

Dynamic Simulation and Neuromuscular Control of Movement: Applications for Predictive Simulations of Balance Recovery

A Dissertation Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Misagh Mansouri Boroujeni
May 2015

Copyright© 2015

Misagh Mansouri Boroujeni
All rights reserved

DEDICATION

This dissertation is dedicated

*To my parents,
Saeid and Mina*

*To my sisters,
Mehrak and Maral*

*And to my wife,
Amanda*

who have always stood beside me, no matter what.

ACKNOWLEDGEMENTS

I would like to sincerely thank my advisor, mentor, and my academic father Dr. Jeffrey A. Reinbolt. You went above and beyond anything I could have expected from an advisor. My life has been genuinely influenced by your confidence and trust in me. Thank you for giving me the opportunity to grow.

I would like to extend my gratitude to my committee members Dr. Songning Zhang, Dr. William Hamel and Dr. Eric Wade for their knowledge, contribution, and their time in guiding me toward graduation.

To all my fellow colleagues at RRG. To Hoa Hoang, Ashley Clark and Kristin Morgan, you really made me feel like I am not thousands of miles away from home.

To my friend and colleague, Dr. Cyril Jon Donnelly, who this project was not possible without his experimental balance data and to Dr. Jos Vanrenterghem and Dr. Mark Robinson, for allocation of their time and resources for the project experiments.

To Dr. Vincent DeSapio and Dr. Behzad Dariush for the helpful conversations and inputs on merging robotics and control methodologies into biomechanics.

To my loving wife, Amanda, who has always stood beside me, every step of the way, and never lost faith in me. Thank you for all the encouragements you gave me. You made this much easier for me. I am so grateful for having you.

Finally, I would like to thank my parents Mina and Saeid, for their unconditional love, encouragement, and support during these years. I am extremely grateful that you all are in my life.

ABSTRACT

Balance is among the most challenging tasks for patients with movement disorders. Study and treatment of these disorders could greatly benefit from combined software tools that offer better insights into neuromuscular biomechanics, and predictive capabilities for optimal surgical and rehabilitation treatment planning. A platform was created to combine musculoskeletal modeling, closed-loop forward dynamic simulation, optimization techniques, and neuromuscular control system design. Spinal (stretch-reflex) and supraspinal (operational space task-based) controllers were developed to test simulation-based hypotheses related to balance recovery and movement control. A corrective procedure (rectus femoris transfer surgery) was targeted for children experiencing stiff-knee gait and how this procedure may affect their balance recovery. Clinical movement analysis and simulation-based approaches were combined to understand the biomechanical consequences of this surgical procedure. The closed-loop controller was extended by merging approaches from robotics and biomechanics. A prioritized multi-task, support-consistent, task-based controller was implemented inside the simulation platform to synthesize human balance. The simulated results were validated with experimental data of healthy adults by defining surrogate response surfaces that represent the patients' primary tasks (e.g., to keep their balance) as function of defined subtasks (e.g. swing leg positions or torso orientations). The potential of using this platform to study, predict functional outcomes and perhaps improve treatments for musculoskeletal conditions is exciting and valuable. This project not only integrates software tools, but also allows integration of neuroscientists, physiologists, biomechanists, and physical therapists to adopt, adapt, and generate new solutions for musculoskeletal conditions.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION	1
1.1. Motivation	1
1.2. Simulation Paradigm	2
1.3. Related Work.....	4
1.4. Specific Aims	6
 CHAPTER 2 A PLATFORM FOR DYNAMIC SIMULATION AND CONTROL OF MOVEMENT BASED ON OPENSIM AND MATLAB	 12
2.1. Introduction	12
2.2. Methods.....	14
2.2.1. <i>Forward Dynamics Simulation of a Musculoskeletal Model</i>	16
2.2.2. <i>Interface between OpenSim and MATLAB/Simulink</i>	17
2.2.3. <i>Open-loop Model Application</i>	18
2.2.4. <i>Closed-loop Model Application</i>	21
2.3. Results	22
2.4. Discussion	24
 CHAPTER 3 SIMULATION AND CONTROL PIPELINE FOR PREDICTIVE PATIENT-SPECIFIC NEUROMUSCULAR SYSTEMS	 30
3.1. Create Musculoskeletal Models	31
3.2. Scale Generic Models using Subjects-Specific Data	32
3.3. Inverse Kinematic Analyses	34
3.4. Forward Dynamic Analyses	34
3.5. Controller Design to Perform Predictive Dynamic Simulations.....	35
3.5.1. <i>Low-level Spinal Stretch-reflex Controller</i>	36
3.5.2. <i>High-level Controller</i>	39
 CHAPTER 4 INFLUENCE OF THE BIARTICULAR RECTUS FEMORIS MUSCLE ON BALANCE RECOVERY IN CHILDREN WITH CEREBRAL PALSY	 41
4.1. Introduction	41
4.2. Methods.....	43
4.2.1. <i>Musculoskeletal Models</i>	43
4.2.2. <i>Biologically-inspired Controller</i>	45
4.2.3. <i>Forward Dynamics Simulations</i>	46
4.2.4. <i>Dynamic Stability Margins</i>	47

