12-2011

COMPARATIVE STUDY ON POSTURE AND ITS INFLUENCES ON HORIZONTAL GROUND REACTION FORCES GENERATED BY MUSCLES: IMPLICATIONS FOR CROUCH GAIT

Hoa Xuan Hoang
hhoang@utk.edu

To the Graduate Council:

I am submitting herewith a thesis written by Hoa Xuan Hoang entitled "COMPARATIVE STUDY ON POSTURE AND ITS INFLUENCES ON HORIZONTAL GROUND REACTION FORCES GENERATED BY MUSCLES: IMPLICATIONS FOR CROUCH GAIT." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Biomedical Engineering.

Jeffrey A. Reinbolt, Major Professor

We have read this thesis and recommend its acceptance:

William R. Hamel, Richard D. Komistek

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

COMPARATIVE STUDY ON POSTURE AND ITS INFLUENCES ON HORIZONTAL GROUND REACTION FORCES GENERATED BY MUSCLES: IMPLICATIONS FOR CROUCH GAIT

A Thesis Presented for

the Master of Science

Degree

The University of Tennessee, Knoxville

By

Hoa Xuan Hoang

December 2011

This thesis is dedicated to my parents.

ACKNOWLEDGEMENTS

I would sincerely like to thank the many people who made this thesis possible, without them none of this would be possible.

I would be utterly lost if not for the support and leadership of Dr. Jeffrey A. Reinbolt throughout this endeavor. His enthusiasm, inspiration, and ability to explain things clearly and simply throughout made studying biomechanics fun and exciting. I am indebted and grateful to him for providing me the opportunity to conduct research at the Reinbolt Research Group.

I would also like to extend my gratitude to Dr. William Hamel and Dr. Richard Komistek for their dedication, knowledge, and instruction in the classroom. I would like to thank both for their time and contribution in fulfillment of their committee responsibilities.

I wish to thank all of my colleagues at the Reinbolt Research Group for providing an interesting and enjoyable environment in which to learn and grow. I can honestly say that there was never a dull moment.

I would like to thank my family for their complete support and for providing a loving environment for me.

Lastly, and most importantly, I wish to thank my parents, Long Kim Hoang and Hoa Thi Pham. They raised me, supported me, taught me, and loved me. To them I dedicate this thesis.

ABSTRACT

Crouch gait decreases walking efficiency due to the increased knee and hip flexion during the stance phase of gait. Crouch gait is generally considered to be disadvantageous for patients with cerebral palsy; however, a crouched posture may afford biomechanical advantages that lead some patients to adopt a crouch gait.

To investigate one possible advantage of crouch gait, a musculoskeletal model created in OpenSim was placed in 15 different postures from upright to severe crouch during initial, middle, and final stance of the gait cycle. A series of optimizations were performed for each posture to maximize ground reaction forces for the 8 compass directions in the transverse plane by modifying muscle forces acting on the model. We compared the areas of the force profiles across all postures.

An overall larger force profile area is allowed by postures from mild crouch (for initial stance) to crouch (for final stance). The overall ability to generate larger ground reaction force profiles represents a mechanical advantage of a crouched posture. This increase in muscle capacity while in a crouched posture may allow a patient to generate new movements to compensate for impairments associated with cerebral palsy, such as motor control deficits.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

1.1 Crouch Gait: Debilitating Movement Abnormality in Children with Cerebral Palsy

Crouch gait is a common condition among children with cerebral palsy, the leading cause of childhood disability affection motor control and development. Cerebral palsy is an umbrella term for non-progressive, non-contagious motor conditions that impair movements in humans. There are several types of cerebral palsy; however, there is currently no cure. The lifetime costs for persons born in 2000 with cerebral palsy in the United States are estimated to total $11.5 billion (average of $912,000 per person) in 2003 US dollars and place great demands on the healthcare system (Honeycutt *et al*., 2004). In a more recent study outside of the US, the average lifetime costs of cerebral palsy are estimated to be even higher at around $1.2 million per person in Europe (Kruse *et al*., 2009).

Crouch gait is a symptom of spastic cerebral palsy and it decreases walking efficiency due to the increased knee and hip flexion during the stance phase of gait (Wren *et al*., 2005). Excessive knee flexion is problematic as it impedes foot clearance, increases the energy requirements of walking (Campbell *et al*., 1978; Rose *et al*., 1990), and increases patellofemoral force (Perry *et al*., 1975). Patients being unable to clear the foot off the ground can suffer tripping or other serious injuries. The energy expenditure indices (EEI)

based on oxygen uptake and heart rate were inefficiently high for children walking with a crouch gait compared to children walking in a normal gait (Rose *et al*., 1990). This increased in energy requirements has been linked to a decreased in functional involvement (Johnston *et al*., 2004) while increased in patellofemoral forces can lead to deteriorated joints and chronic knee pain (Campbell *et al*., 1978; Rose *et al*., 1990; Jahnsen *et al*., 2004). If left untreated, these symptoms can worsen over time (Bell *et al*., 2002).

While there are surgeries to correct crouch gait and decrease excessive knee flexion, it is unpredictable, and often time, unsuccessful. Despite this, patients often time undergo several different surgeries and procedures [9, 10]. Common interventions to treat crouch gait are designed to modify dynamical functions of muscles to try to get patients with crouch gait to walk in as normally as possible. This usually involves intensive physical therapy and strength training to enable them to walk in a more upright (normal) gait pattern. Like surgeries, however, attempts to fix crouch gait in children with cerebral palsy by trying to get them to walk in a upright gait has yielded inconsistent results.

Figure 1-1. Patient with cerebral palsy displaying symptoms of crouch gait. Image courtesy of Gillette Children's Specialty Healthcare Hospital.

1.2 Need for Utilizing Simulation in Biomechanics

Movement abnormalities, such as crouch gait, are very difficult to analyze as many elements of the neuromusculoskeletal system work together to coordinate movement. The musculoskeletal system is a complex multi-joint linkage system. Muscles in the system are able to accelerate joints that they do not cross or body segments they do not attach due to dynamic coupling (Zajac and Gordon, 1989; Kepple *et al*., 1997; Riley and Kerrigan, 1999; Arnold *et al*., 2005; Kimmel and Schwartz, 2006). Furthermore, bi-articulate muscles, such as the hamstrings, cross two joints (hip and the knee in the case of the hamstring) rather than just one joint like uni-articular muscles.

While there is a large quantity of experimental data from clinics that treat movement abnormalities such as cerebral palsy, it remains challenging to understand the causes of movement abnormalities through experiments alone. Several variables that are important in movement dynamics, such as muscle forces and muscle activation, are not usually available in experiments. Even with electromyography (EMG) recordings that can indicate when groups of muscles are active, this activation does not indicate the motion of the body due to dynamic coupling. Dynamic simulation can integrate models with anatomical and physiological elements of the neuromusculosketal system together with experimental data to help understand the mechanisms of movement abnormalities as well as using it as a tool to predict treatment outcomes. Simulation can provide estimates for important variables involved in movement abnormalities. Additionally,

simulation enables cause-effect relationships to be identified and allow researchers to perform “what if” studies.

1.3 Need for Study

The disadvantages of crouch gait are well known; however, it remains challenging to elucidate mechanisms that lead to a crouched posture (Ross and Engsberg, 2007). Several factors have been linked with crouch gait, including muscle weakness, spasticity, tightness (Hoffinger *et al*., 1993; Mcnee *et al*., 2004; Arnold *et al*., 2005), decreased motor control (Gage and Schwartz, 2004), and skeletal deformities (Gage and Schwartz, 2001). Despite being studied for decades, a cause and effect relationship between these factors and crouch gait remains unknown, due to the complexity of the musculoskeletal system (Zajac and Gordon, 1989; Kepple *et al*., 1997; Riley and Kerrigan, 1999; Arnold *et al*., 2005; Kimmel and Schwartz, 2006).

Crouch gait is generally considered to be a negative symptom of cerebral palsy; however, it may afford biomechanical advantages that lead some patients to adopt a crouch gait. An athlete gets lower to increase the ability to produce movement in all directions. Similarly, a standing passenger on a moving train gets lower to increase the ability to resist movement. In each case, the movement was produced or resisted by generating ground reaction forces in the transverse plane. A crouched posture may increase the ground reaction forces, and thereby, allowing an individual to accelerate in

that direction or reject disturbances. This increase in the individual ground reaction forces in the transverse plane will have an overall larger ground reaction force profile area.

A link between crouched gait postures and the capacity of muscles to generate ground reaction forces has several clinical implications. If a crouched posture reduces the capacity of muscles to generate ground reaction forces, patients may have to spend more energy to maintain a crouched posture compared to an upright posture. However, if a crouched posture increases this capacity, patients may be better suited to produce or resist movements to avoid injuries from falling or tripping.

1.4 Focus of Thesis

The focus of this thesis is to use musculoskeletal modeling and optimization technique implemented in C++ to evaluate one possible advantage of crouch gait. The objective was to determine if posture influences muscles capacity to generate ground reaction forces in the transverse plane during initial, middle, and final stance of a gait cycle. This study is a comparative study examining crouched and upright posture and its influences on transverse ground reaction forces. We hypothesized that a crouched posture allows the largest force profile area among postures from upright to severe crouch during the stance phase of gait. This larger transverse ground reaction force profile may show an unrecognized benefit to crouched gait or verify why we crouch in general. Identifying

the relationship between posture and ground reaction forces may show an advantage to adopting a crouched posture to compensate for impairments associated with cerebral palsy.

2 BACKGROUND

2.1 Motion Capture

Motion capture, also known to as motion tracking or mocap, is the use of external devices to record the position and orientation of a real object (usually animal or human subjects) in physical space. The most common type of motion capture system is based on utilizing passive optical technology. Passive refers to markers, which are spheres coated in retroreflective material to reflect light that is transmitted near the camera lens, placed on the subject. Optical refers to the technology used to record 3D data. This involves several high-speed, high-resolution video cameras placed around the subject and experimental area. By placing passive markers on the subject, video cameras record the position of those markers in time and a set of motion data (marker data) can be generated. Motion capture is used in various fields ranging from military to sports and filmmaking. Special effects companies have used this technique to capture the motions of real actors and create realistic animations in movies such as Star Wars, The Lord of the Rings, Avatar, and The Matrix.

2.2 Biomechanical Model

Researchers utilize motion-capture technology to construct biomechanical models of human subjects. The position of internal landmarks such as joint centers may be estimated from the position of the external markers. The markers also enable the

creation of individual segment reference frames to define the position and orientation of each body segment within a Newtonian laboratory reference frame. Marker data collected from an individual during motion capture are used to prescribe the motion of the biomechanical model.

2.3 Kinematics

Human kinematics is an extension of classical dynamics, which is the study of motions of bodies or systems, applied to the human musculoskeletal structures. Human kinematics is the study of the positions, angles, velocities, and accelerations of body segments and joints during motion. With kinematic data and mass-distribution data recorded from experiments, one can study the forces and torques required to produce the recorded motion data.

2.4 OpenSim – Open-source Dynamic Simulation Software

Dynamic simulation software has been used for quite some time and its advantages and values are widely accepted in the field of biomechanics; however, the field is disorganized due to each laboratory developing its own software packages. Furthermore, these simulation software packages are not available to the biomechanics community to be used and evaluated. There are commercial software packages available such as Anybody (MSC Software Corp), Visual 3-D (C-Motion Inc.), Anybody (Anybody Technology), and SIMM (Musculographic Inc.). However, these packages are

costly and access to the source code is not available for researchers to extend the capabilities of these software packages. OpenSim is a freely available, open-source software platform to build musculoskeletal models and create simulations of movement, including inverse dynamic and forward dynamic simulations. The software was developed at Stanford University's Neuromuscular Biomechanics Lab (NMBL). The OpenSim's open-source simulation environment allows researchers to further advance the development of simulation technology as well as allow it to integrate dynamic simulations in the field of biomechanics. There is a large OpenSim community which allows the community to build, exchange, test, analyze, and improve musculoskeletal models and simulations through collaboration.

2.5 Optimization

Optimization involves finding the global minimum or maximum value of an objective function, also known as cost function or energy function, by adjusting a set of design variables which are input values from an allowed set. Simply, optimization is the method of finding the "best value" in a given domain for a given objective function. There are a variety of objective function types and domain types depending on the problem. In biomechanics for example, the objective function may be the ground reaction force the neuromusculoskeletal model generates during a running routine. The ground reaction force is a function of the neuromusculoskeletal model's muscle parameters as well as the neuromusculoskeletal model's kinematic and kinetic parameters. The direction of

the ground reaction force can also be defined if the researcher is only interested in certain directions of the ground reaction force. Limiting the domain by increasing the number of constraints may speed up calculation time as well as decrease chances of finding local minima or maxima. Optimization techniques may be used to modify the design variables of the neuromusculoskeletal model to maximize ground reaction force the model can generate during a running routine.

3 METHODS

3.1 Three-Dimensional Musculoskeletal Model

A three-dimensional musculoskeletal model was first constructed in OpenSim (Delp *et al*., 2007). OpenSim is a robust and powerful open-source software system that allows biomechanists to develop neuromusculoskeletal models and create dynamic simulations of movement. It uses the freely available Simulation Toolkit (SimTK) that provides the essential mathematical and physics-based simulation libraries and components. For example, Simbody™ is the open-source multibody dynamic engine that is packaged with SimTK. Simbody™ can provide results for any set of n-coordinates using its advanced formulation of rigid body mechanics. The underlying source code for OpenSim is available in ANSI C++ (Figure 3-1) with the graphical user interface (GUI) written in Java (Figure 3-2).

The three-dimensional musculoskeletal model consists of 10 rigid body segments: head, trunk, pelvis, and a right and left femur, tibia, and foot segments (Figure 3-3). The lower extremity joints were modeled as follows: the subtalar and ankle joint were pin joints (Inman, 1976), each knee was a pin joint with tibiofemoral and patellofemoral kinematics defined by knee flexion angle (Delp *et al*., 1990), and the hip was a ball-and-socket joint (Anderson and Pandy, 1999). The head and torso were included in the model and were articulated with the pelvis through a ball-and-socket joint (Anderson

and Pandy, 1999). The stance foot (right foot in our study) was a weld joint to the ground while the left foot was free to move. The weld joint was used on the stance foot to allow for the calculation of the ground reaction forces. The arms were not included in the musculoskeletal model, but the mass of the arms was included in the head and torso body segment.

To determine if posture influences muscles capacity, the three-dimensional biomechanical model was constructed with 92 muscle, or "muscles-tendon actuators," in OpenSim (Figure 3-4). Muscles-tendon actuator's paths are defined with the origin and insertion point with intermediate via points if there is muscle wrapping. The force-generating properties of each muscle-tendon actuators are obtain by scaling a generic Hill-type muscle model (Hill, 1938; Zajac, 1989). The Hill-type muscle model is a tendon in series with a muscle. The tendon is represented as a non-linear elastic element while the muscle is represented by a passive elastic element in parallel with an active contractile element (CE). Each muscle-tendon actuator (Figure 3-5) must be scaled using four properties (peak isometric muscle force - F_0^M, optimal muscle-fiber length - L_0^M, pennation angle - α, and tendon slack length - L_S^T) and three curves (normalized passive and active muscle fiber force-length relationship, and normalized tendon force-length relationship) to represent a muscle. Muscle and tendon parameters were from Delp *et al*. (1990, 2007). The values used have been reported in literature from experiments. Their procedure to determine muscle-tendon parameters were similar to Hoy *et al*.

(1990). Physiological cross-ectional area to determine peak isometric muscle force were taken from Friederich *et al*. (1990) and Wikiewics *et al*. (1983). The fiber length and pennation angle were from Friederich and Brand (1990). Peak isometric muscle forces for some of the muscles, such as gluteus maximus, were also taken from Brand *et al*. (1986). The muscle-tendon actuator model produces force for a given muscle length and muscle activation.

Similar neuromusculoskeletal model have been used extensively in research to study cerebral palsy. Hicks *et al*. (2007) modified a similar neuromusculoskeletal model to look at the effects of tibial torsion in patients with cerebral palsy. Arnold *et al*. (2006) examined muscle-tendon lengths and velocities of the hamstrings in the evaluation and treatment of crouch gait with a simpler neuromusculoskeletal model that had only one muscle (hamstring). Steele *et al*. (2010) looked at muscle contributions during single limb support to support and progression in patients with cerebral palsy.

