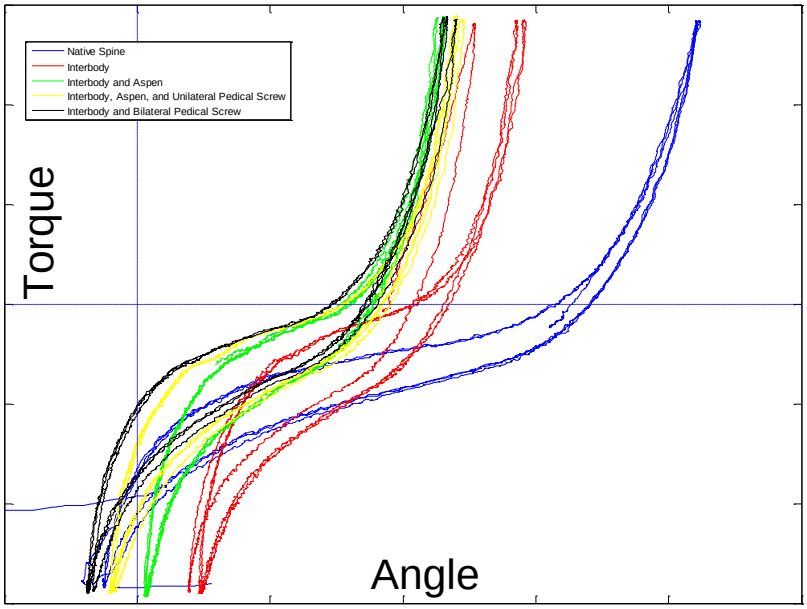


Figure 15: Flexion-extension flexibility curves for a lumbar spine after various surgical interventions

While this and other systems continue to provide insights into spine biomechanics, the current testing methodologies are based on the principle of using the robot or loading frame to perform as a multi-axis material testing system. The existing system was programmed in Kuka Robotic Language (KRL) which is the proprietary language that the manufacturer provides with the robot. KRL is a very useful language for controlling the robot in a typical industrial environment where the robot is required to perform the same pre-planned motion over and over again. Over the years, Kuka has expanded the functionality of the language to include the ability to interface with sensors and couple sensor feedback with the programmed motions. They also sell an add-on for force-torque control. While these changes made the existing body of work possible, it remains a cumbersome tool and suffers from several limitations including:

1. Array size limitations

2. Force-torque control is designed for constant load situations, such as buffing the contours of a fender. It is not a simple process to create and modify dynamic loading trajectories.
3. Inability to manually adjust force control gains while running. Tuning is a slow trial and error process. The gains can be adjusted programmatically but these techniques are not straightforward or even included in any of their documentation.
4. The programming language is proprietary and modifications to the software by future engineers will potentially require a large learning curve.
5. Additional sensors can be added to system, and can potentially be used for control, but these techniques carry significant development efforts for each new type of experiment.
6. Lacks a graphical user interface which is a barrier to the replication, dissemination, and adoption of this software by other laboratories.

For these reasons, and others, the decision was made to leverage our existing LabVIEW (National Instruments, Austin TX, USA) software base of the Cleveland Clinic BioRobotics Lab in the development of the NMS.

Section 2.11 Summary of Background

The testing systems presented in the preceding sections have provided significant insights and benefits into spinal care. This result should not be understated. However, having the ability to ask scientific questions about spinal stability from a neuro-musculoskeletal perspective stands as a novel development in the current state of the art.

A wide range of clinical questions could be considered. Does the pain-modulated motion potentially increase stresses elsewhere in the spine and accelerate other pathologies? Spinal stenosis can result from facet joint degeneration. Would earlier surgical intervention minimize the risk of the onset of this disease? Are there yet to be developed technologies that can aid in minimizing the progression of osteoarthritis (OA)? How effective are these technologies at potentially minimizing pain? Current MS technologies generally treat all specimens the same for testing purposes, but it is highly unlikely that all spines were equally healthy prior to their use in experiments. Could examination and classification of the level of OA or degenerative disc disease (DDD) be used to scale pain for individual specimens and provide more subject specific results on implant performance? These are just a few of the many potential questions that could be asked. There are many more that we have not even conceived yet. In this work we used neural response modulated motion as part of the spinal neuro-musculoskeletal control algorithm to answer a very specific question about facet joint pain. However, this is simply a starting point to ask more pertinent clinical questions and provide more clinically relevant answers with the ultimate goal of providing better treatment for people with spine related health problems.

CHAPTER III

NEUROMUSCULOSKELETAL SIMULATOR SYSTEM DEVELOPMENT

Section 3.01 NMS: A subset of the UMS

Development of the NMS was built upon work done to develop the Universal Musculoskeletal Simulator (UMS) at the Cleveland Clinic. The UMS has been used for cadaveric joint level testing of the foot/ankle, knee, shoulder, and hip [72, 76-77, 81]. Originally built upon a Mikrolar R2000 (Hampton, NH) parallel robot, the UMS has been developed to enable fundamental research such as injury prevention, evaluation of surgical intervention, total joint replacements, and the development of rehabilitation regimens. The UMS can simulate the biomechanics of human motion at any joint through (i) a set of actuators that, when connected to selected tendons traversing a joint, can imitate muscular contractions, and (ii) a robot that can simulate environmentally induced loading/contact of the specimen. Bone strain, soft tissue strain, non-contact video based multi-dimensional strain, joint force/pressure, or other measurements of interest can be recorded for correlation with the muscle or external forces during the loading condition. The benefit of these coupled systems is that they enable biomechanical simulations with joint loading at physiological levels.

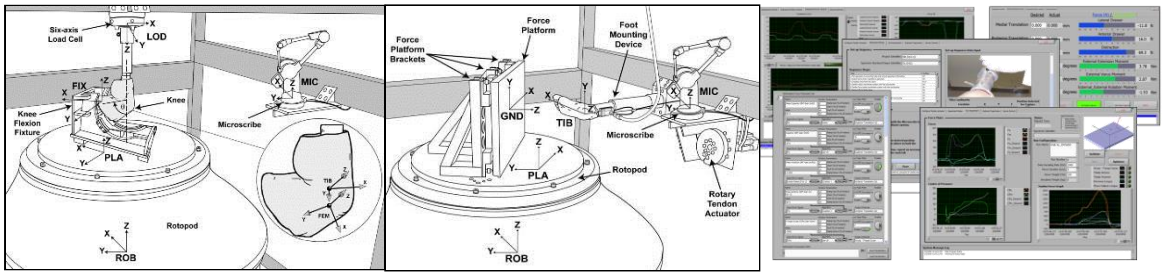


Figure 16: (a) UMS Knee System (b) UMS Foot System (c) Foot software

As a result of the case by case development to suit each new project, the UMS software suite ended up being a collection of similar software applications wherein each application was a copy and slight variation to accommodate a new anatomical joint or study. The end result is a group of applications that are hard to maintain due to the necessity to upgrade all copies with any new features added. The original plan for the development of the NMS was to copy the most developed version of the UMS and make two major additions to it. The first addition would have been to make it compatible with the spine, and the second would have been to make it compatible with the Kuka robot. However, at the time of the development of the system for this dissertation, there was a purchase of a Denso robot in the BioRobotics Laboratory as well as interest from laboratories outside the Cleveland Clinic to purchase a similar system to support their testing needs. These testing needs included other robotic systems and multiple joints. The existing model for new application development, which resulted in multiple copies of similar software, was not sustainable. As a result, the creation of the NMS needed to be aligned with the BioRobotics Laboratory goal of creating a single flexible, scalable, and maintainable UMS software package.

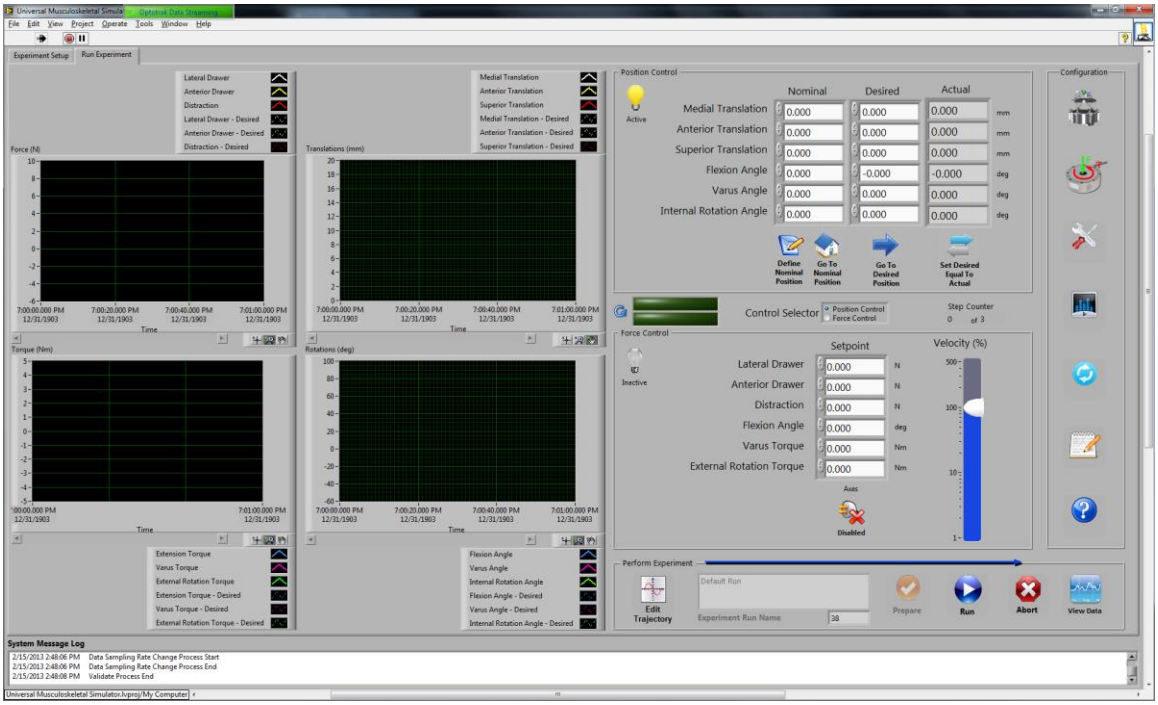


Figure 17: UMS Software User Interface

At the core, the UMS software provides real time 6-DOF kinematic and kinetic control. The joints, and corresponding coordinate systems, are all different, but the general principle is the same. For this reason, one requirement for the newly architected software was flexibility to work with any anatomical joint. A second requirement was flexibility to work with any 6-DOF robot. A third major requirement was the ability to collect data from a wide array of sensors. The user should be able to collect data from most common sensors that would be added to these types of experiments without any further coding. To implement these major requirements, it was clear the system architecture needed to be revisited and Object Oriented Programming (OOP) techniques employed. Compared to traditional procedural coding techniques, OOP provides the benefits of abstraction, encapsulation, inheritance, and polymorphism. My background and training is that of a Mechanical Engineer, so acquiring this new skill set involved a

significant learning curve. The purpose of this dissertation is not to explain the details or benefits of OOP techniques; however, a quick review is beneficial.

Section 3.02 Object Oriented Programming Framework

In OOP methodology, pieces of software are organized into various groups called classes. These classes have properties (pieces of data they store) and methods (i.e. functions that can modify the properties or other data based on the properties). These classes can be in a tree-like structure where a child-class can inherit properties from the respective parent class. For example, the UMS software requirement of needing to be flexible to work with any 6-DOF robot implies a parent class of *6-DOF Robot*. Child classes can be created for each specific type of 6-DOF robot (e.g. Kuka model KR-16, Mikrolar model R-2000, etc.). The methods for the *6-DOF Robot* class may be generic types of commands such as *get position* or *go to position*. It is these high-level functions (methods), not the actual robot drivers, that are used in the UMS software. The hardware specific child classes contain the low-level drivers for talking to that specific robot. However, the high level UMS software doesn't know or care which robot is connected. When the program first starts, an object (the creation of a single copy or instance of that particular child-class) gets created and is assigned to be the *6-DOF Robot*. Therefore, when the function *get position* is called, it will get the position for the robot currently in use. However, if a different robot was configured and assigned to the *6-DOF Robot*, then those respective low-level drivers would be called and retrieve the position from that particular robot. In this way, an effective hardware abstraction layer is created and the hardware is abstracted from the high level software (Figure 18)

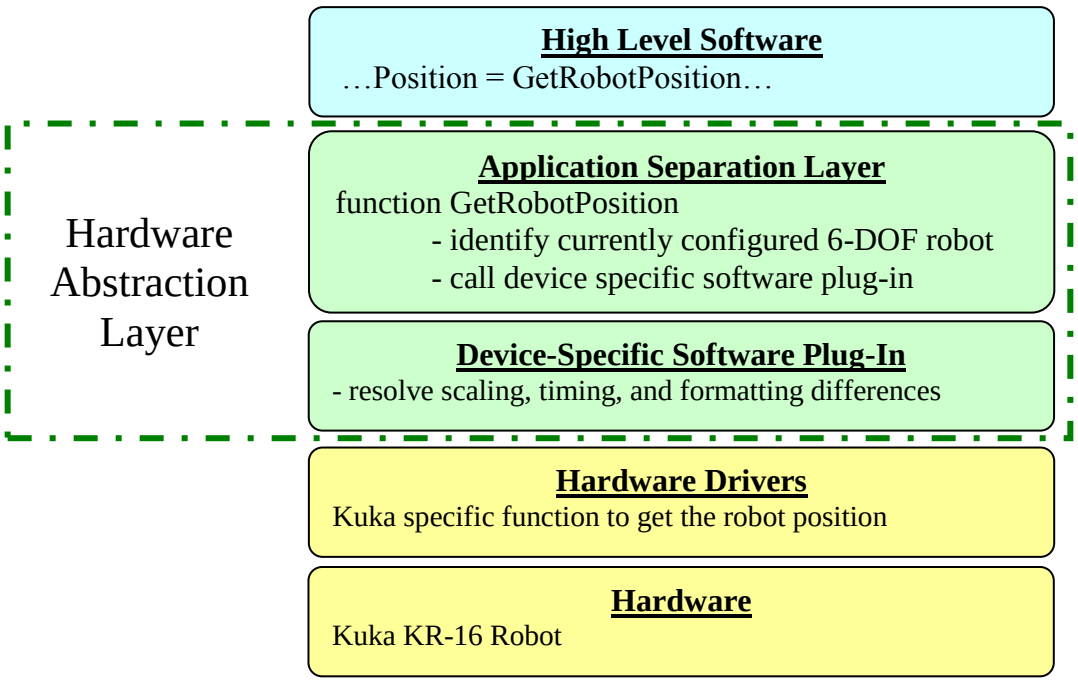


Figure 18: Hardware Abstraction Layer diagram to get position from Kuka robot

There are many advantages that this type of architecture provides. The most obvious one is that separate high-level code containing the unique hardware drivers is not needed. This gives flexibility to work with any 6-DOF robot. Even robotic systems we have not developed hardware drivers for yet can theoretically be integrated with the UMS software without requiring modification (and extensive re-testing) of the high-level code. Another benefit is directly related to software development, debugging, and maintenance. Since each child class is its own unique piece of code with expected inputs and outputs, it can easily be unit tested and verified. Provided it meets the expected inputs and outputs, it is effectively encapsulated to minimize the risk of creating problems in other parts of the code when either the child class , inputs, or outputs are modified.

During the re-architecting process, it became desirable to consider a broader scope of the fundamental building blocks of the UMS. The UMS software, as far as the fundamental functionality and user interface is concerned, is designed for musculoskeletal simulation of human and animal joints for biomechanical research. However, any controlled mechatronic system can typically be broken up into several fundamental building blocks: Sensor, States, Controllers, and Actuators. Sensor data can be used to compute states. State data can be used as an input to a controller. The controller output is sent to an actuator whose influence on the system is likely to change sensor output, closing the loop. It was determined that the creation of a library of Sensor, State, Controller, and Actuator classes would provide a flexible software toolkit. This could be used in the UMS software along with any other data acquisition or control system that may be developed. This idea is not unique [90-93], though this implementation is. Other creations of these types of class libraries tend to be tailored toward solving a specific problem, and they lack the flexibility required for this particular application. It is a fair criticism that this library currently suffers from similar limitations. However, efforts were made to make the parent classes as generic as possible and readily allow for future implementation of more child classes. This implementation is also LabVIEW based. No other LabVIEW based libraries of this type were identified in literature or in web searches.

Section 3.03 Sensor Class

The first library created was the sensor class. The sensor class is effectively a hardware abstraction layer that has a sensor manager to allow the user to configure a

number of possible sensors. For example, the user can choose from analog sensors (single channel, 3 channel, 6 channel, or 2D grid) based configurations. They can also choose to acquire data from digitized sensors. Digitized sensors are defined as sensors whose data does not come in from the Analog to Digital (A/D) board, but whose values are streamed across a network or from other connected peripheral hardware. Digitized sensors require drivers, but those drivers can be wrapped in the hardware abstraction layer. The drivers can also be made to interface with the UMS or any other high level software using the Sensor-State-Controller-Actuator toolkit. The current toolkit allows for the configuration and acquisition of single and multi-channel force, position, and pressure sensors.

The sensor class exploits many of the features already built into the National Instruments (developer of LabVIEW software) hardware and software for analog data acquisition. The National Instruments Measurement and Automation Explorer software application is designed to allow for pre-configuration and saving of channel properties and other data acquisition parameters. These include the sample rate, scaling factors, signal conditioning, etc. If all the sensors for this system were single channel analog sensors, then there would have been no need to develop the sensor class. However, the 6-DOF load cells have 6-axis strain gage bridges whose outputs which require multiplication by a 6x6 calibration matrix for each data acquisition loop. Measurement and Automation Explorer software provides no inherent functionality for these types of combined multi-channel measurements. The software also lacks the ability to

communicate with digitized sensors. Figure 19 shows the hierarchy of sensor classes available in the sensor manager.

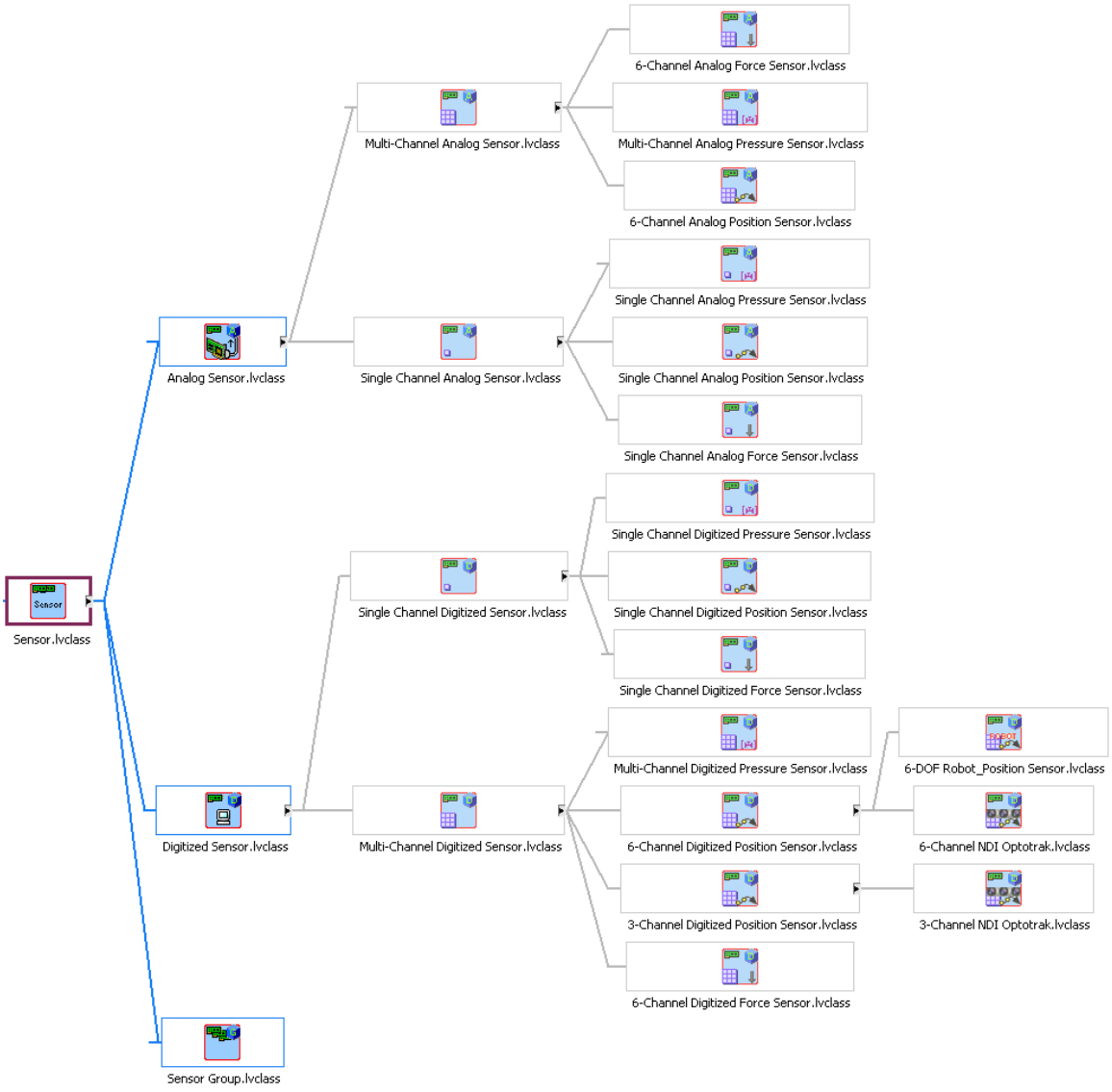


Figure 19: Sensor class hierarchy

Note that the far right column in figure 19 contains examples of hardware specific classes developed as child classes of the sensor class. Each component of this Sensor, State, Controller, Actuator toolkit has a manager application to load, save, add, remove, and modify the sensor objects. Figure 20 is the Sensor Manager application. All of the sensors are added to a list and stored in an object called the Sensor Group. The sensor

classes and the Sensor Manager satisfy the requirement for flexible addition of sensors to any given experiment. It should also be noted that due to the nature of the hardware abstraction layer, it permits virtual sensors to be created and utilized in simulation-based scenarios. In this way, system software can be developed, and significant debugging completed, without having specific hardware connected or available.

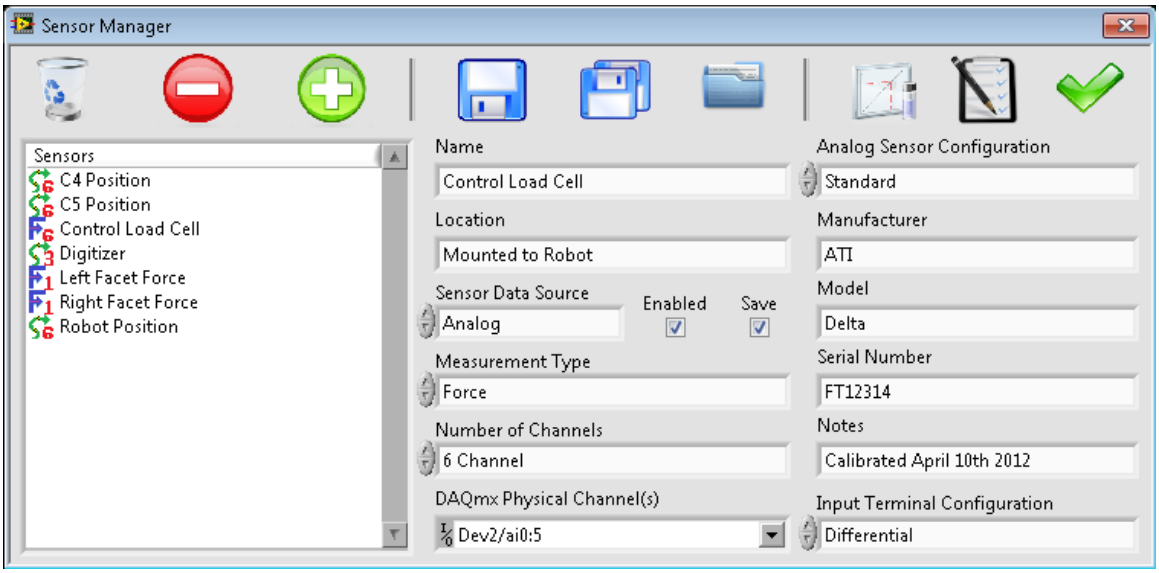


Figure 20: Sensor Manager

The UMS software can minimally operate with two basic sensors, the primary 6-DOF load cell and the robot position. Other than that, the user can select the channel and scaling for any additional sensors to automatically record from them throughout the experiment. Modification of the UMS code is not required.

Section 3.04 State Class

A state is a system variable used to quantify a unique measurement of any particular system. The state class builds system states based on the output of sensors or other states. These states can be the loads transformed to any joint coordinate system

(JCS), or they can be relative spatial relationships. The states can also use input from multiple sensors for redundancy and error checking, if desired. The UMS software requires two states to operate, the JCS kinematics based on the robot position and the JCS loads based on the primary 6-channel load cell. Other than that, the user can implement states from the pre-built templates, or they can code their own states. The system software is able to compute and record the state data for any additionally configured states. An example of a state template would be the 6-DOF kinematic relationship for each FSU for a spine test.

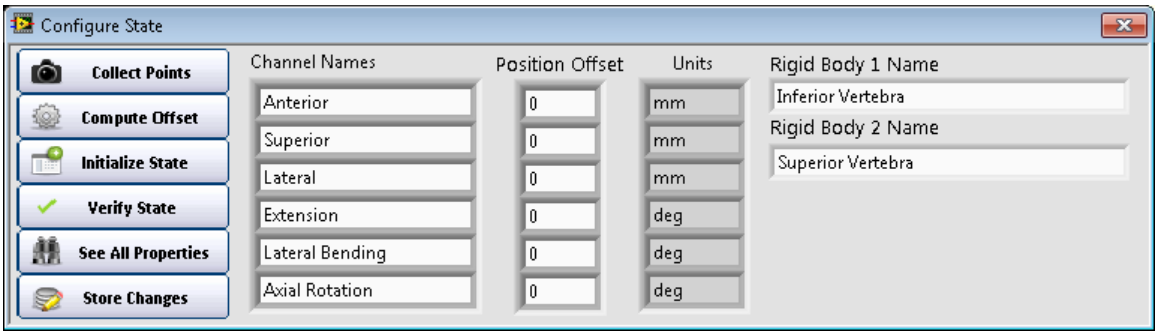


Figure 21: Template for 6-DOF spine FSU position state

By placing motion tracking sensors in each vertebra and selecting the state template from the spine module that computes FSU kinematics, the system will direct the user to digitize the proper anatomical landmarks to compute the proper coordinate systems. Then, during the experiment, the motion tracking sensor data is used to compute the kinematic state of the FSU. The state can also be displayed on the screen and written to a data file. In a similar manner, multiple states can be created for measuring level-by-level spinal kinematics with each one directed at a different spinal level (C4-C5, C5-C6, C6-C7).

States can also be combined with sensors to calculate other states. If the load cell is moving relative to the gravity field, gravity compensation is required to eliminate gravity based sensor cross-talk. Another state template designed for this class is the *6-DOF load* class. This class uses the robot position sensor, the load cell sensor, and the state that contains the relative spatial relationship between the load cell and the anatomical reference frame that the loads are controlled in. Note the *JCS load* state in Figure 20. The information from the sensors and the position state, also built from a template, is combined to compensate for gravity and transform the loads to the anatomical reference frame. This demonstrates how states can be combined while increasing flexibility and minimizing the recalculation of variables.

State classes are also hierarchically designed so that specimen specific classes will inherit properties from the generic classes. Figure 22 contains the class hierarchy of states created for the UMS software for spine testing. Also, similar to the Sensor Manager, the State Manager (Fig. 23) provides an application for the user to load, save, add, remove, and modify the state objects.

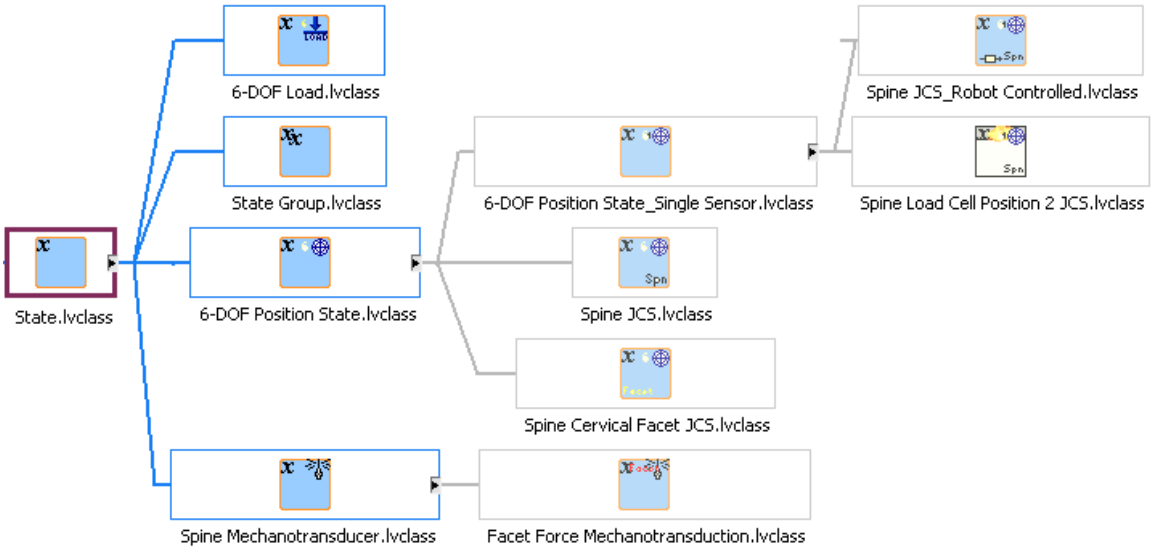


Figure 22: State Class Hierarchy

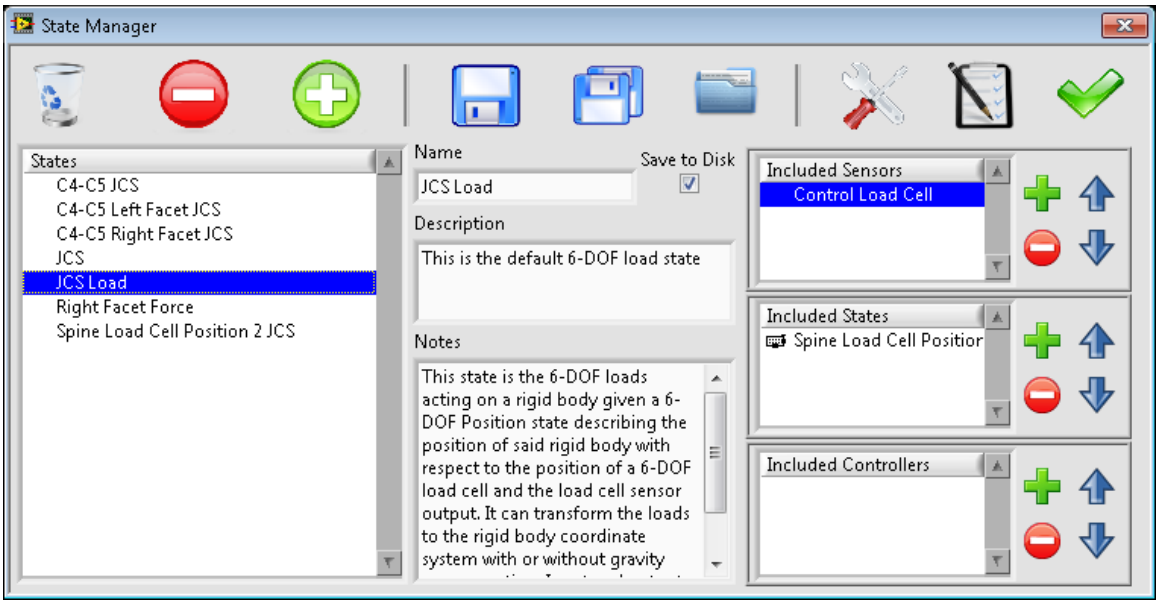


Figure 23: State Manager

For the computation of states in the UMS software, digitization of spatial relationships is required. Any 3-DOF position sensor can be configured in the Sensor Manager and used as the digitizer to provide x, y, z values of points in space. Once all the states are digitized during the setup process and corresponding transformation matrices are computed, the UMS software states are considered initialized. Figure 24 is

an example of digitizing the spatial relationship between the robot base reference frame, the end effector, and the digitization world reference frame.

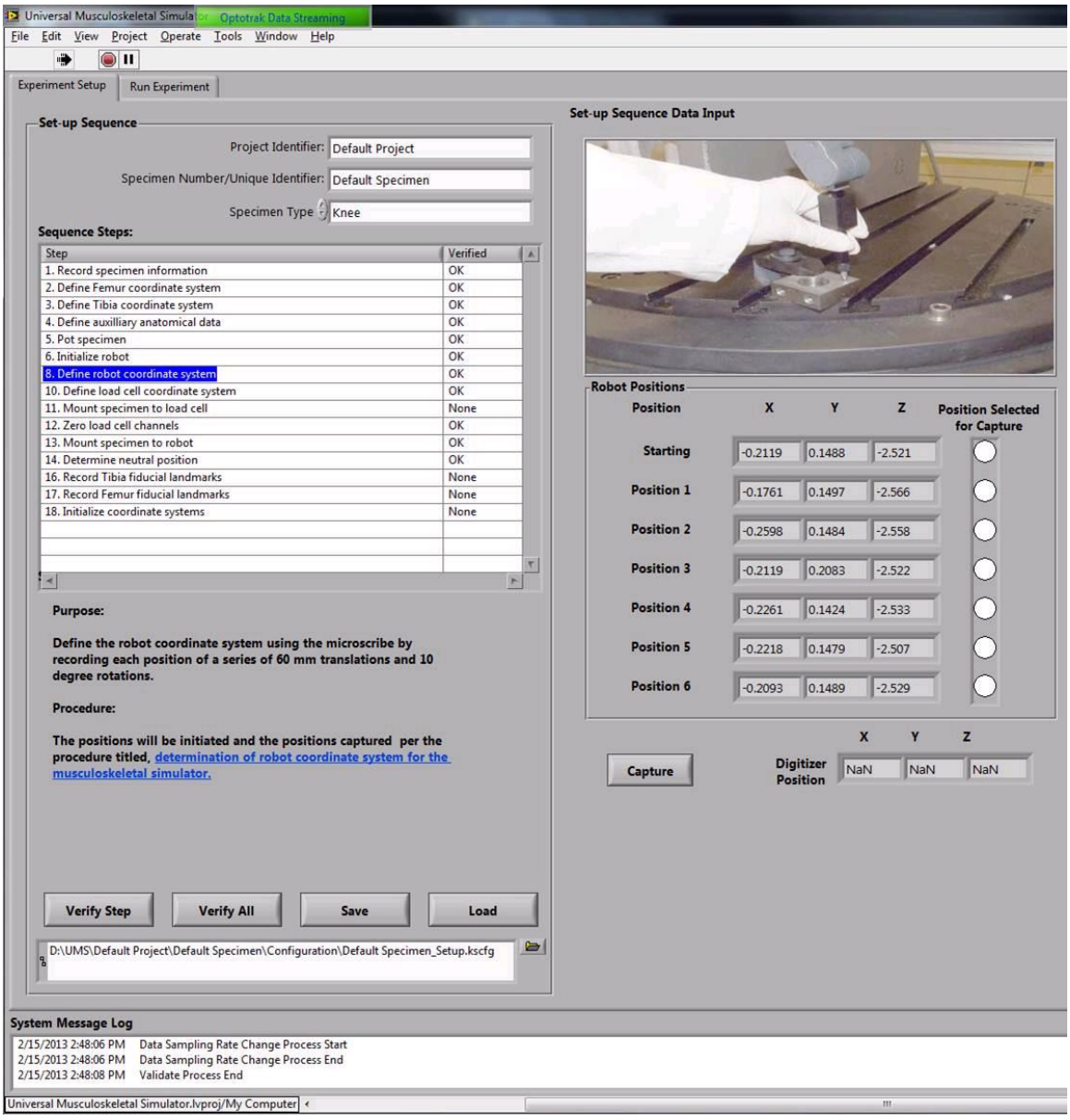


Figure 24: Digitization of state spatial relationships

Section 3.04.1 State Class: 6-DOF Position State

The state objects contain the spatial relationships by which the control can take place. Per Figure 22, the *6-DOF Position State* class is the parent class to a series of

other classes to handle these kinematic relationships. This state is the relative position of two rigid bodies given the position in space of 2 position sensors. It stores the relative (static) relationship between the rigid bodies and the sensors. Then given each sensor position, it can calculate the relative motion. The first position sensor in the included sensor array is the base rigid body (rigid body 1), and the x, y, z, roll, pitch, yaw relationship is the position of rigid body 2 relative to rigid body 1.

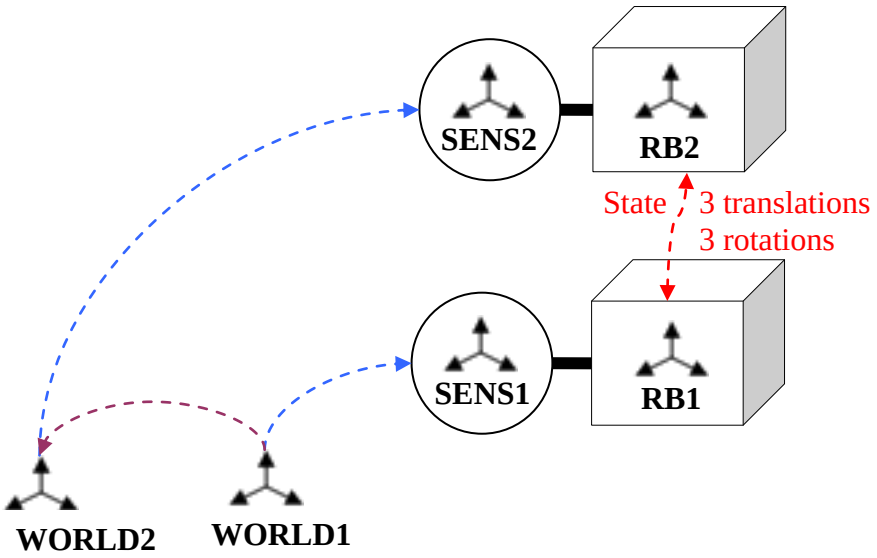


Figure 25: *6-DOF Position State* class functionality

A kinematic chain equation was developed for this state:

$$\mathbf{T}_{RB1,RB2}(\vec{x}) = \mathbf{T}^{-1}{}_{SENS1,RB1} \cdot \mathbf{T}^{-1}{}_{WORLD1,SENS1}(\vec{p}_1) \cdot \mathbf{T}_{WORLD1,WORLD2} \cdot \mathbf{T}_{WORLD2,SENS2}(\vec{p}_2) \cdot \mathbf{T}_{SENS2,RB2} \quad (1)$$

Where: (see Appendix A,B for kinematic chain and transformation matrix notation conventions)

$\vec{x} = Kinematic\ State$

$\vec{p}_1 = Sensor\ 1\ Position$

$\vec{p}_2 = Sensor\ 2\ Position$

The $T_{World1,World2}$ matrix can be an identity matrix which indicates that World1 and World2 are the same location and orientation. However, having this matrix in the

kinematic chain provides the necessary flexibility to allow two distinct position measurement systems to work together to calculate a kinematic state. In the generic instance of this class, the kinematic state is reported using x, y, z, roll, pitch, yaw conventions for translations and rotations. However, in the child classes, these methods can be modified to suit the conventions for the specific rigid bodies. For example, in the *Spine JCS* class, the state relationship is defined as a superior vertebra relative to an inferior vertebra. The digitization points and corresponding coordinate systems are programmed to meet International Society of Biomechanics (ISB) standards [94]. The kinematic state for the relative superior to inferior facet motion also uses this state template. See Appendix B for more details on the creation and reporting of vertebral coordinate systems.

Section 3.04.2 State Class: 6-DOF Position State, Single Sensor

Another common kinematic state framework that was created was the *6-DOF Position State_Single Sensor*. This state is a child class of the *6-DOF Position State* and is the relative position of two rigid bodies given the position in space of a single position sensor. It stores the relative (static) relationship between the rigid bodies and the sensor and then, given the sensor position, it can calculate the relative positions. The position sensor is attached to the relative rigid body (rigid body 2) and the x, y, z, roll, pitch, yaw relationship is the position of the rigid body 2 relative to the fixed rigid body (rigid body 1). This state assumes that rigid body 1 is static and rigid body 2 is dynamic, a requirement for measuring the relative position with a single sensor. This state does not assume that the digitizer for the point collection is in the same reference frame as the

position sensor. However, it does assume that the digitizer is in the World1 reference frame. In other words, the static rigid body (rigid body 1) and the digitizer are in the same World frame. Digitization of rigid body 2 will be done in the World1 reference frame (since that is what the digitizer operates in) but the $T_{World1,World2}$ matrix will need to be utilized to calculate the $T_{Sens2,RB2}$ matrix.

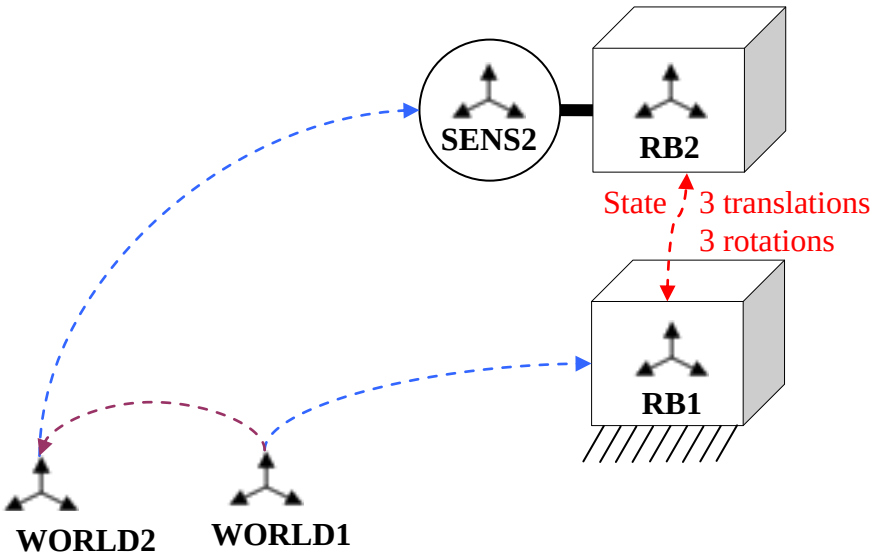


Figure 26: *6-DOF Position State_Single Sensor* class functionality

A kinematic chain equation was developed for this state:

$$\mathbf{T}_{RB1,RB2}(\vec{x}) = \mathbf{T}^{-1}{}_{WORLD1,RB1} \cdot \mathbf{T}_{WORLD1,WORLD2} \cdot \mathbf{T}_{WORLD2,SENS2}(\vec{p}_2) \cdot \mathbf{T}_{SENS2,RB2} \tag{2}$$

Where:

$\vec{x} = Kinematic\ State$

$\vec{p}_2 = Sensor\ 2\ Position$

Note that in equation (1) there are an unlimited number of sensor positions that can produce the same state value. However, in equation (2) the 3 translations and 3 rotations of the sensor 2 position can be directly correlated to 3 translations and 3 rotations of the kinematic state position. The significant benefit of this state

characteristic is that the position of a 6-DOF robot, an object in the *Actuator* class, can also be treated as a 6-DOF sensor. Methods for this state include the forward and inverse kinematics so that if the robot position is known, the kinematic state can also be known. In addition, if a desired kinematic state is required, then the corresponding robot position can be calculated and controlled to that position.

In the generic instance of this class, the kinematic state is reported using x, y, z, roll, pitch, yaw conventions for translations and rotations. However, just like the child classes to the *6-DOF Position State*, these methods can be modified to suit the conventions for the specific rigid bodies. For example, in the *Spine JCS_Robot Controlled* class, the state relationship is a superior vertebra relative to an inferior vertebra. However, unlike the *6-DOF Position State*, this kinematic state can be uniquely identified by robot position.

Section 3.04.3 State Class: 6-DOF Load State

The kinetic state required by the UMS software is the 6-DOF load in the appropriate anatomical reference frame. In its generic form, this state can be defined as the 6-DOF load acting on a rigid body given a 6-DOF load cell sensor output and a *6-DOF Position* state. The *6-DOF Position*, describes the position of that rigid body with respect to the position of the load cell. In some scenarios, the load cell may be moving relative to the gravity field (attached to the end of a robot). Therefore, the mass of an object attached to the load cell can introduce gravitational cross-talk that is not representative of the external loads applied to the rigid body. Computational correction

for this is referred to as gravity compensation. This state is designed to transform the loads from the sensor to the rigid body coordinate system either with or without gravity compensation.

The gravity compensation algorithm requires knowledge of the gravity vector relative to the current load cell orientation. The load cell offsets are subtracted from the output of the sensor. Then, the forces and moments are transformed to the world reference frame where the mass, at a known center of mass from the load cell (also transformed), can be multiplied by the gravity vector and subtracted from the sensor output. Next, the forces and moments can be transformed back to the load cell reference frame where the resulting loads are gravity compensated. The last step transforms the gravity compensated loads to the rigid body. Each step in this process requires transformations of the forces and moments (Eq. 3, 4) from one reference frame (REF1) to another (REF2).

Force transformation: $\mathbf{F}^{REF\,2} = \mathbf{R}_{REF\,2,REF\,1} \cdot \mathbf{F}^{REF\,1}$ (3)

Moment transformation: $\mathbf{M}^{REF\,2} = \mathbf{R}_{REF\,2,REF\,1} \cdot \mathbf{M}^{REF\,1} + \mathbf{t}_{REF\,2,REF\,1} \times \mathbf{F}^{REF\,2}$ (4)

Where:

F = 3D force vector from load cell

M = 3D moment vector from load cell

R = Rotation portion of the transformation matrix

t = translation portion of the transformation matrix

Section 3.05 Controller Class

The controller class provides a means by which one or more states can be controlled. Similar to the Sensor and State classes, the controller class contains a Controller Manager application for the user to load, save, add, remove, and modify the controller objects.