4.3.	Results	49
4.4.	Discussion	54
CHAPTER 5 TASK-BASED CONTROL OF HUMAN MOVEMENT.....		60
5.1.	Introduction	60
5.2.	Musculoskeletal Dynamics	61
5.3.	Task-based Control Formulation.....	61
5.4.	Task-based Control of Movement (Arm-Pole Control Example).....	64
CHAPTER 6 PRIORITIZED TASK-BASED CONTROL OF MOVEMENT WITH SUPPORTING CONTACTS USING OPENSIM AND MATLAB		68
6.1.	Musculoskeletal Model	69
6.2.	Free-floating Dynamic Formulation in Operational Space	70
6.2.1.	<i>Supporting Contact Dynamic Formulation in Operational Space</i>	71
6.2.2.	<i>Kinematic Constraints in Operational Space</i>	73
6.2.3.	<i>Operational Space Control of CoG</i>	74
6.3.	Tasks prioritization in Operational Space	76
6.4.	Task-based Control of Movement (Whole-body Control Example).....	78
CHAPTER 7 SYNTHESIS OF SUBJECT-SPECIFIC TASK-LEVEL MOTIONS FOR PREDICTIVE SIMULATIONS OF BALANCE RECOVERY		83
7.1.	Experimental Balance Recovery Setup	84
7.2.	Processing and Analyzing Experimental Balance Recovery Data.....	86
7.3.	Simulation Pipeline for Synthesizing Subject-specific Movement.....	87
7.4.	Results	93
7.5.	Discussion & Conclusion	94
CHAPTER 8 CONCLUDING REMARKS.....		112
8.1.	Research Innovation	112
8.2.	Summary	113
8.3.	Source Code	116
REFERENCES.....		117
VITA.....		127