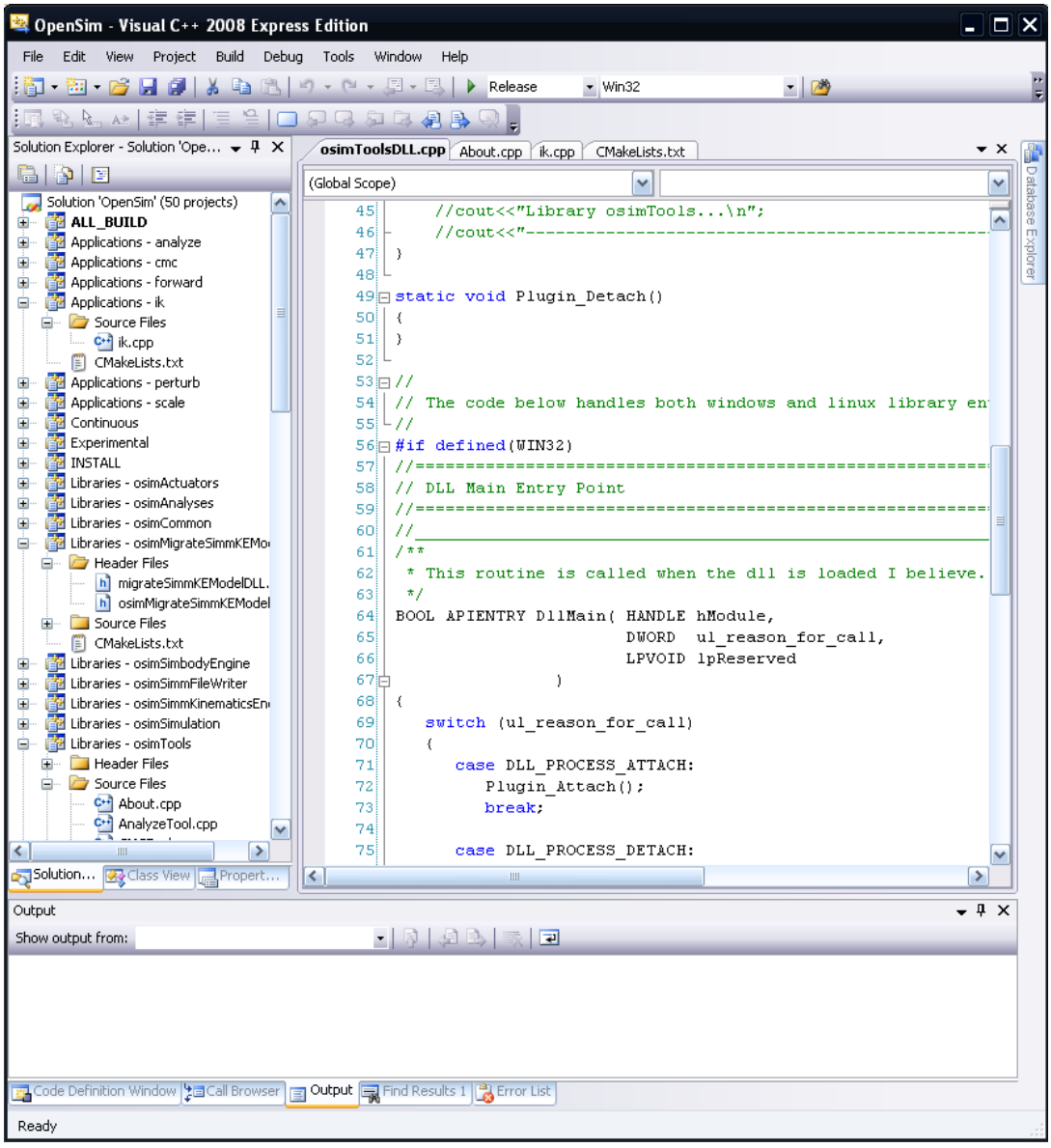

Figure 3-1. OpenSim Source Code in Microsoft Visual C++

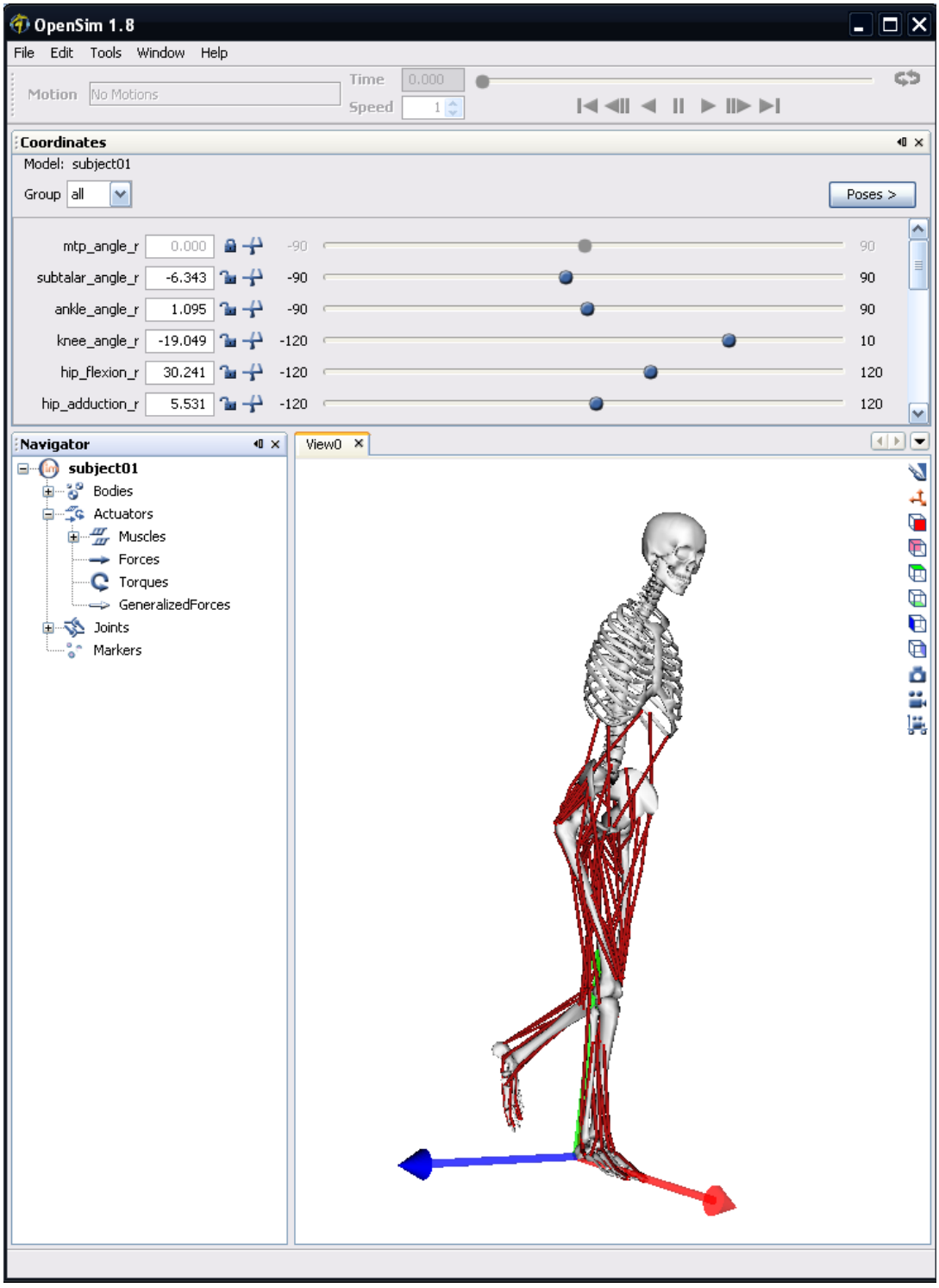

Figure 3-2. OpenSim Graphical User Interface (GUI)

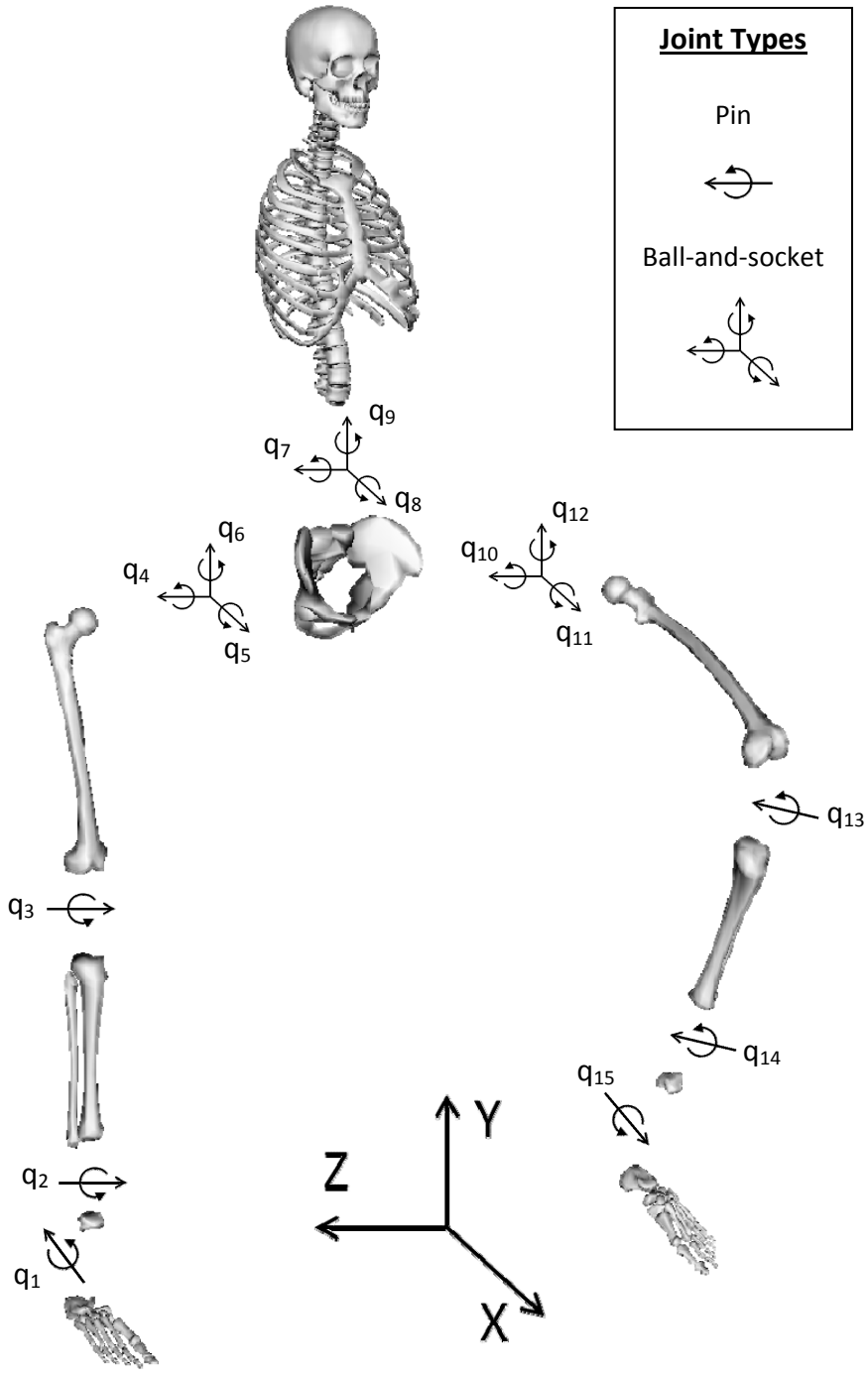

Figure 3-3. The 3-dimensional, 10 segment, 15 DOF kinematic model linkage joined by a set of pin and ball-and-socket joints.

Table 3-1. Degree of freedom for biomechanical model

DOF	Description
q_1	Right ankle inversion-eversion angle
q_2	Right ankle plantarflexion-dorsiflexion angle
q_3	Right knee flexion-extension angle
q_4	Right hip flexion-extension angle
q_5	Right hip adduction-abduction angle
q_6	Right hip internal-external rotation angle
q_7	Trunk anterior-posterior tilt angle
q_8	Trunk elevation-depression angle
q_9	Trunk internal-external rotation angle
q_{10}	Left hip flexion-extension angle
q_{11}	Left hip adduction-abduction angle
q_{12}	Left hip internal-external rotation angle
q_{13}	Left knee flexion-extension angle
q_{14}	Left ankle plantarflexion-dorsiflexion angle
q_{15}	Left ankle inversion-eversion angle

Table 3-2. Mass and Mass Center of Each Body in the Model

		mass center (m)		
Body	mass (kg)	x	y	z
Calcaneus Right	1.20735	0.10271	0.03081	0.00000
Toes Right	0.20921	0.03554	0.00616	-0.01797
Talus Right	0.09659	0.00000	0.00000	0.00000
Tibia Right	3.58100	0.00000	-0.18456	0.00000
Femur Right	8.98404	0.00000	-0.19503	0.00000
Pelvis	11.37517	-0.07244	0.00000	0.00000
Femur Left	8.98404	0.00000	-0.19503	0.00000
Tibia Left	3.58100	0.00000	-0.18456	0.00000
Talus Left	0.09659	0.00000	0.00000	0.00000
Calcaneus Left	1.20735	0.10271	0.03081	0.00000
Toes Left	0.20921	0.03554	0.00616	0.01797
Torso	33.06845	0.00693	0.34551	0.03226

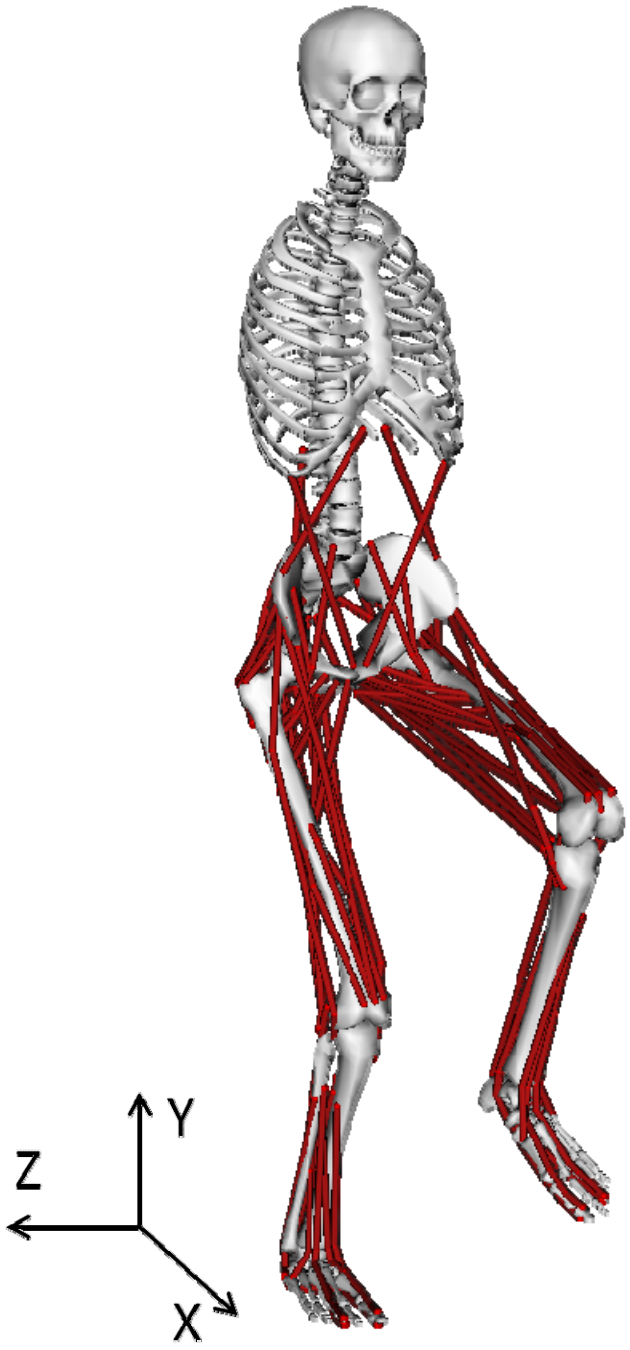

Figure 3-4. The 3-dimension, 10 segment, 15 DOF musculoskeletal model with 92 muscles-tendon actuators (shown in red).

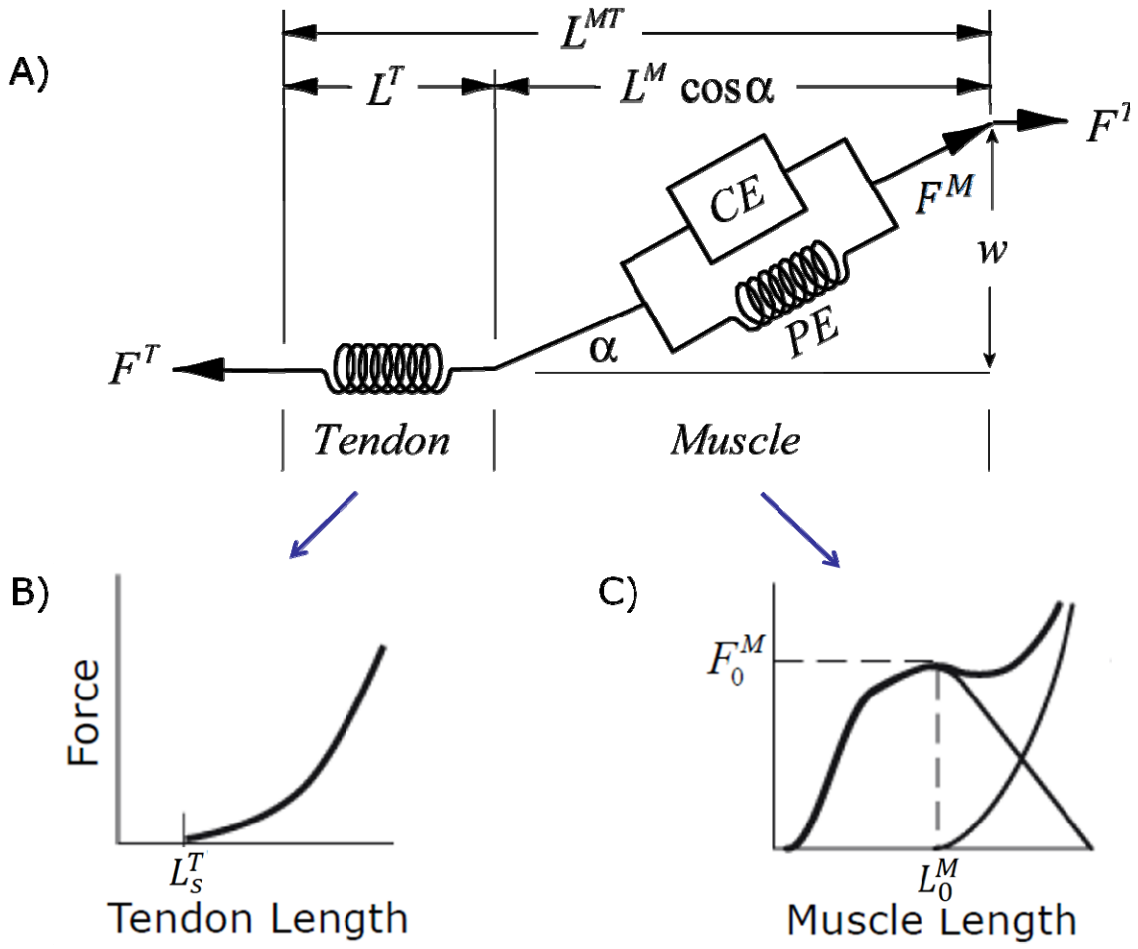

Figure 3-5. Muscle-tendon actuator using a generic Hill-type muscle model (A) with normalized tendon force curve (B) and normalized active and passive muscle force curve (C).

3.2 Data Collection Inclusion Criteria to Define Crouch and Upright Posture

Upright and crouch gait kinematics were recorded in the database at the Center for Gait and Motion Analysis at Gillette Children's Specialty Healthcare, St. Paul, MN and obtained from a previous study (Hicks *et al*., 2008). Subjects with cerebral palsy (aged 6 or older) had to walk with a crouch gait to be included in the study. Arnold *et al*. (2006) defined crouch gait as walking with a knee flexion angle greater than 15° throughout the stance phase with a minimum knee flexion angle of 20° at initial contact. Joints angles of the subjects walking over an entire gait cycle were calculated using a standard clinical protocol to track 3D motion of markers placed on the lower extremity. Joint angles were normalized to a percentage of the gait cycle and averaged for each group. In this study, I used data from the crouch group that exhibit an average of 40° of knee flexion at initial contact. Normal (upright) posture was defined from the average gait data of 83 able-bodied subjects walking at self selected speeds while crouch was defined from the average gait data of 100 subjects with cerebral palsy and crouch gait.