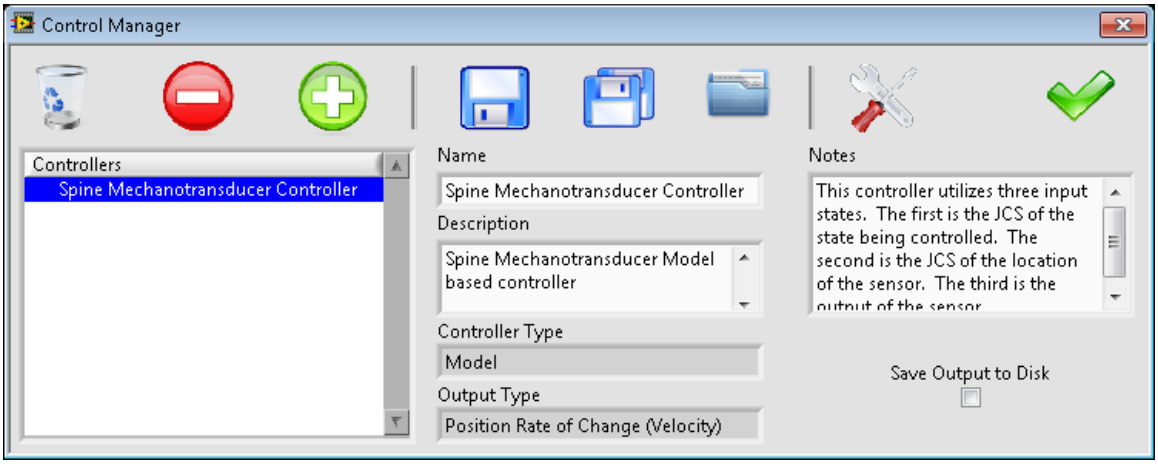


Figure 27: Controller Manager

For commercialization of the UMS software, it is unlikely this window will be made available to the end user. Liability concerns are the primary reason to not allow users to create additional controllers to influence robot motion. However, this ability is the cornerstone of the NMS, and the use of the Controller Manager was essential in the process. More details on the creation of this particular controller can be found in subsequent chapters. In the general application of the Sensor-State-Controller-Actuator toolkit, this window is the means by which the user can access previously designed controllers as templates for specific configurations.

One template developed for this class is the native *6-DOF controller* for the UMS software. This controller provides simultaneous real-time kinematic and kinetic control in

6-DOF. This is accomplished via a hybrid control system that allows the user to choose, for any degree of freedom, whether it is kinematic or kinetic controlled. The controller combines 1) a Proportional-Integral-Derivative (PID) controller for the kinetics, 2) a P controller for the kinematics, and 3) a feedforward component based on changes in the desired kinetics. The hybrid selector is a value from 0 to 1 where 1 is all kinetic and 0 is all kinematic. This is similar to the parallel hybrid scheme as proposed by Raibert and Craig [95]. Using this technique, the user can choose any hybrid value between 0 to 1 (e.g. 0.5). This non-integer hybrid value will result in the robot trying to target both the kinetic and kinematics but achieve neither because it is balancing the two control requirements. To find the desired position, the output of each of the control elements (kinetics, kinematics, feedforward) is treated as velocity, summed for the 6 anatomical kinematic channels, and then numerically integrated. The control system operates in the anatomical reference frame. Thus, after the JCS position is found via integration, it is transformed to the robot position using class methods of the *6-DOF Position State*. Figure 28 contains the control diagram for this particular class template.

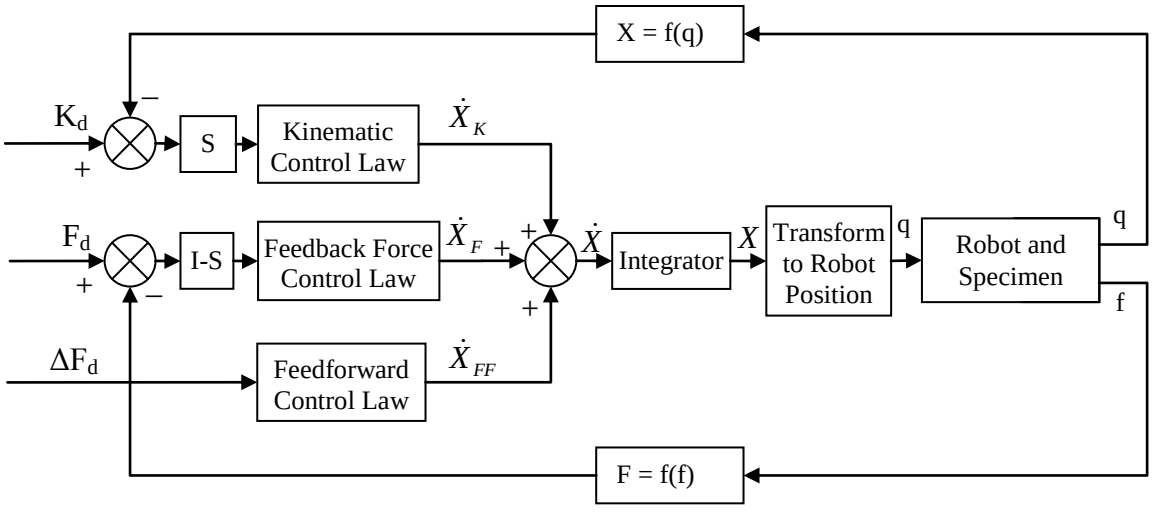


Figure 28: Native UMS controller object

Where:

q = robot pose in joint space

X = specimen kinematic JCS position

f = output of force transducer

F = transformed output of force transducer in specimen JCS

$\dot{X}$ = sum of all controller outputs

K_d = desired JCS kinematics in specimen coordinate system of interest

$\dot{X}_K$ = JCS velocity from kinematic controller

S = diagonal matrix such that when $s_j = 0$, the axis is position controlled, and when $s_j = 1$, it is force controlled. Values between 0 and 1 are also valid and provide simultaneous balance between the two control schemes.

F_d = desired forces in specimen JCS

$\dot{X}_F$ = JCS velocity from feedback force controller

$\dot{X}_{FF}$ = JCS velocity from feedforward force controller

I = identity matrix

Section 3.05.1 Controller Class: Trajectory Editor

The *compute state* class method is used to compute the state based on actual sensor and state inputs. However, to control a state, a desired state must be given. The Sensor-State-Controller-Actuator toolkit contains no native functionality for generating the desired states. The high-level software that employs the toolkit is responsible for generating the desired states. One use of this toolkit could include mobile robotics where

path planning and desired states will be continuously generated based on sensor inputs that drive high level logic. In other cases, a series of desired states can be fully known and a time-based list generated prior to the execution of a task. This is the case for the UMS software. The desired state for the native UMS controller is managed via the Trajectory Editor. The editor is coupled to the two UMS required states: the JCS kinematics based on the robot position, and the JCS loads based on the primary 6-channel load cell. Figure 29 contains a screen shot of the trajectory editor.

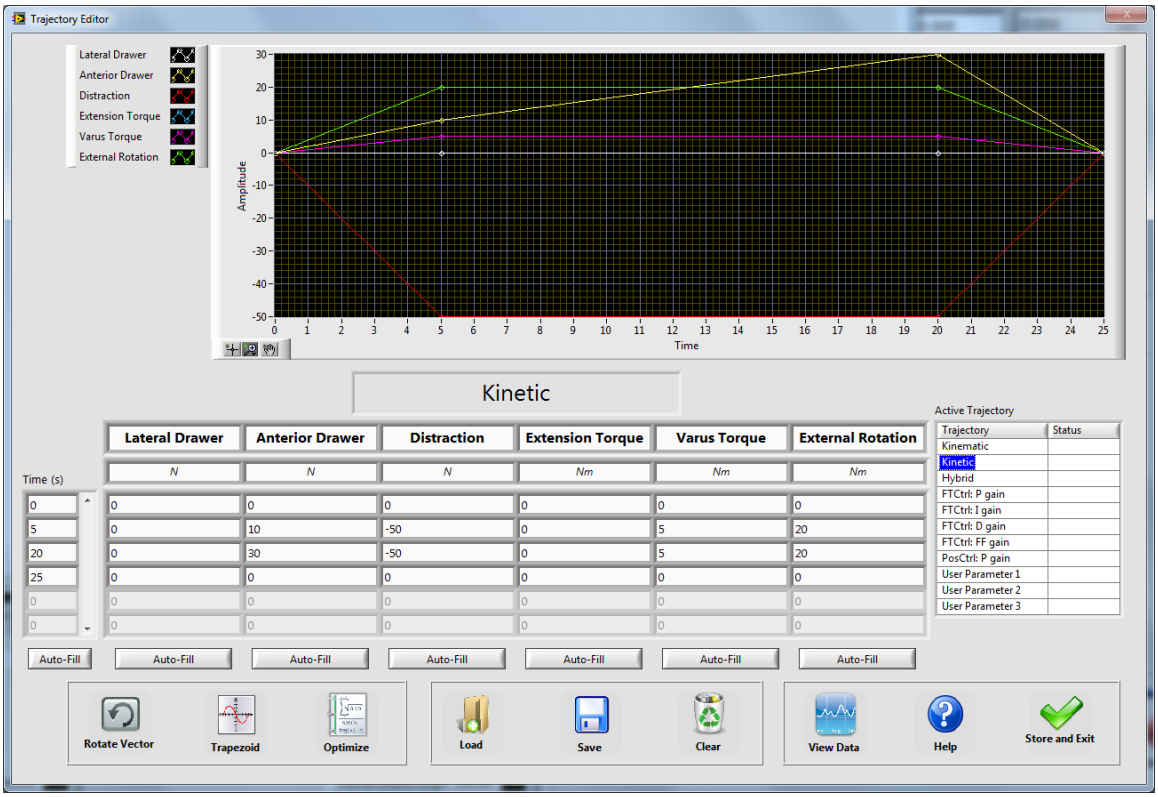


Figure 29: Trajectory Editor

The trajectory editor provides a way to create trajectories for the loads/motions. A trajectory contains:

1. The 6-DOF desired kinematics in the joint coordinate system
2. The 6-DOF desired kinetics in the appropriate anatomical reference frame

3. A hybrid selector for each channel (0 to 1)
4. Proportional gains for each kinetic channel
5. Integral gains for each kinetic channel
6. Derivative gains for each kinetic channel
7. Feed forward gains for each kinetic channel
8. Proportional gains for each kinematic channel
9. Eighteen user configurable parameters that serve as modifiable variables throughout the trajectory execution. These parameters can be used as setpoints or gains for additional actuators or controllers.

The 66 values are defined in the trajectory for a given point in time (row in the editor in figure 29). At the next time point in the trajectory, all the values can be different. This provides a flexible system to execute any type of trajectory that the user wants. The gains can be varied throughout the trajectory to account for any non-linearity in the specimen. The variation in the hybrid value between kinetic and kinematic control can also occur at any time in the trajectory execution. All values are linearly interpolated from one time point to the next. If the user were to perform a spine pure moment test, they would utilize kinetic control and then have the desired torques match a sinusoidal profile on the primary axis. The other axes would be commanded to minimize (0) forces and torques. If the user wanted to perform a study where the center of rotation was constrained, then translation axes would be placed in kinematic control (via hybrid value = 0) and rotations left in kinetic control using a hybrid value of 1. The same trajectory editor can be employed in the same manner for other specimen types. For a knee study, a

loading profile as a function of time for gait, or stair climb, could be entered in the trajectory editor for each degree of freedom. It could then be executed where all channels would be load controlled except for the flexion axis which would be position controlled. In addition, the trajectory editor has functionality for iterative learning via the optimize button.

The optimize button opens up a window where kinematics from previous experiments can be extracted and placed in the current trajectory as the desired kinematics. The hybrid value can be set to some non-integer value between 0 and 1 which will allow the robot to attempt execution of the learned kinematics while still trying to fulfill the desired loading conditions. The other functionality that the optimize window provides is the ability to scale gains and incorporate learned non-linearity from previous runs. In Figure 17, the blue *Velocity* % slider serves as a master override gain to the system. This allows the user to easily modify the overall system gain and, in real time, account for the non-linearity in specimen stiffness. Each time data is written to a file during the experiment, the value of this velocity slider is also recorded. The trajectory optimization process multiplies this velocity value by all the control gains at each setpoint and writes those to the optimized trajectory. In this way, the need for the user to constantly monitor and modify the system gain is reduced in subsequent runs. It also allows the learned gain modifications to be repeatable from test to test since they become inherited by the trajectory.

Trajectories can be created and modified within the Trajectory Editor, but they can also be generated in a spreadsheet program, such as Excel, and imported. The Rotate Vector and Trapezoid buttons are examples of tools to create common trajectories. The Rotate Vector allows the user to create a trajectory where a constant magnitude force vector can be made to sweep across a plane and vary the components of force in the appropriate directions. This has been utilized in hip and shoulder experiments. The most common tool used is the Trapezoid button. This allows the user to create a full factorial experiment for any combinations of loads the user desires. The loading profile is a trapezoid where it linearly ramps, holds, and unloads for each desired loading condition.

Section 3.06 Actuator Class

Similar to the Sensor class, the *Actuator* class serves as a hardware abstraction layer to permit a wide range of actuators to be utilized with little or no modification to higher level system software. The *Actuator* class configures 1 to 6-DOF actuators and allows them to be moved to accomplish a certain task. The distinction to consider with the actuator class is that it uses actuators to accomplish something. A single rotary axis can be used to rotate an item. A 6-DOF robot can be used to fully locate an item in space. In theory, a series of rotary actuators can be stacked to create a 6-DOF robot, though the *Actuator* class is not designed to create an actuator group that "stacks" single axis elements to build the 6-DOF system. The *Actuator* class is designed to quickly and efficiently provide access to the different actuator systems so that they can, in a coordinated fashion, accomplish something. In addition, nothing greater than a 6-DOF system is required since it fully defines a rigid body in space. The number of joints may

exceed 6 for these systems, but the degrees of freedom are still 6. For example, a mobile robot with a 6-DOF robotic arm attached to it can place the object at the end effector at any position and orientation in space. Range of motion constraints of the serial arm may require the coordination of motion between the mobile robot and the arm. The optimization algorithms to provide this coordination and divide up the work will take place in a child class of the *6-DOF actuator* class. Actuators do not need to be limited to motion creation devices. They could be digital or analog outputs of a board. They could be relays and switches. They could be heaters and light sources. It should also be noted that due to the nature of the hardware abstraction layer, it permits virtual actuators to be created and utilized in simulation based scenarios. In this way, system software can be developed, and significant debugging completed, without having any of the safety risks or capital costs associated with moving hardware. Figure 30 contains the current class hierarchy for the *Actuator* class.

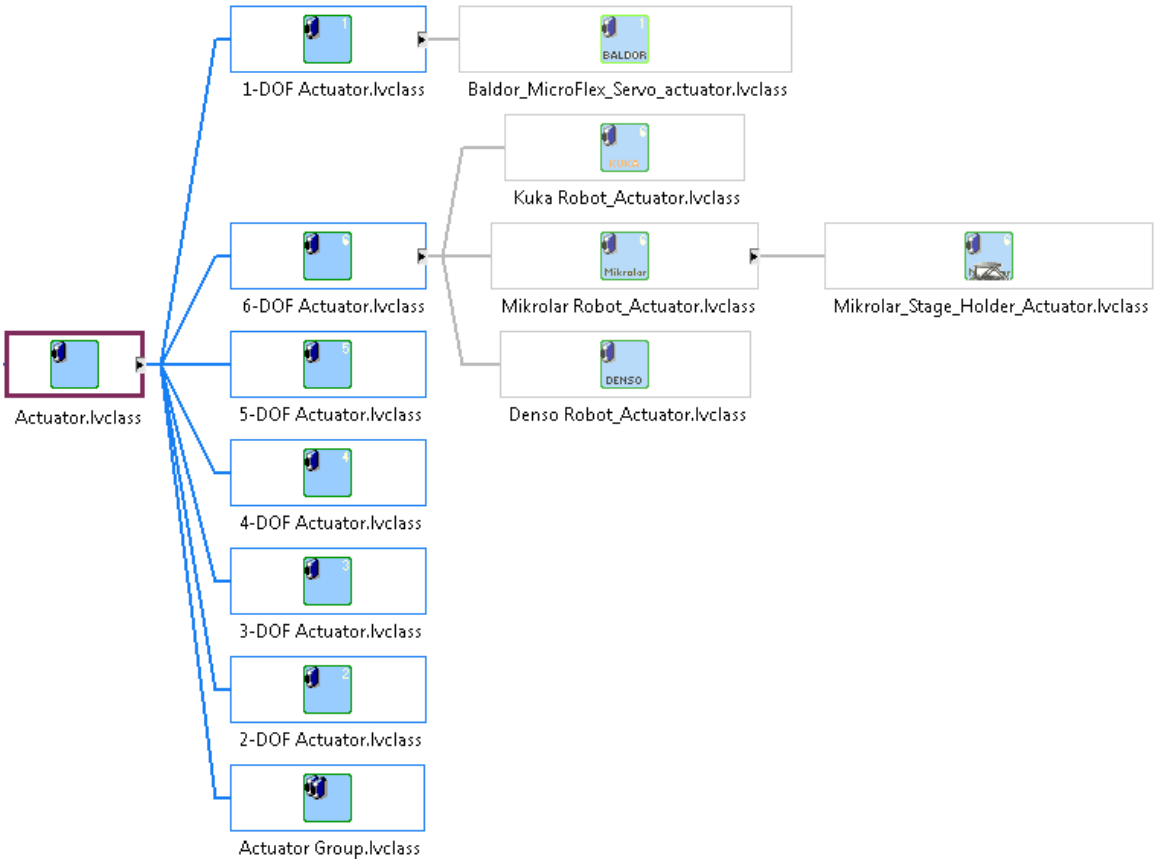


Figure 30: Actuator Class Hierarchy

Some of the key *Actuator* class properties include the number of degrees of freedom, the control type (kinematic or kinetic), the joint type (revolute, prismatic, binary, or continuous), and the coordinate type (cartesian, cylindrical, spherical, or binary). Class methods permit control of the actuator via the joint level (e.g. motor by motor) or by the coordinated motion commands in the selected coordinate type reference frame. This generic framework cannot cover all possible actuators, but hopefully provides enough flexibility to cover a large number of mechatronic systems.

There is a child class of the *Actuator* class called *Actuator Group*. This class is designed to give the UMS, or any other higher level system software, access to additional

actuators that complement the overall behavior of the actuators. In the example above, for the mobile robot with serial arm attached, the *6-DOF actuator* object can place a rigid body at a certain position and orientation in space. However, if the user wants to grasp or release the object, an additional gripper actuator is required. In this case, the *Actuator Group* would contain a *6-DOF actuator* object (mobile robot and serial arm) and a *1-DOF actuator* object (of binary type). The child class of the *1-DOF actuator* object would translate the on-off signals into the appropriate changes to the gripper (e.g. pneumatic valve behavior). The gripper could also be analog in nature and a grip distance could be controlled. In this case, the second object in the group would be replaced with a *1-DOF actuator* object with a Prismatic joint type and Kinematic control type. Alternatively, the gripper could be load controlled (with pressure or load cell feedback) with a Prismatic joint type and Kinetic control type.

Some of the methods in this class use the terminology of "position". The nomenclature is such since in most cases position is the desired outcome of the actuator. However, when the control type is Kinetic, then the "position" name carries a different meaning as it would be considered the kinetic "position" (e.g. force value). Likewise, if it is a Binary control type, the position will be a value of either 0 or 1 where 0 = FALSE and 1 = TRUE.

Section 3.07 UMS Specimen Modules

For the UMS software to be flexible across specimen types, the implementation of each new specimen type will be done using a modular library architecture. Each new

module needs to contain the proper information to acquire spatial relationships and generate coordinate systems. For example, the spine module contains the necessary code to step the user through the process of digitizing the spine anatomy and build coordinate systems compliant with the International Society of Biomechanics. It also contains the code so that the kinetics and kinematics can be recorded and controlled in these same clinically relevant reference frames. In total, the module contains pre-built states, controllers, trajectories, and analysis tools for that particular specimen.

Section 3.08 Hardware Driver Development: Optotrak Drivers

For the sensor hardware abstraction layer, additional work needed to be done to interface with the NDI Optotrak Certus motion tracking system. The LabVIEW drivers that NDI supplied with the Optotrak Certus system provided a basic framework by which to analyze and reverse engineer the communication techniques. However, the code was not in a form that could have been considered drivers, or even used as drivers.

The Optotrak drivers were created to provide a straightforward way for any user of the hardware to connect to the Optotrak Ethernet server and control the acquisition of the data. Communication with the Optotrak server takes place in two distinct forms, transactional and continuous. Transactional communication acquires data at a single point in time. Continuous communication is used to stream data from the Optotrak computer. Figure 31 is an example program to show users of the NDI Optotrak toolkit how to build a simple program to utilize the driver set that was created.

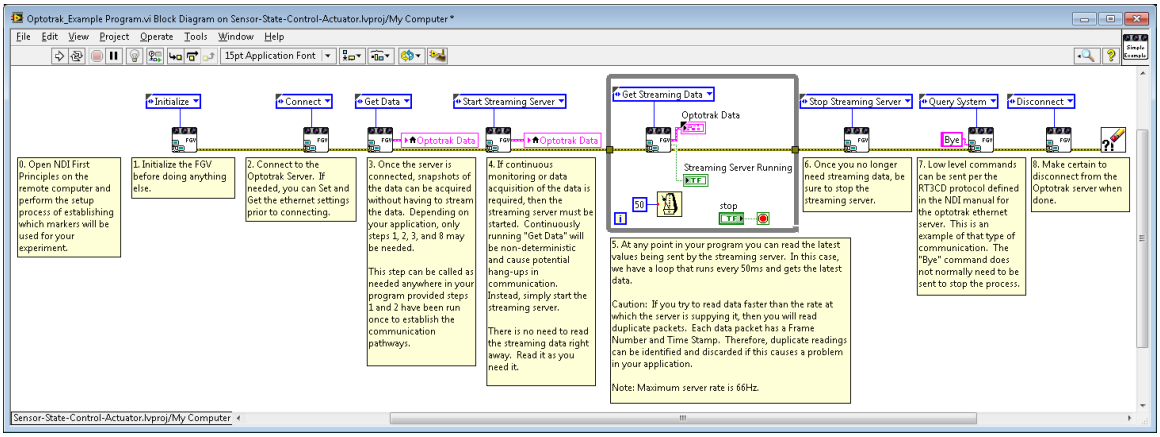


Figure 31: Optotrak Example Program

In addition to the drivers, a high level interface was made for communicating with the Optotrak system. The system was architected and documented in a way to support potential licensing of the toolkit to other users of this hardware. Figure 32 is the Optotrak Advanced Interface.

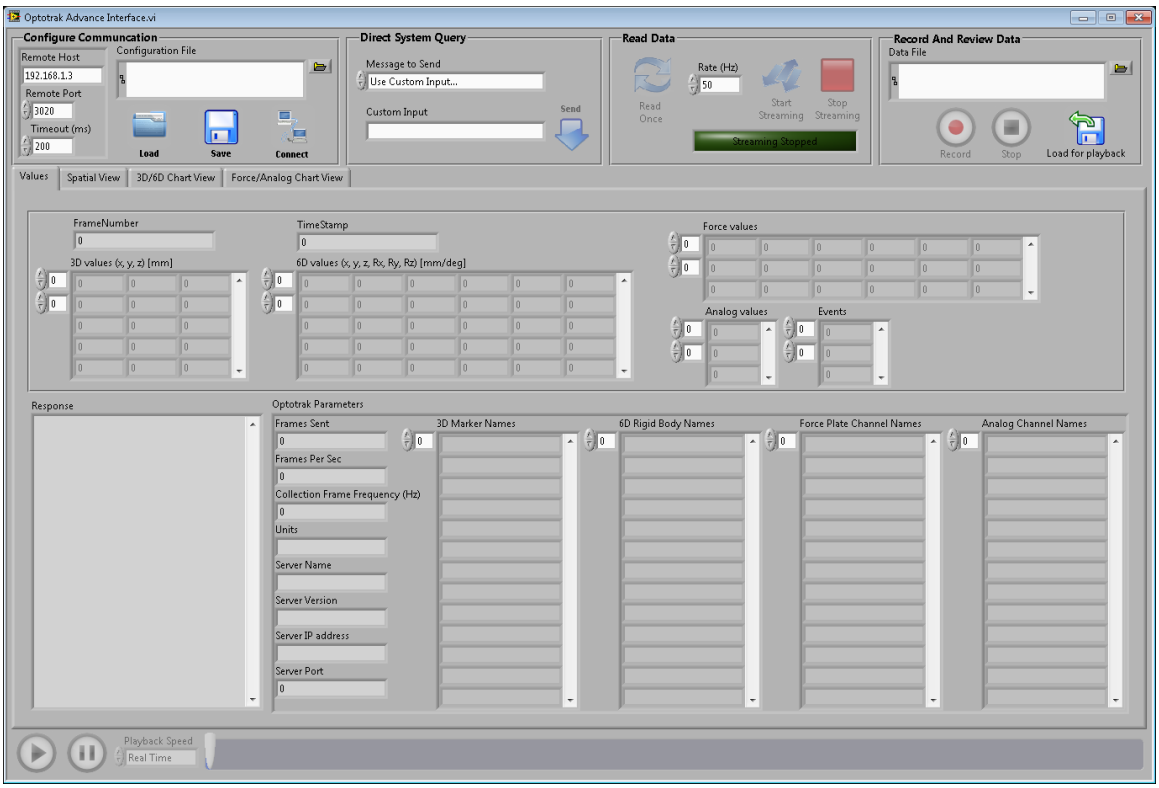


Figure 32: Optotrak Advanced Interface

Section 3.09 Hardware Driver Development: Kuka Drivers

For the actuator hardware abstraction layer, additional work needed to be done to interface with the Kuka KR-16 robot. Significant efforts were made to be able to allow communications between LabVIEW and the Kuka controller. No full functional LabVIEW drivers exist for this purpose, though there is a Matlab based driver set called the Kuka Control Toolbox [96] developed by the University of Siena in Italy. The Kuka Control Toolbox implementation principles are similar to the techniques used by drivers outlined in this section. Another Italian group, Imaging Labs (Lodi, Italy), developed LabVIEW drivers for several commercial robotic systems including Kuka. These drivers handle many important functions in the Kuka communications, though they should be thought of as transactional type drivers only. They are not designed for quick streaming of information. For example, to query the system for current position, it is non-deterministic and could take over 100ms. There is also no direct way to command the robot to a specific position. At a minimum, it requires a two step process of modifying a system variable and then sending a series of commands to run a program on the Kuka controller that calls that variable to determine the desired position. When performing real-time force feedback control, these drivers are insufficient. To overcome this hurdle, custom drivers were created that are based on the Kuka Ethernet Robot Sensor Interface XML (ERX) communication protocol. This communication protocol was developed by Kuka and is a framework for external control. It is based on User Datagram Protocol (UDP) communication methodology and is an exchange of predefined XML messages between the Kuka controller and the remote computer. The messages from the robot controller contain the following information:

```
<Rob Type="KUKA">
  <RIst X="1190.2697" Y="438.8387" Z="1223.4912" A="111.4958" B="38.6900" C="90.1081"/>
  <RSol X="1190.2696" Y="438.8377" Z="1223.4917" A="111.4956" B="38.6897" C="90.1079"/>
  < FTDataSens Fx="0.0000" Fy="0.0000" Fz="0.0000" Mz="0.0000" My="0.0000" Mx="0.0000" />
  < FTDataFrame Fx="0.0000" Fy="0.0000" Fz="0.0000" Mz="0.0000" My="0.0000" Mx="0.0000" />
  <Digin>0</Digin>
  <Digout>0</Digout>
  <Status>0</Status>
  <IPOC>228513818</IPOC>
</Rob>
```

Where:

RIst	=	current actual robot position
Rsol	=	current desired robot position
FTDataSens	=	Output from the 6-axis load cell in the native sensor reference frame
FTDataFrame	=	Output from the 6-axis load cell transformed to the current TOOL reference frame
Digin	=	32-bit integer that represents the current state of digital inputs 1-32
Digout	=	32-bit integer value that represents the current state of digital outputs 1-32
Status	=	Status integer used for communication of various status data
IPOC	=	Interpolation counter

The response messages from the external controller contain the following information.

```
<Sen Type="LABView">
  <EStr>ERX Message! Free config!</EStr>
  <RKorr X="0.0000" Y="0.0000" Z="0.0000" A="0.0000" B="0.0000" C="0.0000" />
  <FTSetpoint Fx="0.0000" Fy="0.0000" Fz="0.0000" Mz="0.0000" My="0.0000" Mx="0.0000" />
  <FTGain Fx="0.0000" Fy="0.0000" Fz="0.0000" Mz="0.0000" My="0.0000" Mx="0.0000" />
  <Command>0</Command>
  <Digout>0</Digout>
  <IPOC>228513818</IPOC>
</Sen>
```

Where:

RKorr = path correction of robot position (change in position relative to robot position when communication first started)

Rsol = current desired robot position

FTSetpoint = Desired setpoint for native Kuka Force/Torque Controller

FTGain = Channel by channel gain for native Kuka Force/Torque Controller

Command = Command integer used to initiate various Kuka controller functions

Digout = 16-bit integer value representing desired state of digital outputs 1-16

IPOC = Interpolation counter (copy of most recent IPOC value received)

These packets are exchanged between the two systems every 12 ms which is dictated by the native control frequency of the Kuka controller. If enough responses from the external system are late, then the Kuka controller stops the program and disables the robot. The IPOC value is a timer signal. The Kuka controller verifies that the response has the IPOC value from the most recent message sent as another way to verify that the external system is still active. Since Windows (Microsoft, Redmond, WA) is a very diplomatic and non-deterministic operating system, this posed a few challenges.

The first step in overcoming these challenges was to purchase components for, and assemble, a computer with sufficient ability based on modern standards. A computer with an AMD 8-core 3.61GHz processor and 16GB of memory running Windows 7 64-bit was selected. Of course, in 2 years this system will be considered slow, but as of when it was purchased in 2012, it was state of the art. The second technique was to use timed loops for the communication loops. These are a programming construct that is a

part of the LabVIEW real-time toolkit but can be used in non-real time situations. Among other things, the timed loops are able to indicate to the Windows operating system a level of priority above any normal function call. The third technique was to utilize a programming construct that stores information in a specific memory location (Fig 33). This is analogous to the use of pointers in the C programming language. Each time an XML packet is read from the controller, the IPOC value is extracted and written into the appropriate location in the response packet (stored in a specific memory location). Then, the message is read from that memory location and returned. The other values in the response packet are also written to that same memory location as they are determined so they can be sent when the whole message is sent. However, the response function does not wait for those values to be updated before sending. The result is that a stale message may be sent, but a message will always be sent. In this way, the Kuka controller is prevented from shutting down the communication if it takes too long to calculate something. This was a key component of making sure the communication was stable and reliable.

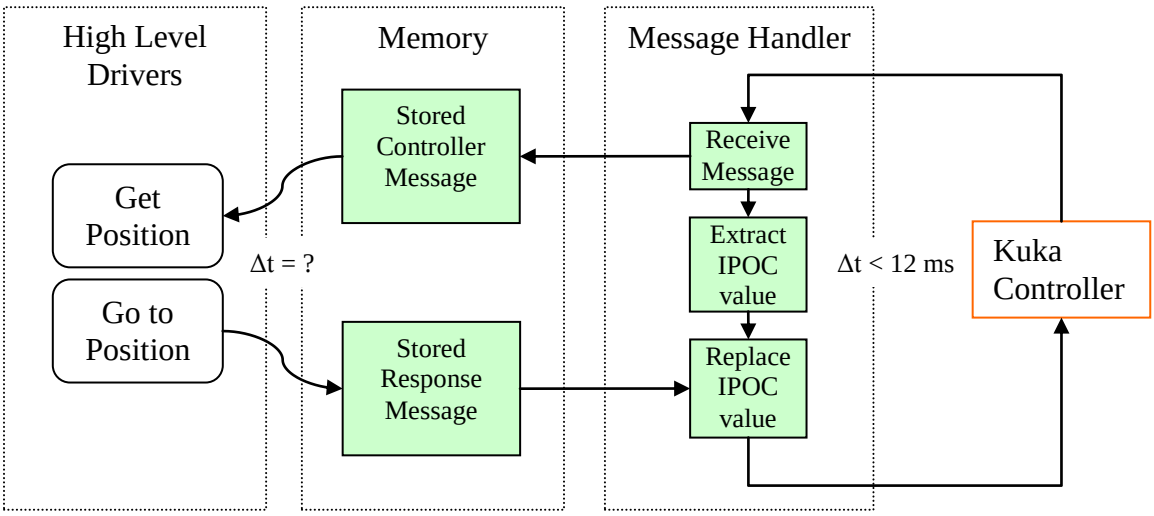


Figure 33: Kuka/Labview low-level continuous communication architecture

A series of functions were built around this “streaming” framework. As a result, the current driver set could be considered a full featured driver set for the Kuka in that it allows for transactional as well and continuous information to be exchanged and the robot can be controlled. In addition to the drivers, a high level interface was made for communicating with the Kuka robot. Figure 34 is the Advanced Kuka Robot Interface that utilizes both the Imaging Labs and Cleveland Clinic drivers for the Kuka robot.

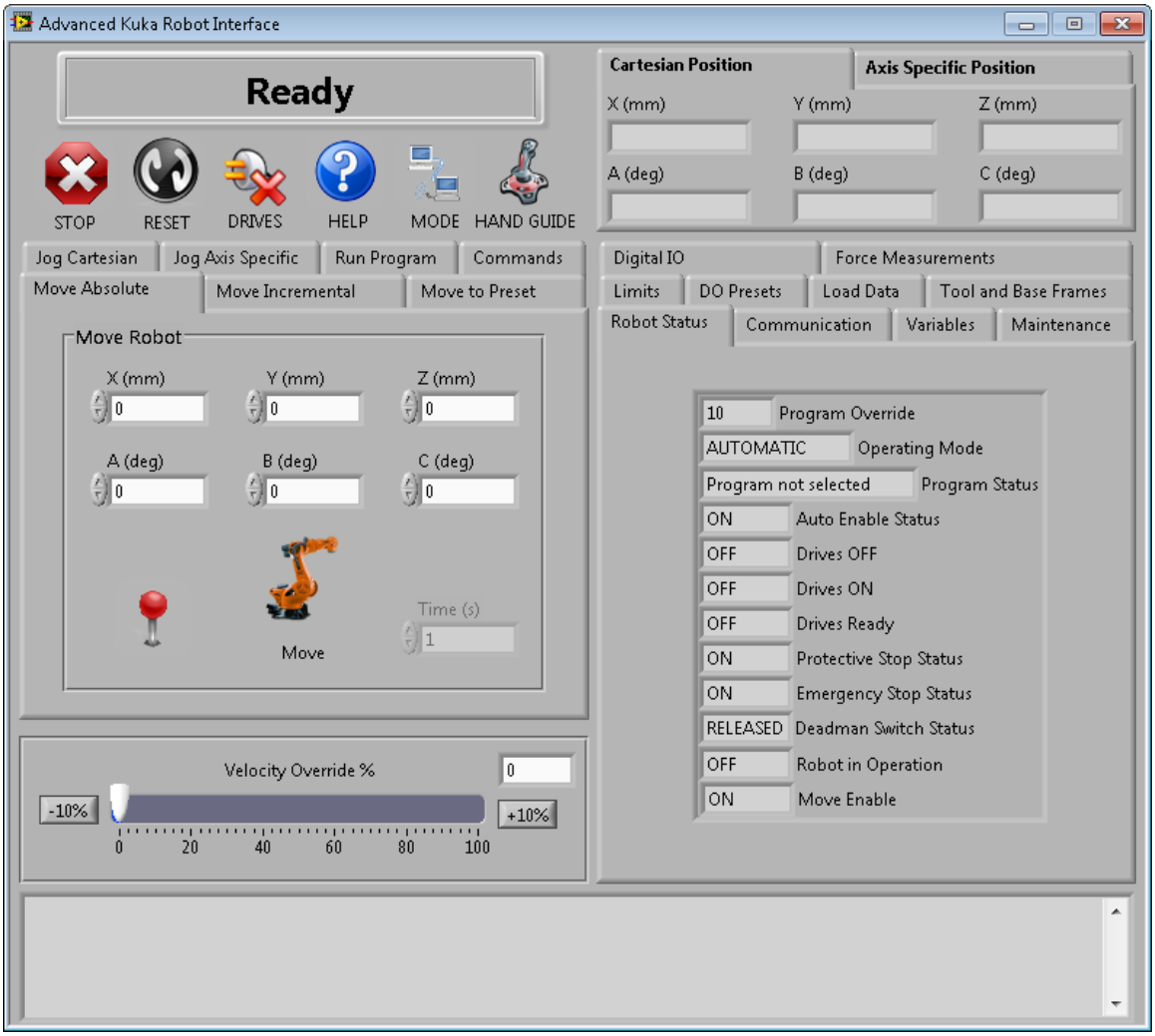


Figure 34: Advanced Kuka Robot Interface

CHAPTER IV
NEUROMUSCULOSKELETAL SIMULATOR CONTROLLER AND NUMERICAL SIMULATION

Section 4.01 Underdetermined Controller

The affect that simulated neural feedback will have on spinal motion is an underdetermined problem. In this particular scenario, the output from a single DOF measurement (facet force) will influence 6-DOF spine motion. Similar to an optimization problem, a set of constraints needed to be placed on the system to derive a unique solution. One could intuitively look at the situation and posit that a move of the superior facet normal to the inferior facet surface would reduce contact force. However, this move could be achieved through either rotation or translation of the superior vertebra. By hand picking the kinematic "solution," it is expected that the scientific value in the experiment will be limited since the result of this pre-ordained solution will be analogous to proving 1=1. The solution we expected is the solution we will get because we programmed it to do so. In order to reduce the influence the system designer had on the kinematic solution, a specimen specific surrogate model and simulation were utilized.

Section 4.02 Specimen Specific Rigid Body Surrogate Model

The conceptual constraint placed on this optimization problem was that the move to "avoid pain" should be the most efficient combination of moves to reduce contact force. It was assumed that a typical human response in a similar scenario would find a similar optimal solution to the problem. It may not be important to find the "proper" *in vivo* response since reasonable responses may produce similar outcomes. Collins' work with comparing common muscle modeling optimization techniques showed that minimal total muscle force, squared muscle force, muscle stress, intra-articular contact force and instantaneous muscle power algorithms predicted remarkably similar patterns of muscle activity over the gait cycle [97]. In this study, the efficient move to reduce contact force was expected to be different depending on the relative positioning of the two vertebrae at any point in time. This efficient move was also expected to be different from specimen to specimen depending on the unique anatomical geometry. To compute what this efficient move needed to be, a specimen specific surrogate model was created and perturbed in simulation. The output of the simulation was a Look-Up Table (LUT) that could be searched for each control loop and the most efficient move extracted.

The model was a rigid body model based on the anatomy digitized during the creation of the state objects. The coordinate systems of interest were the C4 and C5 superior and inferior vertebrae and the facets. The states were the relative kinematics between the C4 and C5 vertebrae, the relative kinematics between the C4 and C5 left facet, and the relative kinematics between the C4 and C5 right facet. The real-time calculation of these states was based on the NDI Optotrak C4 and C5 position markers

placed in the bones. These sensors were also used to determine the appropriate geometric relationships for the model development. Appendix B contains the explicit definitions for these coordinate systems.

The LUT was a 6 dimensional array where each dimension represented a kinematic degree of freedom (3 translations, 3 rotations) and the value stored in the array was an estimate of the contact force. To estimate this force, the distance from the origin on the superior facet to the plane of the inferior facet, along an axis normal to the plane, was calculated. The distance was squared (to estimate non-linear contact forces) and multiplied by an estimated stiffness of 100 N/mm.

$$F_{facet} = \begin{cases} kd^2 & if\ d < 0 \\ 0 & if\ d \geq 0 \end{cases} \tag{5}$$

Where:

d = distance from the origin on the superior facet to the plane of the inferior facet

k = estimated stiffness

Next, the relative relationship between the C4 and C5 vertebrae was mathematically perturbed and the resulting change in the facet kinematics and estimated contact force was computed. By repeating this process for a range of possible C4-C5 kinematic values, the 6 dimensional look-up table was built. The grid size used for the table was 9 elements spread over 15 mm, for translations, and 15 degrees, for rotations. This yielded a total of 9^6 = 531,441 unique table values. This number makes clear the need for a model to be built. To empirically build this table would be extremely time

consuming and labor intensive. In addition, the large number of loading cycles required to build this model would cause tissue degradation and thus reduce the usefulness of the specimen specific features of the model.

To determine an efficient move to reduce facet contact force, the controller used the current C4-C5 position, based on the Optotrak markers and the state coordinate system transformations. It then located the set of LUT values most representative of this current kinematic position. Next, it identified the estimated facet force for the nearest neighbors corresponding to changes in each kinematic degree of freedom. From this, the steepest ascent vector could be identified. The steepest ascent vector was the combination of translations and rotations that most dramatically increased the estimated facet contact force. It was this vector that was used to drive the direction of spine motions as a result of facet contact forces. Intuitively, the steepest descent, not ascent, vector would reduce facet force. The distinction is related to the signs in the controller algorithm where a negative force error multiplied by this vector would go in the opposite direction of the steepest ascent, presumably the steepest descent. During the control loop, the steepest ascent vector is extracted from the LUT. It is then normalized since the purpose of the vector is to guide direction rather than control the magnitude of the motion. The error between the actual and desired facet force values is multiplied by the normalized steepest ascent vector to produce 6-DOF motion of the spine (Equation 6). Figures 35-40 contain screen shots from the various steps in the process of configuring the controller.

$$\vec{\dot{X}}_{Mechanotransducer} = Gain \cdot F_{facet} \cdot \frac{\vec{\dot{X}}_{LUT}}{\left|\vec{\dot{X}}_{LUT}\right|} \tag{6}$$

Where:

$\vec{\dot{X}}_{Mechanotransducer}$ = JCS velocity vector due to mechanotransducer controller

$Gain$ = Mechanotransducer controller gain

F_{facet} = Facet force as measured by the sensor

$\vec{\dot{X}}_{LUT}$ = Steepest ascent vector based on LUT

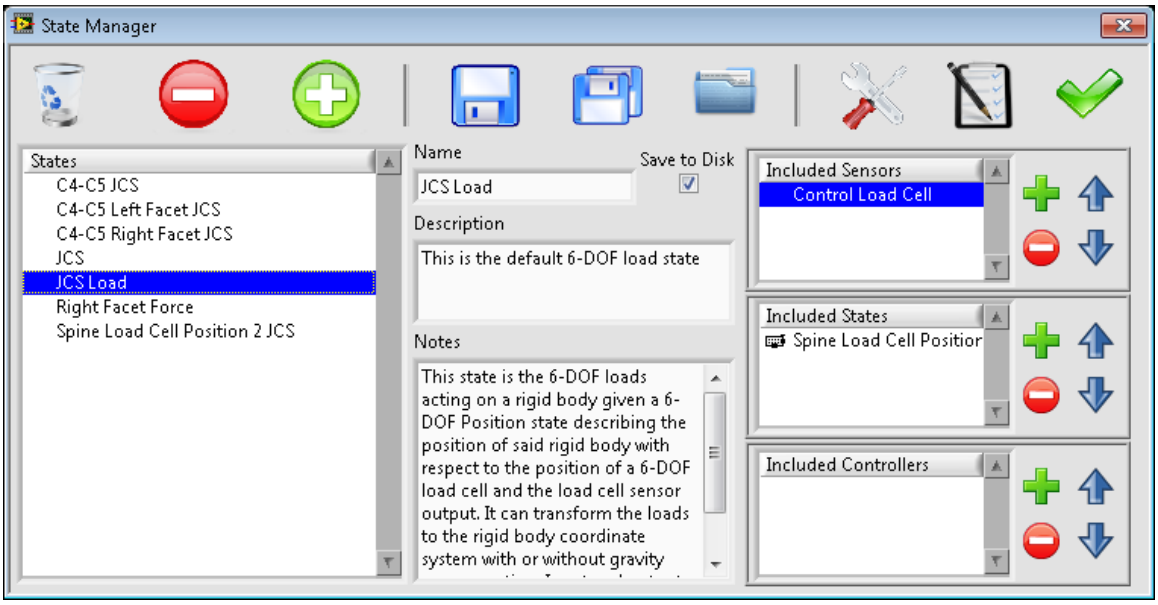


Figure 35: Spine Facet Mechanotransducer Kinematic States

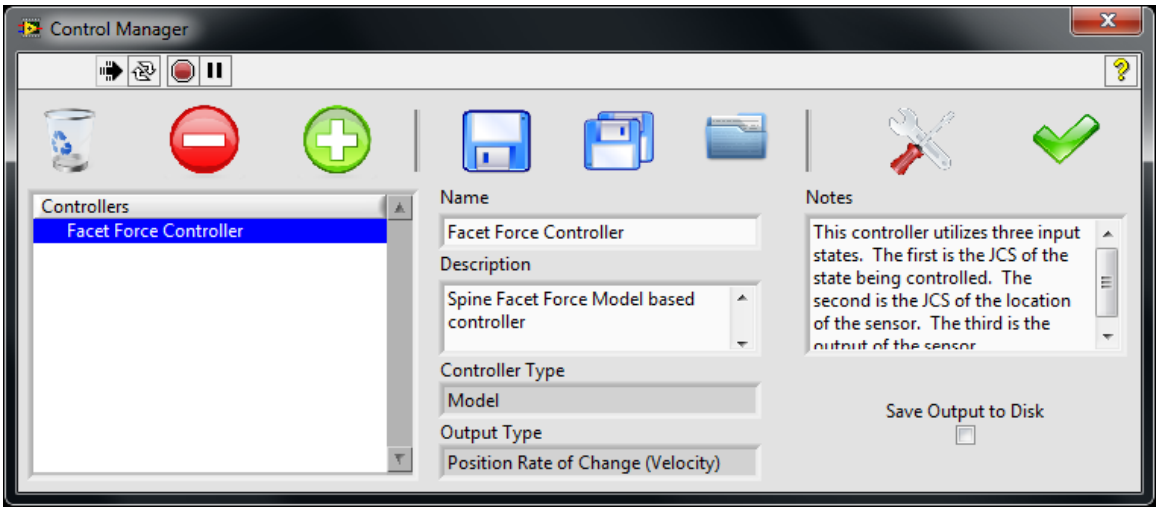


Figure 36: Facet Force Controller added to controller manager

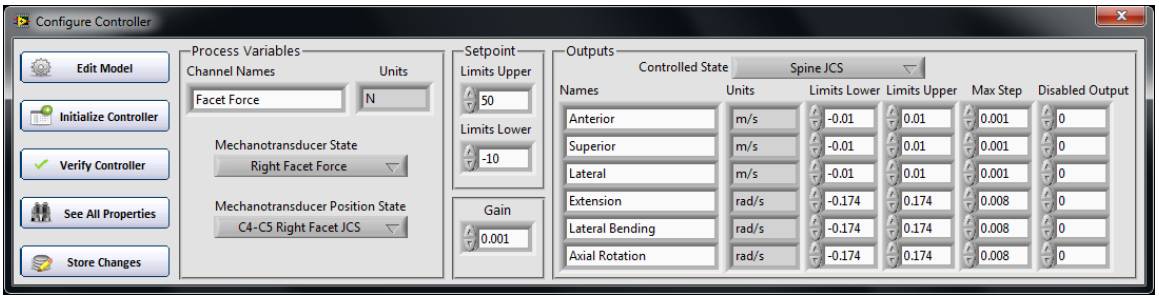


Figure 37: Facet Force Controller Configure Screen

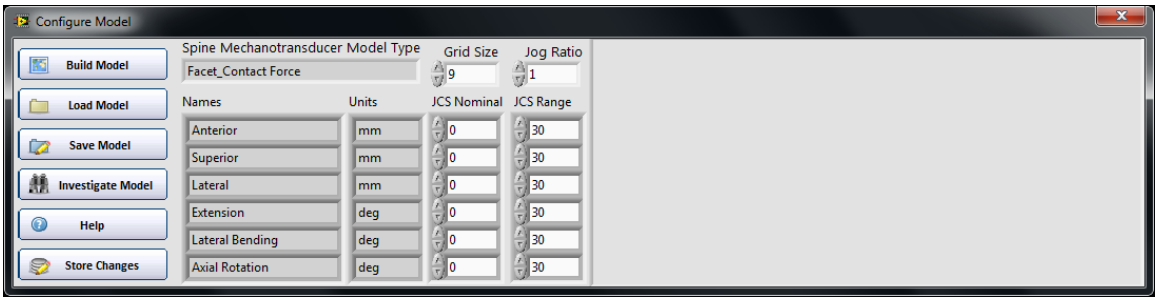


Figure 38: Facet Force Controller Model Parameters Screen

To build the model, the parameters include the grid size, the jog ratio, the nominal JCS value, and the JCS range of values to create kinematic perturbations. The jog ratio is the ratio between translations and rotations for the steepest ascent vector. When the vector is normalized, there exists a bias between the translations and the rotations. This

bias is a result of the differences in units and the fact that they are two different types of motions, which have different effects on the estimated output of the facet force. In the LUT, the translations are in units of meters and the rotations are in radians. The jog ratio value is necessary, though it is a hand selected value that will influence whether the controller favors changes in translations or rotations when modifying the facet force. The simulation section contains more details on how the value was selected.