LIST OF FIGURES

FIGURE 1. 1: DIAGRAM OF THE INDIVIDUAL ELEMENTS BETWEEN NEURAL COMMAND AND MOVEMENT. REHABILITATION STRATEGIES ALTER MANY ELEMENTS AND THE EFFECTS ARE NOT EASILY MEASURED.	2
FIGURE 2. 1: FLOWCHART ILLUSTRATING COMPONENTS OF FORWARD DYNAMICS IN OPENSIM AND MATLAB/SIMULINK. SIMULINK (LARGE BLUE RECTANGLE) IS USED INSTEAD OF THE OPENSIM FORWARD DYNAMICS TOOL. THE OPENSIM TRANSFORMATIONS (ORANGE RECTANGLES) BETWEEN A NEURAL COMMAND AND STATE DERIVATIVES (E.G., JOINT ACCELERATIONS) INVOLVE MUSCULOTENDON DYNAMICS, MUSCULOSKELETAL GEOMETRY, AND MULTIBODY DYNAMICS. INPUT FILES (GREEN RECTANGLES) ARE REQUIRED BY OPENSIM FOR THE MODEL, CONTROLS (IF APPLICABLE), INITIAL STATES, AND EXTERNAL LOADS (IF APPLICABLE). THE SIMULINK INPUT SIGNAL (IF APPLICABLE) AND MATLAB TRANSFORMATION (RED ROUNDED RECTANGLES) BETWEEN JOINT ACCELERATIONS AND OBSERVED MOVEMENT INVOLVES NUMERICAL INTEGRATION. THE INTERFACE CAN USE MATLAB INTEGRATORS INSTEAD OF OPENSIM INTEGRATORS FOR THE FORWARD DYNAMICS. THE NEW S-FUNCTION INTERFACE (RED DOTTED RECTANGLE) TAKES A CONTROLS FILE OR SIMULINK SIGNAL AND COMPUTES STATE DERIVATIVES SUBSEQUENTLY INTEGRATED BY MATLAB.....	15
FIGURE 2. 2: OVERVIEW OF THE NEW S-FUNCTION INTERFACE. (A) THE INTERFACE LINKS THE RAPID MODEL-BASED DESIGN AND CONTROL SYSTEMS STRENGTHS OF MATLAB/SIMULINK WITH THE NUMERICAL SIMULATION AND HUMAN MOVEMENT DYNAMICS STRENGTHS OF OPENSIM. (B) FLOWCHART ILLUSTRATING STAGES FOR EACH SIMULINK SIMULATION. (C) OPENSIM v1.9 (HIGHER VERSIONS NOT SHOWN) METHODS CALLED DURING EACH SIMULATION STAGE.	19
FIGURE 2. 3: OVERVIEW OF AN OPEN-LOOP MODEL APPLICATION FOR ARM FLEXION. (A) THE SIMULINK GRAPHICAL EDITOR WINDOW SHOWS INPUT CONTROLS (U), THE NEW S-FUNCTION INTERFACE (MIDDLE), AND THE OUTPUT STATES (Y). (B) THE CUSTOM SIMULINK BLOCK PARAMETER DIALOG BOX SHOWS EVERY INPUT PARAMETER REQUIRED (OR OPTIONAL) BY THE OPENSIM FORWARD DYNAMICS TOOL TO PERFORM A SIMULATION. (C) THE OUTPUT STATES MAY BE DISPLAYED USING A STATE SELECTOR AND SCOPE WITHIN THE GRAPHICAL EDITOR OR THEY MAY BE LOADED AS A MOTION FILE IN OPENSIM TO VISUALIZE THE SIMULATED MOVEMENT.	20
FIGURE 2. 4: EXAMPLE CLOSED-LOOP SIMULINK MODEL EXTENDING THE OPEN-LOOP CASE WITH CONTROL OF A HUMAN ARM BALANCING A POLE. THE DESIRED POLE ANGLE (MEASURED FROM VERTICAL) WAS ZERO AND USED TO COMPUTE A POLE ANGLE ERROR. THE PROPORTIONAL-INTEGRAL-DERIVATIVE (PID) CONTROLLER, ALONG WITH THE MUSCLE GAIN MATRIX WAS USED TO COMPUTE CONTROL CORRECTION SIGNALS FOR EACH OF THE SIX MUSCLES TO BALANCE THE POLE, DESPITE RANDOM FORCE DISTURBANCES EXERTED ON THE POLE.....	23
FIGURE 2. 5: OPEN-LOOP SIMULATION OF ELBOW FLEXION AND DIFFERENCES BETWEEN MOVEMENTS GENERATED BY OPENSIM AND SIMULINK SIMULATIONS. (A) A SIMPLE HUMAN ARM MODEL WITH 2 DEGREES OF FREEDOM AND 6 MUSCLE-TENDON ACTUATORS (SIX TIME FRAME SERIES OF A ONE SECOND ELBOW FLEXION MOVEMENT SHOWN). (B) SHOULDER ANGLES AND (C) ELBOW ANGLES FOR THE OPENSIM AND SIMULINK SIMULATIONS.	24
FIGURE 2. 6: CLOSED-LOOP SIMULATION OF BALANCING A POLE DESPITE RANDOM FORCE DISTURBANCES. (A) A PID CONTROLLED HUMAN ARM MODEL BALANCING A POLE (FIVE TIME FRAME SERIES FROM 0.25s TO 0.55s SHOWN). (B) RANDOM FORCE DISTURBANCES WERE EXERTED ON THE POLE AND (C) THE POLE ANGLE MEASURED FROM VERTICAL REMAINED SMALL.	25
FIGURE 3. 1: DIFFERENT OPENSIM LAYERS. THE BASE LAYER IS THE COMPUTATIONAL LAYER PROVIDED BY SIMBODY (IN BLUE). SIMBODY COMPONENTS (E.G. INTEGRATORS, OPTIMIZERS) ARE AVAILABLE TO ALL LEVELS WITHIN OPENSIM API. THE NEXT LAYER IS MODELING LAYER (GREEN). THE MODELING LAYER IS FOLLOWED BY ANALYSES LAYER (ORANGE). THE TOP LAYER IS THE APPLICATION LAYER (RED) WHICH IS CONSUMER OF OPENSIM API TO GENERATE APPLICATIONS SUCH AS OPENSIM GUI OR DYNAMIC LIBRARIES. (IMAGE COURTESY: SETH ET AL., 2011).	32
FIGURE 3. 2: SCALING A GENERIC MUSCULOSKELETAL MODEL BASED ON THE EXPERIMENTAL PATIENT-SPECIFIC DATA. MODELS ARE SCALED TO REPRESENT THE LENGTH AND MASS OF EACH SEGMENT OF THE EXPERIMENTAL DATA.	33