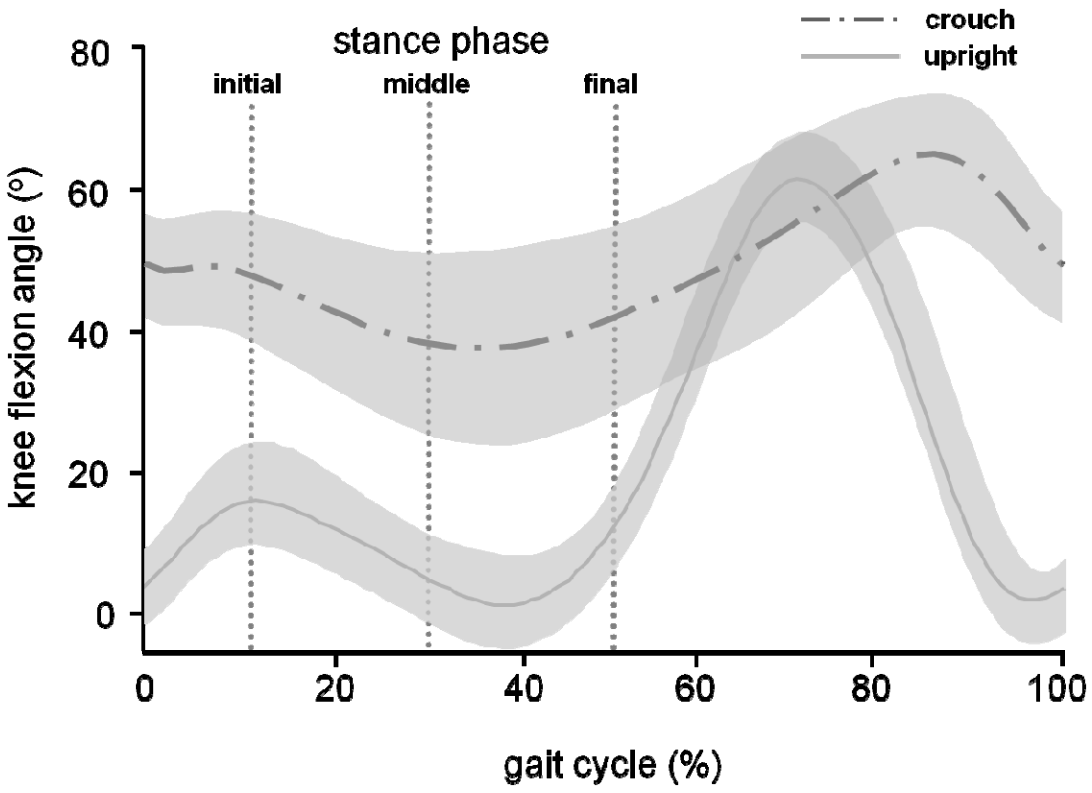

Figure 3-6. Average joint kinematics for upright and crouch gait for the whole gait cycle. The solid line shows the mean values for a group of 83-able bodied children. The dotted line shows the mean values for a group of 100 subjects with cerebral palsy who walked in a crouch gait. Classification of crouch gait is based on the knee flexion angle at initial contact. The bands around both lines show ±1 standard deviation of the mean values. Experimental postures for upright and crouch were taken from the mean values of each group at initial, middle, and final stance.

3.3 Interpolation and Extrapolation to Determine Other Postures

Knee flexion angles for crouch gait shows that subjects adopt a range of gait patterns for walking with a crouch gait (Figure 3-6); the musculoskeletal model was placed in 15 different postures from upright to severe crouch during initial stance at 14% of the gait cycle, middle stance at 32% of the gait cycle, and final stance at 50% of the gait cycle. I linearly interpolated nine additional postures between upright and crouched postures from the experimental data during initial, middle, and final stance (Figure 4). Next, I extrapolated four additional postures (severe crouch) with knee flexion angles greater than crouch. For the initial, middle, and final period of the stance phase, each posture was numbered accordingly: #1 is experimental upright posture, #2 through #10 are interpolated postures, #11 is experimental crouched posture, #12 through #15 are extrapolated postures (severe crouch). The model was placed in a total of 45 postures (15 for each of the three periods) for the study.

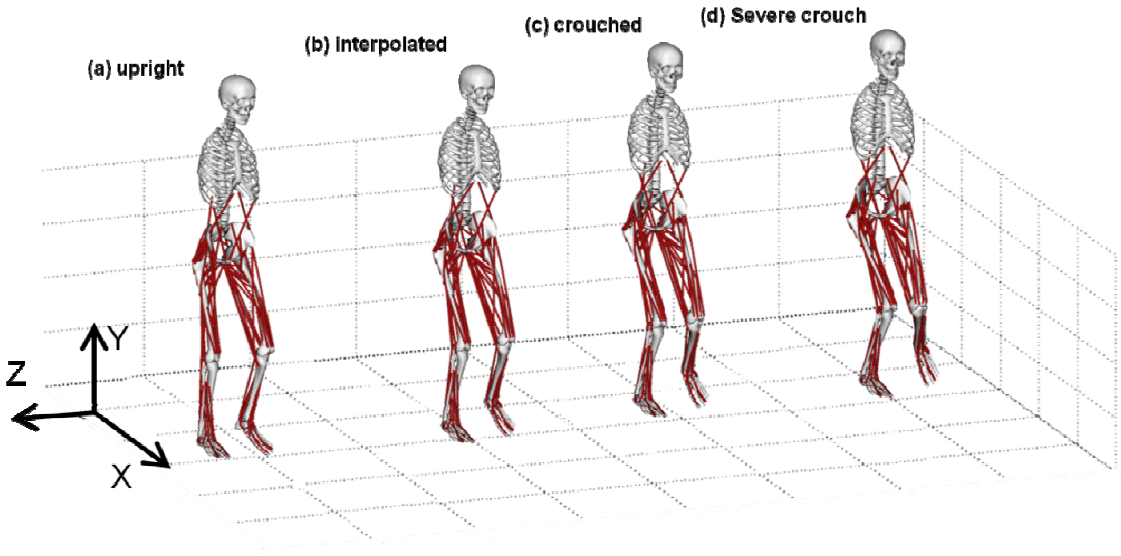

Figure 3-7. Three-dimensional musculoskeletal models placed in 4 (of 15 total) postures during middle stance at 32% gait cycle: (a) experimental upright posture, (b) interpolated posture between experimental upright and crouch data, (c) experimental crouched posture, (d) and extrapolated posture from experimental upright and crouch data (severe crouch).

3.4 Optimization Approach

To determine the relationship between posture and ground reaction forces, a series of optimizations was performed for each posture from upright to severe crouch during initial, middle, and final stance. The optimizer used is an interior point optimizer (IPOPT). IPOPT was developed as a software package for large scale nonlinear optimization (Wächter and Biegler, 2006). It is written in C++ and is available as an open-source software package. IPOPT can find solutions of nonlinear optimization problems of the form

$$\min_{x\ in\ R^n} f(x) \qquad (1)$$

$$such\ that \begin{cases} g_L \leq g(x) \leq g_U \\ x_L \leq x \leq x_U \end{cases} \qquad (2)$$

where the objective function $f(x)$: $R^n \rightarrow R$ and the constraints $g(x)$: $R^n \rightarrow R^m$ are continuously differentiable and can be nonlinear (Wächter and Biegler, 2006). The upper and lower bound of x are x_L and x_U and the upper and lower bound on the constraints are g_L andg_U.

For this study, IPOPT was implemented to find the maximum ground reaction forces in the transverse plane. Ground reaction force is the force exerted on a body in contact with the ground. For most running and prevention studies, the focus is purely on the

vertical ground reaction forces. During walking, however, the ground reaction force will also have a horizontal, or transverse, component parallel to the ground and it is vital to achieve motion and it is the focus of this investigation. To find the maximum transverse ground reaction force, the optimizer can modify the individual muscle-tendon actuator forces acting on the neuromusculoskeletal model. The optimization problem is of the form

$$\underset{f_{muscle}}{\text{maximize}} \quad F_{ground}(f_{muscle}) \tag{3}$$

Such that

$$\frac{F_{ground}}{\|F_{ground}\|} = \hat{v}_{direction} \tag{4}$$

$$0 \leq F_y \tag{5}$$

$$0 \leq \frac{M_z}{F_y} \leq 29cm \tag{6}$$

$$-7cm \leq \frac{-M_z}{F_y} \leq 7cm \tag{7}$$

$$0 \leq f_{muscle} \leq f_{Max\ Isometric} \tag{8}$$

where the objective function is $F_{ground}(f_{muscle})$ and the constraints are discussed below. For each optimization, the ground reaction forces are constraint to be in one of the eight compass directions with the vertical ground reaction force to be greater than or equal to zero. The center of pressure (the point on a body where the total sum of a

pressure acts and causes a force but no moment about that point) was constraint to be under the stance foot of the neuromusculoskeletal model. Each muscle-tendon actuator was constraint to be less than or equal to its maximum isometric force. For each posture during initial, middle, and final stance to find the maximum ground reaction forces in the 8 compass directions.

3.5 C++ Main Program

The open-source IPOPT software package was the optimizer implemented in Microsoft Visual C++ to interface with the neuromusculoskeletal model in OpenSim (Figure 3-8). The neuromusculoskeletal model was first constructed and verified with the number of muscles and total body mass. The posture of the neuromusculoskeletal model was defined based on the posture number (#1-#15). Gravity was set to 9.80665 m/s^2. Next the IPOPT optimizer and target were constructed. The maximum number of iterations was set to 5000 and the convergence tolerance to $1x10^{-6}$. Upper and lower bound for the muscles activation were set from 0 (no activation) to 1 (full activation). The initial guess for muscle activation started at all muscles being activated at 50%. Next, the objective functions of the optimizer were constructed based on equations 4 through equations 8. The optimizer was set to calculate the maximum ground reaction force in the transverse plane based on the objective functions. Once the optimizer determines that the ground reaction force in the transverse plane is the maximum, the results were recorded and written in a text file. If the optimizer fails or was unable to settle on the

maximum value, iterations were increased. If optimizer was still unable to settle on a maximum value, the tolerance was increased to $1x10^{-5}$. This process was looped to run through all of the postures (#1-#15) for the 8 directions of a compass. The “pseudo code” can be seen in Figure 3-9 and the full C++ main code can be seen in Appendix 9.7.

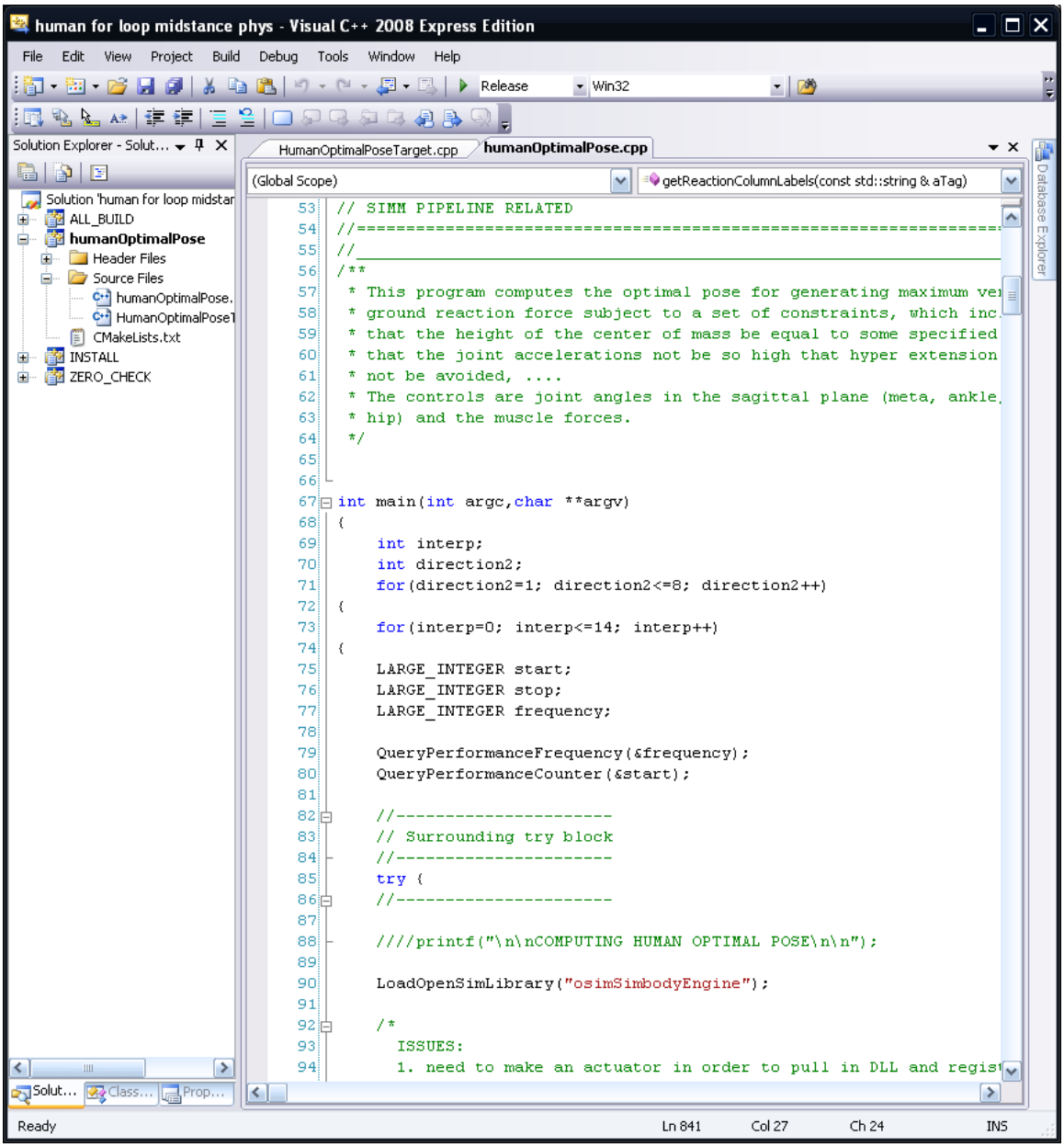

Figure 3-8. Screenshot of the C++ code to find the maximum ground reaction forces in the transverse plane

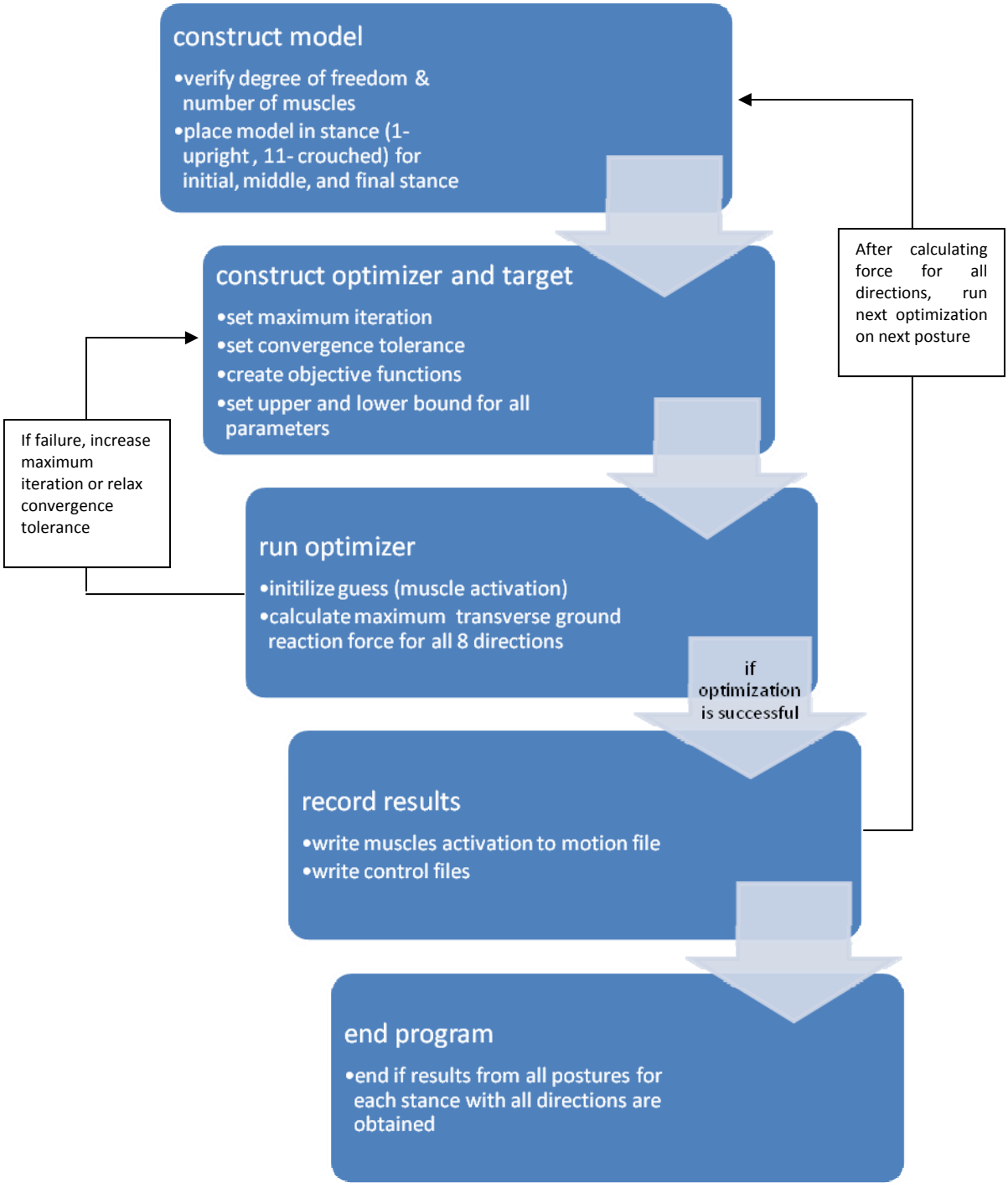

Figure 3-9. Pseudo-code of C++ Program

3.6 Force Profile Generation

A ground reaction force profile was generated for each posture by finding the area of the forces generated in the 8 compass directions (Figure 3-0) in MATLAB®. The force vectors provided the vertices and the area was calculated using polyarea in MATLAB®. Using the generated force profile area from initial, middle, and final stance, the results were interpolated to show the force profile areas over the entire stance phase of gait in MATLAB®.

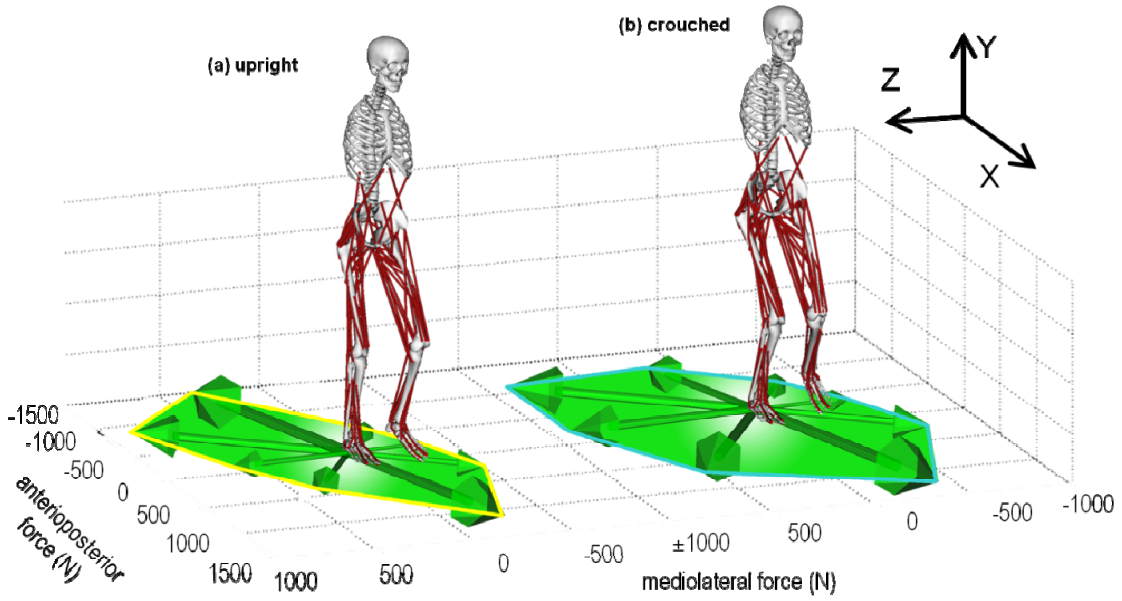

Figure 3-10. Ground reaction force profile generation for upright posture (left model) and crouched posture (right model). The force profile consists of forces in the 8 compass direction generated from each optimization steps.