Once the model parameters were defined, the model was built and the LUT populated by simulating all the unique JCS positions as defined by the parameters. In this case it was 531,441 positions.

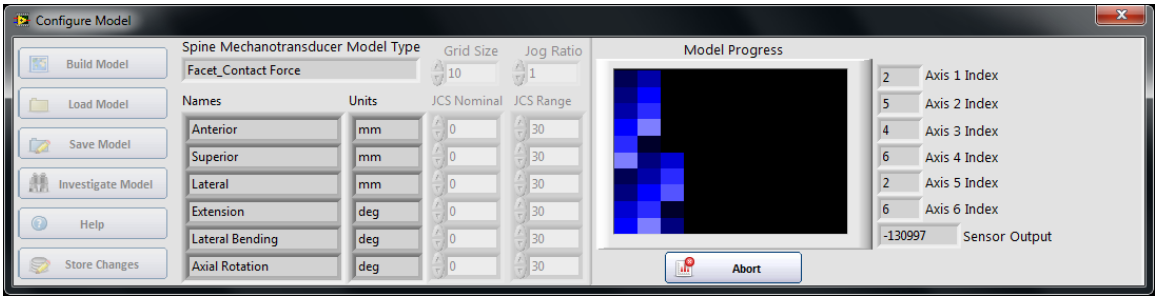


Figure 39: Facet Force Controller Build Model Screen

The LUT is a 6 dimensional matrix. As a result, it is hard to visualize all aspects of the matrix simultaneously. However, it is important to provide the user some visual feedback to ensure the model provided reasonable predictions that can be utilized for the controller. Figure 40 contains several screen shots of the representative 3D graph provided to the user.

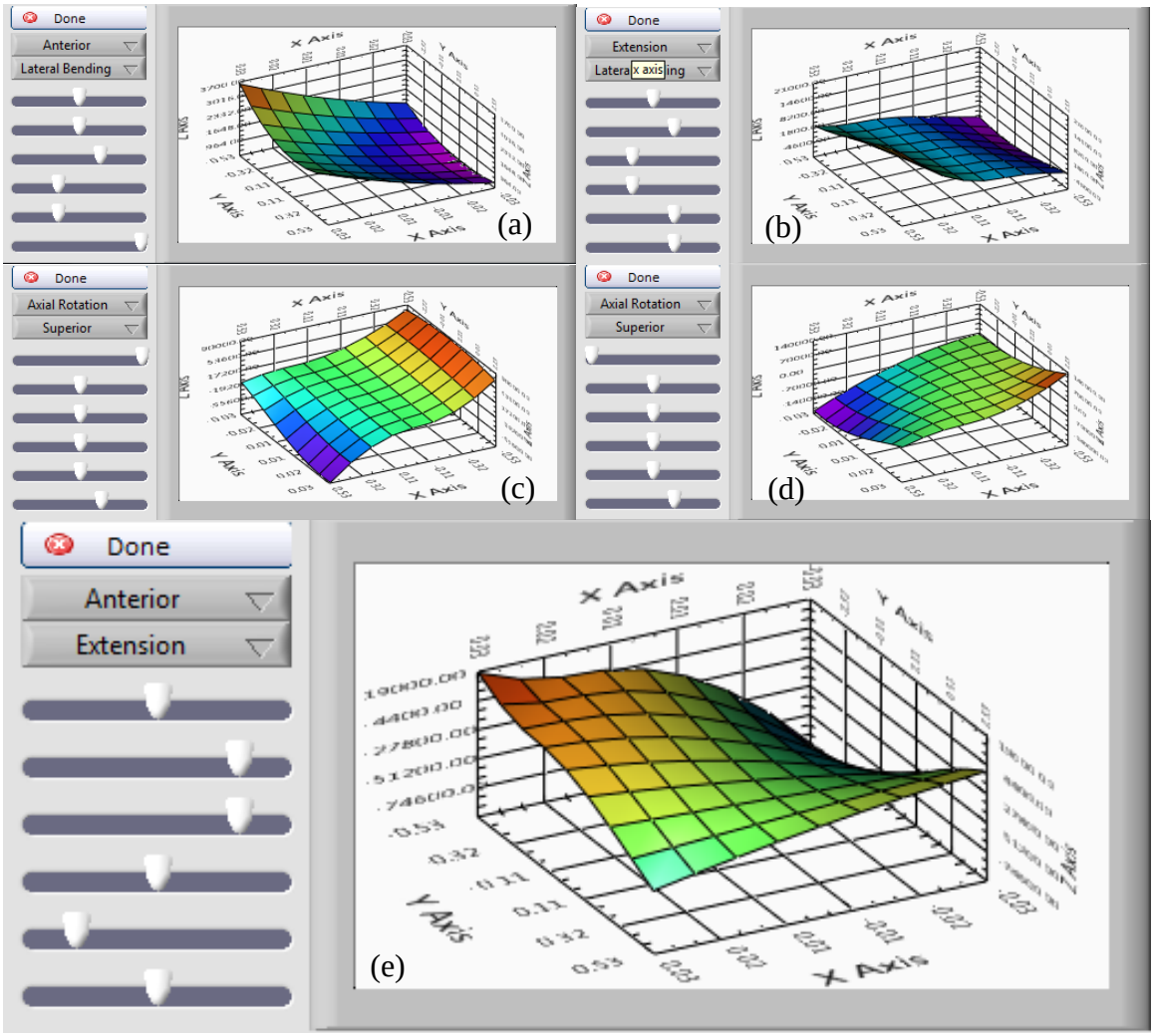


Figure 40: Facet Force Controller Investigate Model Screen

The z axis is the estimated facet force. The x and y axes can be selected as independent dimensions of the LUT. In Figure 40e, they were selected to be the anterior translation and the extension rotation. The sliders beneath the axes selectors allow the user to modify other kinematic values in the LUT and extract the 3D graph at those selected kinematic positions. Figures 40a and 40b have the same channels selected for the x and y axes (anterior and lateral bending), but the sliders have been adjusted to view different regions of the LUT. These suggest the complexity of the facet response.

Figures 40c and 40d are likewise configured for the axial rotation and superior axis and show the differences in the response with the anterior slider at the two ends of the range. The shape of the curves in Figures 40a-e suggest that the motions are coupled, and it makes an extremely strong case for why a model is needed to provide the controller with the steepest ascent vector. For example in Figure 40e, in the anterior translation, direction (x-axis) positive translations (to the left) are estimated to increase the facet force at a range of extension values (far side of graph). At the other end of the extension range (near side of graph), the same translation may produce the opposite effect on facet force. Clearly, a one-size-fits-all vector would produce results that may be ineffective or inconsistent.

The process of developing this specimen specific surrogate model technique, as expected, sometimes yielded results that did not work or were not ideal. The following are several lessons learned during the process that can help others to avoid similar pitfalls. The first change I needed to make to the model was to permit tension in the estimated facet force equation (5).

$$F_{facet} = \begin{cases} kd^2 & if\ d < 0 \\ -kd^2 & if\ d \geq 0 \end{cases} \tag{7}$$

The new equation (7) produces a physically impossible situation where the facet surfaces can support a negative compression force (tension). Recall that the purpose of this model and the LUT is to provide a direction to move the spine to unload the facet. Also, there is no guarantee that the model kinematics and experimental kinematics, as it relates to facet force, will be in perfect agreement. If the controller searches the nearest neighbors in the LUT and finds that they are all zeros, then the steepest ascent vector is 0

and the controller has no solution for the move. It is likely that small numerical round-off errors will be amplified when that particular vector is normalized, and thus the suggested direction will be random and meaningless. As such, tension was permitted in the model in order to produce a reasonable solution in those neutral zone positions where the model and experimental contact/no-contact conditions may be different.

The second refinement was to determine a reasonable grid size. Original versions of the LUT used 13 elements which yielded 13^6 = 4.8 million unique table values. The time required to search the LUT during each control loop increased noticeably and did not show much change in the predicted steepest ascent vector. This was likely due to the model containing low frequency response surfaces (Figure 40). Reducing the grid size to 9 produced a smaller LUT and was sufficiently fast to search and compute the appropriate control values for each control loop (~10-15 ms).

The third refinement was to consider the physiological reality of the steepest ascent vector solution. In fact, the largest component of the vector to reduce facet force was a superior translation that pulled the vertebrae apart. Unfortunately, this solution is not physiological. There are no muscles in the neck that can pull the head up. Secondly, while the muscles can co-contract and create compression, this solution was also not permitted. This was to prevent a compressive ratcheting situation because distraction was not permitted. In addition, for this particular implementation of the NMS where facet force was trying to be reduced, it was unlikely to find a solution where more

compression would be better. As a result, the controller output limits for the superior translation channel were reduced to values near zero (Figure 37).

Section 4.03 Controller Numerical Simulation

When designing any new controller, it is always important to test it in a simulation prior to running it on real hardware. Not only does it protect the operator and hardware for safety and financial reasons, but it also allows for quick iteration, preliminary estimates of controller gains, and potentially easier troubleshooting of the control algorithm. The simulation that was run utilized all the software control tools described in previous chapters. It was also based on the specimen specific digitized geometry of a cervical Sawbones specimen (Pacific Research Laboratories Inc. Vashon, Washington, USA) (Fig 41).

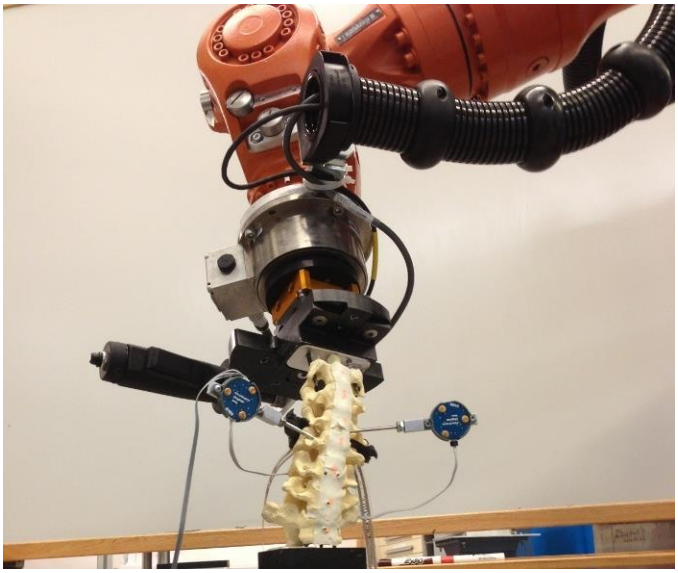


Figure 41: Cervical Sawbones Specimen

The vertebrae and facets were digitized using the NDI Optotrak motion tracking system, and the appropriate coordinate systems were generated. The model was built

based on the right facet geometry, and the LUT was generated and stored as a property of the facet force controller object. The simulated specimen was an FSU whose geometry was based on the digitization process, and whose mechanical properties were defined as a simple diagonal matrix spring model. Due to the slow speed of the spine testing (quasi-static nature of how loads are applied), there were no equations of motion or mass based terms included in the simulation. System dynamics were simulated as time delays between each control loop as the simulation was run in real time. The load was controlled by a proportional controller. The simulation allowed for manual perturbations, manual steady state changes, and automatic execution of a trajectory. Figure 42 is the control diagram for the simulation.

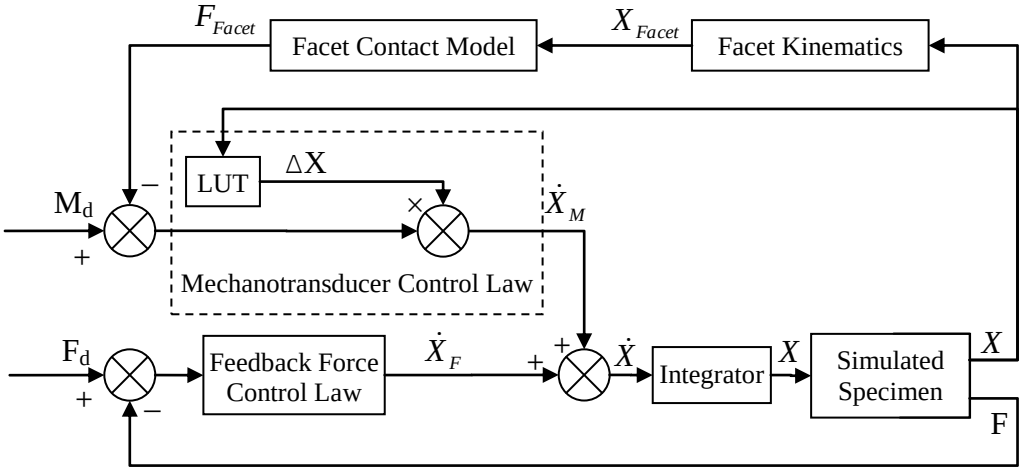


Figure 42: NMS simulation control diagram

Where:

M_d = desired mechanotransducer output (facet force)

F_d = desired forces in inferior facet (C5)

X = specimen kinematic JCS position (C4-C5)

X_{Facet} = simulated facet kinematics based on relative position of JCS

LUT = Look up table search algorithm

ΔX = steepest ascent jog vector based on nearest neighbor search of LUT

$\dot{X}_M$ = JCS velocity from mechanotransducer controller

$\dot{X}_F$ = JCS velocity from feedback force controller

$\dot{X}$ = sum of all controller outputs

F_{Facet} = estimated facet force based on contact model (Eqn. 5)

F = estimated loads in specimen JCS

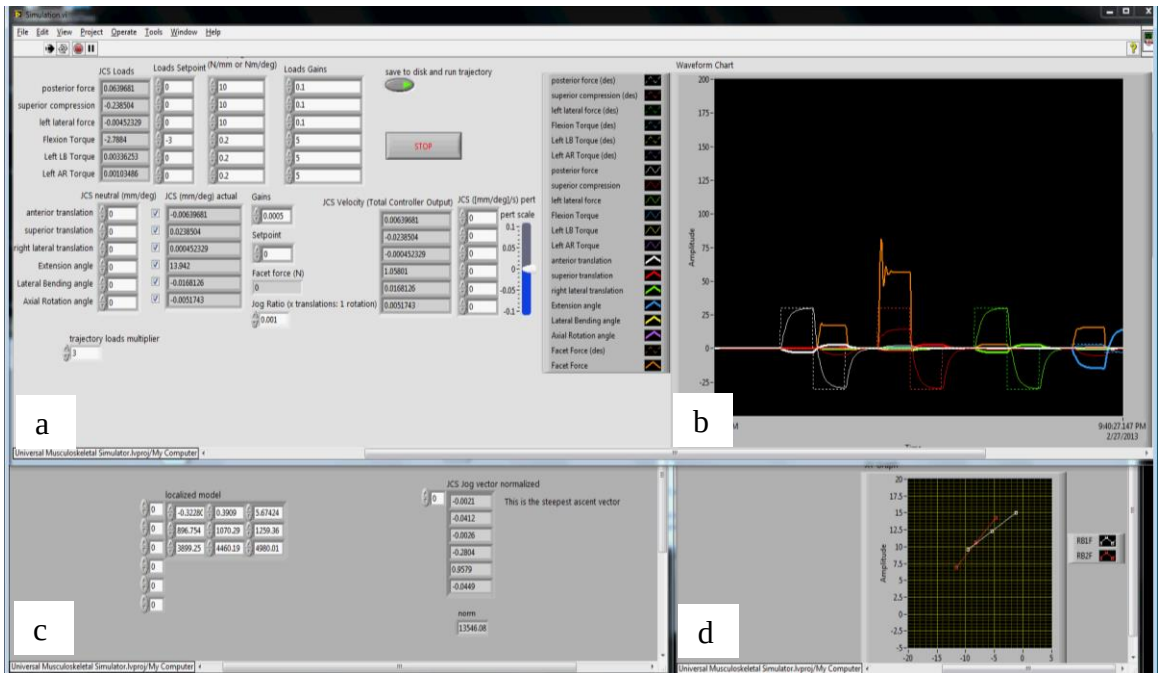


Figure 43: Facet Force Controller Simulation Screen a) simulation control parameters, b) simulated JCS loads and facet force, c) nearest neighbors from LUT resulting in the steepest ascent vector, and d) facet contact surface planes.

Figure 43 shows 4 key screens of the simulation process. Figure 43a contains all the control parameters for the simulation. Note the value of the jog ratio of 0.001. This value was selected by manually perturbing the simulated system and seeing which ratio caused roughly similar responses in the facet force change. My goal was to limit any

cognitive influence on the solutions that the controller found, though this was a tuning parameter that needed defined. As a result, I settled on changing it by order of magnitude differences only. Once this value was selected, it was used consistently throughout simulation and experimentation. It should also be noted that similar perturbations were tried in the experimental setup with cadaveric specimens and the 0.001 value also produced similar satisfactory responses as compared to other values. During the simulation the relative kinematic relationship of the facets could be visualized in the sagittal plane (figure 43d). Each simulated control loop the LUT was searched to identify the steepest ascent vector. Figure 43c shows the nearest neighbor matrix and the resulting vector.

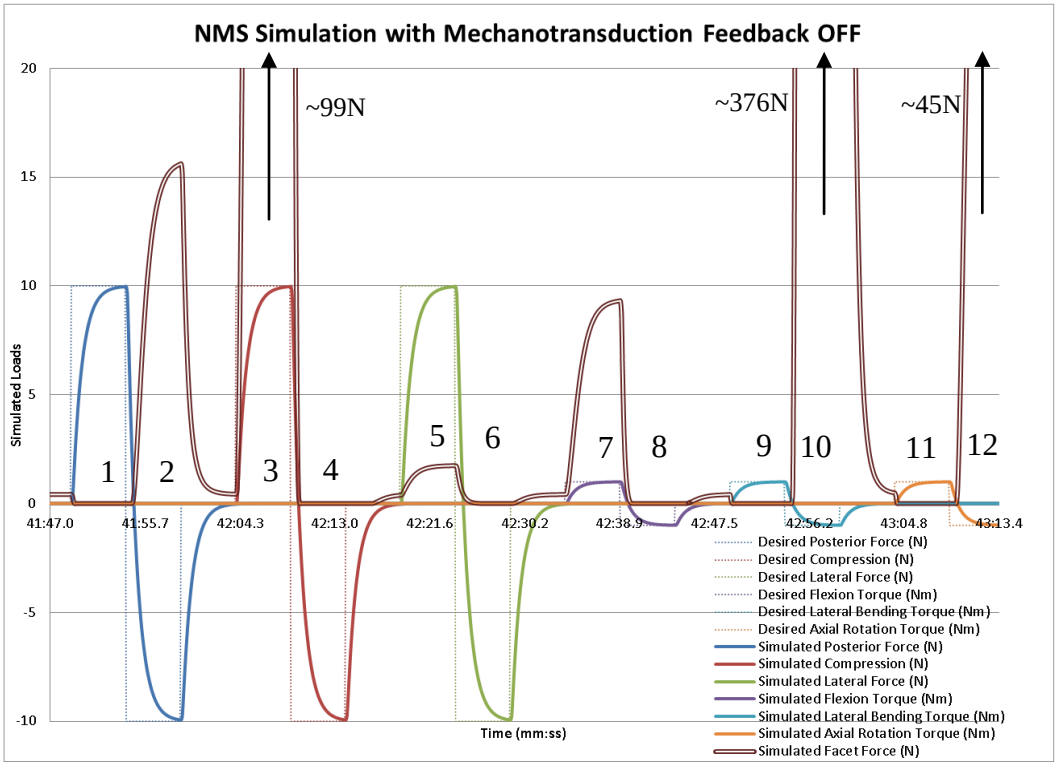


Figure 44: NMS simulation with Neural Feedback Off

The first simulation performed was with the neural feedback turned off (Fig 44).

A trajectory that loaded each degree of freedom individually was created and 12 distinct loading conditions were applied. The forces were perturbed ± 10N, and the torques were perturbed ± 1Nm. The load controller sought to achieve the desired loads by moving the spine and, as a result, loads were transferred across the facet. The facet loads occurred in the JCS loading conditions that one would expect increased loads on the right facet:

- (2) superior vertebra pushing anteriorly on inferior vertebra
- (3) superior vertebra pushing down compressing the inferior vertebra
- (10) right lateral bending torque
- (12) left axial rotation torque

The next simulation was to turn on the neural feedback controller and see if the estimated facet contact forces were reduced (Fig 45).

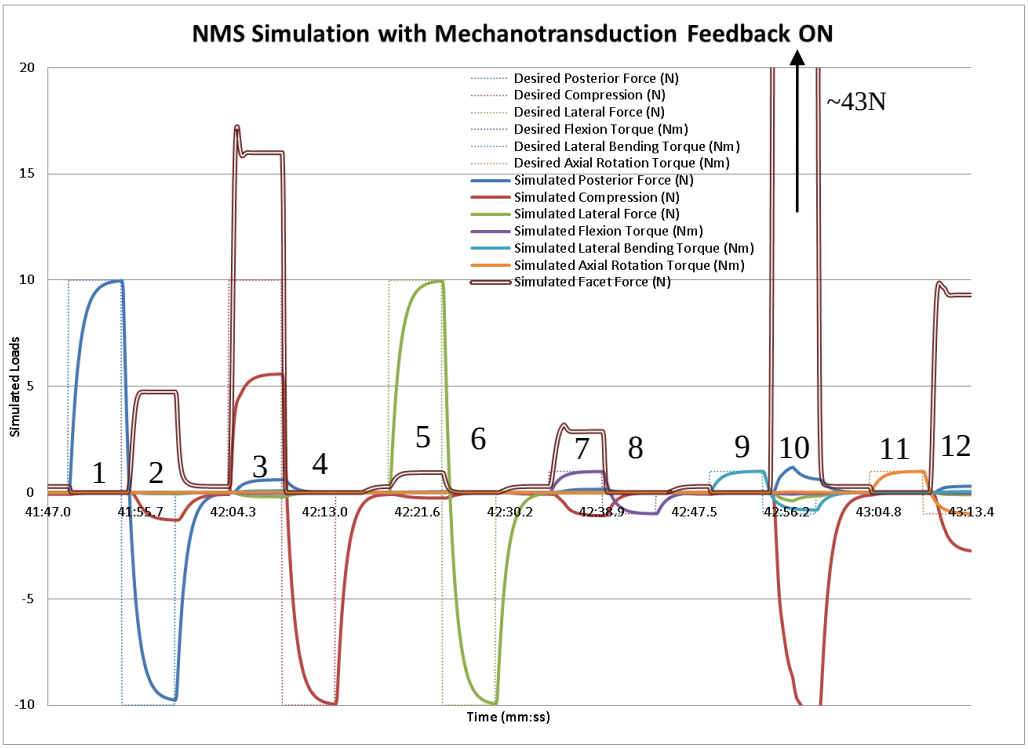


Figure 45: NMS simulation with Neural Feedback On

The neural feedback controller reduced the peak facet forces in all cases. In addition, other off-axis loads increased as part of the kinematic compensatory solution to reducing facet force. In some cases (conditions 2, 3, and 10), the applied loads were also reduced as part of the compensatory solution. This suggested that the neural feedback controller was correctly predicting a steepest ascent vector estimate which was able to reduce facet force. Other variations of this simulation were performed, all with similar results. The gains were modified, and the applied loads were increased. The effect was always a balance of the control system parameters. If the neural feedback control gain was reduced, the facet force would be higher. Likewise, if the proportional gains on the load controller were reduced, the facet force would drop more.

As with any model there are always limitations. This was no exception. This model had no mass based dynamics, no damping, and an oversimplified plant (spine linear spring model). Some of these limitations can be seen in (Figure 44) where the facet contact force of 99 N could be simulated while the overall compressive spine force was only 10 N. This is clearly an artifact and limitation of the simplified spring model, not including estimated facet forces. Thankfully, the point of the simulation was not to estimate spine loads. The objective was to determine if the controller was able to reasonably perform the task of reducing the facet contact force. To this end, it worked. The simulation served as a pipeline from digitizing the anatomy, to building the LUT using the model, and, finally, running the controller that estimates effective compensatory kinematics for the "problem" of high facet force. The results suggest that the pipeline was a good baseline to use for the cadaveric experimentation.

CHAPTER V

CADAVERIC EXPERIMENTATION METHODS

Section 5.01 Overview

The first aim of this study was to develop a robotic neuro-musculoskeletal simulator for cadaveric spine testing which incorporates data from external sensors (e.g. pressure, force, motion tracking) to modulate spinal motion and loading patterns. The second aim was to demonstrate the ability of the neuro-musculoskeletal simulator to use an estimated nociceptive response in unilateral facet arthritis to modulate the motion pattern and elucidate subsequent injurious loading conditions that could lead to contralateral arthritis.

Cadaveric cervical specimens (n = 7) with a mean age of 63.6 years (ages 59-69, 1 female and 6 males) were dissected down to the osteoligamentous structures for C2-T1. Custom fixtures were used to secure the specimens to the robot. Pedicle screws were driven into the most proximal and distal vertebrae, and drywall screws were inserted into the C2 and T1 endplates. The pedicle screws were screwed to rods that were part of the

fixture, and the heads of the drywall screws were embedded in Woodsmetal (a low melting point bismuth and lead alloy). ATI Industrial Automation Inc. (Apex, NC, USA) MC-16 manual tool changers were incorporated into the fixture so that the specimen could be accurately re-mounted between surgeries if the surgery could not be performed with the specimen still on the robot. For this particular experiment, no remounting was required.

Figure 46: Mounted Cadaveric Specimen

Optotrak markers were placed in the C4-C5 vertebral bodies. The NMS controller sensors and states were initialized, and the loading conditions were input into the Trajectory Editor. The thin film force sensors were inserted in both the left and right C4-C5 facets joints per the methodology described in section 5.03. The neural feedback

was assigned to the right facet, and the loading conditions were executed. Data from all sensors and states was recorded throughout the experiment.

Section 5.02 NMS controller

To perform the testing on cadaveric specimens, the native UMS hybrid control algorithm (Fig 28) was combined with the mechanotransducer control algorithm (Fig 42) and the position control algorithm of the robot.

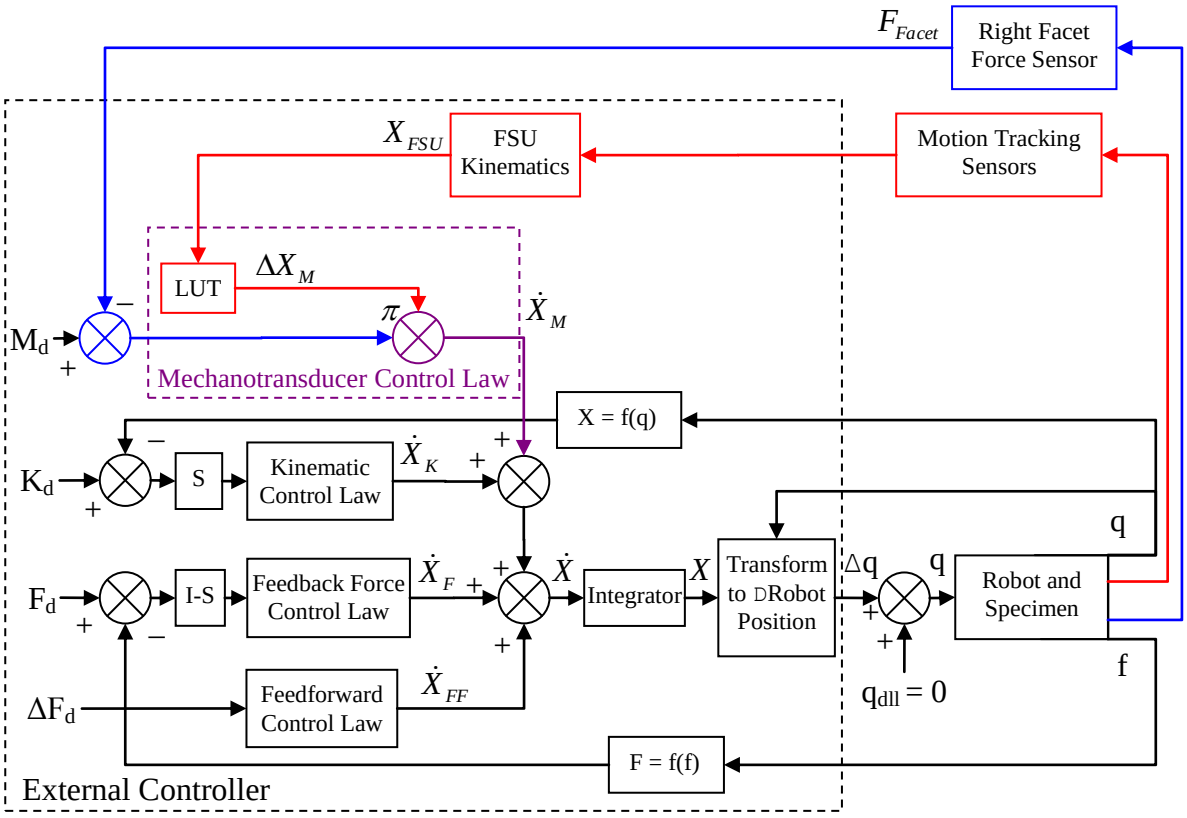


Figure 47: NMS Control Diagram

Where:

q = robot pose in joint space

q_{dll} = desired position at the low level controller

Δq = sum of all changes in robot pose from external controller

f = output of force transducer

F = transformed output of force transducer in specimen JCS

F_{Facet} = measured right facet force based on thin film force sensor. This is the pathway for pain-modulated motion from a force sensor.

M_d = desired mechanotransducer output (facet force)

K_d = desired JCS kinematics in specimen coordinate system of interest

F_d = desired forces in specimen JCS

S = diagonal matrix such that when $s_j = 0$ the axis is position controlled, and when $s_j = 1$, it is force controlled. Values between 0 and 1 are also valid and provide simultaneous balance between the two control schemes.

I = identity matrix

X = specimen kinematic JCS position

X_{FSU} = functional spinal unit kinematics based on motion tracking sensors

ΔX_M = steepest ascent jog vector based on nearest neighbor search of LUT

$\dot{X}$ = sum of all controller outputs

$\dot{X}_M$ = JCS velocity from mechanotransducer controller

$\dot{X}_K$ = JCS velocity from kinematic controller

$\dot{X}_F$ = JCS velocity from feedback force controller

$\dot{X}_{FF}$ = JCS velocity from feedforward force controller

LUT = look up table search algorithm

Note that two position control schemes exist. The main scheme is the one which resides on the low level Kuka controller and handles all the task and joint space conversions to servo the motors to maintain stable control of the robot. The external position controller handles the desired kinematics in the JCS which were created by the trajectory editor. For this reason, and to avoid confusion, it is considered the kinematic control loop. For the low level position controller, the desired position is based on the Kuka desired position ($q_{dll}(t) = 0 \; \forall \; t$) added to the sum of all changes in robot pose ($\triangle$q) sent from the external controller. The $\triangle$q value, in Kuka programming parlance, is the path correction.

The output from the hybrid kinetic and kinematic control feedback controllers is added to the output of the mechanotransducer controller object to produce the change in JCS position that is most likely to satisfy all the system constraints, including the desired loads and the facet force. Similar to the effect seen in the simulation, the relative gain of each term of the controller has the ability to influence the relative importance of each constraint on the system. Though drastically oversimplified, this may not be unlike *in vivo* responses to painful stimuli. Our high level controller (our brain) may cause us to apply the right muscle loads to our neck to achieve a goal, such as "look that way." We may also have a competing controller that can modulate this motion based on the nociceptive sensations. If something is more painful, the muscle loads and corresponding kinematic changes will create an overall different loading state to try to satisfy both high level goals. The control algorithm presented allows for the exploration of the relative importance of these goals by modifying the gains of each term.

Though the facet force drives the magnitude of the controller output, similar to the way action potentials from a nociceptor may, it should also be noted that this implementation of the NMS includes proprioceptive feedback as well. In Figure 47, the blue feedback represents the response of the mechanoreceptors, and the red feedback represents the proprioception. Ultimately, the mechanotransducer control law relies on both types of information to find the best possible kinematic compensatory solution to satisfy the desired loading state.

For the purpose of this experiment, only the force feedback and the mechanotransducer control laws were used. These represent the intent of the brain to achieve certain physiological loads while also reducing facet "pain". The feedforward and the kinematic control laws are a built-in component of the UMS software and though they were not utilized for this study, they could be enabled in future implementations of the NMS.

Section 5.03 Sensor and State Initiation

The following sensor objects were created in the UMS Sensor Manager:

1. C4 Position (Optotrak marker in vertebra)
    a. Standard x, y, z, roll, pitch, yaw convention for transformation
2. C5 Position (Optotrak marker in vertebra)
    a. Standard x, y, z, roll, pitch, yaw convention for transformation
3. Control Load Cell (Attached to the end of the robot)

a. Standard F_x, F_y, F_z, M_x, M_y, M_z convention for transformation

4. Digitizer (Optotrak probe)
    a. Standard x, y, z cartesian convention for position
5. Left Facet Force (Flexiforce sensor)
6. Right Facet Force (Flexiforce sensor)
7. Robot Position
    a. Standard x, y, z, roll, pitch, yaw convention for transformation

Utilizing the UMS State Manager, the following states were created based on the signals from the sensors listed above. See Appendix B for specific anatomical locations for digitization of coordinate systems.

1. JCS (relative position of the C4-C5 vertebrae based on position of robot)
    a. The spine JCS contains the following components:

a	anterior translation of proximal vertebra
s	superior translation of proximal vertebra
l	lateral translation of proximal vertebra (right is positive)
β	lateral bending (tilt right is positive)
γ	axial rotation (head turn to the left is positive)
α	extension

2. JCS Load (loads expressed in the C5 reference frame)
    a. The spine loads contains the following components and are the loads the superior vertebra applies to the inferior vertebra (i.e. C4 pushing on C5):

F_P	posterior shear force
F_C	compression force
F_L	lateral shear force (left is positive)
M_{LB}	lateral bending torque (tilt left is positive)
M_{AR}	axial rotation torque (head turn to the right is positive)
M_F	flexion torque

3. C4-C5 JCS (relative position of the C4-C5 vertebrae based on Optotrak markers)
    a. The C4-C5 JCS contains the same components as the JCS state
4. C4-C5 Right Facet JCS (relative position of the C4-C5 right facets based on Optotrak markers)
    a. The spine facet JCS contains the following components:

a	anterior sliding of superior facet
s	axial separation of superior facet (distraction is positive)
l	lateral shear of superior facet
t	tilt (tilt right is positive for a right facet)
r	axial rotation (right shoulder forward is positive for a right facet)
e	extension

5. C4-C5 Left Facet JCS (relative position of the C4-C5 left facets based on Optotrak markers)
    a. Left Facet JCS contains the same components as the Right Facet JCS state, but are mirrored to maintain the clinically relevant naming conventions.
6. Right Facet Force (force in right facet based on flexiforce sensor value)
7. Spine Load Cell Position 2 JCS (relative position of the load cell relative to C5. Used for calculation of JCS Load and is based on position of robot)
    a. Standard x, y, z, roll, pitch, yaw convention for transformation

The facet force was measured using Flexiforce sensors (100 lb capacity) (Tekscan Inc., Boston, MA, USA). A custom signal conditioner was built by the Cleveland Clinic Electronics Core. The circuit board allowed for the measurement of up to 4 flexiforce sensors and had a trim pot to provide adjustable gain.

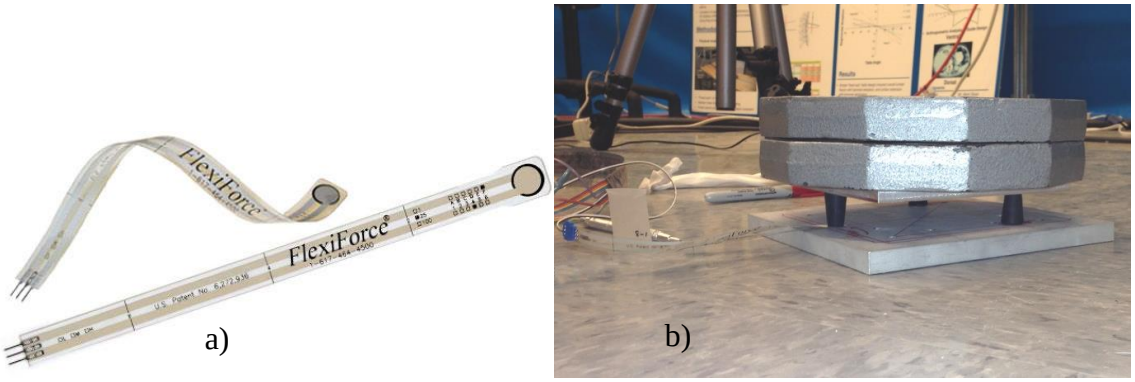


Figure 48: a) Thin Film Force Sensor and b) Calibration

To calibrate the sensors, a custom fixture was made consisting of 3 conical shaped test tube rubber stoppers, two aluminum plates, and deadweights. The diameter of one end of the rubber stoppers was matched to the sensing area. It also provided compliance to distribute the load across the sensing surface. Markings were made on the aluminum base plate at the vertices of an equilateral triangle. These markings were the locations where the rubber stoppers were placed. The sensor being calibrated was placed under one of the rubber stoppers. The top plate was added next and aligned to the base plate using the alignment markings. Finally, the deadweights were stacked on top using the markings for the centroid of the triangle as the center point of the deadweights. The applied force was divided by 3, and the sensor output was measured at 5 points in a load range from 15 to 60 N. A linear fit was applied to the data and the slope was utilized as the scaling factor. The offset was ignored because the top plate imparted some unknown forces on the sensor and the response was very linear. Also, with the sensor unloaded (not in the calibration fixture), the output was always generally very close to zero unless a sensor was experiencing reliability issues. The Right Facet Force state was made using

the Right Facet Force sensor input. The state and controller used the sensor output in volts, and the conversion to Newtons was done in post-processing.

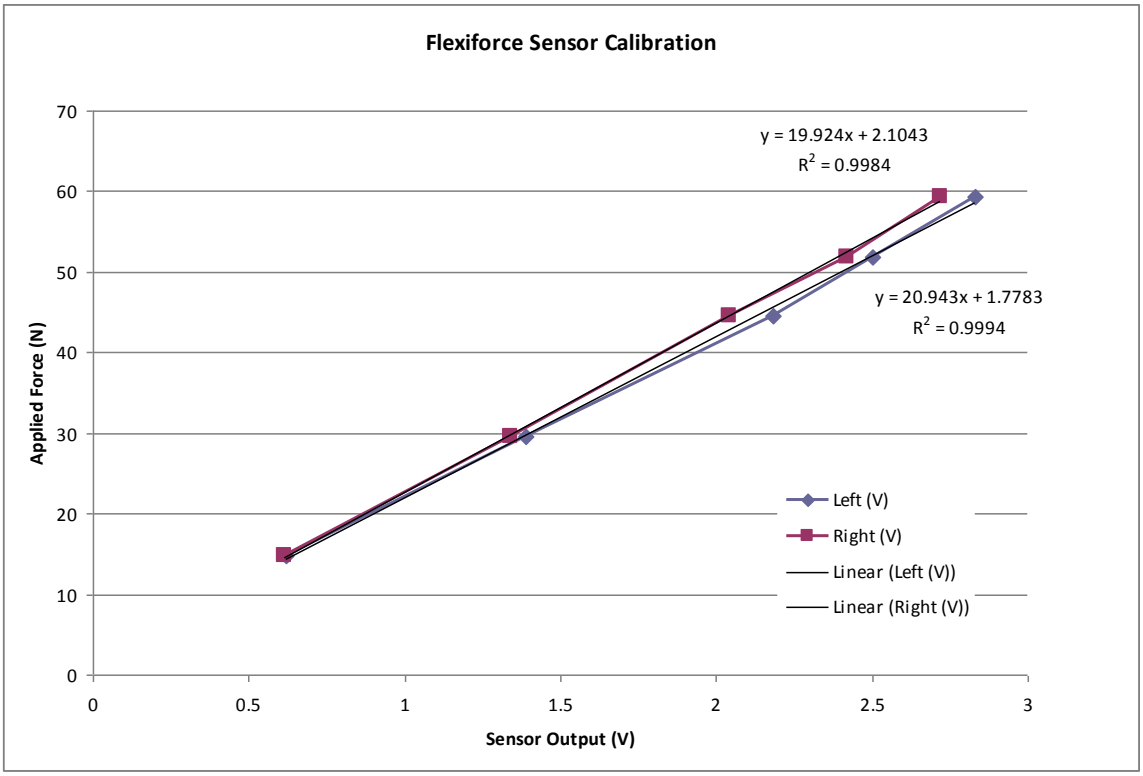


Figure 49: Flexiforce Sensor Calibration Curves: Example

After the specimen was mounted to the robot, the facet force sensors were implanted at both the right and left C4-C5 levels. To minimize the effect that facet capsule resection had on the motion of the spine, the sensors were implanted using a posterior approach. A scalpel was visually placed at the joint line and angled to match the facet surface. It was gently inserted until the capsule was cut wide enough for the sensor. This process was done carefully to avoid resection of the lateral capsule. To prevent the sensors from displacing during the test, a suture was placed through the edges of the sensor lead-in area and attached to the soft tissue just superior to the facet.