FIGURE 3. 3: OVER SIMPLIFIED MODEL OF HUMAN BALANCE CONTROL USING A SIMPLE INVERTED PENDULUM WITH TWO MUSCLE-TENDON ACTUATORS WAS USED TO VALIDATE A MORE COMPLEX CONTROL PROBLEM SUCH AS HUMAN POSTURE BALANCE WITH 23 DOF AND 92 MUSCLE-TENDON ACTUATORS.....	37
FIGURE 3. 4: MUSCLES' ACTIVATIONS DURING A ONE SECOND FORWARD DYNAMIC SIMULATION WITH STRETCH-REFLEX CONTROLLER DUE TO THE INITIAL PENDULUM VELOCITY AS A PERTURBATION TO THE SYSTEM.	38
FIGURE 3. 5: THE INVERTED PENDULUM ROTATION ANGLES (Φ) AND ROTATION VELOCITY DURING A ONE SECOND FORWARD DYNAMIC SIMULATION WITH STRETCH-REFLEX CONTROLLER DUE TO THE INITIAL PENDULUM VELOCITY AS A PERTURBATION TO THE SYSTEM.	38
FIGURE 3. 6: MUSCLES' FIBER LENGTH DURING A ONE SECOND FORWARD DYNAMIC SIMULATION WITH STRETCH-REFLEX CONTROLLER DUE TO THE INITIAL PENDULUM VELOCITY AS A PERTURBATION TO THE SYSTEM.	39
FIGURE 4. 1: THREE-DIMENSIONAL, 23 DEGREE-OF-FREEDOM MUSCULOSKELETAL MODEL WITH 92 MUSCLE-TENDON ACTUATORS (SHOWN IN RED) RECOVERING BALANCE ON A SUPPORT SURFACE TRANSLATING 7.5 CM OVER 0.55 SECONDS WITH A MAXIMUM VELOCITY OF 18 CM/S. THE MUSCULOSKELETAL MODEL WAS USED ALONG WITH A BIOLOGICALLY-INSPIRED CONTROLLER TO CREATE FORWARD DYNAMIC SIMULATIONS OF CHILDREN WITH MILD AND MODERATE CROUCH TO TEST OUR HYPOTHESIS REGARDING BALANCE RECOVERY FOLLOWING RECTUS FEMORIS TRANSFER SURGERY. BIARTICULAR LOCATIONS FOR THE RECTUS FEMORIS MUSCLE ARE SHOWN AS (A) THE PRE-SURGICAL ATTACHMENT TO PATELLA AND (B) THE POST-SURGICAL TRANSFER TO THE INSERTION OF THE SARTORIUS ON THE TIBIA.	44
FIGURE 4. 2: SCHEMATIC OF (A) THE CLOSED-LOOP FORWARD DYNAMIC SIMULATION IN SIMULINK AND (B) THE UNDERLYING ALGORITHMS. THE INPUTS TO THE MUSCULOSKELETAL MODEL, OR NEURAL COMMAND, WERE CALCULATED BASED ON THE LOW-LEVEL (STRETCH-REFLEX CONTROLLER) AND HIGH-LEVEL (COMPUTED MUSCLE CONTROL) FEEDBACK CONTROL SYSTEM. DISTURBANCES (E.G., SUPPORT-SURFACE TRANSLATIONS) MAY ALSO BE INTRODUCED TO THE SYSTEM AS INPUTS. THE OUTPUTS OF THE DYNAMIC SIMULATION ARE THE MODEL'S STATES (I.E., JOINT ANGLES AND VELOCITIES, MUSCLE ACTIVATIONS AND FIBER LENGTHS). THE STRETCH-REFLEX CONTROLLER GAINS, KP AND KD, WERE SELECTED BY A NON-LINEAR LEAST SQUARES OPTIMIZATION MINIMIZING POSITION AND VELOCITY ERRORS BETWEEN SIMULATED AND EXPERIMENTAL CENTER OF MASS DISPLACEMENTS.	48