4 RESULTS

4.1 Results During Initial Stance

The maximum ground reaction forces that were generated were normalized to the model's body weight (BW = 712 N). During initial stance, posterior ground reaction force is the largest for upright (2.1 x BW) while the lateral ground reaction force (1.37 x BW) are larger as posture approaches towards crouch. Anteriorly, posture #9 was able to produce transverse ground reaction force at around 1.5 x BW. Posture #5 (interpolated posture between upright and crouch) allowed the largest transverse ground reaction force averaged over all 8 directions during initial stance (Figure 4-1). The average force of the interpolated posture #5 was 7% larger than upright and 6% larger than crouched. The trend continues as it postures approaches severe crouch, with posture # 5 having a 12% larger ground reaction force.

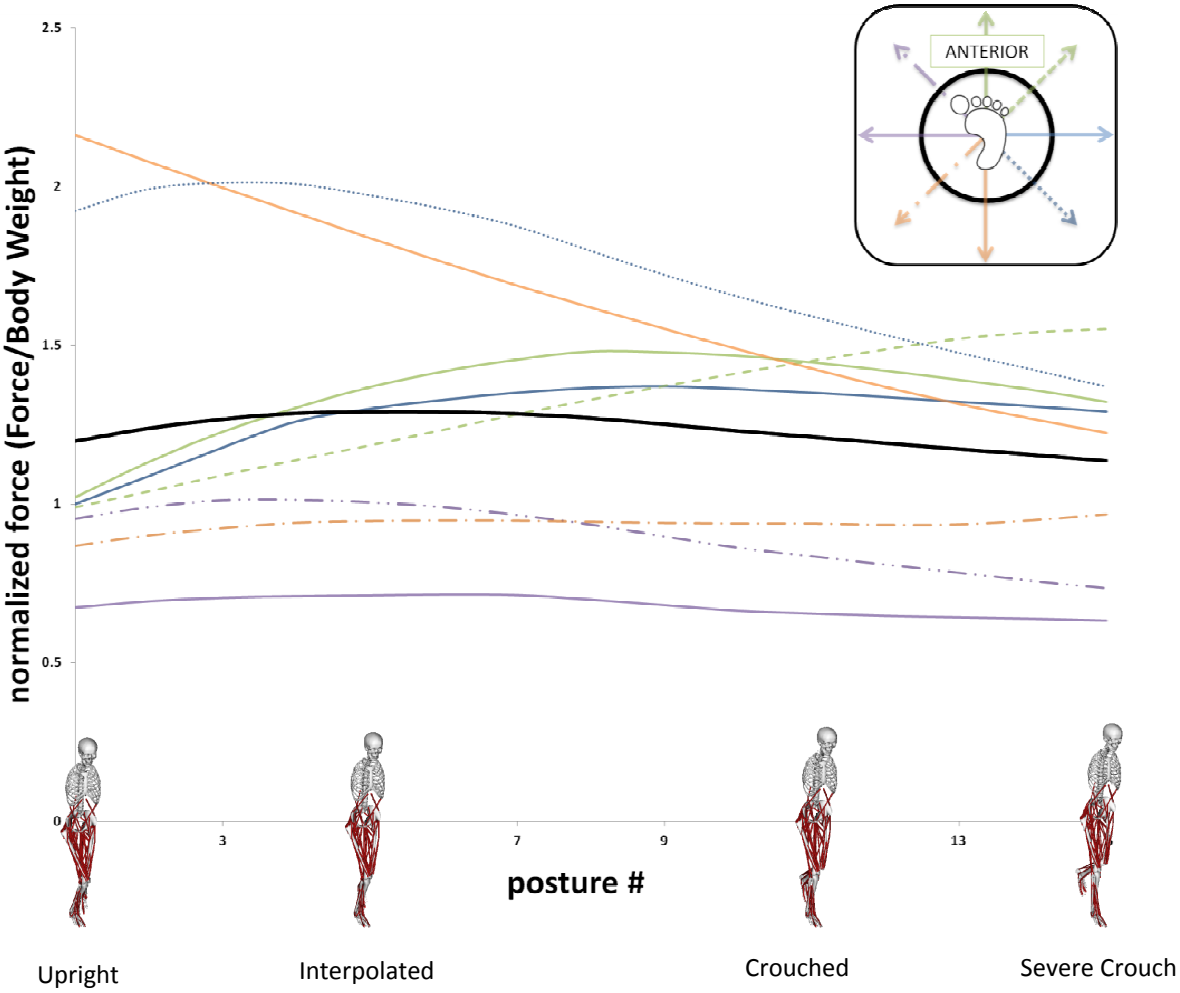

Figure 4-1. Maximum ground reaction forces in the transverse plane for postures during initial stance normalize to model's body weight (712 N). The direction of the forces can be determined by the key to the upper right hand side. For instance, the solid purple line is the ratio of the maximum medial ground reaction force over the model's body weight for all postures during initial stance. The black solid line is the average of all the ground reaction forces in the transverse plane.

4.2 Results During Middle Stance

The crouched posture allowed the largest transverse ground reaction force averaged over all 8 directions during middle stance (Figure 4-2). The average force of crouch (posture 11) was 12% larger than upright (posture 1) and 4% larger than severe crouch (posture 15). The average force of crouch was only slightly larger (<1%) than posture 10. Upright postures (1 - 5) allowed the largest ground reaction forces in the anterior and posterior directions. Posterior ground reaction force decreased as posture went from upright to crouched.

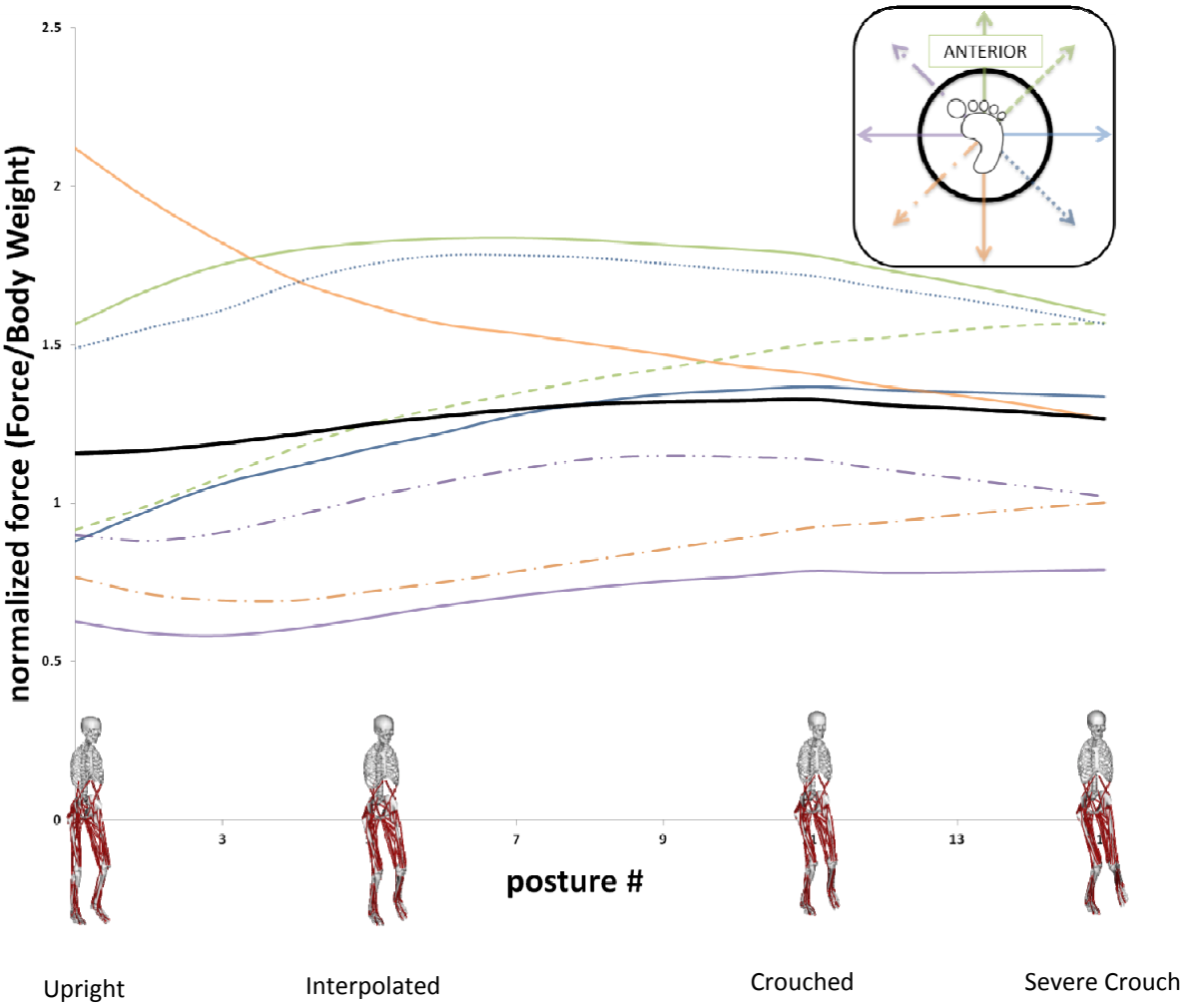

Figure 4-2. Maximum ground reaction forces in the transverse plane for postures during middle stance normalize to model's body weight (712 N). The direction of the forces can be determined by the key to the upper right hand side. For instance, the solid purple line is the ratio of the maximum medial ground reaction force over the model's body weight for all postures during middle stance. The black solid line is the average of all the ground reaction forces in the transverse plane.

4.3 Results During Final Stance

Similar to posture during initial stance, the crouched posture allowed the largest transverse ground reaction force averaged over all 8 directions during final stance (Figure 4-3). However, the average force of crouch (posture 11) was only 5% larger than upright (posture 1) and 3% larger than severe crouch (posture 15). There was a dip in the average ground reaction forces as posture went from upright to interpolated postures and peaked as it approaches crouched.

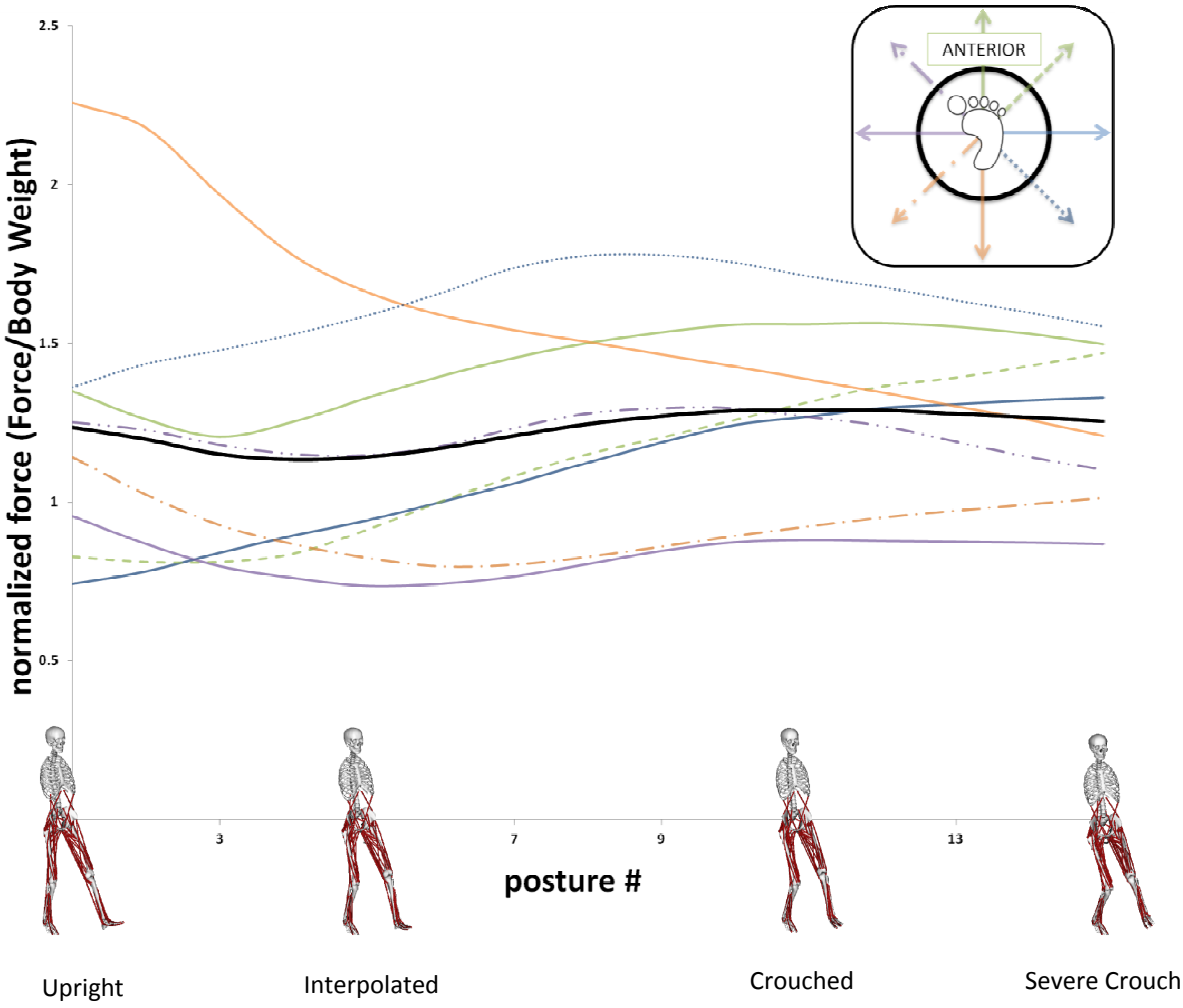

Figure 4-3. Maximum ground reaction forces in the transverse plane for postures during final stance normalize to model's body weight (712 N). The direction of the forces can be determined by the key to the upper right hand side. For instance, the solid purple line is the ratio of the maximum medial ground reaction force over the model's body weight for all postures during final stance. The black solid line is the average ratio of all the ground reaction forces in the transverse plane.

4.4 Maximum Force Profile

The hypothesis was evaluated by comparing the force profile area between postures. From the maximum ground reaction forces generated for postures during initial, middle, and final stance, a force profile area was generated for the whole stance phase. A range of crouched postures allowed the largest ground reaction force profile area during the stance phase of gait (Figure 4-4). Over the stance phase, the maximum force profile areas occurred between mild crouch (#5) and severe crouch postures (#12) from initial stance to final stance. During initial stance, interpolated postures (#4-6) between upright and crouch allowed the largest ground reaction force profiles. These postures produced force profile areas within 1% of each other, with posture #5 being the largest (2.582 kN2). Comparatively, experimental upright (#1) and experimental crouched (#11) postures had 12-13% (2.265 and 2.272 kN2, respectively) smaller force profile areas, and severe crouch (#15) was roughly 23% smaller (1.999 kN2). The crouched posture (#11) during middle stance produced the largest force profile area (2.676 kN2) with this trend continuing to final stance. Postures #8-12 produced force profile area within 2% of each other. During final stance, a posture between crouch and severe crouch (#12) allowed the largest ground reaction force profiles (2.514 kN2); however, this force profile area was less than 2% larger compared to crouch (#11). The force profile area (2.487 kN2) of experimental crouch was 7.3% higher compared to experimental upright and 4% higher than severe crouch during final stance.

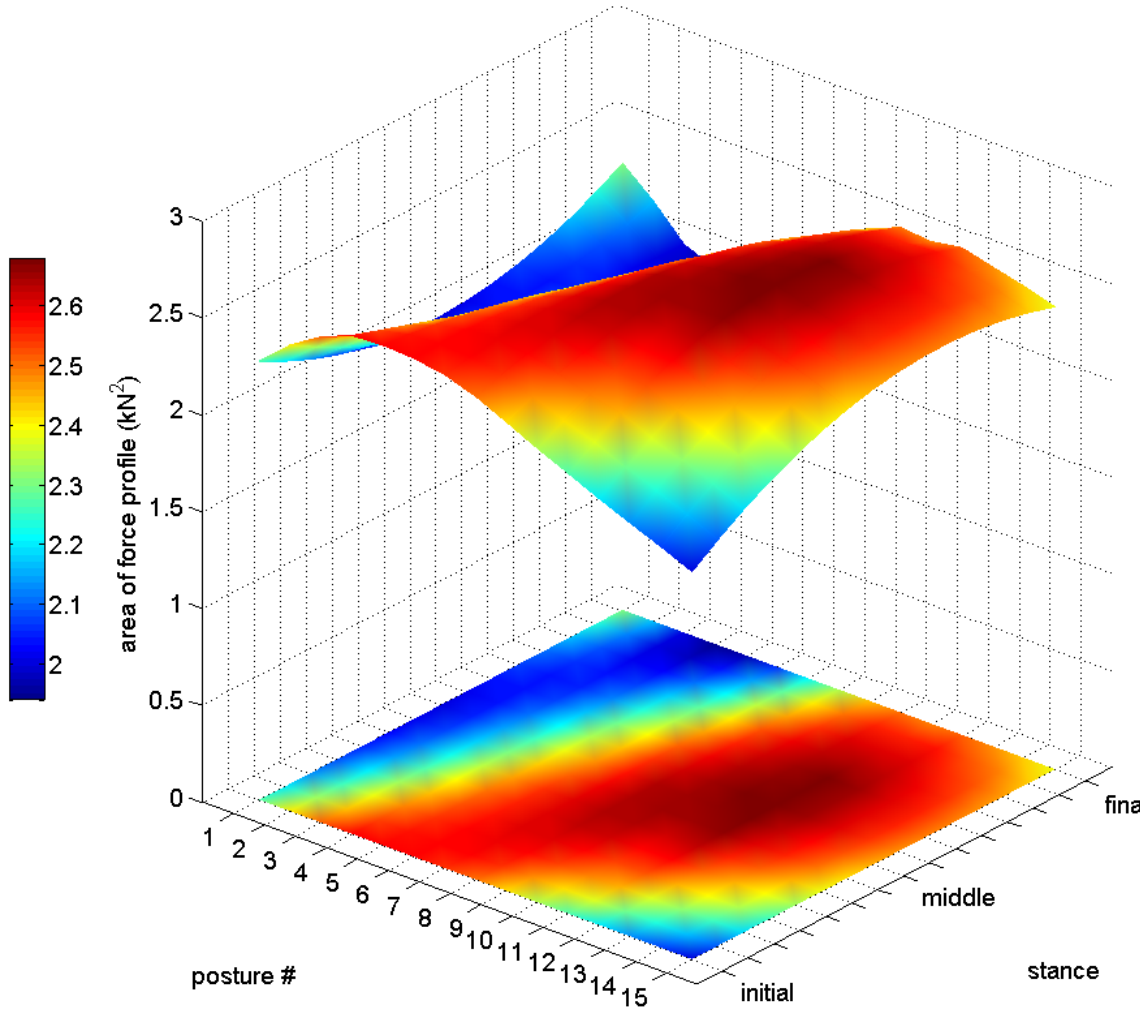

Figure 4-4. Areas of ground reaction force profiles across three parts of stance and across all postures (intermediate force profile areas between initial-middle-final generated with a cubic spline interpolation). Force profile areas throughout stance are from lowest (blue) to highest (red). During early stance, mild crouched postures (#4-6) allowed the greatest forces. During late stance, crouched postures (#9-11) allowed greater forces compared to upright.

5 DISCUSSION

5.1 Assumptions and Research Challenges

There were several assumptions and challenges present in our study and the results should be interpreted in context with our research challenges.

