LINEAR SYSTEM ANALYSES OF THE ROLE OF REFLEX GAIN AND DELAY IN A DYNAMIC HUMAN SPINE MODEL

TIMOTHY C FRANKLIN

Thesis submitted to

the faculty of the Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

In

Engineering Mechanics

Kevin P. Granata, Ph.D., Chair
Scott L. Hendricks, Ph.D.
Michael L. Madigan, Ph.D.

June 8, 2006
Blacksburg, Virginia

Keywords: Reflex, gain, delay, spine, stability

LINEAR SYSTEM ANALYSES OF THE ROLE OF REFLEX GAIN AND DELAY IN A DYNAMIC HUMAN SPINE MODEL

TIMOTHY C FRANKLIN

ABSTRACT

Measurement studies have linked paraspinal muscle reflexes to low back pain. However, the role of reflexes in stabilizing the spine is not clear. Previous studies enlisted biomechanical models to aid in understanding of how intrinsic stiffness stabilizes the spine. This work expands these previous studies by modeling the neuromuscular dynamic control of the spine.

The presence of delay in the reflexive system limits the availability of traditional stability analyses. However it is possible to investigate how reflex delay affects stability of the spine model using methods in linear time delayed stability. Such analyses find the maximum reflex delay, i.e., the delay margin for which stability is possible. Therefore a biomechanical model of the spine was developed that used these methods for stability. The model was able to demonstrate how reflex gains and delays affect stability.

It was shown that increased proportional reflex gain reduced the amount of co-contraction required for stability. However, increased reflex gain required a reduced delay margin of the system. Differential reflex gain had no effect on the amount of co-contraction required for stability. However, it was shown to increase the delay margin for small gains. As the differential reflex gain approached the magnitude of intrinsic muscle damping the trend was reversed, and increased gain caused the delay margin to approach zero. Increased intrinsic muscle damping did not affect the minimum co-contraction required for stability, but was shown to increase the delay margin in all cases.

This study provided a theoretical explanation for the role of reflexes in stabilizing the spine. Results agree with the trends in the published literature regarding patients with low-back pain. Specifically, these patients demonstrate abnormally larger reflex delay. To maintain stability, atypically small reflex gain is necessary. Compensatory co-contraction is required to offset the small reflex gain. Co-contraction and instability is observed in low back pain patients. The results presented here agree with measurement studies, and should aid in the development of hypotheses for future measurement studies.

Acknowledgements

I would like to thank my friends, all of whom have helped me along the way.

Dr. Granata
Without his vision and support none of this work would have been possible.

My family
Who always told me to do my best, and to believe in myself

Sammy, my wife
Who inspired me academically and supported me in so many ways

Table of Contents

ABSTRACT ii
ACKNOWLEDGEMENT iii
TABLE OF CONTENTS iv
LIST OF FIGURES vi
LIST OF TABLES vii
ATTRIBUTION viii

CHAPTER 1 – SIGNIFICANCE **1**
- A. Low back pain **1**
- B. Spinal reflexes are not understood **1**
- C. Specific Aims & Hypotheses **1**

CHAPTER 2 – INTRODUCTION **2**
- A. Goal of research **2**
- B. Previous Models **2**
- C. Reflexes in the spine **6**
- D. Implementation of a model of spine reflexes **8**

CHAPTER 3 – METHODS **9**
- A. The model **9**
  - i. Rigid bodies, rotations 9
  - ii. Intervertebral discs 10
  - iii. Muscle model 12
  - iv. Anatomy 14
- B. Dynamics **15**
  - i. Lagrangian Dynamics 15
  - ii. Generalized Forces 16
  - iii. Linearization 17
  - iv. Linear Time Delay Methods 19
- C. Application **22**
  - i. Optimization 22
  - ii. Experiments 23
  - iii. Computational Techniques 24
- D. 1 DOF Example **26**
  - i. Lagrangian Dynamics 26
  - ii. Generalized Forces 27
  - iii. Linearization 29
  - iv. Linear Time Delayed Stability 31
  - v. Simulation Results 32

CHAPTER 4 – LINEAR TIME DELAY METHODS AND STABILITY OF THE REFLEXIVE HUMAN **35**
- A. Introduction **37**
- B. Methods **39**
- C. Results **46**
- D. Discussion **49**
- E. Acknowledgements **52**

F. Reference List **52**

CHAPTER 5 – ROLE OF REFLEX GAIN AND DELAY ON SPINAL STABILITY - FORWARD DYNAMIC SIMULATION OF SPINE MECHANICS **54**
A. Introduction **56**
B. Methods **58**
C. Results **62**
D. Discussion **63**
E. Reference List **66**

CHAPTER 6 – CONCLUSIONS AND FUTURE WORK **68**
A. Conclusions **68**
i. Reflexes in the spine 68
ii. Methods employed 69
B. Next steps **69**
i. Kinematic optimization 69
ii. Coupled dynamic systems 70
iii. Sensitivity 71
iv. Optimal control theory 71
C. Improvements to the model **72**
i. Six DOF per vertebrae 72
ii. Dynamic pelvis, thighs 73
iii. Articulating thorax 73
iv. Arms 74

REFERENCES **75**

APPENDIX **78**
A. How to build this model **78**
i. Dynamics 78
ii. Numerical Methods 81
iii. Optimization, Integration 85
iv. Model Anatomy 89
v. Delay Margin Function 94
B. How to use this model **98**
i. Sample Procedure 98
ii. Main File 99
iii. Function Prototypes 102
C. Digital Appendix
i. This document
ii. Model development code
iii. Spine model in Matlab code
iv. 1DOF Example

For digital appendix, please contact the Virginia Tech Musculoskeletal Biomechanics Laboratory (www.biomechanics.esm.vt.edu)

List of Figures

Figure 3.A.1 – Rigid body diagram 9

Figure 3.A.2 – Rigid body rotations 9

Figure 3.A.3 – Planar and twist rotations in intervertebral disc model 11

Figure 3.A.4 – Hill type muscle model 12

Figure 3.A.5 – Muscle anatomy drawn with an external force F 13

Figure 3.A.6 – Illustration of a muscle passing through via points 14

Figure 3.B.1 – Nodal force from a muscle with tension T 17

Figure 3.D.1 – 1 DOF system illustration 26

Figure 3.D.2 –Locii plot of eigenvalue magnitude as it changes with frequency 32

Figure 3.D.3 – Unscaled angular position and velocity in radians plotted against time 33

Figure 3.D.4 – D_{ss} plotted vs. time 33

Figure 4.1 – Model Diagram 39

Figure 4.2 – Sample frequency sweep experiment. 44

Figure 4.3 – Normalized state space vs. time following perturbation 46

Figure 4.4 – Normalized state space distance from equilibrium Dss vs. time 46

Figure 4.5 – Co-contraction vs. proportional reflex gain for both load conditions. 47

Figure 4.6 – Metabolic power vs. proportional reflex gain for both load conditions. 47

Figure 4.7 – Delay margin vs. proportional reflex gain for multiple damping coefficients 48

Figure 4.8 – Delay margin vs. differential reflex gain for multiple damping coefficients 48

Figure 4.9 – State space slopes vs. % of delay margin simulated for six random systems. 49

Figure 5.1 – Spine model illustration 58

Figure 5.2 – Metabolic power vs. reflex gain for two loading conditions 62

Figure 5.3 – Delay margin vs. reflex gain for multiple damping coefficients 62

List of Tables

Table 3.D.1 – 1 DOF state space contraction slope vs. delay margin 33

Table 5.1 – Co-contraction percentages vs. reflex gain for two loading conditions 62

Table Ap.A.1 – Coordinate frame codes 84

Table Ap.A.2 – Muscle anatomy sheet 1: muscles in terms of coordinate descriptions 89

Table Ap.A.3 – Muscle anatomy sheet 2: coordinate description translation 91

Attribution

More than one author developed the work described in Chapters 4 and 5. Dr. Kevin Granata provided focus and direction for the work, as well as greatly contributing to the physiological interpretation of the results from the model. Dr. Scott Hendricks provided insight into the interpretation of the linear systems analyses, which led to nonlinear simulations as verification to assumptions made during the model development. Both authors were irreplaceable in this respect, and I formally thank them for their patient generosity.

CHAPTER 1
SIGNIFICANCE

A. Low back pain

Low back pain is a disorder that plagues society, with some estimates placing the annual cost in the United States alone at close to 90 billion dollars. Despite being an active area of research and concern, very little is understood about the mechanisms of low back pain. However, researchers have recently seen a connection between abnormally large reflex delay and patients with low back pain. Therefore it is possible that reflexes play a role in low back pain and may even affect its development. Investigating how reflexes affect the spine is therefore a promising direction in the search for control of low back pain.

B. Spinal reflexes are not understood

While the principal mechanisms of reflex responses have been thoroughly examined via animal studies, the role of these reflexes in the spine is not well understood. It remains unclear what systemic role the combined effects of reflex responses from paraspinal muscles have on the human spine.

The chief theory guiding this work is that normal reflexes may prevent the development of low back pain. Supplemental theories suggest that reflexes allow a subject to decrease the compressive load on intervertebral discs, or that they allow a subject to reject perturbations to stabilize the spine. This work investigated the role of reflexes using a biomechanical model of the spine. The model aided investigation of hypotheses related to the effects of reflexes on the stability of the spine.

C. Specific Aims and Hypotheses

Specific Aim 1

Develop a mathematical model of the spine that includes muscle reflexes.

Hypothesis 1

a. Adding reflexes to the spine allows stability with decreased metabolic power
b. This decreased metabolic power will correspond to reduced co-contraction

Specific Aim 2

Investigate how reflex delay effects stability of the spine

Hypothesis 2

a. Large reflex delays will force the system to rely on co-contraction for stability
b. Small reflex delays will allow the spine to rely solely on reflex for stability.

CHAPTER 2
INTRODUCTION

A. Goal of this work

The goal of this work was to investigate the role of reflexes in the spine through the development and manipulation of a mathematical model. This work drew heavily on previous spine research that focused on the role of intrinsic stiffness in the spine. These previous studies incorporated both experimental measurements and theoretical models. They were able to demonstrate that intrinsic stiffness alone is capable of stabilizing the spine. However these studies neglected to investigate the role of reflexes in the spine. Recent experimental work has shown connections between abnormally large reflex delays and low back pain (Radebold et al 2001), but no modeling study has theorized how these reflex delays affect the stability of the spine. This thesis attempts to fill that gap by using a biomechanical model to investigate the effects of reflexes in the spine.

B. Previous Models

There has been much previous work in the field of mathematical models of spinal stability. This section will review some of the milestones along the way and highlight the advances each model brought to the field. Taking advantage of these advances in previous work simplified the effort required to develop a new computational model of the spine.

i. Bergmark (1989)

Bergmark built a three dimensional (3D) model of the spine that was composed of 6 rigid bodies. These bodies represented the 5 lumbar vertebrae and a rigid thorax, each of which had 3 rotational degrees of freedom (DOF) for 18DOF total. The model used a muscle anatomy in which muscles were approximated as straight lines from origin to insertion. The model was stabilized with intrinsic muscle stiffness K, which was scaled by a non-dimensional stiffness gain q (equation 2.B.1).

$$K = q\frac{F}{L} \qquad (2.B.1)$$

Therefore, intrinsic stiffness was proportional to muscle force F and inversely proportional to muscle length L. All muscles in the model had the same value for the stiffness gain q. The model solved for the required joint torque stiffnesses that would stabilize the spine against gravity and external load. These stiffness calculations were used to find the minimum value of stiffness gain q that would stabilize the system. The model was able to achieve stability even though it only considered intrinsic stiffness (the model did not include reflexes). It was found that the minimum value of q required for stability was a function of muscle force, ie. co-contraction decreased the minimal required value of q. This minimum value of q was found to be greater than 40 for almost all trials.

ii. Crisco and Panjabi (1991)

Crisco and Panjibi built a 2 dimensional (2D) model of the spine composed of 5 single DOF rigid bodies representing the lumbar vertebrae. The model aimed to determine how muscles stabilize the spine in the lateral direction. Hence, it was a frontal plane projection of the lumbar spine. The authors used a linear buckling analysis in their attempts to establish stability in their model. Their analysis required the Hessian matrix (the square matrix of section partial derivatives with respect to the system states) of the total potential energy to be positive definite. This is equivalent to requiring the potential energy of the system to be at a minimum in the equilibrium position. Five different muscle anatomy models were individually investigated, which collectively contained all possible single and two joint muscles, in addition to 1,2,3,4 and 5 joint muscles originating at the pelvis.

The authors found the minimum muscle stiffness each set of muscles required to stabilize the loaded spine (laterally) in the absence of the other four sets. It was assumed that each individual muscle had the same stiffness as all other muscles in the same set. The resulting minimum values were used to compare with the muscle stiffness equation (2.B.1) used by Bergmark (1989). In addition to comparing their theoretical results, the authors performed a literature review of measurement studies to estimate physiological values of q (from Bergmark). From these studies they computed q to be between 0.2 and 40, with a mean value of 10.

iii. Gardner-Morse, Stokes and Laible (1995)

This model expanded the work of Bergmark (1989) to include 6 DOF per joint. The increased DOF allowed the vertebrae to translate as well as rotate relative to neighboring bodies.

The authors developed a model of the lumbar spine that used 6 (DOF) joints which represented the intervertebral discs from the sacrum/pelvis to the T12 on the thorax (for a total of 36 degrees of freedom). The model included 132 muscles which were assumed to have straight paths from origin to insertion. A linear optimization method was used to solve for muscle activations to maximize anterior exertion of the spine at the T12 level while ensuring stability. Stability was determined by performing a buckling analysis between the stiffness matrix and the gravity matrix of the system in equilibrium. In the muscle force model stiffness was assumed to be proportional to activation and modulated by the same stiffness gain q from Bergmark (1989). Only intrinsic stiffness and force were considered in the muscle model, neglecting the role of reflex.

This work found that the intrinsic stiffness of the spine may be sufficient for stability. Further, the study found that only a relatively small stiffness gain ($q = 4.5$) was necessary to provide stability. Based on the range that Crisco and Panjabi (1991) computed from their literature review, the required value of 4.5 is comparatively small. This led the authors to suggest that there is no need for reflexes to stabilize the spine. From these results, it appears that muscle reflexes may not be as critical as our hypotheses suggested.

iv. Cholewicki and McGill (1996)

Though this work was similar to Bergmark's study (1989), it included some major differences. First, the authors used a more complicated muscle anatomy. Additionally, this work incorporated measured electromyography (EMG) data and postural kinematics in an attempt to mimic measured patient data. EMG data assisted the optimization algorithm in finding values of muscle activation that were similar to those measured in the subjects. Similarly measured postural kinematics were used to orient the model to a posture that matched that of the subjects.

In this study, the investigators developed a rigid body model of the spine. The system was treated as a stack of inverted pendulums, where each lumbar vertebrae and a rigid thorax were assigned three rotational DOF (intervertebral shearing and axial compression were ignored). The total model was an 18DOF 3D model of the spine. A symmetric anatomy of 90

muscles was used. Instead of assuming muscles had straight paths between origin and insertion, they were assigned via points that allowed them to wrap around anatomical structures such as other muscles or bony objects. Stability was determined by building a Hessian matrix of the potential energy function as in Crisco and Panjabi (1991). The authors chose to represent muscle stiffness as proportional to muscle activation and scaled by the same stiffness gain q as the previous studies reviewed.

The model was used to investigate movement tasks by analyzing measured human trajectories as a string of successive static equilibrium points. To analyze each point, the model was assigned the equilibrium kinematics. Muscle forces were found using an EMG driven optimization algorithm. This study is particularly interesting because this model used a novel muscle anatomy in addition to using measured data to drive its optimization results. Again, reflexes were ignored in this work and stability of the model was possible in their absence.

v. Shirazi-Adl, Sadouk, Parnianpour, Pop and El-Rich (2002)

This work is similar to that of Cholewicki et al (1996) and Gardner-Morse et al (1995). However, it added kinematic optimization to the spine model. Therefore, it was no longer necessary to specify the kinematics of the spine a priori. The model only required specification of external loads and the orientations of the thorax and pelvis for each experiment. The optimization algorithm found the intermediate lumbar posture (kinematics) that minimized the sum of muscle forces while maintaining stability.

Investigators broke ground by demonstrating that the passive system and the active system can interact in stabilizing the spine. Two spine models were used: a passive finite element model and an active rigid body model with muscles. The passive model was used to find the optimal posture of the spine for a given axial load. This work hinged on the hypothesis that the lordosis angle (a measure of curvature in the lumbar spine) is a function of axial load, in which the curvature of the spine would increase as external load is increased. Once an ideal posture (a posture requiring minimal external forces for equilibrium) was found using the passive model, the active model was then used to solve for optimal muscle activations that provided stability and equilibrium. Any additional compressive loads due to muscle contraction were updated in the passive model, and both models iterated in this fashion until a final solution was reached.

Using novel techniques, this study found interaction between the passive and active systems in the spine. This work highlights the importance of considering the passive aspects of the spine when studying pathology. Briefly, the results agreed with hypotheses that the lordosis angle increases with external load as a result of the spine reducing muscle activations while ensuring stability. Like the previous two studies, this study neglected the role of reflexes and was still able to achieve stability in the model.

All of these models represent a progression of modeling research in the spine. Though each model is slightly different, all were able to achieve stability in the absence of reflexes. This provides evidence for the hypothesis that reflexes may not be necessary for stability. However, while reflexes may not be necessary, it is likely that reflexes offer advantages over intrinsic stiffness alone.

C. Reflexes in the spine

Despite previous modeling studies demonstrating that reflexes are not necessary for stability, their presence *in vivo* suggests that they provide some advantage. Hogan (1984) considered the relative stabilizing role of reflexes and intrinsic stiffness.

> "The required torsional stiffness [for stability] may be provided by negative position feedback [reflex] or by antagonistic coactivation [intrinsic stiffness], or a combination of both. The main difference between the two mechanisms lies in their limitations. Feedback control is limited by transmission delays around neural feedback loops and by the limited bandwidth of open-loop muscle and sensor characteristics. As a result, the maximum feedback gain which may be used to stabilize the system is restricted and the maximum achievable stiffness of the closed-loop system is limited. In contrast, stabilization by antagonistic coactivation is unaffected by neural transmission delays. However, its major limitation is that it incurs an energy cost as the opposing muscles are doing no mechanical work but are consuming metabolic energy. The central nervous system would have to compromise between postural stabilization and metabolic energy consumption." – Hogan (1984)

In his work, Hogan used a model to demonstrate the theoretical possibilities of modulating intrinsic stiffness by cocontracting antagonistic muscles. Like previous work in the spine, Hogan ignored reflex. However the above passage, found in the introduction of his paper, underscores the belief that reflexes may play an important role in stability.

There have been measurement studies in which investigators have attempted to quantify the stabilizing effects of reflexes in biomechanics. Kearney *et al* (1997) performed one such landmark study in an investigation of ankle stiffness. This study used nonlinear systems identification techniques to quantify the contribution of reflexes to stability *in vivo*. The authors perturbed a subject's ankle with a random sequence of angular displacements while subjects performed isotonic contractions. Using systems identification techniques, they were able to demonstrate that reflexes were responsible for a significant amount of the stiffness in the system. More recent work by Moorhouse et al. (2006) used Kearney's experimental methods to study the human torso. The results of this study showed a significant portion of torso stiffness was attributable to muscle reflexes. Some subjects were even shown to be unstable in the absence of reflex! These studies provide clear evidence that reflexes play an important role in stability.

The importance of reflexes in the spine was further elucidated by the work of Radebold et al. (2001) who showed a connection between abnormally delayed reflexes and low back pain. In their study, paraspinal reflex delay times were compared between patients with low back pain and healthy controls. They showed that the delay times of patients were significantly larger than those of normal subjects. One hypothetical explanation of this finding is that patients with large reflex delays are forced to recruit more co-contraction to stabilize the spine with intrinsic stiffness. The increased recruitment of co-contraction would result in increased compressive forces in the spine, commonly considered a precursor for low back pain (Norman et al 1998). This hypothesis is qualitatively confirmed by a study by Van Dieen et al (2003), which showed that patients with low back pain have higher levels of co-contraction than their healthy counterparts. This hypothesis also agrees with those of Hogan (1984), who predicted that reflex delay would cause a tradeoff between reflexes and intrinsic stiffness in stabilizing the spine.

It was our goal to build a computational model that will aid in understanding the tradeoff between reflex and intrinsic stiffness for stabilizing control of the spine. To model reflexes in the spine, it was first necessary to understand the nature of reflexes *in vivo*. Paraspinal reflexes have been characterized as a delayed 2^{nd} order transfer function of muscle stretch rate (Granata et al 2004). This transfer function may be expanded into components that depend on muscle length, stretch rate, and an integration of length about equilibrium. Such a feedback control law is similar to a proportional-integral-differential (PID) controller. Since reflexes represent an increase in stimulus frequency to the motor unit, one must not only consider neurological delay

but also the dynamics of the muscle itself. Therefore, to model muscle reflexes accurately it is necessary to use some dynamic muscle equations in addition to 2nd order transfer functions.

D. Implementing a model of reflexes in the spine

To achieve the goals of this work, it was necessary to build upon previous studies without including unnecessary complications. Therefore we chose a system that was similar to that of Cholewicki et al (1996). It contained the same 18DOF and 90-muscle anatomy (including via points). Muscles had the same intrinsic stiffness modulated by stiffness gain q as the previous studies reviewed. The primary difference was that the model was expanded to include reflex activations as well. The spine was not allowed to optimize its kinematics as in Shirazi-Adl *et al* (2002). However, it was implemented in such a way as to ease further work in this direction.

Muscle reflex recruitment was modeled as proportional to stretch and stretch rate, which is similar to the 2nd order transfer function of velocity while neglecting the integral I term (leaving a P D controller). The integral term was neglected because it would have inhibited a linear stability analyses. By definition an integral controller is not linear. Reflex activations were time delayed to allow an investigation of the effects of reflex delay on the stability of the spine. To determine the stability of the delayed reflexes methods developed by Chen (1995) were used. This work allowed any system with both instantaneous (intrinsic) as well as delayed (reflexive) components to be analyzed for stability about equilibrium. For a given system, the analysis solved for the time delay that caused instability in the system. This method was sufficient for the investigation of the impact of time delays on the stability of the spine.

The goal of this work was to develop a mathematical model of the spine that included muscle reflex activation. The model was sufficient not only to investigate stability of the spine with and without reflexes (specific aim 1), but it was also possible to investigate how reflex delay affected this stability (specific aim 2).

CHAPTER 3
METHODS

A. Model

i. Rigid Bodies and Rotations

The model was a 6 body representation of the spine in three dimensions (3D). This representation is similar to the work by Bergmark (1989) and Cholewicki et al. (1996), in which three rotational degrees of freedom (DOF) were assigned per body, for a total of 18 DOF. Translation between vertebrae was ignored. The bodies composed the five lumbar vertebra (L5,L4,L3,L2,L1) and a rigid body representing the thoracic and cervical spine (T12-C1) (Figure 3.A.1). The geometry of the spine was similar to that of 6 stacked inverted pendulums sitting on a pelvis. The pelvis had kinematic freedom, which meant it technically added another three DOF to the system. However, since the pelvis was not assigned any dynamic properties, it was fixed during simulations.

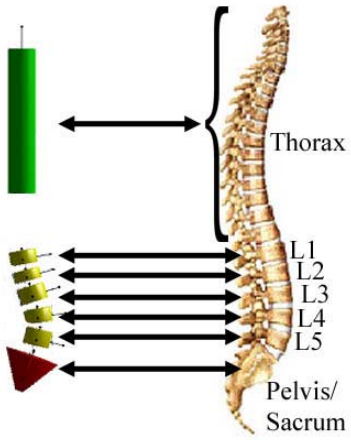


Figure 3.A.1
Rigid Body Diagram

The three DOF of each body correspond to a 1-2-3 rotation of that body (Figure 3.A.2). This means the body fixed coordinate system was determined relative to the inertial coordinate frame by rotating first about axis 1, then about the rotated axis 2, and lastly about the twice rotated axis 3. Axis 1 was orthogonal to the sagital plane, pointing right; axis 2 was orthogonal to the frontal plane, pointing anterior; and axis 3 was orthogonal to the transverse plane, pointing up. Using three Euler rotations in dynamics has the disadvantage of introducing singularities related to the orientation of the body. The 1-2-3 rotation was chosen because this singularity (when the 2 rotation reached 90^{o}) corresponded to a physiologically unrealistic orientation. The singularity point is illustrated in equation 3.A.i.1, where β represents the 2 rotation. If a body were to reach this singularity, the body

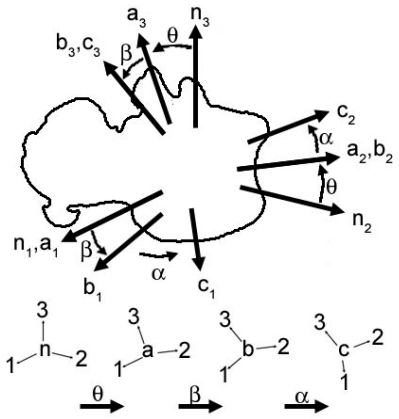


Figure 3.A.2
Rigid Body Rotations
Rotate θ about n1, then β about a2, and lastly α about b3, leaving the fully rotated c coordinate frame.

would appear to be oriented parallel to the ground, a posture that is unlikely when the pelvis is oriented upright.

$$\begin{pmatrix} \theta' \\ \beta' \\ \alpha' \end{pmatrix} = \frac{1}{\cos[\beta]} * \begin{pmatrix} \cos[\alpha] & -\sin[\alpha] & 0 \\ \cos[\beta]\sin[\alpha] & \cos[\alpha]\cos[\beta] & 0 \\ -\cos[\alpha]\sin[\beta] & \sin[\alpha]\sin[\beta] & \cos[\beta] \end{pmatrix} \begin{pmatrix} \omega 1 \\ \omega 2 \\ \omega 3 \end{pmatrix} \quad (3.A.i.1)$$

Each rigid body was modeled as a cylinder with an elliptical cross sectional area. This allowed each body to have three-dimensional inertia approximating a cross section of the trunk. The spine was made to pivot about the center of these cross sections and the mass of each rigid body was similarly applied at the center of these sections. These approximations were made to simplify of the system. Dimensions of the rigid bodies were taken from Liu et al. (1971). The equilibrium kinematics used in this study were an average of the upright standing posture of the lumbar spine presented in Cholewicki et al (1996) and Stokes and Gardner-Morse (1999). In hindsight, better rotational inertia should have been used in which the spine pivots towards the edge of the cylindrical body. Along the same lines, the mass should be applied anterior to the pivot point of each body, which would be a more physiologically correct assumption (Liu et al 1971). However, these issues should not greatly affect the results of the model.

ii. Intervertebral Discs

Very little research has been conducted on intervertebral discs, making it difficult to accurately model them in the spine. However, there have been some studies on these critical components of the spine (Stokes et al 2002, Izambert et al 2003). These studies guided the development of the discs implemented in this model. Previous investigators (Goto et al 2002, Shirazi-Adl et al 2002, Fagan et al 2002) have developed finite element models of the intervertebral discs with approximations of their constitutive material properties. While these models allowed deformation (translation and rotation) of the discs, such complexity was beyond the scope and goals of this project. As such, a simpler disc model was needed.

Two studies in particular influenced the design of the disc model. Stokes et al (2002) designed a six DOF measurement device with which they tested the rotational and translational stiffness of cadaveric discs. The result of their study indicated that the rotational stiffness of the disc was similar in all directions (facet joints were removed from the vertebrae). Unfortunately, these researchers did not report damping. Izambart et al. (2003) investigated translational

damping and stiffness. They found that during low frequency oscillation the damping coefficient was approximately 2 orders of magnitude less than the stiffness coefficient. From these two studies, we developed a 3 rotational DOF model of the intervertebral disc that had the same rotational stiffness (50Nm/rad) and rotational damping (0.5 Nms/rad) about all three axes.

A lumped parameter model of the intervertebral disc was designed that provided an approximation of the physiological disc while maintaining computational simplicity. It was assumed that anterior-posterior and medio-lateral rotations, here termed planar rotations, would behave similarly. Note that this assumption neglects the effects of vertebral facet joints, which would likely show asymmetry between these two rotations. The term "planar rotation" was chosen because such rotations cause the planes from neighboring vertebrae to intersect. "Twist rotations" were so called because unlike planar rotations, two parallel vertebrae under a twist rotation will remain parallel. Therefore the model of the intervertebral discs broke relative rotations between two vertebrae into planar rotations and twist rotations (Figure 3.A.3).

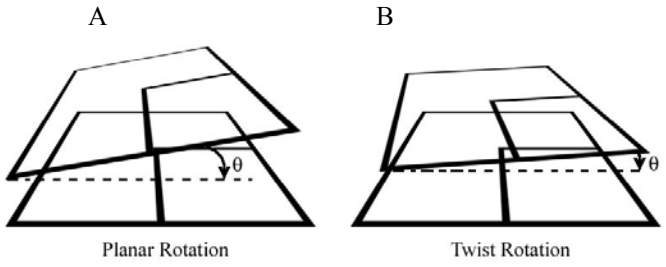


Figure 3.A.3
Planar and Twist rotations in intervertebral disc model

The model first found the planar rotation to be the smallest angle between the two vertebral planes (Figure 3.A.3). To find the twist rotation, a 'dummy plane' originally equal to one of the vertebral planes was rotated by the planar rotation angle, making it parallel to the second vertebral plane. The twist angle was the angle between these two parallel planes. Equal and opposite moments were calculated for each vertebra as the addition of the scaled moment vectors from both the planar and twist calculation. The magnitude of the moment (M) was calculated by assuming that rotational stiffness (K) and damping (B) were linear with angle (θ) (equation 3.A.ii.2).

$$M = K \cdot \theta + B \cdot \dot{\theta} \qquad (3.A.ii.2)$$

iii. Muscle Model

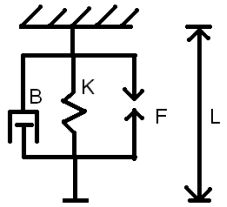


Figure 3.A.4
Hill type muscle model of length L with parallel stiffness K, damping B, and force F

Over 50 years of research have been dedicated to understanding the function of mammalian muscle tissue. This research has generated a variety of mathematical models of muscle contraction. Some of the earlier work considered Hill type muscle models (Hill 1938), which contained a contractile element (force generator) in parallel with a passive element (spring K and damper B) (Zajac 1989). Some of these models also included a series spring intended to represent a tendon, though this element is excluded in our work (Figure 3.A.4). More recent research recognizes that the viscoelastic properties of the muscle are nonlinear (Zajac 1989). The most advanced work in muscle models includes the dynamic properties of a muscle. For example, following frequency modulated muscle activation there are calcium molecular dynamics whose time delays and saturation effects modify the response of muscle. A muscle contraction model by Brown et al. (2000) includes 5 state variables to represent these dynamic processes. Since the goal of our work was to evaluate the spine model in equilibrium posture under constant load, the Hill type model was sufficient (equation 3.A.iii.1, Figure 3.A.4). The Hill model effectively approximates the linear behavior of a nonlinear muscle model about equilibrium.

$$f_m\left(L_m(t)\right) = F + K \cdot \left(L_m(t) - L_{mo}\right) + B \cdot L_m'(t) \qquad (3.A.iii.1)$$

Where muscle force f_m is a function of constant force F, stiffness K, damping B, muscle length $L_m(t)$, and muscle stretch rate $L_m'(t)$.