FIGURE 4. 3: THE EXTRAPOLATED CENTER OF MASS OF A PRE-SURGICAL MODEL FOLLOWING POSTERIOR SUPPORT-SURFACE TRANSLATION AS A FUNCTION OF SIMULATION TIME. CoM_{extrp} WAS CHOSEN OVER CoM AND CENTER OF PRESSURE (CoP) IN DEFINING THE STABILITY MARGINS BECAUSE MODELS MAY BE MOVING TOWARD A FALL THAT OCCURS AFTER THE SIMULATION TIME DESPITE THE CoP BEING LOCATED WITHIN THE BoS THROUGHOUT THE SIMULATION. THE FUNCTIONS B AND TtB ARE DEFINED BASED ON HOF ET AL., 2005 AND THE MINIMUM VALUES OF THESE FUNCTIONS WERE USED AS AN INDICATOR OF HOW STABLE THE SIMULATIONS ARE AT EACH TIME STEP.	50
FIGURE 4. 4: EXTRAPOLATED CENTER OF MASS DISPLACEMENTS RELATIVE TO THE SUPPORT-SURFACE DURING ANTERIOR (A & B) AND POSTERIOR (C & D) TRANSLATIONS (SHADED REGIONS) FOR SIMULATIONS BEFORE (PRE) AND AFTER (POST, UNILATERAL AND BILATERAL) RECTUS FEMORIS TENDON TRANSFER FOR MILD (A & C) AND MODERATE (B & D) CROUCHED POSTURES. THE DISPLACEMENTS FOR SIMULATIONS OF THE TYPICALLY DEVELOPING (TD) MODEL WITHOUT A SPASTIC STRETCH-REFLEX ARE SHOWN AS WELL.	52
FIGURE 4. 5: SIMULATED CENTER OF MASS (CoM) DISPLACEMENTS FOR TYPICALLY DEVELOPING (THIN SOLID LINE) AND PRE-SURGICAL, SPASTIC CEREBRAL PALSY (THICK SOLID LINE) MODELS WITH MILD CROUCH COMPARED WITH EXPERIMENTAL CoM DISPLACEMENTS OF HEALTHY ADULTS WITH UPRIGHT POSTURE FROM TING, 2007 (SQUARE DOT LINE) AND HENRY, ET AL., 1998 (DASH LINE) DURING A) ANTERIOR AND B) POSTERIOR SUPPORT-SURFACE TRANSLATIONS. TO COMPARE THE BALANCE RECOVERIES ACROSS SUPPORT-SURFACE TRANSLATION MAGNITUDES, THE CoM DISPLACEMENTS WERE NORMALIZED BY THE SUPPORT-SURFACE TRANSLATIONS IN EACH STUDY.	56
FIGURE 5. 1: A TASK-LEVEL NEUROMUSCULAR FEEDBACK CONTROLLER (A) TASK-LEVEL COMMAND INPUT STATES. (B) TASK SPACE FEEDBACK LINEARIZATION USING ESTIMATION OF THE DYNAMIC PARAMETERS. (C) TASK SPACE FEEDBACK LOOP FOR TRACKING TASK SPACE COMMANDS. (D) POSTURE SPACE CONTROL INPUT. (E) STATIC OPTIMIZATION OF MUSCLE ACTIVATIONS. (F) ACTIVATION FEEDBACK LOOP FOR TRACKING OPTIMIZATION SOLUTION. (G) OUTPUT AS NEURAL EXCITATIONS. (IMAGE COURTESY: GOLDFARB, 2014).	64