Biomechanical Model Selection

Our musculoskeletal model did not incorporate any skeletal abnormalities, such as tibial torsion, commonly seen in children with cerebral palsy walking with crouch gait (Novacheck *et al*., 2010). Muscle paths are altered in skeletal deformities which may contribute to misalignment of the body (Cornell, 1997; Laplaza *et al*., 1993)]. Our study was focused on examining the different postures and their influence on ground reaction forces. Incorporating bone deformities such as tibial torsion would add additional variables to the investigation, making it difficult to elucidate the effects of ground reaction force relating to the different postures. Finally, the arms in our musculoskeletal model were omitted due to the lack of an upper extremity model with muscles. However, the mass properties of the arms were included in the torso. In a running simulation (Hamner *et al*., 2010), the arms accounted for less than 1% of both the maximum horizontal and vertical mass center acceleration and therefore its contribution to propulsion and support were minimal.

Static Optimization

First, the optimization procedure implemented to calculate the maximum ground reaction force was static rather than dynamic optimization. Dynamic optimization involves minimizing or maximizing the cost objective function over a period of time; this was not implemented in our study as the model was placed in a given posture while the muscles were able to generate force. Hence, static optimization was better suited for our study. Anderson and Pandy (2001) showed that static optimization was equivalent to dynamic optimization in biomechanics.

5.2 Comparison of Results with Literature and Experiments

Vertical Ground Reaction Force During Crouch

Our study is fundamentally different from Hicks *et al*. (2008), which examined the effect of crouch postures on the capacity of muscles to extend the hip and knee joints. Their study used induced acceleration analysis (Zajac and Gordan, 1989) to determine the joint angular accelerations towards extension resulting from the application of 1 N muscle force to the musculoskeletal model. The joint angular accelerations resulting from the induced acceleration analysis reflects the influence of muscle geometry and posture on the capability of each muscle's contribution to extend the hip and knee joints. Their study showed almost the entire major hip and knee extensors' capacities were reduced in crouch gait. This finding suggests a reduction in the ability to generate

vertical ground reaction force. In this study, optimization was used to maximize horizontal ground reaction forces in the transverse plane without regard for the vertical ground reaction force. However, a vertical ground reaction force is necessary to achieve the horizontal ground reaction forces. This study suggests an increase in the ability to generate these horizontal forces.

Muscle Activation Generated with Optimization versus Experimental EMG data

The optimal muscle activations to achieve the maximum ground reaction forces in the transverse plane were compared to EMG data for normal walking kinematics (Besier *et al*, 2009) and crouch gait kinematics (Steele, 2010) and can be seen in Table 5-1. The rectus femoris activations obtained from experimental EMG were quite smaller than the activations found through optimization. They were about twice as large compared to the experimental EMG. Muscle activations obtained from optimization for biceps femoris long head, semimembranosus, and gastrocnemius were higher than experimental measured EMG activations, but were within 1 standard deviation of the experimentally measured mean. The higher activations obtained from the optimization may be attributed to our body's tendency to minimize energy consumption. The optimization, however, is trying to maximize ground reaction forces in the transverse plane and is, therefore, not concerned with energy consumption.

Table 5-1. Experimental EMG muscle activation for upright walking and walking in crouch gait and activation generated from optimizer during middle stance (32% of whole gait cycle)

	mean activation (0 – 1)			
muscle	exp. upright (Besier *et al*., 2009)	exp. crouch (Steele *et al*., 2010)	opt. upright	opt. crouch
rectus femoris	0.20 ± 0.10	0.35 ± 0.10	0.4012	0.6096
biceps femoris long head	0.35 ± 0.15	0.45 ± 0.15	0.5137	0.4288
semimembranosus	0.41 ± 0.10	0.30 ± 0.10	0.5002	0.3750
gastrocnemius	0.42 ± 0.05	0.60 ± 0.15	0.3757	0.5568

5.3 Sensitivity Analysis

To test the sensitivity of the neuromusculoskeletal model, total body mass was scaled by ±3% of the total body mass in the scale tool in OpenSim. The mass scaled models were placed in 15 postures as before and the transverse ground reaction force profile was generated for each posture during middle stance. Similarly, the neuromusculoskeletal model was scaled by ±3% of each individual body segments and the transverse ground reaction force profile was generated for each posture during middle stance. Force profile area in the transverse plane for the scaled models were generated and compared to the un-scaled model.

For the mass scaled (0.97) model, changing the mass did not yield vastly different results and had similar trends to the un-scaled model. The only values it changed were the ground reaction force profiles for postures #9-11 and it was only by 0.51%. Having a more crouched posture (severe crouch) did not particularly alter the force profile by changing the total body mass. This same trend can be seen in the other mass scaled (1.03) model created as well.

The segment length scaled (0.97 & 1.03) models had very similar trends to each other. The biggest percent difference between the segment length scaled and un-scaled models was only around 1.38%. However, it is interesting to note that the “smaller” scaled model was able to produce larger transverse ground reaction force profile area

compared to both the scaled and “bigger” model. The model is not particularly sensitive to changing the mass and body segments. With a 3% change in either segment length or mass, the results were only off by 1.38% or less.

Table 5-2. Percent difference for model scaled by 0.97 of the total mass and 0.97 of each body lengths

posture #	un-scaled model Force Profile Area (kN^2)	mass of model scaled by 0.97		length of bodies scaled by 0.97	
		Force Profile Area (kN^2)	% diff	Force Profile Area (kN^2)	% diff
1	2.0076	2.0076	0.00	2.0135	0.29
2	2.0472	2.0472	0.00	2.0539	0.32
3	2.1371	2.1371	0.00	2.1450	0.36
4	2.2558	2.2558	0.00	2.2641	0.36
5	2.3815	2.3815	0.00	2.3921	0.44
6	2.4795	2.4795	0.00	2.4910	0.46
7	2.5688	2.5687	0.00	2.5837	0.57
8	2.6327	2.6327	0.00	2.6513	0.70
9	2.6532	2.6632	0.37	2.6838	1.15
10	2.6605	2.6764	0.59	2.6973	1.38
11	2.6764	2.6605	0.59	2.6829	0.24
12	2.6255	2.6255	0.00	2.6483	0.86
13	2.5813	2.5813	0.00	2.6038	0.87
14	2.5264	2.5264	0.00	2.5484	0.87
15	2.4536	2.4536	0.00	2.4803	1.08

Table 5-3. Percent difference for model scaled by 1.03 of the total mass and 1.03 of each body lengths

posture #	un-scaled model Force Profile Area (kN^2)	mass of model scaled by 1.03		length of bodies scaled by 1.03	
		Force Profile Area (kN^2)	% diff	Force Profile Area (kN^2)	% diff
1	2.0076	2.0076	0.00	2.0015	0.30
2	2.0472	2.0472	0.00	2.0405	0.33
3	2.1371	2.1371	0.00	2.1292	0.37
4	2.2558	2.2558	0.00	2.2472	0.38
5	2.3815	2.3815	0.00	2.3691	0.52
6	2.4795	2.4795	0.00	2.4673	0.49
7	2.5688	2.5687	0.00	2.5530	0.62
8	2.6327	2.6327	0.00	2.6146	0.69
9	2.6532	2.6632	0.37	2.6435	0.36
10	2.6605	2.6764	0.59	2.6566	0.15
11	2.6764	2.6605	0.59	2.6400	1.36
12	2.6255	2.6255	0.00	2.6042	0.81
13	2.5813	2.5813	0.00	2.5607	0.80
14	2.5264	2.5264	0.00	2.5004	1.03
15	2.4536	2.4536	0.00	2.4253	1.15

5.4 Ground Reaction Forces during Walking

To compare the ground reaction forces generated from the optimization to a real life walking situation, an experiment was prepared to collect data of transverse ground reaction forces for healthy individuals walking. The force plate used was AMTI measured at 1200 Hz. There were a total of 6 participants in the experiment. Participants had various bodyweights, so the ground reaction force obtained was normalized by body weight of the individual. Other variables such as height of the participant, shoes worn by the participant, etc. were not controlled. Participants were asked to walk at a self selected speed to land his or her right foot over a force plate. Once the right foot makes contact, the participants were asked to do either a side step to the medial, lateral, anterior, or posterior. For each direction, 3 samples were collected for each participant. Participants were allowed to practice before data was collected.

The normalize force obtained from the experiment can be seen in Table 5-4. Results obtained looked similar to the force generated for upright posture during final stance. For crouch posture, the experimental mean were smaller than the generated normalized forces from the optimizer, although not unreasonable. The optimizer was able to generate larger normalized ground reaction force in the lateral direction compared to medial for the crouched posture. Participants in this experiment did not feel this way as they felt like they were more unbalanced side stepping laterally. For the upright posture, however, the optimizer was able to generate higher forces in the

medial direction, similar to the trend seen in the experiment. The experimental mean in the medial direction (0.85 ± 0.07) was higher than the lateral direction (0.68 ± 0.09). Only the normalized force in the posterior direction generated by the optimizer were exceptionally high (2.2 x BW) compared the experimental mean (0.93 ± 0.16 x BW) and the experimental max (1.23 x BW). Regarding generating posterior ground reaction force, some participants stated that they felt they could produce the highest force in this direction. This "feeling" might be a result of stepping more vertically in this direction and not generating as high of transverse ground reaction force. The normalized ground reaction force in the vertical direction was over 2.5 times body weight.

This experiment generally produced lower normalized ground reaction forces. There may be several reasons for this. Participants were unable to step completely in the desire direction. For example, participants stepping in the posterior direction always had component medially and laterally. This is true for the other side stepping procedures as well. Furthermore, the optimizer is able to activate any muscles to within its peak isometric force to attempt to maximize these transverse ground reaction forces. While we have gross motor control of our bodies, we are unable to activate individual muscles in our bodies like the optimizer can to the neuromuscular model.

Table 5-4. Normalized transverse ground reaction force obtained from experiment

	normalize force (F/BW)			
direction	experimental mean	experimental max	opt. upright (final stance)	opt. crouch (final stance)
anterior	1.07 ± 0.25	1.25	1.34	1.56
posterior	0.93 ± 0.16	1.23	2.2	1.38
medial	0.85 ± 0.07	0.93	0.95	0.88
lateral	0.68 ± 0.09	0.76	0.74	1.27

5.5 Groundwork for Creating Predictive Software for Patients with Cerebral Palsy

The treatment of crouch gait and cerebral palsy is complex, with outcomes being unpredictable, and often times unsuccessful. Treatments are often given without quantitative data to justify treatments. Treatments forcing patients with cerebral palsy into an upright posture may not be beneficial. There may be more basic rational that patient reverts to a crouch posture. The brain is affected in patients with cerebral palsy. This decreased in control may cause the brain to use a more basic controller than what is normally available. The increased in transverse ground reaction force profile area for a crouch posture may allow a subject to be more balance and allow a subject to move more medial and lateral to compensate for decreased in motor control of patients with cerebral palsy.

The tools developed from this study can be used in a clinical environment to predict possible outcomes for patients suffering from crouch gait and cerebral palsy. Data for this study was average for all patients included in the study, but a patient-specific model could be generated for individuals in a clinical setting. The neuromusculoskeletal model can be scaled for any individuals to help predict the transverse ground reaction forces that the patient can generate given their gait analysis. For instance, a patient with cerebral palsy with crouch gait may not need to be completely upright in a normal gait. The patient would go in the clinic to take gait data to generate a patient-specific

neuromuscular model. We may find that a slightly less crouch posture may help the patient generate a larger ground reaction force profile. This may help in postural balance due to the decreased in motor control. Optimization techniques, however, only allow researchers to determine the best transverse ground reaction force given the muscles parameters, but it would be impossible for clinicians to tell patients to "activate" only certain muscles during walking. However, this lays the foundation for utilizing optimization techniques to help answer questions that may lead to possibly predicting outcomes in the future.

5.6 Evolutionary reason for Crouch

Not only are there implications for crouch gait in biomechanics, but the increased in transverse ground reaction force profile area may be the result of evolution that we crouch in general. Furthermore, several other species of bipedal animals such as ostriches walk in a crouch gait. As humans, we crouch for all sorts of activities. Our ancestors probably crouched when they sensed danger. This allowed them to equally react and run for a given danger in any direction. Crouch may allow other animals this same advantage for a larger ground reaction force profile area for an enhanced "flight" phase in reaction to danger. For us in the modern time, we crouch when we are unbalance on a train or a bus. In sports, the "ready" stance for most activities is a crouch posture. This allows the athlete to move in all directions rather that just anterior and

posterior as in an upright posture. It also allows an athlete to oppose other athletes as seen in American football.

6 CONCLUSION

6.1 Mechanical Advantage of Crouch Gait

The goal of this comparative study was to examine how posture influences ground reaction forces generated by muscles. We found that the force profile area for initial stance was highest for postures near mild crouch and decreases as postures move towards upright and crouch. The force profile area increased during middle stance as postures change from mild crouch to crouch and decreased as postures move beyond crouch to severe crouch. The trend for final stance was similar to that of middle stance except that upright showed a slight increase. Our results show that postures between mild crouch and severe crouch postures were able to produce the largest force profile area during the stance phase of gait.

Despite the research challenges, we can draw several conclusions from this study. First, the overall ability to generate larger ground reaction forces and force profile areas represents a mechanical advantage of a crouched posture. This advantage results from an increased capacity of muscles to generate ground reaction forces. This increase in muscle capacity while in a crouched posture may allow a patient to generate new movements to compensate for impairments associated with cerebral palsy, such as motor control deficits. Furthermore, this increase in muscle capacity to generate

horizontal ground reaction forces may also rationalize the advantage an athlete gains when adopting a crouch posture in sports.

6.2 Future Work

There are several possible directions this thesis can continue in for future studies.

Patient-specific Neuromusculoskeletal Model

Currently, the neuromusculoskeletal model is not patient-specific and the kinematic data used to determine postures is averaged from the population of this study that met the inclusion criteria. Furthermore, the neuromusculoskeletal model was used for both upright postures as well as crouch gait postures. However, a more robust and improved model can be developed using patient specific data. X–ray computed tomography (CT) and magnetic resonance imaging (MRI) can be used to determine bone surfaces. Several studies have investigated estimating muscles attachments and parameters. Kaptein and van der Helm (2004) estimated muscle attachments contours through deformation of bones meshes obtained from CT and MR images. Scheys *et al*. (2009) presented a novel approach to define line-of-actions for muscles using non-rigid registration between atlas images and MR images. Using patient-specific information, an “optimal” model can be developed to be used to predict treatment outcomes.

Predicting Treatment Outcomes

Patient-specific model is a powerful tool for clinicians and biomechanist to predict treatment outcomes of patients with cerebral palsy. A possible retrospective study would include measuring a patient's gait and data before treatment and comparing the data after treatment. A force profile can be generated for the patient "pre" and "post" treatment to determine if the patient has improved force generation profile. There may be factors or some connection that can be used to predict treatment outcomes for future patients.

Implementation of Optimization Techniques in other Fields

Neuromusculoskeletal modeling is not reserved just for studying human movement; this work can be implemented into other fields such as evolutionary biology. Techniques from this study can be used to study other bipedal animals to understand the trade-off between weight support and maneuverability. This trade-off determines how animals choose postures for different body sizes/morphologies and different behaviors (Biewener, 1989). As shown from this study, crouch posture increases horizontal ground reaction forces. Other studies, however, have shown that there is a compromise in energy costs expenditure and lower vertical ground reaction force generation during crouch (Hicks *et al*., 2008; Rose *et al*., 1990). While others have investigated the effect of added mass on metabolism (Taylor *et al*., 1980) or muscle energetics (Ellerby & Marsh, 2006), the influence of mass on control of muscle forces is poorly understood,

especially in a comparative context. This future study can provide new knowledge about how land animals support their mass and may inspire novel ways for legged robots to carry loads or propose alternatives to managing musculoskeletal health in obese individuals.

This thesis has shown that computer simulations are valuable tools for analyzing movement and its application to understanding and treating movement abnormalities. However, there is considerable amount of future works required to create patient-specific models and using it to predict treatment outcomes. Insights gained from utilizing computer simulations can be used to improve the quality of life for those suffering from movement abnormalities as well as other musculoskeletal and neuromusculoskeletal disorders.

7 GLOSSARY

Abduction	Movement away from the midline of the body in the coronal plane.
Acceleration	The time rate of change of velocity.
Adduction	Movement towards the midline of the body in the coronal plane.
Ankle motion	The ankle angles reflect the motion of the foot segment relative to the shank segment.
Anterior	The front or before, also referred to as ventral.
Coccyx	The tailbone located at the distal end of the sacrum.
Constraint functions	Specific limits that must be satisfied by the optimal design.
Degree of freedom (DOF)	A single coordinate of relative motion between two bodies. Such a coordinate responds without constraint or imposed motion to externally applied forces or torques. For translational motion, a DOF is a linear coordinate along a single direction. For rotational motion, a DOF is an angular coordinate about a single, fixed axis.
Design variables	Variables that change to optimize the design.
Distal	Away from the point of attachment or origin.
Dorsiflexion	Movement of the foot towards the anterior part of the tibia in the sagittal plane.
Eversion	A turning outward.
Extension	Movement that rotates the bones comprising a joint away from each other in the sagittal plane.
Femur	The longest and heaviest bone in the body. It is located between the hip joint and the knee joint.
Final Stance	The period of time just before foot leaves contact with the ground.
Flexion	Movement that rotates the bones comprising a joint towards each other in the sagittal plane.
Force	A push or a pull and is produced when one object
Force plate	A transducer that is set in the floor to measure about some specified point, the force and torque applied by the foot to the ground. These devices provide measures of the three components of the resultant ground reaction force vector and the three components of the resultant torque vector.
Gait	A manner of walking or moving on foot.

Generalized coordinates	A set of coordinates (or parameters) that uniquely describes the geometric position and orientation of a body or system of bodies. Any set of coordinates that are used to describe the motion of a physical system.
Hip motion	The hip angles reflect the motion of the thigh segment relative to the pelvis.
Inferior	Below or at a lower level (towards the feet).
Initial stance	The period of time when the foot first contact with the ground.
Inverse dynamics	Analysis to determine the forces and torques necessary to produce the motion of a mechanical system, given the topology of how bodies are connected, the kinematics, the mass properties, and the initial condition of all degrees of freedom.
Inversion	A turning inward.
Kinematics	Those parameters that are used in the description of movement without consideration for the cause of movement abnormalities. These typically include parameters such as linear and angular displacements, velocities and accelerations.
Kinetics	General term given to the forces that cause movement. Both internal (muscle activity, ligaments or friction in muscles and joints) and external (ground or external loads) forces are included. The moment of force produced by muscles crossing a joint, the mechanical power flowing to and from those same muscles, and the energy changes of the body that result from this power flow are the most common kinetic parameters used.
Knee abduction-adduction	Motion of the long axis of the shank within the coronal plane as seen by an observer positioned along the anterior-posterior axis of the thigh.
Knee flexion-extension	Motion of the long axis of the shank within the sagittal plane as seen by an observer positioned along the medial-lateral axis of the thigh.
Knee internal-external rotation	Motion of the medial-lateral axis of the shank with respect to the medial-lateral axis of the thigh within the transverse plane as viewed by an observer positioned along the longitudinal axis of the shank.
Knee motion	The knee angles reflect the motion of the shank segment relative to the thigh segment.