Intrinsic muscle stiffness K was equal to the stiffness used by Bergmark (1989), and the muscle stiffness of every muscle was modulated by the same dimensionless stiffness gain q. Intrinsic muscle damping B was similar to the intrinsic stiffness, and the muscle damping of every muscle was modulated by the same damping gain b (with units of seconds). Muscle force F was equal to the maximum muscle force f_o modulated by a continuous muscle activation α

(which varied from 0 to 1) (equation 3.A.iii.2). Maximum muscle force f_o was defined as the maximum muscle stress of 46 N/cm^2 multiplied by the muscle cross sectional area (Gardner-Morse et al 1995).

$$K = q\frac{F}{L} \quad B = b\frac{F}{L} \quad F = f_o \cdot \alpha \qquad (3.A.iii.2)$$

By inserting the assigned force, stiffness, and damping from equation 3.A.iii.2 into the muscle model in equation 3.A.iii.1, we are left with a simplified muscle model (3.A.iii.3).

$$f_m(\alpha(t), L_m(t)) = f_o \cdot \alpha(t) \cdot \left(q \cdot \left(\frac{L_m(t) - L_{mo}}{L_{mo}} \right) + b \cdot \frac{L_m'(t)}{L_{mo}} + 1 \right) \qquad (3.A.iii.3)$$

Where muscle force f_m is a function of muscle activation α, muscle length $L_m(t)$, velocity $L_m'(t)$, stiffness gain q, damping gain b, maximum muscle force f_o and equilibrium muscle length L_{mo}.

While this muscle model is simple compared with dynamic nonlinear models, we considered an interesting muscle activation. Rather than using open loop or constant muscle activation, reflexes were added that allowed the muscle activation to change with the stretch and stretch rate of the muscle (3.A.iii.2). These feedback pathways were considered an approximation of the 2nd order transfer functions calculated in measurement studies and were meant to simplify implementation of the model. The feedback was similar to the proportional-differential (PD) linear controllers often used in the robotics industry (see section 2.D for a discussion the reflex model). Therefore total muscle activation α was equal to the sum of steady state activation α_o and reflex activation α_r. Reflex activation allowed muscle activations to be modified in response to changes in length, thus creating a closed system.

$$\alpha(t) = \alpha(\alpha_o, L_m(t)) = \alpha_o + \alpha_r(\alpha_o, L_m(t))$$
$$where \quad \alpha_r(\alpha_o, L_m(t)) = \alpha_o \cdot \left(G_p \cdot \left(\frac{L_m(t-\tau) - L_{mo}}{L_{mo}} \right) + G_d \cdot \frac{L_m'(t-\tau)}{L_{mo}} \right) \qquad (3.A.iii.2)$$

The muscle activation α is a function of steady state activation α_o, reflex activation α_r, proportional reflex gain G_p, and differential reflex gain G_d

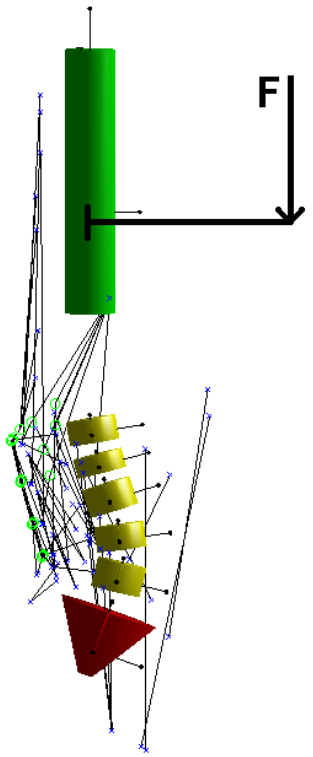


Figure 3.A.5
Muscle Anatomy drawn with an external force F. The right side anatomy is excluded for clarity.

iv. Anatomy

A muscle anatomy similar to the work of Cholewicki et al (1996) was used (See section 2.B.iv). In their work origin and insertion locations of 90 muscles were described in body fixed coordinate frames (Figure 3.A.5). In addition to these points, many of the muscles passed through "via points" that were also described in body fixed coordinates. Via points for the muscles may be thought of as similar to eyelets for shoelace. The lace is allowed to pass freely through the eyelet; however when it passes through at an angle, the lace applies a force to the eyelet (Figures 3.A.6, 3.B.1). These via points allowed the modeled muscles to behave like anatomical muscle, which must pass over bony locations or other muscle tissues between the origin and insertion.

While the anatomic origins and insertions of muscles were obtained in Cholewicki et al (1995), cross sectional areas of the muscles were obtained elsewhere. Stokes and Gardner-Morse (1999) provided the cross sectional areas of the paraspinal musculature. Thus we obtained the muscle anatomy and cross sectional areas from the literature. Muscle tissue is composed of both fast and slow twitch fibers. In addition to having different force responses, these two types of muscle consume different amounts of metabolic power. Unfortunately, there was insufficient data for the distribution of fast and slow twitch fibers in each of the muscles. Therefore, the assumption was made that all of the muscles in the spine were mixed with 50% of each fiber type. While this approximation affected the calculation of metabolic power consumed by the muscles, we did not feel these approximations dramatically affected the results of this study.

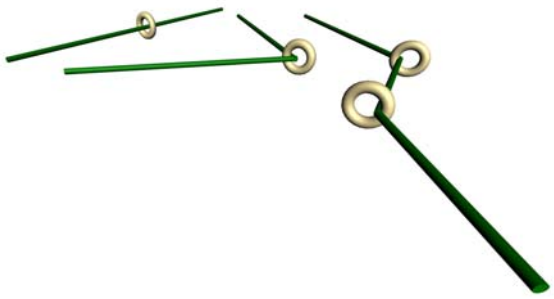


Figure 3.A.6
Illustration of a muscle passing through via points.

B. Dynamics

i. Lagrange's Equations

Before performing any stability analysis on the spine model, it is necessary to mathematically represent how the model moves. Dynamic equations of motion (EOM) of the system were derived using the Lagrange derivative. The equations of motion were derived by performing the Lagrange derivative on the Lagrangian function L, where L=T-V (T is the kinetic energy in the system, and V is the potential energy in the system). The Lagrange derivative of the Lagrangian is equal to the generalized forces vector Q (3.B.i.1), which was calculated for each degree of freedom q_i. Generalized forces are those forces in the system that are not present in the potential energy function (V). This is an important point, as this work calculated passive properties (intrinsic muscle stiffness, intervertebral disc stiffness) as generalized forces *instead* of potential functions, to support an anatomy that could be easily modified without changing any of the code in the model. Finding the equations of motion required the entire system to be represented as either kinetic energy, potential energy, or generalized forces.

$$EOM_i = \frac{d}{dt} \cdot \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i \qquad (3.B.i.1)$$

As stated above, some of the elements in the model, such as the intervertebral disc stiffnesses and muscle stiffnesses, can be represented as either generalized forces or as potential energy. For simplicity in model development, they were represented as generalized forces. As such, potential energy V (equation 3.B.i.2) contains only energy due to gravity.

$$V = \sum_{j=1}^{6} g \cdot m_j \cdot \{G_j\}_3 \qquad (3.B.i.2)$$

where j is the body index, g is gravity, m_j is mass of body j, and $\{G_j\}_3$ is the vertical component of the center of mass of body j.

The kinetic energy T of the system was calculated in equation 3.B.i.3.

$$T = \sum_{j=1}^{6} \left(\frac{1}{2} \cdot m_j \cdot \dot{G}_j \bullet \dot{G}_j + \frac{1}{2} {}^N\omega_j^B \bullet I_j \bullet {}^N\omega_j^B \right) \qquad (3.B.i.3)$$

where j is the body index, m is mass, and $\dot{G}$ is the velocity vector of the center of mass, ${}^{N}\omega^{B}$ is the angular velocity vector of the body in body fixed coordinates, and I_j is the rotational inertia matrix of body j about the center of mass.

The potential and kinetic energy were represented symbolically which allowed the Lagrange derivative to be performed symbolically as well (see Appendix A.1). A matrix representation of the coupled equations was found. The accelerations of the angles were expressed in equation 3.B.i.4. It is important to understand that the generalized force vector Q_v was not found symbolically and was numerically evaluated for all evaluations of $\ddot{q}$.

$$\begin{aligned} &M_m(q)\cdot\ddot{q}+C_v(q,\dot{q})+G_v(q)=Q_v \\ &\ddot{q}=(M_m(q))^{-1}\cdot(Q_v-C_v(q,\dot{q})-G_v(q)) \end{aligned} \qquad (3.B.i.4)$$

where M_m is the mass matrix, C_v is the cross terms and coriolis vector, G_v is the gravity vector, and Q_v is the vector of generalized forces. The generalized coordinate vector q is a vector containing all 18 kinetic degrees of freedom (6x3 rotation angles). Equation 3.B.i.4 allowed us to solve for angular accelerations as a function of the system state and generalized forces.

ii. Generalized Forces

Solving the equations of motion required the generalized forces Q_i for all i degrees of freedom. Note that these were implemented numerically. The contribution of the k^{th} force acting on body j (F_{jk}) on the i^{th} generalized force is shown below (3.B.ii.1), where r_{jk} is the vector between the point of application and the body fixed origin. This method was used for both external forces and muscle forces. Forces applied at via points were calculated as demonstrated in Figure 3.B.1. Intervertebral disc moments were applied to the bodies as pure moments.

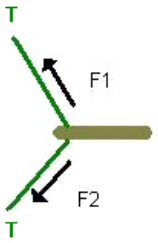
$$Q_i=\sum_{j=1}^{6}\left(\left(\sum_{k=1}^{M_j}F_{jk}\right)\cdot\frac{\partial v_j}{\partial\dot{q}_i}+\left(\sum_{k=1}^{M_j}r_{jk}\times F_{jk}\right)\cdot\frac{\partial\omega_j}{\partial\dot{q}_i}\right) \qquad (3.B.ii.1)$$

Where:

i=DOF (index)	*j*=Body (index)
k=Forces acting on body *j* (index)	F_{jk}=Force k acting on body *j*
r_{jk}=Moment arm of force F_{jk}	v_j=Linear velocity of the origin of body *j*
ω_j=Angular velocity of body *j*	q_i=DOF *i*
Q_i=Generalized forces for DOF *i*	

iii. Linearization

We performed a linear stability analysis of the system. At this point in the analysis there exists the full equations of motion including generalized forces. As stated above only a portion of the equations of motion were solved symbolically, while the generalized forces were found numerically. Therefore equation 3.B.ii.1 was iterated during every evaluation of the acceleration of the system. As such, the following paragraphs describe how the linearization was calculated numerically.



$$F_1 = P_{i-1} - P_i$$
$$F_2 = P_{i+1} - P_i$$
$$Fnode = T * \left(\frac{F_1}{\|F_1\|} + \frac{F_2}{\|F_2\|} \right)$$

Figure 3.B.1
Nodal force from a muscle with tension T

The dynamics of the spine model are nonlinear, therefore a linear analysis required 'linearizing' the system about an equilibrium point. We will investigate the method of finite differencing for finding the linearized representation of a system. It is critical to ensure that the system is in equilibrium before carrying out a linearization. Therefore let us assume that the system is in equilibrium for the following investigation. For a general problem, the differential states of a state vector x may be described as:

$$x = \left[\theta_1,...,\theta_n,\dot{\theta}_1,...,\dot{\theta}_n\right]^t$$
$$\dot{x} = f(x) = f\left(\theta_1,...,\theta_n,\dot{\theta}_1,...,\dot{\theta}_n\right) \qquad (3.B.iii.1)$$
$$x = \bar{x} + \Delta x$$

where $\bar{x}$ is the equilibrium state, and Δx is the state vector about equilibrium. Disturbing only state θ_1 by $\Delta\theta_1$ and leaving the other states at equilibrium, we perform the first taylor series expansion:

$$\dot{x} = f\left(\bar{\theta}_1 + \Delta\theta_1,...,\bar{\theta}_n,\dot{\bar{\theta}}_1,...,\dot{\bar{\theta}}_n\right)$$
$$\dot{x} \approx f(\bar{x}) + \left(\frac{\partial f(x)}{\partial \theta_1}\bigg|_{x=\bar{x}} \right) \cdot \Delta\theta_1 \qquad (3.B.iii.2)$$

This can be repeated for all states using the vector q as a disturbed version of the state vector x.

$$q = x - \bar{x} = \left[\Delta\theta_1,...,\Delta\theta_n,\Delta\dot{\theta}_1,...,\Delta\dot{\theta}_n\right]^t. \qquad (3.B.iii.3)$$

It is possible to build a matrix J such that:

$$\dot{x} \approx f(\bar{x}) + J \cdot q \qquad (3.B.iii.4)$$

Since we know the system is in equilibrium, we know that acceleration is zero at the equilibrium point, thus we have:

$$f(\bar{x}) = 0$$

resulting in the following linear system:

$$\dot{x} = J \cdot q \tag{3.B.iii.5}$$

Lastly, since the equilibrium position is constant we know the time derivative of q is equal to the time derivative of x.

$$\begin{aligned} q &= x - \bar{x} \\ \dot{q} &= \dot{x} = J \cdot q \end{aligned} \tag{3.B.iii.6}$$

This leaves us with a closed linear system with Jacobian *J*. In the spine model *J* was a 36x36 matrix representing the 36 state variables (for 18 degrees of freedom). *J* was assembled by taking numerical partial derivatives about the equilibrium position. The symbolic J matrix is shown below:

$$J = \begin{bmatrix} 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0_{18x18} & 0 & \cdots & 1_{18x18} \\ \frac{\partial \ddot{\theta}_1}{\partial \theta_1} & \cdots & \frac{\partial \ddot{\theta}_1}{\partial \theta_{18}} & \frac{\partial \ddot{\theta}_1}{\partial \dot{\theta}_1} & \cdots & \frac{\partial \ddot{\theta}_1}{\partial \dot{\theta}_{18}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \ddot{\theta}_{18}}{\partial \theta_1} & \cdots & \frac{\partial \ddot{\theta}_{18}}{\partial \theta_{18}} & \frac{\partial \ddot{\theta}_{18}}{\partial \dot{\theta}_1} & \cdots & \frac{\partial \ddot{\theta}_{18}}{\partial \dot{\theta}_{18}} \end{bmatrix} \quad \text{for} \quad q = \begin{bmatrix} \Delta\theta_1 \\ \vdots \\ \Delta\theta_{18} \\ \Delta\dot{\theta}_1 \\ \vdots \\ \Delta\dot{\theta}_{18} \end{bmatrix} \tag{3.B.iii.7}$$

The 18x18 top right block of J is the identity matrix, as it represents the partials of the velocities with respect to themselves. Similarly, the top left of J is the zero matrix, illustrating that the velocities do not depend on positions. To calculate the lower 18x36 half of J, all 36 state variables are varied independently and the 18 acceleration functions are compared to their equilibrium values. An example of how a partial derivative is computed numerically is shown below:

$$\frac{\partial \ddot{\theta}_1}{\partial \theta_1} = \frac{f_1\left(\bar{\theta}_1 + \Delta\theta_1, \ldots, \bar{\theta}_n, \dot{\bar{\theta}}_1, \ldots, \dot{\bar{\theta}}_n\right) - f_1(\bar{x})}{\Delta\theta_1} \tag{3.B.iii.8}$$

Where $\Delta\theta_1$ is a small disturbance of θ_1 about the equilibrium position $\bar{x}$.

The linear system is stable if all of the eigenvalues of its Jacobian have negative real components. If the nonlinear system is within an infinitesimally small region about the equilibrium point, called the linear region, then the nonlinear system will be stable if the linear

system is stable. The size of the linear region is different for all systems, however the region was assumed to be large enough to be physiologically relevant.

iv. Linear Time Delay Methods

While the linear analysis performed above was useful to understand stability of the spine model, we know that the muscle activation reflex used in our muscle model had time delayed feedback. Therefore, it was desirable to understand how this time delayed feedback will affect the linear stability of the system. Chen (1995) developed an eigenvalue method for determining the time delay for which such a delayed system becomes unstable. In this section we expand our linear analysis to include Chen's time delayed stability analysis.

To carry out a linear time delayed stability analysis, the system must be in the following form:

$$\dot{x}(t) = J_i \cdot x(t) + J_d \cdot x(t-\tau) \qquad \text{(3.B.iv.1)}$$

To achieve this form the dynamics must be separated into instantaneous and delayed components. Since all of the delayed components come from the muscle model, we investigated splitting the muscle model into instant and delayed force. By inserting the muscle activation equation (3.A.iii.2) into the muscle force equation (3.A.iii.1) the muscle force model is consolidated into one equation (3.B.iv.2).

$$f_m(\alpha_o, L_m(t)) = f_o \cdot \alpha_o \cdot \left(1 + G_p \cdot \left(\frac{L_m(t-\tau) - L_{mo}}{L_{mo}}\right) + G_d \cdot \frac{L_m'(t-\tau)}{L_{mo}}\right) \cdot \left(q \cdot \left(\frac{L_m(t) - Lm_o}{L_{mo}}\right) + b \cdot \frac{L_m'(t)}{L_{mo}} + 1\right)$$

(3.B.iv.2)

Expanding this equation and removing higher order terms we had the muscle force function in the following form:

$$f_m(\alpha_o, Lm(t)) \approx f_o \cdot \alpha_o + f_o \cdot \alpha_o \cdot \left(q \cdot \left(\frac{L_m(t) - L_{mo}}{L_{mo}}\right) + b \cdot \frac{L_m'(t)}{L_{mo}}\right) + f_o \cdot \alpha_o \cdot \left(G_p \cdot \left(\frac{L_m(t-\tau) - L_{mo}}{L_{mo}}\right) + G_d \cdot \frac{L_m'(t-\tau)}{L_{mo}}\right)$$

$$f_m(\alpha_o, Lm(t)) \approx \{f_{const} + k_i \cdot (L_m(t) - L_{mo}) + d_d \cdot L_m'(t)\} + \{k_d \cdot (L_m(t-\tau) - L_{mo}) + d_d \cdot L_m'(t-\tau)\}$$

$$f_m(\alpha_o, Lm(t)) \approx f_{mi}(\alpha_o, L_m(t)) + f_{md}(\alpha_o, L_m(t-\tau))$$

(3.B.iv.3)

This manipulation grouped muscle forces as a function of the instantaneous state into f_{mi}, and forces as a function of the delayed state into f_{md}. In the above equation, f_{const} is steady state force, k_i and d_i are instantaneous stiffness and damping, and k_d and d_d are delayed stiffness and

damping, respectively. The separated muscle forces are then carried through the generalized force vector Q. This results in an instantaneous and delayed generalized force vector. The instantaneous generalized force vector Q_{vi} contained intervertebral disc forces, external forces, and instantaneous muscle forces f_{mi}, while delayed generalized force vector Q_{vd} contained only the delayed muscle forces f_{md}.

$$\begin{aligned} Q_v &= Q_{vi} + Q_{vd} \\ \ddot{\theta}(t) &= M_m^{-1}(Q_v - G_v - C_v) = M_m^{-1}(Q_{vi} - G_v - C_v) + M_m^{-1}(Q_{vd}) \end{aligned} \qquad \text{(3.B.iv.4)}$$

This results in acceleration due to instantaneous states and delayed states.

$$\ddot{\theta}(t) = \ddot{\theta}_i(t) + \ddot{\theta}_d(t) \qquad \text{(3.B.iv.5)}$$

Ideally $\ddot{\theta}_d(t)$ would be composed entirely of components due to the delayed state. Instead, it was composed of delayed muscle forces and instantaneous components. Since the moment arms in Q_{vd} and the mass matrix M_m were instantaneous, $\ddot{\theta}_d(t)$ contains both instantaneous and delayed components. To build the delayed Jacobian J_d an assumption of the relative phase between the instantaneous and delayed components was needed. Since the oscillatory period of the spine is on the order of 1s, and physiological reflex delay is approximately 80ms (Radebold et al 2001), it was assumed that these instantaneous components moved in phase with the muscle forces. Therefore all instantaneous components in M_m and the moment arms in Q_{vd} were converted to delayed components for the sake of linearizing the system. (See section 4.C or 3.D.v for verification of this method with nonlinear simulation results)

Now that we had acceleration due to instantaneous and delayed components in two separate equations, we could build our instantaneous Jacobian J_i and our delayed Jacobian J_d in the same manor as in section 3.B.iii. Since we assumed the delayed acceleration to be composed entirely of delayed components, we performed finite differencing on the entire delayed acceleration vector.

$$J_i = \begin{bmatrix} \frac{\partial \dot{\theta}_i(t)}{\partial \theta(t)} & \frac{\partial \dot{\theta}_i(t)}{\partial \dot{\theta}(t)} \\ \frac{\partial \ddot{\theta}_i(t)}{\partial \theta(t)} & \frac{\partial \ddot{\theta}_i(t)}{\partial \dot{\theta}(t)} \end{bmatrix} \quad J_d = \begin{bmatrix} \frac{\partial \dot{\theta}_d(t)}{\partial \theta(t-\tau)} & \frac{\partial \dot{\theta}_d(t)}{\partial \dot{\theta}(t-\tau)} \\ \frac{\partial \ddot{\theta}_d(t)}{\partial \theta(t-\tau)} & \frac{\partial \ddot{\theta}_d(t)}{\partial \dot{\theta}(t-\tau)} \end{bmatrix} \qquad \text{(3.B.iv.6)}$$

The goal was to find the time delay τ that caused the system in 3.B.iv.1 (restated below for clarity) to become unstable.

$$\dot{x}(t) = J_i \cdot x(t) + J_d \cdot x(t-\tau)$$

The methods detailed here (from Chen 1995) made the assumption that the time delayed system is stable in the zero delay (τ=0) case. This means that the eigenvalues of $J=J_i+J_d$ were all in the left complex plane, and thus they all had negative real components. Since we knew that they had negative real components, we wanted to find the time delay at which one of the eigenvalues crossed the imaginary axis. At this point, the eigenvalue would have no real component and the system would be marginally stable. The delay margin was the smallest time delay for which one of the eigenvalues crossed the imaginary axis. To solve for this time delay, we had to transform to the Laplace domain:

$$sX = J_i \cdot X + J_d \cdot X \cdot e^{-s\cdot\tau} \qquad \text{(3.B.iv.7)}$$

Since we were searching for the eigenvalue with a real component of zero, we modified the Laplace variable s to have no real component.

$$s \to j\omega \quad \Rightarrow \quad j\omega \cdot X = J_i \cdot X + J_d \cdot X \cdot e^{-j\omega\tau} \qquad \text{(3.B.iv.8)}$$

This equation (3.B.iv.8) was manipulated into a generalized eigenvalue problem:

$$\left(j\omega \cdot I - J_i\right) \cdot X = \lambda \cdot J_d \cdot X \quad \lambda = e^{-j\omega\tau} = e^{-j\theta} \qquad \text{(3.B.iv.9)}$$

Equation 3.B.iv.9 then allowed us to solve for the set of time delays at which eigenvalues crossed the imaginary axis. The smallest delay in this set was the delay margin of the system. To find the delays, we treated frequency ω as the independent variable, and generalized eigenvalue λ as the dependant variable. We varied ω from 0 to ω_{max} and searched instances in which the magnitude of λ was unity. At such frequencies, λ had the form of $e^{-j\theta}$ (with a magnitude of unity), as in equation 3.B.iv.9. This signified a crossing event, and the corresponding time delay was the phase (θ) of λ divided by the frequency ω. This problem is stated mathematically in equation 3.B.iv.10. (See section 3.C.iii.c for numerical techniques to tackle this problem)

$$\omega = [0, \omega_{\max}) \quad \forall \omega_i \ni \lambda\left(\left(j\omega_i \cdot I - J_i\right), J_r\right) = e^{-j\theta_i} \quad \tau_i = \frac{\theta_i}{\omega_i} \qquad \text{(3.B.iv.10).}$$

C. Applications

We developed the methods for building and analyzing the spine model. However the use of these methods have not yet been addressed. This section will attempt to address the general nature of the methods, and how experiments may take advantage of them.

i. Optimization

We have discussed stability as a function of muscle forces without addressing how to acquire these muscle activations. In this section we will discuss the possibilities of using optimization methods to establish the muscle activations of the model. The advantage of using these methods is two fold: they allow one to specify a cost function that is to be minimized and they allow one to constrain the solution to enforce a required condition.

A constrained optimization was chosen to solve for the muscle activations α_o that minimized metabolic power $P(\alpha_o)$ while ensuring the spine to be in equilibrium and stable with zero time delay. In our implementation the fmincon algorithm (Mathworks) was chosen because it was easily available, simple to use, and switched between multiple methods as it searched the space. A mathematical representation of our optimization problem is shown below:

$$\text{Find } \alpha_o \ni \begin{cases} \min \sum P(\alpha_o) & \text{Cost Function} \\ \ddot{\theta}(\bar{x}, \alpha_o) = 0 & \text{Equilibrium Constraint} \\ \text{Re}(\lambda(J_i + J_d)) < 0 & \text{Stability Constraint} \end{cases} \qquad (3.C.i.1)$$

The power equation $P(\alpha_o)$ derived from Anderson (1999) calculated the power required to maintain muscle contraction. It was a function of muscle activation and was proportional to muscle mass. As stated previously, it was assumed that all muscles in the model contained equal amounts of slow and fast twitch fibers.

$$P(\alpha_o) = m \cdot \left(\frac{74}{2} \cdot Sin\left(\frac{\alpha_o \cdot \pi}{2} \right) + \frac{111}{2} \cdot \left(1 - Cos\left(\frac{\alpha_o \cdot \pi}{2} \right) \right) \right) \qquad (3.C.i.2)$$

An optimization algorithm requires a starting point and a domain over which to search. For this model the initial seed location was not found to affect the solution. All muscle activations were bounded over the domain [0,1), which ranges from no activation to full activation. The optimization algorithm evaluated the cost function and the constraint functions across multiple iterations of the muscle activation vector. It attempted to find the muscle

activation vector which simultaneously minimized the cost function while satisfying the constraint functions.

One criticism of optimization methods is that they tend to find 'local minima' instead of 'global minima'. In other words, instead of finding the absolute best solution, optimization algorithms tend to find the best solution in the region in which they search. One way to mitigate this limitation is to perform multiple iterations of the optimization at random initial locations. If all of the iterations result in the same solution, the solution is likely to be the global minimum.

ii. Experiments

The fully developed model may be used to perform 'experiments'. The word experiment may be misleading, as the word is typically used to describe measurements of the physical world. However, learning the qualitative nature of a model requires performing experiments in which controlled variables (such as reflex gain) are varied, while observations are made. No knowledge of the natural world is gained by performing experiments with a model. Instead, knowledge about the model is gained, and such knowledge may then be compared with real world measurement. Should results from the spine model match physiological measurement, the model may then be a reasonable approximation to the physiological system.

There are two advantages to having a model that appears to match real world data. First, the fact that the model is a reasonable approximation of the natural world is knowledge that may help to identify the governing variables in the natural system. Second, the model may be used to generate future hypotheses to test in the natural world. If the model is able to predict the real world there is strong evidence that real world looks like the model.

With these advantages in mind, designing experiments should attempt to recreate a condition that may be tested in the real world. Unfortunately, experiments on human subjects must be limited in scope. Specifically, no experiments may be performed in which a subject is aversely affected. Therefore, one way to study the human system is to observe how pathological individuals differ from normal individuals. The pathology is the control variable, and the two populations may be observed to qualify differences between them. It is possible to test the spine model by 'giving it a pathology.' If observations following this change are qualitatively identical to differences between the pathological and normal populations, there is evidence that

the governing structure of the model is similar to the physiological system. Experiments on the spine model should be designed with this method of verification in mind.

iii. Computational Techniques

While the methods described in this text are sufficient to create a similar model, there are many pitfalls in the implementation that may be avoided through some numerical techniques. A few of the most important techniques will be discussed here.

a. Optimization-Jacobian

The optimization algorithm described in the previous section is an efficient way to find muscle activations that satisfy desired criteria. However, if the algorithm were implemented as described above it would be a computational disaster. Jacobians J_i and J_d are functions of muscle activation, and thus must be evaluated for every new muscle activation vector. As it takes several seconds to evaluate each Jacobian, it would take much time to complete the optimization. By exploiting a characteristic of the muscle model, it was possible to simplify these computations.

The fact that the muscle force model scales with activation is a critical feature that allows for simplifying computation of a Jacobian. As the optimization varies the muscle activations we wish to take the new activation vector and quickly find the Jacobian. We may do so by finding 'partial Jacobians' which represent the effects of each individual muscle with full activation (α_o=1), while all other muscle activations are zero. This method allowed the 36x36 Jacobian to be assembled by expanding it into a 36x36xM matrix (for M muscles). The full Jacobian was recovered by taking the dot product of this matrix along the M dimension with the muscle activation vector. The key advantage was that this 36x36xM matrix was constant (as a function of the initial conditions only) and therefore only needed to be evaluated once.

b. Compiled Binaries

Computing the Lagrange equations of motion as specified in 3.B.i on a system of this size will result in equations that are difficult to manage. For example, printing the equations of motion of the 18DOF spine in 12pt font would take over 800 pages. Such equations are not well suited for interpreted languages such as Matlab. Among other inefficiencies, un-compiled code

wastes time recalculating the same sin(x) that repeats itself 1000 times. By porting the code to the C language, it may be compiled into a binary program using an optimizing compiler and later accessed as a binary from Matlab. It is not unreasonable to expect execution of compiled code to be over 100 times faster than the interpreted code, especially with such large equations. (See appendix A.ii for details on creating binary mex files)

c. Frequency Scaling in the Linear Time Delay Stability Criterion

The linear time delay methods are limited in their complexity by realizing that a maximum frequency ω_{max} exists. Since a frequency sweep involves testing the solution at multiple points in the frequency domain, a maximum frequency simplifies the analyses because there is no need to search for solutions at frequencies larger than this value. However, this advantage does not help if the value of ω_{max} is unknown. Therefore, a more practical method must be obtained for rapidly sweeping the frequency domain over an unspecified range.

A technique that was employed in this work was to have rescaling resolution over the frequency domain. With knowledge of the system in question, one can estimate a reasonable value of ω_{max}. For example, in this work the maximum frequency was below 100 rad/s. One should define a temporary ω_{max} as the estimated value multiplied by 10~100 (as a safety margin). There should be no question that this new ω_{max} is larger than the true ω_{max} in the system. By performing a frequency sweep of the entire region [0, ω_{max}) with a low resolution (~1000 frequencies tested) every crossing should be observed coarsely. The highest frequency crossing is a rough estimate of the true ω_{max}. Therefore ω_{max} should be redefined to something slightly larger than the highest crossing. A high resolution sweep (~10000 frequencies tested) should be performed on the revised range, and crossings should be interpolated. This method results in very high accuracy of crossing instances, and allows the time delayed stability analysis to be entirely automated for systems with frequencies within the range specified.