FIGURE 5. 2: SCHEMATIC MODEL OF THE TASK-BASED NEUROMUSCULAR CONTROLLER IMPLEMENTED THROUGH THE OPENSIM/MATLAB INTERFACE. THE INTERFACE (DOTTED LINES IN A) CALLS THE OPENSIM ARM MODEL AND THEN THE CLOSED-LOOP TASK FORCE IS CALCULATED AS AN INPUT TO THE INTERFACE. THE TASK FORCE THEN WAS USED TO CALCULATE THE ACTUATION FORCE (JOINT TORQUES) USING MULTI-BODY DYNAMICS AND LATER THE MUSCLE EXCITATIONS USING STATIC OPTIMIZATION. ALL THESE STEPS ARE IMPLEMENTED INSIDE THE INTERFACE BY CALLING OPENSIM LIBRARIES.	65
FIGURE 5. 3: DISTURBANCE WAS INTRODUCED TO THE CENTER OF MASS OF THE POLE TO TEST THE CONTROLLER ROBUSTNESS IN PRESENCE OF EXTERNAL PERTURBATION.	66
FIGURE 5. 4: THE RESULTANT MUSCLE ACTIVATIONS CALCULATED THROUGH THE TASK-BASED CONTROLLER IMPLEMENTED DURING 10 SECOND SIMULATIONS OF HAND POSITION CONTROL AND POLE BALANCING TASKS. ..	67
FIGURE 6. 1: FREE-FLOATING MUSCULOSKELETAL MODEL. PELVIS BODY (FREE-FLOATING BASE) HAS 6 DEGREES OF FREEDOM IN SPACE (3 TRANSLATIONAL AND 3 ROTATIONAL).	69
FIGURE 6. 2: FOOT-GROUND CONTACT MODELED AS HUNT CROSSLY FORCES INSIDE THE OPENSIM'S MODEL FILE. FIVE CONTACT POINTS (TWO ON EACH SIDE OF THE HEEL AND TWO ON EACH SIDE OF THE METATARSAL BONES AND ONE ON THE TIP OF THE TOES) WERE DEFINED ACCORDING TO ANDERSON AND PANDY, 1999, TO REPRESENT FOOT-GROUND CONTACT.	72
FIGURE 6. 3: SCHEMATIC OF THE TASK-BASED NEUROMUSCULAR CONTROLLER IMPLEMENTED THROUGH OPENSIM/MATLAB INTERFACE IN THE SIMULINK ENVIRONMENT. THE INTERFACE (DOTTED LINES IN A) CALLS THE OPENSIM 2392 GAIT MODEL, INITIAL CONTROLS, INITIAL STATES FILES, AND PERFORM A FORWARD DYNAMICS SIMULATION WITH A CLOSED-LOOP TASK-BASED FEEDBACK CONTROL IN OPERATIONAL SPACE USING THE DEFINED TASK FORCES REQUIRED FOR PERFORMING A SEQUENCE OF TASKS. THE TASK FORCE IS USED TO CALCULATE THE ACTUATION FORCE (JOINT TORQUES) THROUGH MULTI-BODY SYSTEM JACOBIAN TRANSPOSE. MUSCLE EXCITATIONS ARE CALCULATED USING STATIC OPTIMIZATION.	77