Lateral	Away from the body's longitudinal axis, or away from the mid-sagittal plane.
Markers	Active or passive objects (balls, hemispheres or disks) aligned with respect to specific bony landmarks used to help determine segment and joint position in motion capture.
Medial	Toward the body's longitudinal axis, or toward the mid-sagittal plane.
Middle Stance	The period of time between initial foot contact with the ground and just before foot leaves the ground.
Mid-sagittal plane	The plane that passes through the midline and divides the body or body segment into the right and left halves.
Model parameters	A set of coordinates that uniquely describes the model segments lengths, joint locations, and joint orientations, also referred to as joint parameters. Any set of coordinates that are used to describe the geometry of a model system.
Moment of force	The moment of force is calculated about a point and is the cross product of a position vector from the point to the line of action for the force and the force. In two-dimensions, the moment of force about a point is the product of a force and the perpendicular distance from the line of action of the force to the point. Typically, moments of force are calculated about the center of rotation of a joint.
Motion capture	Interpretation of computerized data that documents an individual's motion.
Objective functions	Figures of merit to be minimized or maximized.
Parametric	Of or relating to or in terms of parameters, or factors that define a system.
Passive markers	Joint and segment markers used during motion capture that reflect visible or infrared light.
Pelvis	Consists of the two hip bones, the sacrum, and the coccyx. It is located between the proximal spine and the hip joints.
Posterior	The back or behind, also referred to as dorsal.
Proximal	Toward the point of attachment or origin.
Range of motion	Indicates joint motion excursion from the maximum angle to the minimum angle.

Sacrum	Consists of the fused components of five sacral vertebrae located between the 5th lumbar vertebra and the coccyx. It attaches the axial skeleton to the pelvic girdle of the appendicular skeleton via paired articulations.
Sagittal plane	The plane that divides the body or body segment into the right and left parts.
Skin movement artifacts	The relative movement between skin and underlying bone.
Stance phase	The period of time when the foot is in contact with the ground.
Subtalar joint	Located between the distal talus and proximal calcaneous, also known as the talocalcaneal joint.
Superior	Above or at a higher level (towards the head).
Swing phase	The period of time when the foot is not in contact with the ground.
Talus	The largest bone of the ankle transmitting weight from the tibia to the rest of the foot.
Tibia	The large medial bone of the lower leg, also known as the shinbone. It is located between the knee joint and the talocrural joint.
Transverse plane	The plane at right angles to the coronal and sagittal planes that divides the body into superior and inferior parts.
Velocity	The time rate of change of displacement.

8 LIST OF REFERENCES

[1] Arnold, A.S., Anderson, F.C., Pandy, M.G., Delp, S.L., 2005. Muscular contributions to hip and knee extension during the single limb stance phase of normal gait: a framework for investigating the causes of crouch gait. Journal of Biomechanics 38, 2181–2189.

[2] Arnold, S., Liu, M., Schwartz, M., Õunpuu, S., Delp, S., 2006. The role of estimating muscle-tendon lengths and velocities of the hamstrings in the evaluation and treatment of crouch gait, Gait & Posture, Volume 23: 273-281.

[3] Bell, K.J., Ounpuu, S., DeLuca, P.A., Romness, M.J., 2002. Natural progression of gait in children with cerebral palsy. Journal of Pediatric Orthopaedics 22, 677–682.

[4] Brand, R. A, Pedersen, D. R., and Friederich, J. A, 1986. "The sensitivity of muscle force predictions to changes in physiologic cross-sectional area," J. Biomech., vol. 19: 589-596.

[5] Campbell, J. and Ball, J, 1978. Energetics: application to the study and management of locomotors disabilities - Energetics in walking in cerebral palsy, Orthopedic Clinics of North America, Volume 9: 374-377.

[6] Cornell, M.S., 1995. The hip in cerebral palsy, Developmental Medicine and Child Neurology, Volume 37, 3-18.

[7] Delp SL, Anderson FC, Arnold AS, Loan P, Habib A, John CT, Guendelman E, Thelen DG., 2007. OpenSim: open-source software to create and analyze dynamic simulations of movement. IEEE Trans Biomed Eng.; 54 (11), 1940-50.

[8] Friederich, J. A. and Brand, R A., 1990. "Muscle fiber architecture in the human lower limb," J. Biomech., vol. 23: 91-95.

[9] Gage, J.R., Schwartz, M.H., 2001. Dynamic deformities and lever arm considerations. In: Paley, D. (Ed.), Principles of Deformity. Springer, Berlin, 761–775.

[10] Gage, J.R., Schwartz, M.H., 2004. Pathological gait and lever-arm dysfunction. In: Gage, J.R. (Ed.), The Treatment of Gait Problems in Cerebral Palsy. Mac Keith Press, London, 180–204.

[11] Hamer, S.R., Seth, A., Delp, S.L., 2010. Muscle contributions to propulsion and support during running. Journal of Biomechanics 43 (14), 2709-2716.

[12] Hicks, J., Arnold, A., Anderson, F., Schwartz, M., Delp, S., 2007. The effect of excessive tibial torsion on the capacity of muscles to extend the hip and knee during single-limb stance, Gait & Posture, Volume 26: 546-552.

[13] Hicks, J.L., Schwartz, M., Arnold, A.S., and Delp S.L., 2008. Crouched postures reduce the capacity of muscles to extend the hip and knee during the single-limb stance phase of gait. Journal of Biomechanics 41, 960–967.

[14] Hill, A. V., 1938. “The Heat of Shortening and the dynamic of Constants of Muscle,” Proceedings of the Royal Society of London Biological Sciences, Volume 126: 136-195.

[15] Hoffinger, S.A., Rab, G.T., Abou-Ghaida, H., 1993. Hamstrings in cerebral palsy crouch gait. Journal of Pediatric Orthopaedics 13, 722–726.

[16] Honeycutt, A., Dunlap, L., Chen, H., al Homsi, G., Grosse, S., Schendel, D., 2004. Economic Costs Associated with Mental Retardation, Cerebral Palsy, Hearing Loss, and Vision Impairment --- United States. Morbidity and Mortality Weekly Report 53 (03), 57-59.

[17] Hoy, M. G., Zajac, F. E., and Gordon, M. E., 1990. "A musculoskeletal model of the human lower extremity: the effect of muscle, tendon, and moment ann on the moment-angle relationship of musculotendon actuators at the hip, knee, and ankle," J. Biomech., vol. 23: 157-169

[18] Inman, 1976 V.T. Inman, The Joints of the Ankle, Williams and Wilkins Co., Baltimore, MD.

[19] Jahnsen, R., Villien, L., Aamodt, G., Stanghelle, J.K., Holm, I., 2004. Musculoskeletal pain in adults with cerebral palsy compared with the general population. Journal of Rehabilitation Medicine 36, 78–84.

[20] Johnston, T. E., Moore, S. E., Quinn, L. T. and Smith, B. T.,2004. Energy cost of walking in children with cerebral palsy: relation to the Gross Motor Function Classification System, Developmental Medicine & Child Neurology, 46(1), 34–38

[21] Kruse, M., Michelsen, S., Flachs, E., Bronnum-Hansen, H., Madsen, M., Uldall, P., 2009. Lifetime Costs of Cerebral Palsy, Developmental Medicine & Child Neurology, 51(8), 662-628.

[22] Laplaza, F.J., Root, L., Tassanawipas, A., Glassser, D.B., 1993. Femoral Torsion and neck-shaft angles in cerebral palsy, Journal of Pediatric Orthopedics, Volume 13, 192-199.

[23] McNee, A.E., Shortland, A.P., Eve, L.C., Robinson, R.O., Gough, M., 2004. Lower limb extensor moments in children with spastic diplegic cerebral palsy. Gait and Posture 20, 171–176.

[24] Pandy, M., and Anderson, F., 1999. Dynamic optimization of human gait, Journal of Biomechanics, Volume 31: 115.

[25] Perry, J., Antonelli, D., Ford, W., 1975. Analysis of knee-joint forces during the flexed-knee stance. Journal of Bone and Joint Surgery, Volume 57A, 961-967.

[26] Rose, J., Gamble, J.G., Burgos, A., Medeiros, J., Haskell, W.L., 1990. Energy expenditure index of walking for normal children and for children with cerebral palsy. Developmental Medicine and Child Neurology 32, 333–340.

[27] Rose, J., Gamble, J.G., Medeiros, J., Burgos, A., Haskell, W.L., 1989. Energy cost of walking in normal children and in those with cerebral palsy: comparison of heart rate and oxygen uptake. Journal of Pediatric Orthopaedics 9, 276–279.

[28] Ross, S.A., Engsberg, J.R., 2007. Relationships Between Spasticity, Strength, Gait, and the GMFM-66 in Persons With Spastic Diplegia Cerebral Palsy, Archives of Physical Medicine and Rehabilitation, 88(9), 1114-1120

[29] Steele, K., Seth, A., Hicks, J., Schwartz, M., Delp, S., 2010. Muscle contributions to support and progression during single-limb stance in crouch gait, Journal of Biomechanics, Volume 43: 2099-2105.

[30] Tom F. Novacheck, Joyce P. Trost, Sue Sohrweide, 2010. Examination of the Child with Cerebral Palsy, Orthopedic Clinics of North America, Volume 41, Issue 4, Orthopedic Management of Cerebral Palsy, 469-488.

[31] Wächter, A. and Biegler, L. T., 2006. On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming, Mathematical Programming 106(1): 25-57

[32] Wickiewicz, T. L., Roy, R R, Powell, P. L., and Edgerton, V. R, 1983. "Muscle architectuie of the human lower limb," CUn. Orthop. Rei. Res., vol. 179: 275-283.

[33] Wren, T.A., Rethlefsen, S., Kay, R.M., 2005. Prevalence of specific gait abnormalities in children with cerebral palsy: influence of cerebral palsy subtype, age, and previous surgery. Journal of Pediatric Orthopaedics 25, 79–83.

[34] Zajac, F. E., 1989. "Muscle and tendon: properties, models, scaling, and application to biome¬chanics and motor control," CRC Critical Reviews in Biomedical Engineering. Volume. 17, 359-411.

9 APPENDIX

9.1 Individual Muscle-tendon Actuators Maximum Isometric Force

Table 9-1. Maximum Isometric Force for Each Individual Muscle-tendon Actuators

No.	Muscle	Max Isometric Force (N)
1	glut_med1_r	819
2	glut_med2_r	573
3	glut_med3_r	653
4	glut_min1_r	270
5	glut_min2_r	285
6	glut_min3_r	323
7	semimem_r	1288
8	semiten_r	410
9	bifemlh_r	896
10	bifemsh_r	804
11	sar_r	156
12	add_long_r	627
13	add_brev_r	429
14	add_mag1_r	381
15	add_mag2_r	343
16	add_mag3_r	488
17	tfl_r	233
18	pect_r	266
19	grac_r	162
20	glut_max1_r	573
21	glut_max2_r	819
22	glut_max3_r	552
23	iliacus_r	1073
24	psoas_r	1113
25	quad_fem_r	381
26	gem_r	164
27	peri_r	444
28	rect_fem_r	1169
29	vas_med_r	1294
30	vas_int_r	1365

31	vas_lat_r	1871
32	med_gas_r	1558
33	lat_gas_r	683
34	soleus_r	3549
35	tib_post_r	1588
36	flex_dig_r	310
37	flex_hal_r	322
38	tib_ant_r	905
39	per_brev_r	435
40	per_long_r	943
41	per_tert_r	180
42	ext_dig_r	512
43	ext_hal_r	162
44	ercspn_r	2500
45	intobl_r	900
46	extobl_r	900
47	glut_med1_l	819
48	glut_med2_l	573
49	glut_med3_l	653
50	glut_min1_l	270
51	glut_min2_l	285
52	glut_min3_l	323
53	semimem_l	1288
54	semiten_l	410
55	bifemlh_l	896
56	bifemsh_l	804
57	sar_l	156
58	add_long_l	627
59	add_brev_l	429
60	add_mag1_l	381
61	add_mag2_l	343
62	add_mag3_l	488
63	tfl_l	233
64	pect_l	266
65	grac_l	162

66	glut_max1_l	573
67	glut_max2_l	819
68	glut_max3_l	552
69	iliacus_l	1073
70	psoas_l	1113
71	quad_fem_l	381
72	gem_l	164
73	peri_l	444
74	rect_fem_l	1169
75	vas_med_l	1294
76	vas_int_l	1365
77	vas_lat_l	1871
78	med_gas_l	1558
79	lat_gas_l	683
80	soleus_l	3549
81	tib_post_l	1588
82	flex_dig_l	310
83	flex_hal_l	322
84	tib_ant_l	905
85	per_brev_l	435
86	per_long_l	943
87	per_tert_l	180
88	ext_dig_l	512
89	ext_hal_l	162
90	ercspn_l	2500
91	intobl_l	900
92	extobl_l	900