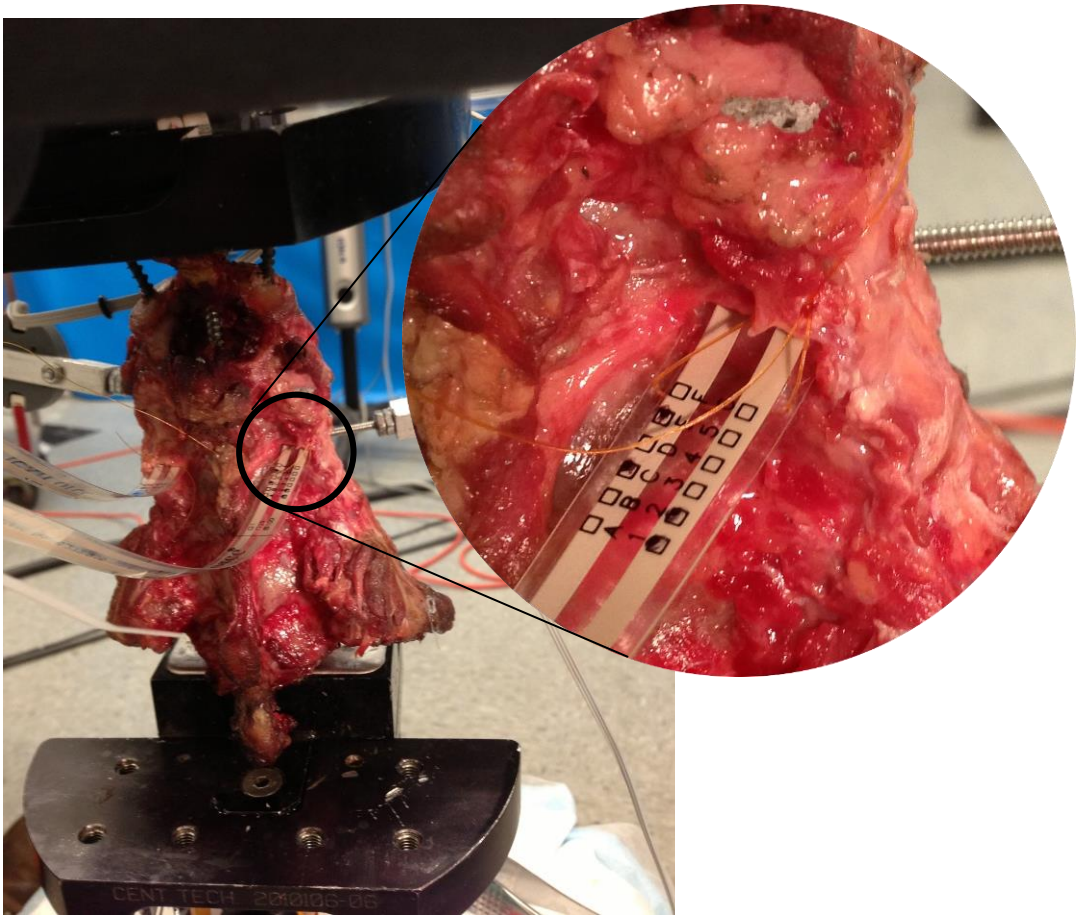


Figure 50: Facet Force Sensor Placement and Attachment

Pilot testing had shown that if the sensors were not sutured into place, they could work their way out of the joint. In addition, the suturing technique of using both edges to form a triangular attachment was an iterative solution that reduced the motion of the sensor within the joint from run to run. Figure 51a shows a specimen with the spine resected at the C4-C5 level and folded forward to view the inferior surface of the C4 facets and where the sensors rest on the C5 facet. Figure 51b shows the sensor next to the C5 facet.

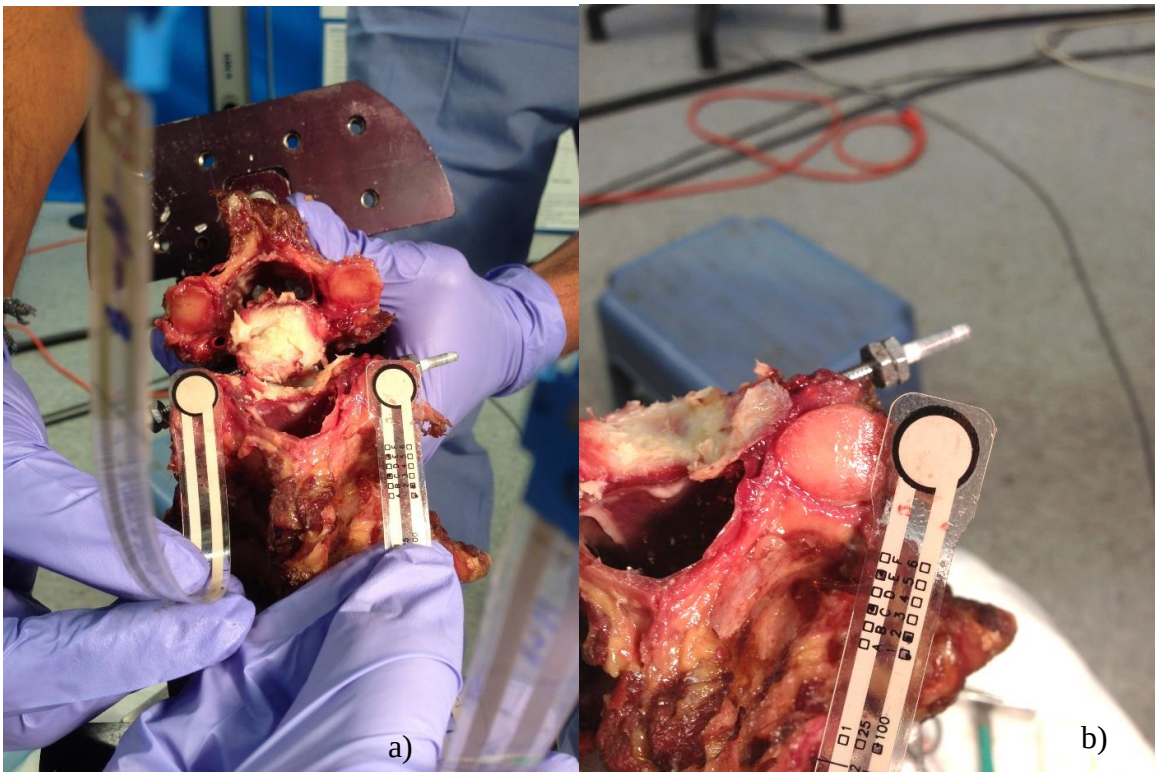


Figure 51: Facet Force Sensor a) placement and b) size

The Kuka KR-16 robot, in addition to serving as the actuator in the system, was also treated as a position sensor. The 6-DOF positions from the robot were utilized in the state calculations. Motion tracking markers were placed into the C4 and C5 vertebral bodies, and the NDI Optotrak Certus measured the motion of those markers in 3D space. The Optotrak was also used to digitize the relative spatial relationships of the robot and all the other rigid bodies in the system. Note in the list of states created that JCS and C4-C5 JCS states were theoretically calculating the same thing. However, the position sensors used in each state were different. As mentioned in section 3.04.2, control of the state using the robot is accomplished via the single position sensor state. Since the specimen was a multi-segment spine, C4 was not rigidly attached to the robot and C5 was not rigidly attached to the base. As a result the JCS state kinematics were more globally

approximate of the entire specimen and are, in actuality, the relative kinematics of C2-T1 based on digitizing C4 and C5. The C4-C5 JCS state was based on the sensors rigidly attached to the vertebra and was a much more accurate representation of the actual joint level kinematics. For this reason its kinematic state served as the input to the LUT for the NMS controller.

The loads in the system were measured using a 6-DOF load cell (Delta IP-65 SI-330-30, ATI Industrial Automation Inc., Apex, NC, USA) (Range Fx, Fy = ±330N; Fz = ±990N; Mx, My, Mz = ±30Nm) which was attached to the end of the robot.

Section 5.04 Loading Conditions

Since this work can be described as a proof-of-concept, the loads applied to the spine were not required to mimic a specific *in vivo* situation. The goal was to provide repeatable loading conditions, of physiological magnitudes, capable of isolating various effects to analyze any interactions in a controlled manner. There were 8 loading conditions. The loads selected were pure moments (± 2 Nm) in the 3 primary rotational axes (flexion-extension, lateral bending, and axial rotation) along with 2 combined torques designed to increase right facet forces (2 Nm right lateral bending with 2 Nm left axial rotation and 2 Nm extension w/ 2 Nm left axial rotation). In addition, a 40 N head weight was added to simulate the compressive load imparted by the mass of the head. The original plan was to execute all loading conditions with the NMS facet force feedback off and then repeat the same test with it turned on at various gain levels. Pilot testing had shown inconsistent facet force results from run to run and it was thought that

the artifact was due to the sensor shifting between each loading and unloading cycle. To minimize this artifact the order of the loading conditions was modified so that a specific loading condition was applied. Then, while still at that same loading condition, the facet force feedback controller was enabled. In this way direct visualization and comparison of the pain-modulated motion could be captured. To understand the non-linear effects that the facet force controller had on the resulting motions and loads each loading condition was subjected to three levels of nociceptive sensitivity (none, high, medium). To modulate this sensitivity the gain value of the facet force feedback controller was adjusted. When nociceptive sensitivity was "no" the gain was 0 and the controller was effectively disabled. The "high" sensitivity gain value was based on pilot testing and was defined as the highest value at which the controller gain could be reasonably set to allow the system to remain stable. A value of 0.07 was selected. The "medium" sensitivity was 50% of the "high" sensitivity gain (0.035). The UMS trajectory editor was used to build the corresponding trajectory files for these loading conditions and the User Parameters were used to define the target facet force and the gain value.

The loading profile was a trapezoid shape with the following parameters.

1. Ramp to loading condition: 10 seconds
2. Hold with sensitivity = "no": 35 seconds
3. Hold with sensitivity = "high": 25 seconds
4. Hold with sensitivity = "medium": 18 seconds
5. Unload/load to next loading condition: 10 seconds

The nature of the testing is quasi-static and therefore the hold time on the trapezoid profile was designed to allow the loads to settle so that the system state would be a very repeatable condition. The largest state changes took place in getting the spine to the loading condition, causing this time to be the longest. The other changes took less time to stabilize, causing the values required to be less.

Section 5.05 Osteophytes: Facet Force Mechanical Short Circuit

During the course of testing it was noticed that some specimens had little or no facet force measurable by the thin film force sensors. The source of this phenomenon was found to be due to the osteophytes that surrounded the facet joint. The effect of this mechanical short-circuit has been identified in other studies as well [68, 98-100]. The superior and inferior C4-C5 osteophytes were contacting to transfer the load between vertebrae without transferring the load across the facet contact surface. Figure 52a shows some of the articulating surfaces from "normal" facets. In contrast, Figure 52b shows the facets of specimen C111406. Note the irregular shaped edges, the osteophytes around the perimeter, and the arthritis. The arthritis is most pronounced in the left facet articulating surfaces. The discolored area is the missing cartilage. The images also show that the sensing area of the Flexiforce sensor is smaller than the articulating surface of the arthritic facet. These morphological properties of the arthritic facet illustrate some of the difficulties with acquiring facet force measurements.

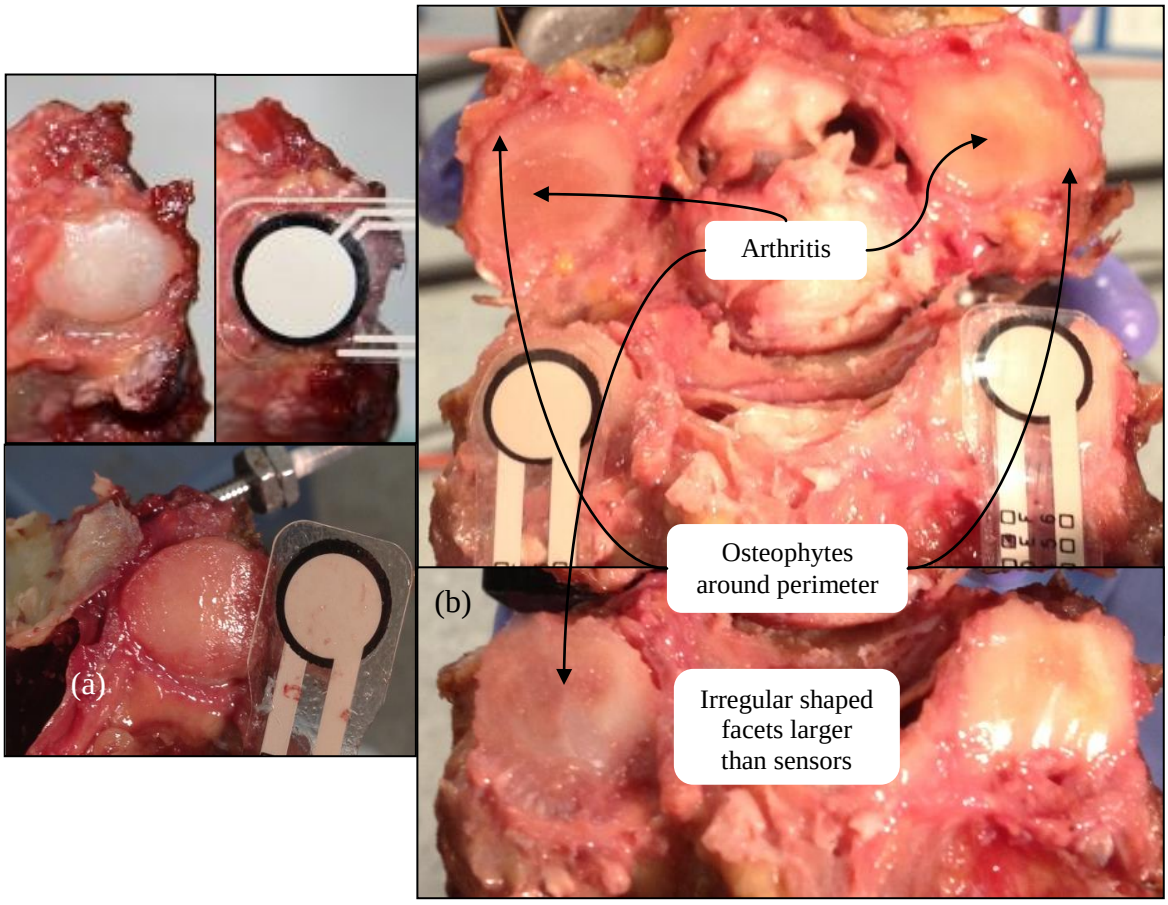


Figure 52: a) “Normal” Facets and b) Facets with Osteophytes

In typical biomechanical pure moment testing this mechanical short circuit would not have been recognized and may or may not have influenced the results. In this study, the absence of facet contact force would prevent the completion of the testing for that specimen. In these cases, a rongeur was used to remove enough osteophytes so that some amount of contact force could be measured. The intent was not to remove all the osteophytes since this might have modifed the specimen in an undesirable way. Figure 53 shows a specimen with the lateral osteophytes fully removed.

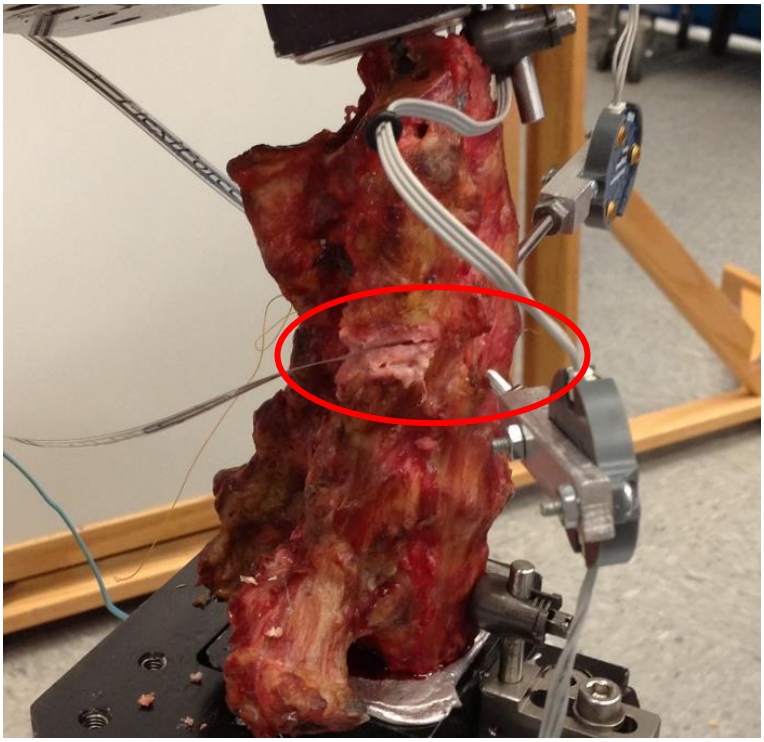


Figure 53: Spine with Fully Removed Lateral Facet Osteophytes

Section 5.06 Analysis Methods

Once the system was initialized, the loading conditions were repeated for a total of 3 times. It should be noted that the specimens typically were loaded additionally to verify the sensors were able to measure the force and osteophytes did not need to be removed. In addition, the neural feedback controller was built upon a specimen specific LUT. Each specimen was pre-tested at select loading conditions to verify that the LUT was providing a reasonable solution when the controller was enabled. Based on pilot experiments, these initialization loads and the first complete loading cycle were considered pre-conditioning and the data was discarded. Data from both the second and third cycles were kept and all kinetics and kinematics were analyzed at the points of interest. To gather the point-of-interest data, the time based data was first zero-phase low

pass filtered at 1 Hz. The next step was to extract an average of the last 5 seconds of data from a given loading condition/sensitivity value. Finally, the values from the second and third runs were averaged. The averages of the kinetics and kinematics between "no" and "high", as well as "no" and "medium" sensitivity to neural contact force were compared using a t-test. Statistical significance was identified for all measurements of $p < 0.05$. In addition, a histogram of the standard deviations of the second and third runs was created for each measurement to verify that the resulting curve matched the form of a typical f-distribution. If they did not match that form it would suggest that averaging the second and third runs would be an invalid statistical method. For example, if one pair of repeated runs had a difference in the measurement that was 10 times that of another repeated pair then the technique of averaging would mask this phenomenon. However, if the differences between the repeated runs were all similar across all conditions then averaging would be valid.

CHAPTER VI

CADAVERIC EXPERIMENTATION RESULTS

Section 6.01 Overview

Table III and IV contain the detailed information on each specimen, sensor scaling, and the notes associated with each set of runs. These notes were made at the conclusion of each pair of runs and represent qualitative observations. Two of the specimens (C111406 and C120252) required removal of osteophytes to achieve some amount of facet contact force to utilize the NMS control algorithm. Also note the qualitative observations regarding the trade-off of facet force from the right to left. Upon dissection of one of these specimens the facet arthritis and osteophytes were evident (Fig 52 and 53). When compared to a normal specimen the thin film force sensors were not large enough to sense the full force being transferred.

Table III: Specimen Parameters and Sensor Scale Factors

Specimen Number	Age (yrs)	Gender	Cause of Death	Height (in)	Weight (lbs)	Left Sensor number	Left Sensor Scale Factor (N/V)	Right Sensor number	Right Sensor Scale Factor (N/V)
1008217N	68	Male	Heart Condition	69	220	SN 11-8	17.1105	SN 6-6	14.908
1010614N	69	Female	COPD	66	100	SN 11-8	22.506	SN 13-6	48.872
1208452	65	Male	Vascular Demenrtia	70	129	SN 1-3	18.772	SN 6-6	9.113
C111406	61	Male	Septic Shock: Pneumonia	62	100	SN 11-8	26.6	SN 11-6	19.795
C112132	58	Male	Cardiac Arrest; Coronary Artery Disease	71	275	SN 11-8	14.862	SN 13-6	25.969
C120252	65	Male	Malignant Arrythmia; Subdural Hemotoma; Malignant Brain Tumor; Non-Hodgkins Lymphoma	67	150	SN 11-8	19.924	SN 13-6	20.943
C120837	59	male	Mycardial Infarction; HTN	70	240	SN 1-3	15.433	SN 6-6	14.29

Table IV: Experiment Run Notes

Specimen Number	Comments
1008217N	None
1010614N	Good run. C4 is missing the spinous process. This is likely why this specimen can go further into extension. It also may explain why it, unlike the other ones, will trade off right and left facet loads in extension. In the other ones, perhaps the stack of spinous processes are taking the majority of the load. In this one, since one is missing, the facets seem to carry the load. Very interesting
1208452	Good one. Good sensor response this time too. Use this video for close ups
C111406	Sensor has a very low output. This specimen has a degenerated C4-C5 right side facet. This is unfortunate since that is the point of this study. This specimen could have been considered one to exclude, but since we have put this much effort into getting it ready we tested it. We removed a large number of osteophytes to get the sensor in. Even after that, the sensor was barely being compressed at maximum torques. We then removed more osteophytes (which meant the lateral capsule was removed as well) and the sensor output increased some so we ran a sample test. We still got a very low sensor output. Then we removed anterior osteophytes to get sufficient load on the facet. In looking at the response, the compensatory strategy is different. In the other spines the solution included posterior translation. In this one the facets are much flatter (i.e. more horizontal). As a result the solution was more lateral bending and no noticeable posterior translation.
C112132	This one was interesting in that facet force was seen on both sides in flexion and extension. The NMS controller traded off the facet forces. As the right force was increased by the nociceptor avoidance algorithm, the left facet force climbed. This was in flexion and extension. On a previous specimen it was only in extension and was through to be due to the missing spinous process. However, this specimen (fully intact) suggests this may be a normal phenomenon for some individuals.
C120252	Initially the sensors indicated very low (negligible) output with the applied torques. Osteophytes were removed on the lateral (right and left) sides of the C4-C5 facets. This increased the force to roughly 0.7V. With the same sensors on the last specimen these same torques produced facet forces at 4.5V. This is a large variance. It is likely there are more osteophytes short circuiting the facet force measurement. At this point, the controller works, albeit less effectively, due to the force shielding of the sensors.
C120837	None

The neural feedback controller creates kinetic and kinematic changes as a result of the simulated "pain" feedback driving modulated, or compensatory, kinematics. Tables C1 – C3 show the mean and standard deviations of the state differences between the "no" sensitivity condition and the "medium" (50%) or "high" (100%) sensitivity conditions. These state differences are the estimated compensatory responses to reduce the force, or simulated pain, in the right facet joint. Bold p values represent statistically significant differences. Italicized values represent states that were trending toward significance but

were not less than 0.05. The changes in kinetics and kinematics are the average of the compensatory responses for all specimens. Figures 55 - 86 are the graphical representations of the results in this table.

Section 6.02 Compensatory Facet Kinetics

Figure 54 contains a representative plot of right facet force vs. time for the combined right lateral bending and left axial rotation loading condition. This also shows how the point of interest data was extracted from the time based data.

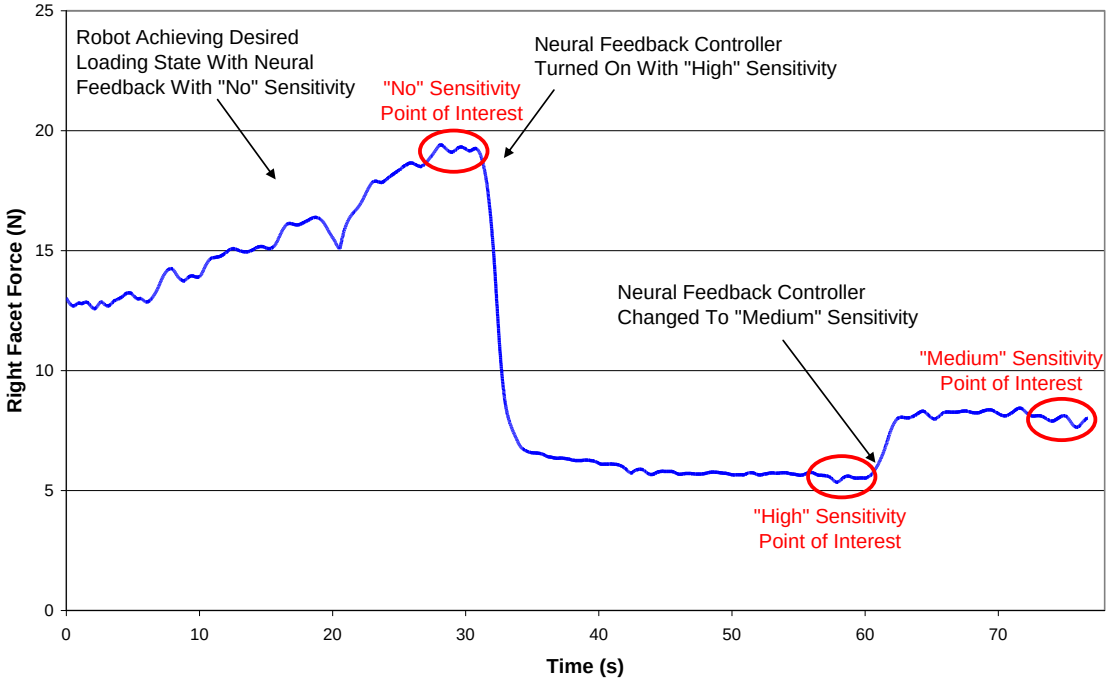


Figure 54: Representative Plot of Right Facet Force vs. Time for the Combined Right Lateral Bending and Left Axial Rotation Loading Condition

Note the drop in facet force with the controller enabled at “high” sensitivity. Next, with the sensitivity reduced to 50% of the “high” value the facet force increased, but not to 50% of the change from “no” to “high” sensitivity. This suggests a non-linear relationship in the compensatory response to simulated facet pain.

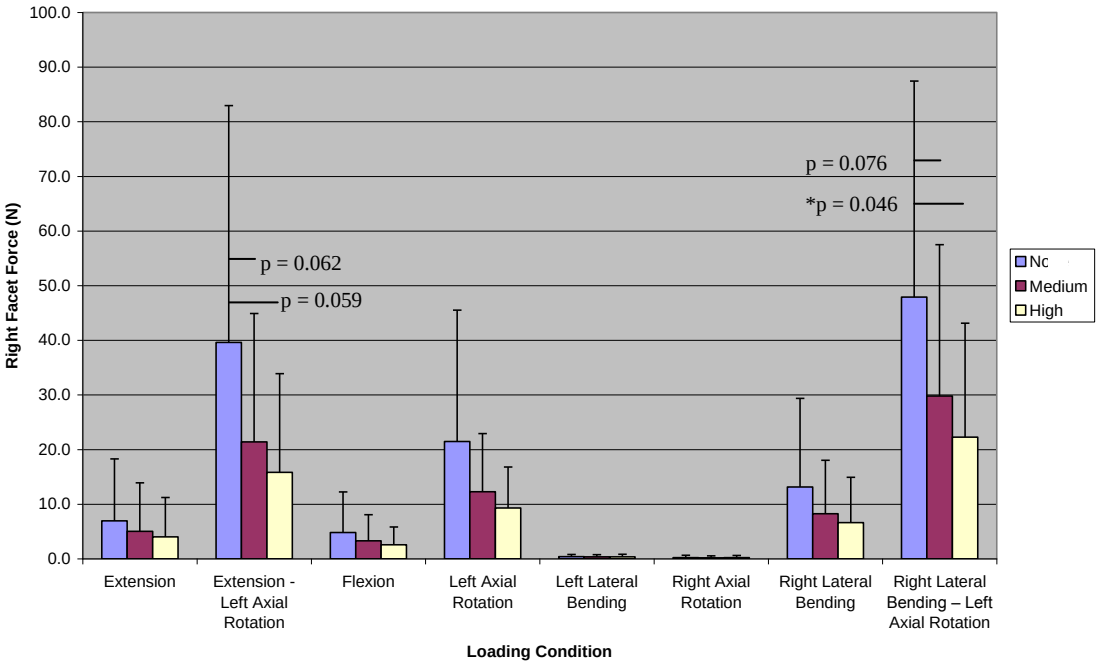


Figure 55: Neural Feedback Controller Affect on Right Facet Force

In all loading conditions the neural feedback controller reduced right facet force (Table C1) and Figure 55). The reductions were found to be nearly significant in the combined Extension – Left Axial Rotation medium and high sensitivity conditions (18.2 N; p = 0.062 and 23.8 N; p = 0.059) and they were significant in the combined Right Lateral Bending – Left Axial Rotation high sensitivity condition (25.6 N; p = 0.046). The differences between "medium" and "high" sensitivity were found to be non-linear in the facet force reduction. The average force reduction as a percentage of the "no" condition load was found to be 37% (± 7%) for the "medium" sensitivity and 51% (± 7%) for the "high" sensitivity for all conditions (excluding Left Lateral Bending and Right Axial Rotation because they showed no appreciable right facet force).

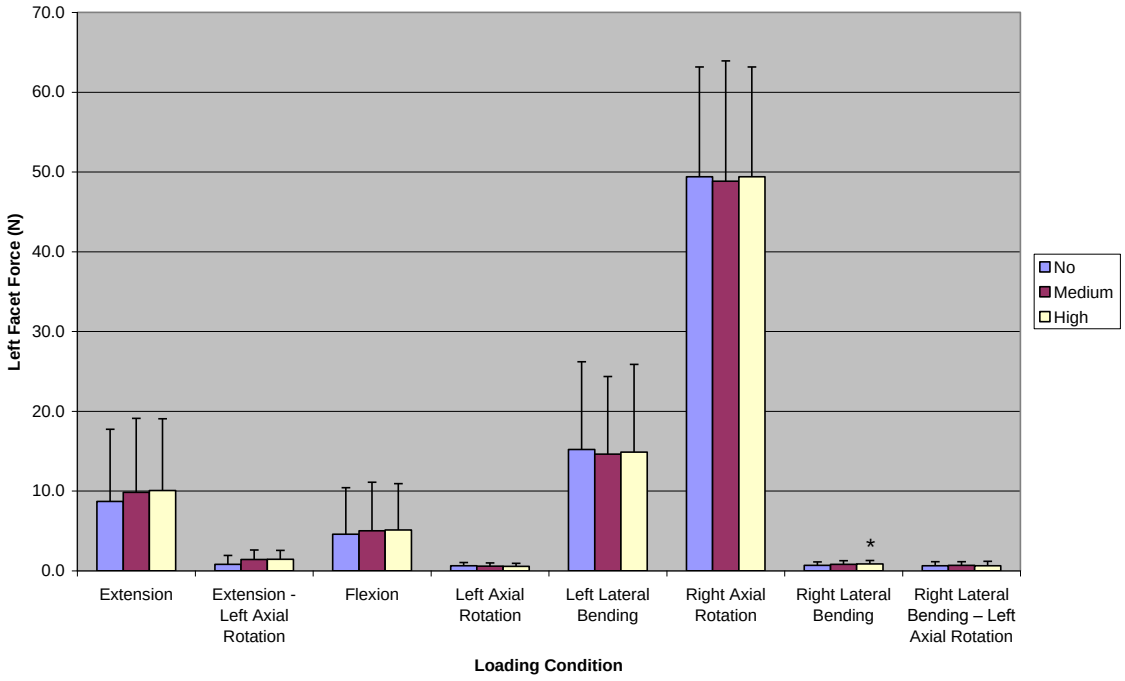


Figure 56: Neural Feedback Controller Effect on Left Facet Force

Increases in left facet force as a result of the neural feedback controller avoiding loads on the right facet were only significant in the Right Lateral Bending loading condition ($p = 0.047$), though the increase was only 0.18 N. Some specimens showed the effect (Table C1) more than others and in some loading conditions where both facets could be equally load sharing (e.g. flexion and extension) the averages showed a slight increase in left facet force (Figure 56). However, on average across specimens this was not found to be a significant side effect of the compensatory motion for the loading conditions tested.

Section 6.03 Compensatory JCS Kinetics

As the right facet was unloaded the loads were transferred to other structures in the spine. The compensatory JCS kinetics represent the changes in the loads relative to the prescribed loads on the spine. The compensatory kinetics may include increased shear loads or torques. The controller may also reduce the applied forces or torques that make up that particular loading condition. When reviewing the plots it must be considered that for some loading conditions a positive change in torque is not necessarily an increase in load *when it is the active loading condition*. For example, Right Axial Rotation torque is assigned a -2 Nm value to achieve a Left Axial Rotation torque of 2 Nm. Therefore, if the neural feedback controller "increases" the Right Axial Rotation torque by 0.5 Nm it is actually reducing the torque from -2 Nm to -1.5 Nm. However, if Axial Rotation torque is not the active loading condition (i.e. desired = 0 Nm) an increase of 0.5 Nm is truly an increase in load of 0.5 Nm.

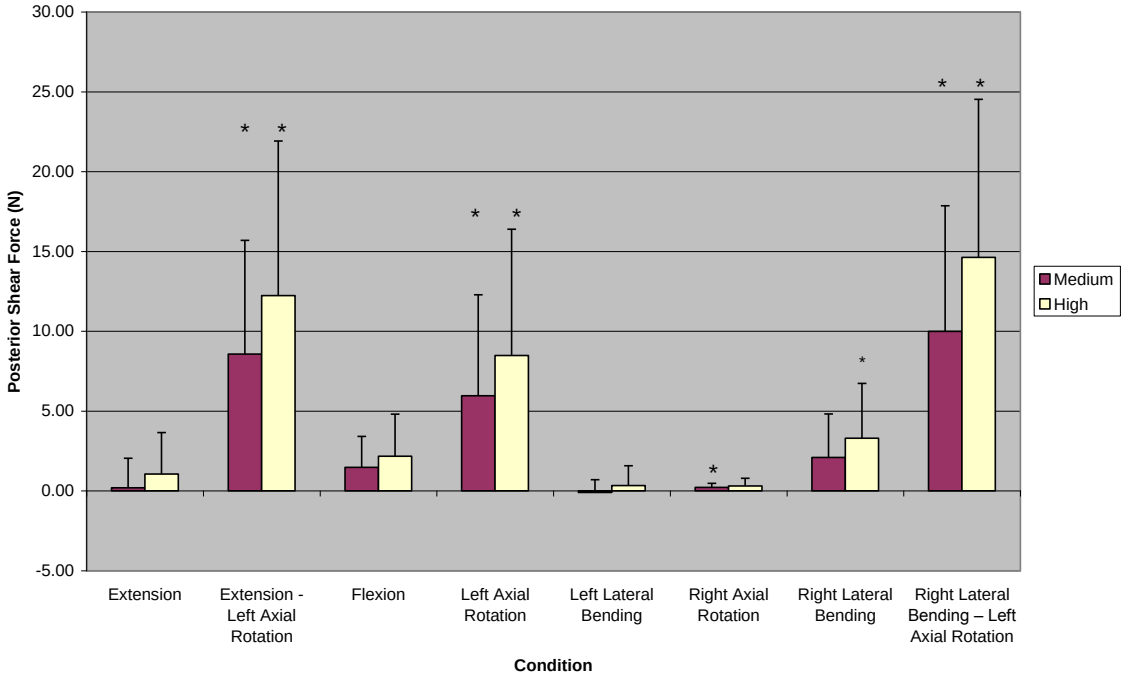


Figure 57: Neural Feedback Controller Effect on Posterior Shear Force

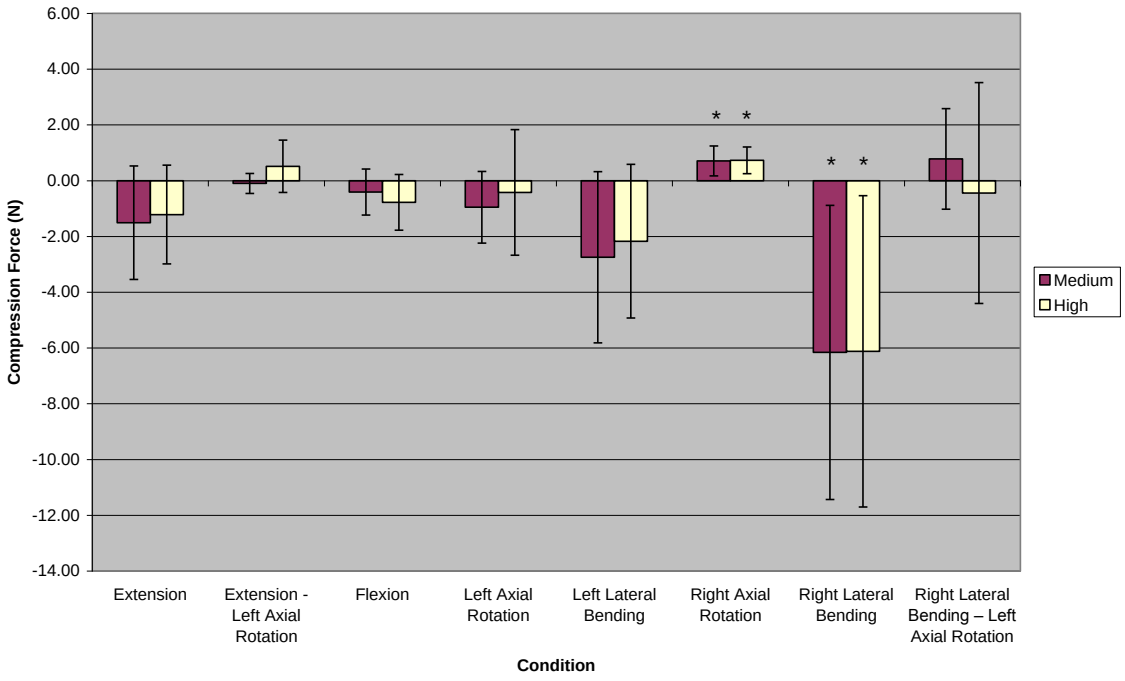


Figure 58: Neural Feedback Controller Effect on Compression Force

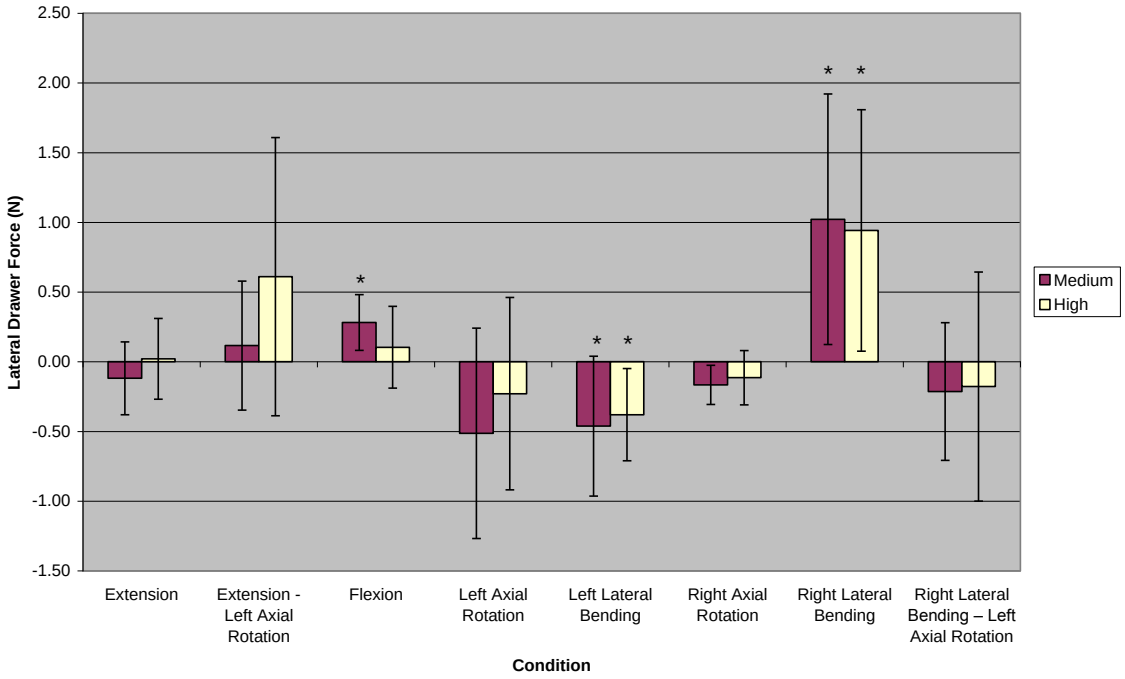
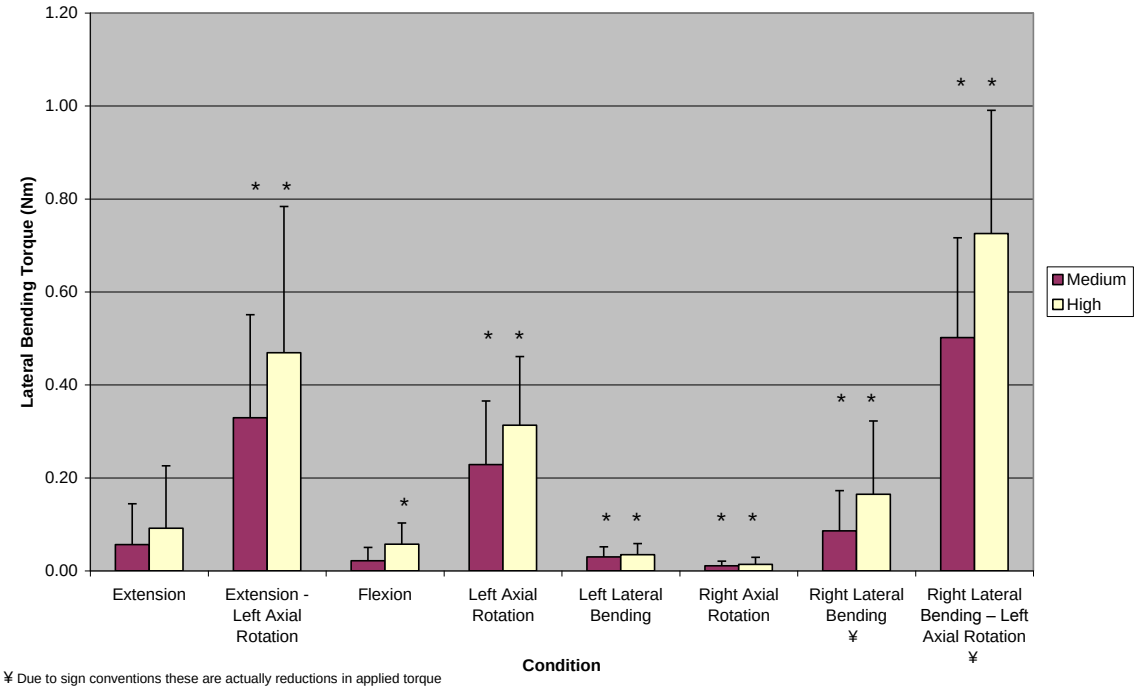


Figure 59: Neural Feedback Controller Effect on Lateral Shear Force



¥ Due to sign conventions these are actually reductions in applied torque

Figure 60: Neural Feedback Controller Effect on Lateral Bending Torque

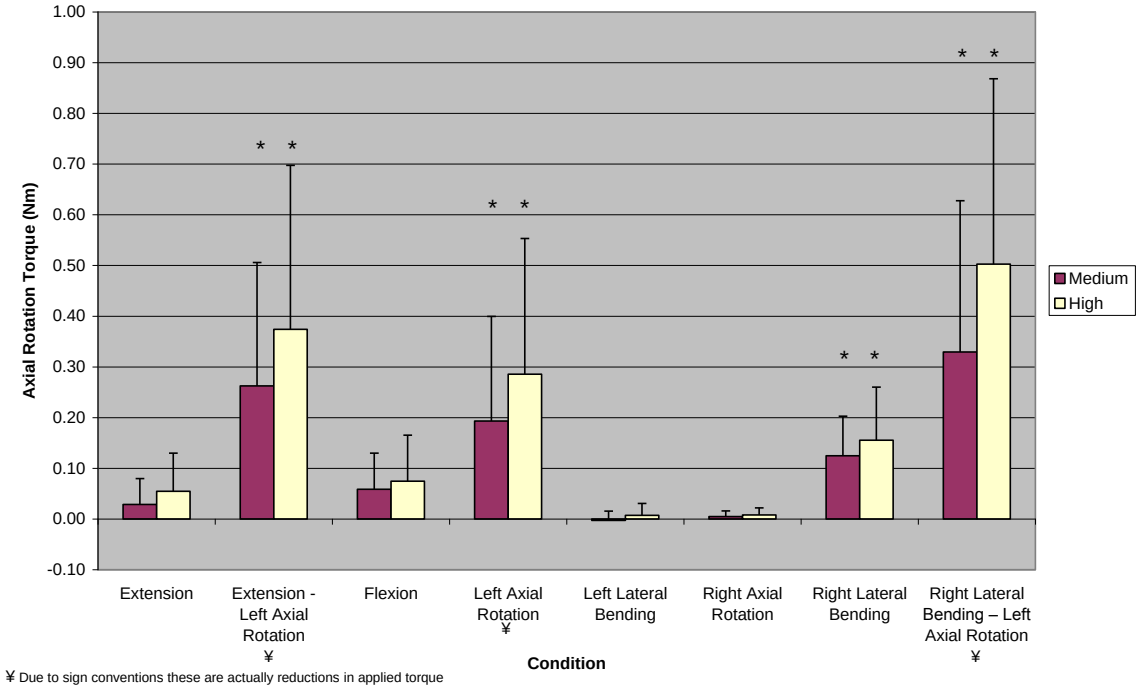


Figure 61: Neural Feedback Controller Effect on Axial Rotation Torque

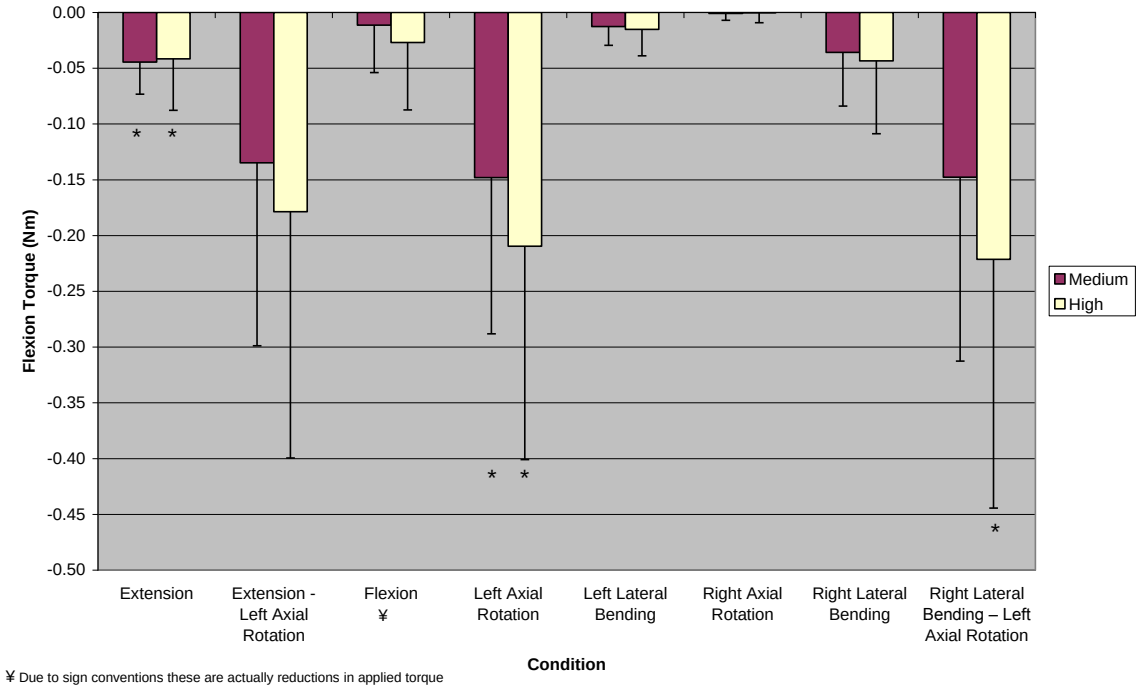


Figure 62: Neural Feedback Controller Effect on Flexion Torque

The neural feedback controller generally produced significant increases in Posterior shear loads (Figure 57). These changes even occurred in cases where

significant changes were not identified in right facet force (e.g. Left Axial Rotation and Right Lateral Bending conditions). The controller also tended to produce significant changes in lateral bending, axial rotation, and flexion torques (Figures 60-62). If the torques were the active loading condition then they were reduced. If they were not the active loading condition (i.e. desired torque = 0 Nm) then they increased. In the combined loading conditions (i.e. those having significant or trending reductions in right facet force at "high" sensitivity) there was commonality in compensatory kinetic changes. Posterior shear loads increased as much as 12.2 N ($p = 0.016$) and 14.6 N ($p = 0.008$) in the Extension – Left Axial Rotation and Right Lateral Bending – Left Axial Rotation conditions respectively. The Axial Rotation torques dropped by 0.26 Nm ($p = 0.022$) and 0.50 Nm ($p = 0.011$) respectively. The Lateral Bending torque increased by 0.47 Nm ($p = 0.008$) and decreased by 0.73 Nm ($p < 0.001$) respectively. The Extension torque increased by 0.22 Nm ($p = 0.039$) in the Right Lateral Bending – Left Axial Rotation condition.