FIGURE 6. 4: DIFFERENT TASKS DEFINED FOR THE SINGLE-LEG BALANCE EXAMPLE. ALL THESE TASKS DEFINED WITHIN THE SUPPORTING CONTACT CONSTRAINT. THREE TASK FORCE VECTORS WERE DEFINED. THE FIRST TASK WAS EXERTED ON THE WHOLE BODY CENTER OF MASS. THE SECOND TASK FORCE WAS EXERTED ON THE SWING LEG FOOT AND THE THIRD TASK FORCE WAS EXERTED ON OUR MODEL HEAD. THE ARROWS ON THE FOOT-GROUND CONTACTS ARE NOT THE TASK FORCES. THEY SHOW THE FOOT-GROUND CONTACT FORCES DUE TO GRAVITATIONAL FORCES. THE RESULTANT GROUND REACTION FORCES ARE SUMMATION OF ALL THOSE FIVE CONTACT FORCES AND ARE APPLIED TO THE CENTER OF PRESSURE OF THE MODEL.	79
FIGURE 6. 5: A) THE DESIRED WHOLE-BODY CoM TRAJECTORY DEFINED FOR THE MUSCULOSKELETAL SYSTEM THROUGH THE POINTS (1-4). THE MODEL IS SUPPOSE TO GO THROUGH THESE POINT WHILE MAINTAINING BALANCE ON ONE FOOT. B) RIGHT LEG (STANCE LEG) JOINT TORQUES REQUIRED TO PERFORM THE ASSIGNED TASK.	81
FIGURE 6. 6: THE REFERENCE WHOLE-BODY CoM PATH (X, Y AND Z), WAS DEFINED IN OUR BALANCE CONTROL EXAMPLE INSIDE SIMULINK.	82
FIGURE 7. 1: CAREN SYSTEM (MOTekMEDICAL), SINGLE-LEG SUPPORT SETUP.	85
FIGURE 7. 2: STATIC POSE FOR THE SECOND SUBJECT SHOWS PLACEMENT OF REFLECTIVE MARKERS AND THE FORCE PLATE SETUP ON THE CAREN PLATFORM.	85
FIGURE 7. 3: TASK-BASED CONTROL OF BALANCE SIMULATIONS FLOWCHART USING SUBJECT-SPECIFIC DATA.	89
FIGURE 7. 4: TASK-BASED CONTROL OF BALANCE SYNTHESIS FLOWCHART USING SUBJECT-SPECIFIC DATA.	90
FIGURE 7. 5: QUADRATIC RESPONSE SURFACES FOR SUBJECT 1'S CoM TO LEFT CALCANEUS (V_2) IN X, Y AND Z DIRECTION OVER PROJECTION OF V_1 (EXPERIMENTAL (MAGENTA) AND SIMULATED (BLACK)).	97
FIGURE 7. 6: QUADRATIC RESPONSES SURFACE FOR SUBJECT 1'S CoM TO TORSO (V_3) IN X, Y AND Z DIRECTION OVER PROJECTION OF V_1 (EXPERIMENTAL (MAGENTA) AND SIMULATED (BLACK)).	98
FIGURE 7. 7: QUADRATIC RESPONSE SURFACES FOR SUBJECT 2'S CoM TO LEFT CALCANEUS (V_2) IN X, Y AND Z DIRECTION OVER PROJECTION OF V_1 (EXPERIMENTAL (MAGENTA) AND SIMULATED (BLACK)).	99
FIGURE 7. 8: QUADRATIC RESPONSE SURFACES FOR SUBJECT 2'S CoM TO LEFT TORSO (V_3) IN X, Y AND Z DIRECTION OVER PROJECTION OF V_1 (EXPERIMENTAL (MAGENTA) AND SIMULATED (BLACK)).	100