D. One DOF Example

To understand the analyses performed in this study, it is helpful to visualize the dynamic equations. However, to present the equations for the full 18 DOF system would require hundreds of pages and be of little benefit to the reader. Therefore, let us instead consider the single inverted pendulum system in Figure 3.D.1. This system is simple enough that all of the equations used in the analyses may be displayed symbolically. (See appendix C for Mathematica and Matlab programs supporting this work)

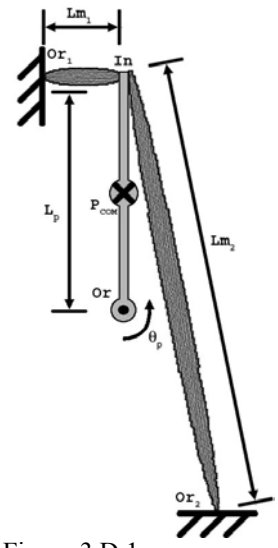


Figure 3.D.1
1 DOF system diagram

In this system the pendulum rotates about the pivot point *Or* and is stabilized by muscles M1 and M2. Muscle M1 attaches at origin Or_1 and muscle M2 attaches at origin Or_2. Both muscles share the insertion point $I_n(t)$ at the end of the pendulum. Gravity is applied at the center of mass of the pendulum $P_{com}(t)$. Muscles 1 and 2 produce positive (tensile) muscle force $fm_1(t)$ and $fm_2(t)$ along unit force vectors $F_1(t)$ and $F_2(t)$, respectively. Muscle force *fm* is a function of muscle activation $\alpha(t)$ and muscle length $Lm(t)$ ($fm(t)=fm(\alpha(t),Lm(t))$).

The system will be explored with the following parameters at equilibrium:
Lengths: L_p=1, Lm_1=0.5, and Lm_2=2.
Angles: θ_p=0°(upright), 90° between the pendulum and M1, 10° between the pendulum and M2.
Mass: m=1

i. Lagrangian Dynamics

Kinematics

The location of the pendulum center of mass $\vec{P}_{com}(t)$ and its time derivative $\dot{\vec{P}}_{com}(t)$ were:

$$\vec{P}_{com}(t)=\left\{-\frac{L}{2}\cdot Sin(\theta_p(t)),\frac{L}{2}\cdot Cos(\theta_p(t)),0\right\}\quad \dot{\vec{P}}_{com}(t)=\left\{-\frac{L}{2}\cdot Cos(\theta_p)\cdot\theta_p'(t),-\frac{L}{2}\cdot Sin(\theta_p)\cdot\theta_p'(t),0\right\} \qquad (3.D.i.1)$$

and the unit force vectors of muscles M1 and M2 acting on moment arm R were:

$$\hat{F}_1(t)=\frac{Or_1-I_n(t)}{|Or_1-I_n(t)|}\quad \hat{F}_2(t)=\frac{Or_2-I_n(t)}{|Or_2-I_n(t)|}\quad \vec{R}(t)=I_n(t)-Or \qquad (3.D.i.2)$$

Kinetics

The kinetic energy T and potential energy V of the system are shown below. A curly bracket subscript such as $\{\}_n$ is used to refer to the n^{th} component of the enclosed vector.

$$\begin{matrix} T = \frac{1}{2}\cdot m\cdot\left(\dot{\bar{P}}_{com}(t)\bullet\dot{\bar{P}}_{com}(t)\right)+\frac{1}{2}\cdot I_o\cdot\dot{\theta}_p(t)^2 \\ V = g\cdot m\cdot\left\{\bar{P}_{com}(t)\right\}_2 \end{matrix} \quad\Rightarrow\quad \begin{matrix} T = \frac{m\cdot L^2}{6}\cdot\dot{\theta}_p(t)^2 \\ V = \frac{g\cdot m\cdot L}{2}\cdot Cos\left(\theta_p(t)\right) \end{matrix} \qquad (3.D.i.3)$$

Equations of Motion

Apply the Lagrange Derivative for the equation of motion (EOM):

$$EOM = \frac{d}{dt}\left(\frac{\partial T}{\partial\theta_p'(t)}\right)-\frac{\partial T}{\partial\theta_p(t)}+\frac{\partial V}{\partial\theta_p(t)} \quad EOM = \frac{m\cdot L}{6}\cdot\left(2\cdot L\cdot\ddot{\theta}_p(t)-3\cdot g\cdot Sin\left(\theta_p(t)\right)\right) \qquad (3.D.i.4)$$

We can represent the EOM in 'matrix' form and apply a generalized force vector Q_v. Note that the matrix and vectors depicted in this equation are actually scalars since this problem has only a single degree of freedom. However, in the full problem M_m is a matrix and G_v, C_v, and Q_v are vectors. Also note that while C_v is zero in this example, it is non-zero in the higher degree of freedom system.

$$\begin{matrix} M_m\cdot\ddot{\theta}_p(t)+G_v+C_v = Q_v \\ M_m = \frac{m\cdot L^2}{3} \quad G_v = -\frac{g\cdot m\cdot L}{2}\cdot Sin\left(\theta_p(t)\right) \quad C_v = 0 \quad Q_v = fm_1(t)\cdot\left\{\bar{R}(t)\times\hat{F}_1(t)\right\}_3+fm_2(t)\cdot\left\{\bar{R}(t)\times\hat{F}_2(t)\right\}_3 \end{matrix} \qquad (3.D.i.5)$$

Q_v is a complicated function of $\theta_p(t)$. However, evaluated at equilibrium ($\theta_p(t)$=0) it has the following simple form:

$$Q_v = fm_1 - fm_2\cdot Sin\left(\frac{\pi}{18}\right) \qquad (3.D.i.6)$$

ii. Generalized Forces

The values of force in M1 and M2, fm_1 and fm_2 respectively, were determined from the muscle force model detailed in section 3.A.iii. The force model separated into instantaneous and delayed components is restated here, from section 3.B.iv.

$$fm(\alpha_o, Lm(t))\approx f_o\cdot\alpha_o+f_o\cdot\alpha_o\cdot\left(q\cdot\left(\frac{Lm(t)-Lm_o}{Lm_o}\right)+b\cdot\frac{Lm'(t)}{Lm_o}\right)+f_o\cdot\alpha_o\cdot\left(G_p\cdot\left(\frac{Lm(t-\tau)-Lm_o}{Lm_o}\right)+G_d\cdot\frac{Lm'(t-\tau)}{Lm_o}\right)$$

$$fm(\alpha_o, Lm(t))\approx fmi(\alpha_o, Lm(t))+fmd(\alpha_o, Lm(t-\tau))$$

The full equation of generalized force Q_v as a function of $\theta_p(t)$ is long, even in this simple example. However in the interest of clarity, the equation is shown below. It should be noted that the approximation from the previous section whereby higher order terms were neglected is carried through this step. As such, muscle force was separated into an instantaneous and delayed component. The complexity in this equation is due to the number of square roots associated with the length and velocity components of the muscle force equations, as well as the unit force vectors. We can break it into components:

$$Q_v = [fmi_1(t) + fmd_1(t-\tau)] \cdot \left\{\bar{R}(t) \times \hat{F}_1(t)\right\}_3 + [fmi_2(t) + fmd_2(t-\tau)] \cdot \left\{\bar{R}(t) \times \hat{F}_2(t)\right\}_3 \qquad (3.D.ii.1)$$

$\bar{R}(t) =$

```
{-1. L Sin[θp[t]], L Cos[θp[t]], 0.}
```

$\hat{F}_1(t) =$

$$\left\{\frac{-1.\,\mathrm{Lml} + L\,\mathrm{Sin}[\theta p[t]]}{\sqrt{(L - 1.\,L\,\mathrm{Cos}[\theta p[t]])^2 + (-1.\,\mathrm{Lml} + L\,\mathrm{Sin}[\theta p[t]])^2}},\ \frac{L - 1.\,L\,\mathrm{Cos}[\theta p[t]]}{\sqrt{(L - 1.\,L\,\mathrm{Cos}[\theta p[t]])^2 + (-1.\,\mathrm{Lml} + L\,\mathrm{Sin}[\theta p[t]])^2}},\ 0.\right\}$$

$\hat{F}_2(t) =$

$$\left\{\frac{0.173648\,\mathrm{Lm2} + L\,\mathrm{Sin}[\theta p[t]]}{\sqrt{(L - 0.984808\,\mathrm{Lm2} - 1.\,L\,\mathrm{Cos}[\theta p[t]])^2 + (0.173648\,\mathrm{Lm2} + L\,\mathrm{Sin}[\theta p[t]])^2}},\ \frac{L - 0.984808\,\mathrm{Lm2} - 1.\,L\,\mathrm{Cos}[\theta p[t]]}{\sqrt{(L - 0.984808\,\mathrm{Lm2} - 1.\,L\,\mathrm{Cos}[\theta p[t]])^2 + (0.173648\,\mathrm{Lm2} + L\,\mathrm{Sin}[\theta p[t]])^2}},\ 0.\right\}$$

$\left\{\bar{R}(t) \times \hat{F}_1(t)\right\}_3 =$

$$\frac{L\,(\mathrm{Lml}\,\mathrm{Cos}[\theta p[t]] - 1.\,L\,\mathrm{Sin}[\theta p[t]])}{\sqrt{(L - 1.\,L\,\mathrm{Cos}[\theta p[t]])^2 + (-1.\,\mathrm{Lml} + L\,\mathrm{Sin}[\theta p[t]])^2}}$$

$\left\{\bar{R}(t) \times \hat{F}_2(t)\right\}_3 =$

$$\frac{L\,(-0.173648\,\mathrm{Lm2}\,\mathrm{Cos}[\theta p[t]] + (-1.\,L + 0.984808\,\mathrm{Lm2})\,\mathrm{Sin}[\theta p[t]])}{\sqrt{(L - 0.984808\,\mathrm{Lm2} - 1.\,L\,\mathrm{Cos}[\theta p[t]])^2 + (0.173648\,\mathrm{Lm2} + L\,\mathrm{Sin}[\theta p[t]])^2}}$$

$fmi_1(t) =$

$$\mathrm{fo}\alpha\mathrm{l}\left(1. + \frac{q\left(-1.\sqrt{\mathrm{Lml}^2} + \sqrt{L^2(-1. + \mathrm{Cos}[\theta p[t]])^2 + (\mathrm{Lml} - 1.\,L\,\mathrm{Sin}[\theta p[t]])^2}\right)}{\sqrt{\mathrm{Lml}^2}} + \frac{(1.\,b\,L^2\,\mathrm{Sin}[\theta p[t]] + \mathrm{Cos}[\theta p[t]]\,(-1.\,b\,L\,\mathrm{Lml} + 0.\,b\,L^2\,\mathrm{Sin}[\theta p[t]]))\,\theta p'[t]}{\sqrt{\mathrm{Lml}^2}\,\sqrt{L^2(-1. + \mathrm{Cos}[\theta p[t]])^2 + (\mathrm{Lml} - 1.\,L\,\mathrm{Sin}[\theta p[t]])^2}}\right)$$

$fmd_1(t) =$

$$\frac{\mathrm{fo}\alpha\mathrm{l}\left(GP\left(-1.\sqrt{\mathrm{Lml}^2} + \sqrt{L^2(-1. + \mathrm{Cos}[\theta p[t]])^2 + (\mathrm{Lml} - 1.\,L\,\mathrm{Sin}[\theta p[t]])^2}\right) + \frac{(1.\,GDL^2\,\mathrm{Sin}[\theta p[t]] + \mathrm{Cos}[\theta p[t]]\,(-1.\,GDL\,\mathrm{Lml} + 0.\,GDL^2\,\mathrm{Sin}[\theta p[t]]))\,\theta p'[t]}{\sqrt{L^2(-1. + \mathrm{Cos}[\theta p[t]])^2 + (\mathrm{Lml} - 1.\,L\,\mathrm{Sin}[\theta p[t]])^2}}\right)}{\sqrt{\mathrm{Lml}^2}}$$

$fmi_2(t) =$

$$fo\,\alpha 2\left(1.+\frac{1.\,q\left(-1.\sqrt{Lm2^2}+\sqrt{(-1.\,L+0.984808\,Lm2+L\,Cos[\theta p[t]])^2+(0.173648\,Lm2+1.\,L\,Sin[\theta p[t]])^2}\right)}{\sqrt{Lm2^2}}+\right.$$

$$\left.\frac{0.5\,b\,(L\,(2.\,L-1.96962\,Lm2)\,Sin[\theta p[t]]+Cos[\theta p[t]]\,(0.347296\,L\,Lm2+0.\,L^2\,Sin[\theta p[t]]))\,\theta p'[t]}{\sqrt{Lm2^2}\,\sqrt{(-1.\,L+0.984808\,Lm2+L\,Cos[\theta p[t]])^2+(0.173648\,Lm2+1.\,L\,Sin[\theta p[t]])^2}}\right)$$

$$fmd_2(t)=$$

$$\frac{fo\,\alpha 2\left(1.\,GP\left(-1.\sqrt{Lm2^2}+\sqrt{(-1.\,L+0.984808\,Lm2+L\,Cos[\theta p[t]])^2+(0.173648\,Lm2+1.\,L\,Sin[\theta p[t]])^2}\right)+\frac{0.5\,GD\left(L\,(2.\,L-1.96962\,Lm2)\,Sin[\theta p[t]]+Cos[\theta p[t]]\,(0.347296\,L\,Lm2+0.\,L^2\,Sin[\theta p[t]])\right)\theta p'[t]}{\sqrt{(-1.\,L+0.984808\,Lm2+L\,Cos[\theta p[t]])^2+(0.173648\,Lm2+1.\,L\,Sin[\theta p[t]])^2}}\right)}{\sqrt{Lm2^2}}$$

iii. Linearization

Before constructing the linear system, we had to ensure that the system was in equilibrium. Plugging numerical parameter values into the nonlinear differential equation and setting acceleration equal to zero, we were left with an equilibrium condition (equation 3.D.iii.1). This condition found α_1 to be linearly related to α_2, which we should expect since an increase of force in one muscle would require a proportional increase of force in the opposite muscle to prevent motion.

$$\ddot{\theta}_p(t)=M_m^{-1}(Q_v-G_v-C_v)=0 \quad \Rightarrow \quad \alpha_1=0.1736\cdot\alpha_2 \tag{3.D.iii.1}$$

We wished to represent our system in linear state space form for the state vector $x(t)=[\theta_p(t), \theta_p'(t)]^t$. This system included both an instantaneous Jacobian J_i and a delayed Jacobian J_d.

$$\dot{x}(t)=J_i\cdot x(t)+J_d\cdot x(t-\tau) \tag{3.D.iii.2}$$

However, our system was represented by a nonlinear differential equation:

$$\ddot{\theta}_p(t)=M_m^{-1}(Q_v-G_v-C_v)=\frac{3}{m\cdot L^2}\left(Q_v+\frac{g\cdot m\cdot L}{2}\cdot Sin(\theta_p(t))\right) \tag{3.D.iii.3}$$

Since we separated muscle force into instantaneous and delayed components, we used these componenets to separate acceleration into acceleration caused by instantaneous states ($\ddot{\theta}_{pi}(t)$) and acceleration caused by delayed states ($\ddot{\theta}_{pd}(t)$).

$$\ddot{\theta}_p(t)=\ddot{\theta}_{pi}(t)+\ddot{\theta}_{pd}(t)$$
$$\ddot{\theta}_{pi}(t)=\frac{3}{m\cdot L^2}\left(f_{1i}(t)\cdot\left(\vec{R}(t)\times\hat{F}_1(t)\right)+f_{2i}(t)\cdot\left(\vec{R}(t)\times\hat{F}_2(t)\right)+\frac{g\cdot m\cdot L}{2}\cdot Sin\left(\theta_p(t)\right)\right) \qquad \text{(3.D.iii.4)}$$
$$\ddot{\theta}_{pd}(t)=\frac{3}{m\cdot L^2}\left(f_{1d}(t-\tau)\cdot\left(\vec{R}(t)\times\hat{F}_1(t)\right)+f_{2d}(t-\tau)\cdot\left(\vec{R}(t)\times\hat{F}_2(t)\right)\right)$$

To create the linear system, we first applied our equilibrium condition (3.D.iii.1), which is necessary for the system to be linear. Next, we computed the Jacobian matrices J_i and J_d (see section 3.B.iv). As stated in section 3.B.iv, due to the assumption that the system moves much slower than the reflex delay, instantaneous states in the delayed acceleration were converted to delayed states. Therefore delayed acceleration $\ddot{\theta}_{pd}(t)$ was changed from 3.D.iii.4 to equation 3.D.iii.5.

$$\ddot{\theta}_{pd}(t)=\frac{3}{m\cdot L^2}\left(f_{1d}(t-\tau)\cdot\left(\vec{R}(t-\tau)\times\hat{F}_1(t-\tau)\right)+f_{2d}(t-\tau)\cdot\left(\vec{R}(t-\tau)\times\hat{F}_2(t-\tau)\right)\right) \qquad \text{(3.D.iii.5)}$$

Instantaneous Jacobian J_i was found by taking the partial derivatives of $\dot{\theta}_{pi}(t)$ and $\ddot{\theta}_{pi}(t)$ with respect to $\theta_p(t)$ and $\dot{\theta}_p(t)$, evaluated at equilibrium ($\theta_p(t)$=0, $\dot{\theta}_p(t)$=0). Similarly the delayed Jacobian J_d was found by taking the partial derivatives of $\dot{\theta}_{pd}(t)$ and $\ddot{\theta}_{pd}(t)$ with respect to $\theta_p(t-\tau)$ and $\dot{\theta}_p(t-\tau)$, evaluated at equilibrium ($\theta_p(t-\tau)$=0, $\dot{\theta}_p(t-\tau)$=0). The velocity $\dot{\theta}_p(t)$ was equal to the instantaneous velocity $\dot{\theta}_{pi}(t)$ and was not a function of the delayed velocity $\dot{\theta}_{pd}(t)$. Therefore the top right value of J_i is one, while for J_d the value is zero.

$$J_i=\begin{bmatrix}\frac{\partial\dot{\theta}_{pi}(t)}{\partial\theta_p(t)} & \frac{\partial\dot{\theta}_{pi}(t)}{\partial\dot{\theta}_p(t)}\\ \frac{\partial\ddot{\theta}_{pi}(t)}{\partial\theta_p(t)} & \frac{\partial\ddot{\theta}_{pi}(t)}{\partial\dot{\theta}_p(t)}\end{bmatrix}\quad J_d=\begin{bmatrix}\frac{\partial\dot{\theta}_{pd}(t)}{\partial\theta_p(t-\tau)} & \frac{\partial\dot{\theta}_{pd}(t)}{\partial\dot{\theta}_p(t-\tau)}\\ \frac{\partial\ddot{\theta}_{pd}(t)}{\partial\theta_p(t-\tau)} & \frac{\partial\ddot{\theta}_{pd}(t)}{\partial\dot{\theta}_p(t-\tau)}\end{bmatrix}$$
$$J_i=\begin{bmatrix}0 & 1\\ \frac{3g}{2}+\alpha_2\cdot f_o\cdot(1.499-q\cdot 1.087) & -\alpha_2\cdot f_o\cdot b\cdot 1.087\end{bmatrix}=\begin{bmatrix}0 & 1\\ K_i & B_i\end{bmatrix} \qquad \text{(3.D.iii.6)}$$
$$J_d=\begin{bmatrix}0 & 0\\ -\alpha_2\cdot f_o\cdot G_p\cdot 1.087 & -\alpha_2\cdot f_o\cdot G_d\cdot 1.087\end{bmatrix}=\begin{bmatrix}0 & 0\\ K_d & B_d\end{bmatrix}$$

For intrinsic stability, in which the system is stable for all time delays, it is required that $K_i<0$ and $B_i<0$ (neglecting reflex). Looking at K_i, it is clear that the minimum value of α_2 for stability is dependent on the size of q. Since the maximum value of α_2 is unity (as defined in the

muscle model), for g=9.8 and f_o=1 the minimum value of q is 14.9. With an infinite value of q, the minimum value of α_2 that permits stability approaches zero. However the system does not need to be intrinsically stable. As long as $K_i+K_d<0$ and $B_i+B_d<0$, the system will be stable for any time delay below the delay margin. The delay margin is determined by performing a linear time delay stability analysis. If the system is stable regardless of time delay, the corresponding delay margin is infinite.

iv. Linear Time Delayed Stability

Taking the system above (equation 3.D.iii.6), we chose the following parameter values: g=10, f_o=1, q=20, b=2, α_2=0.5, G_p=15, G_d=3, resulting in the following system:

$$\dot{x}(t) = J_i \cdot x(t) + J_d \cdot x(t-\tau) \qquad (3.D.iv.1)$$

$$J_i = \begin{bmatrix} 0 & 1 \\ 4.88 & -1.09 \end{bmatrix} \quad J_d = \begin{bmatrix} 0 & 0 \\ -8.15 & -1.63 \end{bmatrix} \qquad (3.D.iv.2)$$

We first consided the system with zero time delay by setting τ=0 (3.D.iv.3).

$$\dot{x}(t) = J \cdot x(t) \quad J = J_i + J_d \quad J = \begin{bmatrix} 0 & 1 \\ -3.27 & -2.72 \end{bmatrix} \qquad (3.D.iv.3)$$

The eigenvalues of matrix J are $\{-1.36 \pm 1.20\,j\}$.

Clearly the system was stable when time delay τ=0 since the eigenvalues of J had negative real components. However, the system may not be stable for all time delays. We attempted to find the maximum delay the system can tolerate. The Laplace transform of equation 3.D.iv.1 is:

$$sX = J_i \cdot X + J_d \cdot X \cdot e^{-s\cdot\tau} \qquad (3.D.iv.4)$$

From equation 3.D.iv.3 it is clear that when τ=0, the poles are in the left complex plane. We therefore wished to find the time delay in which the Laplace variable s has no real component (s=jω), placing it on the imaginary axis. There may be multiple solutions to this problem that yield multiple time delays. The delay margin of the system is the smallest time delay solution, representing the first pole to cross the imaginary axis.

$$s \to j\omega \quad j\omega \cdot X = J_i \cdot X + J_d \cdot X \cdot e^{-j\omega\tau} \qquad (3.D.iv.5)$$

Manipulating equation (3.D.iv.5) into a generalized eigenvalue problem we have:

$$\left(j\omega \cdot I - J_i\right) \cdot X = \lambda \cdot J_d \cdot X \quad \lambda = e^{-j\omega\tau} = e^{-j\theta} \qquad \text{(3.D.iv.6)}$$

A frequency sweep was then performed on the variable ω to find solutions to this problem 3.D.iv.6. This sweep involved testing frequencies from 0 to some maximum ω_{max} for instances in which the generalized eigenvalue λ had a magnitude of unity. Whenever the magnitude of λ was not unity, it was clear that it did not have the form of $e^{-j\omega\tau}$ and was therefore not a solution to 3.D.iv.6. Therefore every frequency for which $|\lambda|=1$ *was* a solution to the problem (Figure 3.D.2). This indicates an imaginary axis crossing of a pole in the system in equation 3.D.iv.4. The corresponding time delay τ at which this crossing occurred was the ratio of the phase of λ to the frequency at which it had a magnitude of unity. This is stated mathematically in equation 3.D.iv.7. The computational complexity of this problem was limited by the fact a ω_{max} exists, bounding the range of frequencies that needed to be tested (see section 3.C.iii.c).

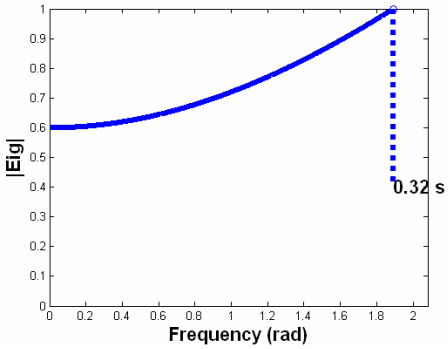


Figure 3.D.2 Locii plot of eigenvalue magnitude as it changes with frequency. Note only λ_1 is shown as λ_2 magnitude is infinite.

$$\omega = [0, \omega_{max}) \quad \forall \omega_i \ni \lambda\left(\left(j\omega_i \cdot I - J_i\right), J_r\right) = e^{-j\theta_i} \quad \tau_i = \frac{\theta_i}{\omega_i} \qquad \text{(3.D.iv.7)}$$

The frequency sweep yielded the result illustrated in Figure 3.D.1. Only a single crossing was found at a frequency of 1.894 radians. At the frequency of 1.894 radians, the generalized eigenvalue problem reduced to the following:

$$\begin{bmatrix} 1.894 \cdot j & -1 \\ -4.88 & 1.09 + 1.894 \cdot j \end{bmatrix} \cdot X = \lambda \begin{bmatrix} 0 & 0 \\ -8.15 & -1.63 \end{bmatrix} \cdot X \qquad \text{(3.D.iv.8)}$$

with eigenvalues: $\lambda_1 = 0.825 - 0.566j = e^{-j \cdot 0.6013} \quad \lambda_2 = -\infty$

and delay margin: $\tau = \frac{0.6013}{1.894} = 0.3175s$

v. Simulation Results

Once a delay margin was found, we wished to test this result with the full nonlinear dynamic system. If the method was successful, the nonlinear system would be stable when

simulated with time delays of less than the delay margin (0.3175), and unstable when simulated with delays larger than the delay margin.

To quantify the stability of the system, we calculated the distance of the normalized state vector x(t) from the equilibrium position $\bar{x}=[\theta_p(t)=0,\ \dot{\theta}_p(t)=0]$ as a function of time (D_{ss} from equation 3.D.v.1). Angular position and velocity were scaled by their own mean to normalize them. If they were not scaled, the velocity amplitude would have been greater than the position amplitude (by approximately the oscillatory frequency) and the distance D_{ss} would therefore have appeared oscillatory (Figure 3.D.3).

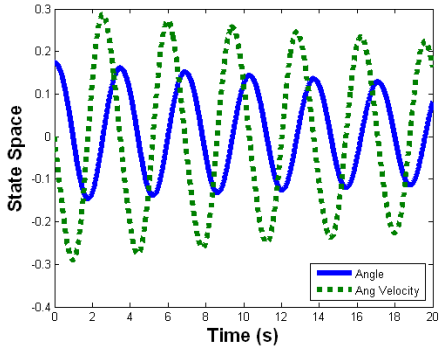


Figure 3.D.3 Unscaled angular position and velocity in radians plotted against time (Trial 2)

If distance D_{ss} had decreased with time the system was stable. Similarly if D_{ss} increased with time the system was unstable. Therefore the slope of the distance from equilibrium indicated if the system was stable. A negative slope indicated stability, while a positive slope indicated instability. System integrations began with $\theta_p=10^o$, thus simulating the system with a step perturbation.

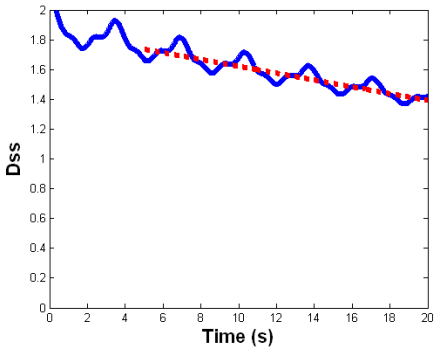


Figure 3.D.4 D_{ss} plotted vs. time (Trial 2)

Table 3.D.1

Delay margin vs. State space slope

Trial	% Delay Margin	State Space Slope
1	50%	-0.50
2	99%	-0.022
3	100%	0.0148
4	101%	0.035
5	150%	1.45

$$D_{ss} = \sqrt{\left(\frac{\theta_p(t)}{mean(\theta_p(t))}\right)^2 + \left(\frac{\dot{\theta}_p(t)}{mean(\dot{\theta}_p(t))}\right)^2} \qquad (3.D.v.1)$$

The system exhibited a positive slope for delays larger than the delay margin, and a negative slope for delays smaller than the delay margin (Table 3.D.1). From these simulation results it was clear that the fully nonlinear system was stable for time delay of less than the delay margin, and was unstable for time delays larger than the delay margin. We therefore concluded that the linearization and assumptions did not seem to aversely affect results.

CHAPTER 4
Linear Time Delay Methods and Stability of the Reflexive Human Spine

To Be Submitted: IEEE Transactions on Neural Systems and Rehabilitation Engineering

Linear Time Delay Methods and Stability of the Reflexive Human Spine

Timothy C. Franklin, MS.
Kevin P. Granata, Ph.D.
Scott L. Hendricks, Ph.D.

Musculoskeletal Biomechanics Laboratories
Department of Engineering Science & Mechanics
School of Biomedical Engineering and Science
Virginia Polytechnic Institute & State University
219 Norris Hall (0219)
Blacksburg, VA 24061

Address all correspondence to: K.P. Granata, Ph.D.
Musculoskeletal Biomechanics Laboratories
Department of Engineering Science & Mechanics
School of Biomedical Engineering and Science
Virginia Polytechnic Institute & State University
219 Norris Hall (0219)
Blacksburg, VA 24061
Phone: (540) 231-5316
FAX: (540) 231-4547
Granata@VT.edu

Keywords: Low-Back, Spine, Linear, Stability, Reflex

Word Count: 3883

Abstract

Linear time delayed stability methods were applied to a biomechanical model of the human musculoskeletal spine. These were used to investigate effects of reflex gain and reflex delay on stability. The equations of motion represented a dynamic 18 DoF rigid body model with time-delayed reflexes. Muscle recruitment was identified be minimizeing metabolic power with constraints of equilibrium and linear stability of the model with zero time delay. The recruitment pattern and associated muscle forces were used to find the delay margin, i.e. the maximum reflex delay for which the system was stable. Results demonstrated that intrinsic stiffness necessary for stability declined with increased proportional reflex gain. The maximum acceptable proportional reflex gain was limited by reflex delay, i.e. long reflex delay requires smaller maximum reflex gain to avoid instability. As differential reflex gain increased the delay margin also showed a small increase in acceptable reflex dealy. However, differential reflex gain with values near the magnitude of intrinsic damping caused the delay margin to approach zero. Results were verified through simulations of the fully nonlinear time delayed system. The linear methods accurately found the delay margin below which the nonlinear system was asymptotically stable. These methods may aid future investigations into the role of reflexes in musculoskeletal stability.

A. Introduction

The role of reflexes in the stability of the human spine is not well understood. Previous studies investigating spinal stability incorporated both experimental measurements and theoretical models. However those studies neglected the role of reflexes. We are aware of no existing models to investigate how these reflexes may affect the stability of the spine.

Previous models suggest that intrinsic muscle stiffness due to steady state contraction may be sufficient to stabilize the spine. Several rigid body spine models have been implemented to investigate this problem (Bergmark 1989, Gardner-Morse *et al* 1995, Cholewicki *et al* 1996). Intrinsic muscle stiffness K was modeled as K=q*F/L, for muscle stiffness gain q, muscle force F, and muscle length L (Bergmark 1989). It has been shown that stability may be possible with a value of q as low as 4.5 (Gardner-Morse *et al* 1995). Others conclude that q must be at least 8.4 (Kiefer *et al* 1998). The disparity in predicted minimum values is a result of differences in modeled anatomy and design parameters. While all of these models were able to achieve stability in the absence of reflexes, it is likely that the presence of reflexes offers advantages over intrinsic stiffness alone. For example, it has been suggested that reflexes allow the spine to maintain stability with reduced metabolic expenditure (Hogan 1984).

Reflex contributes to spinal stability in the manner of feedback control. Kearney *et al* (1997) quantified the stabilizing effects of reflexes in a biomechanical investigation of ankle stiffness. Nonlinear systems identification techniques demonstrated that reflexes were responsible for a significant fraction of the stiffness in the system. Empirical measurements of the torso by Moorhouse *et al* (2006) used similar methods to quantify reflex contribution to human torso dynamics. Results showed that 42% of torso stiffness was attributable to muscle reflexes. In fact, many of the experimental subjects were unstable in the absence of reflexes. These studies provided clear evidence that reflexes play an important role in stability.