Table V: Compensatory Kinetics for Simulated Right Facet Pain

Loading Condition	Posterior Shear Force	Compression Force	Lateral Shear Force	Lateral Bending Torque	Axial Rotation Torque	Flexion Torque
Extension						x
Extension - Left Axial Rotation	x			x	x	
Flexion			x	x		
Left Axial Rotation	x			x	x	x
Left Lateral Bending		x	x	x		
Right Axial Rotation	x	x	x	x		
Right Lateral Bending	x	x	x	x	x	
Right Lateral Bending – Left Axial Rotation	x			x	x	x

Table V shows the qualitative accumulation of the significant compensatory kinetics for each loading condition. The table allows for a quick reference of which compensatory loads were common among all specimens in reducing force (simulated pain) in the right facet.

Section 6.04 Compensatory JCS Kinematics

The neural feedback controller kinetic changes were a result of the pain-modulated (i.e. compensatory) kinematics. The kinematics were analyzed in three ways. The first was a view of the regional kinematics. This was the output of the JCS state which was the digitization of the C4-C5 vertebrae but calculated based on robot position. It is most correctly thought of as the relative position of the C2-T1 vertebrae, or kinematics of the whole cervical spinal region. The second was a review of the actual kinematics of the C4-C5 FSU based on the Optotrak markers. The last kinematic measurements were the localized right and left facet motions.

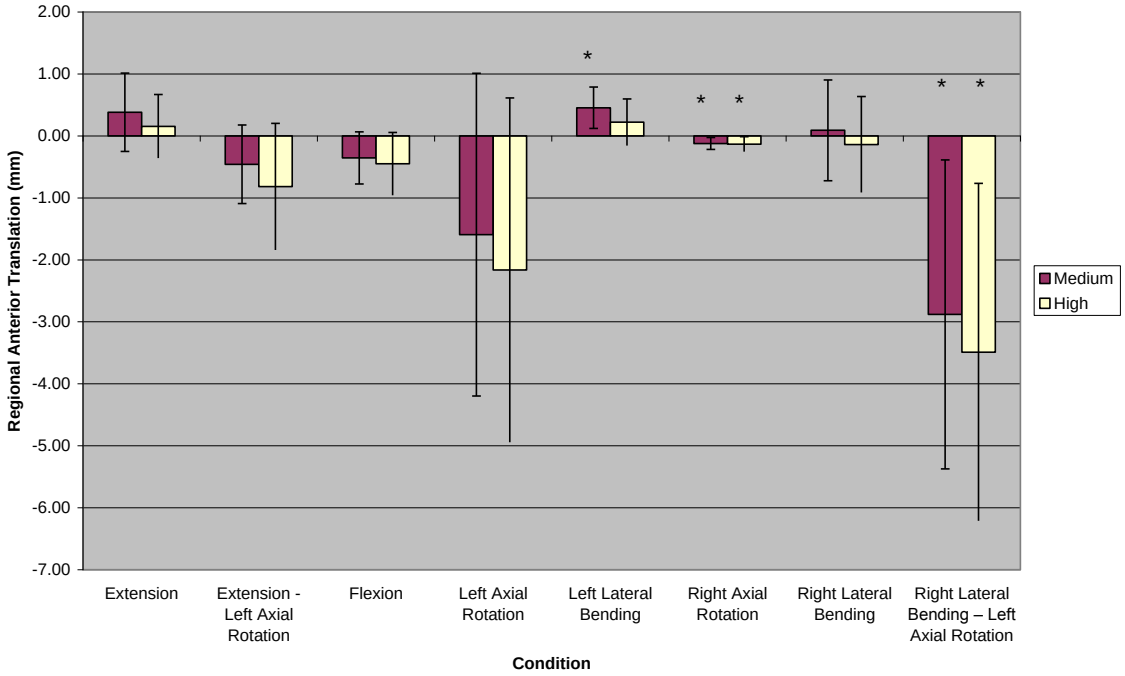


Figure 63: Neural Feedback Controller Effect on Regional Anterior Translation

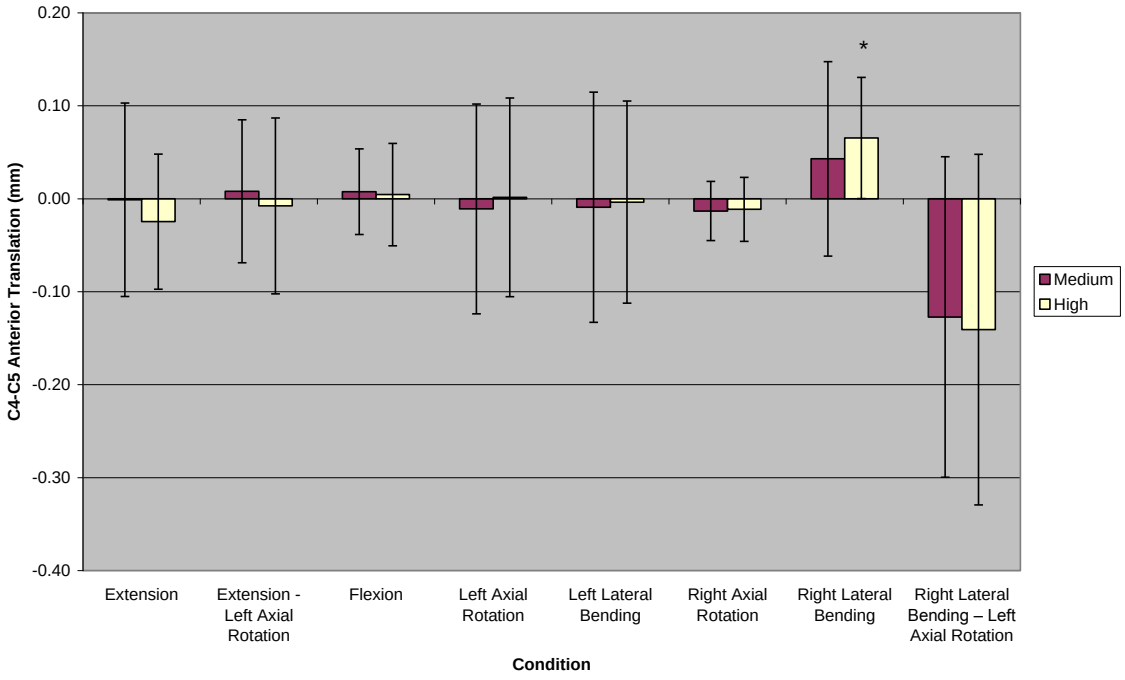


Figure 64: Neural Feedback Controller Effect on C4-C5 Anterior Translation

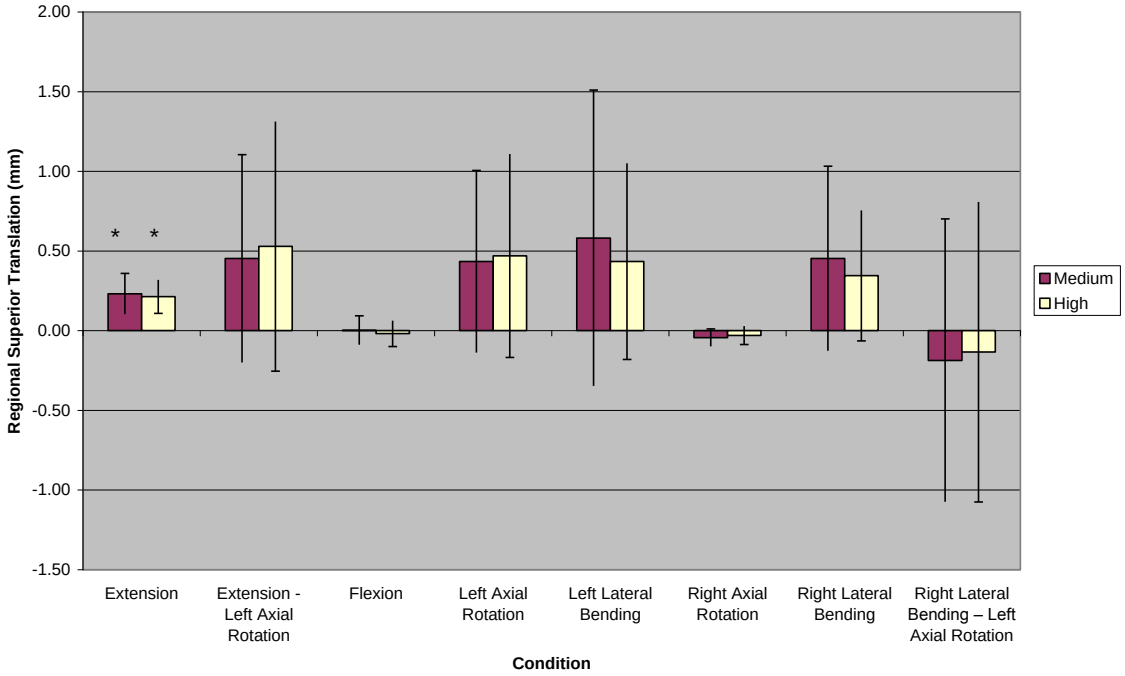


Figure 65: Neural Feedback Controller Effect on Regional Superior Translation

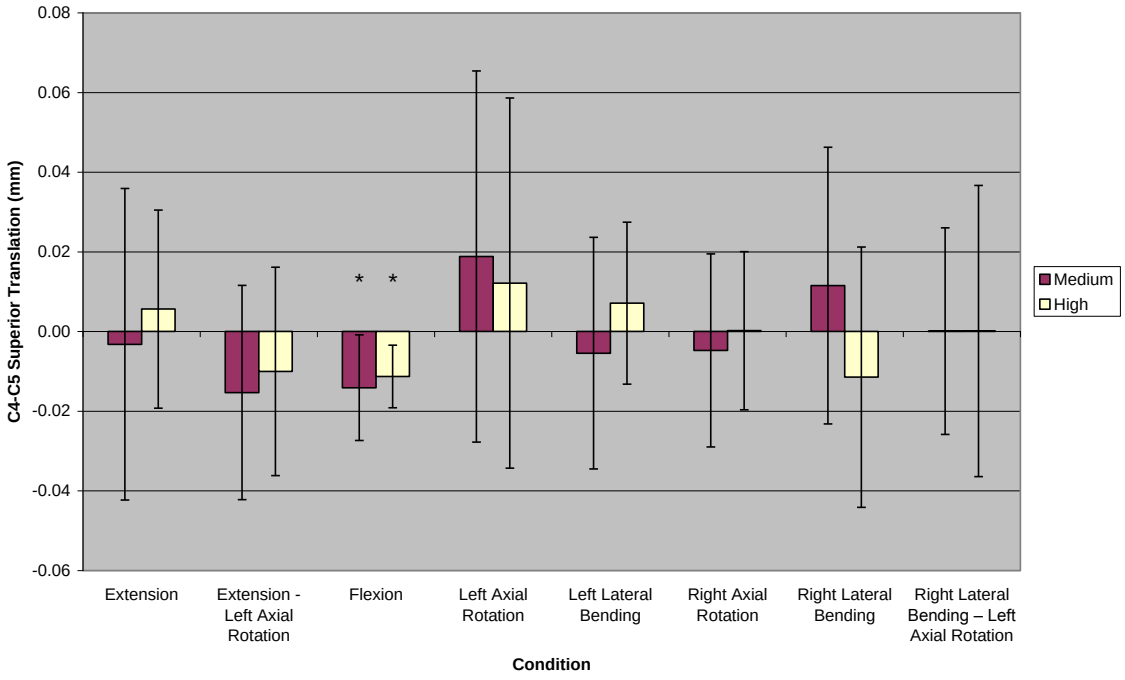


Figure 66: Neural Feedback Controller Effect on C4-C5 Superior Translation

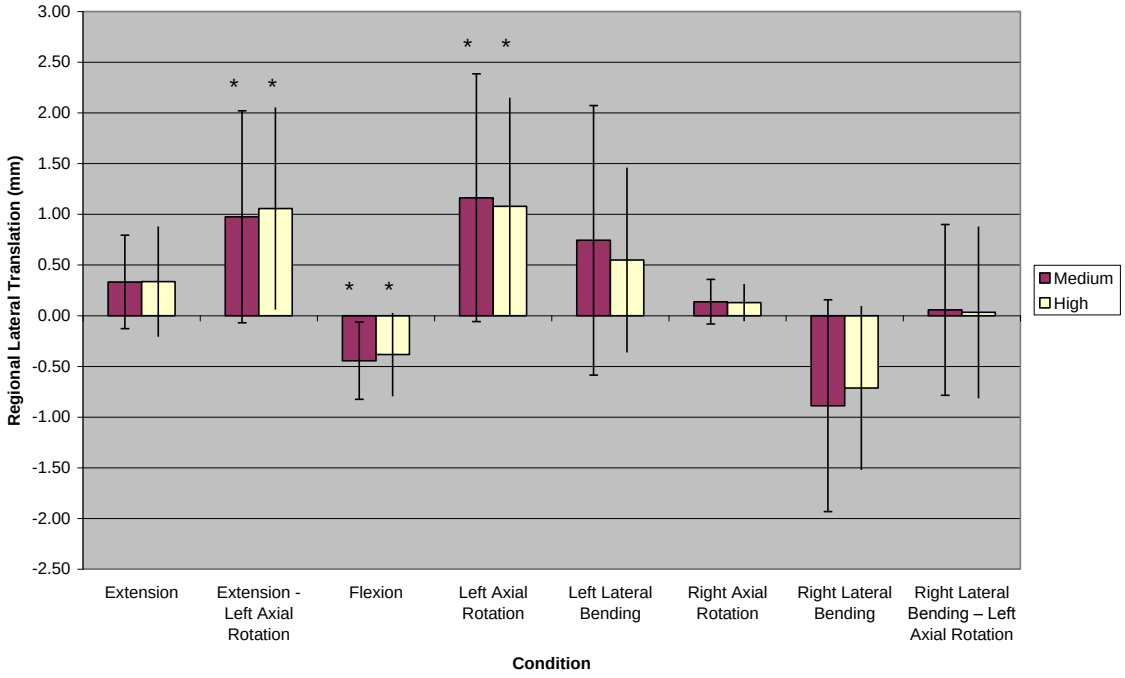


Figure 67: Neural Feedback Controller Effect on Regional Lateral Translation

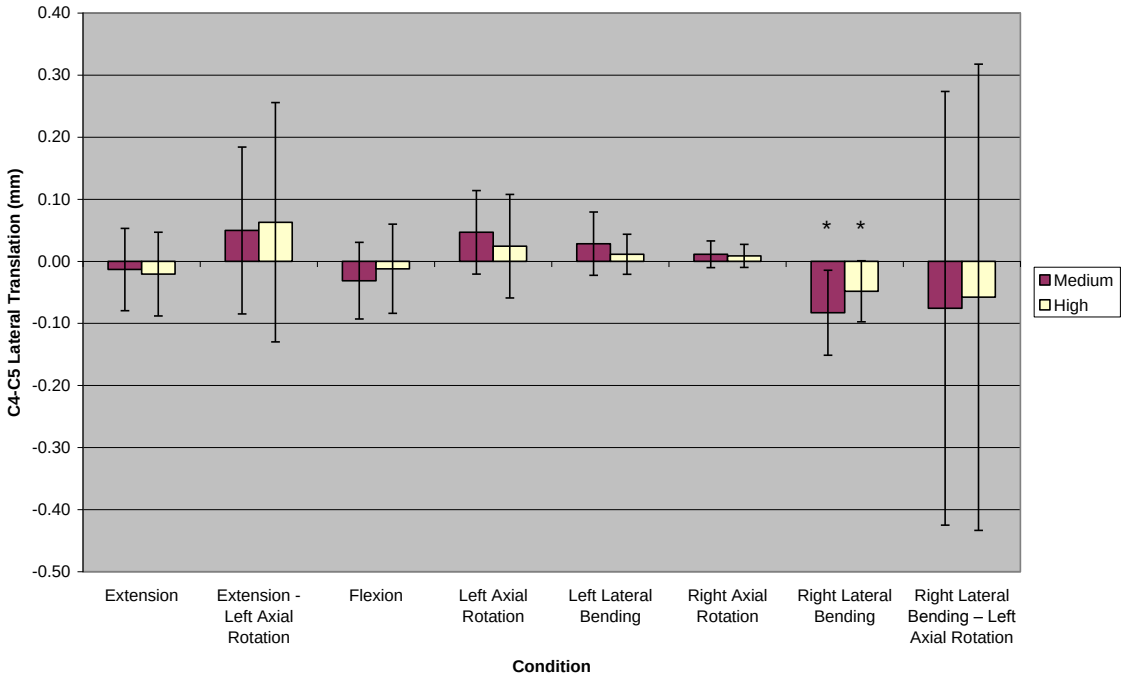


Figure 68: Neural Feedback Controller Effect on C4-C5 Lateral Translation

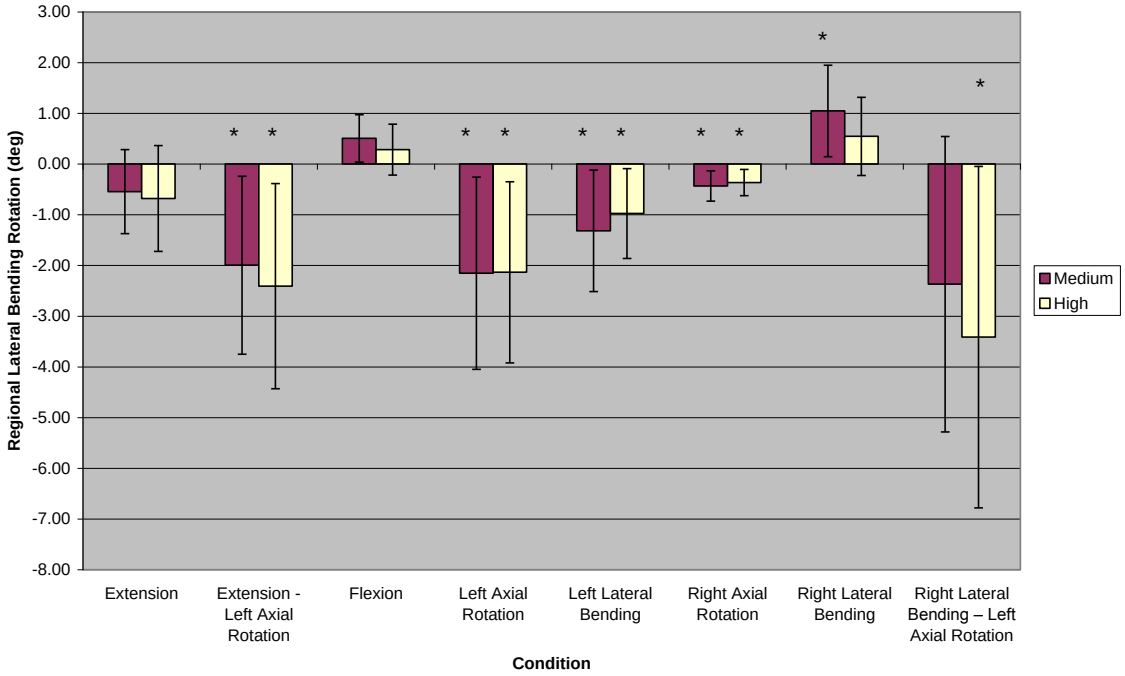


Figure 69: Neural Feedback Controller Effect on Regional Lateral Bending Rotation

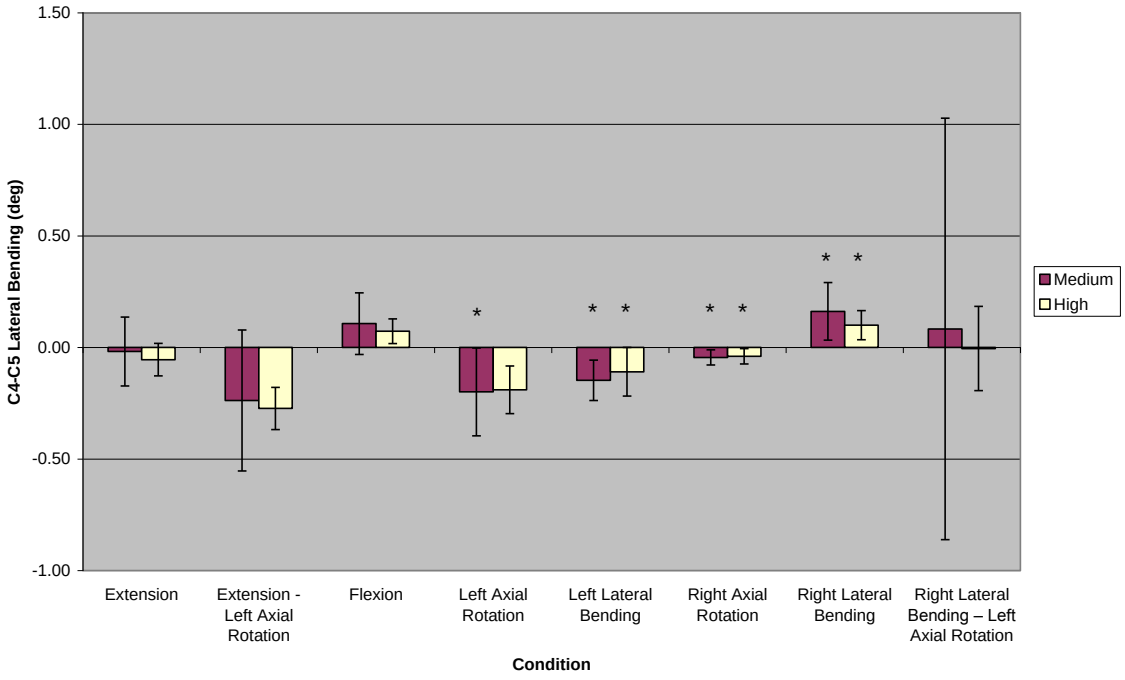


Figure 70: Neural Feedback Controller Effect on C4-C5 Lateral Bending Rotation

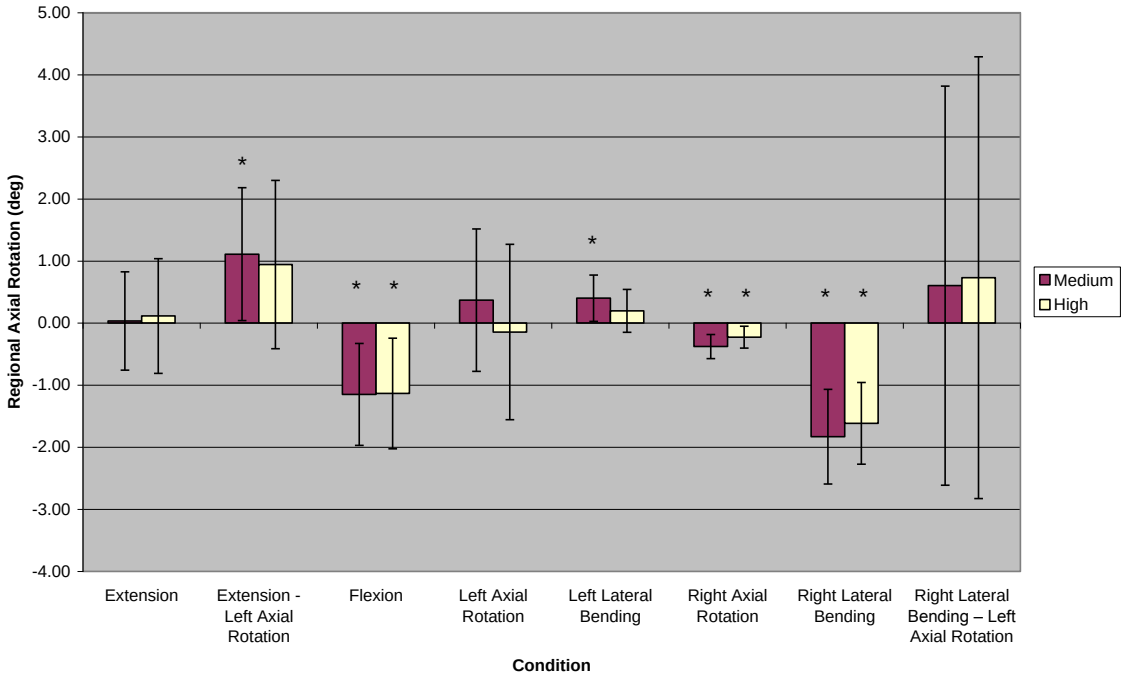


Figure 71: Neural Feedback Controller Effect on Regional Axial Rotation

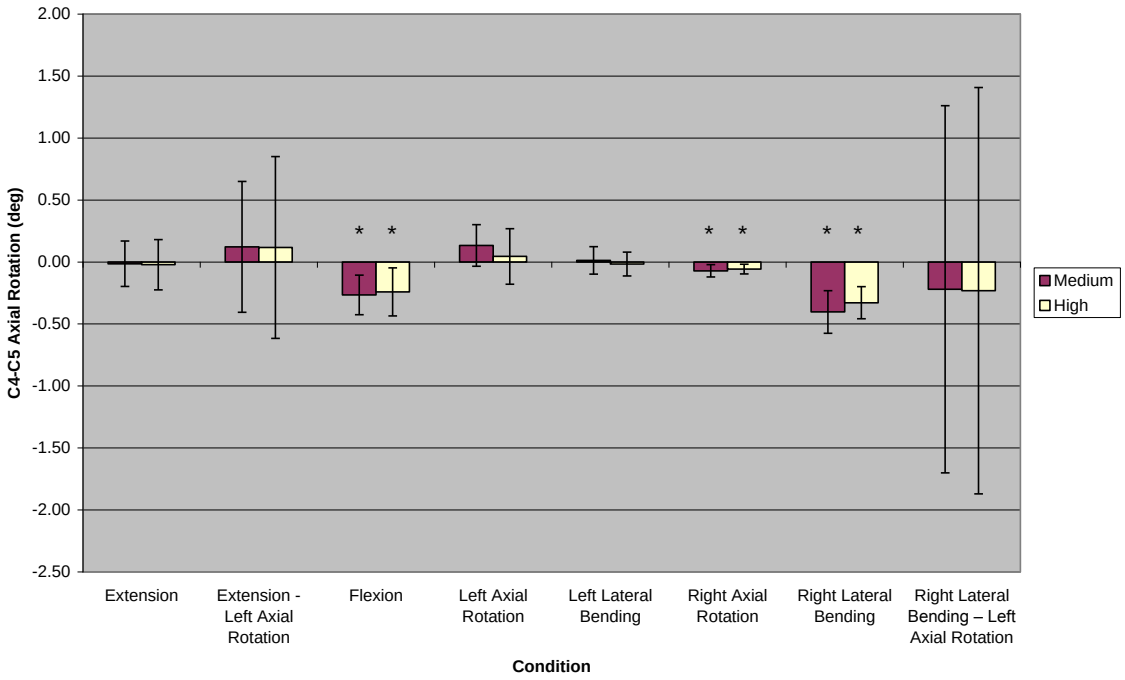


Figure 72: Neural Feedback Controller Effect on C4-C5 Axial Rotation

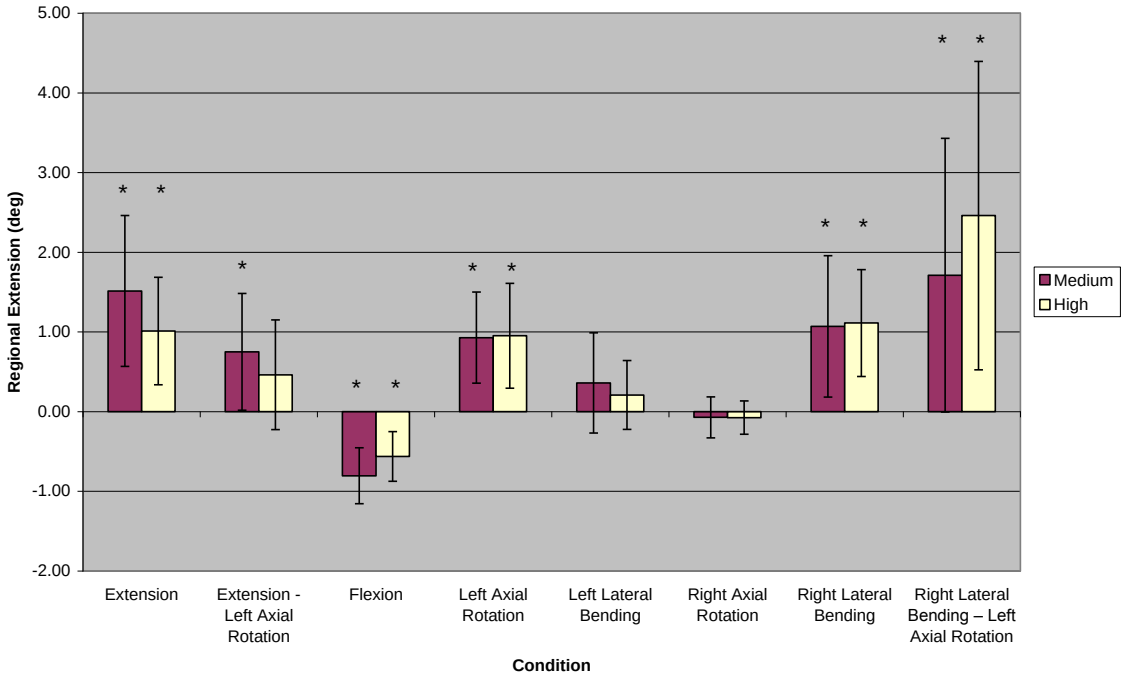


Figure 73: Neural Feedback Controller Effect on Regional Extension

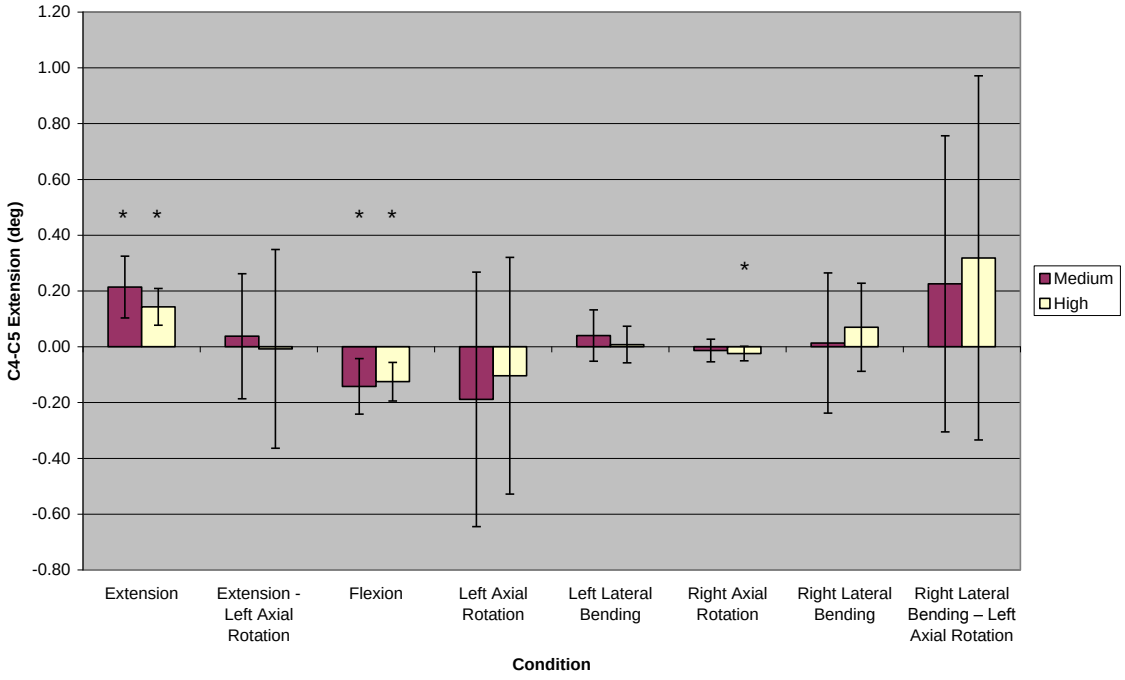


Figure 74: Neural Feedback Controller Effect on C4-C5 Extension

In the regional and C4-C5 compensatory kinematics each state showed statistically significant changes in at least one loading condition. The motions were larger for the regional kinematics (Fig 63, 65, 67, 69, 71, 73) when compared to the C4-C5 (Fig 64, 66, 68, 70, 72, 74) since it was a combination of the motions for all the levels from C2 to T1 rather than just the C4-C5 FSU. There were also more loading conditions with significant changes in the regional kinematics when compared to the C4-C5 FSU. In general, the directions of the compensatory kinematics were similar between the C4-C5 and regional analysis. There is also correlation between the kinetic and kinematic changes. For example the increase in posterior shear force (Fig 57) corresponds to the 3.5 mm ($p = 0.015$) increase in regional posterior translation (Fig 64). Table VI and VII are the qualitative accumulation of the significant regional and C4-C5 compensatory kinematics for each loading condition. The table allows for a quick reference of which compensatory motions were common among all specimens to help reduce force (simulated pain) in the right facet. The superior translation state was not included in the tables as this degree of freedom was not permitted to be modified by the neural feedback controller due to the non-physiological nature of it. In addition, the two loading conditions without any right facet force were not included in the table as these loading conditions were unlikely to have compensatory responses with any clinical significance.

Table VI: Regional Compensatory Kinematics for Simulated Right Facet Pain

Loading Condition	Anterior	Lateral	Lateral Bending	Axial Rotation	Extension
Extension					x
Extension - Left Axial Rotation		x	x	x	x
Flexion	x	x		x	x
Left Axial Rotation		x	x		x
Right Lateral Bending			x	x	x
Right Lateral Bending – Left Axial Rotation	x		x		x

Table VII: C4-C5 Compensatory Kinematics for Simulated Right Facet Pain

Loading Condition	Anterior	Lateral	Lateral Bending	Axial Rotation	Extension
Extension					x
Extension - Left Axial Rotation					
Flexion				x	x
Left Axial Rotation			x		
Right Lateral Bending	x	x	x	x	
Right Lateral Bending – Left Axial Rotation					

These tables are able to identify which kinematic parameters have significant compensatory responses. However, they should not be interpreted in isolation without understanding the sign and relative magnitude of the response. For example, table VI shows that extension was a common compensatory response across all loading conditions. However, Figure 73 and 74 show that the response for the Extension loading condition is to move in more in extension. By contrast, the response for the Flexion loading condition is to move in more flexion. Figures 75 and 76 are 3D plots that combine the "high" sensitivity condition data from Figures 63-74 and Table VI-VII.

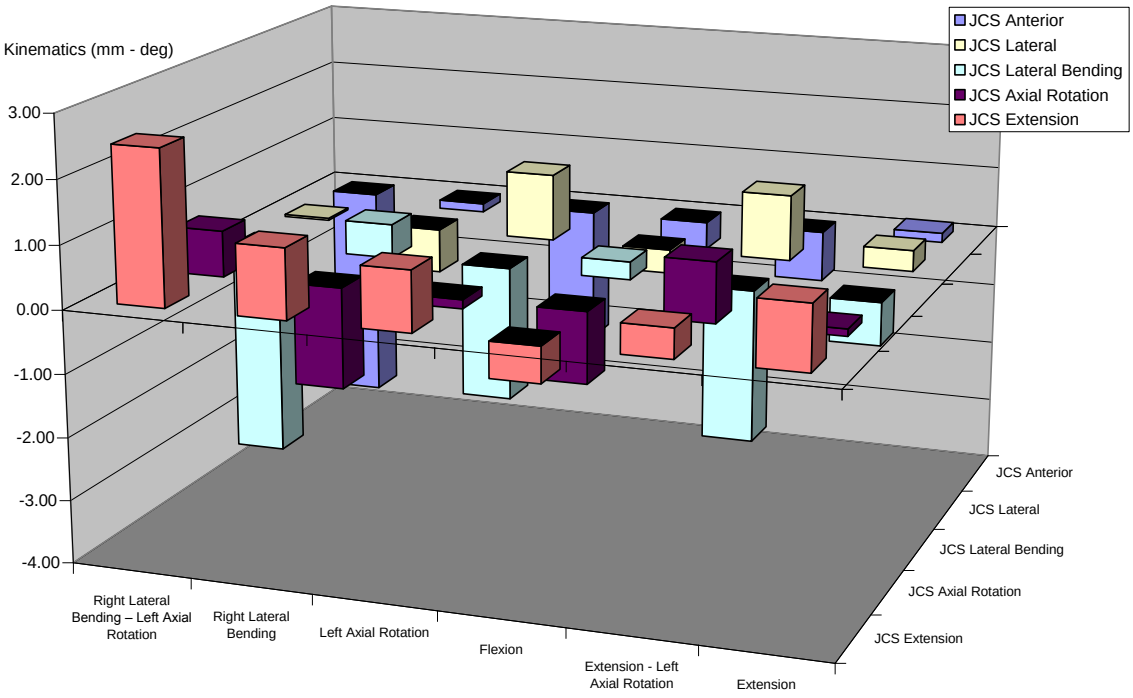


Figure 75: Neural Feedback Controller Effect on Regional Kinematics

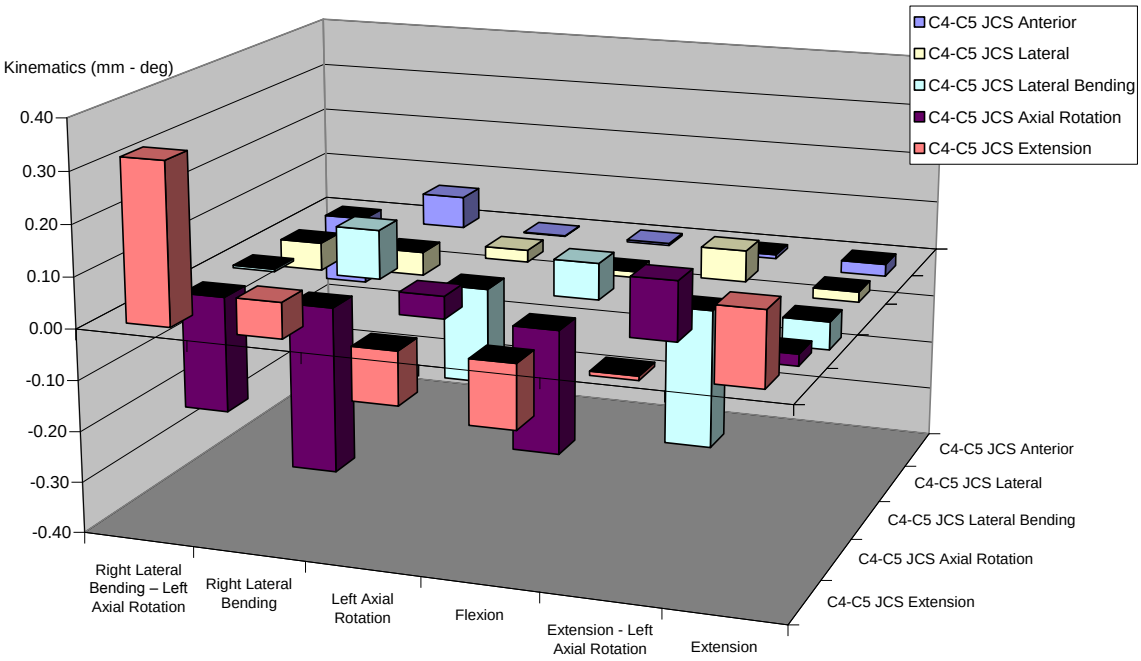


Figure 76: Neural Feedback Controller Effect on C4-C5 Kinematics

Figures 75 and 76 show how the sign and magnitudes of the compensatory kinematic response vary with the loading condition. It also shows the similarity in responses between the two kinematic data sets.

Section 6.05 Compensatory Facet Kinetics

The kinematics for both the left and right facets was recorded during the experiment. The difference between the “no” and “high”, and “no” and “medium” sensitivity conditions was calculated and the compensatory small joint level motions were plotted in Figures 77-88.