9.2 Initial Stance Results

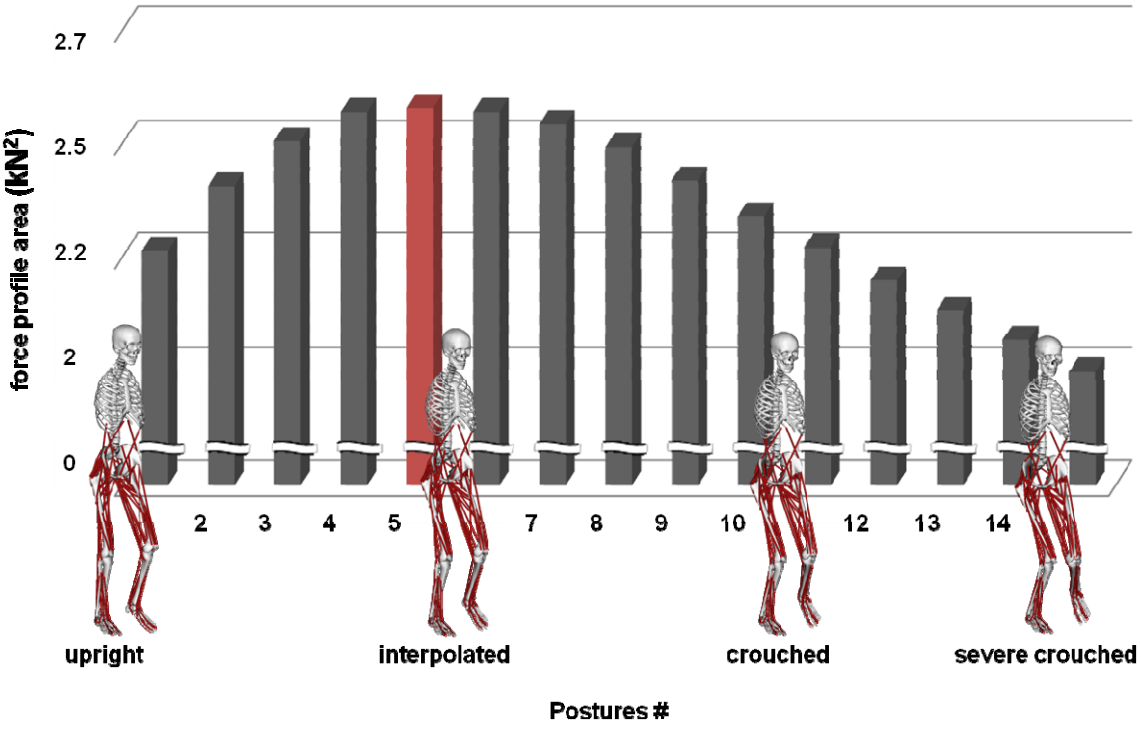

Figure 9-1. Initial Stance Force Profile Area

9.3 Middle Stance Results

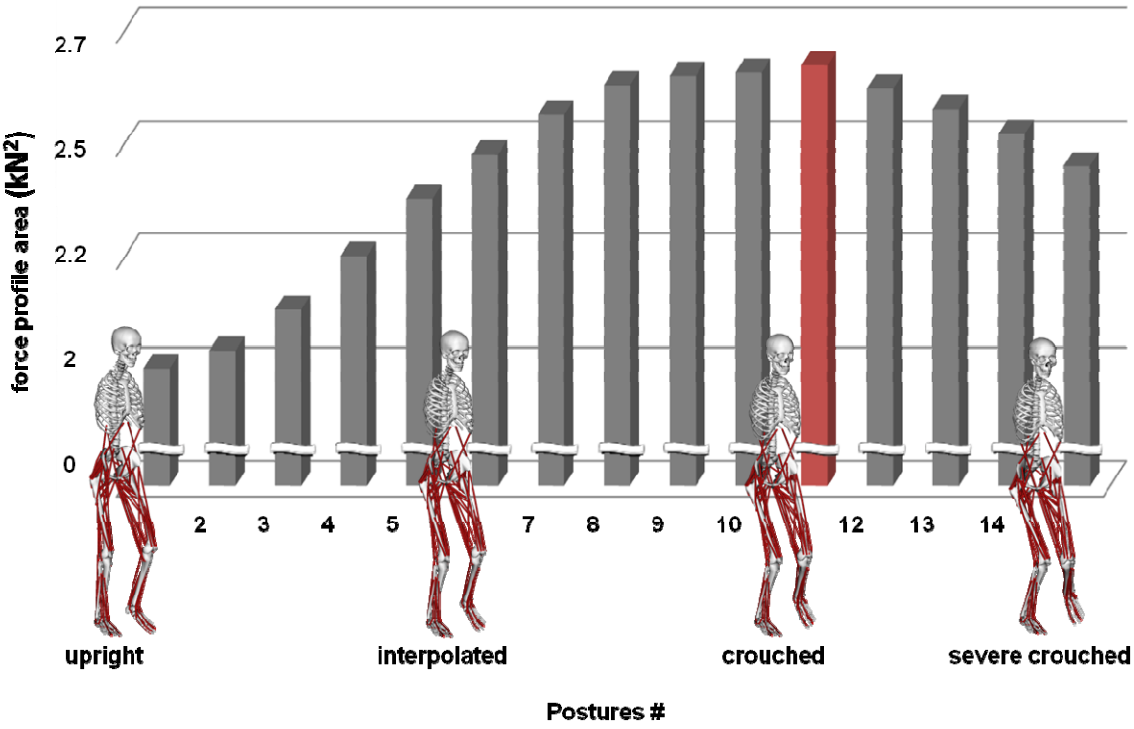

Figure 9-2. Middle Stance Force Profile Area

9.4 Final Stance Results

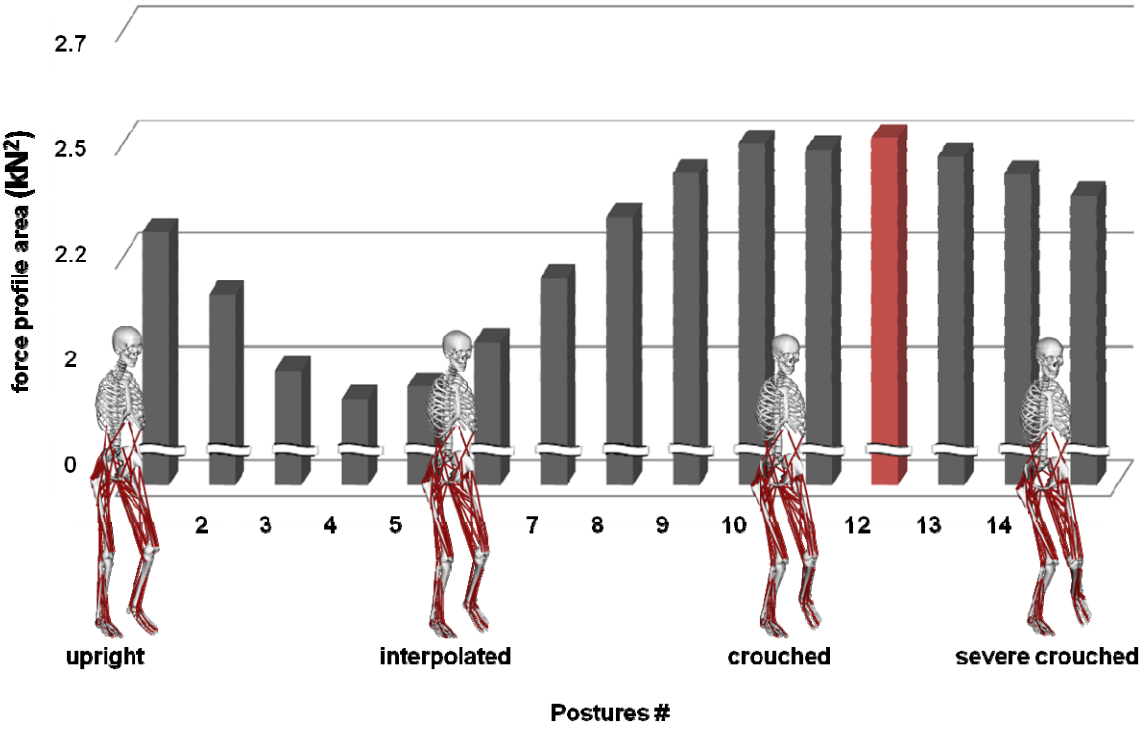

Figure 9-3. Final Stance Force Profile Area

9.5 Force Profile Surface Initial Stance to Middle Stance

Table 9-2. Force Profile surface - Initial Stance to Middle Stance

		Stance Phase							
		1	2	3	4	5	6	7	8
Posture #	1	2.26509	2.23934	2.21359	2.18784	2.16210	2.13635	2.11060	2.00760
	2	2.40699	2.37101	2.33503	2.29906	2.26308	2.22710	2.19112	2.04720
	3	2.50756	2.47051	2.43347	2.39642	2.35938	2.32233	2.28528	2.13710
	4	2.57292	2.54121	2.50950	2.47779	2.44607	2.41436	2.38265	2.25580
	5	2.58163	2.56161	2.54160	2.52159	2.50158	2.48156	2.46155	2.38150
	6	2.57322	2.56385	2.55448	2.54510	2.53573	2.52636	2.51699	2.47950
	7	2.54630	2.54855	2.55080	2.55305	2.55530	2.55755	2.55980	2.56880
	8	2.49279	2.50678	2.52077	2.53476	2.54875	2.56275	2.57674	2.63270
	9	2.41913	2.44254	2.46595	2.48935	2.51276	2.53617	2.55957	2.65320
	10	2.34146	2.37336	2.40527	2.43717	2.46907	2.50098	2.53288	2.66050
	11	2.27150	2.31199	2.35248	2.39297	2.43346	2.47395	2.51444	2.67640
	12	2.19971	2.24229	2.28487	2.32745	2.37003	2.41260	2.45518	2.62550
	13	2.13328	2.17808	2.22288	2.26768	2.31249	2.35729	2.40209	2.58130
	14	2.06879	2.11455	2.16031	2.20607	2.25183	2.29759	2.34335	2.52640
	15	1.99966	2.04505	2.09044	2.13584	2.18123	2.22663	2.27202	2.45360

9.6 Force Profile Surface Middle Stance to Final Stance

Table 9-3. Force Profile Surface - Middle Stance to final Stance

		Stance Phase							
		8	9	10	11	12	13	14	15
Posture #	1	2.03741	2.06723	2.09704	2.12685	2.15667	2.18648	2.30573	2.03741
	2	2.05937	2.07154	2.08371	2.09588	2.10805	2.12022	2.16890	2.05937
	3	2.12365	2.11020	2.09675	2.08330	2.06985	2.05641	2.00261	2.12365
	4	2.22412	2.19245	2.16077	2.12910	2.09742	2.06575	1.93905	2.22412
	5	2.34002	2.29854	2.25706	2.21558	2.17409	2.13261	1.96669	2.34002
	6	2.43781	2.39613	2.35444	2.31275	2.27107	2.22938	2.06263	2.43781
	7	2.53242	2.49603	2.45965	2.42326	2.38688	2.35049	2.20495	2.53242
	8	2.60343	2.57416	2.54489	2.51562	2.48635	2.45708	2.34000	2.60343
	9	2.63153	2.60986	2.58819	2.56652	2.54485	2.52318	2.43651	2.63153
	10	2.64469	2.62888	2.61307	2.59725	2.58144	2.56563	2.50238	2.64469
	11	2.65751	2.63862	2.61973	2.60084	2.58195	2.56306	2.48750	2.65751
	12	2.61438	2.60325	2.59213	2.58101	2.56988	2.55876	2.51427	2.61438
	13	2.57047	2.55963	2.54880	2.53796	2.52713	2.51629	2.47295	2.57047
	14	2.51714	2.50788	2.49862	2.48936	2.48009	2.47083	2.43379	2.51714
	15	2.44697	2.44034	2.43371	2.42708	2.42045	2.41382	2.38730	2.44697