The contribution of reflexes for control of spinal stability and injury is supported by clinical evidence. Recent measurements have shown significant correlations between abnormal reflex response and low back pain. Radebold *et al* (2001) observed a significant difference in reflex response and associated stabilizing control in patients with low back pain versus age-matched asymptomatic control subjects. Others conclude that reflex characteristics can be used to correctly classify subjects into low-back pain and asymptomatic groups with 83% accuracy (Reeves *et al* 2005). It has been shown by Van Dieen *et al* (2003), that patients with low back pain had higher levels of co-contraction than their healthy counterparts. The increased recruitment of co-contraction may result in increased compressive forces in the spine, commonly considered a precursor for low back pain (Norman *et al* 1998). The authors concluded that this recruitment strategy in patients with low-back pain may be used to compensate for limitations elsewhere in the stabilizing control system such as impaired reflex dynamics.

We are aware of no existing biomechanical models of spinal stability that incorporate the effects of neuromuscular reflex. The absence of reflex response in these models is due, in part, to the challenge of investigating the stability of dynamic systems with feedback delay. Therefore, the goal of this study was to implement a computational model that would aid in understanding the role of reflexes in stabilizing control of the spine. We implemented a biodynamic model to represented five lumbar vertebrae and a thorax, with 90 active muscles. Muscles dynamics included viscoelastic intrinsic stiffness modulated by stiffness gain q as in previous studies. However muscle dynamics were expanded to include reflex activations that were proportional to time-delayed muscle stretch and stretch rate similar to a delayed proportional-differential feedback controller. Linear time-delayed systems analyses were used to determine stability of the delayed system (Chen 1995). This work allowed any system with instantaneous (intrinsic) and delayed (reflexive) components to be analyzed for stability about equilibrium. For a prescribed biomechanical configuration the analysis solved for the time delay at the threshold between stability and instability in the system. Results were validated with forward-dynamic simulations of the fully nonlinear time-delayed dynamics of the model.

B. Methods

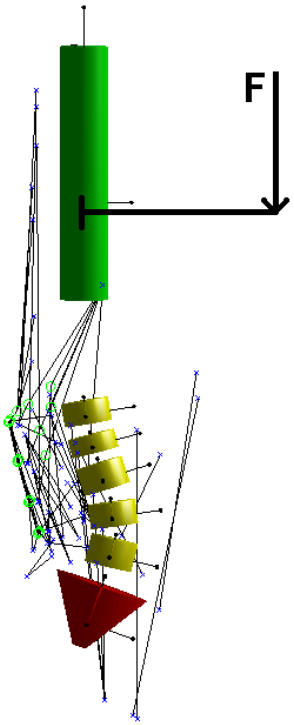


Figure 4.1. Model Diagram

i. Model Anatomy and Geometry

The model represented a three dimensional (3D) 18 degree of freedom (DoF) spine consisting of the five rigid lumbar vertebrae and a rigid thoracic segment. Passive intervertebral disc reaction moments were incorporated as 3D lumped parameter viscoelastic elements between neighboring vertebral bodies. The disc model was assigned stiffness of 50Nm/rad calculated from Stokes et al. (2002). Intervertebral damping was assigned 0.5Nms/rad, 2 orders of magnitude less than the stiffness (Kasra et al. 1992, Izambert et al. 2003). A 3D anatomy of 90 muscles was used, with physiologic properties and attachment points on the pelvis, vertebra and thoracic body described in the literature (Cholewicki and McGill 1996, Stokes and Gardner-Morse 1999). For each muscle the anatomy described the origin, insertion, and intermediate via locations in coordinate frames fixed to each vertebral body. The via points allow representation of muscle curvature as they to wrap around virtual biological structures. This provided

muscle lengths, stretch rates, muscle unit force vectors, as well as normal contact forces from the via points. External loads to simulate a lifting task were applied vertically at a location 20cm anterior to the trunk at the T4 level (Figure 1).

Equations of motion were calculated with the Lagrange derivative. Muscle forces, intervertebral disc moments, and conservative external loads were considered generalized forces. Therefore the potential energy function V contained only energy due to gravity

$$V = \sum_{j=1}^{6} m_j \cdot \vec{g} \cdot \vec{P}_j \tag{1}$$

m_j was the mass of the j[th] body, $\vec{g}$ was the gravity vector, and $\vec{P}_j$ was the center of mass of the j[th] body.

Each body was assigned the inertial properties of a cylinder with an elliptical cross sectional area (Lui et al 1971). The corresponding inertia matrices were used in the kinetic energy function T

$$T = \sum_{j=1}^{6} \left(\frac{1}{2} \cdot m_j \cdot \dot{\vec{P}}_j \bullet \dot{\vec{P}}_j + \frac{1}{2} {}^{N}\vec{\omega}_j^{B} \bullet I_j \bullet {}^{N}\vec{\omega}_j^{B} \right) \tag{2}$$

$\dot{\vec{P}}_j$ was the velocity vector of the center of mass of body j, ${}^{N}\vec{\omega}_j^{B}$ was the inertial frame angular velocity vector of the body j in body fixed coordinates, and I_j was the inertia matrix of body j.

External forces, muscle forces and intervertebral disc moments were applied by means of the generalized force vector $\vec{Q}_v$. Generalized forces for all 18 DoF were found numerically for each evaluation of acceleration. The contribution of the k[th] force acting on body j (F_{jk}) on the n[th] generalized dimension is shown below, where r_{jk} is the vector between the point of application and the vertebral body origin. Muscle forces were applied to their origin, insertion, and intermediate via point locations. Intervertebral disc moments were applied to the bodies as pure moments through the second term in equation 3.

$$\mathrm{Q}_{v_n} = \sum_{j=1}^{6} \left(\left(\sum_{k=1}^{N_j} F_{jk} \right) \cdot \frac{\partial v_j}{\partial \dot{q}_n} + \left(\sum_{k=1}^{N_j} r_{jk} \times F_{jk} \right) \cdot \frac{\partial \omega_j}{\partial \dot{q}_n} \right) \tag{3}$$

N_j represented the number of loads applied to body j, v_j was the velocity vector of the body fixed origin of body j, ω_j was the angular velocity vector of body j, and $\dot{q}_n$ was the time derivative of the nth DoF.

The forward dynamic system was represented in differential matrix form with respect to the vector of kinematic angles $\vec{\theta}$

$$\ddot{\vec{\theta}} = \mathrm{M_m}\left(\vec{\theta}\right)^{-1} \cdot \left\{ \vec{\mathrm{Q}}_v\left(\vec{\theta}, \dot{\vec{\theta}}, \vec{\alpha}_m\right) - \vec{\mathrm{C}}_v\left(\vec{\theta}, \dot{\vec{\theta}}\right) - \vec{\mathrm{G}}_v\left(\vec{\theta}\right) \right\} \tag{4}$$

where $\dot{\vec{\theta}}$ and $\ddot{\vec{\theta}}$ are vectors of rigid body angular velocity and accelerations. $\mathbf{M}_\mathrm{m}$ represented an 18x18 inertial mass matrix, $\vec{G}_v$ was a vector of gravitational forces, $\vec{C}_v$ was a vector of velocity related forces, and $\vec{Q}_v$ was a vector of generalized forces influenced by muscle activation vector $\vec{\alpha}_m$.

ii. Muscle Force and Reflex Model

Tensile muscle force was similar to a Hill-type muscle model, which contained a contractile element composed of a force generator F_m, in parallel with a viscoelastic element was composed of a spring K_m and a damper B_m (Zajac 1989).

$$f_m\left(L_m(t)\right) = F_m + K_m \cdot \left(L_m(t) - L_{mo}\right) + B_m \cdot \dot{L}_m(t) \tag{5}$$

Muscle force f_m was the sum of steady-state contractile force F_m and contributions from viscoelastic strain. Therefore, f_m was a function of muscle length $L_m(t)$, equilibrium muscle length L_{mo}, and muscle stretch rate $\dot{L}_m(t)$. Steady-state contractile force F_m was estimated as the maximum isometric muscle force f_{mo} modulated by myoelectric muscle activation α_m. Activation varied from $\alpha_m = 0$ for off to 1 for fully activated muscle.

$$F_m = f_{m_o} \cdot \alpha_m,\ K_m = q^{F_m}\!\!\diagup_{L_{m0}},\ B_m = b^{F_m}\!\!\diagup_{L_{m0}} \tag{6}$$

Intrinsic stiffness K_m of each muscle m=1:90 was represented by methods of Bergmark (1989). Stiffness of every muscle was modulated by the same dimensionless stiffness gain q. Intrinsic muscle damping B_m was similar to the intrinsic stiffness, wherein the

muscle damping of every muscle was modulated by the same damping gain *b,* with units of time. Substituting the assigned force, stiffness, and damping from equation 6 into the muscle model in equation 5 resulted in the muscle model in equation 7.

$$f_m(L_m(t)) = f_{mo} \cdot \alpha_m(t) \cdot \left(q \cdot \left(\frac{L_m(t) - L_{mo}}{L_{mo}} \right) + b \cdot \frac{\dot{L}_m(t)}{L_{mo}} + 1 \right) \tag{7}$$

Maximum muscle force f_{mo} was assigned a value of maximum muscle stress 46N/cm^2 multiplied by muscle cross sectional area (Gardner-Morse et al 1995). Crisco and Panjabi (1991) performed a literature review and estimated the value of q to be between 0.5 and 40 with an average value of 10. In this study the average q value of 10 was used. Muscle activation α_m was the sum of a steady state component α_{mo} and a time-delayed reflex component α_{mr}. Reflex was proportional to time-delayed muscle stretch modulated by proportional reflex gain G_p, and delayed stretch rate modulated by differential reflex gain G_d.

$$\begin{aligned} &\alpha_m(\alpha_{mo}, L_m(t)) = \alpha_{mo} + \alpha_{mr}(\alpha_{mo}, L_m(t)) \\ &\textit{where} \quad \alpha_{mr}(\alpha_{mo}, L_m(t)) = \alpha_{mo} \cdot \left(G_p \cdot \left(\frac{L_m(t-\tau) - L_{mo}}{L_{mo}} \right) + G_d \cdot \frac{\dot{L}_m(t-\tau)}{L_{mo}} \right) \end{aligned} \tag{8}$$

iii. Linear Stability Methods

Linear stability methods were used to optimize steady-state muscle activation, solve for stability, and to find the delay margin of the system. A linearized version of the system dynamics may be represented with two Jacobians. These include an instantaneous Jacobian J_i and a delayed Jacobian J_d, for a state vector $x(t)=[\vec{\theta}(t), \dot{\vec{\theta}}(t)]^t$.

$$\dot{x}(t) = J_i \cdot x(t) + J_d \cdot x(t-\tau) \tag{9}$$

To build these Jacobians it was necessary to separate the system dynamics into instantaneous and delayed components. Instantaneous dynamics depend on only the instantaneous state x(t), while the delayed dynamics depend on only the delayed state x(t-τ). By substituting equation 8 into equation 7 and eliminating high order terms, the muscle force was separated into instant force f_{mi}, and delayed force f_{md}.

$$f_m(\alpha_{mo}, L_m(t)) \approx f_{mi}(\alpha_{mo}, L_m(t)) + f_{md}(\alpha_{mo}, L_m(t-\tau)) \tag{10}$$

Carrying the separate muscle force quantities through the generalized force equation resulted in instantaneous generalized forces $\vec{Q}_{vi}$, and delayed generalized forces $\vec{Q}_{vd}$.

$$\vec{Q}_v = \vec{Q}_{vi} + \vec{Q}_{vd} \tag{11}$$

Substituting equation (11) into the generalized forces from equation (4) resulted in acceleration due to the instantaneous system $\ddot{\vec{\theta}}_i(t)$, and acceleration due to the delayed system $\ddot{\vec{\theta}}_d(t)$.

$$\ddot{\vec{\theta}}(t) = M_m^{-1}\left(\vec{Q}_{vi} - \vec{G}_v - \vec{C}_v\right) + M_m^{-1}\left(\vec{Q}_{vd}\right) = \ddot{\vec{\theta}}_i(t) + \ddot{\vec{\theta}}_d(t) \tag{12}$$

The instantaneous Jacobian contained partial derivatives of instantaneous acceleration $\ddot{\vec{\theta}}_i(t)$ with respect to the instantaneous state x(t). Therefore this Jacobian included contributions from external load, gravity, steady-state muscle contraction force, stiffness and damping, and intervertebral disc stiffness and damping. The delayed Jacobian contained changes in accelerations due to the delayed system state x(t-τ), i.e. delayed reflex. While muscle force was dependant on only the delayed state, the muscle moment arms (and inertia matrix) depended on the instantaneous state. However, the delayed Jacobian J_d combined delayed reflex response with instantaneous kinematic states, i.e. muscle moment arms. The oscillatory period of the spine is approximately 1s, while a typical physiological reflex delay is approximately 60ms (Reeves *et al* 2005). Therefore by assuming the total system oscillatory period was large compared with reflex delay τ, the phase margin was deemed negligible and these instantaneous states were converted to delayed states to calculate J_d.

The delay margin is the time delay at which instability occurs. Chen (1995) developed a frequency sweeping method for finding the delay margin in the Laplace domain.

$$sX = J_i \cdot X + J_d \cdot X \cdot e^{-s\cdot\tau} \tag{13}$$

The case in which time delay is zero is equivalent to an eigenvalue problem, where I is the identity matrix.

$$\left(s \cdot I - \left(J_i + J_d\right)\right) \cdot X = 0 \tag{14}$$

Since we require the system to be stable at zero time delay, the poles of equation 14 are located in the left half of the complex s-plane. Therefore finding the delay margin is equivalent to solving for the delay at which one of the poles crosses the imaginary axis. On this axis the real component of the eigenvalue is zero. Hence equation 13 may be expressed as a generalized eigenvalue problem with the real component of the Laplace variable set to zero.

$$\begin{aligned} &s \to j\omega \\ &\left(j\omega \cdot I - J_i\right) \cdot X = \lambda \cdot J_d \cdot X \quad \lambda = e^{-j\omega\tau} = e^{-j\theta} \end{aligned} \tag{15}$$

where θ is the phase of the eigenvalue. The frequency domain must be swept for solutions such that:

$$\omega = [0, \omega_{\max}) \quad \forall \omega_i \ni \lambda\left(\left(j\omega_i \cdot I - J_i\right), J_r\right) = e^{-j\theta_i} \quad \tau_i = \frac{\theta_i}{\omega_i} \tag{16}$$

This frequency sweep results in a set of solutions, each corresponding to a time delay τ_i. The minimum τ_i is the delay margin of the system. An example of a frequency sweep is illustrated in Figure 2. In this figure, magnitudes of eigenvalues were plotted as they changed in the frequency domain. The calculated time delays were printed at locations for which eigenvalue magnitudes crossed unity.

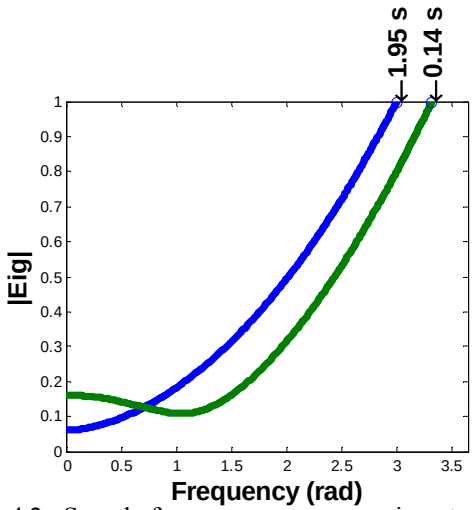


Figure 4.2. Sample frequency sweep experiment. Generalized eigenvalue magnitude loci are plotted as they migrate with increasing frequency. Delays corresponding to unit crossings are drawn

iv. Muscle activation

Constructing the system Jacobians required a priori vector of steady state muscle activation $\vec{\alpha}_{mo}$. This was achieved by using the optimization algorithm fmincon (Mathworks) to find the set of steady state activations that minimized metabolic power (Anderson, 1999) and resulted in Jacobians such that the zero time delay system was stable (equation 14). Metabolic power was the total sum of muscle energy expenditure due to maintenance of muscle force. For simplicity an equal distribution of slow and fast twitch muscle fibers were assumed when calculating power (equation 17), where m_i was mass of muscle i, and $\vec{\alpha}_{mo_i}$ was steady-state activation of muscle i.

$$\mathrm{P} = \sum_{i=1}^{90} m_i \cdot \left(\frac{74}{2} \cdot Sin\left(\frac{\alpha_{mo_i} \cdot \pi}{2} \right) + \frac{111}{2} \cdot \left(1 - Cos\left(\frac{\alpha_{mo_i} \cdot \pi}{2} \right) \right) \right) \qquad (17)$$

Each solution was repeated 3 times with a randomized initial activation vector. For every solution a consistent minimum was found. The resulting instantaneous and delayed Jacobians were then used to determine the delay margin at the prescribed set of $\vec{\alpha}_{mo}$. Co-contraction necessary to achieve stability was quantified using methods described elsewhere (Granata et al 2005). Briefly, co-contraction was calculated as the percentage of total muscle force relative to that required for equilibrium. Thus, zero co-contraction corresponded to the minimal total muscle force required to maintain equilibrium while ignoring stability.

v. Nonlinear verification

The methods used to find the muscle activation levels and the delay margin were developed with the linearized system. However in developing the delayed Jacobian J_d an assumption was made regarding the relative phase margin between the oscillatory period of the system and the muscle reflex delay. In light of this assumption it was desirable to test the results of the linear analysis using the fully nonlinear time delayed system. Therefore, the nonlinear forward dynamics of the spine were simulated with time delays slightly below and above the delay margin.

Six systems were simulated with randomly chosen parameters for external load, intrinsic damping gain, proportional reflex gain, and differential reflex gain. Random

parameters were bounded into the following intervals: External load (0-200), muscle damping gain (0.2-20), proportional reflex gain (0-100), and differential reflex gain (0-10). Stability was calculated by observing the state space expansion following a kinematic disturbance of 0.01^{o} extension of the thoracic body. All systems were simulated for 5 seconds, which was verified to be greater than 10 times the delay margin in all systems. The state space about equilibrium ($\bar{x}$) was normalized (Figure 3) by dividing angular positions by the mean amplitude of all angular positions, and angular velocities by the mean angular velocity amplitude. The distance D_{ss} of the normalized state space from equilibrium was found (equation 18). The slope of D_{ss} versus time was used to determine if the state space was expanding or contracting toward equilibrium (Figure 4).

$$D_{ss} = \sqrt{\sum_{i=1}^{18}\left(\left(\frac{\theta_i(t)-\bar{\theta}_i}{mean\left(\bar{\theta}(t)-\bar{\theta}\right)}\right)^2+\left(\frac{\dot{\theta}_i(t)-\dot{\bar{\theta}}_i}{mean\left(\dot{\bar{\theta}}(t)-\dot{\bar{\theta}}\right)}\right)^2\right)} \qquad (18)$$

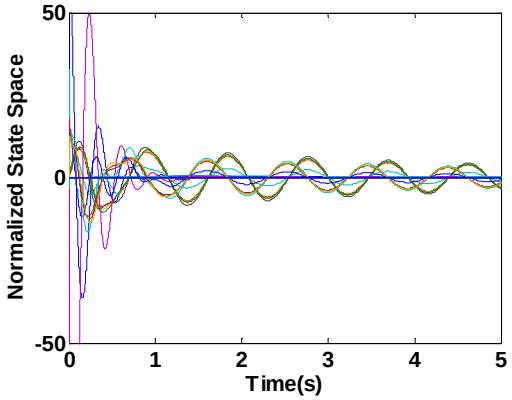


Figure 4.3. Normalized state space vs. time following perturbation

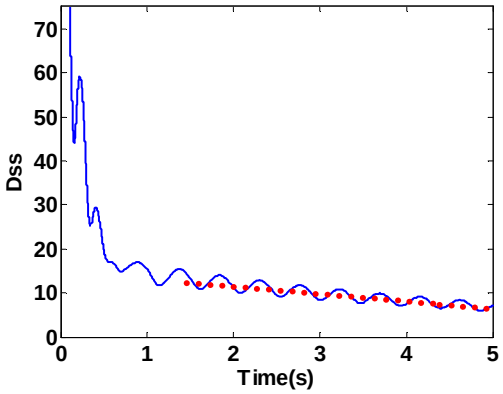


Figure 4.4. Normalized state space distance from equilibrium Dss vs. time

C. Results

Reflexes were found to affect the stability of the spine. Increasing the proportional reflex gain G_p provided stability with reduced intrinsic stiffness. This reduction in intrinsic stiffness was manifest in decreased antagonistic muscle co-contraction (Figure 5). Since co-contracted muscle has an associated metabolic cost, increased proportional reflex gain G_p allowed the optimization algorithm to stabilize the model with reduced

metabolic power consumption (Figure 6). Neither differential reflex gain G_d nor the intrinsic damping gain b influenced the optimization results, co-contaction, or metabolic power consumption of the static lifting task. Reflexes were necessary to maintain stability in some conditions. For example, the model was unstable for a 200N load when proportional reflex gain G_p was less than 10. While increasing the intrinsic stiffness gain q would allow this loading condition to stabilize in the absence of reflex, another loading or posture condition could be found that would require reflexes for stability. Hence, reflexes may often be a necessary condition for spinal stability.

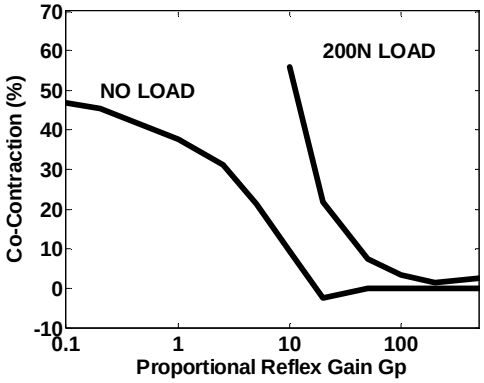


Figure 4.5. Co-contraction vs. proportional reflex gain for both load conditions.

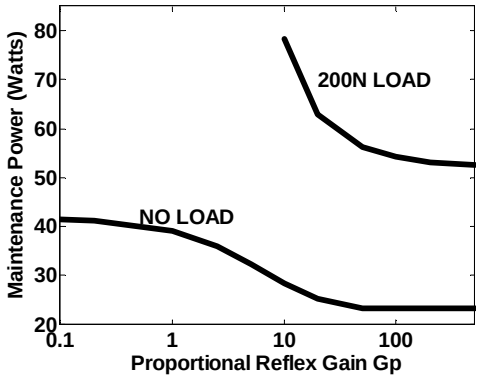


Figure 4.6. Metabolic power vs. proportional reflex gain for both load conditions.

Proportional reflex gain G_p was inversely related to the delay margin (Figure 7). This relationship demonstrated that the system was stable for only small time delays when utilizing large values of reflex gain G_p. The intrinsic damping gain b caused shifts in this relationship. Increased damping caused the system to be stable for larger time delays. Small values of differential gain G_d were shown to provide small increases in the delay margin of the system. However, as these values approached the magnitude of the intrinsic damping coefficient b, the trend was reversed and the delay margin approached zero (Figure 8). External load did not influence the delay margin.

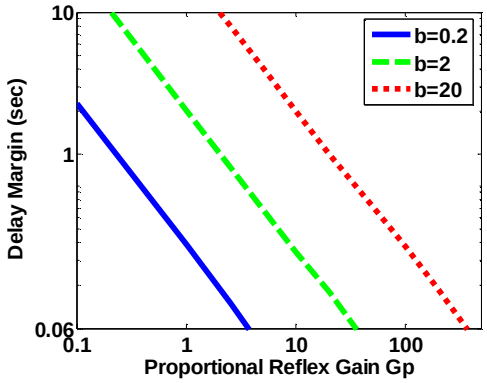


Figure 4.7. Delay margin vs. proportional reflex gain for multiple damping coefficients, terminated at 60ms.

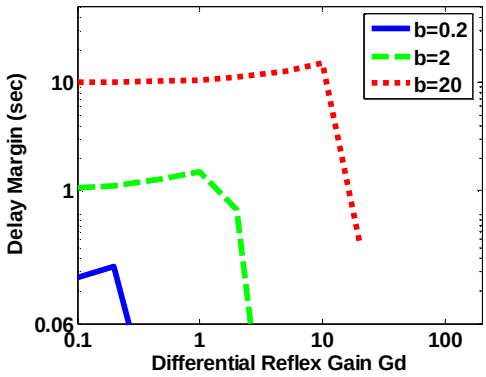


Figure 4.8. Delay margin vs. differential reflex gain for multiple damping coefficients, terminated at 60ms.

The nonlinear simulation confirmed that the linear method successfully predicted the delay margin below which the system remained stable. When reflex delay was 99% of the delay margin the slope of the state space distance from equilibrium, D_{ss}, was negative for all systems, i.e. a stable attractor. This indicated an asymptotically stable system for reflex delays less than the delay margin. Additionally, all of the systems had a positive slope for larger values of simulated delay, i.e. spinal instability. However, the systems varied in the amount of time delay at which this nonlinear system moved from stable to unstable (Figure 9). Therefore, while this method effectively predicted a stable delay margin, it did not predict the upper bound for the nonlinear delay margin. The resultant slopes of the six randomly chosen systems as a function of their respective delay margin are illustrated in Figure 9.

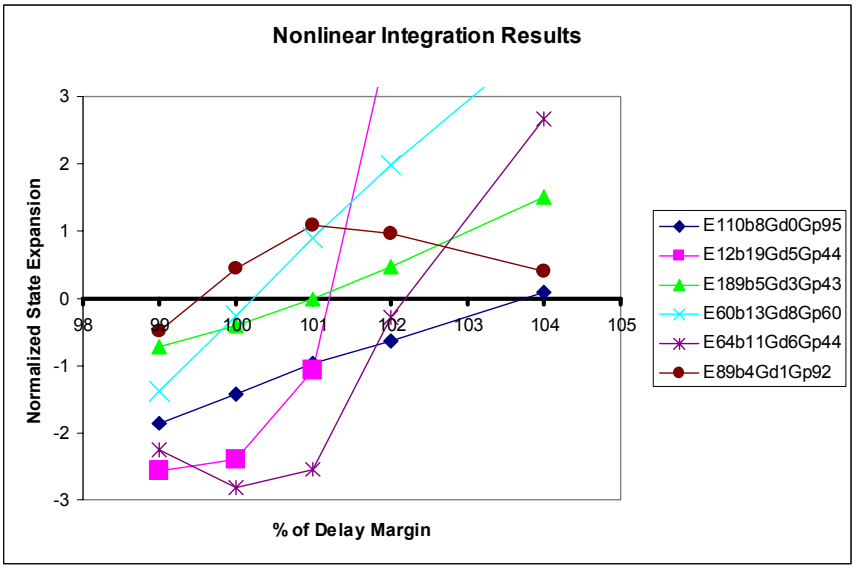


Figure 4.9. State space slopes vs. % of delay margin simulated for six random systems. E=external load, b=damping gain, Gd=differential reflex gain, Gp=proportional reflex gain

D. Discussion

Methods have been demonstrated to aid investigations regarding the role of reflexes in the spine stability. Time-delayed linear analyses accurately predicted a conservative estimate of the delay margin of the nonlinear system. Forward-dynamic, nonlinear integration results supported the analyses. When reflex delay was shorter then the delay margin, these simulations showed the existence of the equilibrium as a state-space attractor. This indicated that the system was asymptotically stable about equilibrium for reflex delays less than the delay margin. This was the expected result and indicated that the linearization assumptions did not negatively affect results. All systems were shown to diverge from equilibrium when simulated with reflex delay larger than the delay margin. The percentage of the delay margins at which the slopes became positive was different for all systems. Since each system had a different damped natural frequency, it is possible that this discrepancy was related to the assumed relative phase between delayed and instantaneous states. Nonetheless, predicted maximum reflex delay from linearized analyses were typically accurate within 4% of the fully nonlinear results.

Proportional reflex gain provided active feedback control for spinal stability. This allowed the system to achieve stability while simultaneously reducing the requirements for open loop intrinsic stiffness. Hence the reflex response reduced the requirements for steady-state co-

contraction and improved metabolic power efficiency. However, reflex delay limited the maximum amount of feedback contribution to effective stiffness (Figure 7). This effect was predicted by Hogan (1984), who suggested the tradeoff for utilizing reflexes or co-contraction for stability was related to their limitations, namely reflex delay and metabolic power. For given intrinsic damping b and reflex delay τ, the maximum proportional reflex gain G_p was the value at which the delay margin was equal to the reflex delay. Any further increase in G_p required a reflex delay larger than the delay margin, suggesting instability.

Intrinsic muscle damping b and differential reflex gain influenced the maximum reflex delay of the system. Intrinsic muscle damping b increased the delay margin in all cases. Greater damping slowed the damped natural frequencies in the system, forcing the system to have a larger oscillatory period. This caused a shift in the delay margin because the slower system was able to tolerate larger values of reflex delay. Differential reflex gain G_d did not strongly contribute to stability of the system. While small values of G_d slightly increased the delay margin of the system, large gains drove the delay margin toward zero. Therefore only small gains positively affected stability of the system, with large gains destabilizing the system for any reflex delay.

The results of this study provide insight into clinical behavior of patients with low back pain, and risk factors for propospective injury. When compared with asymptomatic control subjects, low-back pain patients have: a) larger reflex delay (Reese 2005), b) impaired postural control (Radebold *et al* 2001), and c) increased antagonistic muscle co-contraction (van Dieen *et al* 2003). As our study demonstrated, large physiological reflex delays required small proportional reflex gain to avoid spinal instability. With small reflex gain values, greater intrinsic stiffness was required and hence greater co-contraction was predicted in individuals with long reflex delay. This implies a causal effect between large reflex delay and increased muscle co-contraction observed in patients with low-back pain. In fact, Van Dieen *et al* (2003) concluded that the co-contraction recruitment strategy in patients with low-back pain may be used to compensate for limitations elsewhere in the stabilizing control system such as impaired reflex dynamics. Additionally the model demonstrated that proportional reflex gain may be necessary to stabilize large loads, which require more stiffness than the intrinsic system can

provide. Lifting heavy loads is a recognized risk factor for low-back pain (National Research Counsel, 2001). Individuals with large reflex delays may be without risk under small external loads. However, under heavy external loads the stable delay margin may be brief compared to their physiological reflex delay. This agrees with epidemiologic data wherein reflex delay in healthy individuals accurately predicts prospective risk of low-back injury (Reeves *et al* 2005). Therefore the model also implies a relationship between large reflex delay and instability in the spine.