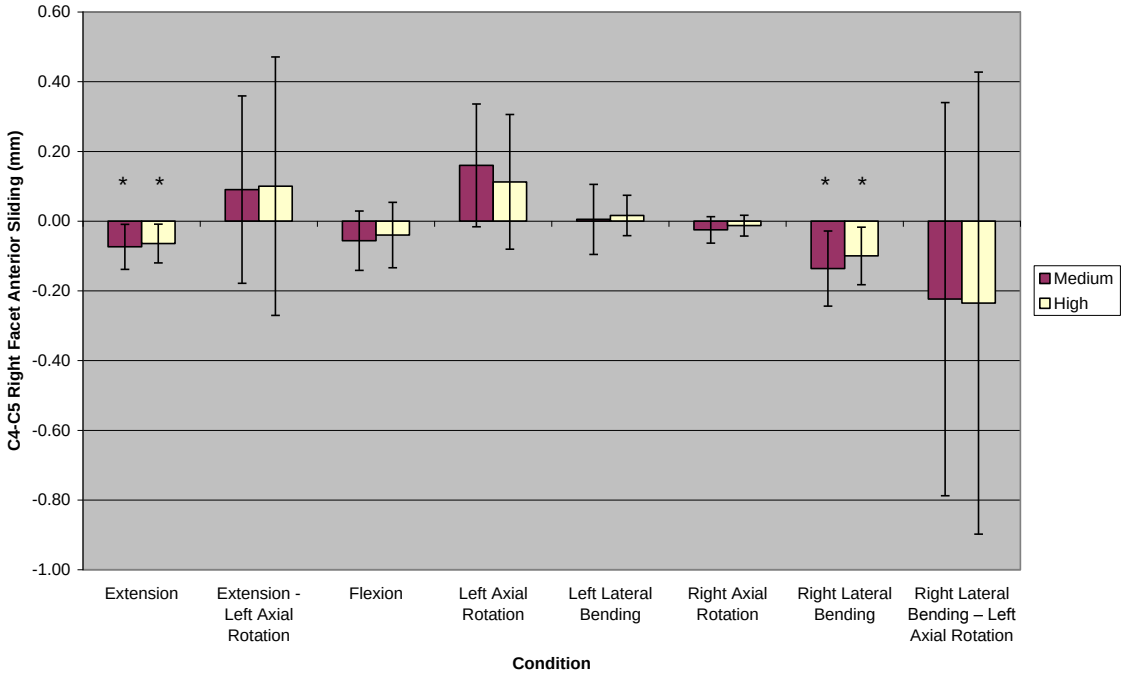


Figure 77: Neural Feedback Controller Effect on C4-C5 Right Facet Anterior Sliding

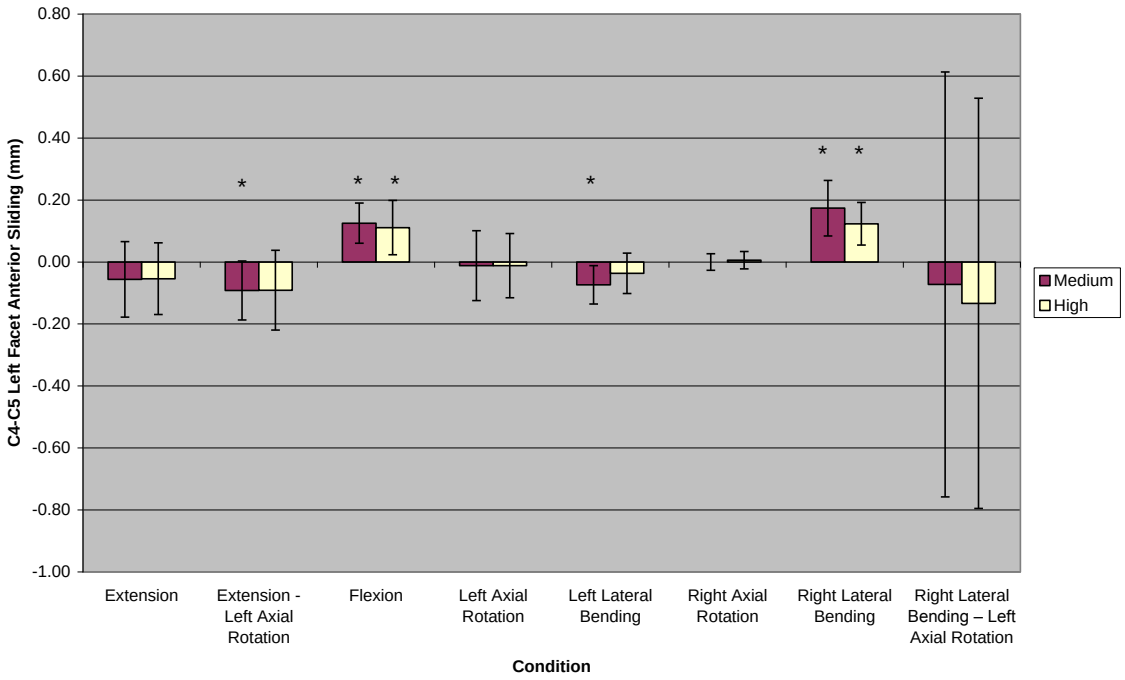


Figure 78: Neural Feedback Controller Effect on C4-C5 Left Facet Anterior Sliding

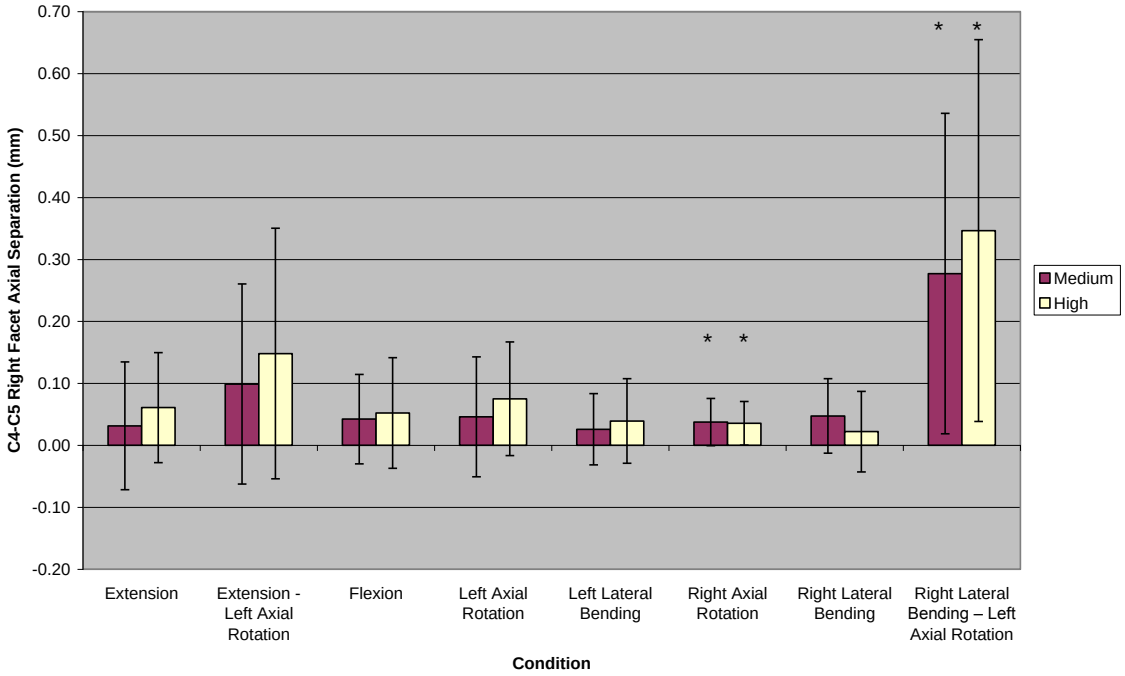


Figure 79: Neural Feedback Controller Effect on C4-C5 Right Facet Axial Separation

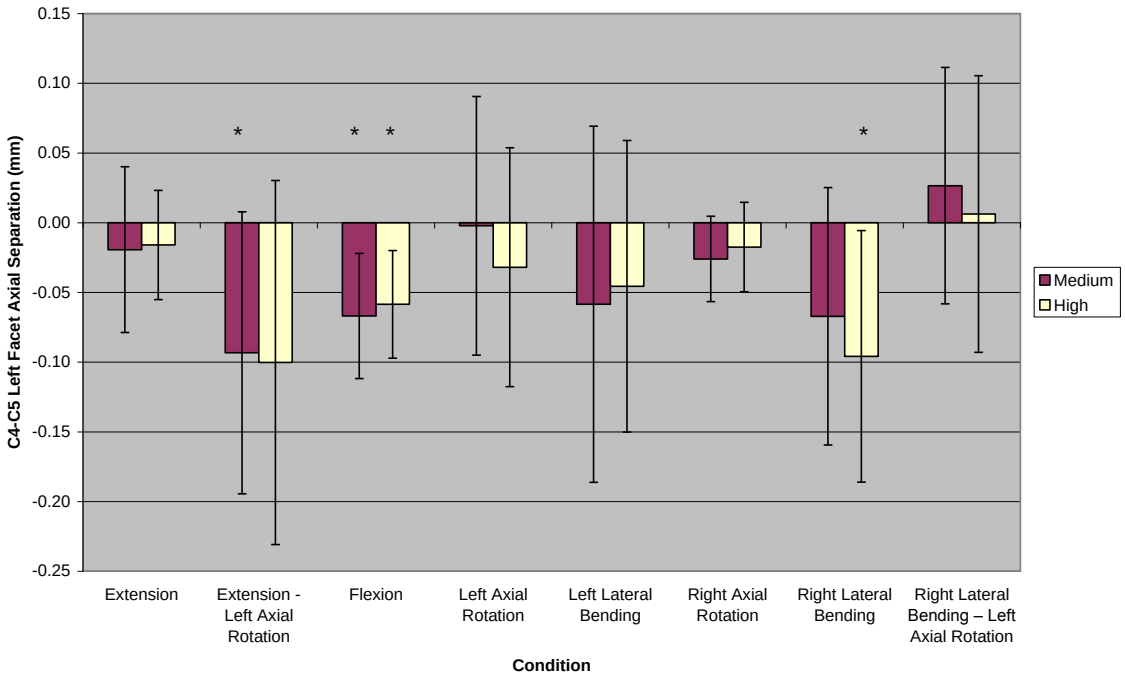


Figure 80: Neural Feedback Controller Effect on C4-C5 Left Facet Axial Separation

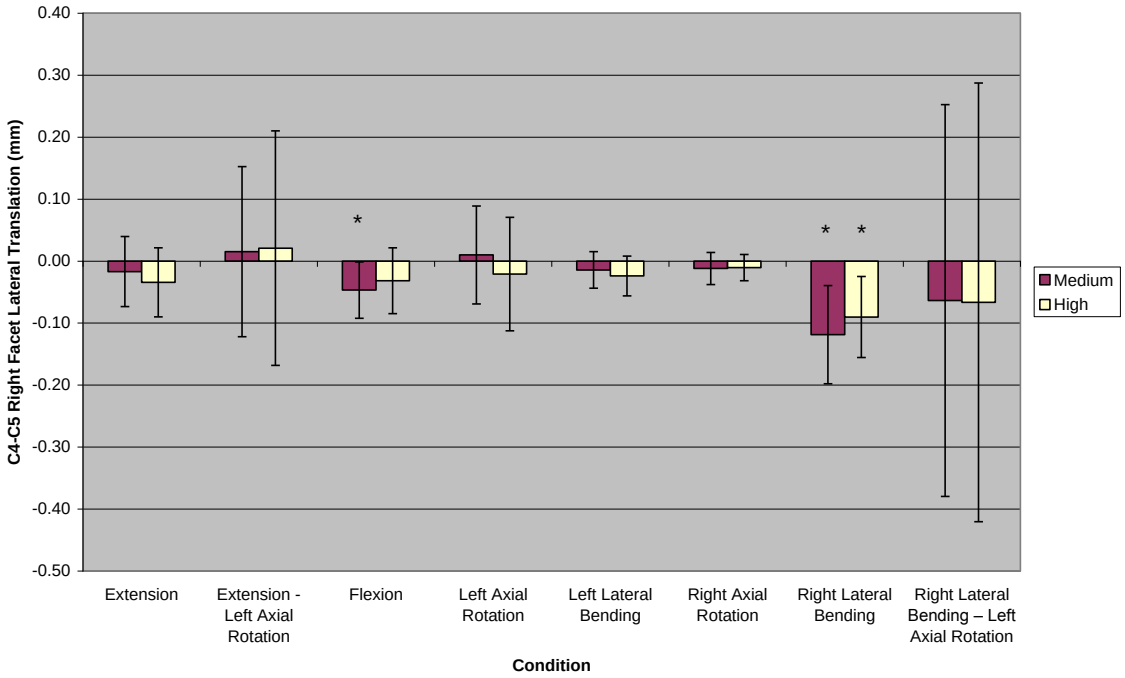


Figure 81: Neural Feedback Controller Effect on C4-C5 Right Facet Lateral Translation

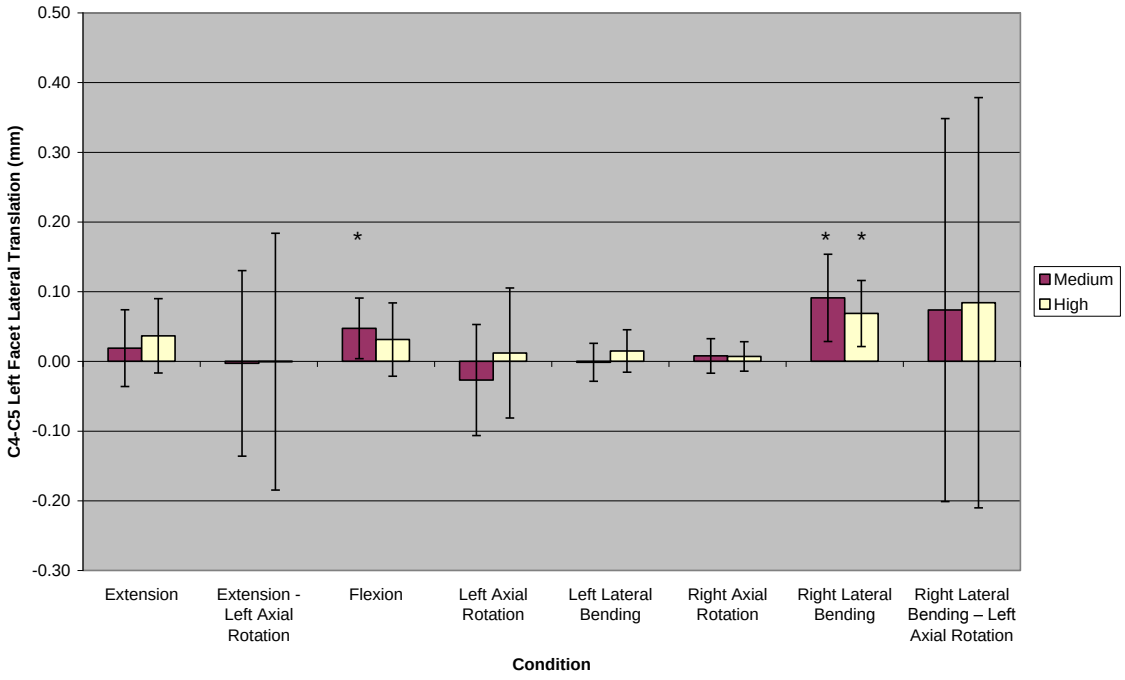


Figure 82: Neural Feedback Controller Effect on C4-C5 Left Facet Lateral Translation

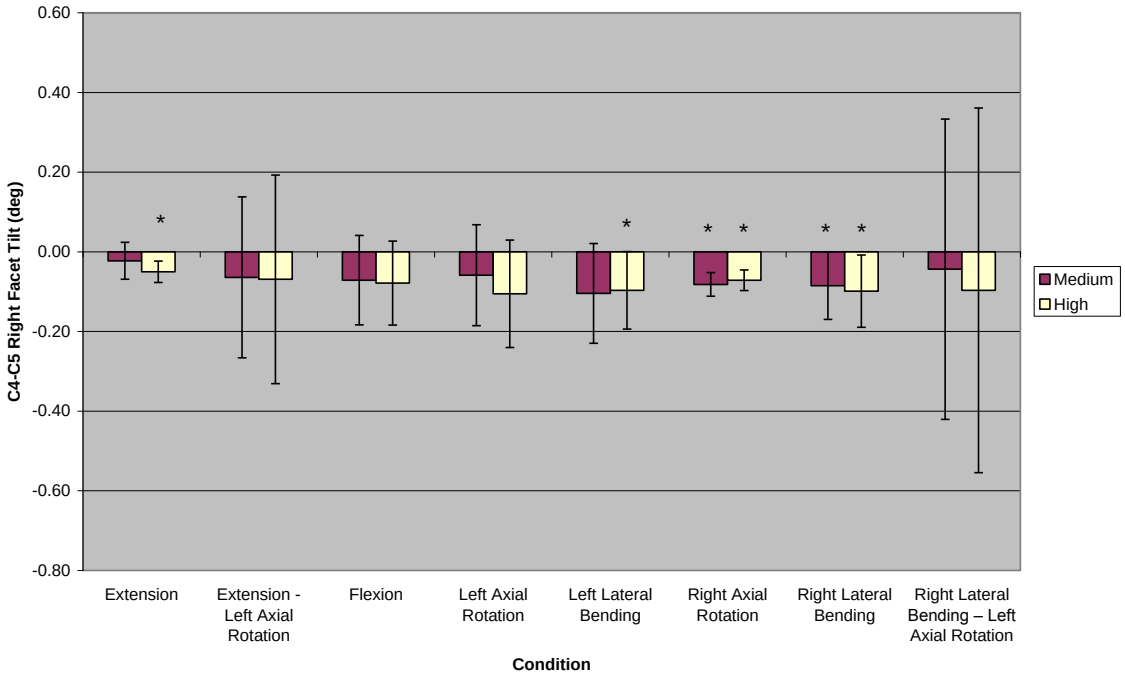


Figure 83: Neural Feedback Controller Effect on C4-C5 Right Facet Tilt

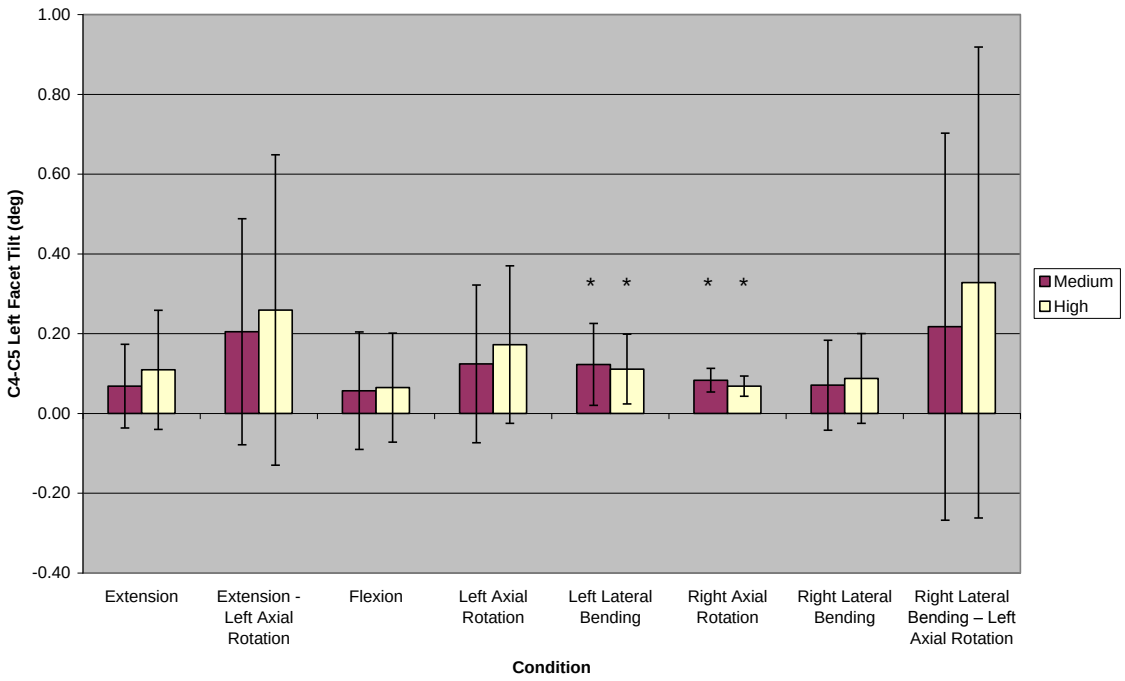


Figure 84: Neural Feedback Controller Effect on C4-C5 Left Facet Tilt

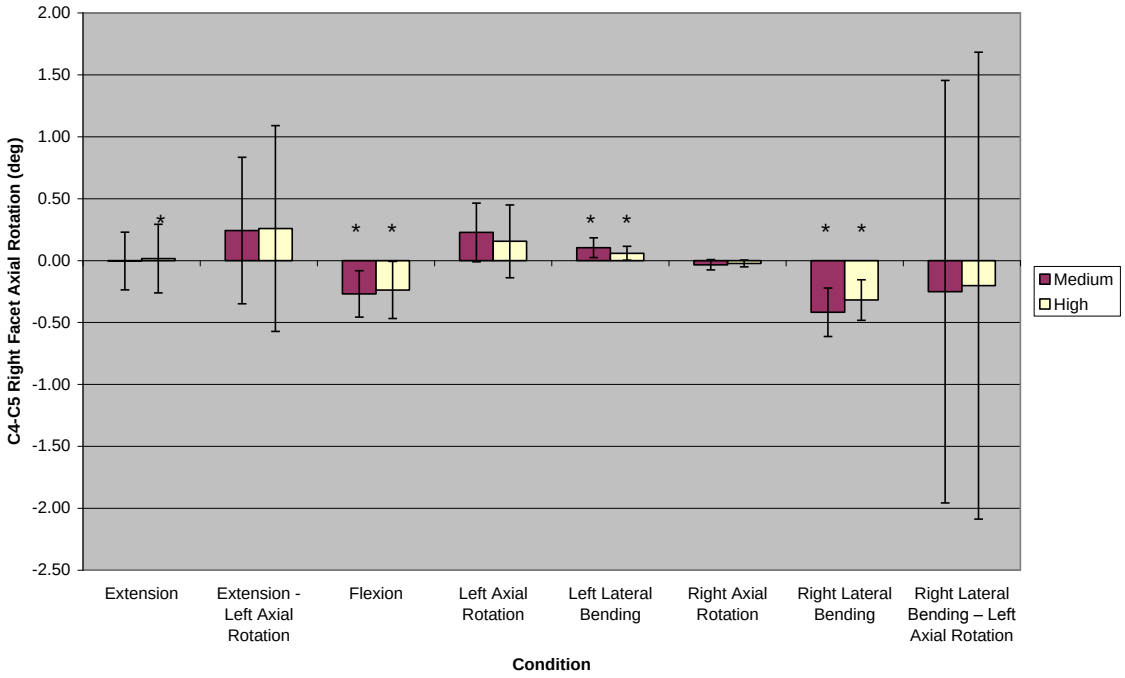


Figure 85: Neural Feedback Controller Effect on C4-C5 Right Facet Axial Rotation

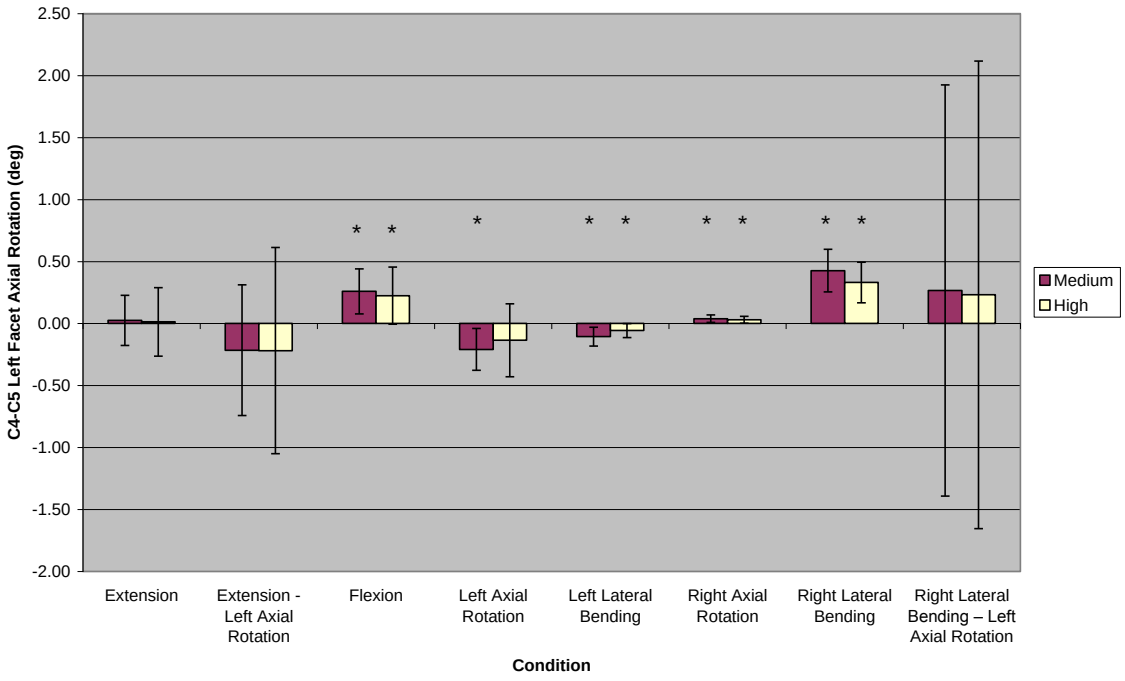


Figure 86: Neural Feedback Controller Effect on C4-C5 Left Facet Axial Rotation

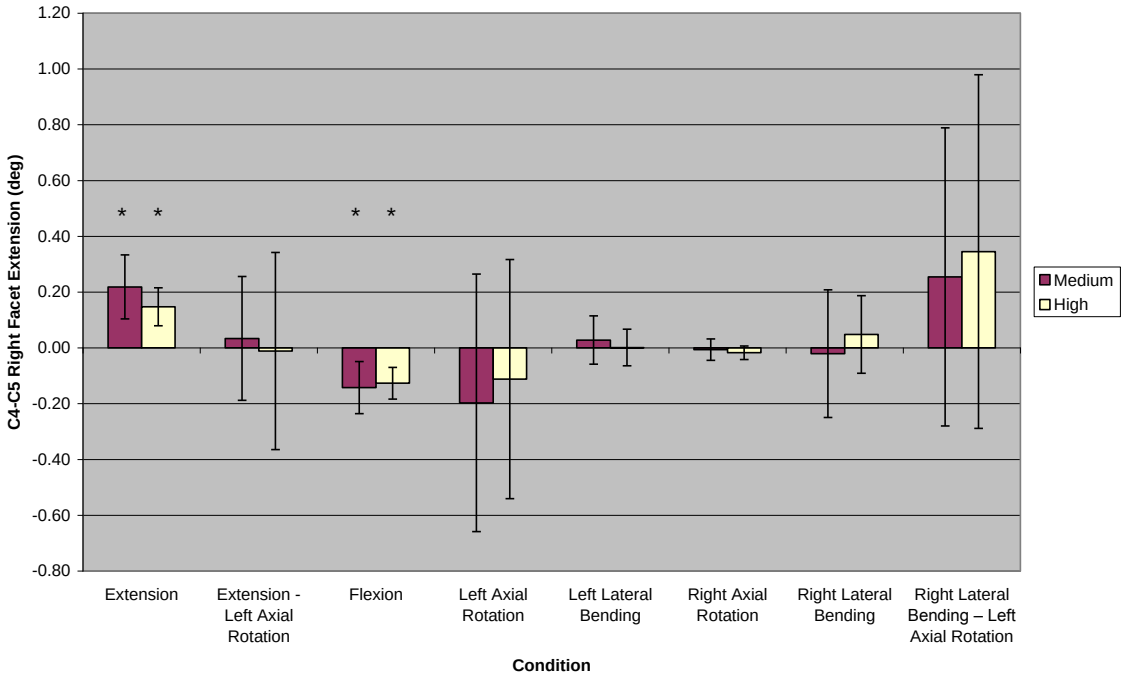


Figure 87: Neural Feedback Controller Effect on C4-C5 Right Facet Extension

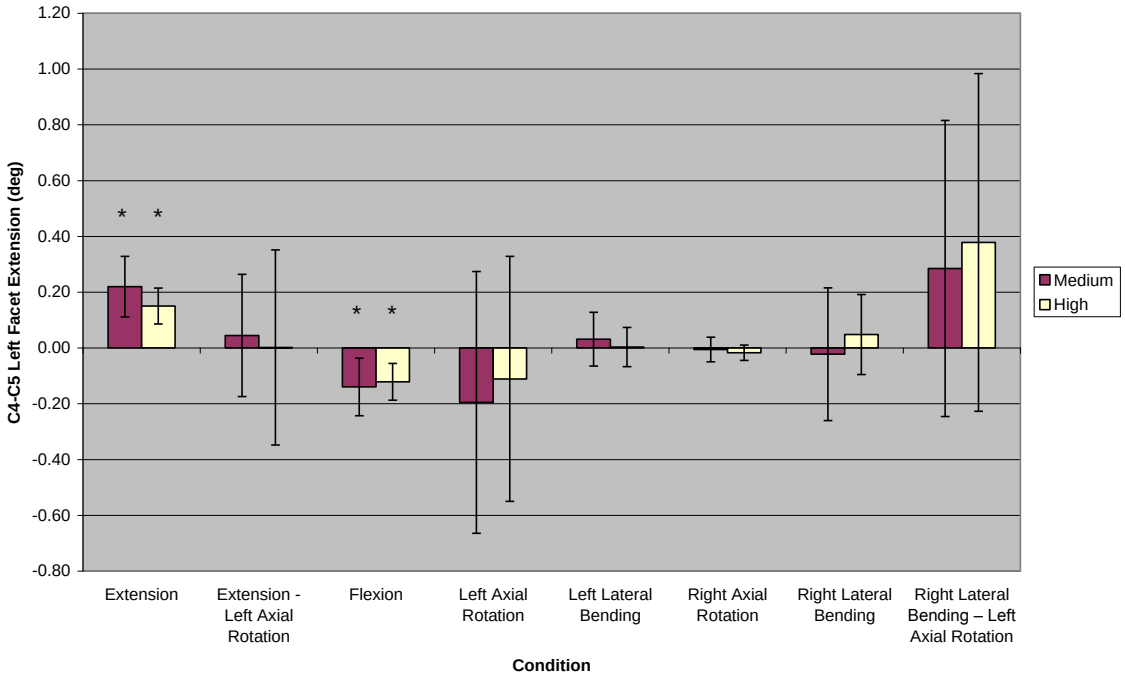


Figure 88: Neural Feedback Controller Effect on C4-C5 Left Facet Extension

Figures 77 – 88 show that changes in facet kinematics were significant across several loading conditions. There also seemed to be an equal and opposite response in most of the loading conditions. For example, for the axial separation states (Figures 77 and 78) the compensatory motion for reducing right facet force was to increase the axial separation in the right facet, and decrease it in the left facet. The same phenomenon occurred in the other degrees of freedom with the exception of extension (Figures 87 and 88). In this case, the compensatory motions were very similar. Table VIII is the qualitative accumulation of the significant compensatory facet kinematics for both sides for each loading condition. The table allows for a quick reference of which compensatory motions were common among all specimens to help reduce force (simulated pain) in the right facet.

Table VIII: Facet Compensatory Kinematics for Simulated Right Facet Pain

	Anterior Sliding		Axial Separation		Lateral Shear		Tilt		Axial Rotation		Extension	
Loading Condition	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right
Extension		x						x			x	x
Extension - Left Axial Rotation	x		x									
Flexion	x		x		x	x			x	x	x	x
Left Axial Rotation							x					
Right Lateral Bending	x	x	x		x	x		x	x	x		
Right Lateral Bending – Left Axial Rotation				x								

Similar to tables VI and VII, table VIII should not be interpreted in isolation. Figure 89 is a 3D plot that combines the “high” sensitivity condition data from Figures 77-88 and Table VIII.

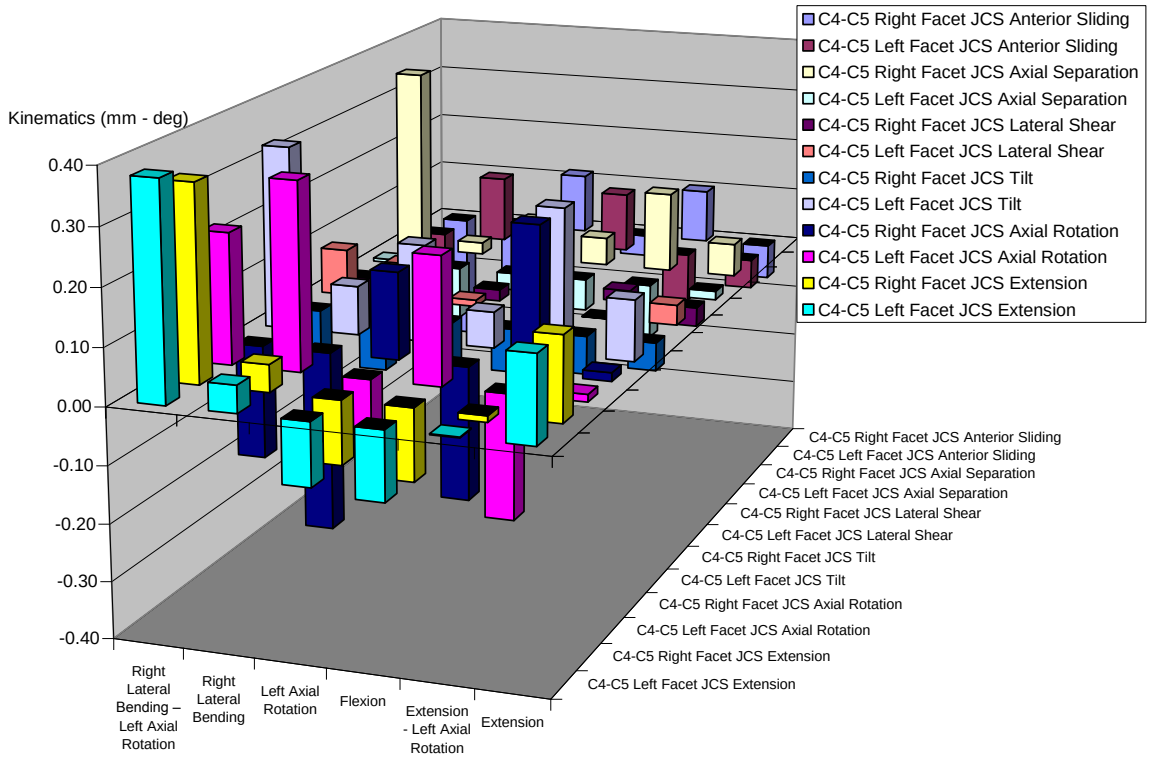


Figure 89: Neural Feedback Controller Effect on C4-C5 Facet Kinematics

Figure 89 is similar to a map that shows the regions of high effect, such as those in the Right Lateral Bending – Left Axial Rotation condition. Other relationships can be established using this type of plot. For example, the paired response of the right and left facets in extension, for both magnitude and sign, can be seen in the first two rows of data in the foreground. This suggests that the flexion-extension compensatory response is similar between the two facet kinematic states.

Section 6.06 Repeated Pairs f-distribution

To verify that the averaging of the repeated tests was a statistically valid method, a histogram of the standard deviations of the repeated pairs was created for each of the 32 recorded states. Figure 90 is a representative plot for the Posterior Shear Force kinetic state. In this and the other plots the f-distribution is evident suggesting the method is valid.

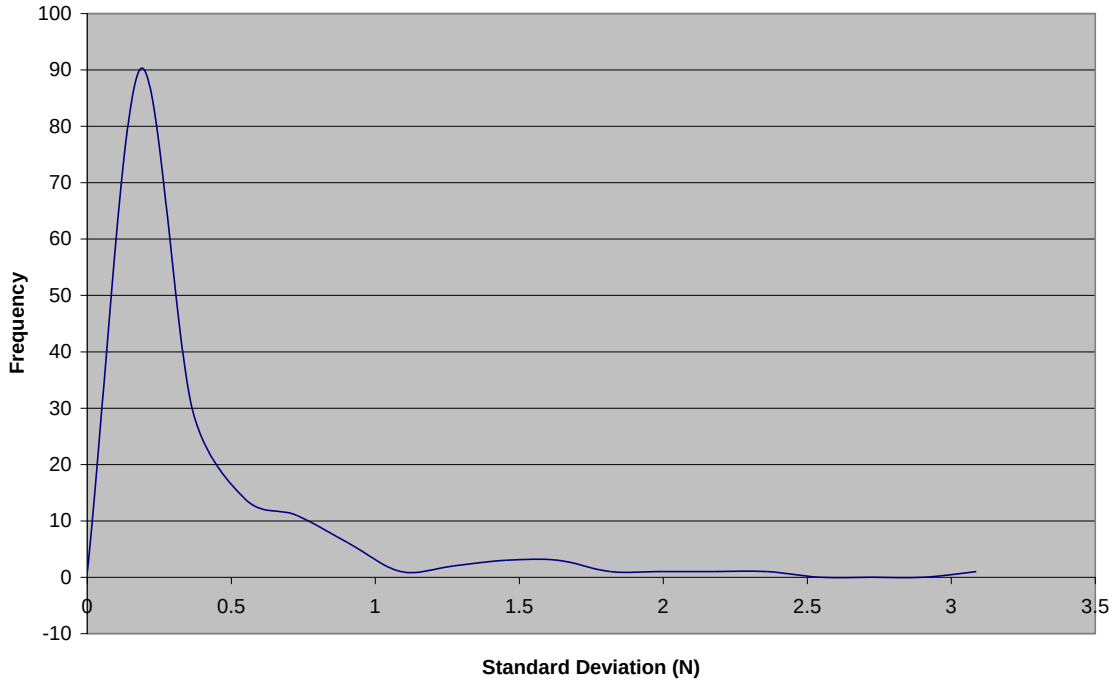


Figure 90: Distribution of the Standard Deviations of the Posterior Shear Force Repeated Averages

CHAPTER VII

DISCUSSION

Section 7.01 Efficacy of the Neuromusculoskeletal Controller

One goal in the development of the Neuro-Musculoskeletal Simulator was to estimate how natural pain avoidance of the patient could transfer the load to other structures and potentially increase the risk for other problems. These problems could be along the lines of excessive joint force, ligament/capsular strains, annular strains, or fractures [101] among others. They would most likely be chronic in nature as the repeated pain avoidance responses could create abnormal loading patterns. The novel NMS described in this work has demonstrated, through simulation and cadaveric experimentation, that it is able to incorporate data from external sensors (e.g. force, motion tracking) to modulate spinal motion and loading patterns. In addition, the NMS exhibited the ability to use an estimated nociceptive response in unilateral facet arthritis to elucidate statistically significant compensatory kinetic and kinematic changes. One of those changes was to reduce the simulated pain in the "painful" facet by decreasing the facet contact force by 51% (Fig 55). The response was found to be non-linear with

respect to the gains of the controller. This suggests that, for the simulated pathology, small nociceptive inputs may result in significant compensatory responses.

Other notable compensatory kinetic changes were spine shear load increases as high as 14.7 N (Fig. 57) and off-axis torque increases of 0.46 Nm (Fig. 60). These correspond to an increase of approximately 37% and 18% of the total applied force and torque respectively. In addition to the NMS identifying significant differences that may be clinically relevant, the system was also able to identify changes that were statistically significant but likely clinically insignificant. For example, in the Table C1, force and torque changes as small as 0.23 N and 0.01 Nm were found to be significant ($p = 0.05$ and $p = 0.028$ respectively). In Table C2, kinematic changes as small as 0.01 mm and 0.02 deg were identified as significant ($p = 0.009$ and $p = 0.049$ respectively). The ability for the NMS to identify such small changes suggests the high level of resolution and repeatability the system is able to provide to researchers who will use it to answer clinical questions.

Section 7.02 Clinical Implications of Compensatory Responses

This is the first known study to attempt to model and simulate compensatory mechanisms for unilateral facet arthritis. To transfer the findings of this study into clinical practice would be premature without future studies. However, it does provide insights for understanding potential implications of this particular pathology. These insights are based on the assumption that the LUT provided solutions that were somewhat representative of *in vivo* responses. The work by Collins [97] comparing distinctly

different gait optimization algorithms that predicted remarkably similar patterns of muscle activity over the gait cycle suggests that the LUT solution may be a reasonable outcome.

One of the insights gained during this study was that there were significant changes in the loads between the "no" sensitivity and the "high" and "medium" sensitivity conditions. The 37% increase in applied forces and the 18% increase in applied torques suggest there may be non-trivial clinical implications to these compensatory responses. These increased loads were transferred to other structures. Our hypothesis was that one such structure to bear the loads would be the left facet. However, no significant increases were found in the left facet force (Fig. 56). In two loading conditions, flexion and extension, the averages showed slight increases in left facet force. These increases were likely due to the fact that the loading conditions were not biasing the load to one facet or the other. When the right facet would unload, it is conceivable that some load could be transferred to the left. This theory is also supported by Figures 77 and 78, which showed that the compensatory kinematic change in the right facet was increased Axial Separation while the change in the left facet was decreased Axial Separation. This was true for all the loading conditions. If the left facet was not already in contact, then the other vertebral kinematic degrees of freedom (e.g. posterior translation) were able to compensate and transfer the load to the disc or ligaments before the left facet articulating surfaces would contact and bear the load. Exactly which structures were bearing these loads is uncertain and is worthy of future studies.

Another discovery from this work, albeit unintentional, was the measureable reduction in facet force that the facet osteophytes provided. For the spines with osteophytes needing removal, the arthritis is clear. Typically, this level of degeneration presents as a sign of other issues. The disc was likely degenerated, which increased the loads on the facets and then eventually formed the facet osteophytes. The lesson for me was that compensatory changes to pathologies can include morphological changes.

Section 7.03 Limitations

The concept of what the NMS is simulating is not trivial and it would be naïve to suggest that this is the final tool required to fully understand the interplay of the 3 sub-systems of spinal stability. Motor control is a very complex problem, even if it involves just an agonist and antagonist muscle acting on a hinge joint. In the cervical spine, there are multiple DOF and centers of rotation along with a very large number of motor units. For this reason, it is easy to see that one potential limitation of this study is that the simplifications and assumptions made have the potential to simplify the problem to a level to which the results carry little meaning. I believe that this is not the case and the choices made with respect to sensors, the motion tracking system, the robot, and the control algorithm are sufficiently complex to elucidate useful answers applicable to some clinical questions.

When clinicians have reviewed proposals regarding the use of the NMS, a typical response is, “You are not measuring pain. This is not that simple.” I emphatically agree. However, this should not be a reason to not try. Even though pain is not easily measured,

quantified, standardized, localized, or explained, it is not impossible. In a recent New England Journal of Medicine article, Wager et al. developed a technique for quantifying heat related pain by identifying a pattern of Functional Magnetic Resonance Imaging activity across brain regions [102]. The NMS-simulated "nociceptive" responses are limited to models of mechanically induced pain. They are also limited to all the assumptions outlined in section 2.07 including:

1. Assume facet osteoarthritis has a nociceptive response related to force applied to the joint articulating surface.
2. Assume sleeping nociceptors in the subchondral bone are activated due to the inflammation caused by joint loading without articular cartilage.
3. Assume joint force increases are proportional to nociceptive response increases which are proportional to increases in perceived pain.
4. Assume that Humphrey's [71] mechanotransduction model applies wherein force, as measured by the sensor, increases with mechanical loading as does the likelihood that stresses will rise near nociceptive free nerve endings.

Limitations 1 and 2 apply specifically to this work regarding osteoarthritis. In addition, assumptions 3 and 4 apply to future uses of the NMS for pain based responses. Consider, however, that the NMS is not just limited to pain-related responses. Mechanoreceptor-based neural responses are the model for NMS feedback. Other uses could include proprioceptive or vibration based inputs. The mechanoreceptor/nociceptor model used for this study was assumed to be a slowly adapting (sustained) type where the action potential output would not change with time. However, some mechanoreceptors

can be rapidly adaptive and mainly respond to transient stimuli. This was not modeled, but future work could incorporate such features.

Another limitation to this study was the ability to measure facet force. The presence and subsequent removal of osteophytes introduced a specimen-based variation that had a large impact on the output of the facet force used as the input to the neural feedback controller. In fact, the range of measured right facet forces across all specimens was 7 to 113 N for the combined Right Lateral Bending - Left Axial Rotation "no" sensitivity loading condition. The large variance has direct implications on the controller output and this is likely the source of the large standard deviations in the compensatory responses. Secondly, even though large portions of this variance can be attributed to the osteophytes [99-100], some portions of this variation can be attributed to the calibration technique. It is important to recall that the goal of the experiment was not to quantify facet contact force. Instead, the study was designed to show that facet contact force can be modified via sensor feedback and, for that purpose, relative changes are valid measures. The third problem with the facet measurement was the fragility of the sensors. The sensors were not robust and it was not uncommon to have a sensor stop working during an experimental run. This required re-testing and recalibration of the sensors. In some cases, sensors were post-calibrated as a matter of practicality. In some of those cases, the sensor output was significantly different than the sensor that failed. The averaged runs always used the same sensor and calibration factor. The analyzed data has the correctly calibrated values, but the control system was using a sensor output with different sensitivity in the feedback loop. The implications are that while this is unlikely

to have affected the values reported for the facet force, the magnitude of the compensatory neural feedback would have been different. This limitation was unlikely to change the major conclusions of this study as the variation in peak facet forces (Table C1) due to specimen differences was much greater than the variations in the calibration factors (Table III).

The order of the applied loading conditions was not randomized and this may contribute to some limitations in the interpretation of the results. The repeated measures helped to reduce some of the risk, though. One potential problem with switching the order of pain feedback sensitivity always from "no", to "high", to "medium", while maintaining a constant load, is that creep in the soft tissues may be a source of some of the statistically significant, low-magnitude changes in loading conditions that did not apply any load to the right facet (i.e. Left Lateral Bending and Right Axial Rotation). It may not be that the compensatory effects produced kinematic changes of 0.03 deg. It may simply be that under this loading condition the tissues would creep an additional 0.03 deg while held. Because the system is under load control, the phenomenon will not be evident in the kinetic plots. However, the kinematic plots can provide evidence of this. For example, in Figures 69 and 71, it is evident that the "medium" sensitivity changes are greater than the "high" sensitivity changes for some loading conditions. Recall that the "medium" condition followed "high" by 18 seconds. This counterintuitive result of "medium" moving more could be real or it could be a creep artifact. Fortunately, knowing this is possible can provide reassurance with the validity of the results for the loading conditions where the "high" sensitivity resulted in greater motion.

It also helps to bound the effect by knowing that the creep effects are unlikely to be greater than the small differences between "medium" and "high" changes such as those found in Figures 69 and 71.

Section 7.04 Future work: Engineering

Completing a project of this magnitude and understanding all the limitations and assumptions that went into it make most engineers want to improve upon it in the future. One assumption made was that the neural feedback controller could not vary the superior translation of the vertebrae. This was because it was obvious that it was not physiological to have muscles extend and distract the neck. There may have been other non-obvious compensatory motions that should also be limited by the controller output. One way to identify these would be to insert a rigid body model of the full cervical spine, complete with muscle elements, in the process of creating the LUT. The relationship between the muscles and the 6-DOF motions of the C2 – T1 vertebrae would need to be established. However, by using this technique, the steepest ascent vector from the LUT would be better bounded by physiological solutions.

The NMS framework has the potential for incorporating a large variety of sensors to serve as a surrogate for neural feedback. All sensors that could be used are going to be many orders of magnitude greater in size than the mechanoreceptors they are imitating. In addition, adding more sensors will increase the risk of experimental unreliability. The output of mechanoreceptors is not always monolithic information provided to the CNS. Action potentials from neighboring receptors are combined to provide a greater wealth of

information to the CNS. It is possible to imagine the creation of a new state object in the NMS system that is based on the output of a virtual sensor. The virtual sensor may be a finite element model of a single nociceptor where the model boundary conditions are based on the real sensor outputs of position and loads. Similar work has been done by Halloran et al. in combining rigid body modeling with deformable finite element models [103]. In this case, the rigid body model would be replaced with a real specimen, and the deformable model would be a small targeted area where a physical sensor would be unable to serve as a surrogate for the mechanoreceptor. In addition, the finite element model doesn't need to represent just a single nociceptor. The model could be a distributed framework of virtual sensors working in concert to drive compensatory responses in the NMS. Having virtual sensors would also allow for multiple pathologies to be simulated on the same spine where surgical interventions are applied. By the press of a button, the test could be transformed from one set of neural responses to another. These virtual sensor networks could also be combined to simulate compounding pathologies.

Section 7.05 Future work: Clinical

Based on the new UMS architecture, the creation of the NMS is the addition of a controller object focused on the simulated neural mechanoreceptor feedback. This controller object is intended to be a simplified representation of neural pathways that affect the spine motion. As mentioned in the background chapter, it is understood that this is a very simplified representation and is not intended to be used as a measure of validation that these responses are similar to how humans may modify their motion. This

type of work is intended for the future and requires the NMS framework to exist to even attempt those types of studies.

Other future studies envisioned include measuring disc force during these types of pain-modulated simulations. It has been shown that metabolism of the cells in the intervertebral disc can be inhibited by very low and high pressures [104]. In addition, high pressures stimulate the production of matrix degrading enzymes [105]. Therefore, as the mechanical loading exceeds some limit, it can start to degenerate a disc. Under this pretense, it may be possible to use intradiscal pressure measurements to predict the potential for disc degeneration. Where the link between pressure and pain is indirect via the following chain: High Intervertebral Pressure -> Disc Degeneration -> Disc Pain. The stress profiling method [106] presented the possibility that chronic back pain was due to a decompressed nucleus pulposus combined with multiple stress concentrations in the annulus. Considering the nerve endings in the outer anulus, along with the stress concentrations, this is likely a source for discogenic pain. It is also understood that mechanical stimulation can sensitize nerve endings due to degenerated nucleus pulposus cells releasing cytokines. This concept suggests that spine pain is modulated by both mechanical and chemical effects [107]. Discogenic pain is not likely from a single source, and combining all the nociceptors in the disc to estimate compensatory responses would be a useful application of the virtual sensor network framework. It need not all be limited to a single structure either. With real or virtual sensors, the neural response from the facets and the disc can be combined to study interactions of these pathologies. Virtual sensors could be added on as a way to include estimates of muscle nociceptive

responses. The NMS, while not the final solution, may have the ability to help unveil some of the unknown aspects of compensatory responses and is not limited to the facet joint question focused on in this study.