9.7 Main Program in Microsoft Visual C++

```
//*****************************************************************************
********
// humanOptimalPose.cpp
// This file contains the main routine for computing optimal pose and
// ground reaction forces that can be generated for.
//*****************************************************************************
********

//=============================================================================
// INCLUDES
//=============================================================================
#include <iostream>
#include <direct.h>
#include <OpenSim/Common/rdMath.h>
#include <OpenSim/Common/Mtx.h>
#include <OpenSim/Common/IO.h>
#include <OpenSim/Common/Storage.h>
#include <OpenSim/Simulation/Model/BodySet.h>
#include <OpenSim/Simulation/Model/CoordinateSet.h>
#include <OpenSim/Common/rdOptimizationTarget.h>
#include <simmath/Optimizer.h>
#include "HumanOptimalPoseTarget.h"
#include <SimTKCommon/Constants.h>
#include <OpenSim/Actuators/GeneralizedForceAtv.h>
#include <OpenSim/Common/LoadOpenSimLibrary.h>
#include <OpenSim/Simulation/Model/AbstractMuscle.h>
#include "convert.h"

using namespace std;
using namespace OpenSim;


//=============================================================================
// DEFINES
//=============================================================================
#define MAXLEN 2048

//=============================================================================
// INTERNAL GLOBALS
//=============================================================================
static char tmp[MAXLEN],tmp1[MAXLEN];
static Model *_Model = NULL;

//=============================================================================
// DECLARATIONS
//=============================================================================
Storage* generateHumanOptimalPoseFile();
char* getControlDescription();
char* getReactionDescription();
Array<string> getControlColumnLabels(const string &aTag);
Array<string> getReactionColumnLabels(const string &aTag);
```

```
//=========================================================================
==
// SIMM PIPELINE RELATED
//=========================================================================


int main(int argc,char **argv)
{
      int interp;
      int direction2;
      for(direction2=1; direction2<=8; direction2++)
{
      for(interp=0; interp<=14; interp++)
{
      LARGE_INTEGER start;
      LARGE_INTEGER stop;
      LARGE_INTEGER frequency;

      QueryPerformanceFrequency(&frequency);
      QueryPerformanceCounter(&start);

      //----------------------
      // Surrounding try block
      //----------------------
      try {
      //----------------------

      ////printf("\n\nCOMPUTING HUMAN OPTIMAL POSE\n\n");

      LoadOpenSimLibrary("osimSimbodyEngine");


      GeneralizedForceAtv *atv = new GeneralizedForceAtv();
      delete atv;

      // CONSTRUCT THE MODEL
      _Model  =  new  Model("human_1792_open_loop.osim");  //for  initial
and mid stance
      _Model->setup();
      _Model->printDetailedInfo(cout);
      ActuatorSet *actSet = _Model->getActuatorSet();
      AbstractDynamicsEngine &engine = _Model->getDynamicsEngine();
      BodySet *bodySet = engine.getBodySet();

      int nstates = _Model->getNumStates();
      //// cout << "nstates: " << nstates << endl;
      int ncontrols = _Model->getNumControls();
      //// cout << "ncontrols: " << ncontrols << endl;

      // VARIABLES
      int i,a;
      int nq = _Model->getNumCoordinates();
      int nu = _Model->getNumSpeeds();
      int ny = _Model->getNumStates();
```

```
	int na = _Model->getNumActuators();
	int nb = _Model->getNumBodies();
	int nmus = 92;
	int ndofs = 14;
	int nx = ndofs + nmus;
	double t=0.0;
	double *q = new double[nq];  for(i=0;i<nq;i++) q[i]=0.0;
	double *qang = new double[nu];  for(i=0;i<nu;i++) qang[i]=0.0;
	double  *qAngAndForce  =  new  double[nu+6+na];    for(i=0;i<nu;i++)
qAngAndForce[i]=0.0;
	double *u = new double[nu];  for(i=0;i<nu;i++) u[i]=0.0;
	double *dqdt = new double[nq];
	double *dudt = new double[nu];
	double *y = new double[ny];
	double *dy = new double[ny];
	SimTK::Vector x(nx);
	double frc[3],trq[3],acc[3],fg[3];;

	// COMPUTE TOTAL BODY MASS
	double massTotal=0.0;
	AbstractBody *body;
	for(i=1;i<nb;i++) {
		body = bodySet->get(i);
		massTotal += body->getMass();
	}


	// CHECK NUMBER OF MUSCLES

	if(na<=0) {
		printf("The model must be actuated by 1 or more muscles...
quitting.\n");
		exit(0);
	}

	// OUTPUT STORAGE
	// generalized coordinate storage
	Storage *poseStore = generateHumanOptimalPoseFile();
	// control
	Storage *controlStore = new Storage();
	controlStore->setName("HumanOptimalGroundReaction");
	controlStore->setDescription(getControlDescription());
	controlStore-
>setColumnLabels(getControlColumnLabels("activation"));
	// reaction
	Storage *reactionStore = new Storage();
	reactionStore->setName("HumanOptimalGroundReaction");
	reactionStore->setDescription(getReactionDescription());
	reactionStore->setColumnLabels(getReactionColumnLabels("fx"));

	// CONSTRUCT OPTIMIZER AND TARGET
	HumanOptimalPoseTarget *target = new
		HumanOptimalPoseTarget(nx,_Model);
	SimTK::OptimizerAlgorithm algorithm = SimTK::InteriorPoint;
```

```
      SimTK::Optimizer           *optimizer           =           new
SimTK::Optimizer(*target,algorithm);
      optimizer->setDiagnosticsLevel(3);
      optimizer->setMaxIterations(5000);
      double ConvgTol = 1.0e-5;
      optimizer->setConvergenceTolerance(ConvgTol);

      // STATES
      target->setQ(q);  // sets pointer
      target->setU(u);  // sets pointer
      target->setY(y);  // sets pointer
      target->setActivationConstant(0.050);
      target->setDX(1.0e-6);

      // UPPER AND LOWER BOUNDS
      double min =  0.0;            // When using mucles.
      double max =  1.0;
      SimTK::Vector lower(nx),upper(nx);
      lower = min;
      upper = max;
      target->setParameterLimits(lower,upper);

      // CENTER OF MASS HEIGHTS
      double trunkAng;
      double M,ICM[3][3];
      SimTK::Vec3 COM(0);
      engine.getSystemInertia(&M,COM,ICM);
      ////printf("COM = %lf %lf %lf\n",COM[0],COM[1],COM[2]);
      double initialHeight,finalHeight;
      // FINAL ------------------

      // INITIAL STANCE
      for(i=0;i<nq;i++) q[i]=0.0;
      //// cout << "\nTop Configuration" << endl;

      // Normal Mean: upright
      for(i=0;i<nq;i++) q[i]=0.0;
      //// Initial
      cout << "\n Initial Stance" << endl;
      double                    normal_mean_14_q[11]                    =
{11.89295382,4.02955269,3.142365187,30.24057565,5.531013419,0.604470773
,19.04943742,0.282528417,3.154743166,1.095314854,-3.866155672};
      double    normal_mean_64_q[11]    =    {11.91092833,-4.139727891,-
3.446532362,8.056052816,-6.61425588,-5.496872729,47.01785257,-
2.464463848,5.101164785,-20.22711601,-1.968889635};
      q[0] = q[13] = 0.0; // mtp_angle
      q[1] = -6.34262078; q[12] = 0.0; // subtalar_angle
      q[2]  =  normal_mean_14_q[9];    q[11]  =  normal_mean_64_q[9];  //
ankle_angle
      q[3] = -normal_mean_14_q[6]; q[10] = -normal_mean_64_q[6]; // (-)
knee_angle
      q[4]  =  normal_mean_14_q[3];  q[7]  =  normal_mean_64_q[3];  //
hip_flexion
```

```
      q[5]  =  normal_mean_14_q[4];  q[8]  =  normal_mean_64_q[4];  //
hip_adduction
      q[6]  =  normal_mean_14_q[5];  q[9]  =  normal_mean_64_q[5];  //
hip_rotation
      q[14]    =    -1.840295;    //    -1.840295//    -9.88451436;    //
lumbar_extension
      q[15] = 2.445147; // 2.445147 // 0.93527406; // lumbar_bending
      q[16]   =   0.322333321;   //   0.322333321   //   5.86548147;   //
lumbar_rotation


      CoordinateSet          *coordinateSet          =          _Model-
>getDynamicsEngine().getCoordinateSet();

      for(i=0;i<coordinateSet->getSize();i++) {
            //// cout << coordinateSet->get(i)->getName() << " = " <<
q[i] << endl; // print out name & angles
      }

      engine.convertDegreesToRadians(q,q);

      engine.setConfiguration(q,u);

      _Model->getDynamicsEngine().getSystemInertia(&M,COM,ICM);
      trunkAng = (q[2]+q[3]+q[4]+q[14])*SimTK_RADIAN_TO_DEGREE;
      printf("\n\nM = %lf\n",M);
      printf("COM = %lf %lf %lf\n",COM[0],COM[1],COM[2]);
      printf("trunk angle = %lf\n",trunkAng);
      finalHeight = COM[1];

      // INITIAL ------------------
      for(i=0;i<nq;i++) q[i]=0.0;
      double offset = 0.0;
      //// cout << "\nBottom Configuration" << endl;

      // Normal Mean:
      for(i=0;i<nq;i++) q[i]=0.0;

      // Severe Mean: (crouch)
      //// Midstance
      for(i=0;i<nq;i++) q[i]=0.0;



//initial stance
switch(interp){

case 0:

q[0] = 0 ;  q[1] = -6.34262078 ;    q[2] = 1.095314854 ;    q[3]   =   -
19.04943742 ;     q[4] = 30.24057565 ;    q[5] = 5.531013419 ;    q[6]
= 0.604470773 ;   q[7] = 8.056052816 ;    q[8] = -6.61425588 ;    q[9]
= -5.496872729 ;  q[10] = -47.01785257 ;  q[11] = -20.22711601 ;  q[12]
```

```
= 0 ; q[13] = 0 ; q[14] = -1.840295 ;     q[15]    =    2.445147    ;
      q[16] = 0.322333321 ;
break;

case 1:

q[0] = 0 ;  q[1] = -6.161502192 ;   q[2] = 2.8034307736 ;   q[3]   =   -
21.761175896 ;    q[4] = 31.406020485 ;   q[5] = 5.1733570928 ;   q[6]
= 1.8005394657 ;  q[7] = 8.7965021284 ;   q[8] = -6.0309944219 ;  q[9]
= -3.7910233231 ; q[10] = -47.271034811 ; q[11]   =   -17.5375141566   ;
      q[12] = 0 ; q[13] = 0 ; q[14] = -1.840295 ;     q[15]  =  2.445147
;     q[16] = 0.322333321 ;
break;

case 2:

q[0] = 0 ;  q[1] = -5.980383604 ;   q[2] = 4.5115466932 ;   q[3]   =   -
24.472914372 ;    q[4] = 32.57146532 ;    q[5] = 4.8157007666 ;   q[6]
= 2.9966081584 ;  q[7] = 9.5369514408 ;   q[8] = -5.4477329638 ;  q[9]
= -2.0851739172 ; q[10] = -47.524217052 ; q[11]   =   -14.8479123032   ;
      q[12] = 0 ; q[13] = 0 ; q[14] = -1.840295 ;     q[15]  =  2.445147
;     q[16] = 0.322333321 ;
break;

case 3:

q[0] = 0 ;  q[1] = -5.799265016 ;   q[2] = 6.2196626128 ;   q[3]   =   -
27.184652848 ;    q[4] = 33.736910155 ;   q[5] = 4.4580444404 ;   q[6]
= 4.1926768511 ;  q[7] = 10.2774007532 ;  q[8] = -4.8644715057 ;  q[9]
= -0.379324511300001 ;  q[10] = -47.777399293 ; q[11]  =  -12.1583104498
;     q[12] = 0 ; q[13] = 0 ; q[14] = -1.840295 ;     q[15]  =  2.445147
;     q[16] = 0.322333321 ;
break;

case 4:

q[0] = 0 ;  q[1] = -5.618146428 ;   q[2] = 7.9277785324 ;   q[3]   =   -
29.896391324 ;    q[4] = 34.90235499 ;    q[5] = 4.1003881142 ;   q[6]
= 5.3887455438 ;  q[7] = 11.0178500656 ;  q[8] = -4.2812100476 ;  q[9]
= 1.3265248946 ;  q[10] = -48.030581534 ; q[11] = -9.4687085964 ; q[12]
= 0 ; q[13] = 0 ; q[14] = -1.840295 ;     q[15] = 2.445147 ;      q[16]
= 0.322333321 ;
break;

case 5:

q[0] = 0 ;  q[1] = -5.43702784 ;    q[2] = 9.635894452 ;    q[3]   =   -
32.6081298 ;      q[4] = 36.067799825 ;   q[5] = 3.742731788 ;    q[6]
= 6.5848142365 ;  q[7] = 11.758299378 ;   q[8] = -3.6979485895 ;  q[9]
= 3.0323743005 ;  q[10] = -48.283763775 ; q[11] = -6.779106743 ;  q[12]
= 0 ; q[13] = 0 ; q[14] = -1.840295 ;     q[15] = 2.445147 ;      q[16]
= 0.322333321 ;
break;
```

```
case 6:

q[0] = 0 ;  q[1] = -5.255909252 ;   q[2] = 11.3440103716 ;  q[3]   =   -
35.319868276 ;    q[4] = 37.23324466 ;    q[5] = 3.3850754618 ;   q[6]
= 7.7808829292 ;  q[7] = 12.4987486904 ;  q[8] = -3.1146871314 ;  q[9]
= 4.7382237064 ;  q[10] = -48.536946016 ; q[11] = -4.0895048896 ; q[12]
= 0 ; q[13] = 0 ; q[14] = -1.840295 ;     q[15] = 2.445147 ;      q[16]
= 0.322333321 ;
break;

case 7:

q[0] = 0 ;  q[1] = -5.074790664 ;   q[2] = 13.0521262912 ;  q[3]   =   -
38.031606752 ;    q[4] = 38.398689495 ;   q[5] = 3.0274191356 ;   q[6]
= 8.9769516219 ;  q[7] = 13.2391980028 ;  q[8] = -2.5314256733 ;  q[9]
= 6.4440731123 ;  q[10] = -48.790128257 ; q[11] = -1.3999030362 ; q[12]
= 0 ; q[13] = 0 ; q[14] = -1.840295 ;     q[15] = 2.445147 ;      q[16]
= 0.322333321 ;
break;

case 8:

q[0] = 0 ;  q[1] = -4.893672076 ;   q[2] = 14.7602422108 ;  q[3]   =   -
40.743345228 ;    q[4] = 39.56413433 ;    q[5] = 2.6697628094 ;   q[6]
= 10.1730203146 ; q[7] = 13.9796473152 ;  q[8] = -1.9481642152 ;  q[9]
= 8.1499225182 ;  q[10] = -49.043310498 ; q[11] = 1.2896988172 ;  q[12]
= 0 ; q[13] = 0 ; q[14] = -1.840295 ;     q[15] = 2.445147 ;      q[16]
= 0.322333321 ;
break;

case 9:

q[0] = 0 ;  q[1] = -4.712553488 ;   q[2] = 16.4683581304 ;  q[3]   =   -
43.455083704 ;    q[4] = 40.729579165 ;   q[5] = 2.3121064832 ;   q[6]
= 11.3690890073 ; q[7] = 14.7200966276 ;  q[8] = -1.3649027571 ;  q[9]
= 9.8557719241 ;  q[10] = -49.296492739 ; q[11] = 3.9793006706 ;  q[12]
= 0 ; q[13] = 0 ; q[14] = -1.840295 ;     q[15] = 2.445147 ;      q[16]
= 0.322333321 ;
break;

case 10:

q[0] = 0 ;  q[1] = -4.5314349 ;     q[2] = 18.17647405 ;    q[3]   =   -
46.16682218 ;     q[4] = 41.895024 ;      q[5] = 1.954450157 ;    q[6]
= 12.5651577 ;    q[7] = 15.46054594 ;    q[8] = -0.781641299 ;   q[9]
= 11.56162133 ;   q[10] = -49.54967498 ;  q[11] = 6.668902524 ;   q[12]
= 0 ; q[13] = 0 ; q[14] = -1.840295 ;     q[15] = 2.445147 ;      q[16]
= 0.322333321 ;
break;

case 11:

q[0] = 0 ;  q[1] = -4.350316312 ;   q[2] = 19.8845899696 ;  q[3]   =   -
48.878560656 ;    q[4] = 43.060468835 ;   q[5] = 1.5967938308 ;   q[6]
```

```
= 13.7612263927 ; q[7] = 16.2009952524 ;  q[8]   =   -0.1983798409   ;
      q[9] = 13.2674707359 ;  q[10] = -49.802857221 ; q[11]            =
9.3585043774 ;    q[12] = 0 ; q[13] = 0 ; q[14] = -1.840295 ;     q[15]
= 2.445147 ;      q[16] = 0.322333321 ;
break;

case 12:

q[0] = 0 ;  q[1] = -4.169197724 ;   q[2] = 21.5927058892 ;  q[3]   =   -
51.590299132 ;    q[4] = 44.22591367 ;    q[5] = 1.2391375046 ;   q[6]
= 14.9572950854 ; q[7] = 16.9414445648 ;  q[8] = 0.3848816172 ;   q[9]
= 14.9733201418 ; q[10] = -50.056039462 ; q[11] = 12.0481062308 ; q[12]
= 0 ; q[13] = 0 ; q[14] = -1.840295 ;     q[15] = 2.445147 ;      q[16]
= 0.322333321 ;
break;

case 13:

q[0] = 0 ;  q[1] = -3.988079136 ;   q[2] = 23.3008218088 ;  q[3]   =   -
54.302037608 ;    q[4] = 45.391358505 ;   q[5]   =  0.881481178400001   ;
      q[6] = 16.1533637781 ;  q[7] = 17.6818938772 ;  q[8]             =
0.9681430753 ;    q[9] = 16.6791695477 ;  q[10] = -50.309221703 ; q[11]
= 14.7377080842 ; q[12] = 0 ; q[13] = 0 ; q[14] = -1.840295 ;     q[15]
= 2.445147 ;      q[16] = 0.322333321 ;
break;

case 14:

q[0] = 0 ;  q[1] = -3.806960548 ;   q[2] = 25.0089377284 ;  q[3]   =   -
57.013776084 ;    q[4] = 46.55680334 ;    q[5]   =  0.523824852200001   ;
      q[6] = 17.3494324708 ;  q[7] = 18.4223431896 ;  q[8]             =
1.5514045334 ;    q[9] = 18.3850189536 ;  q[10] = -50.562403944 ; q[11]
= 17.4273099376 ; q[12] = 0 ; q[13] = 0 ; q[14] = -1.840295 ;     q[15]
= 2.445147 ;      q[16] = 0.322333321 ;
break;

}
mkdir("InitialStanceResults");
string aDir="InitialStanceResults";

std::string dirstring = stringify(direction2);

      for(i=0;i<coordinateSet->getSize();i++) {
            //// cout << coordinateSet->get(i)->getName() << " = " <<
q[i] << endl; // print out name & angles
      }

      engine.convertDegreesToRadians(q,q);

      engine.setConfiguration(q,u);

      engine.getSystemInertia(&M,COM,ICM);
      trunkAng = (q[2]+q[3]+q[4]+q[14])*SimTK_RADIAN_TO_DEGREE;
      printf("\n\nM = %lf\n",M);
```

```
        printf("COM = %lf %lf %lf\n",COM[0],COM[1],COM[2]);
        printf("trunk angle = %lf\n\n",trunkAng);
        initialHeight = COM[1];

        // GRAVITY
        SimTK::Vec3 g(0, -9.80665, 0);
        SimTK::Vec3 g0(0);
        _Model->getGravity(g);
        printf("\ngravity = %lf %lf %lf\n\n\n",g[0],g[1],g[2]);

        // INITIAL GUESS

        ////activation for direction 1,2,...etc.
        //double activation[92];

double activationx[92];

        for (i=0;i<na;i++)      {
              //x[i+ndofs]=activationx[i];   //manually   set   x's   from
abstract
              x[i+ndofs]=.5;
        }
        //// cout << "Activation (x): " << x << endl;
        target->ExtractXs(q,x);

        // MAX UPPER AND LOWER
        //   1)   subtalar;   2)   ankle;   3)   knee;   4)   hip_flexion;   5)
hip_adduction; 6) hip_rotation; 7) lumbar_extension; 8) lumbar_bending
        double subtalarLower = -15.0 * SimTK_DEGREE_TO_RADIAN;
        double subtalarUpper = 15.0 * SimTK_DEGREE_TO_RADIAN;
        double ankleLower = -5.0 * SimTK_DEGREE_TO_RADIAN;
        double ankleUpper = 30.0 * SimTK_DEGREE_TO_RADIAN;
        double kneeLower = -95.0 * SimTK_DEGREE_TO_RADIAN;
        double kneeUpper = 5.0 * SimTK_DEGREE_TO_RADIAN;
        double hipFlexionLower = -5.0 * SimTK_DEGREE_TO_RADIAN;
        double hipFlexionUpper = 95.0 * SimTK_DEGREE_TO_RADIAN;
        double hipAdductionLower = -12.5 * SimTK_DEGREE_TO_RADIAN;
        double hipAdductionUpper = 2.5 * SimTK_DEGREE_TO_RADIAN;
        double hipRotationLower = -15.0 * SimTK_DEGREE_TO_RADIAN;
        double hipRotationUpper = 15.0 * SimTK_DEGREE_TO_RADIAN;
        double lumbarExtensionLower = -45.0 * SimTK_DEGREE_TO_RADIAN;
        double lumbarExtensionUpper = 5.0 * SimTK_DEGREE_TO_RADIAN;
        double lumbarBendingLower = -5.0 * SimTK_DEGREE_TO_RADIAN;
        double lumbarBendingUpper = 5.0 * SimTK_DEGREE_TO_RADIAN;

        // Practically fix the ankle, knee, and hip angles
        double tightBounds = 1e-6 * SimTK_DEGREE_TO_RADIAN;

        // LOOP OVER CONFIGURATIONS
        double status;
        char outName[MAXLEN];
        int nAction = na + 1;
        double *outControl = new double[nAction];
        double *outReaction = new double[13];
```

```
double dHeight= 0;
double dBound = 1e-6*SimTK_DEGREE_TO_RADIAN;
double comHeight;
int loop;
double time = 0.0;

// Print out key variables
cout << "dHeight: " << dHeight << endl;
cout << "dBound: " << dBound << endl;
cout << "ConvgTol: " << ConvgTol << endl;
cout << "DIRECTION: " << direction2 << endl;
cout << "Interpolation#: " << interp << endl;
for(loop=0,comHeight=initialHeight;loop<1;comHeight+=dHeight) {
      target->setDirection(direction2);
      time += 1.0;

      //// LOOPING
      if(comHeight<initialHeight) comHeight=initialHeight;
      if(comHeight>finalHeight) comHeight=finalHeight;
      if(dHeight>0.0) {
            if(comHeight>=(finalHeight)) {
                  dHeight *= -1.0;
                  loop++;
            }
      } else {
            if(comHeight<=(initialHeight)) {
                  dHeight *= -1.0;
                  loop++;
            }
      }

      // SET DESIRED COM HEIGHT
      target->setHeight(comHeight);

      target->ExtractQs(x,q);
      // BOUNDS

// subtalar
lower[0] = q[1]-dBound;
upper[0] = q[1]+dBound;
// ankle
lower[1] = q[2]-dBound;
upper[1] = q[2]+dBound;
// knee
lower[2] = q[3]-dBound;
upper[2] = q[3]+dBound;
// hip_flexion
lower[3] = q[4]-dBound;
upper[3] = q[4]+dBound;
// hip_adduction
lower[4] = q[5]-dBound;
upper[4] = q[5]+dBound;
// hip_rotation
lower[5] = q[6]-dBound;
```

```
	upper[5] = q[6]+dBound;
	// lumbar_extension
	lower[6] = q[14]-dBound;
	upper[6] = q[14]+dBound;
	// lumbar_bending
	lower[7] = q[15]-dBound;
	upper[7] = q[15]+dBound;

		lower[8] = q[12]-dBound;
		upper[8] = q[12]+dBound;
		lower[9] = q[11]-dBound;
		upper[9] = q[11]+dBound;
		lower[10] = q[10]-dBound;
		upper[10] = q[10]+dBound;
		lower[11] = q[7]-dBound;
		upper[11] = q[7]+dBound;
		lower[12] = q[8]-dBound;
		upper[12] = q[8]+dBound;
		lower[13] = q[9]-dBound;
		upper[13] = q[9]+dBound;

	//// SET MUSCLE FORCES BOUNDS
		target->setParameterLimits(lower,upper);

		//--------------------
		// OPTIMIZE
		string good="good";
		try {
			status = optimizer->optimize(x);
		}
		catch (const SimTK::Exception::Base &ex) {
				good="fail";
				cout << ex.getMessage() << endl;
				cout  <<  "****OPTIMIZATION  FAILED...******"  <<
endl;
				cout << endl;
				cout << endl;

		}
		//--------------------

		printf("\n-----\n");
		cout << "RESULTS FOR CENTER OF MASS HEIGHT: " << comHeight
<< endl;

		// RECORD GENERALIZED COORDINATES
		target->ExtractQs(x,q);

		for(i=0;i<nu;i++) {
			qang[i] = q[i]*SimTK_RADIAN_TO_DEGREE;
		}

		// RECORD MUSCLE CONTROLS
		for(a=0;a<na;a++) {
```

```
                outControl[a] =0.0;
                //outControl[a+nmus] =0.0;
                if(a<nmus) {
                        outControl[a] = x[ndofs+a];
                        //outControl[a+nmus] = x[ndofs+a];
                }
        }
        outControl[nAction-1] = -g[1] * massTotal;

        //// CENTER OF PRESSURE


        // RECORD OPTIMAL REACTION FORCES
        massTotal                       =                       target-
>computeGroundReactions(x,acc,frc,trq,true);


        _Model->getGravity(g);
        for(i=0;i<3;i++) fg[i] = massTotal*g[i];
        for(i=0;i<3;i++) {
                outReaction[i] = fg[i];
                outReaction[3+i] = 0.0;
                outReaction[6+i] = frc[i]; // m*a[i] - fg[i];
                outReaction[9+i] = trq[i];
        }
        //printf("Total mass = %lf\n",massTotal);

        // COMPUTE COM
        engine.setConfiguration(q,u);
        engine.getSystemInertia(&M,COM,ICM);


        // APPEND TO STORAGE
        double frcScaleFactor = 1.0;
        qang[nu-3]=COM[0];qang[nu-2]=COM[1];qang[nu-1]=COM[2];
        for(i=0;i<nu;i++) qAngAndForce[i]=qang[i];
        for(i=0;i<3;i++) qAngAndForce[nu+i]=frc[i]*frcScaleFactor;
        for(i=0;i<3;i++) qAngAndForce[nu+3+i]=COM[i];
        for(i=0;i<na;i++) qAngAndForce[nu+3+3+i]=outControl[i];
        poseStore->append(time,nu+6+nAction-1,qAngAndForce);
        controlStore->append(COM[1],nAction,outControl);
        reactionStore->append(COM[1],12,outReaction);

        // WRITE OUTPUT

        std::string dH = stringify(dHeight);
        std::string CTol = stringify(ConvgTol);
        std::string dB = stringify(dBound*SimTK_RADIAN_TO_DEGREE);
        std::string interpstring = stringify(interp);

stringstream aBaseNamestream;
        aBaseNamestream << good << "_" << aDir <<"_" << dirstring
<< "_" << interpstring << "_" << dH << "dH_" << CTol << "Tol_" << dB <<
"Bound_humanOptimalPose_";
```

```
string aBaseName=aBaseNamestream.str();

Storage::printResult(poseStore,aBaseName+"q",aDir,0,".mot");

Storage::printResult(controlStore,aBaseName+"Control",aDir,0,".xls");

Storage::printResult(reactionStore,aBaseName+"reaction",aDir,0,".xls");

      }

      // CLEANUP
      printf("\n\ndone.\n\n");

      //-----------------------------
      // Catch any thrown exceptions
      //-----------------------------
      } catch(Exception x) {
            x.print(cout);
            return(-1);
      }
      //-----------------------------

      QueryPerformanceCounter(&stop);
      double         duration1         =         (double)(stop.QuadPart-
start.QuadPart)/(double)frequency.QuadPart;
      cout << "Total time = " << (duration1) << " seconds" << endl;

      //return(0);
}
}
return(0);
}

//_____________________________________________________________________
_______
/**
 * Generating a default pose file.
 */
Storage* generateHumanOptimalPoseFile()
{
      cout << "Generating default optimal pose file..." << endl;
      printf("Generating default optimal pose file... ");
      printf("configuration...\n");
      Storage *poseStore = new Storage(100,"optimalpose");

      // DESCRIPTION

      strcpy(tmp,"datarows 1000\n");
      strcat(tmp,"datacolumns 119\n");
      strcat(tmp,"otherdata 1\n");
      strcat(tmp,"range 0.000000 0.000000\n");
      poseStore->setDescription(tmp);

      // COLUMN LABELS
```

```
	int i;
	CoordinateSet *coordinateSet = _Model->getDynamicsEngine().getCoordinateSet();
	int nq = coordinateSet->getSize();
	Array<string> labels("",nq+1);
	labels.append("time");
	for(i=0;i<nq;i++) {
		labels.append(coordinateSet->get(i)->getName());
	}
	labels.append("ground_force_vx");
	labels.append("ground_force_vy");
	labels.append("ground_force_vz");
	labels.append("ground_force_px");
	labels.append("ground_force_py");
	labels.append("ground_force_pz");

	ActuatorSet *actSet = _Model->getActuatorSet();
	int na = actSet->getSize();
	for(i=0;i<na;i++) {
		string label = actSet->get(i)->getName();
		label += ".activation";
		labels.append(label);
	}

	poseStore->setColumnLabels(labels);

	return(poseStore);
}

//_____________________________________________________________________________
/**
 * Get the description of the contents of the reaction output file.
 */
char* getControlDescription()
{
	strcpy(tmp,"\nThis file contains the controls applied by the muscles ");
	strcat(tmp,"and gravity to the Human model.\n");
	strcat(tmp,"\nUnits are S.I. Units (meters, kg, Newtons, ...).\n\n");

	return(tmp);
}
//_____________________________________________________________________________
/**
 * Get the description of the contents of the reaction output file.
 */
char* getReactionDescription()
{
	strcpy(tmp,"\nGround reaction generated in response ");
	strcat(tmp,"to forces applied by muscles and gravity to the Human model.\n");
	strcat(tmp,"The 1st 6 values are the reactions cause by gravity.\n");
```

```
	strcat(tmp,"The 2nd 6 values are the reactions cause by
everything else.\n");
	strcat(tmp,"\nUnits are S.I. Units (meters, kg, Newtons,
...).\n\n");

	return(tmp);
}
//_____________________________________________________________________
/**
 * Get the columns labels for the force output file.
 */
Array<string> getControlColumnLabels(const string &aTag)
{
	ActuatorSet *actSet = _Model->getActuatorSet();
	int na = actSet->getSize();
	Array<string> labels;
	labels.append("Pose#");
	for(int i=0;i<na;i++) {
		string label = actSet->get(i)->getName();
		if(aTag!="") {
			label += ".";
			label += aTag;
		}
		labels.append(label);
	}
	labels.append("Gravity");

	return(labels);
}//_____________________________________________________________________
/**
 * Get the columns lables for the reaction output file.
 */
Array<string> getReactionColumnLabels(const string &aTag)
{
	Array<string> labels;

	// POSE
	labels.append("Pose#");

	// GRAVITY
	labels.append("g_fx");
	labels.append("g_fy");
	labels.append("g_fz");
	labels.append("g_mx");
	labels.append("g_my");
	labels.append("g_mz");

	// GRAVITY + MUSCLES
	labels.append("m_fx");
	labels.append("m_fy");
	labels.append("m_fz");
	labels.append("m_mx");
	labels.append("m_my");
	labels.append("m_mz");
```

```
        return(tmp);
        cout << "end" << endl;
}

//}
```

10 VITA

Hoa Xuan Hoang was born in 1985 in Gia Kiem, Viet-Nam. His parents are Long Kim Hoang and Hoa Thi Pham. He has an older brother, Duy, and three younger siblings: Van, Binh, and Anna. In 1991, his family moved to Knoxville, TN where he graduated in 2004 from Knoxville Catholic High School in the top 10% of his class. He enrolled at the University of Tennessee where he received the Chancellor's Scholarship and admitted in the Chancellor's Honors Program. While at the University of Tennessee, Hoa was inducted into Pi Tau Sigma, an international Mechanical Engineering Honors Society, and Tau Beta Pi, a national engineering honor society. Hoa participated in the Engineering co-op program and worked 4 rotations at Kimberly Clark. In 2008, Hoa graduated with honors from the Chancellor's Honor Program with a Bachelor of Science degree in Mechanical Engineering. In 2009, Hoa began his graduate studies in Biomedical Engineering and was appointed a graduate research assistantship in the Reinbolt Research Group at the University of Tennessee. He plans to continue his graduate education and research activities through the pursuit of a Doctor of Philosophy in Engineering.