Results from these simulations should be interpreted in light of the model limitations. The muscle force model ignored nonlinear effects of muscle force-versus-length and force-versus-velocity profiles. While this would not affect the linear system analyses, nonlinear stiffness and damping may have hardening properties that would result in increased stability in the nonlinear system. However, the small intervertebral movement studied here (less than 0.01 degrees) were adequately represented without these nonlinear effects. Reflexes increased in proportion to steady-state recruitment as described in empirical measurements (Matthews 1986, Rogers 2006). However, the transient decay of the reflex response was ignored, as the dynamic response characteristics were considered slow when compared to the reflex delay (Moorhouse 2006). Nonetheless, future research should include the myoelectric dynamics of reflex when investigating stability. All muscles were represented with identical reflex gains. Future studies should include gamma neuromotor drive to represent modulation of reflex gain in individual muscle groups. Finally, the model did not allow the optimization algorithm to modify the equilibrium kinematics. Previous investigators have shown that allowing optimal adjustment of intervertebral angles at equilibrium may reduce the steady-state muscle forces needed for stability by allowing the passive system to support some of the external load (Shirazi-Adl *et al* 2002).

These simulations demonstrated the importance of reflexes in spinal stability. Specifically, they provided insight into the relative roles of proportional and differential reflex feedback. The linear time delayed techniques illustrated how reflex delay affected the stability of the model by limiting the maximum reflex gains. Future analyses of spinal stability should further investigate the contributions of neuromuscular reflex.

E. Acknowledgement

This research was supported by a grant R01 AR46111 from NIAMS of the National Institutes of Health.

F. Reference List

Anderson,F.C. (1999) A dynamic optimization solution for a complete cycle of normal gait. Ph.D. Univ Texas, Austin.

Bergmark A. (1989) Stability of the lumbar spine: A study in mechanical engineering. *Acta Orthop.Scand.Suppl.* **230,** 1-54.

Chen,J. (1995) On Computing the Maximal Delay Intervals for Stability of Linear Delay Systems. *Ieee Transactions on Automatic Control* **40,** 1087-1093.

Cholewicki,J. and McGill,S.M. (1996) Mechanical stability on the in vivo lumbar spine: Implications for injury and chronic low back pain. *Clin.Biomech.* **11,** 1-15.

Crisco,J.J. and Panjabi,M.M. (1991) The intersegmental and multisegmental muscles of the lumbar spine: A biomechanical model comparing lateral stabilizing potential. *Spine* **16,** 793-799.

Gardner-Morse,M.G., Stokes,I.A.F., and Laible,J.P. (1995) Role of muscles in lumbar stability in maximum extension efforts. *J.Orthop.Res.* **13,** 802-808.

Granata,K.P., Lee,P.E., and Franklin,T.C. (2005) Co-contraction recruitment and spinal load during isometric trunk flexion and extension. *Clin Biomech.(Bristol., Avon.)* **20,** 1029-1037.

Izambert,O., Mitton,D., Thourot,M., and Lavaste,F. (2003) Dynamic stiffness and damping of human intervertebral disc using axial oscillatory displacement under a free mass system. *Eur.Spine J.* **12,** 562-566.

Kasra,M., Shirazi-Adl,A., and Drouin,G. (1992) Dynamics of human lumbar intervertebral joints. Experimental and finite-element investigations. *Spine* **17,** 93-102.

Kearney R.E. and Stein R.B. (1997) Identification of intrinsic and reflex contributions to human ankle stiffness dynamics. *IEEE Trans.Biomed.Eng* **44,** 493-504.

Kiefer, A., A. Shirazi-Adl, et al. (1998). "Synergy of the human spine in neutral postures." Eur Spine J **7**: 471-479.

Liu, Y. K., J. M. Laborde, et al. (1971). "Inertial properties of a segmented cadaver trunk: their implications in acceleration injuries." Aerosp Med **42**(6): 650-7.

Matthews,P.B.C. (1986) Observations on the automatic compensation of reflex gain on varying the pre-existing level of motor discharge in man. J.Physiol. 374, 73-90.

Moorhouse,K.M. (2005) Role of intrinsic and Reflexive Dynamics in the Control of Spinal Stability. Ph.D. Virginia Tech.

National Research Council. Musculoskeletal Disorders and the Workplace. Washington DC: National Academy Press, 2001.

Norman, R., R. Wells, et al. (1998). "A comparison of peak vs cumulative physical work exposure risk factors for the reporting of low back pain in the automotive industry." Clin Biomech (Bristol, Avon) 13(8): 561-573.

Radebold,A., Cholewicki,J., Polzhofer,G.A., and Green,T.P. (2001) Impaired postural control of the lumbar spine is associated with delayed muscle response times in patients with chronic idiopathic low back pain. *Spine* **26,** 724-730.

Reeves, N., J. Cholewicki, et al. (2005). "Muscle reflex classification of low-back pain." Journal of Electromyography and Kinesiology **15**: 53-60.

Rogers, E. L. and K. P. Granata (2006). "Disturbed paraspinal reflex following prolonged flexion-relaxation and recovery." Spine **31**(7): 839-45.

Shirazi-Adl, A., S. Sadouk, et al. (2002). "Muscle force evaluation and the role of posture in human lumbar spine under compression." Eur Spine J 11: 519-526.

Stokes,I.A. and Gardner-Morse,M. (1999) Quantitative anatomy of the lumbar musculature. *J.Biomech.* **32,** 311-316.

Stokes,I.A.F., Gardner-Morse,M.G., Churchill,D., and Laible,J.P. (2002) Measurement of spinal motion segment stiffness matrix. *J.Biomechanics* **32,** 517-521.

Van Dieen,J.H., Cholewicki,J., and Radebold,A. (2003) Trunk muscle recruitment patterns in patients with low back pain enhance the stability of the lumbar spine. *Spine* **28,** 834-841.

Zajac, F. E. (1989). "Muscle and Tendon: Properties, Models, Scaling, and Application to Biomechanics and Motor Control." Critical Reviews in Biomedical Engineering **17**(4): 360-410.

CHAPTER 5
Reflex Gain and Delay on Spinal Stability:
Forward Dynamics Simulation of Spinal Mechanics

Submitted to: Journal of Biomechanics *27 March 2006*

Role of Reflex Gain and Delay on Spinal Stability: Forward Dynamics Simulation of Spinal Mechanics.

Timothy C. Franklin, MS.
Kevin P. Granata, Ph.D.

Musculoskeletal Biomechanics Laboratories
Department of Engineering Science & Mechanics
School of Biomedical Engineering and Science
Virginia Polytechnic Institute & State University
219 Norris Hall (0219)
Blacksburg, VA 24061

Address all correspondence to: K.P. Granata, Ph.D.
Musculoskeletal Biomechanics Laboratories
Department of Engineering Science & Mechanics
School of Biomedical Engineering and Science
Virginia Polytechnic Institute & State University
219 Norris Hall (0219)
Blacksburg, VA 24061
Phone: (540) 231-5316
FAX: (540) 231-4547
Granata@VT.edu

Keywords: Low-Back, Spine, Stability, Reflex

Word Count: 3125

Abstract

The goal of this study was to investigate the role of reflex and reflex time delay in muscle recruitment and spinal stability. A dynamic biomechanical model of the musculoskeletal spine with reflex response was implemented to investigate the relationship between reflex gain, co-contraction, and stability in the spine. The first aim of the study was to investigate how reflex gain affected the levels of co-contraction predicted in the model. It was found that reflexes allowed the model to stabilize with less antagonistic co-contraction and hence lower metabolic power than when limited to intrinsic stiffness alone. In fact, without reflexes there was no feasible recruitment pattern that could maintain spinal stability when the torso was loaded with 200 N external load. Reflex delay is manifest in the paraspinal muscles and represents the time from a perturbation to the onset of reflex activation. The second aim of the study was to investigate the relationship between reflex delay and the maximum tolerable reflex gain. The maximum acceptable upper bound on reflex gain decreased logarithmically with reflex delay. Thus, increased reflex delay and reduced reflex gain requires greater antagonistic co-contraction to maintain spinal stability. The results of this study may help understanding of how patients with retarded reflex delay utilize reflex for stability, and may explain why some patients preferentially recruit more intrinsic stiffness than healthy subjects.

A. Introduction

Intrinsic stiffness and reflex response may both contribute to stability of the spine. This is related to the fact that musculoskeletal stiffness is composed of both intrinsic and reflexive components (Kearney R.E. and Stein R.B., 1997). However, previous investigations of spine stability focused on the stabilizing potential of only the intrinsic components (Gardner-Morse *et al.*, 1995; Cholewicki and McGill, 1996). These studies showed that intrinsic stiffness may provide sufficient stiffness for stabilizing the spine under a variety of loading conditions. Therefore it is possible that reflex is not a necessary component of stability but may be desirable for improved and efficient stabilizing control.

The recruitment of stiffness must change for different loading tasks. There are loads under which large amounts of stiffness are required for stability. For example, a vertical load applied at shoulder level requires much more stabilizing stiffness than the same load applied at a waist level (El Rich and Shirazi-Adl, 2005). To recruit stiffness for stability, intrinsic stiffness may be increased through the co-activation of antagonistic muscles (Lee *et al.*, 2006). However, the need for co-contraction may be attenuated by reflex response. Reflex is defined as a change in muscle activation in response to an external perturbation thereby leading to a change in force. Hence, this reflexive change in force is phenomenologically similar to stiffness and has been called reflex stiffness (Kearney R.E. and Stein R.B., 1997). Cholewicki (Cholewicki *et al.*, 2000) coined the term effective stiffness to define the combined effects of intrinsic muscle stiffness and reflex contributions to spinal dynamics. Reflexes may increase the effective stiffness of the spine beyond that provided by the intrinsic stiffness alone. As reflex is observed experimentally there may be advantages to recruiting reflexes instead of intrinsic stiffness alone (Granata *et al.*, 2004).

One advantage of reflexes is that they may allow stability of the spine with reduced energy expenditure as compared with intrinsic stiffness alone. Muscle activation is associated with metabolic cost that is proportional to the volume and activation level of the muscle (Hatze H., 1977). Therefore while intrinsic stiffness may be increased through steady state co-contraction, this requires increased energy consumption. Recall that reflex on the other hand is a change in muscle activation that does not require steady

state energy consumption, only a brief muscle activation. The contribution of reflex to effective stiffness is modulated by the reflex gain. Reflex gain defines the amount of change in muscle activation for a given perturbation. As such, the first specific aim of this study was to investigate the relationship between reflex gain, co-contraction, and stability in the spine. We hypothesized that reflexes allow more energy efficient control of stability than steady state recruitment of intrinsic stiffness alone. To test our hypothesis we used a model of the spine to investigate how reflex gain influences spinal stability and associated energy expenditure.

While reflexes may aid in stability, reflex delay may adversely affect stability of the spine. The reflexive system may be viewed as a feedback control mechanism that acts to stabilize the spine. When the spine is perturbed the neuromuscular reflex response changes muscle activations to reject the perturbation and return the system to its original configuration. Reflex delay describes the time from a perturbation to the onset of reflex activation. If reflex delay were large compared with the natural oscillatory period of the spine, then the reflex may cease to stabilize the system and in fact may destabilize it. This effect has been observed in patients with low back pain wherein abnormally large reflex delays were related to impaired postural control (Radebold *et al.*, 2001). The effect of reflex delay on stability may additionally depend on reflex gain. Therefore the second specific aim of this study was to investigate the relationship between reflex gain and the maximum limit of reflex delay. It was hypothesized that small reflex gains will tolerate large values of reflex delay while large reflex gains will only tolerate small values of reflex delay. Investigations of these hypotheses provide insight into the relationship between reflex gain, delay, and stability of the musculoskeletal system.

B. Methods

Model Anatomy and Geometry

A dynamic model of the spine was implemented to investigate the stabilizing role of reflexes. This model was unique in that it was a forward dynamic simulation of spinal mechanics and thus solved differential kinematics for given force inputs. The model represented a three dimensional 18 degree of freedom spine consisting of five lumbar vertebrae and a thoracic segment. Gravitational trunk mass was distributed among the rigid bodies as described in the literature (Liu *et al.*, 1971). Passive intervertebral reaction moments were incorporated as lumped parameter viscoelastic models between neighboring vertebral bodies. The disc model was assigned stiffness of 50Nm/rad (Stokes *et al.*, 2002). Intervertebral damping was assigned 0.5Nms/rad, 1% of the stiffness value (Kasra *et al.*, 1992; Izambert *et al.*, 2003). A three dimensional anatomy of 90 muscles was used wherein lines of action were represented with the centriod-line approach with many muscles passing through nodal points to account for curved lines of action. The anatomy described origin and insertion locations in vertebral body fixed coordinate frames described in the literature (Cholewicki and McGill, 1996; Stokes and Gardner-Morse, 1999). These provided muscle lengths, velocities, and generalized force vectors associated with spinal kinematics. Two loading conditions were investigated in this study: upright unloaded and upright with 200N external flexion load. External load was applied vertically 20cm anterior to the trunk at the T4 level (Figure 1).

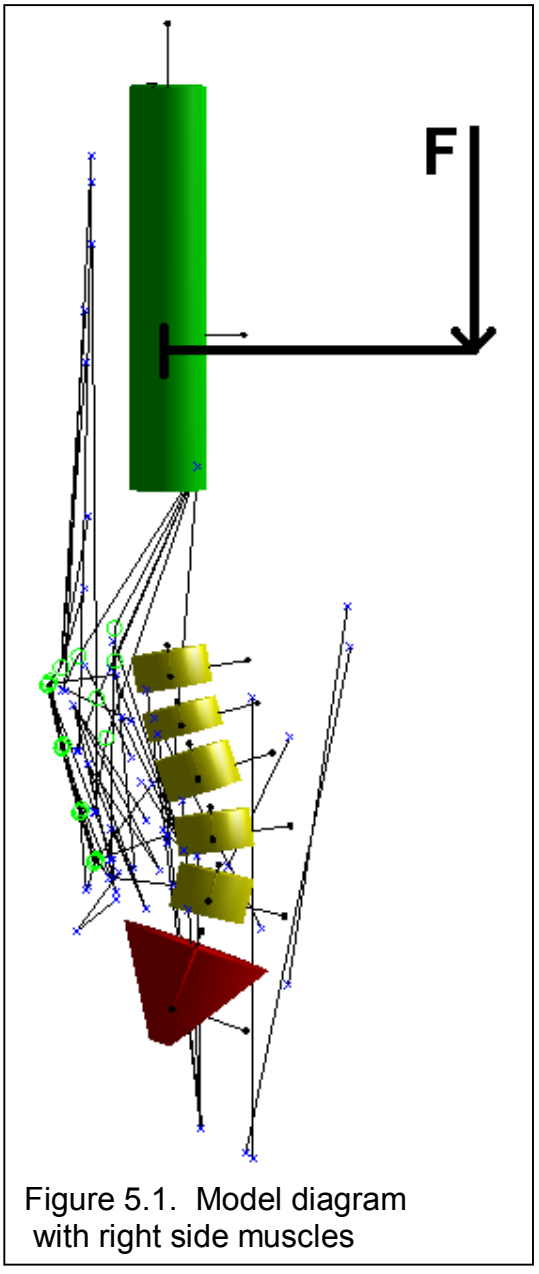


Figure 5.1. Model diagram with right side muscles

Both external and internal forces were modeled as generalized forces Q acting on each segment. The system dynamics were represented in differential matrix form with respect to the vector of kinematic angles $\theta(t)$

$$\ddot{\theta} = \mathrm{M}(\theta)^{-1} \cdot \{\mathrm{Q}(\theta, \dot{\theta}, \alpha) - \mathrm{C}(\theta, \dot{\theta}) - \mathrm{G}(\theta)\} \quad (1)$$

where M represents an 18x18 inertial mass matrix, G is a vector of gravitational forces, C is a vector of velocity related forces, and Q is a vector of generalized forces.

Muscle Force and Reflex Model

Muscle contraction force was represented by a muscle model that treated stiffness and damping as a variable influenced by steady-state muscle activation and posture. Muscle force f_m was represented as a function of myoelectric activation □(t), muscle length l_m, and muscle velocity v_m

$$f_m(\alpha(t), l_m, v_m) = f_o \cdot \alpha(t) \cdot \left\{1 + q \cdot \left(\frac{l_m - l_o}{l_o}\right) + b \cdot \frac{v_m}{l_o}\right\} \quad (2)$$

where f_o was the maximum isometric force, l_o was the muscle length at equilibrium posture, q was the intrinsic stiffness coefficient, and b was the damping coefficient. Maximum isometric force f_o for a muscle was computed as the cross sectional area multiplied by the maximum muscle stress of 46N/cm^2 (Gardner-Morse *et al.*, 1995). The constant value q describes the change in active muscle stiffness as a function of steady-state force and muscle length (Bergmark A., 1989). Summary of available literature by Crisco and Panjabi (1991) estimated the value q to be between 0.5 and 40 with an average value of 10. Hence, our model implemented q = 10. The constant damping coefficient b describes the change in muscle damping as a function of steady-state force and muscle length, a relationship similar to the stiffness coefficient q. The effects of the damping coefficient value are explored in this study.

Muscle activation $\alpha(t)$ was represented as the combination of steady state muscle activation α_o and a reflex muscle activation α_R. Reflex activation has been measured and characterized as response function to kinetic and kinematic disturbances (Granata *et al.*, 2004). Therefore, reflex activation α_R was modeled as proportional to change in muscle length delayed by a constant reflex time delay τ. Reflex response was designed to increase proportional to steady state activation α_o as per published measurements (Matthews, 1986). Reflex activation was assigned a scalar reflex gain G_R, which allowed modulation of the reflex response.

$$\alpha(t) = \alpha_o + \alpha_r(t), \quad where \quad \alpha_r(t) = \alpha_o \cdot G_R \cdot \left\{ \frac{l(t-\tau) - l_o}{l_o} \right\} \tag{3}$$

Stability and Muscle Activation Pattern

Stability of the system was determined by analyzing the Jacobian of the model dynamics. The dynamics in equation (1) can be expressed as $\dot{x}(t) = F\{x(t)\}$ for the vector $x(t) = \left[\theta(t)^t, \dot{\theta}(t)^t\right]^t$ where $\dot{x}(t)$ and $\dot{\theta}(t)$ are the time derivatives of the state vector x(t) and posture θ(t) respectively. In the resulting linear model the time derivative of state vector x(t) is the product of the Jacobian, J_i, with the state vector.

$$\dot{x}(t) = J_i \cdot x(t) \tag{4}$$

The 36x36 Jacobian matrix J_i was computed from the gradient of the nonlinear state differential function $F\{\cdot\}$ with respect to x(t) evaluated at equilibrium. Hence, the Jacobian is a representation of the linearized dynamics in equation (1) about the equilibrium point. The system was asymptotically stable if the eigenvalues of matrix J_i had negative real components.

The system Jacobian was a function of muscle recruitment and therefore required *a priori* values of steady-state activation □$_o$. These activation values were calculated using an algorithm that minimized metabolic power subject to constraints which guaranteed stability and equilibrium. Metabolic power was calculated as the power of maintaining muscle activation, thus assuming the model to be in steady state (Anderson, 1999). The analyses were seeded with a randomized initial set of activations and each solution was repeated three times to assure a consistent minimum was found. Repeated solutions were identical for the randomized seeds.

Time Delay Stability

System dynamics were expanded to include a delayed Jacobian related to the delayed state x(t-τ) for time delay τ.

$$\dot{x}(t) = J_i \cdot x(t) + J_d \cdot x(t-\tau) \tag{5}$$

The instantaneous Jacobian J_i represented effects of gravity, intrinsic stiffness, and steady-state muscle force, while the delayed Jacobian J_d represented effects from the

reflex response. For the system with given instantaneous Jacobian J_i and delayed Jacobian J_d that is stable with no time delay, there exists a maximum tolerable delay □, called the delay margin. If a time delay larger than the delay margin is applied then the system becomes unstable.

The model implemented a method by Chen (Chen, 1995) for computing the delay margin of a linear system. Briefly, we examined at what time delays an eigenvalue in the delayed system (equation 5) crosses the imaginary axis, i.e. becomes unstable. We defined the system to be stable when time delay $\tau=0$ resulted in the eigenvalues of matrix $[J_i + J_d]$ having negative real components. Modifying the time delay changes the eigenvalues. When the time delay equals the delay margin, one of these eigenvalues crosses the imaginary axis causing instability. To determine the delay margin at which this crossing occurs, the problem is reformulated into a generalized eigenvalue problem. Eigenvalue frequencies are sought for which a solution to the generalized eigenvalue problem is also a solution corresponding to an imaginary axis eigenvalue crossing in the original problem (equation 5). This method yields a set of eigenvalues and frequencies, each corresponding to a crossing event. The ratio of each eigenvalue's phase and the frequency of crossing results in a set of time delays corresponding to crossings. The delay margin is the minimum value in this set.

Procedure Outline

The procedure employed in this study found the muscle activation pattern and delay margin for a given external load, reflex gain, and damping coefficient. The muscle activation set α_0 that minimized metabolic power at stable equilibrium was found by initially assuming reflex time delay of zero (specific aim 1). The resulting co-contraction level necessary to achieve stability was quantified using methods described elsewhere (Granata *et al.*, 2005a). These activations were used to solve for the delay margin of the system, i.e. maximum permissible reflex delay (specific aim 2). The process was replicated for the two external loading conditions, and a range of reflex gains G_R and damping coefficients b.

C. Results

Reflex allowed the model to stabilize with lower metabolic power than with intrinsic stiffness alone. In both loading conditions (0 N, 200 N external load) a range of reflex gain and damping parameters were examined. Muscle activation sets were found that minimized metabolic power while providing stability to the model with zero reflex time delay. As reflex gain increased, the metabolic power required for stability decreased until it reached the minimum metabolic power. This minimum metabolic power is the power required to maintain equilibrium (Figure 2). A solution with a metabolic power greater than equilibrium was attributed to antagonistic muscle co-contraction. Therefore with small reflex gain steady state co-contraction was required for stability (Table 1) leading to greater metabolic power than that required for equilibrium alone. The minimum metabolic power was identical for all tested values of the damping coefficient.

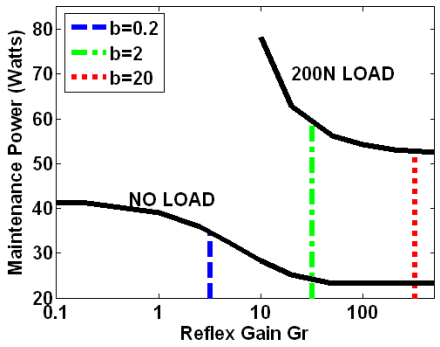


Figure 5.2. Metabolic power vs. reflex gain for both load conditions. When reflex delay is considered, vertical lines represent maximum tolerable reflex gain for the associated damping coefficient.

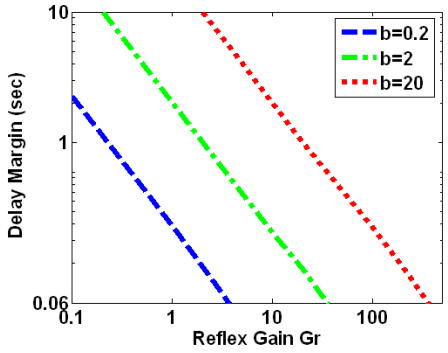


Figure 5.3. Delay margin vs. reflex gain for multiple damping coefficients. The figure is terminated at 60ms. The relationships presented were identical for both load conditions.

Load	G_R=1	G_R=10	G_R=100
0 N	38%	9.4%	0%
200 N	(no stable solution)	56%	3.4%

Table 5.1. Co-contraction percentages for both loading conditions at multiple reflex gains

In addition to minimizing metabolic power reflexes allowed the system to stabilize when intrinsic stiffness alone was inadequate. Stability in the unloaded condition was feasible for all values of reflex gain including the absence of reflex (G_r=0) (Figure 2). Recall however that co-contraction necessarily increased with smaller values of reflex gain. Conversely, stability was not feasible for all values of reflex gain at 200N

of external flexion load. The intrinsic muscle stiffness chosen in our study q=10 was insufficient to stabilize the loaded spine for any reflex gain less than G_R=10.

Delay adversely affected stability by limiting the reflex gain the system could tolerate. Instability occurred when the physiological delay was greater than the system delay margin. To model this effect a physiological reflex delay of 60ms was chosen based on measurements in the paraspinal muscles (Reeves 2005). Any system with a delay margin below 60ms was interpreted as unstable. The delay margin of the system was logarithmically related to the reflex gain (Figure 3) and was identical for both the 0N and 200 N loading conditions. The maximum tolerable reflex gain was the value at which the delay margin fell below 60ms. The presence of a maximum reflex gain limited the effectiveness of reflex to decrease metabolic power for control of spinal stability (Figure 2).

Damping influenced the delay margin of the system and maximum tolerable reflex gain. An increase in the damping coefficient shifted the relationship between reflex gain and the delay margin (Figure 3). This shift changed the maximum reflex gain at which the delay margin fell below 60ms. Larger damping coefficients resulted in larger maximum tolerable reflex gains. The importance of the damping coefficient was illustrated in the 200N loaded system. In this condition stability could not be achieved with the smallest damping coefficient (b=0.2) but stable conditions were observable for b=2 and b=20. When examining the small damping coefficient the maximum tolerable reflex gain was G_R=3.2, while stability required a reflex gain greater than G_R=10.

D. Discussion

This study demonstrated that reflexes play an important role in the stability of the spine. We have shown that reflexes reduce antagonistic muscle co-contraction necessary for stability thereby lowering the metabolic power required to perform an exertion. Additionally the model suggests there are conditions which cannot be stabilized with intrinsic stiffness alone. This agrees with empirical measurement wherein it was observed that intrinsic stiffness alone may be insufficient to stabilize the torso in upright postures (Moorhouse, 2005). The results of our study show that the ability of reflexes to

augment effective stiffness can stabilize the spine in these configurations. However, the benefits of reflex are limited by reflex time delay.

The delay margin of the spine did not change with external load. Only two loading conditions were investigated in this study. However, for these two conditions the relationship between delay margin, reflex gain, and damping were identical. If this relationship is in fact load invariant, then there may be an optimal reflex gain based on physiological reflex delay, system damping, and posture. This highlights the potential risk associated with fatigue and prolonged flexion posture, which have been shown to influence reflex gain (Granata *et al.*, 2005b; Madigan *et al.*, 2006). In applications where the maximum reflex gain is already utilized for stability any need to further increase reflex gain (for example in response to decreased intrinsic stiffness) may result in instability.

Patients with greater than normal reflex delay may require more co-contraction than their healthy counterparts. Evidence suggests that patients with low back pain may have abnormally long reflex delays (Radebold *et al.*, 2001). The delay margin is logarithmically related to the maximum tolerable reflex gain. Hence, abnormally long reflex delay may require a reduction in the maximum reflex gain. To stabilize the same system with reduced reflex gain, this study has shown that the model must co-contract antagonistic muscles. Van Dieen (Van Dieen *et al.*, 2003) and Marras (Marras W.S. *et al.*, 2001) observed that patients with low back pain recruit greater than normal levels of co-contraction. In fact, Van Dieen (Van Dieen *et al.*, 2003) concluded that this co-contraction strategy may be used to compensate for limitations elsewhere in the stabilizing control system such as impaired reflex dynamics. Our results support this conclusion, i.e. long reflex delay and reduced reflex gain require increased stabilizing co-contraction. Hence, these patients may be required to use less energy efficient motor recruitment strategies for control of spinal stability.

There are several limitations that must be addressed with further work. First, the stability analyses assumed the system behaved like its linearized representation. This assumption was the foundation of the time delay stability criterion, which enabled the computation of the delay margin. Future work should consider using time domain stability methods which may use the nonlinear dynamics of the model. Second, the

intrinsic stiffness parameter q was chosen to be 10, the average value suggested by the available literature (Crisco and Panjabi, 1991). It has also been suggested that smaller values of q are sufficient for stability (Gardner-Morse *et al.*, 1995). Due to our parameter choice, we found that intrinsic stiffness alone was unable to stabilize the 200N external flexion load. If a larger value of q was chosen, then the system may have achieved stability without reflex. However, for a larger value of q a different load could have been chosen which intrinsic stiffness alone could not have stabilized, thus repeating the results in this study. Third, results were based on the assumption that the system was stable with reflexes of zero time delay. There are conditions wherein it is conceivable that the spine may be unstable with zero time delay but stable with $\tau>0$. This possibility may obscure results. However, the assumption was made for computational efficiency and its affect will be explored at a later date. Finally, the maximum reflex gain was found which allowed stability while minimizing metabolic power. This reflex gain was dangerously close to the gain which destabilized the system. Any recruitment error could cause instability. A sensitivity analysis which assumes some error in recruitment could be used to find the ideal reflex gain to simultaneously minimize metabolic power and risk of injury.

In conclusion, results of this study provide insight into the effects of reflex on the stability of the spine. The analyses demonstrated that reflexes are an important resource for stabilizing the spine and that there are advantages to recruiting reflexes over intrinsic stiffness by co-contraction. For a given reflex delay this study found that a subject has an optimal reflex gain that minimizes metabolic power while assuring stability. This result may aid in understanding of how patients with retarded reflex delay utilize muscle recruitment for stability.

E. Reference List

Anderson,F.C. (1999) A dynamic optimization solution for a complete cycle of normal gait. Ph.D. Univ Texas, Austin.

Bergmark A. (1989) Stability of the lumbar spine: A study in mechanical engineering. *Acta Orthop.Scand.Suppl.* **230,** 1-54.

Chen,J. (1995) On Computing the Maximal Delay Intervals for Stability of Linear Delay Systems. *Ieee Transactions on Automatic Control* **40,** 1087-1093.

Cholewicki,J. and McGill,S.M. (1996) Mechanical stability on the in vivo lumbar spine: Implications for injury and chronic low back pain. *Clin.Biomech.* **11,** 1-15.

Cholewicki,J., Simons,A.P.D., and Radebold,A. (2000) Effects of external trunk loads on lumbar spine stability. *J.Biomechanics* **33,** 1377-1385.

Crisco,J.J. and Panjabi,M.M. (1991) The intersegmental and multisegmental muscles of the lumbar spine: A biomechanical model comparing lateral stabilizing potential. *Spine* **16,** 793-799.

El Rich,M. and Shirazi-Adl,A. (2005) Effect of load position on muscle forces, internal loads and stability of the human spine in upright postures. *Comput.Methods Biomech.Biomed.Engin.* **8,** 359-368.

Gardner-Morse,M.G., Stokes,I.A.F., and Laible,J.P. (1995) Role of muscles in lumbar stability in maximum extension efforts. *J.Orthop.Res.* **13,** 802-808.

Granata,K.P., Lee,P.E., and Franklin,T.C. (2005a) Co-contraction recruitment and spinal load during isometric trunk flexion and extension. *Clin Biomech.(Bristol., Avon.)* **20,** 1029-1037.

Granata,K.P., Rogers,E., and Moorhouse,K. (2005b) Effects of static flexion-relaxation on paraspinal reflex behavior. *Clin Biomech.(Bristol., Avon.)* **20,** 16-24.

Granata,K.P., Slota,G.P., Bennett,B.E., and Kang,H.G. (2004) Paraspinal muscle reflex dynamics. *J.Biomechanics* **37,** 241-247.

Hatze H. (1977) A myocybernetic control model of skeletal muscle. *Biol.Cybern.* **25,** 103-119.

Izambert,O., Mitton,D., Thourot,M., and Lavaste,F. (2003) Dynamic stiffness and damping of human intervertebral disc using axial oscillatory displacement under a free mass system. *Eur.Spine J.* **12,** 562-566.

Kasra,M., Shirazi-Adl,A., and Drouin,G. (1992) Dynamics of human lumbar intervertebral joints. Experimental and finite-element investigations. *Spine* **17,** 93-102.