Tables V – VIII and figures 75, 76, and 89 may provide the roadmap for how this data can be used for clinical correlation. These data map specific compensatory responses to the loading conditions for a given pathology. It may be possible to use this data to validate the NMS responses. It also could be used in reverse for diagnostic purposes. For a given set of potential pathologies, a patient's movements could be analyzed and mapped back to a most likely diagnosis or pain source. A certain combination of motions that differ from a "normal" population dataset could be identified. Or for unilateral problems, comparison of left and right motions may be used, with the patient serving as their own control. The measurement may be a coarse range of motion of the head, or it may need to include some fluoroscopic data that measures movement of specific structures. If head motions are sufficient, the link between the compensatory motion and the pathology data tables could be the heart of a front-line test for non-operative health care providers. A low-cost instrumented hat device with accelerometer-based motion capture, when coupled with the pathology mapping table, could allow for preliminary interpretation of the problem that will direct future testing and follow-up care.

Other ideas for future uses for this technology include applications in robotically assisted surgery where an NMS framework, including virtual sensors, could serve as an

additional feedback mechanism that could potentially improve safety or surgical techniques (e.g. de-rotation of scoliotic spines). The uses of the NMS are not limited to spine. The current framework allows for easy transfer of the techniques to other joints. One can imagine that the complex interplay of the bones and muscles of the ankle and foot may be modulated with neural feedback that modifies muscle activation patterns during gait.

In applying the NMS technologies to future clinical questions, there could be a secondary use to the resulting compensatory responses. As was seen with the facet osteophytes in this study, compensatory changes to pathologies can include morphological changes. The NMS could be a tool to work backwards to identify the pathology that may have initiated morphological changes. Fujie used estimates of anterior cruciate ligament loads in the knee to build finite element models to suggest that the formation of the Resident's ridge bone structure near the ligament attachment can be biomechanically explained by the ligament force-induced bone remodeling [108]. In a similar fashion, compensatory kinetics and kinematics could be used in such studies.

Section 7.06 Conclusions

The NMS presented is a novel development that improves upon the current state of the art for spine biomechanics research. What is presented is the first successful iteration of this new type of biomechanical simulator. It is fully expected that future versions of these types of simulators will greatly exceed the functionality and usefulness of this one by using greater computational power, more representative compensatory response algorithms, refined sensors, and improved control algorithms. It is also expected that other researchers will develop improved systems and techniques to incorporate estimates of neural feedback into musculoskeletal biomechanical testing. It is important to consider that while one goal of this project was to analyze potential implications of unilateral facet arthritis, the main goal was to produce and demonstrate the Neuro-Musculoskeletal Simulator which forms the basis for asking and answering a myriad of clinical questions.

REFERENCES

[1] M. M. Panjabi, "Clinical spinal instability and low back pain," *J Electromyogr Kinesiol,* vol. 13, pp. 371-9, Aug 2003.

[2] A. G. Sawa and N. R. Crawford, "The use of surface strain data and a neural networks solution method to determine lumbar facet joint loads during in vitro spine testing," *J Biomech,* vol. 41, pp. 2647-53, Aug 28 2008.

[3] N. R. Crawford, *et al.*, "An apparatus for applying pure nonconstraining moments to spine segments in vitro," *Spine (Phila Pa 1976),* vol. 20, pp. 2097-100, Oct 1 1995.

[4] J. T. Lysack, *et al.*, "A continuous pure moment loading apparatus for biomechanical testing of multi-segment spine specimens," *J Biomech,* vol. 33, pp. 765-70, Jun 2000.

[5] S. Vadapalli, *et al.*, "Effect of lumbar interbody cage geometry on construct stability: a cadaveric study," *Spine (Phila Pa 1976),* vol. 31, pp. 2189-94, Sep 1 2006.

[6] W. Tawackoli, *et al.*, "The effect of compressive axial preload on the flexibility of the thoracolumbar spine," *Spine (Phila Pa 1976),* vol. 29, pp. 988-93, May 1 2004.

[7] M. M. Panjabi, "Hybrid multidirectional test method to evaluate spinal adjacent-level effects," *Clin Biomech (Bristol, Avon),* vol. 22, pp. 257-65, Mar 2007.

[8] M. M. Panjabi, *et al.*, "Critical load of the human cervical spine: an in vitro experimental study," *Clin Biomech (Bristol, Avon),* vol. 13, pp. 11-17, Jan 1998.

[9] T. Miura, *et al.*, "A method to simulate in vivo cervical spine kinematics using in vitro compressive preload," *Spine (Phila Pa 1976),* vol. 27, pp. 43-8, Jan 1 2002.

[10] A. G. Patwardhan, *et al.*, "Load-carrying capacity of the human cervical spine in compression is increased under a follower load," *Spine (Phila Pa 1976),* vol. 25, pp. 1548-54, Jun 15 2000.

[11] A. G. Patwardhan, *et al.*, "A follower load increases the load-carrying capacity of the lumbar spine in compression," *Spine (Phila Pa 1976),* vol. 24, pp. 1003-9, May 15 1999.

[12] S. M. Renner, *et al.*, "Novel model to analyze the effect of a large compressive follower pre-load on range of motions in a lumbar spine," *J Biomech,* vol. 40, pp. 1326-32, 2007.

[13] P. A. Cripton, *et al.*, "In vitro axial preload application during spine flexibility testing: towards reduced apparatus-related artefacts," *J Biomech,* vol. 33, pp. 1559-68, Dec 2000.

[14] H. S. Ahn and D. J. DiAngelo, "Biomechanical testing simulation of a cadaver spine specimen: development and evaluation study," *Spine (Phila Pa 1976),* vol. 32, pp. E330-6, May 15 2007.

[15] B. W. Cunningham, *et al.*, "The effect of spinal destabilization and instrumentation on lumbar intradiscal pressure: an in vitro biomechanical analysis," *Spine (Phila Pa 1976),* vol. 22, pp. 2655-63, Nov 15 1997.

[16] P. Gedet, *et al.*, "Minimizing errors during in vitro testing of multisegmental spine specimens: considerations for component selection and kinematic measurement," *J Biomech,* vol. 40, pp. 1881-5, 2007.

[17] K. A. Gillespie and J. P. Dickey, "Biomechanical role of lumbar spine ligaments in flexion and extension: determination using a parallel linkage robot and a porcine model," *Spine (Phila Pa 1976),* vol. 29, pp. 1208-16, Jun 1 2004.
[18] B. P. Kelly, *et al.*, "Biomechanical comparison of a novel C1 posterior locking plate with the harms technique in a C1-C2 fixation model," *Spine (Phila Pa 1976),* vol. 33, pp. E920-5, Nov 15 2008.
[19] H. Schmidt, *et al.*, "Which axial and bending stiffnesses of posterior implants are required to design a flexible lumbar stabilization system?," *J Biomech,* vol. 42, pp. 48-54, Jan 5 2009.
[20] I. A. Stokes, *et al.*, "Measurement of a spinal motion segment stiffness matrix," *J Biomech,* vol. 35, pp. 517-21, Apr 2002.
[21] M. R. Walker and J. P. Dickey, "New methodology for multi-dimensional spinal joint testing with a parallel robot," *Med Biol Eng Comput,* vol. 45, pp. 297-304, Mar 2007.
[22] H. J. Wilke, *et al.*, "A universal spine tester for in vitro experiments with muscle force simulation," *Eur Spine J,* vol. 3, pp. 91-7, 1994.
[23] R. Alberts, *et al.*, "Single-unit artificial intervertebral disc," *J Neurosurg Spine,* vol. 1, pp. 95-100, Jul 2004.
[24] H. de Visser, *et al.*, "A robotic testing facility for the measurement of the mechanics of spinal joints," *Proc Inst Mech Eng H,* vol. 221, pp. 221-7, Apr 2007.
[25] L. G. Gilbertson, *et al.*, "Improvement of accuracy in a high-capacity, six degree-of-freedom load cell: application to robotic testing of musculoskeletal joints," *Ann Biomed Eng,* vol. 27, pp. 839-43, Nov-Dec 1999.
[26] D. J. Goertzen and G. N. Kawchuk, "A novel application of velocity-based force control for use in robotic biomechanical testing," *J Biomech,* vol. 42, pp. 366-9, Feb 9 2009.
[27] G. N. Kawchuk, *et al.*, "Identification of spinal tissues loaded by manual therapy: a robot-based serial dissection technique applied in porcine motion segments," *Spine (Phila Pa 1976),* vol. 35, pp. 1983-90, Oct 15 2010.
[28] T. L. Schulte, *et al.*, "The effect of dynamic, semi-rigid implants on the range of motion of lumbar motion segments after decompression," *Eur Spine J,* vol. 17, pp. 1057-65, Aug 2008.
[29] L. Tian, "Intelligent self-tuning of PID control for the robotic testing system for human musculoskeletal joints test," *Ann Biomed Eng,* vol. 32, pp. 899-909, Jun 2004.
[30] L. Tian, "Control methods and the performance of the robotic testing system for human musculoskeletal joints," *Ann Biomed Eng,* vol. 32, pp. 889-98, Jun 2004.
[31] L. Tian, "An intelligent control method based on fuzzy logic for a robotic testing system for the human spine," *J Biomech Eng,* vol. 127, pp. 807-12, Oct 2005.
[32] L. Tian and L. G. Gilbertson, "The study of control methods for the robotic testing system for human musculoskeletal joints," *Comput Methods Programs Biomed,* vol. 74, pp. 211-20, Jun 2004.
[33] L. G. Gilbertson, *et al.*, "New Methods To Study Lumbar Spine Biomechanics: Delineation Of In Vitro Load-Displacement Characteristics By Using A Robotic/Ufs Testing System With Hybrid Control," *Operative Techniques in Orthopaedics,* vol. 10, pp. 246-253, 2000.

[34] H. Gray. (1918, March 10th). *Anatomy of the Human Body*. Available: Bartleby.com, 2000. http://www.bartleby.com/107/illus111.html and http://www.bartleby.com/107/illus385.html
[35] D. B. Jenkins and W. H. Hollinshead, *Hollinshead's functional anatomy of the limbs and back,* 7th ed. Philadelphia: Saunders, 1998.
[36] H. Brisby, "Pathology and possible mechanisms of nervous system response to disc degeneration," *J Bone Joint Surg Am,* vol. 88 Suppl 2, pp. 68-71, Apr 2006.
[37] H. G. Schaible, *et al.*, "Joint pain," *Exp Brain Res,* vol. 196, pp. 153-62, Jun 2009.
[38] A. K. Reinthal, *et al.*, "Improved ambulation and speech production in an adolescent post-traumatic brain injury through a therapeutic intervention to increase postural control," *Pediatr Rehabil,* vol. 7, pp. 37-49, Jan-Mar 2004.
[39] R. A. Schmidt, "Motor schema theory after 27 years: reflections and implications for a new theory," *Res Q Exerc Sport,* vol. 74, pp. 366-75, Dec 2003.
[40] S. Roberts, *et al.*, "Mechanoreceptors in intervertebral discs. Morphology, distribution, and neuropeptides," *Spine (Phila Pa 1976),* vol. 20, pp. 2645-51, Dec 15 1995.
[41] A. Murai, *et al.*, "Modeling and identification of human neuromusculoskeletal network based on biomechanical property of muscle," *Conf Proc IEEE Eng Med Biol Soc,* vol. 2008, pp. 3706-9, 2008.
[42] R. T. Raikova and B. I. Prilutsky, "Sensitivity of predicted muscle forces to parameters of the optimization-based human leg model revealed by analytical and numerical analyses," *J Biomech,* vol. 34, pp. 1243-55, Oct 2001.
[43] D. G. Thelen, *et al.*, "Neuromusculoskeletal models provide insights into the mechanisms and rehabilitation of hamstring strains," *Exerc Sport Sci Rev,* vol. 34, pp. 135-41, Jul 2006.
[44] H. S. Bahat, *et al.*, "The effect of neck pain on cervical kinematics, as assessed in a virtual environment," *Arch Phys Med Rehabil,* vol. 91, pp. 1884-90, Dec 2010.
[45] J. AbouHassan, *et al.*, "Can postural modification reduce kinetic and kinematic loading during the bowing postures of Islamic prayer?," *Ergonomics,* vol. 53, pp. 1446-54, Dec 2010.
[46] W. S. Marras, *et al.*, "The role of dynamic three-dimensional trunk motion in occupationally-related low back disorders. The effects of workplace factors, trunk position, and trunk motion characteristics on risk of injury," *Spine (Phila Pa 1976),* vol. 18, pp. 617-28, Apr 1993.
[47] J. D. Dubois, *et al.*, "Effect of experimental low back pain on neuromuscular control of the trunk in healthy volunteers and patients with chronic low back pain," *J Electromyogr Kinesiol,* Jun 2 2011.
[48] G. L. Shum, *et al.*, "Back pain is associated with changes in loading pattern throughout forward and backward bending," *Spine (Phila Pa 1976),* vol. 35, pp. E1472-8, Dec 1 2010.
[49] M. A. Adams, "Biomechanics of back pain," *Acupunct Med,* vol. 22, pp. 178-88, Dec 2004.
[50] R. F. McLain, "Mechanoreceptor endings in human cervical facet joints," *Iowa Orthop J,* vol. 13, pp. 149-54, 1993.
[51] R. F. McLain and K. Raiszadeh, "Mechanoreceptor endings of the cervical, thoracic, and lumbar spine," *Iowa Orthop J,* vol. 15, pp. 147-55, 1995.

[52] R. F. McLain and J. G. Pickar, "Mechanoreceptor endings in human thoracic and lumbar facet joints," *Spine (Phila Pa 1976),* vol. 23, pp. 168-73, Jan 15 1998.
[53] C. Chen, *et al.*, "Distribution of A-delta and C-fiber receptors in the cervical facet joint capsule and their response to stretch," *J Bone Joint Surg Am,* vol. 88, pp. 1807-16, Aug 2006.
[54] J. M. Cavanaugh, *et al.*, "Pain generation in lumbar and cervical facet joints," *J Bone Joint Surg Am,* vol. 88 Suppl 2, pp. 63-7, Apr 2006.
[55] K. M. Szadek, *et al.*, "Possible nociceptive structures in the sacroiliac joint cartilage: An immunohistochemical study," *Clin Anat,* vol. 23, pp. 192-8, Mar 2010.
[56] M. Hussain, *et al.*, "Reduction in segmental flexibility because of disc degeneration is accompanied by higher changes in facet loads than changes in disc pressure: a poroelastic C5-C6 finite element investigation," *Spine J,* vol. 10, pp. 1069-77, Dec 2010.
[57] H. B. Chen, *et al.*, "Biomechanics of whiplash injury," *Chin J Traumatol,* vol. 12, pp. 305-14, Oct 2009.
[58] L. Oliveira, *et al.*, "Cervical Facet Degeneration After Total Disc Replacement: 280 Levels in 162 Patients—5-Year Follow-up," *Neurosurgery Quarterly,* vol. 21, pp. 17-21 10.1097/WNQ.0b013e3182039fae, 2011.
[59] C. S. Kuo, *et al.*, "Biomechanical analysis of the lumbar spine on facet joint force and intradiscal pressure--a finite element study," *BMC Musculoskelet Disord,* vol. 11, p. 151, 2010.
[60] T. P. Hedman, "A new transducer for facet force measurement in the lumbar spine: benchmark and in vitro test results," *J Biomech,* vol. 25, pp. 69-80, Jan 1992.
[61] C. A. Niosi, *et al.*, "The effect of dynamic posterior stabilization on facet joint contact forces: an in vitro investigation," *Spine (Phila Pa 1976),* vol. 33, pp. 19-26, Jan 1 2008.
[62] D. C. Wilson, *et al.*, "Accuracy and repeatability of a new method for measuring facet loads in the lumbar spine," *J Biomech,* vol. 39, pp. 348-53, 2006.
[63] C. M. Wiseman, *et al.*, "The effect of an interspinous process implant on facet loading during extension," *Spine (Phila Pa 1976),* vol. 30, pp. 903-7, Apr 15 2005.
[64] U. K. Chang, *et al.*, "Changes in adjacent-level disc pressure and facet joint force after cervical arthroplasty compared with cervical discectomy and fusion," *J Neurosurg Spine,* vol. 7, pp. 33-9, Jul 2007.
[65] G. R. Buttermann, *et al.*, "An experimental method for measuring force on the spinal facet joint: description and application of the method," *J Biomech Eng,* vol. 113, pp. 375-86, Nov 1991.
[66] M. J. Schendel, *et al.*, "Experimental measurement of ligament force, facet force, and segment motion in the human lumbar spine," *J Biomech,* vol. 26, pp. 427-38, Apr-May 1993.
[67] J. B. Welcher, *et al.*, "Development of a versatile intra-articular pressure sensing array," *Med Eng Phys,* Apr 6 2011.

[68] N. V. Jaumard, *et al.*, "Pressure measurement in the cervical spinal facet joint: considerations for maintaining joint anatomy and an intact capsule," *Spine (Phila Pa 1976)*, vol. 36, pp. 1197-203, Jul 1 2011.
[69] J. J. Liau, *et al.*, "The influence of inserting a Fuji pressure sensitive film between the tibiofemoral joint of knee prosthesis on actual contact characteristics," *Clin Biomech (Bristol, Avon)*, vol. 16, pp. 160-6, Feb 2001.
[70] T. A. Zdeblick, *et al.*, "Cervical stability after sequential capsule resection," *Spine (Phila Pa 1976)*, vol. 18, pp. 2005-8, Oct 15 1993.
[71] J. D. Humphrey, "Stress, strain, and mechanotransduction in cells," *J Biomech Eng*, vol. 123, pp. 638-41, Dec 2001.
[72] W. K. Barsoum, *et al.*, "Robotic testing of proximal tibio-fibular joint kinematics for measuring instability following total knee arthroplasty," *J Orthop Res*, vol. 29, pp. 47-52, Jan 2011.
[73] V. Musahl, *et al.*, "Varying femoral tunnels between the anatomical footprint and isometric positions: effect on kinematics of the anterior cruciate ligament-reconstructed knee," *Am J Sports Med*, vol. 33, pp. 712-8, May 2005.
[74] N. Diermann, *et al.*, "Rotational instability of the knee: internal tibial rotation under a simulated pivot shift test," *Arch Orthop Trauma Surg*, vol. 129, pp. 353-8, Mar 2009.
[75] S. L. Woo, *et al.*, "Use of robotic technology for diathrodial joint research," *J Sci Med Sport*, vol. 2, pp. 283-97, Dec 1999.
[76] L. D. Noble, Jr., *et al.*, "Design and validation of a general purpose robotic testing system for musculoskeletal applications," *J Biomech Eng*, vol. 132, p. 025001, Feb 2010.
[77] D. G. Lee and B. L. Davis, "Assessment of the effects of diabetes on midfoot joint pressures using a robotic gait simulator," *Foot Ankle Int*, vol. 30, pp. 767-72, Aug 2009.
[78] D. V. Boguszewski, *et al.*, "Investigating the effects of anterior tibial translation on anterior knee force in the porcine model: is the porcine knee ACL dependent?," *J Orthop Res*, vol. 29, pp. 641-6, May 2011.
[79] R. W. Colbrunn, *et al.*, "Impingement and Stability of Total Hip Arthroplasty Versus Femoral Head Resurfacing Using a Cadaveric Robotics Model," *J Orthop Res*, p. 0, Mar 13 2013.
[80] C. F. Scifert, *et al.*, "Experimental and computational simulation of total hip arthroplasty dislocation," *Orthop Clin North Am*, vol. 32, pp. 553-67, vii, Oct 2001.
[81] S. G. Kaar, *et al.*, "Effect of humeral head defect size on glenohumeral stability: a cadaveric study of simulated Hill-Sachs defects," *Am J Sports Med*, vol. 38, pp. 594-9, Mar 2010.
[82] M. Wellmann, *et al.*, "Effect of coracoacromial ligament resection on glenohumeral stability under active muscle loading in an in vitro model," *Arthroscopy*, vol. 24, pp. 1258-64, Nov 2008.
[83] N. A. Sharkey and A. J. Hamel, "A dynamic cadaver model of the stance phase of gait: performance characteristics and kinetic validation," *Clin Biomech (Bristol, Avon)*, vol. 13, pp. 420-433, Sep 1998.

[84] Z. M. Li, *et al.*, "A robot-assisted study of intrinsic muscle regulation on proximal interphalangeal joint stiffness by varying metacarpophalangeal joint position," *J Orthop Res,* vol. 24, pp. 407-15, Mar 2006.
[85] F. Techy, *et al.*, "Properties of an interspinous fixation device (ISD) in lumbar fusion constructs: a biomechanical study," *Spine J,* Mar 13 2013.
[86] J. C. Dick, *et al.*, "A biomechanical comparison evaluating the use of intermediate screws and cross-linkage in lumbar pedicle fixation," *J Spinal Disord,* vol. 7, pp. 402-7, Oct 1994.
[87] J. Kikkawa, *et al.*, "Biomechanical evaluation of a posterolateral lumbar disc arthroplasty device: an in vitro human cadaveric model," *Spine (Phila Pa 1976),* vol. 35, pp. 1760-8, Sep 1 2010.
[88] P. Mageswaran, *et al.*, "Hybrid dynamic stabilization: a biomechanical assessment of adjacent and supraadjacent levels of the lumbar spine," *J Neurosurg Spine,* vol. 17, pp. 232-42, Sep 2012.
[89] J. A. Tang, *et al.*, "Pure moment testing for spinal biomechanics applications: fixed versus 3D floating ring cable-driven test designs," *J Biomech,* vol. 45, pp. 706-10, Feb 23 2012.
[90] D. J. Sutton, *et al.*, "Object Interaction Language (OIL): An intent-based language for programming self-organized sensor/actuator networks," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 6113-6118.
[91] J. Budenske and M. Gini, "Sensor explication: knowledge-based robotic plan execution through logical objects," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 27, pp. 611-625, 1997.
[92] J. Domaszewicz, *et al.*, "ROVERS: pervasive computing platform for heterogeneous sensor-actuator networks," in *World of Wireless, Mobile and Multimedia Networks, 2006. WoWMoM 2006. International Symposium on a*, 2006, pp. 6 pp.-620.
[93] M. D. Naish and E. A. Croft, "Open architecture for intelligent multisensor integration in industrial applications," pp. 33-41, 1997.
[94] G. Wu, *et al.*, "ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion--part I: ankle, hip, and spine. International Society of Biomechanics," *J Biomech,* vol. 35, pp. 543-8, Apr 2002.
[95] M. H. Raibert and J. J. Craig, "Hybrid Position/Force Control of Manipulators," *Journal of Dynamic Systems, Measurement, and Control,* vol. 103, pp. 126-133, 1981.
[96] F. Chinello, *et al.*, "KCT: a MATLAB toolbox for motion control of KUKA robot manipulators," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 4603-4608.
[97] J. J. Collins, "The redundant nature of locomotor optimization laws," *J Biomech,* vol. 28, pp. 251-67, Mar 1995.
[98] M. Al-Rawahi, *et al.*, "Mechanical function of vertebral body osteophytes, as revealed by experiments on cadaveric spines," *Spine (Phila Pa 1976),* vol. 36, pp. 770-7, May 1 2011.

[99] E. Toh, *et al.*, "The effect of anterior osteophytes and flexural position on thoracic trabecular strain," *Spine (Phila Pa 1976)*, vol. 26, pp. 22-6, Jan 1 2001.
[100] N. V. Jaumard, *et al.*, "Contact pressure in the facet joint during sagittal bending of the cadaveric cervical spine," *J Biomech Eng*, vol. 133, p. 071004, Jul 2011.
[101] K. Siemionow, *et al.*, "Identifying serious causes of back pain: cancer, infection, fracture," *Cleve Clin J Med*, vol. 75, pp. 557-66, Aug 2008.
[102] T. D. Wager, *et al.*, "An fMRI-based neurologic signature of physical pain," *N Engl J Med*, vol. 368, pp. 1388-97, Apr 11 2013.
[103] J. P. Halloran, *et al.*, "Adaptive surrogate modeling for efficient coupling of musculoskeletal control and tissue deformation models," *J Biomech Eng*, vol. 131, p. 011014, Jan 2009.
[104] H. Ishihara, *et al.*, "Effects of hydrostatic pressure on matrix synthesis in different regions of the intervertebral disk," *J Appl Physiol*, vol. 80, pp. 839-46, Mar 1996.
[105] T. Handa, *et al.*, "Effects of hydrostatic pressure on matrix synthesis and matrix metalloproteinase production in the human lumbar intervertebral disc," *Spine (Phila Pa 1976)*, vol. 22, pp. 1085-91, May 15 1997.
[106] M. A. Adams, *et al.*, "The internal mechanical functioning of intervertebral discs and articular cartilage, and its relevance to matrix biology," *Matrix Biol*, vol. 28, pp. 384-9, Sep 2009.
[107] K. Olmarker, *et al.*, "Changes in spontaneous behavior in rats exposed to experimental disc herniation are blocked by selective TNF-alpha inhibition," *Spine (Phila Pa 1976)*, vol. 28, pp. 1635-41; discussion 1642, Aug 1 2003.
[108] H. Y. Fujie, S; Nakata, K; Shino, K "Resident's Ridge Formation Due To ACL Force-Induced Bone Remodeling," presented at the Orthopedic Research Society, San Antonio, Texas, 2013.
[109] M. M. Panjabi, *et al.*, "Cervical facet joint kinematics during bilateral facet dislocation," *Eur Spine J*, vol. 16, pp. 1680-8, Oct 2007.

Appendix

APPENDIX A. KINEMATIC CHAIN EQUATIONS

The following equation is derived from Figure A1 for the foot and ankle simulator.

$$\mathbf{T}_{ROB,MIC} = \mathbf{T}_{ROB,PLA}(\mathbf{q}) \cdot \mathbf{T}_{PLA,GND} \cdot \mathbf{T}_{GND,TIB}(a,m,s,r,t,o) \cdot \mathbf{T}_{TIB,MIC}$$

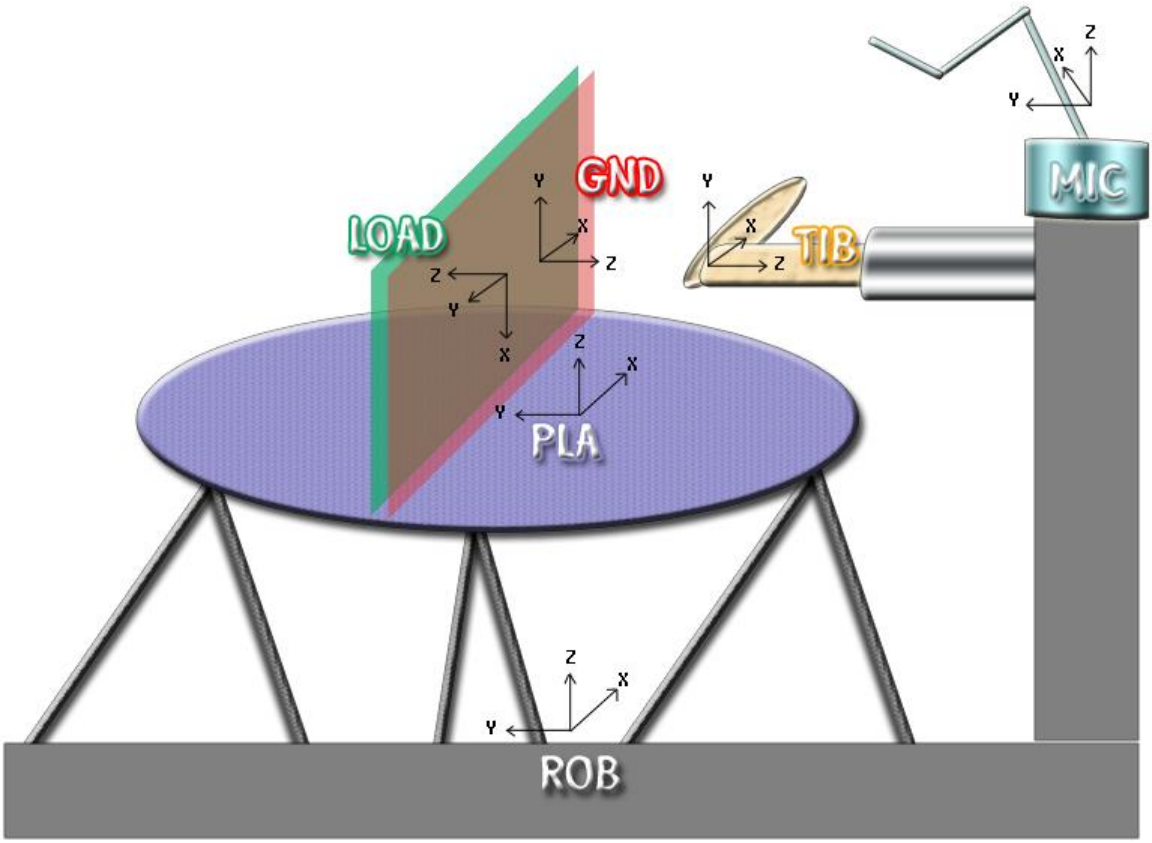


Figure A1. Coordinate systems used in this document for the foot and ankle simulator.

The following equation is the derived from Figure A2. This is the generic static/dynamic bone configuration. Each equation in the derivation is a reduction of terms to minimize static transformations.

$$\mathbf{T}_{ROB,MIC} = \mathbf{T}_{ROB,PLA}(q) \cdot \mathbf{T}_{PLA,FIX}(\theta) \cdot \mathbf{T}_{FIX,HOL}(\phi) \cdot \mathbf{T}_{HOL,DYB} \cdot \mathbf{T}_{DYB,STB}(\bar{\mathrm{g}}) \cdot \mathbf{T}_{STB,MIC}$$

$$\mathbf{T}_{ROB,STB} = \mathbf{T}_{ROB,PLA}(q) \cdot \mathbf{T}_{PLA,FIX}(\theta) \cdot \mathbf{T}_{FIX,HOL}(\phi) \cdot \mathbf{T}_{HOL,DYB} \cdot \mathbf{T}_{DYB,STB}(\bar{\mathrm{g}})$$

$$\mathbf{T}_{ROB,STB} = \mathbf{T}_{ROB,PLA}(q) \cdot \mathbf{T}_{PLA,FIX}(\theta) \cdot \mathbf{T}_{FIX,DYB}(\phi) \cdot \mathbf{T}_{DYB,STB}(\bar{\mathrm{g}})$$

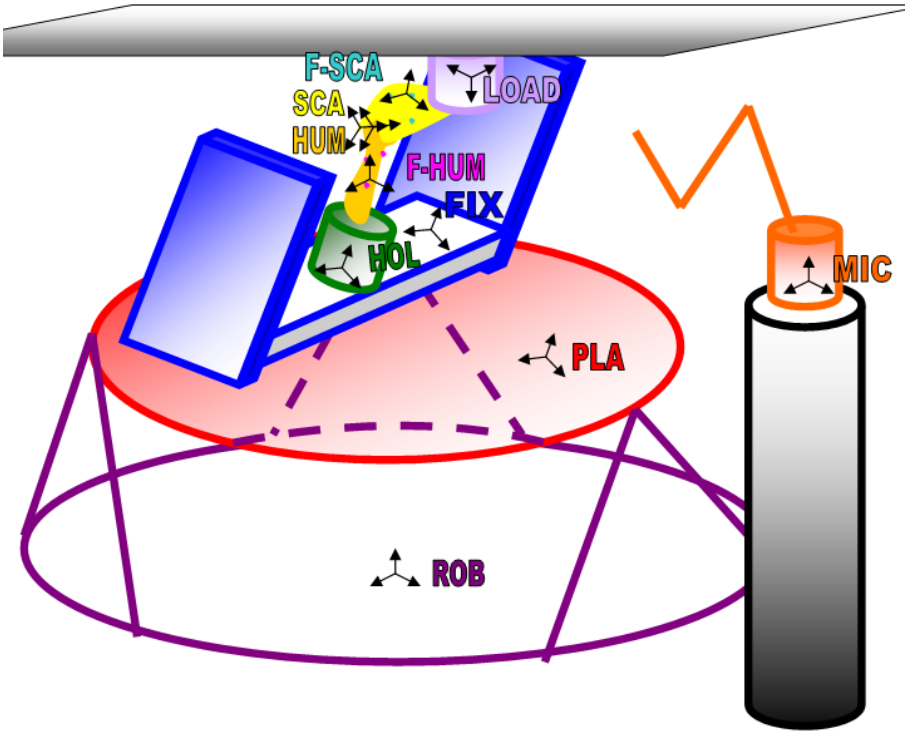


Figure A2. Coordinate systems used in this document for a generic two bone joint simulator. Note: The fixture coordinate frame is directly above the holder coordinate and the z-axes are aligned, not on the side of the fixture as suggested by the sketch

The following equation is the derived from Figure A2 and A3. This is the generic static/dynamic bone configuration regardless of robot and of specimen. However, this kinematic chain only applies to controlling 6-DOF robot position in order to achieve a given 6-DOF two bone Joint Coordinate System. Additional DOF for the robot or bone can be included, but lower level optimization must take place in order to have a unique solution.

For Generic:

$$\mathbf{T}_{ROB,WORLD} = \mathbf{T}_{ROB,HOL}(\vec{q}) \cdot \mathbf{T}_{HOL,DYB} \cdot \mathbf{T}_{DYB,STB}(\vec{g}) \cdot \mathbf{T}_{STB,WORLD}$$

Additional kinematics for the Mikrolar 8 axis system (robot, stage, holder)

$$\mathbf{T}_{ROB,HOL}(\vec{q}) = \mathbf{T}_{ROB,PLA}(\vec{h}) \cdot \mathbf{T}_{PLA,FIX}(\theta) \cdot \mathbf{T}_{FIX,HOL}(\phi)$$

For Kinetics Generic:

$\mathbf{T}_{STB,STLD} = T^{-1}_{WORLD,STB} \cdot \mathbf{T}_{WORLD,STLD}$ to find STB load using static load cell

$\mathbf{T}_{DYB,DYLD} = T^{-1}_{WORLD,DYB} \cdot \mathbf{T}_{WORLD,DYLD}$ to find DYB load using dynamic load cell. Assumes that the WORLD based transformation matrices were collected when robot was in same position. Gravity compensation will need to be applied to the output of the load cell.

$\mathbf{T}_{STB,DYLD} = \mathbf{T}^{-1}_{DYB,STB}(\vec{g}) \cdot \mathbf{T}_{DYB,DYLD}$ to find STB load using dynamic load cell

$\mathbf{T}_{DYB,STLD} = \mathbf{T}_{DYB,STB}(\vec{g}) \cdot \mathbf{T}_{STB,STLD}$ to find DYB load using static load cell

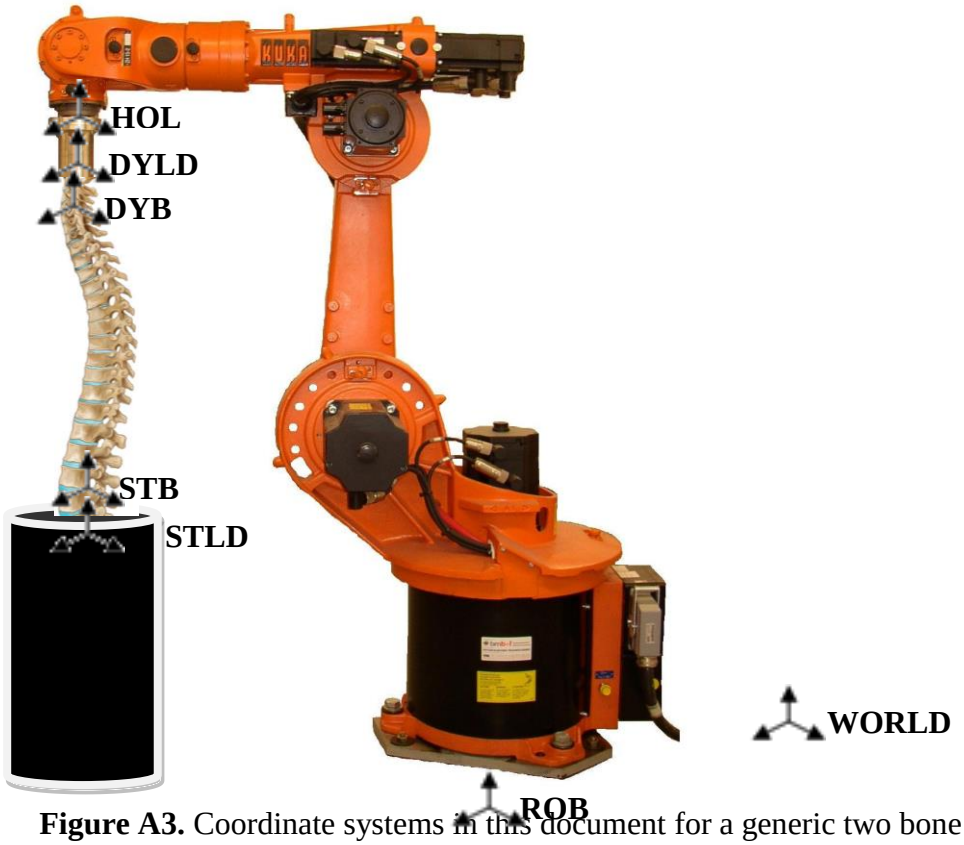


Figure A3. Coordinate systems in this document for a generic two bone joint simulator.

APPENDIX B. COORDINATE SYSTEM DEFINITIONS

Notations

$\mathbf{r}_P^A = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$	Position of a point P, expressed in coordinate frame A
$\mathbf{T}_{A,B} = \begin{pmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$	Matrix to transform B-coordinates into A-coordinates: $\mathbf{r}_P^A = \mathbf{T}_{A,B} \cdot \mathbf{r}_P^B$ and $\mathbf{T}_{A,C} = \mathbf{T}_{A,B} \cdot \mathbf{T}_{B,C}$
$\mathbf{R}_{A,B} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix}$	Rotational part of $\mathbf{T}_{A,B}$
$\mathbf{t}_{A,B} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$	Translational part of $\mathbf{T}_{A,B}$. Physical meaning: the position of B's origin expressed in the A reference frame.
$\mathbf{F}^A = \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix}$	Force expressed in components relative to coordinate frame A. The symbol $\mathbf{M}^A$ is defined similarly.

B.1 UMS_Make T_ROB_HOL

Purpose:
Generates the transformation matrix between the non-moving robot coordinate system ROB and the end of the robot that holds whatever is attached to it HOL. This matrix is dynamic and is a function of the 6-DOF robot position coordinates q.

Algorithm:

$$\mathbf{q} = (q_1, q_2, q_3, q_4, q_5, q_6) \text{ (x,y,z,roll,pitch,yaw)}$$

$$\mathbf{T}_{ROB,HOL}(\mathbf{q}) = \begin{pmatrix} \cos q_6 & -\sin q_6 & 0 & q_1 \\ \sin q_6 & \cos q_6 & 0 & q_2 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos q_5 & 0 & \sin q_5 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin q_5 & 0 & \cos q_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos q_4 & -\sin q_4 & 0 \\ 0 & \sin q_4 & \cos q_4 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos q_6 & -\sin q_6 & 0 & q_1 \\ \sin q_6 & \cos q_6 & 0 & q_2 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos q_5 & \sin q_5 \sin q_4 & \sin q_5 \cos q_4 & 0 \\ 0 & \cos q_4 & -\sin q_4 & 0 \\ -\sin q_5 & \cos q_5 \sin q_4 & \cos q_5 \cos q_4 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos q_6 \cos q_5 & \cos q_6 \sin q_5 \sin q_4 - \sin q_6 \cos q_4 & \cos q_6 \sin q_5 \cos q_4 + \sin q_6 \sin q_4 & q_1 \\ \sin q_6 \cos q_5 & \sin q_6 \sin q_5 \sin q_4 + \cos q_6 \cos q_4 & \sin q_6 \sin q_5 \cos q_4 - \cos q_6 \sin q_4 & q_2 \\ -\sin q_5 & \cos q_5 \sin q_4 & \cos q_5 \cos q_4 & q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

B.2 UMS_Make T_WORLD_ROB

Purpose:
Generates the transformation matrix between the non-moving robot coordinate system ROB and the world coordinate system WORLD. This matrix is constant as long as the position digitization system remains in the same place. A1 is the neutral position of the robot. A2-A7 are the positions of a common point on the robot when it is translated from neutral to translations in x, y, z (A2-A4) and rotated in the roll, pitch, yaw orientations (A5-A7).

Algorithm:
1. The x-axis is defined by the normalized vector pointing from A_1 to A_2.

$$\vec{X}_{ROB} = \frac{\vec{A}_2 - \vec{A}_1}{\left|\left(\vec{A}_2 - \vec{A}_1\right)\right|}$$

2. Make the temporary y-axis be defined by the normalized vector pointing from A_1 to A_3.

$$\vec{Y}_{ROB-Temp} = \frac{\vec{A}_3 - \vec{A}_1}{\left|\left(\vec{A}_3 - \vec{A}_1\right)\right|}$$

3. The z-axis is the axis mutually perpendicular to both the X and Y axes.

$$\vec{Z}_{ROB} = \vec{X}_{ROB} \times \vec{Y}_{ROB-Temp}$$

4. Make the final Y axis such that all axes are orthogonal.

$$\vec{Y}_{ROB} = \vec{Z}_{ROB} \times \vec{X}_{ROB}$$

5. Generate three 3x3 matrices R_x, R_y, and R_z representing rotation alpha about the X_{ROB}, Y_{ROB}, and Z_{ROB} axes. Alpha is the amount of rotation done by the robot to collect points A5-A7.

R_x = axis_angle(X_{ROB},alpha). See elsewhere for function "axis_angle" that generates a rotation matrix.

R_y= axis_angle(Y_{ROB},alpha)

R_z = axis_angle(Z_{ROB},alpha)

6. Solve 3D coordinates of robot origin from nine equations: P_rotated = O_{ROB} + R*(P-O_{ROB}), applied to all three rotations R. The 9 equations are not independent, but we let least squares deal with that as follows:

a. Make 3x3 identity matrix I

b. Make a 9x3 matrix $A = \begin{bmatrix} I - R_x \\ I - R_y \\ I - R_z \end{bmatrix}$

c. Make a 9x1 column vector $b = \begin{bmatrix} \vec{A}_5 - R_x \cdot \vec{A}_1 \\ \vec{A}_6 - R_y \cdot \vec{A}_1 \\ \vec{A}_7 - R_z \cdot \vec{A}_1 \end{bmatrix}$

d. Solve O_{ROB} from the overdetermined linear system $A*O_{ROB} = b$, using linear least squares method. Compute norm of residuals: norm($A*O_{ROB}$-b) and display on screen. Give warning if not sufficiently close to zero.

7. Put results in a 4x4 matrix

$$T_{WORLD,ROB} = \begin{bmatrix} \vec{X}_{ROB} & \vec{Y}_{ROB} & \vec{Z}_{ROB} & \vec{O}_{ROB} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

B.3 UMS_Matrix2Robot

Purpose:
Extract robot holder pose coordinates from the 4x4 T_ROB_HOL matrix.

Algorithm:

$\mathbf{q} = (q_1, q_2, q_3, q_4, q_5, q_6)$ (x,y,z,roll,pitch,yaw)

T_{rc} are the row and column coordinates of the T_ROB_HOL matrix

$$q_1 = T_{14}$$
$$q_2 = T_{24}$$
$$q_3 = T_{34}$$
$$q_4 = \text{atan2}(T_{33}, T_{32})$$
$$q_5 = \text{atan2}(\sqrt{T_{11}^2 + T_{21}^2}, -T_{31})$$
$$q_6 = \text{atan2}(T_{11}, T_{21})$$

B.4 Make JCS Transformation Matrix

Purpose:

This generates the transformation matrix between the most proximal (PROX) and most distal (DIS) vertebra. This can apply to a Functional Spinal Unit (FSU) or a series to connected vertebra. This dynamic matrix is a function of the 6-DOF Joint Coordinate System (JCS) rotations and translations.

References:

The JCS definition is based on the ISB 2002 standard [94] with one slight modification regarding the definition of the origin. The definition of the individual vertebra coordinate systems are defined below. The relative relationship between these coordinate systems establishes the JCS.

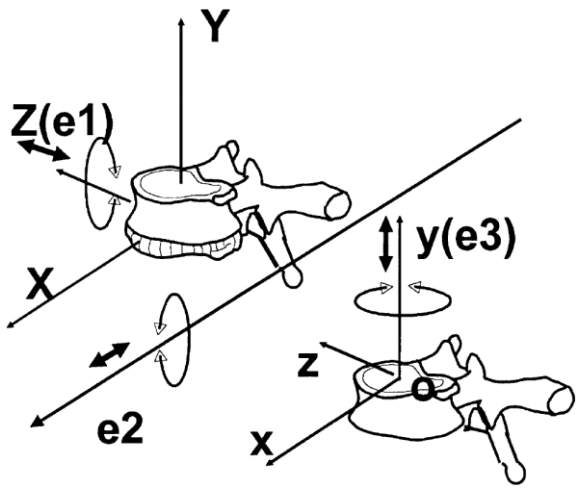


Figure B1. Vertebral Coordinate Systems

Vertebral coordinate system:

Superior (y): The line passing through the centers of the vertebra's upper and lower endplates, and pointing cephalad.

Lateral (z): The line parallel to a line joining similar landmarks on the bases of the right and left pedicles, and pointing to the right.

Anterior (x): The line perpendicular to the Y- and Z-axis, and pointing anteriorly.

Origin (o): The origin of the individual vertebra is a point along the y axis that is midpoint between the upper and lower endplates. Note: The ISB standard does not define an origin for an individual vertebra.

Joint Coordinate System:

Spine JCS contains the following components:

a	anterior translation of proximal vertebra
s	superior translation of proximal vertebra
l	lateral translation of proximal vertebra (right is positive)
β	lateral bending (tilt right is positive)
γ	axial rotation (head turn to the left is positive)
α	Extension

For the neuroscience discipline the following terminology is the translation to the biomechanical sciences.
Superior: Towards head = rostral
Inferior: Towards feet = caudal
Anterior: Forward = ventral
Posterior: Backward = dorsal

Origin (O): The origin of the joint is defined as a point that is the midpoint of the two vertebral origins that define the proximal and distal vertebra.
Note: The ISB standard defines the origin as the intersection of the proximal and distal y axes in the reference, neutral position. It requires that the neutral position must be specified, and must be in a position where the vertebral y axes are coplanar. If the y axes are parallel (do not intersect at the common origin O) the y axes are constrained to be collinear, and the origin O is the mid-point between adjacent endplates. Since the vertebral y axis from one vertebra to another are not guaranteed to be co-planar in a practical neutral position (i.e. zero load condition) it is proposed that a variation of the standard will be implemented. The axis intersection point will not be used and the mid-point between adjacent endplates will be estimated as the midpoint of the two vertebral origins. Though these are not guaranteed to be the same, they are likely close enough and will allow for multiple vertebral kinematics to be calculated without having to have two origins per vertebra.

Extension (α): The extension axis is the axis fixed to the proximal vertebra and coincident with the Z-axis of the proximal vertebra coordinate system. Extension is positive; Flexion is negative.

Lateral Bending (β): The lateral bending axis is the floating axis, the common axis perpendicular to the flexion and axial rotation axes. Right leaning is positive: Left leaning is negative.

Axial Rotation (γ): The axial rotation axis is the axis fixed to the distal vertebra and coincident with the y-axis of the distal vertebra coordinate system. Head turn to the left is positive, Head turn to the right is negative.

<u>Order of rotation and translation:</u>
Order of rotation and translations are important to understand and apply the kinematics of the JCS. The order of rotations and translations are given below.

1: α rotations and l translations are applied simulatenously

2: β rotations and *a* translations are applied simulatenously

3: γ rotations and *s* translations are applied simulatenously

Algorithm:

$$\mathbf{T}_{DIS,PROX}(a,s,l,\alpha,\beta,\gamma) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & \cos\beta & -\sin\beta & 0 \\ 0 & \sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\gamma & 0 & \sin\gamma & 0 \\ 0 & 1 & 0 & s \\ -\sin\gamma & 0 & \cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos\alpha & -\cos\beta\sin\alpha & \sin\alpha\sin\beta & a\cos\alpha \\ \sin\alpha & \cos\alpha\cos\beta & -\cos\alpha\sin\beta & a\sin\alpha \\ 0 & \sin\beta & \cos\beta & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\gamma & 0 & \sin\gamma & 0 \\ 0 & 1 & 0 & s \\ -\sin\gamma & 0 & \cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos\alpha\cos\gamma - \sin\alpha\sin\beta\sin\gamma & -\sin\alpha\cos\beta & \cos\alpha\sin\gamma + \sin\alpha\sin\beta\cos\gamma & a\cos\alpha - s\sin\alpha\cos\beta \\ \sin\alpha\cos\gamma + \cos\alpha\sin\beta\sin\gamma & \cos\alpha\cos\beta & \sin\alpha\sin\gamma - \cos\alpha\sin\beta\cos\gamma & a\sin\alpha + s\cos\alpha\cos\beta \\ -\cos\beta\sin\gamma & \sin\beta & \cos\beta\cos\gamma & l + s\sin\beta \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

B.5 Extract JCS Values From Transformation Matrix

Purpose:

After applying the kinematic chain equation of the system to determine the transformation matrix of the joint, the JCS values must be extracted so that the kinematics can be interpreted or controlled.

Algorithm:

Extract $(a, s, l, \beta, \gamma, \alpha)$ from the 4x4 matrix $T_{DIS,PROX}$. This is essentially the inverse of the make JCS transformation function. T_{rc} are the row and column coordinates of the $T_{DIS,PROX}$ matrix

$$\alpha = -\text{atan2}\left(\frac{T_{12}}{T_{22}}\right)$$

$$\gamma = -\text{atan2}\left(\frac{T_{31}}{T_{33}}\right)$$

$$\beta = \text{atan2}\left(\frac{T_{32}}{\sqrt{T_{12}^2 + T_{22}^2}}\right)$$ will be between –pi/2 +pi/2

$$a = T_{14}\cos\alpha + T_{24}\sin\alpha$$

$$s = -\frac{T_{14}\sin\alpha - T_{24}\cos\alpha}{\cos\beta}$$

$$l = T_{34} - s\sin\beta$$

B.6 Make Transformation Matrix WORLD_VERT

Purpose:

Create the transformation matrix of an individual vertebra relative to the digitizer world reference frame.

8 points will be collected.

E_1 = Most anterior point on the superior (rostral) end plate.

E_2 = Most anterior point on the inferior (caudal) end plate.

E_3 = Left most point on the superior (rostral) end plate of the vertebral body (not the pedicles)

E_4 = Left most point on the inferior (caudal) end plate of the vertebral body (not the pedicles)

E_5 = Right most point on the superior (rostral) end plate of the vertebral body (not the pedicles)

E_6 = Right most point on the inferior (caudal) end plate of the vertebral body (not the pedicles)

E_7 = Left most point on the transverse processes (select similar structure as E_8)

E_8 = Right most point on the transverse processes (select similar structure as E_7)

Algorithm:

1. The temporary y-axis is pointed superiorly along two points on the anterior surface of the vertebral body and is defined by the normalized vector pointing from E_2 to E_1.

$$\vec{E}_{y,Temp} = \frac{\vec{E}_1 - \vec{E}_2}{\left|\vec{E}_1 - \vec{E}_2\right|}$$

2. Adjust the superior end plate points by creating a superior end plate plane. This is normal to the temporary y axis and is the average of the distance between the superior amounts of the anterior and lateral points. This is because in the cervical spine the end plate is saddle shaped and not a plane that can be made with 3 points. This adjustment will be small with lumbar spine, and may not be needed. However, it should not cause any problems to do it and it will yield an algorithm that is flexible for all spinal regions.

$$\vec{E}_{y,\sup\_adjust} = \vec{E}_{y,Temp} \times \left[\frac{\vec{E}_{y,Temp} \cdot \left[\left(\frac{\vec{E}_3 + \vec{E}_5}{2} \right) - \vec{E}_1 \right]}{2} \right]$$

3. Apply the adjustment to points E1, E3, and E5 on the superior end plate.

$$\vec{E}_{1,adjust} = \vec{E}_1 + \vec{E}_{y,\sup\_adjust}$$

$$\vec{E}_{3,adjust} = \vec{E}_3 - \vec{E}_{y,\sup\_adjust}$$

$$\vec{E}_{5,adjust} = \vec{E}_5 - \vec{E}_{y,\sup\_adjust}$$

2. Adjust the inferior end plate points in the same manner.

$$\vec{E}_{y,\text{inf}\_adjust} = \vec{E}_{y,Temp} \times \left[\frac{\vec{E}_{y,Temp} \cdot \left[\left(\frac{\vec{E}_4 + \vec{E}_6}{2} \right) - \vec{E}_2 \right]}{2} \right]$$

3. Apply the adjustment to points E2, E4, and E6 on the superior end plate.

$$\vec{E}_{2,adjust} = \vec{E}_2 + \vec{E}_{y,\text{inf}\_adjust}$$

$$\vec{E}_{4,adjust} = \vec{E}_4 - \vec{E}_{y,\text{inf}\_adjust}$$

$$\vec{E}_{6,adjust} = \vec{E}_6 - \vec{E}_{y,\text{inf}\_adjust}$$

4. Given the adjusted points E1, E3, and E5, perform an elliptical fit through those points to find the plane. To do this, the points need to be placed in a plane for the 2D elliptical calculation to take place.

$$\vec{E}_{x,\text{sup}adjust} = \frac{\vec{E}_{3,adjust} - \vec{E}_{1,adjust}}{\left| \vec{E}_{3,adjust} - \vec{E}_{1,adjust} \right|}$$

$$\vec{E}_{y\_temp,\text{sup}adjust} = \frac{\vec{E}_{5,adjust} - \vec{E}_{1,adjust}}{\left| \vec{E}_{5,adjust} - \vec{E}_{1,adjust} \right|}$$

$\vec{E}_{z,\text{sup}adjust} = \vec{E}_{x,\text{sup}adjust} \times \vec{E}_{y\_temp,\text{sup}adjust}$ Then normalize it.

$\vec{E}_{y,\text{sup}adjust} = \vec{E}_{z,\text{sup}adjust} \times \vec{E}_{x,\text{sup}adjust}$ Then normalize it.

Build matrix

$$T_{WORLD,\text{sup}adjust} = \begin{bmatrix} \vec{E}_{x,\text{sup}adjust} & \vec{E}_{y,\text{sup}adjust} & \vec{E}_{z,\text{sup}adjust} & \vec{E}_{1,adjust} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5. Find the adjusted end positions in the adjust reference plane. (in this case z is superior and should have values of 0 so we can drop it and just use x, y, for the ellipse fit.

$$\vec{E}_{3ellipse,\text{sup}adjust} = T^{-1}_{WORLD,\text{sup}adjust} \times \vec{E}_{3,adjust}$$

$$\vec{E}_{5ellipse,\text{sup}adjust} = T^{-1}_{WORLD,\text{sup}adjust} \times \vec{E}_{5,adjust}$$

Extract the x and y values from the points and put it into an array with 0, 0 for the E1 adjusted point.

6. Run ellipse fit and calculate centroid of the ellipse. Use these xy values to calculate the centroid in the world reference frame.

$$\vec{E}_{centroid,\text{sup}adjust} = T_{WORLD,\text{sup}adjust} \times \vec{E}_{\text{sup}centroid}$$

7. Given the adjusted points E2, E4, and E6, perform an elliptical fit through those points to find the plane. To do this, the points need to be placed in a plane for the 2D elliptical calculation to take place.

$$\vec{E}_{x,\text{inf}\,adjust} = \frac{\vec{E}_{4,adjust} - \vec{E}_{2,adjust}}{\left| \vec{E}_{4,adjust} - \vec{E}_{2,adjust} \right|}$$

$$\vec{E}_{y\_temp,\inf adjust} = \frac{\vec{E}_{6,adjust} - \vec{E}_{2,adjust}}{\left|\vec{E}_{6,adjust} - \vec{E}_{2,adjust}\right|}$$

$\vec{E}_{z,\inf adjust} = \vec{E}_{x,\inf adjust} \times \vec{E}_{y\_temp,\inf adjust}$ Then normalize it.

$\vec{E}_{y,\inf adjust} = \vec{E}_{z,\inf adjust} \times \vec{E}_{x,\inf adjust}$ Then normalize it.

Build matrix

$$T_{WORLD,\inf adjust} = \begin{bmatrix} \vec{E}_{x,\inf adjust} & \vec{E}_{y,\inf adjust} & \vec{E}_{z,\inf adjust} & \vec{E}_{2,adjust} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

8. Find the adjusted end positions in the adjust reference plane. (in this case z is inferior and should have values of 0 so we can drop it and just use x, y, for the ellipse fit.

$\vec{E}_{4ellipse,\inf adjust} = T^{-1}_{WORLD,\inf adjust} \times \vec{E}_{4,adjust}$

$\vec{E}_{6ellipse,\inf adjust} = T^{-1}_{WORLD,\inf adjust} \times \vec{E}_{6,adjust}$

Extract the x and y values from the points and put it into an array with 0, 0 for the E2 adjusted point (the origin).

9. Run ellipse fit and calculate centroid of the ellipse. Use these xy values to calculate the centroid in the world reference frame.

$\vec{E}_{centroid,\inf adjust} = T_{WORLD,\inf adjust} \times \vec{E}_{\inf centroid}$

10. The second temporary y-axis is pointed superiorly as a vector between these two centroids. It is defined by the normalized vector pointing from $E_{inf\ centroid}$ to $E_{sup\ centroid}$.

$$\vec{E}_{y,temp2} = \frac{\vec{E}_{\sup centroid} - \vec{E}_{\inf centroid}}{\left|\vec{E}_{\sup centroid} - \vec{E}_{\inf centroid}\right|}$$

11. The origin is the midpoint between these two centroids.

$$\vec{O}_E = \frac{\vec{E}_{\sup centroid} + \vec{E}_{\inf centroid}}{2}$$

12. The z-axis is pointed laterally to the right by a vector connecting the two transverse process points. It is defined by the normalized vector pointing from E_7 to E_8.

$$\vec{E}_z = \frac{\vec{E}_8 - \vec{E}_7}{\left|\vec{E}_8 - \vec{E}_7\right|}$$

13. The x-axis is pointed anteriorly. It is defined by the normalized vector orthogonal to E_z and E_{ytemp2}.

$\vec{E}_x = \vec{E}_{y,temp2} \times \vec{E}_z$ Then normalize it.

14. The y-axis is pointed superiorly. It is defined by the normalized vector orthogonal to E_z and E_x.

$\vec{E}_y = \vec{E}_z \times \vec{E}_x$ Then normalize it.

15. T_WORLD_VERT is defined as the rotations and translations from the world coordinate system to the vertebra coordinate system. Put the vertebra axes and origin in a 4x4 matrix.

$$T_{WORLD,VERT} = \begin{bmatrix} \vec{E}_x & \vec{E}_y & \vec{E}_z & \vec{O}_E \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Algorithm Verification:
Use the following inputs to verify the algorithm.

$$\vec{E}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \quad \vec{E}_2 = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \quad \vec{E}_3 = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} \quad \vec{E}_4 = \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix} \quad \vec{E}_5 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \quad \vec{E}_6 = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \quad \vec{E}_7 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$$

$$\vec{E}_8 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

They should yield the following transformation matrix:

$$T_{WORLD,VERT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

B.7 Make Cervical Facet JCS Transformation Matrix

Purpose:

This generates the transformation matrix between the superior (SUP) and inferior (INF) cervical or thoracic facets of a given functional spinal unit. This dynamic matrix is a function of the 6-DOF Joint Coordinate System (JCS) rotations and translations.

References:

The JCS definition is based on Panjabi's work [109] with some modifications regarding the definition of the rotations. The definition of the individual facet coordinate systems are defined below. The relative relationship between these coordinate systems establishes the facet JCS.

This function requires an input indicating if it is the right or left facet.
The coordinate system meets the following convention though the labels are wrong.

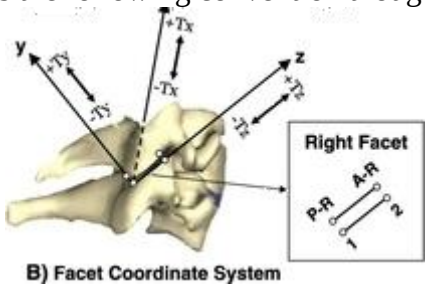


Figure B2. Facet Coordinate Systems

Facet coordinate system:

Axial Separation (y): The line normal to the facet surface, and pointing posterior/superior.

Lateral Shear (z): The line parallel to a line joining similar landmarks on the right and left facets, and pointing laterally.

Anterior Sliding (x): The line perpendicular to the Y- and Z-axis line that points in the anterior/superior direction.

Origin (o): The origin of the individual facet is a point that is the centroid of an elliptical fit to the facet surface.

Joint Coordinate System: (superior facet relative to inferior)

Spine Facet JCS contains the following components:

- *a* anterior sliding of superior facet
- *s* axial separation of superior facet (distraction is positive)
- *l* lateral shear of superior facet
- *t* tilt (tilt right is positive for a right facet)

r axial rotation (right shoulder forward is positive for a right facet)
e Extension

Origin (O): The origin of the joint is defined as the origin of the inferior facet.

Extension (*e*): The extension axis is the axis fixed to the superior facet and coincident with the Z-axis of the superior facet coordinate system. Extension is positive; Flexion is negative.

tilt (*t*): The tilt axis is the floating axis, the common axis perpendicular to the extension and axial rotation axes. For a right facet, right leaning is positive, and left leaning is negative.

Axial Rotation (*r*): The axial rotation axis is the axis fixed to the inferior facet and coincident with the y-axis of the inferior facet coordinate system. For a right facet, right shoulder forward is positive, and left shoulder forward is negative.

Order of rotation and translation:
Order of rotation and translations are important to understand and apply the kinematics of the JCS. The order of rotations and translations are given below.

1: *e* rotations and all translations are applied simulatenously
2: *t* rotations
3: *r* rotations

Algorithm:

Use the standard x, y, z, roll, pitch, yaw convention with the JCS vector defined above.

B.8 Extract Cervical Facet JCS Values From Transformation Matrix

Purpose:

After applying the kinematic chain equation of the system to determine the transformation matrix of the joint, the JCS values must be extracted so that the kinematics can be interpreted or controlled.

Algorithm:

Use the standard x, y, z, roll, pitch, yaw convention with the JCS vector.

B.9 Make Transformation Matrix WORLD_CFACET

Purpose:

Create the transformation matrix of an individual cervical or thoracic facet relative to the digitizer world reference frame. This function requires an input indicating if it is the right or left facet.

4 points will be collected.
E_1 = Most anterior/superior point on the facet
E_2 = Most lateral point on the facet
E_3 = Most posterior/inferior point on the facet
E_4 = Most lateral point on the contralateral facet

Algorithm:

1. The z-axis is pointed laterally along the two contralateral points on the facet surface and is defined by the normalized vector pointing from E_4 to E_2.

$$\vec{E}_z = \frac{\vec{E}_2 - \vec{E}_4}{\left|\vec{E}_2 - \vec{E}_4\right|}$$

4. Given the points E1, E2, and E3, perform an elliptical fit through those points to find the plane. To do this, the points need to be placed in a plane for the 2D elliptical calculation to take place.

$$\vec{E}_{x,adjust} = \frac{\vec{E}_1 - \vec{E}_2}{\left|\vec{E}_1 - \vec{E}_2\right|}$$

$$\vec{E}_{y\_temp,adjust} = \frac{\vec{E}_3 - \vec{E}_2}{\left|\vec{E}_3 - \vec{E}_2\right|}$$

$\vec{E}_{z,adjust} = \vec{E}_{x,adjust} \times \vec{E}_{y\_temp,adjust}$ Then normalize it.

$\vec{E}_{y,adjust} = \vec{E}_{z,adjust} \times \vec{E}_{x,adjust}$ Then normalize it.

Build matrix

$$T_{WORLD,adjust} = \begin{bmatrix} \vec{E}_{x,adjust} & \vec{E}_{y,adjust} & \vec{E}_{z,adjust} & \vec{E}_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5. Find the adjusted end positions in the adjust reference plane. (in this case z is out of plane and should have values of 0 so we can drop it and just use x, y, for the ellipse fit.

$\vec{E}_{1ellipse,adjust} = T^{-1}_{WORLD,adjust} \times \vec{E}_1$

$\vec{E}_{3ellipse,adjust} = T^{-1}_{WORLD,adjust} \times \vec{E}_3$

Extract the x and y values from the points and put it into an array with 0, 0 for the E2 adjusted point.

6. Run ellipse fit and calculate centroid of the ellipse. Use these xy values to calculate the centroid (origin) in the world reference frame. $\vec{E}_{centroid,adjust}$

$\vec{O}_E = T_{WORLD,adjust} \times \vec{E}_{centroid,adjust}$

10. The temporary x-axis is pointed antero-superiorly. It is defined by the normalized vector pointing from E_3 to E_1.

$$\vec{E}_{x,temp} = \frac{\vec{E}_1 - \vec{E}_3}{\left|\vec{E}_1 - \vec{E}_3\right|}$$

13. The y-axis is pointed posterior/superior . It is defined by the normalized vector orthogonal to E_z and E_{xtemp}.

$\vec{E}_y = \vec{E}_z \times \vec{E}_{x,temp}$ Then normalize it.

14. The x-axis is pointed antero-superiorly. It is defined by the normalized vector orthogonal to E_z and E_y.

$\vec{E}_x = \vec{E}_y \times \vec{E}_z$ Then normalize it.

15. T_WORLD_CFACET is defined as the rotations and translations from the world coordinate system to the cervical facet coordinate system. Put the facet axes and origin in a 4x4 matrix.

$$T_{WORLD,CFACET} = \begin{bmatrix} \vec{E}_x & \vec{E}_y & \vec{E}_z & \vec{O}_E \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Algorithm Verification:

Use the following inputs to verify the algorithm.

$$\vec{E}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \vec{E}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \vec{E}_3 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \quad \vec{E}_4 = \begin{bmatrix} 0 \\ 0 \\ -2 \end{bmatrix}$$

They should yield the following transformation matrix:

$$T_{WORLD,CFACET} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

B.10 UMS Make T_WORLD_LOAD.vi

Purpose:

This generates the transformation matrix between the world coordinate system (WORLD) and Load cell coordinate system LOAD. This matrix is constant as long as the load cell is mounted to the frame.

(1) Using position digitization data, compute the orientation of the Z axis of the load cell. This axis is perpendicular to the plane defined by the points B1-B3:

(1a) $\vec{b}_z = \frac{\left(\vec{B}_2 - \vec{B}_1\right)}{\left|\left(\vec{B}_2 - \vec{B}_1\right)\right|} \times \frac{\left(\vec{B}_3 - \vec{B}_1\right)}{\left|\left(\vec{B}_3 - \vec{B}_1\right)\right|}$. This will produce a Z axis which points away from the surface of the load cell.

2) Using position digitization data, compute the position of the Z axis of the load cell. This is the axis of the cylinder surface defined by the points B4-B6. Because the axis is vertical or nearly vertical in the world coordinate system, the axis position vector $\vec{a}$ can be effectively parameterized as:

$$\vec{a} = \begin{pmatrix} p_1 \\ p_2 \\ 0 \end{pmatrix}$$

$\vec{a}$ is found by fitting a cylinder to six points, the original points B4-B6, plus three extra points which are the same points, but shifted by adding $\vec{b}_z$ to each point. The cylinder's axis will therefore be aligned with

A good initial guess for the iterative minimization is to use the center of gravity of the triangle formed by B1, B2, and B3: p1 = (B1x+B2x+B3x)/3, p2 = (B1y+B2y+B3y)/3.

(3) Project B1 onto the cylinder axis to get the origin of the load cell:

$$\vec{O}_{LB} = \vec{a} + \left((\vec{B}_1 - \vec{a}) \cdot \vec{b}_z\right)\vec{b}_z$$

(4) Use points B7 and B8 to define the X and Y axes of the load cell.

(4a) Compute distance D between C7 and C8 along the load cell axis:

$$D = (\vec{B}_8 - \vec{a}) \cdot \vec{b}_z - (\vec{B}_7 - \vec{a}) \cdot \vec{b}_z$$

(4b) Move B7 by this amount, so B7 and B8 are at the same load cell Z coordinate:

$$\vec{B}_{7a} = \vec{B}_7 + D\vec{b}_z$$

(4c) Now the Y axis of the load cell points from B7a to B8

$$\vec{b}_y = \vec{B}_8 - \vec{B}_{7a}$$

(4d) The X axis of the load cell is obtained by cross product:

$$\vec{b}_x = \vec{b}_y \times \vec{b}_z$$

(5) Put the load cell axes and origin together in a 4x4 matrix:

$$\mathbf{T}_{WORLD,LOAD} = \begin{pmatrix} \vec{b}_x & \vec{b}_y & \vec{b}_z & \vec{O}_{LB} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

APPENDIX C. COMPENSATORY KINETIC AND KINEMATIC DIFFERENCES

Tables C1 – C3 show the mean and standard deviations of the state differences between the "no" sensitivity condition and the "medium" (50%) or "high" (100%) sensitivity conditions. These state differences are the estimated compensatory responses to reduce the force, or simulated pain, in the right facet joint. Bold p values represent statistically significant differences. Italicized values represent states that were trending toward significance but were not less than 0.05. The changes in kinetics and kinematics are the average of the compensatory responses for all specimens. Figures 55 - 89 are the graphical representations of the results in this table.

Table C1: Kinetic State Differences at Each Loading Condition

Condition	Sensitivity (Gain%)	Left Facet Force N	Right Facet Force N	JCS Load Posterior Shear Force N	JCS Load Compression N	JCS Load Left Lateral Shear Force N	JCS Load Left Lateral Bending Torque Nm	JCS Load Right Axial Rotation Torque Nm	JCS Load Flexion Torque Nm
Extension	50% mean	1.16	-1.90	0.20	-1.51	-0.12	0.06	0.03	-0.04
Extension	50% std	2.45	3.75	1.85	2.03	0.26	0.09	0.05	0.03
Extension	50% p value	0.258	0.229	0.784	0.097	0.275	0.139	0.184	**0.006**
Extension	100% mean	1.38	-2.90	1.05	-1.22	0.02	0.09	0.05	-0.04
Extension	100% std	2.94	5.05	2.61	1.77	0.29	0.13	0.08	0.05
Extension	100% p value	0.260	0.179	0.327	0.119	0.857	0.120	0.105	**0.054**
Extension - Left Axial Rotation	50% mean	0.62	-18.21	8.57	-0.10	0.12	0.33	0.26	-0.13
Extension - Left Axial Rotation	50% std	0.88	21.07	7.12	0.36	0.46	0.22	0.24	0.16
Extension - Left Axial Rotation	50% p value	0.111	*0.062*	**0.019**	0.489	0.533	**0.008**	**0.029**	0.072
Extension - Left Axial Rotation	100% mean	0.65	-23.81	12.23	0.51	0.61	0.47	0.37	-0.18
Extension - Left Axial Rotation	100% std	0.95	27.04	9.68	0.94	1.00	0.31	0.32	0.22
Extension - Left Axial Rotation	100% p value	0.121	*0.059*	**0.016**	0.197	0.157	**0.008**	**0.022**	0.076
Flexion	50% mean	0.43	-1.51	1.48	-0.41	0.28	0.02	0.06	-0.01
Flexion	50% std	0.95	2.70	1.94	0.83	0.20	0.03	0.07	0.04
Flexion	50% p value	0.278	0.190	0.090	0.240	**0.010**	0.093	0.072	0.506
Flexion	100% mean	0.52	-2.24	2.17	-0.78	0.10	0.06	0.07	-0.03
Flexion	100% std	1.14	4.21	2.65	1.00	0.29	0.05	0.09	0.06
Flexion	100% p value	0.274	0.210	0.073	0.085	0.385	**0.016**	0.073	0.282
Left Axial Rotation	50% mean	-0.06	-9.18	5.96	-0.95	-0.51	0.23	0.19	-0.15
Left Axial Rotation	50% std	0.09	14.32	6.31	1.29	0.75	0.14	0.21	0.14
Left Axial Rotation	50% p value	0.151	0.141	**0.047**	0.097	0.121	**0.004**	**0.048**	**0.031**
Left Axial Rotation	100% mean	-0.07	-12.17	8.49	-0.42	-0.23	0.31	0.29	-0.21
Left Axial Rotation	100% std	0.13	17.94	7.91	2.25	0.69	0.15	0.27	0.19
Left Axial Rotation	100% p value	0.191	0.123	**0.030**	0.636	0.413	**0.001**	**0.030**	**0.027**

Left Lateral Bending	50% mean	-0.59	-0.02	-0.10	-2.75	-0.46	0.03	0.00	-0.01
Left Lateral Bending	50% std	1.08	0.10	0.80	3.07	0.50	0.02	0.02	0.02
Left Lateral Bending	50% p value	0.199	0.561	0.764	*0.056*	*0.051*	**0.011**	0.716	0.094
Left Lateral Bending	100% mean	-0.34	-0.02	0.34	-2.17	-0.38	0.03	0.01	-0.02
Left Lateral Bending	100% std	0.99	0.07	1.24	2.76	0.33	0.02	0.02	0.02
Left Lateral Bending	100% p value	0.404	0.522	0.498	0.082	**0.023**	**0.009**	0.449	0.139
Right Axial Rotation	50% mean	-0.56	-0.06	0.23	0.71	-0.17	0.01	0.00	0.00
Right Axial Rotation	50% std	1.98	0.18	0.25	0.54	0.14	0.01	0.01	0.01
Right Axial Rotation	50% p value	0.482	0.426	**0.050**	**0.013**	**0.020**	**0.028**	0.292	0.711
Right Axial Rotation	100% mean	0.00	0.00	0.31	0.73	-0.11	0.01	0.01	0.00
Right Axial Rotation	100% std	1.78	0.16	0.48	0.48	0.20	0.02	0.01	0.01
Right Axial Rotation	100% p value	1.000	0.938	0.143	**0.007**	0.171	*0.051*	0.181	0.943
Right Lateral Bending	50% mean	0.13	-4.88	2.10	-6.16	1.02	0.09	0.12	-0.04
Right Lateral Bending	50% std	0.22	7.72	2.72	5.27	0.90	0.09	0.08	0.05
Right Lateral Bending	50% p value	0.165	0.146	0.088	**0.021**	**0.024**	**0.039**	**0.006**	0.096
Right Lateral Bending	100% mean	0.18	-6.51	3.30	-6.12	0.94	0.16	0.16	-0.04
Right Lateral Bending	100% std	0.19	9.62	3.43	5.58	0.87	0.16	0.10	0.07
Right Lateral Bending	100% p value	**0.047**	0.124	**0.044**	**0.027**	**0.028**	**0.033**	**0.008**	0.129
Right Lateral Bending – Left Axial Rotation	50% mean	0.05	-18.13	9.99	0.78	-0.21	0.50	0.33	-0.15
Right Lateral Bending – Left Axial Rotation	50% std	0.08	22.44	7.87	1.80	0.49	0.21	0.30	0.16
Right Lateral Bending – Left Axial Rotation	50% p value	0.114	*0.076*	**0.015**	0.296	0.296	**0.001**	**0.027**	*0.055*
Right Lateral Bending – Left Axial Rotation	100% mean	0.00	-25.63	14.63	-0.44	-0.18	0.73	0.50	-0.22
Right Lateral Bending – Left Axial Rotation	100% std	0.10	26.98	9.89	3.96	0.82	0.27	0.37	0.22
Right Lateral Bending – Left Axial Rotation	100% p value	0.942	**0.046**	**0.008**	0.777	0.587	**0.000**	**0.011**	**0.039**

Table C2: Kinematic Vertebral State Differences at Each Loading Condition

Condition	Sensitivity (Gain%)	JCS Anterior mm	JCS Superior mm	JCS Lateral mm	JCS Lateral Bending deg	JCS Axial Rotation deg	JCS Extension deg	C4-C5 JCS Anterior mm	C4-C5 JCS Superior mm	C4-C5 JCS Lateral mm	C4-C5 JCS Lateral Bending deg	C4-C5 JCS Axial Rotation deg	C4-C5 JCS Extension deg
Extension	50% mean	0.38	0.23	0.33	-0.54	0.03	1.51	0.00	0.00	-0.01	-0.02	-0.01	0.21
Extension	50% std	0.63	0.13	0.46	0.83	0.79	0.95	0.10	0.04	0.07	0.15	0.18	0.11
Extension	50% p value	0.161	**0.003**	0.105	0.134	0.914	**0.005**	0.977	0.835	0.615	0.767	0.840	**0.002**
Extension	100% mean	0.15	0.21	0.33	-0.68	0.11	1.01	-0.02	0.01	-0.02	-0.05	-0.02	0.14
Extension	100% std	0.51	0.10	0.54	1.04	0.92	0.67	0.07	0.02	0.07	0.19	0.20	0.07
Extension	100% p value	0.461	**0.002**	0.154	0.136	0.755	**0.007**	0.403	0.572	0.450	0.471	0.779	**0.001**
Extension - Left Axial Rotation	50% mean	-0.46	0.45	0.98	-2.00	1.11	0.75	0.01	-0.02	0.05	-0.24	0.12	0.04
Extension - Left Axial Rotation	50% std	0.63	0.65	1.05	1.75	1.07	0.73	0.08	0.03	0.13	0.32	0.53	0.22
Extension - Left Axial Rotation	50% p value	0.104	0.116	**0.049**	**0.024**	**0.034**	**0.035**	0.794	0.182	0.366	0.093	0.565	0.674
Extension - Left Axial Rotation	100% mean	-0.82	0.53	1.06	-2.41	0.94	0.46	-0.01	-0.01	0.06	-0.27	0.12	-0.01
Extension - Left Axial Rotation	100% std	1.02	0.78	1.00	2.02	1.36	0.69	0.09	0.03	0.19	0.44	0.73	0.36
Extension - Left Axial Rotation	100% p value	0.078	0.124	**0.031**	**0.020**	0.116	0.126	0.835	0.350	0.421	0.155	0.689	0.954
Flexion	50% mean	-0.36	0.00	-0.44	0.51	-1.15	-0.81	0.01	-0.01	-0.03	0.11	-0.27	-0.14
Flexion	50% std	0.42	0.09	0.38	0.47	0.82	0.35	0.05	0.01	0.06	0.14	0.16	0.10
Flexion	50% p value	0.066	0.957	**0.022**	0.029	**0.010**	**0.001**	0.679	**0.031**	0.230	0.087	**0.005**	**0.009**
Flexion	100% mean	-0.45	-0.02	-0.38	0.28	-1.13	-0.56	0.00	-0.01	-0.01	0.07	-0.24	-0.13
Flexion	100% std	0.51	0.08	0.41	0.50	0.89	0.31	0.06	0.01	0.07	0.17	0.19	0.07
Flexion	100% p value	*0.056*	0.564	**0.048**	0.185	**0.015**	**0.003**	0.839	**0.009**	0.671	0.288	**0.016**	**0.003**

Left Axial Rotation	50% mean	-1.59	0.43	1.16	-2.15	0.37	0.93	-0.01	0.02	0.05	-0.20	0.13	-0.19
Left Axial Rotation	50% std	2.60	0.57	1.22	1.90	1.15	0.57	0.11	0.05	0.07	0.20	0.17	0.46
Left Axial Rotation	50% p value	0.156	0.091	**0.045**	**0.024**	0.426	**0.005**	0.805	0.326	0.116	**0.037**	0.080	0.315
Left Axial Rotation	100% mean	-2.17	0.47	1.08	-2.14	-0.14	0.95	0.00	0.01	0.02	-0.19	0.04	-0.10
Left Axial Rotation	100% std	2.78	0.64	1.07	1.79	1.41	0.66	0.11	0.05	0.08	0.22	0.22	0.42
Left Axial Rotation	100% p value	0.085	0.099	**0.037**	**0.020**	0.797	**0.009**	0.972	0.515	0.471	0.062	0.618	0.540
Left Lateral Bending	50% mean	0.45	0.58	0.74	-1.32	0.40	0.36	-0.01	-0.01	0.03	-0.15	0.01	0.04
Left Lateral Bending	50% std	0.34	0.93	1.33	1.20	0.37	0.63	0.12	0.03	0.05	0.09	0.11	0.09
Left Lateral Bending	50% p value	**0.012**	0.149	0.190	**0.027**	**0.030**	0.181	0.850	0.637	0.193	**0.005**	0.784	0.295
Left Lateral Bending	100% mean	0.22	0.43	0.55	-0.98	0.20	0.21	0.00	0.01	0.01	-0.11	-0.02	0.01
Left Lateral Bending	100% std	0.38	0.62	0.91	0.88	0.35	0.43	0.11	0.02	0.03	0.06	0.10	0.07
Left Lateral Bending	100% p value	0.173	0.111	0.163	**0.027**	0.183	0.249	0.932	0.391	0.393	**0.003**	0.657	0.774
Right Axial Rotation	50% mean	-0.12	-0.04	0.14	-0.43	-0.38	-0.07	-0.01	0.00	0.01	-0.04	-0.07	-0.01
Right Axial Rotation	50% std	0.10	0.05	0.22	0.30	0.19	0.26	0.03	0.02	0.02	0.03	0.05	0.04
Right Axial Rotation	50% p value	**0.014**	0.079	0.151	**0.009**	**0.002**	0.484	0.315	0.622	0.220	**0.013**	**0.008**	0.413
Right Axial Rotation	100% mean	-0.13	-0.03	0.13	-0.37	-0.23	-0.08	-0.01	0.00	0.01	-0.04	-0.06	-0.02
Right Axial Rotation	100% std	0.12	0.06	0.18	0.26	0.17	0.21	0.03	0.02	0.02	0.03	0.04	0.03
Right Axial Rotation	100% p value	**0.025**	0.220	0.114	**0.009**	**0.014**	0.376	0.415	0.980	0.268	**0.010**	**0.008**	**0.049**

Right Lateral Bending	50% mean	0.09	0.45	-0.89	1.05	-1.83	1.07	0.04	0.01	-0.08	0.16	-0.40	0.01
Right Lateral Bending	50% std	0.81	0.58	1.04	0.90	0.76	0.89	0.10	0.03	0.07	0.13	0.17	0.25
Right Lateral Bending	50% p value	0.782	0.084	0.065	**0.022**	**0.001**	**0.019**	0.320	0.414	**0.019**	**0.016**	**0.001**	0.895
Right Lateral Bending	100% mean	-0.14	0.34	-0.71	0.55	-1.61	1.11	0.07	-0.01	-0.05	0.10	-0.33	0.07
Right Lateral Bending	100% std	0.77	0.41	0.81	0.77	0.66	0.67	0.07	0.03	0.05	0.12	0.13	0.16
Right Lateral Bending	100% p value	0.649	0.067	0.058	0.110	**0.001**	**0.005**	**0.038**	0.389	**0.041**	0.066	**0.001**	0.285
Right Lateral Bending – Left Axial Rotation	50% mean	-2.88	-0.19	0.06	-2.37	0.60	1.71	-0.13	0.00	-0.08	0.08	-0.22	0.23
Right Lateral Bending – Left Axial Rotation	50% std	2.49	0.89	0.84	2.91	3.21	1.72	0.17	0.03	0.35	0.94	1.48	0.53
Right Lateral Bending – Left Axial Rotation	50% p value	**0.022**	0.598	0.863	0.075	0.637	**0.039**	0.098	0.994	0.587	0.824	0.707	0.304
Right Lateral Bending – Left Axial Rotation	100% mean	-3.49	-0.13	0.03	-3.41	0.73	2.46	-0.14	0.00	-0.06	-0.01	-0.23	0.32
Right Lateral Bending – Left Axial Rotation	100% std	2.72	0.94	0.85	3.37	3.56	1.93	0.19	0.04	0.38	1.04	1.64	0.65
Right Lateral Bending – Left Axial Rotation	100% p value	**0.015**	0.719	0.924	**0.036**	0.606	**0.015**	0.096	0.993	0.697	0.990	0.721	0.245

Table C3: Kinematic Facet State Differences at Each Loading Condition

Condition	Sensitivity (Gain%)	C4-C5 Left Facet JCS Anterior Sliding mm	C4-C5 Left Facet JCS Axial Separation mm	C4-C5 Left Facet JCS Lateral Shear mm	C4-C5 Left Facet JCS Tilt deg	C4-C5 Left Facet JCS Axial Rotation deg	C4-C5 Left Facet JCS Extension deg	C4-C5 Right Facet JCS Anterior Sliding mm	C4-C5 Right Facet JCS Axial Separation mm	C4-C5 Right Facet JCS Lateral Shear mm	C4-C5 Right Facet JCS Tilt deg	C4-C5 Right Facet JCS Axial Rotation deg	C4-C5 Right Facet JCS Extension deg
Extension	50% mean	-0.06	-0.02	0.02	0.07	0.02	0.22	-0.07	0.03	-0.02	-0.02	0.00	0.22
Extension	50% std	0.12	0.06	0.06	0.10	0.20	0.11	0.06	0.10	0.06	0.05	0.23	0.11
Extension	50% p value	0.267	0.421	0.399	0.136	0.760	**0.002**	**0.023**	0.449	0.458	0.243	0.960	**0.002**
Extension	100% mean	-0.05	-0.02	0.04	0.11	0.01	0.15	-0.06	0.06	-0.03	-0.05	0.02	0.15
Extension	100% std	0.12	0.04	0.05	0.15	0.21	0.06	0.06	0.09	0.06	0.03	0.28	0.07
Extension	100% p value	0.261	0.320	0.119	0.101	0.876	**0.001**	**0.022**	0.120	0.153	**0.002**	0.890	**0.001**
Extension - Left Axial Rotation	50% mean	-0.09	-0.09	0.00	0.20	-0.22	0.04	0.09	0.10	0.02	-0.06	0.24	0.03
Extension - Left Axial Rotation	50% std	0.10	0.10	0.13	0.28	0.53	0.22	0.27	0.16	0.14	0.20	0.59	0.22
Extension - Left Axial Rotation	50% p value	**0.043**	**0.050**	0.955	0.105	0.321	0.609	0.408	0.156	0.779	0.432	0.321	0.701
Extension - Left Axial Rotation	100% mean	-0.09	-0.10	0.00	0.26	-0.22	0.00	0.10	0.15	0.02	-0.07	0.26	-0.01
Extension - Left Axial Rotation	100% std	0.13	0.13	0.18	0.39	0.74	0.35	0.37	0.20	0.19	0.26	0.83	0.35
Extension - Left Axial Rotation	100% p value	0.110	0.088	0.994	0.129	0.466	0.990	0.502	0.101	0.781	0.511	0.443	0.935
Flexion	50% mean	0.13	-0.07	0.05	0.06	0.26	-0.14	-0.06	0.04	-0.05	-0.07	-0.27	-0.14
Flexion	50% std	0.06	0.04	0.04	0.15	0.18	0.10	0.09	0.07	0.05	0.11	0.19	0.09
Flexion	50% p value	**0.002**	**0.008**	**0.028**	0.346	**0.009**	**0.012**	0.131	0.172	**0.034**	0.143	**0.009**	**0.007**
Flexion	100% mean	0.11	-0.06	0.03	0.06	0.22	-0.12	-0.04	0.05	-0.03	-0.08	-0.24	-0.13
Flexion	100% std	0.09	0.04	0.05	0.14	0.23	0.07	0.09	0.09	0.05	0.11	0.23	0.06

Flexion	100% p value	**0.016**	**0.007**	0.167	0.257	**0.044**	**0.003**	0.303	0.173	0.165	0.095	**0.034**	**0.001**
Left Axial Rotation	50% mean	-0.01	0.00	-0.03	0.12	-0.21	-0.20	0.16	0.05	0.01	-0.06	0.23	-0.20
Left Axial Rotation	50% std	0.11	0.09	0.08	0.20	0.17	0.47	0.18	0.10	0.08	0.13	0.24	0.46
Left Axial Rotation	50% p value	0.785	0.951	0.408	0.148	**0.017**	0.314	*0.053*	0.256	0.752	0.265	0.044	0.301
Left Axial Rotation	100% mean	-0.01	-0.03	0.01	0.17	-0.13	-0.11	0.11	0.08	-0.02	-0.11	0.15	-0.11
Left Axial Rotation	100% std	0.10	0.09	0.09	0.20	0.22	0.44	0.19	0.09	0.09	0.13	0.29	0.43
Left Axial Rotation	100% p value	0.769	0.362	0.748	0.060	0.158	0.528	0.175	0.073	0.566	0.083	0.213	0.515
Left Lateral Bending	50% mean	-0.07	-0.06	0.00	0.12	-0.11	0.03	0.00	0.03	-0.01	-0.10	0.10	0.03
Left Lateral Bending	50% std	0.06	0.13	0.03	0.10	0.08	0.10	0.10	0.06	0.03	0.13	0.08	0.09
Left Lateral Bending	50% p value	**0.019**	0.271	0.903	**0.019**	**0.010**	0.420	0.904	0.279	0.246	0.069	**0.014**	0.428
Left Lateral Bending	100% mean	-0.04	-0.05	0.01	0.11	-0.06	0.00	0.02	0.04	-0.02	-0.10	0.06	0.00
Left Lateral Bending	100% std	0.07	0.10	0.03	0.09	0.04	0.07	0.06	0.07	0.03	0.10	0.06	0.07
Left Lateral Bending	100% p value	0.188	0.292	0.243	**0.015**	**0.013**	0.915	0.492	0.178	0.093	**0.038**	**0.033**	0.963
Right Axial Rotation	50% mean	0.00	-0.03	0.01	0.08	0.04	-0.01	-0.03	0.04	-0.01	-0.08	-0.03	-0.01
Right Axial Rotation	50% std	0.03	0.03	0.02	0.03	0.03	0.04	0.04	0.04	0.03	0.03	0.04	0.04
Right Axial Rotation	50% p value	0.975	0.065	0.437	**0.000**	**0.014**	0.747	0.130	**0.040**	0.267	**0.000**	0.068	0.666
Right Axial Rotation	100% mean	0.01	-0.02	0.01	0.07	0.03	-0.02	-0.01	0.04	-0.01	-0.07	-0.02	-0.02
Right Axial Rotation	100% std	0.03	0.03	0.02	0.03	0.03	0.03	0.03	0.04	0.02	0.03	0.03	0.02
Right Axial	100% p value	0.613	0.199	0.410	**0.000**	**0.043**	0.149	0.291	**0.037**	0.231	**0.000**	0.061	0.101

Rotation													
Right Lateral Bending	50% mean	0.17	-0.07	0.09	0.07	0.43	-0.02	-0.14	0.05	-0.12	-0.09	-0.42	-0.02
Right Lateral Bending	50% std	0.09	0.09	0.06	0.11	0.17	0.24	0.11	0.06	0.08	0.08	0.20	0.23
Right Lateral Bending	50% p value	**0.002**	0.102	**0.008**	0.148	**0.001**	0.810	**0.016**	0.082	**0.007**	**0.037**	**0.001**	0.816
Right Lateral Bending	100% mean	0.12	-0.10	0.07	0.09	0.33	0.05	-0.10	0.02	-0.09	-0.10	-0.32	0.05
Right Lateral Bending	100% std	0.07	0.09	0.05	0.11	0.13	0.14	0.08	0.06	0.07	0.09	0.16	0.14
Right Lateral Bending	100% p value	**0.003**	**0.031**	**0.009**	0.084	**0.001**	0.412	**0.019**	0.402	**0.011**	**0.028**	**0.002**	0.397
Right Lateral Bending – Left Axial Rotation	50% mean	-0.07	0.03	0.07	0.22	0.27	0.28	-0.22	0.28	-0.06	-0.04	-0.25	0.25
Right Lateral Bending – Left Axial Rotation	50% std	0.69	0.08	0.27	0.48	1.66	0.53	0.56	0.26	0.32	0.38	1.71	0.53
Right Lateral Bending – Left Axial Rotation	50% p value	0.789	0.440	0.505	0.281	0.686	0.205	0.334	**0.030**	0.613	0.768	0.709	0.254
Right Lateral Bending – Left Axial Rotation	100% mean	-0.13	0.01	0.08	0.33	0.23	0.38	-0.24	0.35	-0.07	-0.10	-0.20	0.35
Right Lateral Bending – Left Axial Rotation	100% std	0.66	0.10	0.29	0.59	1.81	0.61	0.66	0.31	0.35	0.46	1.89	0.63
Right Lateral Bending – Left Axial Rotation	100% p value	0.613	0.875	0.478	0.192	0.746	0.150	0.384	**0.025**	0.636	0.595	0.786	0.200