Kearney R.E. and Stein R.B. (1997) Identification of intrinsic and reflex contributions to human ankle stiffness dynamics. *IEEE Trans.Biomed.Eng* **44,** 493-504.

Lee,P.J., Rogers,E.L., and Granata,K.P. (2006) Active trunk stiffness increases with co-contraction. *J.Electromyogr.Kinesiol.* **16,** 51-57.

Madigan,M.L., Herrmann,C., Davidson,B., and Granata,K.P. Effects of Lumbar Extensor Fatigue on Paraspinal Lumbar Reflexes. Clin.Biomech. 2006.

Ref Type: In Press

Marras W.S., Davis K.G., Ferguson S.A., Lucas B.R., and Gupta P. (2001) Spine loading characteristics of patients with low back pain compared with asymptomatic individuals. *Spine* **26,** 2566-2574.

Matthews,P.B.C. (1986) Observations on the automatic compensation of reflex gain on varying the pre-existing level of motor discharge in man. *J.Physiol.* **374,** 73-90.
Moorhouse,K.M. (2005) Role of intrinsic and Reflexive Dynamics in the Control of Spinal Stability. Ph.D. Virginia Tech.
Radebold,A., Cholewicki,J., Polzhofer,G.A., and Green,T.P. (2001) Impaired postural control of the lumbar spine is associated with delayed muscle response times in patients with chronic idiopathic low back pain. *Spine* **26,** 724-730.
Stokes,I.A. and Gardner-Morse,M. (1999) Quantitative anatomy of the lumbar musculature. *J.Biomech.* **32,** 311-316.
Stokes,I.A.F., Gardner-Morse,M.G., Churchill,D., and Laible,J.P. (2002) Measurement of spinal motion segment stiffness matrix. *J.Biomechanics* **32,** 517-521.
Van Dieen,J.H., Cholewicki,J., and Radebold,A. (2003) Trunk muscle recruitment patterns in patients with low back pain enhance the stability of the lumbar spine. *Spine* **28,** 834-841.

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

A. Conclusions

The goal of this work was to develop a biomechanical model of spinal stability and to investigate how reflexes affect the stability of the spine. One should note that this study did not include experimental measurements, but rather used a theoretical model. However, assuming the model of the spine is a reasonable approximation of the physiological spine, the results should be useful in formulating hypotheses for experimental measurement.

i. Reflexes in the spine

The results of this study confirmed both hypotheses presented in Chapter 1. The first hypothesis suggested that adding reflexes to the spine would decrease metabolic power required for stability. The work presented in Chapter 5 showed that as proportional muscle reflex gain was increased, the optimization algorithm found stability could be achieved with less antagonistic co-contraction and metabolic power (verifying hypothesis 1 a,b). The second hypothesis suggested that large reflex delays would force the neuromuscular system to rely on co-contraction to maintain stability. This hypothesis was confirmed in Chapter 5 by recognizing that reflex gain and delay margin were inversely related. Such a relationship meant that for any reflex delay there was a maximum reflex gain, and this maximum reflex gain was very small for large reflex delay. Therefore a system with large reflex delay had little (if any) stiffness due to reflex, and was forced to rely on co-contraction for stability (hypothesis 2, a). The corollary hypothesis (2,b) stated that for small reflex delay, the system should be able to rely entirely on reflexes for stability. This hypothesis was confirmed by recognizing that in the limiting case a system with zero delay was stable for any delay margin. Such a system could therefore recruit infinite values of reflex gain, and hence stiffness due to reflex. As reflex gain approached infinite values, zero co-contraction was needed to stabilize the model.

ii. Methods Employed

The methods presented in this paper may be used to model a variety of problems in biomechanics. Most of the analyses in this work focused on stability of the spine. These analyses, however, will work for any biomechanical system in which the effect of reflexes on stability about equilibrium is concerned. The methods may need to be modified for specific problems. For example, in section 3.B.iv it was assumed that the oscillatory period of the spine was large compared with reflex delay, and thus the mass matrix and muscle moment arms were assumed to be in phase with the reflexes. In a fast moving system this approximation may not hold. However, most of the methods need no such modification and this work should provide the grounds for further analysis of the role of reflexes in biomechanical stability.

B. Next Steps

The general nature of the reflexive spine model provides for a wide variety of experiments. This section will detail some suggested future experiments to take full advantage of the model.

i. Kinematic Optimization

The current spine model has fixed kinematics (posture), which is used to calculate stability. The source of the fixed kinematics has not been addressed because it is not clear how to accurately measure in vivo spine kinematics. Some researchers have realized a way to circumvent this problem by implementing a novel solution whereby they allowed the kinematics of the spine to vary as part of the muscle activation optimization algorithm (Shirazi-Adl et al 2002). The concept is supported by the idea that the human spine optimizes posture simultaneous with muscle activations to minimize metabolic power while ensuring stability. Easing the spine into a posture that takes advantage of passive structures such as the intervertebral discs (for instance by increasing the lordotic angle of the lumbar spine) may be one example of how this concept manifests itself in the in vivo spine.

To allow for kinematic optimization of our spine model, the muscle activation optimization algorithm must exist inside an outer kinematics optimization algorithm. The

outside loop would modify the kinematics of the spine. The inside loop would solve for muscle activations that minimized metabolic power subject to the stability of the prescribed kinematics. In section 3.C.iii, a numerical strategy was discussed whereby the Jacobian was built quickly by taking the dot product of a 3D matrix with the muscle activation vector. Use of this method to speed up the optimization is not as effective with kinematic optimization, as that 'constant' 3D matrix is a function of the kinematics. Therefore for each iteration of the outer loop (kinematic loop), another 'constant' 3D Jacobian must be calculated. This method was briefly implemented to gauge its feasibility. From this work, it was apparent that the solution did not quickly converge on optimal kinematics and muscle activations. Therefore, performing this experiment would likely be computationally expensive. However, given sufficient computational resources, this method may converge on an optimal kinematic posture with matching optimal muscle activations.

ii. Coupled Dynamic Systems

To date, the experiments that have been performed on the spine model involve searching for parameters that assure stability under an isotonic external load. These experiments have neglected the obvious possibility of an external load with dynamic properties. One example of such a dynamic load is a box that must be pushed against friction. The box has an inertial load proportional to acceleration and a force of friction that is a function of sliding velocity. To move the box, the spine model must supply a force equal to the dynamic load. Interesting experiments may be performed by coupling a dynamic load to the spine model.

Since the linearized system is generated numerically, increasing the degrees of freedom of the linear system to include a coupled system should not be an obstacle. The equilibrium state of the system can have a non-zero velocity. In such a case, the linearized state vector (see section 3.B.iii) defined about equilibrium will describe only the change in velocity and position relative to the spine. The two systems may be coupled through generalized force vectors. In the example of pushing a box, we may even include a passive spring between the box and trunk to model the stiffness of articulating arms. This stiffness will likely modify the stability of the system to maintain

its equilibrium position (and remain at constant velocity). Such a small modification to the model offers possibilities in understanding the stability of pushing versus pulling of sliding boxes, among other possible experiments.

iii. Sensitivity

All of the parameters in the spine model have been treated as values of infinite precision. However physiological muscle activation, which is a frequency-modulated neural pulse train, is unlikely to possess infinite precision and will therefore have some amount of noise in the signal. There is a possibility of muscle recruitment error as a consequence of this noise in the control signal that may be detrimental to a system that cannot stabilize such error. Thus, an interesting new direction for this project would be to allow the model to expect recruitment error and observe how it modifies the optimized muscle activations to ensure stability.

To reprogram the model to 'expect' stochastic behavior, it is possible to have the optimization algorithm check multiple Jacobians for stability. Each of these Jacobians could correspond to the spine model with some small amount of recruitment error. In the limiting case, the optimization algorithm would need to check every combination of individual muscle recruitment error. A simpler method may be to check M Jacobians (for M muscles), each corresponding to one muscle having a given percentage of recruitment error. The optimization algorithm may then be constrained to force all M Jacobians to be stable. Hypothetically, the model will increase intrinsic stiffness, or co-contraction, to stabilize against the uncertainty in muscle recruitment. It may then be possible to test these results by integrating the fully nonlinear model while injecting 'recruitment error' into muscle activations.

iv. Optimal control theory

All of the stability analyses employed have taken advantage of the linearized system dynamics. Since the fully nonlinear system equations are available, it may be interesting to investigate more complex control. For example, it may be revealing to investigate methods of optimal control theory to move the spine model through some desired cyclic movement. One example of this movement experiment would be to have

the model pick a box off of a table, lift it, and place it back on the table. This experiment is very similar to existing biomechanical experiments investigating stability of cyclic motion of the *in vivo* human trunk (Granata and England 2006). In these experiments, human subjects perform such a task before and after an intervention, such as fatigue. It should prove interesting to compare the results of a theoretical model to experimental results.

The implementation of such optimal control theory often requires use of a shooting method. A shooting method involves integrating the dynamic system following an estimate of the control solution. These integrations are computationally expensive. Therefore, moving into the field of optimal control theory may require parallel computing resources. If there were M computers (for M muscles), each computer may integrate the model with a small perturbation to one of the muscles from the initial solution. Using such methods, the optimization algorithm may dramatically decrease the time required to compute a solution.

C. Improvements to the model

The model implemented in this work is broad in usage, and may therefore find application in a variety of experiments. Still, there is room to expand the model's generality and perhaps improve the approximation of the physiological system.

i. Six DOF per vertebrae

Increasing the degrees of freedom of the model will allow translation of the vertebrae, in addition to rotation. Translation is important because it is one of the common failure mechanisms in the spine. For example, spondylolisthesis is a common low back injury that involves shearing of a vertebral joint in the anterior direction (Jayakumar et al 2006). Another common injury of the spine is a herniated disc, which is believed to be triggered by compression of the spinal column (Iencean 2000). These additional degrees of freedom will allow the intervertebral model to be expanded from a lumped parameter model into a finite element model. Such a model will allow for distributed loading and investigations of both strain and strain rate in the local tissues.

ii. Dynamic Pelvis, Thighs

While the goal of this model is to investigate spine mechanics and not whole body mechanics, it is necessary to recognize that the spine has muscles in common with the pelvis and femur. A significant number of spine muscles have their origin on the pelvis, most of which lie on the iliac crest (Cholewicki et al, 1996). Similarly the femur is a point of paraspinal muscle attachment, with the psois being the largest muscle it has in common with the spine. In the spine model developed in this work, the femur and pelvis are kinematically fixed and the rest of the spine is free to move on top of them. In the physiologic spine, moments acting on the spine through common muscles create reaction moments on the pelvis or femur. By neglecting these reaction moments, we assumed these bodies are fixed in such a way that makes it impossible for the muscles to move them. This is an unrealistic condition to recreate in a laboratory and therefore should not exist in the model. Both bodies should be assigned dynamic properties to allow them to move kinetically. If an experiment may benefit from the reduced model in which those bodies are fixed, it is trivial to kinematically constrain them. However in the interest of generality these degrees of freedom should exist in the model.

iii. Articulated Thorax

It is possible that the assumption that the thorax moves as a rigid body negatively impacts the model results. As one example of how this assumption could affect the biofidelity of the model, consider a subject with trunk flexion. The forward leaning curvature of the thorax (kyphosis) will lengthen the erector spinae muscles, thus creating a hardening intrinsic stiffness. Additionally this flexion in the thorax will decrease the rotational inertia of the spine about the pelvis. By modeling the thorax as a rigid body these effects are ignored (though lumbar spine deformation has similar effects). While it is possible to argue for or against the importance of an articulating thorax in a spine model, the degrees of freedom should be increased when it is apparent that the physiological spine (*in vivo*) behaves differently than the theoretical model.

iv. Arms

The addition of arms to a spine model is an area that also needs to be explored. A principal goal of spine research is to understand how industrial environments cause low back pain in the workforce. Toward this end, researchers often investigate the stability of the spine while lifting a load (for example, picking up a box). Unfortunately, the current spine model is incapable of 'lifting a load'. While external loads may be applied to any rigid body of the model, applying a vertical load to the thorax (at shoulder level) is not a good approximation of a person lifting a load with their hands. A person lifting a load with their hand is effectively supporting the load at the end of a pendulum (their arm), which is a stable structure when 'hanging' from a pivot point (their shoulder). Additionally, the neuromuscular structure of the arms provides viscoelastic damping between the load and shoulders (Rancourt and Hogan 2001). It is therefore possible that the dynamics of such a lifting experiment are more stable than a comparable experiment where the load is applied at shoulder level. This hypothetical increase in stability is due to the stabilizing nature of the pendulum-like arm. This hypothesis, and the many hypotheses that follow, suggest that adding arms to the model is necessary to advance the understanding of the causes of low back pain in industrial settings.

References

Anderson, F. C. (1999). "A Dynamic Optimization Solution for a Complete Cycle of Normal Gait." PhD Thesis, University of Texas.

Arjmand, N. and A. Shirazi-Adl (2005). "Biomechanics of changes in lumbar posture in static lifting." Spine 30(23): 2637-48.

Bergmark, A. (1989). "Stability of the lumbar spine: a study in mechanical engineering." Acta Orthop Scand 60([Suppl 230]): 1-54.

Brown, I. E. and G. E. Loeb (2000). "Measured and modeled properties of mammalian skeletal muscle. IV. Dynamics of activation and deactivation." Journal of Muscle Research and Cell Motility **21**: 33-47.

Chen, J. (1995). "On Computing the Maximal Delay Intervals for Stability of Linear Delay Systems." IEEE Transactions On Automatic Control **40**(6): 1087-1093.

Cholewicki, J. and S. M. McGill (1996). "Mechanical stability of the in vivo lumbar spine: implications for injury and chronic low back pain." Clinical Biomechanics **11**(1): 1-15.

Cholewicki, J., S. Silfies, et al. (2005). "Delayed Trunk Muscle Reflex Responses Increase the Risk of Low Back Injuries." SPINE **30**(23): 2614-2620.

Crisco, J. and M. Panjabi (1991). "The Intersegmental and Multisegmental Muscles of the Lumbar Spine: A Biomechanical Model Comparing Lateral Stabilizing Potential." SPINE **16**(7): 793-799.

Fagan, M. J., S. Julian, et al. (2002). "Patient-specific spine models. Part 1: finite element analysis of the lumbar intervertebral disc - a material sensitivity study." Proceedings Institution of Mechanical Engineers **216**(H): 299-314.

Gardner-Morse, M., I. A. F. Stokes, et al. (1995). "Role of Muscles in Lumbar Spine Stability in Maximum Extension Efforts." Journal of Orthopaedic Research **13**: 802-808.

Goto, K., N. Tajima, et al. (2002). "Mechanical analysis of the lumbar vertebrae in a three-dimensional finite element method model in which intradiscal pressure in the nucleus pulposus was used to establish the model." Journal of Orthopaedic Science **7**: 243-6.

Granata, K., G. Slota, et al. (2004). "Paraspinal muscle reflex dynamics." Journal of Biomechanics **37**: 241-247.

Granata, K. P., P. E. Lee, et al. (2005). "Co-contraction recruitment and spinal load during isometric trunk flexion and extension." Clinical Biomechanics **20**: 1029-1037.

Hogan, N. (1984). "Adaptive Control of Mechanical Impedance by Coactivation of Antagonistic Muscles." IEEE Transactions On Automatic Control **AC-29**(8): 681-690.

Iencean, S. M. (2000). "Lumbar intervertebral disc herniation following experimental intradiscal pressure increase." Acta Neurochir (Wien) **142**(6): 669-76.

Izambert O., M. D., Thourot M., Lavaste F. (2003). "Dynamic stiffness and damping of human intervertebral disc using axial oscillatory displacement under a free mass system." Eur Spine J **12**: 562-566.

Jayakumar, P., C. Nnadi, et al. (2006). "Dynamic degenerative lumbar spondylolisthesis: diagnosis with axial loaded magnetic resonance imaging." Spine **31**(10): E298-301.

Kasra, M., A. Shirazi-Adl, et al. (1992). "Dynamics of Human Lumbar Intervertebral Joints: Experimental and Finite-Element Investigations." SPINE **17**(1): 93-102.

Kearney, R. E., R. B. Stein, et al. (1997). "Identification of Intrinsic and Reflex Contributions to Human Ankle Stiffness Dynamics." IEEE Transactions On Biomedical Engineering **44**(6): 493-504.

Lee, P., E. Rogers, et al. (2006). "Active trunk stiffness increases with co-contraction." Journal of Electromyography and Kinesiology 16: 51-57.

Liu, Y. K., J. M. Laborde, et al. (1971). "Inertial properties of a segmented cadaver trunk: their implications in acceleration injuries." Aerosp Med **42**(6): 650-7.

Moorhouse, K. and K. Granata (2006). "Role of Reflex Dynamics in Spine Stability: Intrinsic Muscle Stiffness Alone is Insufficient for Stability." Journal of Biomechanics **In Press**.

Norman, R., R. Wells, et al. (1998). "A comparison of peak vs cumulative physical work exposure risk factors for the reporting of low back pain in the automotive industry." Clin Biomech (Bristol, Avon) **13**(8): 561-573.

Radebold, A., J. Cholewicki, et al. (2001). "Impaired Postural Control of the Lumbar Spine is Associated With Delayed Muscle Response Times in Patients With Chronic Idiopathic Low Back Pain." SPINE **26**(7): 724-730.

Rancourt, D. and N. Hogan (2001). "Stability in force-production tasks." J Mot Behav **33**(2): 193-204.

Shirazi-Adl, A., S. Sadouk, et al. (2002). "Muscle force evaluation and the role of posture in human lumbar spine under compression." Eur Spine J **11**: 519-526.

Stokes, I., M. Gardner-Morse, et al. (2002). "Measurement of a spinal motion segment stiffness matrix." Journal of Biomechanics **35**: 517-521.

van Dieen, J. H., J. Cholewicki, et al. (2003). "Trunk Muscle Recruitment Patterns in Patients With Low Back Pain Enhance the Stability of the Lumbar Spine." SPINE **28**(8): 834-841.

Zajac, F. E. (1989). "Muscle and Tendon: Properties, Models, Scaling, and Application to Biomechanics and Motor Control." Critical Reviews in Biomedical Engineering **17**(4): 360-410.

APPENDIX A
HOW TO BUILD THIS MODEL

This chapter provides directions to recreate the spine model, or any similar biomechanical model.

i. Dynamics

Section 3.B discussed how the equations of motion were computed symbolically. In our implementation this computation was performed in Mathematica, a program that specializes in symbolic computation and simplification of complicated equations. Below is the code used to generate the dynamics, which were used for numerical evaluations of angular accelerations in the model.

Mathematica Code

Goal: Find M_m matrix and C_v, G_v vectors (see section 3.B.i)

```
6 Pend general model
For use with muscle file & Matlab
See: Cholewicki 1996 Clin. Biomech. Article for attachments
Tim Franklin 6-2-05
With 123 angle rotations

----------Initialize Workspace
ClearAll["Global`*"];
Remove["Global`*"];
Off[General::spell1];
Off[General::spell];

----------Define Rotation Matrices, Vector and Length functions
R1[θ_] := {{1, 0, 0}, {0, Cos[θ], Sin[θ]}, {0, -Sin[θ], Cos[θ]}};
R2[θ_] := {{Cos[θ], 0, -Sin[θ]}, {0, 1, 0}, {Sin[θ], 0, Cos[θ]}};
R3[θ_] = {{Cos[θ], Sin[θ], 0}, {-Sin[θ], Cos[θ], 0}, {0, 0, 1}};
Vec[Vdr_, Vor_] := (Vdr - Vor)/Sqrt[(Vdr - Vor).(Vdr - Vor)];
Len[Vdr_, Vor_] := Sqrt[(Vdr - Vor).(Vdr - Vor)];

----------Define Inertial Coordinate System
1=X,2=Y,3=Z
n1 = {1, 0, 0};
n2 = {0, 1, 0};
n3 = {0, 0, 1};

----------Define Body Fixed (c) Coordinate Systems (p=pelvis, 0=thorax, 1=L1,…,5=L5)
cp1 = Transpose[R3[Ap[t]].R2[Bp[t]].R1[Tp[t]]].n1;
cp2 = Transpose[R3[Ap[t]].R2[Bp[t]].R1[Tp[t]]].n2;
cp3 = Transpose[R3[Ap[t]].R2[Bp[t]].R1[Tp[t]]].n3;
c51 = Transpose[R3[A5[t]].R2[B5[t]].R1[T5[t]]].n1;
c52 = Transpose[R3[A5[t]].R2[B5[t]].R1[T5[t]]].n2;
c53 = Transpose[R3[A5[t]].R2[B5[t]].R1[T5[t]]].n3;
c41 = Transpose[R3[A4[t]].R2[B4[t]].R1[T4[t]]].n1;
c42 = Transpose[R3[A4[t]].R2[B4[t]].R1[T4[t]]].n2;
c43 = Transpose[R3[A4[t]].R2[B4[t]].R1[T4[t]]].n3;
c31 = Transpose[R3[A3[t]].R2[B3[t]].R1[T3[t]]].n1;
c32 = Transpose[R3[A3[t]].R2[B3[t]].R1[T3[t]]].n2;
c33 = Transpose[R3[A3[t]].R2[B3[t]].R1[T3[t]]].n3;
c21 = Transpose[R3[A2[t]].R2[B2[t]].R1[T2[t]]].n1;
```

```
c22 = Transpose[R3[A2[t]].R2[B2[t]].R1[T2[t]]].n2;
c23 = Transpose[R3[A2[t]].R2[B2[t]].R1[T2[t]]].n3;
c11 = Transpose[R3[A1[t]].R2[B1[t]].R1[T1[t]]].n1;
c12 = Transpose[R3[A1[t]].R2[B1[t]].R1[T1[t]]].n2;
c13 = Transpose[R3[A1[t]].R2[B1[t]].R1[T1[t]]].n3;
c01 = Transpose[R3[A0[t]].R2[B0[t]].R1[T0[t]]].n1;
c02 = Transpose[R3[A0[t]].R2[B0[t]].R1[T0[t]]].n2;
c03 = Transpose[R3[A0[t]].R2[B0[t]].R1[T0[t]]].n3;

------------------Define Angular Velocities
NωAp = Tp'[t] Transpose[R1[Tp[t]]].n1;
ApωBp = Bp'[t] Transpose[R2[Bp[t]].R1[Tp[t]]].n2;
BpωCp = Ap'[t] Transpose[R3[Ap[t]].R2[Bp[t]].R1[Tp[t]]].n3;
NωA5 = T5'[t] Transpose[R1[T5[t]]].n1;
A5ωB5 = B5'[t] Transpose[R2[B5[t]].R1[T5[t]]].n2;
B5ωC5 = A5'[t] Transpose[R3[A5[t]].R2[B5[t]].R1[T5[t]]].n3;
NωA4 = T4'[t] Transpose[R1[T4[t]]].n1;
A4ωB4 = B4'[t] Transpose[R2[B4[t]].R1[T4[t]]].n2;
B4ωC4 = A4'[t] Transpose[R3[A4[t]].R2[B4[t]].R1[T4[t]]].n3;
NωA3 = T3'[t] Transpose[R1[T3[t]]].n1;
A3ωB3 = B3'[t] Transpose[R2[B3[t]].R1[T3[t]]].n2;
B3ωC3 = A3'[t] Transpose[R3[A3[t]].R2[B3[t]].R1[T3[t]]].n3;
NωA2 = T2'[t] Transpose[R1[T2[t]]].n1;
A2ωB2 = B2'[t] Transpose[R2[B2[t]].R1[T2[t]]].n2;
B2ωC2 = A2'[t] Transpose[R3[A2[t]].R2[B2[t]].R1[T2[t]]].n3;
NωA1 = T1'[t] Transpose[R1[T1[t]]].n1;
A1ωB1 = B1'[t] Transpose[R2[B1[t]].R1[T1[t]]].n2;
B1ωC1 = A1'[t] Transpose[R3[A1[t]].R2[B1[t]].R1[T1[t]]].n3;
NωA0 = T0'[t] Transpose[R1[T0[t]]].n1;
A0ωB0 = B0'[t] Transpose[R2[B0[t]].R1[T0[t]]].n2;
B0ωC0 = A0'[t] Transpose[R3[A0[t]].R2[B0[t]].R1[T0[t]]].n3;

------------------Define Relative Angular Velocity Vectors
NωCp = NωAp + ApωBp + BpωCp;
NωC5 = NωA5 + A5ωB5 + B5ωC5;
NωC4 = NωA4 + A4ωB4 + B4ωC4;
NωC3 = NωA3 + A3ωB3 + B3ωC3;
NωC2 = NωA2 + A2ωB2 + B2ωC2;
NωC1 = NωA1 + A1ωB1 + B1ωC1;
NωC0 = NωA0 + A0ωB0 + B0ωC0;
NωCprel = NωCp.Transpose[R3[Ap[t]].R2[Bp[t]].R1[Tp[t]]];
NωC5rel = NωC5.Transpose[R3[A5[t]].R2[B5[t]].R1[T5[t]]];
NωC4rel = NωC4.Transpose[R3[A4[t]].R2[B4[t]].R1[T4[t]]];
NωC3rel = NωC3.Transpose[R3[A3[t]].R2[B3[t]].R1[T3[t]]];
NωC2rel = NωC2.Transpose[R3[A2[t]].R2[B2[t]].R1[T2[t]]];
NωC1rel = NωC1.Transpose[R3[A1[t]].R2[B1[t]].R1[T1[t]]];
NωC0rel = NωC0.Transpose[R3[A0[t]].R2[B0[t]].R1[T0[t]]];

------------------Define Origin of Each Body
Orp = 0 n1;
Or5 = Orp + cent1 cp1 + cent2 cp2 + cent3 cp3;
Or4 = Or5 + L5 c53;
Or3 = Or4 + L4 c43;
Or2 = Or3 + L3 c33;
Or1 = Or2 + L2 c23;
Or0 = Or1 + L1 c13;

------------------Define Center of Mass (COM) of Each Body
G5 = Or5 + L5/2 c53;
G4 = Or4 + L4/2 c43;
G3 = Or3 + L3/2 c33;
G2 = Or2 + L2/2 c23;
G1 = Or1 + L1/2 c13;
G0 = Or0 + L0/2 c03;

------------------Define Velocities of COM
vG5 = D[G5, t];
vG4 = D[G4, t];
vG3 = D[G3, t];
vG2 = D[G2, t];
```

```
vG1 = D[G1, t];
vG0 = D[G0, t];

----------Define 3D Inertial Matrix of Each Body
I5 = {{1/4 m5 lb^2 + 1/12 m5 L5^2, 0, 0}, {0, 1/4 m5 la^2 + 1/12 m5 L5^2,
      0}, {0, 0, 1/4 m5 (la^2 + lb^2)}};
I4 = {{1/4 m4 lb^2 + 1/12 m4 L4^2, 0, 0}, {0,
    1/4 m4 la^2 + 1/12 m4 L4^2, 0}, {0, 0, 1/4 m4 (la^2 + lb^2)}};
I3 = {{1/4 m3 lb^2 + 1/12 m3 L3^2, 0, 0}, {0, 1/4 m3 la^2 + 1/12 m3 L3^2,
      0}, {0, 0, 1/4 m3 (la^2 + lb^2)}};
I2 = {{1/4 m2 lb^2 + 1/12 m2 L2^2, 0, 0}, {0,
    1/4 m2 la^2 + 1/12 m2 L2^2, 0}, {0, 0, 1/4 m2 (la^2 + lb^2)}};
I1 = {{1/4 m1 lb^2 + 1/12 m1 L1^2, 0, 0}, {0, 1/4 m1 la^2 + 1/12 m1 L1^2,
      0}, {0, 0, 1/4 m1 (la^2 + lb^2)}};
I0 = {{1/4 m0 lb^2 + 1/12 m0 L0^2, 0, 0}, {0,
    1/4 m0 la^2 + 1/12 m0 L0^2, 0}, {0, 0, 1/4 m0 (la^2 + lb^2)}};

----------Define Kinetic Energy
T = 1/2 m5 vG5.vG5 + 1/2 m4 vG4.vG4 + 1/2 m3 vG3.vG3 + 1/2 m2 vG2.vG2 + 1/2
    m1 vG1.vG1 +
    1/2 m0 vG0.vG0 + 1/2 NωC5rel.I5.NωC5rel + 1/2
    NωC4rel.I4.NωC4rel + 1/2 NωC3rel.I3.NωC3rel +
    1/2 NωC2rel.I2.NωC2rel + 1/2
    NωC1rel.I1.NωC1rel + 1/2 NωC0rel.I0.NωC0rel;

----------Define Potential Energy
V = g m5 G5[[3]] + g m4 G4[[3]] + g m3 G3[[3]] + g m2 G2[[3]] + g
    m1 G1[[3]] + g m0 G0[[3]];

----------Take Lagrange Derivative for Equations of Motion (EM)
(*Lagrange Derivative*)
EL[θ_] := D[D[T, θ'[t]], t] - D[T, θ[t]] + D[V, θ[t]];
ELT5 = EL[T5]; ELB5 = EL[B5]; ELA5 = EL[A5];
ELT4 = EL[T4]; ELB4 = EL[B4]; ELA4 = EL[A4];
ELT3 = EL[T3]; ELB3 = EL[B3]; ELA3 = EL[A3];
ELT2 = EL[T2]; ELB2 = EL[B2]; ELA2 = EL[A2];
ELT1 = EL[T1]; ELB1 = EL[B1]; ELA1 = EL[A1];
ELT0 = EL[T0]; ELB0 = EL[B0]; ELA0 = EL[A0];

----------Define State Vector q
q[t] = {T5[t],
    B5[t], A5[t], T4[t], B4[t], A4[t], T3[t], B3[t], A3[t], T2[t], B2[t],
    A2[t], T1[t], B1[t], A1[t], T0[t], B0[t], A0[t]};
q'[t] = D[q[t], t];
q''[t] = D[q'[t], t];

----------Define Mass Matrix Mm
Mdd[EL_] := {D[EL, T5''[
   t]], D[EL, B5''[t]], D[EL, A5''[t]], D[EL, T4''[t]], D[EL, B4''[
      t]], D[EL, A4''[t]], D[EL, T3''[t]], D[EL, B3''[t]], D[EL, A3''[
      t]], D[EL, T2''[t]], D[EL, B2''[t]], D[EL, A2''[t]], D[EL, T1''[
      t]], D[EL, B1''[t]], D[EL, A1''[t]], D[EL, T0''[t]], D[EL, B0''[
      t]], D[EL, A0''[t]]} // Simplify;
Mm = {Mdd[
   ELT5], Mdd[ELB5], Mdd[ELA5], Mdd[ELT4], Mdd[ELB4], Mdd[ELA4], Mdd[
      ELT3], Mdd[ELB3], Mdd[ELA3], Mdd[ELT2], Mdd[ELB2], Mdd[ELA2], Mdd[
      ELT1], Mdd[ELB1], Mdd[ELA1], Mdd[ELT0], Mdd[ELB0], Mdd[ELA0]} //
      Simplify;

----------Define Gravity Vector Gm
Gm = {g D[
   ELT5, g], g D[ELB5, g], g D[ELA5, g], g D[ELT4, g], g D[
      ELB4, g], g D[ELA4, g], g D[ELT3, g], g D[ELB3, g], g D[
      ELA3, g], g D[ELT2, g], g D[ELB2, g], g D[ELA2, g], g D[
      ELT1, g], g D[ELB1, g], g D[ELA1, g], g D[ELT0, g], g D[
      ELB0, g], g D[ELA0, g]} // Simplify;

----------Define Crossterms/Coriolis Vector Cm
CSimp = {T5''[t] -> 0, B5''[t] -> 0, A5''[t] -> 0, T4''[t] -> 0, B4''[t] -> 0,
     A4''[t] -> 0, T3''[t] -> 0, B3''[t] -> 0, A3''[t] -> 0, T2''[t] -> 0,
```

```
   B2''[t] -> 0, A2''[t] -> 0, T1''[t] -> 0, B1''[t] -> 0, A1''[
     t] -> 0, T0''[t] -> 0, B0''[
     t] -> 0, A0''[t] -> 0, cent1 -> 0, cent2 -> 0};
CmT5 = ELT5 - Mm[[1]].q''[t] - Gm[[1]];
CmB5 = ELB5 - Mm[[2]].q''[t] - Gm[[2]];
CmA5 = ELA5 - Mm[[3]].q''[t] - Gm[[3]];
CmT4 = ELT4 - Mm[[4]].q''[t] - Gm[[4]];
CmB4 = ELB4 - Mm[[5]].q''[t] - Gm[[5]];
CmA4 = ELA4 - Mm[[6]].q''[t] - Gm[[6]];
CmT3 = ELT3 - Mm[[7]].q''[t] - Gm[[7]];
CmB3 = ELB3 - Mm[[8]].q''[t] - Gm[[8]];
CmA3 = ELA3 - Mm[[9]].q''[t] - Gm[[9]];
CmT2 = ELT2 - Mm[[10]].q''[t] - Gm[[10]];
CmB2 = ELB2 - Mm[[11]].q''[t] - Gm[[11]];
CmA2 = ELA2 - Mm[[12]].q''[t] - Gm[[12]];
CmT1 = ELT1 - Mm[[13]].q''[t] - Gm[[13]];
CmB1 = ELB1 - Mm[[14]].q''[t] - Gm[[14]];
CmA1 = ELA1 - Mm[[15]].q''[t] - Gm[[15]];
CmT0 = ELT0 - Mm[[16]].q''[t] - Gm[[16]];
CmB0 = ELB0 - Mm[[17]].q''[t] - Gm[[17]];
CmA0 = ELA0 - Mm[[18]].q''[t] - Gm[[18]];
Cm = {CmT5, CmB5, CmA5, CmT4, CmB4, CmA4, CmT3, CmB3, CmA3, CmT2, CmB2,
      CmA2, CmT1, CmB1, CmA1, CmT0, CmB0, CmA0} /. CSimp;

------------------Define Partial Derivatives (For Generalized Forces Calculation)
QL[θ_] := {D[D[Or5,
   t], θ'[t]], D[D[Or4, t], θ'[t]], D[D[Or3,
    t], θ'[t]], D[D[Or2, t], θ'[t]], D[D[Or1,
    t], θ'[t]], D[D[Or0, t], θ'[t]]};
QR[θ_] := {D[NωC5, θ'[t]], D[NωC4, θ'[t]], D[NωC3,
    θ'[t]], D[NωC2, θ'[t]], D[NωC1, θ'[t]], D[NωC0, θ'[t]]};
QLθ5 = QL[T5];
QLβ5 = QL[B5]; QLα5 = QL[A5];
QLθ4 = QL[T4]; QLβ4 = QL[B4]; QLα4 = QL[A4];
QLθ3 = QL[T3]; QLβ3 = QL[B3]; QLα3 = QL[A3];
QLθ2 = QL[T2]; QLβ2 = QL[B2]; QLα2 = QL[A2];
QLθ1 = QL[T1]; QLβ1 = QL[B1]; QLα1 = QL[A1];
QLθ0 = QL[T0]; QLβ0 = QL[B0]; QLα0 = QL[A0];
QRθ5 = QR[T5]; QRβ5 = QR[B5]; QRα5 = QR[A5];
QRθ4 = QR[T4]; QRβ4 = QR[B4]; QRα4 = QR[A4];
QRθ3 = QR[T3]; QRβ3 = QR[B3]; QRα3 = QR[A3];
QRθ2 = QR[T2]; QRβ2 = QR[B2]; QRα2 = QR[A2];
QRθ1 = QR[T1]; QRβ1 = QR[B1]; QRα1 = QR[A1];
QRθ0 = QR[T0]; QRβ0 = QR[B0]; QRα0 = QR[A0];

------------------Vector of Partial Derivatives (Again, For Generalized Forces)
QLm = {{QLθ5}, {QLβ5}, {QLα5}, {QLθ4}, {QLβ4}, {QLα4}, {QLθ3}, {QLβ3}, \
{QLα3}, {QLθ2}, {QLβ2}, {QLα2}, {QLθ1}, {QLβ1}, {QLα1}, {QLθ0}, {QLβ0}, \
{QLα0}} // Simplify;
QRm = {{QRθ5}, {QRβ5}, {QRα5}, {QRθ4}, {QRβ4}, {QRα4}, {QRθ3}, {QRβ3}, {
      QRα3}, {QRθ2}, {QRβ2}, {QRα2}, {QRθ1}, {QRβ1}, {QRα1}, {QRθ0}, {
     QRβ0}, {QRα0}} // Simplify;

------------------Vector of Body Origins (For Use in Generalized Force Algorithm)
Om = {Or5, Or4, Or3, Or2, Or1, Or0};
```

ii. Numerical Methods

All experiments on the spine model (performed or suggested) require working with the model using numerical methods. Matlab was used as the primary program for its ease of use and the availability of several 'canned' analysis tools that further simplify its use.

Matlab was the programming language of choice due to its ease of use and the availability of tools. Among the multitude of tools that were used were: eig (an

eigenvalue solver), fmincon (a nonlinear constrained optimization algorithm), ode45 (a variable step instantaneous integrator), and dde23 (a variable step delayed differential integrator). One drawback of Matlab is that it is an interpreted instead of a compiled language. It is therefore slow and inefficient when compared with other languages. The resulting dynamics from Appendix A.i were too long for matlab functions. These equations, which would cover 800 pages if printed, were therefore implemented as binary mex files which matlab accessed for numerical values.

Mex files are compiled binary files, which in matlab have an interface identical to matlab functions. The mex files written for this project were all written in c. They were compiled with the Microsoft Visual C (MSVC) compiler, although any compiler would have worked. However, MSVC employs optimization algorithms that result in the compiled binary being incredibly efficient (for example, it avoids computing the same value twice).

Use of mex files in place of matlab files is not without drawbacks. Mex files require the programmer to have some experience with real data types (matlab does not enforce strong datatyping, for instance requiring variables to be declared as: int, double, float, char, etc). They also require the programmer to work directly with memory allocation, and passing by reference. While managing memory is not terribly difficult, small errors can cause 'undetermined symptoms' as your programs overwrite or delete random blocks of memory. Writing mex files in c requires including the matlab library "mex.h", which has all of the functions required to interface with matlab. Matlab prefers if memory assignment and deletion happens through their functions, so that variables reside in memory Matlab has allocated.

Example Mex File: ExMex.c

This file will demonstrate by example how to program a mex binary for Matlab. The main function is mexFunction(), while the sub-function housing our actual program is vAdd(). Also note the presence of functions prefixed with 'mx', these are functions defined in the mex.h library and interface with Matlab directly. For instance, mxCalloc allocates memory in the Matlab stack, while mxFree removes the allocation.

To compile in Matlab, type:
Setup compiler
mex -setup
Compile the function
mex ExMex.c
Run the function
c=exmex(ones(10,1),2*ones(10,1));

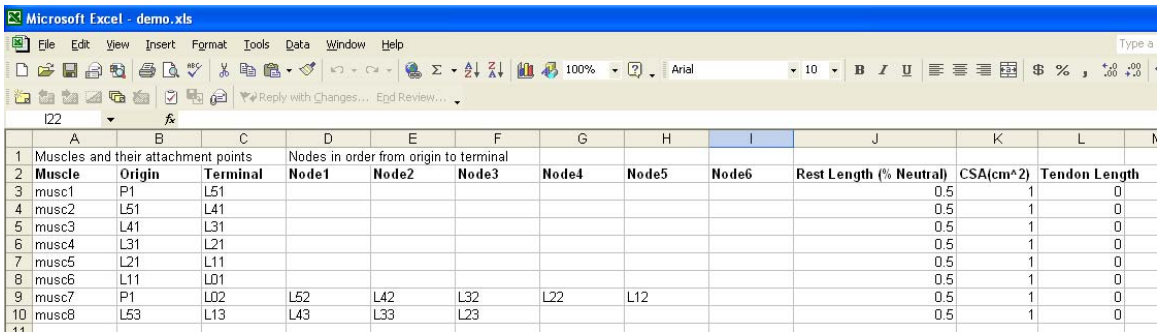
ExMex.c

```
/*==========================================================
  Add two vectors, display the sum of all elements in c
  c=ExMex(a,b)
  Vectors a and b must be the same size
 *==========================================================*/
#include <math.h>
#include "mex.h"

/* Input Arguments */
#define a_IN            prhs[0]
#define b_IN            prhs[1]
/* Output Arguments */
#define c_OUT                   plhs[0]

#if !defined(MAX)
#define    MAX(A, B)            ((A) > (B) ? (A) : (B))
#endif
#if !defined(MIN)
#define    MIN(A, B) ((A) < (B) ? (A) : (B))
#endif

void vAdd(double *a, double *b, double *c, int len, int row)
{
        double *TV;                                     //Temporary Vector Pointer
        int i;
        double sum=0;
        TV=mxCalloc(100,sizeof(double));                //Temp Vector Allocate

        printf("Beginning Addition\n");
        for(i=0;i<(len*row);i++)
        {
                c[i]=a[i]+b[i];
                sum=sum+c[i];
        }
        printf("Addition Ended, total: %6f \n", sum);

        mxFree(TV);                                     //Temp Vector DeAllocate
        return;
}

void mexFunction( int nlhs, mxArray *plhs[], int nrhs, const mxArray*prhs[] )
{
   double *a, *b, *c;
        int col, row;

   /* Check for proper number of arguments */
   if (nrhs != 2) {
        mexErrMsgTxt("2 input arguments required.");
   } else if (nlhs > 1) {
        mexErrMsgTxt("Too many output arguments (1 max).");
   }else if (mxGetM(a_IN)>mxGetM(b_IN)||mxGetM(b_IN)>mxGetM(a_IN)) {
        mexErrMsgTxt("Vectors must be same size");
   }else if (mxGetN(a_IN)>mxGetN(b_IN)||mxGetN(b_IN)>mxGetN(a_IN)) {
        mexErrMsgTxt("Vectors must be same size");
   }
        //Length of vector in
        col = mxGetM(a_IN);
```

```
        row = mxGetN(a_IN);
    /* Assign pointers to the various parameters */
        a = mxGetPr(a_IN);
        b = mxGetPr(b_IN);
    /* Create a matrix for the return argument */
        c_OUT = mxCreateDoubleMatrix(col, row, mxREAL);
        c = mxGetPr(c_OUT);
    /* Do the actual computations in a subroutine */
    vAdd(a,b,c,col,row);
    return;
}
```

Muscle Programming

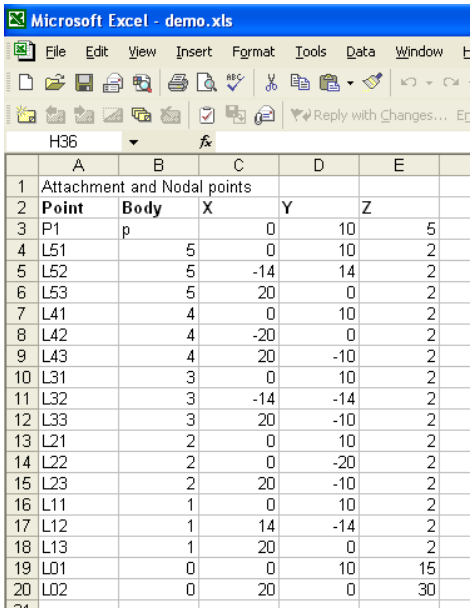
The muscle anatomy may be programmed using excel workbooks that are parsed by the spine model in Matlab. Below is an example of Sheet1 in the muscle workbook. As one can see attachment points are actually just descriptions of points explicitly specified in Sheet2. This allows for ease in programming the muscle configuration.



Microsoft Excel - demo.xls

	A	B	C	D	E	F	G	H	I	J	K	L
1	Muscles and their attachment points			Nodes in order from origin to terminal								
2	**Muscle**	**Origin**	**Terminal**	**Node1**	**Node2**	**Node3**	**Node4**	**Node5**	**Node6**	**Rest Length (% Neutral)**	**CSA(cm^2)**	**Tendon Length**
3	musc1	P1	L51							0.5	1	0
4	musc2	L51	L41							0.5	1	0
5	musc3	L41	L31							0.5	1	0
6	musc4	L31	L21							0.5	1	0
7	musc5	L21	L11							0.5	1	0
8	musc6	L11	L01							0.5	1	0
9	musc7	P1	L02	L52	L42	L32	L22	L12		0.5	1	0
10	musc8	L53	L13	L43	L33	L23				0.5	1	0

Below is an example of Sheet2 in the muscle workbook. In this sheet, the points are assigned a body and a location on that body. All locations are in centimeters (cm).



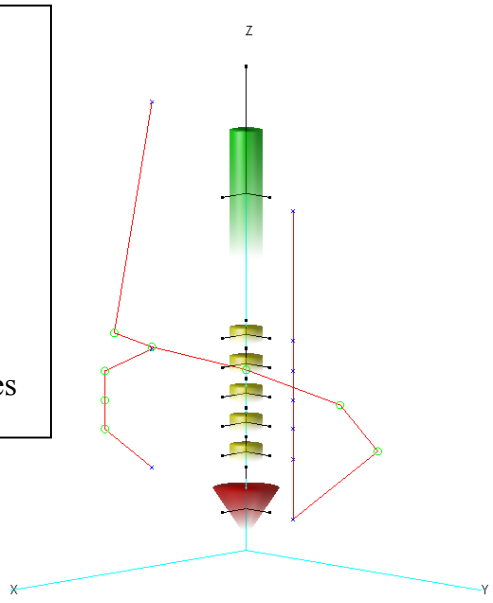
Microsoft Excel - demo.xls

	A	B	C	D	E
1	Attachment and Nodal points				
2	**Point**	**Body**	**X**	**Y**	**Z**
3	P1	p	0	10	5
4	L51	5	0	10	2
5	L52	5	-14	14	2
6	L53	5	20	0	2
7	L41	4	0	10	2
8	L42	4	-20	0	2
9	L43	4	20	-10	2
10	L31	3	0	10	2
11	L32	3	-14	-14	2
12	L33	3	20	-10	2
13	L21	2	0	10	2
14	L22	2	0	-20	2
15	L23	2	20	-10	2
16	L11	1	0	10	2
17	L12	1	14	-14	2
18	L13	1	20	0	2
19	L01	0	0	10	15
20	L02	0	20	0	30

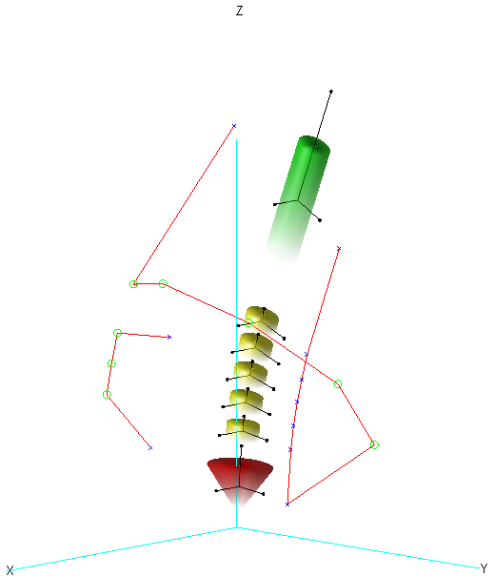
Body specifications

L5:	Body 5
L4:	Body 4
L3:	Body 3
L2:	Body 2
L1:	Body 1
Up. Trunk:	Body 0
Pelvis:	Body p
Absolute:	Body n

Table Ap.A.1 – Coordinate frame codes



The above example muscle anatomy contains 8 muscles. The anatomy is illustrated in two kinematic positions at the right. The first 6 muscles are attached end to end up the abdominal section, while the remaining two demonstrate the use of via points (or nodes). Muscle 7 swirls around the body, with each node traversing sequential vertebrae. Muscle 8 (far left) shows a realistic use of nodal points by a muscle in the back. In the figures nodal points are drawn with o's and muscle terminals are drawn with x's.



iii. Optimization, Integration

The optimization algorithm found the muscle activation set α, which minimized metabolic power while ensuring the model was in equilibrium and stable. Stability was determined by ensuring that the eigenvalues of the zero delay Jacobian had negative real components (see section 3.C.i). The Mathworks algorithm fmincon was used in this model. To program this algorithm the following code was used:

In optimizationscript.m

```
…
   eps=0.00001*pi/180;    %Epsilon angle and angular vel. disturbance for stability
   %-----Optimization Settings
         Aeq=[];
         Beq=[];
         %Intialize Globals
         PrevEig=zeros(18,1);
         JPrevEig=zeros(36,1);
         optiter=0;
         cequal=zeros(18,1);
         len=size(Mc,1);
         Ain=-eye(size(Mc,1));
         Bin=zeros(len,1);
         Ov=zeros(len,1);
         lb=zeros(len,1);
         hb=ones(len,1);
   Un=ones(size(Mc,1),1);
   S=Stateo;
   Pang=Pango;
%-----Equality Assembly
   Uo=zeros(size(Mc,1),1);
   Aeq=zeros(18,size(Mc,1));
   Beq=zeros(18,1);
   Beq=-spineaccP(0,S,Pang,AAPOA,Mc,Ec,Lo,CSA,Uo,Pn,EFo,Vars,IVD);
```

```
    for i=1:size(Mc,1)
        Uo=zeros(size(Mc,1),1);
        Uo(i)=1;
        Aeq(:,i)=spineaccA(0,S,Pang,AAPOA,Mc,Ec,Lo,CSA,Uo,Pn,EFo,Vars,IVD);
    end
%-----Jacobian Assembly
    Jpi=zeros(36,36);
    Jpi(1:18,19:36)=eye(18,18);
    Jai=zeros(size(Mc,1),36,36);
    for h=1:36
        Sd=S;
        Sd(h)=Sd(h)+eps;
        for i=1:size(Mc,1)
            Uo=zeros(size(Mc,1),1);
            Uo(i)=1;
            Jai(i,19:36,h)=(spineaccA(0,Sd,Pang,AAPOA,Mc,Ec,Lo,CSA,Uo,Pn,EFo,Vars,IVD)-Aeq(:,i))/eps;
        end
        Uo=zeros(size(Mc,1),1);
        Jpi(19:36,h)=(spineaccP(0,Sd,Pang,AAPOA,Mc,Ec,Lo,CSA,Uo,Pn,EFo,Vars,IVD)+Beq)/eps;
    end
    %-----Reflex
    Jpr=zeros(36,36);
    Jar=zeros(size(Mc,1),36,36);
    for h=1:36
        Sd=S;
        Sd(h)=Sd(h)+eps;
        for i=1:size(Mc,1)
            Uo=zeros(size(Mc,1),1);
            Uo(i)=1;
            Jar(i,19:36,h)=(spineaccR(0,Sd,Pang,AAPOA,Mc,Ec,Lo,CSA,Uo,Pn,EFo,Vars,IVD))/eps;
        end
    end
    %-----For Optimization (no time delay assumption)
    Jp=Jpi+Jpr;
    Ja=Jai+Jar;
%-----Stiffness Matrix Assembly (NOT CURRENTLY USED!!)
    %-----Intrinsic
    Kpi=zeros(18,18);
    Kai=zeros(size(Mc,1),18,18);
    for h=1:18
        Sd=S;
        Sd(h)=Sd(h)+eps;
        for i=1:size(Mc,1)
            Uo=zeros(size(Mc,1),1);
            Uo(i)=1;
            Kai(i,:,h)=(spineaccA(0,Sd,Pang,AAPOA,Mc,Ec,Lo,CSA,Uo,Pn,EFo,Vars,IVD)-Aeq(:,i))/eps;
        end
        Uo=zeros(size(Mc,1),1);
        Kpi(:,h)=(spineaccP(0,Sd,Pang,AAPOA,Mc,Ec,Lo,CSA,Uo,Pn,EFo,Vars,IVD)+Beq)/eps;
    end
    %-----Reflex
    Kpr=zeros(18,18);
    Kar=zeros(size(Mc,1),18,18);
    for h=1:18
        Sd=S;
        Sd(h)=Sd(h)+eps;
        for i=1:size(Mc,1)
            Uo=zeros(size(Mc,1),1);
            Uo(i)=1;
            Kar(i,:,h)=(spineaccR(0,Sd,Pang,AAPOA,Mc,Ec,Lo,CSA,Uo,Pn,EFo,Vars,IVD))/eps;
        end
    end
    %-----For Optimization (no time delay assumption)
    Kp=Kpi+Kpr;
    Ka=Kai+Kar;
   clear options
   options=optimset('LargeScale','off','MaxFunEvals',1e5,'MaxIter',200,'TolX',1e-6,'TolFun',1e-3,'TolCon',1e-
4,'Display','iter','Diagnostics','off');
%-----Run Optimization routine
    Un=rand(size(Mc,1),1);      %Random initial seed
```

```
   options=optimset('LargeScale','off','MaxFunEvals',1e5,'MaxIter',400,'TolX',1e-3,'TolFun',1e-3,'TolCon',1e-
4,'Display','off','Diagnostics','off','MaxSQPIter',2000);
[Un,fval,exitflag,output,lambda,grad,hessian]=fmincon(@costfun,Un,Ain,Bin,Aeq,Beq,lb,hb,@nlcon,options,Stateo,Kp,Ka,Jp,Ja,Lo,
CSA);
…
```

costfun.m

```
function Out=costfun(Un,S,Kp,Ka,Jp,Ja,Lo,CSA)
%Cost function for muscle contraction optimization
%-----Muscle Maintanance Heat Rate
%    M~mi*Ui
%    mi=CSA.*Lo*rho - mass
%    Out=watts (J/s)
   rho=1000; %kg/m^3 (approx same as water)
   mass=rho*(CSA/100/100).*Lo;
   aslow=sin(pi/2*Un);
   afast=1-cos(pi/2*Un);
   %Minimize Power
   Out=sum(mass.*(74*0.5*aslow+111*0.5*afast));    %W=J/s
```

nlcon.m

```
function [c,ceq]=nlcon(Un,S,Kp,Ka,Jp,Ja,Lo,CSA)
%[c,ceq]=nlcon(Un,S,Pang,NLAeq,NLBeq,NLA,NLB,eps,M,V,FextFo,Pos,MapIN,MuscL,FPos,FMapIN)
%Nonlinear constraints
%c: nonlinear inequalities c(Unum)<0
%ceq: nonlinear equalities ceq(Unum)=0
%This function runs after cost function

% global pPrevEig nPrevEig cequal optiter
global PrevEig JPrevEig optiter
   optiter=optiter+1;
%--------------Equality
   ceq=0;
%--------------Inequality:PrevEig current, since costfun run first
%-----Jacobian Eigenvalues
   J=Jp;
   for i=1:36
      J(19:36,i)=J(19:36,i)+(Un'*Ja(:,19:36,i))';
   end
      ap=eig(J);
      b=zeros(36,1);
      p=zeros(36,36);
      for j=1:36
         dist=JPrevEig-ap(j);
         p(j,:)=sqrt(real(dist).^2+imag(dist).^2)';
      end
      for j=1:36
         [q,indc]=min(min(p));
         [q,indr]=min(p(:,indc));
         b(indc)=ap(indr);
         p(:,indc)=1e12;
         p(indr,:)=1e12;
      end
      JPrevEig=b;
%-----Stiffness Eigenvalues
%     K=Kp;
%     for i=1:18
%        K(:,i)=K(:,i)+(Un'*Ka(:,:,i))';
%     end
%        ap=eig(K);
%        b=zeros(18,1);
%        p=zeros(18,18);
%        for j=1:18
%           dist=PrevEig-ap(j);
%           p(j,:)=sqrt(real(dist).^2+imag(dist).^2)';
%        end
%        for j=1:18
%           [q,indc]=min(min(p));
%           [q,indr]=min(p(:,indc));
%           b(indc)=ap(indr);
```

```
%           p(:,indc)=1e12;
%           p(indr,:)=1e12;
%       end
%       PrevEig=b;
%    eigs=eig(J);
%    c=real(eig(J));
   c=real(b)+1e-4;      %Allow 1e-4 slop
%    c=-1;               %Enable to evaluate equilibrium only
%    ceq=imag(b);
%-----
%    optplot(Un,c);
```

Once the optimized muscle activations have been found, the fully nonlinear time delayed system may be simulated. In our implementation the integrator dde23 was used. This is the only integrator included with Matlab that allows for delayed differential equations. The code to execute the integration is below.

from spinesim.m

```
            %-----Add Kinematic Disturbance for simulation
               StateDist=Stateo;
               StateDist(1)=StateDist(1)+0.01*pi/180;
            %-----Define Simulation Timing
                  Tstep=0.01;
                  Tfinal=5;
                  Tode=0:Tstep:Tfinal;
                  Lags=[Td];
               %-----Set Simulation Options
                  options=odeset('OutputFcn',@odebarplot,'Events',@extrema,'RelTol',1e-6,'AbsTol',1e-9,'Stats','on');
               %-----Run Simulation
                  sol=dde23(@spinedyn,Lags,StateDist,Tode,options,Pango,AAPOA,Mc,Ec,Lo,CSA,Un,Pn,EFo,Vars,IVD);
                  Time=sol.x;
                  State=sol.y';
```

spinedyn.m

```
function [dS]=spinedyn(t,S,Z,Pango,AAPOA,Mc,Ec,Lo,CSA,Uo,Po,EFo,Vars,IVD)
%Spine simulation program
%Dynamics file
%-----Define Open Loop Pelvis Kinematics
   Pang=Pango;
%-----Define Muscle Activation and External Loads
%      This is where open loop code or control programs
%      should be implemented.
   U=Uo;
   P=Po;
   EF=EFo;
%Force Perturbation
%    if(t<0)
%       EF=EFo;
%    elseif(t<0.01)
%       EF(1)=EF(1);
%    else
%       EF=EFo;
%    end
%-----Kinematics-Rotation matrices, muscle lengths/vels, etc..
   [Ma,Ea,ML,MV,Gl,Gr,Rm,Or,dRm,dOr,RBV,dRBV] = kinematicsc(S,Pang,Vars,Mc,Ec);
   [MaR,EaR,MLR,MVR,GlR,GrR,RmR,OrR,dRmR,dOrR,RBVR,dRBVR] = kinematicsc(Z,Pang,Vars,Mc,Ec);
%-----Solve for Muscle Forces
   A=S(37:end);
   [MF,dA]=muscmodel(U,P,A,ML,MV,Lo,CSA);
   [MFR,dAR]=muscmodelR(U,P,A,MLR,MVR,Lo,CSA);
   MF=MF+MFR;
```

```
%-----Passive Dynamics (Intervertebral Disks)
   [PRQ]=intervertc(RBV,dRBV,AAPOA,IVD);
   PRQ=PRQ(1:6,:);
%-----Read Model Dynamic Matrices
   [Mm,Cv,Gv]=dynmatsc(S,Pang,Vars);
%-----Solve Generalized Forces for given Muscle and Ext. Loads
   Q=genforcesc(Ma,real(MF),Ea,EF,PRQ,Gl,Gr,Or);
%-----Solve Differential Kinematics
   qdd=Mm\(Q-Cv-Gv);
%-----Assemble Differential State Vector
   dS=zeros(length(S),1);
   dS(1:18)=S(19:36);
   dS(19:36)=qdd;
   dS(37:end)=dA;
```

---

iv. Model Anatomy

Sheet 1 and 2 of the muscle anatomy are printed in this section. The anatomy is identical to that supplied in Cholewicki et al (1995), however coordinates have been transformed into their body fixed coordinates. Cross sectional areas may be found in Stokes et al (2002).

Table Ap.A.2 – Muscle anatomy sheet 1: muscles in terms of coordinate descriptions

Muscles and their attachment points			Nodes in order from origin to terminal						
Muscle	**Origin**	**Terminal**	**Node1**	**Node2**	**Node3**	**Node4**	**Node5**	**Node6**	**CSA(cm^2)**
RRectAb	RPEL3	RRIB3							5
RextOb1	RPEL4	RRIB4							3.4
RextOb2	RPEL5	RRIB5							3.4
RIntOb1	RPEL6	RRIB6							2.8
RIntOb2	RPEL7	RRIB7							2.8
RparsL5	RPEL9	RL53							2
RparsL4	RPEL9	RL43	RL410						2
RParsL3	RPEL9	RL33	RL410						2
RParsL2	RPEL9	RL23	RL411	RL310	RL210	RL19			2
RParsL1	RPEL9	RL13	RL412	RL311	RL211	RL110			2
RIlioLum	RPEL10	RRIB8	RL413	RL312	RL212	RL111	RRIB23		1
RLongTP	RPEL11	RRIB9	RL312	RL212	RL111	RRIB23			1.25
RLongTL5	RL54	RRIB10	RL212	RL111	RRIB23				1.25
RLongTL4	RL44	RRIB11	RL111	RRIB23					1.25
RLongTL3	RL34	RRIB12							1.25
RLongTL2	RL24	RRIB13							1.25
RLongTL1	RL14	RRIB14							1.25
RQuadLP	RPEL12	RRIB15							2.8
RQuadL1	RPEL12	RL15							2.8
RQuadL2	RPEL12	RL25							2.8
RQuadL3	RPEL12	RL35							2.8
RQuadL4	RPEL12	RL45							2.8
RLatDP	RPEL16	RRIB17							2.5
RLatDL5	RPEL15	RRIB17	RRIB22						2.5

RLatDL4	RL48	RRIB17	RRIB21					2.5
RLatDL3	RL38	RRIB17	RRIB20					2.5
RLatDL2	RL28	RRIB17	RRIB19					2.5
RLatDL1	RL17	RRIB17	RRIB18					2.5
RMultPL5	RPEL14	RL55						1
RMultPL4	RPEL14	RL46						1
RMultPL3	RPEL13	RL36						1
RMultPL2	RPEL13	RL26						1
RMultPL1	RPEL13	RL16						1
RMultL5L3	RL56	RL36						1
RMultL5L2	RL56	RL26						1
RMultL5L1	RL56	RL16						1
RMultL4L2	RL47	RL26						1
RMultL4L1	RL47	RL16						1
RMultL3L1	RL37	RL16						1
RMultL2T12	RL27	RRIB16						1
RPsoaL5	RPEL17	RL57						3
RPsoaL4	RPEL17	RL49						3
RPsoaL3	RPEL17	RL39						3
RPsoaL2	RPEL17	RL29						3
RPsoaL1	RPEL17	RL18						3
LRectAb	LPEL3	LRIB3						5
LExtOb1	LPEL4	LRIB4						3.4
LExtOb2	LPEL5	LRIB5						3.4
LIntOb1	LPEL6	LRIB6						2.8
LIntOb2	LPEL7	LRIB7						2.8
LParsL5	LPEL9	LL53						2
LParsL4	LPEL9	LL43	LL410					2
LParsL3	LPEL9	LL33	LL410					2
LParsL2	LPEL9	LL23	LL411	LL310	LL210	LL19		2
LParsL1	LPEL9	LL13	LL412	LL311	LL211	LL110		2
LIlioLum	LPEL10	LRIB8	LL413	LL312	LL212	LL111	LRIB23	1
LLongTP	LPEL11	LRIB9	LL312	LL212	LL111	LRIB23		1.25
LLongTL5	LL54	LRIB10	LL212	LL111	LRIB23			1.25
LLongTL4	LL44	LRIB11	LL111	LRIB23				1.25
LLongTL3	LL34	LRIB12						1.25
LLongTL2	LL24	LRIB13						1.25
LLongTL1	LL14	LRIB14						1.25
LQuadLP	LPEL12	LRIB15						2.8
LQuadL1	LPEL12	LL15						2.8
LQuadL2	LPEL12	LL25						2.8
LQuadL3	LPEL12	LL35						2.8
LQuadL4	LPEL12	LL45						2.8
LLatDP	LPEL16	LRIB17						2.5
LLatDL5	LPEL15	LRIB17	LRIB22					2.5
LLatDL4	LL48	LRIB17	LRIB21					2.5
LLatDL3	LL38	LRIB17	LRIB20					2.5
LLatDL2	LL28	LRIB17	LRIB19					2.5
LLatDL1	LL17	LRIB17	LRIB18					2.5

LMultPL5	LPEL14	LL55	1
LMultPL4	LPEL14	LL46	1
LMultPL3	LPEL13	LL36	1
LMultPL2	LPEL13	LL26	1
LMultPL1	LPEL13	LL16	1
LMultL5L3	LL56	LL36	1
LMultL5L2	LL56	LL26	1
LMultL5L1	LL56	LL16	1
LMultL4L2	LL47	LL26	1
LMultL4L1	LL47	LL16	1
LMultL3L1	LL37	LL16	1
LMultL2T12	LL27	LRIB16	1
LPsoaL5	LPEL17	LL57	3
LPsoaL4	LPEL17	LL49	3
LPsoaL3	LPEL17	LL39	3
LPsoaL2	LPEL17	LL29	3
LPsoaL1	LPEL17	LL18	3

Table Ap.A.3 – Muscle anatomy sheet 2: coordinate description translation

Attachment and Nodal points

Point	**Body**	**X**	**Y**	**Z**
LL11	1	0.00	0.00	0.00
LL12	1	0.00	0.00	3.62
LL13	1	-2.60	-3.67	2.33
LL14	1	-0.20	-7.14	1.93
LL15	1	-3.60	-3.95	2.67
LL16	1	-0.50	-6.75	0.87
LL17	1	-0.20	-7.32	2.01
LL18	1	-1.90	-1.87	0.96
LL19	1	-8.00	-8.23	2.42
LL110	1	-4.30	-8.23	2.42
LL111	1	-1.50	-8.23	2.42
LL21	2	0.00	0.00	0.00
LL22	2	0.00	0.00	3.86
LL23	2	-2.70	-3.37	2.36
LL24	2	-0.20	-7.10	1.38
LL25	2	-3.80	-3.98	2.75
LL26	2	-0.50	-6.78	0.35
LL27	2	-1.50	-3.96	0.03
LL28	2	-0.20	-7.29	1.44
LL29	2	-2.00	-1.87	1.08
LL210	2	-7.80	-8.22	1.82
LL211	2	-4.10	-8.22	1.82
LL212	2	-1.50	-8.22	1.82
LL31	3	0.00	0.00	0.00
LL32	3	0.00	0.00	3.79
LL33	3	-3.00	-3.24	2.54
LL34	3	-0.20	-7.15	1.95
LL35	3	-3.80	-3.88	2.88
LL36	3	-0.50	-6.62	0.61
LL37	3	-1.50	-4.31	0.42
LL38	3	-0.20	-7.34	2.00
LL39	3	-2.10	-1.65	0.87
LL310	3	-7.60	-8.32	2.21
LL311	3	-3.90	-8.32	2.21
LL312	3	-1.50	-8.32	2.21
LL41	4	0.00	0.00	0.00
LL42	4	0.00	0.00	3.70
LL43	4	-4.00	-3.20	2.30
LL44	4	-0.20	-6.80	1.30
LL45	4	-4.40	-3.40	2.70
LL46	4	-0.50	-6.50	0.40
LL47	4	-1.50	-3.90	0.10
LL48	4	-0.20	-7.00	1.30
LL49	4	-2.30	-1.40	0.70
LL410	4	-5.00	-8.10	1.20
LL411	4	-7.40	-8.10	1.20
LL412	4	-3.80	-8.10	1.20
LL413	4	-1.50	-8.10	1.20
LL414	4	0.00	1.80	1.10
LL415	4	0.00	-1.90	1.20
LL416	4	0.00	-2.60	1.50
LL417	4	-3.60	-3.20	2.30
LL418	4	-2.00	-3.40	1.00

LL419		4	-1.60	-3.40	0.90	LRIB8	0	-8.40	-5.63	3.45
LL420		4	0.00	-7.00	0.90	LRIB9	0	-5.00	-5.28	8.45
LL421		4	0.00	-6.00	0.70	LRIB10	0	-2.00	-5.38	18.95
LL422		4	0.00	-7.00	1.20	LRIB11	0	-2.00	-5.41	22.45
LL423		4	0.00	1.60	0.30	LRIB12	0	-2.00	-4.95	26.96
LL424		4	0.00	0.10	0.60	LRIB13	0	-2.00	-4.99	31.16
LL425		4	0.00	-1.60	0.60	LRIB14	0	-2.00	-5.00	32.96
LL51		5	0.00	0.00	0.00	LRIB15	0	-7.20	-3.70	-0.03
LL52		5	0.00	0.00	3.89	LRIB16	0	-0.50	-5.39	-1.35
LL53		5	-5.00	-2.64	2.30	LRIB17	0	-12.00	1.69	11.52
LL54		5	-0.20	-6.06	1.19	LRIB18	0	-6.50	-5.99	-1.06
LL55		5	-0.50	-5.91	0.71	LRIB19	0	-6.50	-4.76	-3.84
LL56		5	-1.50	-3.95	0.51	LRIB20	0	-6.50	-4.14	-6.54
LL57		5	-2.30	-1.19	1.08	LRIB21	0	-6.50	-3.61	0.87
LL58		5	0.00	1.92	2.51	LRIB22	0	-6.50	-3.59	-1.33
LL59		5	0.00	-1.62	2.42	LRIB23	0	-1.50	-6.99	-1.56
LL510		5	0.00	-2.51	2.87	LRIB24	0	0.00	10.98	2.20
LL511		5	-3.60	-2.64	2.30	LRIB25	0	0.00	4.28	1.74
LL512		5	-2.00	-2.57	3.37	LRIB26	0	0.00	-2.59	-0.72
LL513		5	-2.40	-3.51	3.70	RL11	1	0.00	0.00	0.00
LL514		5	0.00	-5.61	1.75	RL12	1	0.00	0.00	3.62
LL515		5	0.00	-4.34	2.38	RL13	1	2.60	-3.67	2.33
LL516		5	0.00	-3.84	2.75	RL14	1	0.20	-7.14	1.93
LPEL1	p		0.00	0.00	0.00	RL15	1	3.60	-3.95	2.67
LPEL2	p		0.00	0.00	10.65	RL16	1	0.50	-6.75	0.87
LPEL3	p		-3.00	7.80	-2.54	RL17	1	0.20	-7.32	2.01
LPEL4	p		-13.00	3.50	11.52	RL18	1	1.90	-1.87	0.96
LPEL5	p		0.00	8.39	-2.60	RL19	1	8.00	-8.23	2.42
LPEL6	p		-12.50	-0.01	14.77	RL110	1	4.30	-8.23	2.42
LPEL7	p		-12.00	6.44	8.63	RL111	1	1.50	-8.23	2.42
LPEL8	p		-13.00	3.50	11.52	RL21	2	0.00	0.00	0.00
LPEL9	p		-6.00	-6.93	11.70	RL22	2	0.00	0.00	3.86
LPEL10	p		-6.80	-8.04	10.60	RL23	2	2.70	-3.37	2.36
LPEL11	p		-3.30	-8.05	10.50	RL24	2	0.20	-7.10	1.38
LPEL12	p		-9.00	-3.01	14.95	RL25	2	3.80	-3.98	2.75
LPEL13	p		-3.60	-6.71	11.88	RL26	2	0.50	-6.78	0.35
LPEL14	p		-1.50	-7.71	7.76	RL27	2	1.50	-3.96	0.03
LPEL15	p		-3.00	-5.61	12.98	RL28	2	0.20	-7.29	1.44
LPEL16	p		-6.00	-4.19	15.16	RL29	2	2.00	-1.87	1.08
LPEL17	p		-8.20	4.41	-2.22	RL210	2	7.80	-8.22	1.82
LPEL18	p		0.00	-8.14	10.61	RL211	2	4.10	-8.22	1.82
LPEL19	p		-6.20	-3.64	14.61	RL212	2	1.50	-8.22	1.82
LRIB1		0	0.00	0.00	0.00	RL31	3	0.00	0.00	0.00
LRIB2		0	0.00	0.00	32.50	RL32	3	0.00	0.00	3.79
LRIB3		0	-7.00	11.80	-0.39	RL33	3	3.00	-3.24	2.54
LRIB4		0	-12.50	-1.15	-5.51	RL34	3	0.20	-7.15	1.95
LRIB5		0	-10.50	5.34	-3.95	RL35	3	3.80	-3.88	2.88
LRIB6		0	-7.00	7.86	-6.43	RL36	3	0.50	-6.62	0.61
LRIB7		0	0.00	11.78	2.61	RL37	3	1.50	-4.31	0.42

RL38		3	0.20	-7.34	2.00
RL39		3	2.10	-1.65	0.87
RL310		3	7.60	-8.32	2.21
RL311		3	3.90	-8.32	2.21
RL312		3	1.50	-8.32	2.21
RL41		4	0.00	0.00	0.00
RL42		4	0.00	0.00	3.70
RL43		4	4.00	-3.20	2.30
RL44		4	0.20	-6.80	1.30
RL45		4	4.40	-3.40	2.70
RL46		4	0.50	-6.50	0.40
RL47		4	1.50	-3.90	0.10
RL48		4	0.20	-7.00	1.30
RL49		4	2.30	-1.40	0.70
RL410		4	5.00	-8.10	1.20
RL411		4	7.40	-8.10	1.20
RL412		4	3.80	-8.10	1.20
RL413		4	1.50	-8.10	1.20
RL414		4	0.00	1.80	1.10
RL415		4	0.00	-1.90	1.20
RL416		4	0.00	-2.60	1.50
RL417		4	3.60	-3.20	2.30
RL418		4	2.00	-3.40	1.00
RL419		4	1.60	-3.40	0.90
RL420		4	0.00	-7.00	0.90
RL421		4	0.00	-6.00	0.70
RL422		4	0.00	-7.00	1.20
RL423		4	0.00	1.60	0.30
RL424		4	0.00	0.10	0.60
RL425		4	0.00	-1.60	0.60
RL51		5	0.00	0.00	0.00
RL52		5	0.00	0.00	3.89
RL53		5	5.00	-2.64	2.30
RL54		5	0.20	-6.06	1.19
RL55		5	0.50	-5.91	0.71
RL56		5	1.50	-3.95	0.51
RL57		5	2.30	-1.19	1.08
RL58		5	0.00	1.92	2.51
RL59		5	0.00	-1.62	2.42
RL510		5	0.00	-2.51	2.87
RL511		5	3.60	-2.64	2.30
RL512		5	2.00	-2.57	3.37
RL513		5	2.40	-3.51	3.70
RL514		5	0.00	-5.61	1.75
RL515		5	0.00	-4.34	2.38
RL516		5	0.00	-3.84	2.75
RPEL1	p		0.00	0.00	0.00
RPEL2	p		0.00	0.00	10.65
RPEL3	p		3.00	7.80	-2.54
RPEL4	p		13.00	3.50	11.52
RPEL5	p		0.00	8.39	-2.60
RPEL6	p		12.50	-0.01	14.77
RPEL7	p		12.00	6.44	8.63
RPEL8	p		13.00	3.50	11.52
RPEL9	p		6.00	-6.93	11.70
RPEL10	p		6.80	-8.04	10.60
RPEL11	p		3.30	-8.05	10.50
RPEL12	p		9.00	-3.01	14.95
RPEL13	p		3.60	-6.71	11.88
RPEL14	p		1.50	-7.71	7.76
RPEL15	p		3.00	-5.61	12.98
RPEL16	p		6.00	-4.19	15.16
RPEL17	p		8.20	4.41	-2.22
RPEL18	p		0.00	-8.14	10.61
RPEL19	p		6.20	-3.64	14.61
RRIB1		0	0.00	0.00	0.00
RRIB2		0	0.00	0.00	32.50
RRIB3		0	7.00	11.80	-0.39
RRIB4		0	12.50	-1.15	-5.51
RRIB5		0	10.50	5.34	-3.95
RRIB6		0	7.00	7.86	-6.43
RRIB7		0	0.00	11.78	2.61
RRIB8		0	8.40	-5.63	3.45
RRIB9		0	5.00	-5.28	8.45
RRIB10		0	2.00	-5.38	18.95
RRIB11		0	2.00	-5.41	22.45
RRIB12		0	2.00	-4.95	26.96
RRIB13		0	2.00	-4.99	31.16
RRIB14		0	2.00	-5.00	32.96
RRIB15		0	7.20	-3.70	-0.03
RRIB16		0	0.50	-5.39	-1.35
RRIB17		0	12.00	1.69	11.52
RRIB18		0	6.50	-5.99	-1.06
RRIB19		0	6.50	-4.76	-3.84
RRIB20		0	6.50	-4.14	-6.54
RRIB21		0	6.50	-3.61	0.87
RRIB22		0	6.50	-3.59	-1.33
RRIB23		0	1.50	-6.99	-1.56
RRIB24		0	0.00	10.98	2.20
RRIB25		0	0.00	4.28	1.74
RRIB26		0	0.00	-2.59	-0.72

Vertebral body heights
From Lui (1971)

```
Ll=.3212*.617;        %Length of Lumbar region
Lt=.6788*.617;        %Length of Upper Trunk region
Thorax=Lt;
L1=0.1913*Ll;
L2=0.1518*Ll;
L3=0.2260*Ll;
L4=0.1987*Ll;
L5=0.2322*Ll;
```

Lordosis Angles
Quiet standing lordosis angles, averaged from Stokes and Gardner-Morse (1999) and those derived from Cholewicki et al (1996)

```
Pelvis=-20*pi/180;
L5=-14*pi/180;
L4=8*pi/180;
L3=18*pi/180;
L2=15*pi/180;
L1=10*pi/180;
Thorax=0*pi/180;
```

Intervertebral angles
From Arjmand and Shirazi-Adl (2005)

```
Pc=0*pi/180;    %Pelvis Change
Tc=0*pi/180;    %Thorax Change
Dc=Tc-Pc;
Pelvis=Pelvis+Pc;          %Pelvis
L5=L5+Pc+0.14*Dc;
L4=L4+Pc+0.40*Dc;
L3=L2+Pc+0.63*Dc;
L2=L3+Pc+0.79*Dc;
L1=L1+Pc+0.92*Dc;
Thorax=Thorax+Tc;          %Thorax
```

v. Delay Margin Function

The source code for finding the delay margin is shown below. This function requires only the two input Jacobians Ji and Jd (section 3.B.iv) and employs a rescaling resolution over the frequency domains (section 3.C.iii.c).

Using this function:

```
Td=tdinvest(Ji,Jr,DrawSet)
```

Where drawset=1 to draw result figures, or drawset=0 to draw no figures.

```
function [Td]=tdinvest(Ji,Jr,DrawSet);
%-----Time Delay Investigation
   fmax=tdfmax(Ji,Jr);
   len=size(Ji,1);
   PrevEig=zeros(len,1);
   f=linspace(0,fmax,1000);
   rho=zeros(size(f));
   lamshist=zeros(len,length(f));
   lammags=zeros(len,length(f));

   for i=1:length(f)
      lams=eig(sqrt(-1)*f(i)*eye(len)-Ji,Jr);
      %-----Sort Eigenvalues
         b=zeros(len,1);
         p=zeros(len,len);
         for j=1:len
            dist=PrevEig-lams(j);
            for k=1:len
               if(isnan(dist(k)))
                  dist(k)=Inf;
               end
            end
            p(j,:)=sqrt(real(dist).^2+imag(dist).^2)';
         end
         for j=1:len
            [q,indc]=min(min(p));
            [q,indr]=min(p(:,indc));
            b(indc)=lams(indr);
            p(:,indc)=NaN;
            p(indr,:)=NaN;
         end
         PrevEig=b;  %Sorted Eigenvalues
      %-----
      lamshist(:,i)=b;
      lammags(:,i)=sqrt(real(b).^2+imag(b).^2);
      lammags=min(lammags,1e9*ones(size(lammags)));
      rho(i)=min(lammags(:,i));
   end
   if(DrawSet==1)
      figure(10)
      plot(f,lammags(:,:),'LineWidth',5);axis([f(1) f(end) 0 1])
      xlabel('Frequency (rad)','FontSize',16,'FontWeight','bold')
      ylabel('|Eig|','FontSize',16,'FontWeight','bold')
      hold on;
      td=0;
      k=1;
      l=0;li=-.1;
      for i=1:length(Ji)
         traji=lammags(i,:);
         for j=1:length(f)-1
            if((traji(j)<1)&(traji(j+1)>1))
               perc=(1-traji(j))/(traji(j+1)-traji(j));
               fc=perc*(f(j+1)-f(j))+f(j);
               td(k)=tdfind(fc,Ji,Jr);
               plot(fc,1,'o')
               l=l+li;
               text(fc,0.5+l,[num2str(round(min(td(k))*100)/100) ' s'],'FontSize',16,'FontWeight','bold')
               plot([fc fc],[1 0.5+l],':','LineWidth',5)
               k=k+1;
            elseif((traji(j)>1)&(traji(j+1)<1))
               perc=(1-traji(j))/(traji(j+1)-traji(j));
               fc=perc*(f(j+1)-f(j))+f(j);
               td(k)=tdfind(fc,Ji,Jr);
               plot(fc,1,'o')
               l=l+li;
               text(fc,0.5+l,[num2str(round(min(td(k))*100)/100) ' s'],'FontSize',16,'FontWeight','bold')
```

```
            plot([fc fc],[1 0.5+l],':','LineWidth',5)
            k=k+1;
        end
        if(l<=-.4)
            li=.1;
        elseif(l>=.4)
            li=-.1;
        end
    end
end

    hold off
else
    td=0;
    k=1;
    l=0;li=-.1;
    for i=1:length(Ji)
        traji=lammags(i,:);
        for j=1:length(f)-1
            if((traji(j)<1)&(traji(j+1)>1))
                perc=(1-traji(j))/(traji(j+1)-traji(j));
                fc=perc*(f(j+1)-f(j))+f(j);
                td(k)=tdfind(fc,Ji,Jr);
                l=l+li;
                k=k+1;
            elseif((traji(j)>1)&(traji(j+1)<1))
                perc=(1-traji(j))/(traji(j+1)-traji(j));
                fc=perc*(f(j+1)-f(j))+f(j);
                td(k)=tdfind(fc,Ji,Jr);
                l=l+li;
                k=k+1;
            end
            if(l<=-.4)
                li=.1;
            elseif(l>=.4)
                li=-.1;
            end
        end
    end
end
Td=min(td);
```

```
function [fmax]=tdfmax(Ji,Jr);
    fmax=0;
    len=size(Ji,1);
    fi=1000;
    fmult=1;
    cont=1;
    k=1;
while(cont==1)
    f=linspace(0,fi*fmult,1000);
    rho=zeros(size(f));
    for i=1:length(f)
        lams=eig(sqrt(-1)*f(i)*eye(len)-Ji,Jr);
        lammags=sqrt(real(lams).^2+imag(lams).^2);
        rho(i)=min(lammags);
    end
    for i=1:(length(f)-1)
        if((rho(i)<1)&(rho(i+1)>1))
            fmax=1.1*f(i);
        end
    end
    if((k>10)|(fmax>(f(end)/3)))
        cont=0;
    end
    if(rho(end)>1)
        fmult=fmult/100;
    else
        fmult=fmult+1;
```

```
    end
    k=k+1;
end
```

```
function [td]=tdfind(f,Ji,Jr);
    lams=eig(sqrt(-1)*f*eye(size(Ji,1))-Ji,Jr);
    lammags=sqrt(real(lams).^2+imag(lams).^2);
    %Find nearest to |1|
    dist=lammags-1;
    [C,I]=min(abs(dist));
    lamda=lams(I);
    theta=atan2(-imag(lamda),real(lamda));
    if(theta<0)
        theta=theta+2*pi;
    elseif(theta>2*pi)
        theta=theta-2*pi;
    end
    td=theta/f;
```

APPENDIX B
HOW TO USE THIS MODEL

This chapter will provide directions to use the model developed in this work.

i. Sample Procedure

An infinite amount of experiments may be performed on the model. Instead of addressing any number of these, one experiment will be overviewed. Procedures to perform other experiments are left to the reader (however see section 6.B for suggested experiments).

Experiment: Optimize muscle activations and find the delay margin for a fixed posture spinesim.m is the main program file. Most of the model settings are located in this document. In here it is possible to choose dynamic parameters (mass, lengths, etc), muscle anatomy, external forces, and more.

1. Modify spinesim.m
    a. Change masses, lengths, gravity, intervertebral stiffness, etc.
    b. Choose a muscle anatomy and external forcing system
    c. Define equilibrium spine posture
    d. Choose External Load
    e. Set: OptSet=1 (optimize), SimFlag=0 (simulate)
2. Modify muscle parameters
    a. Set stiffness (q) and damping (b) in muscmodel.m
    b. Set Reflex gains Gp and Gd in muscmodelR.m
3. Run spinesim.m

The program will follow the following procedure:

1. Optimize muscle activations to stabilize the spine for (τ=0)
2. Find delay margin

The resultant delay margin will be written to the screen at the end of the program.

ii. Main File - spinesim.m

```
%Spine Simulation Program
%Muscle and External Force Model
%Tim Franklin
%Virginia Tech


   disp(['Simulation Started:   ' datestr(clock,21)])

%-----Define Simulation Variables

   %Distances in meters (m), masses in kilograms (kg)
   Ll=.3212*.617;         %Length of Lumbar region
   Lt=.6788*.617;         %Length of Upper Trunk region

   V=struct;              %All Model Parameters
   V.L0=0.419;             %L0 is the thorax, called L0 for simplicity later
   V.L1=0.1913*Ll;
   V.L2=0.1518*Ll;
   V.L3=0.2260*Ll;
   V.L4=0.1987*Ll;
   V.L5=0.2322*Ll;
   V.la=0.12;              %Radius of trunk, ML
   V.lb=0.06;              %Radius of trunk, AP
   V.cent1=0;              %Position of the L5 w.r.t. pelvis center of rotation
   V.cent2=0;
   V.cent3=0.1067;        %Treat Pelvis as pendulum...Location of L5 pivot in P
   V.g=9.8;               %Gravity positive
   V.m0=20;                %Mass in kg
   V.m1=2;
   V.m2=2;
   V.m3=2;
   V.m4=2;
   V.m5=2;
   Vars=[V.L0,V.L1,V.L2,V.L3,V.L4,V.L5,V.la,V.lb,V.cent3,...
      V.g,V.m0,V.m1,V.m2,V.m3,V.m4,V.m5];

%-----Intervertebral Disc Parameters
   IKp=50;    %Planar
   IDp=0.5;
   IKt=50;    %Twist
   IDt=0.5;
   IVD=zeros(1,4);
   IVD=[IKp,IDp,IKt,IDt];

%-----Initialize Position Vectors

   Stateo=zeros(36,1);     %States vector initial
   Pango=zeros(9,1);       %Pelvis vector initial

%-----Read Muscle and Ext Force Models
   %Muscles may be added as .xls or .csv, but musc be added differently in
   %either case.  muscleCorg and forceCorg convert .xls to the matrix
   %format.

   M=struct;E=struct;

   M=addmuscfile(M,'.\ChMusc\CholeMusc.xls');    %Cholwicki muscle file

   E=addforcefile(E,'LoadLiftHigh.xls');
%     E=addforcefile(E,'LoadLiftLow.xls');

%-----Organize into matrices for C code

   Mc=muscleCorg(M);
   Ec=forceCorg(E);

%     Mc=addmusccsv(Mc,'..\Muscle\Stokes\CSV\InterRot.csv');
```

```
%    Mc=addmusccsv(Mc,'..\Muscle\Stokes\CSV\InterTrans.csv');
%    Mc=addmusccsv(Mc,'..\Muscle\Stokes\CSV\InterSpin.csv');

%-----

%-----Define State Variables
   %State:[T5,B5,A5,T4,B4,A4,T3,B3,A3,T2,B2,A2,T1,B1,A1,T0,B0,A0]'
   %Pang:[Tp,Bp,Ap,Tp',Bp',Ap',Tp",Bp",Ap"]'

   V.SperM=1;                    %Number of state variables per muscle
   V.States=38+V.SperM*length(M);     %Number of state variables
   Stateo(37:V.States,1)=0;

   %-----Physiological Lordosis Angles

      [AAPOA]=getaapoa();              %Anat. AP-Offset Angles (Lordosis)
      Pango(1)=AAPOA(7);
      Stateo(1)=AAPOA(6);
      Stateo(4)=AAPOA(5);
      Stateo(7)=AAPOA(4);
      Stateo(10)=AAPOA(3);
      Stateo(13)=AAPOA(2);
      Stateo(16)=AAPOA(1);

%-----Establish Rest Length for Muscles

   [Lo,CSA]=muscpropmat(Stateo,Pango,Vars,Mc,Ec);

%-----Draw and display muscles
   DrawSet=0;     %1=draw, 0=don't draw

   if(DrawSet==1)
      drawspinemat(Stateo,Pango,Mc,Ec,V,zeros(size(Mc,1),1));
      drawnow
      pause(1);
   end

%-----Set Initial Values for External Forces-Used in Equilib and in Simulation
   %Negative=Pulling (tension), Positive=Pushing

   EFo=zeros(length(E),1);
   EFo(1)=200;

%-----Solve for Equilibrium and Stable Muscle Activation Levels, etc
%    OptimSet=0;    %0=Don't Optimize, 1=Optimize

   Uo=zeros(size(Mc,1),1);Un=Uo;
   Po=zeros(size(Mc,1),1);Pn=Po;
   Sn=zeros(18,1);
   flag=0;
   if(OptimSet==1)
      starttime=clock;
      optimizationscript
      disp(['Optimization Elapsed Time: ' etimev(clock,starttime)]);
      Stateo(1:18)=Stateo(1:18)+Sn;
   else
      load OptimSOL.mat;
      if(length(Un)==length(Uo))
         flag=1;
      else
         disp('Must Run Optimization First!!');
         flag=0;
      end
   end
   if(DrawSet==1&flag==1)
      drawspinemat(Stateo,Pango,Mc,Ec,V,(Un/max(Un)).^(1/3));
   end

%-----Build Instant and Reflex Jacobians, for Time Delay analysis
%            tdsimulate=0;
```

```
if(OptimSet==1)
   Td=inf;
   if(flag==1)
      Ji=Jpi;
      for i=1:36
         Ji(19:36,i)=Ji(19:36,i)+(Un'*Jai(:,19:36,i))';
      end
      Jr=Jpr;
      for i=1:36
         Jr(19:36,i)=Jr(19:36,i)+(Un'*Jar(:,19:36,i))';
      end
      disp(['Finding Time Delay...'])
      if(max(max(abs(Jr)))>1e-6)
         Td=tdinvest(Ji,Jr,DrawSet);
      end
   end
end


%-----Display Results

[Ma,Ea,ML,MV,Gl,Gr,Rm,Or,dRm,dOr,RBV,dRBV] = kinematicsc(Stateo,Pango,Vars,Mc,Ec);
[MF,dA]=muscmodel(Un,Pn,zeros(size(Un)),ML,MV,Lo,CSA);
disp(['T: ' num2str(round(min(Td)*100000)/100000) '   P: ' num2str(EnMain) '   F: ' num2str(sum(MF))]);

%-----Save Optimization Solution

save OptimSOL.mat Un Pn Sn Td EnMain MF

%-----For Optimize Only, without simulation (save time)
%    simflag=0;     %flag=1 simulate, flag=0 don't simulate

%-----Add Kinematic Disturbance for simulation

StateDist=Stateo;
StateDist(1)=StateDist(1)+0.01*pi/180;

%-----Define Simulation Timing

if(simflag==1)

   Tstep=0.01;
   Tfinal=5;
   Tode=0:Tstep:Tfinal;
   Lags=[lag];

   %-----Set Simulation Options

   options=odeset('OutputFcn',@odebarplot,'Events',@extrema,'RelTol',1e-6,'AbsTol',1e-9,'Stats','on');

   %-----Run Simulation

   disp('Simulating...');
   starttime=clock;
   sol=dde23(@spinedyn,Lags,StateDist,Tode,options,Pango,AAPOA,Mc,Ec,Lo,CSA,Un,Pn,EFo,Vars,IVD);
   Time=sol.x;
   State=sol.y';
   disp(['Simulation Elapsed Time: ' etimev(clock,starttime)]);
   disp(['Lag Simulated: ' num2str(Lags(1))]);
   Stated=State-ones(length(Time),1)*Stateo';
%-----
else
   disp('Simulation not run')
end
```

iii. Function Prototypes

Most of the important functions will be explained below with function prototypes.

spinesim.m

Script main function of program.

[M]=addmuscfile(M,filename)

modify struct M with muscle descriptions from excel file.

[E]=addforcefile(E,filename)

modify struct E with external force descriptions from excel file.

[Mc]=muscleCorg(M)

convert struct description of muscles into a matrix (for use in C functions)

[Ec]=forceCorg(E)

convert struct description of external forces into a matrix (for use in C functions)

[AAPOA]=getappoa()

output anatomical lordosis angles of kinematic posture.

[Ma,Ea,ML,MV,Gl,Gr,Rm,Or,dRm,dOr,RBV,dRBV] = kinematicsc(S,Pang,Vars,Mc,Ec)

take states S, pelvis angles Pang, Variable vector Vars, muscle Mc and forces Ec, output rotated versions of Mc, Ec, muscle lengths and stretch rates ML and MV, and helper variables for generalized forces calculation

[MF,dA]=muscmodel(U,P,A,ML,MV,Lo,CSA);

take activations U variable gains P, differential states A, muscle lengths ML, velocities MV, initial lengths Lo, and cross sectional areas CSA. Output muscle force MF and differential activation states dA.

[PRQ]=intervertc(RBV,dRBV,AAPOA,IVD);

input basis vectors of vertebrae, lordosis angles, output moments of intervertebral discs

[Mm,Cv,Gv]=dynmatsc(S,Pang,Vars);

input states S, pelvis angles Pang, and variables Vars, output dynamics Mm, Cv, and Gv

Q=genforcesc(Ma,MF,Ea,EF,PRQ,Gl,Gr,Or);

Input muscle rotated anatomy, muscle forces, rotated external force locations, external forces, intervertebral disc moments, and helper variables. Output generalized forces.