FIGURE 7. 9: KINEMATIC DIFFERENCES BETWEEN EXPERIMENTAL (ORANGE) AND SIMULATED MOVEMENT (BLUE) FOR A) SUBJECT 1 AND B) SUBJECT 2. THIS FIGURE IS CONSTRUCTED FROM THE LAST FRAME OF THE SIMULATIONS TO QUALITATIVELY SHOW HOW WELL THE EXPERIMENTS AND SYNTHESIZED MOTION MATCH.	101
FIGURE 7. 10: SUBJECT 1’S RIGHT LEG AND LEFT LEG HIP AND KNEE JOINTS KINEMATICS (EXPERIMENTAL (DASH) AND SYNTHESIZED (SOLID)).....	102
FIGURE 7. 11: SUBJECT 1’S PELVIS ROTATIONS AND TRANSLATION JOINT KINEMATICS (EXPERIMENTAL (DASH) AND SYNTHESIZED (SOLID)).....	103
FIGURE 7. 12: SUBJECT 2’S RIGHT LEG AND LEFT LEG HIP AND KNEE JOINTS KINEMATICS (EXPERIMENTAL (DASH) AND SYNTHESIZED (SOLID)).....	104
FIGURE 7. 13: SUBJECT 2’S PELVIS ROTATIONS AND TRANSLATION JOINT KINEMATICS (EXPERIMENTAL (DASH) AND SYNTHESIZED (SOLID)).....	105
FIGURE 7. 14: SUBJECT 1’S COM DISPLACEMENT IN ANTERIOR-POSTERIOR DIRECTION FROM THE EXPERIMENT (GREEN) AND THE CORRESPONDING SIMULATION (BLUE).....	106
FIGURE 7. 15: SUBJECT 2’S COM DISPLACEMENT IN ANTERIOR-POSTERIOR DIRECTION FROM THE EXPERIMENT (GREEN) AND THE CORRESPONDING SIMULATION (BLUE).....	107
FIGURE 7. 16: EXPERIMENTAL (FIRST THREE ROWS) EMG VERSUS THE SIMULATED (LAST ROW) MUSCLE ACTIVITIES (BICEPS FEMORIS, MEDIAL GASTROCNEMIUS, GLUTEUS MAXIMUS AND GLUTEUS MEDIUS) FOR SUBJECT 1. EXPERIMENTAL RAW EMGs (FIRST row, RED), RECTIFIED EMGs (SECOND row, ORANGE), EXPERIMENTAL FILTERED EMGs (THIRD row, GREEN) AND SIMULATED MUSCLE ACTIVATIONS (LAST row, BLUE).....	108
FIGURE 7. 17: EXPERIMENTAL (FIRST THREE ROWS) EMG VERSUS THE SIMULATED (LAST ROW) MUSCLE ACTIVITIES (RECTUS FEMORIS, SEMITENDINOSUS, VASTUS LATERALIS, VASTUS MEDIALIS) FOR SUBJECT 1. EXPERIMENTAL RAW EMGs (FIRST row, RED), EXPERIMENTAL RECTIFIED EMGs (SECOND row, ORANGE), EXPERIMENTAL FILTERED EMGs (THIRD row, GREEN) AND SIMULATED MUSCLE ACTIVATIONS (LAST row, BLUE).....	109
FIGURE 7. 18: EXPERIMENTAL (FIRST THREE ROWS) EMG VERSUS THE SIMULATED (LAST ROW) MUSCLE ACTIVITIES (BICEPS FEMORIS, MEDIAL GASTROCNEMIUS, GLUTEUS MAXIMUS AND GLUTEUS MEDIUS) FOR SUBJECT 2. EXPERIMENTAL RAW EMGs (FIRST row, RED), RECTIFIED EMGs (SECOND row, ORANGE), EXPERIMENTAL FILTERED EMGs (THIRD row, GREEN) AND SIMULATED MUSCLE ACTIVATIONS (LAST row, BLUE).....	110
FIGURE 7. 19: EXPERIMENTAL (FIRST THREE ROWS) EMG VERSUS THE SIMULATED (LAST ROW) MUSCLE ACTIVITIES (RECTUS FEMORIS, SEMITENDINOSUS, VASTUS LATERALIS, VASTUS MEDIALIS) FOR SUBJECT 2. EXPERIMENTAL RAW EMGs (FIRST row, RED), EXPERIMENTAL RECTIFIED EMGs (SECOND row, ORANGE), EXPERIMENTAL FILTERED EMGs (THIRD row, GREEN) AND SIMULATED MUSCLE ACTIVATIONS (LAST row, BLUE).....	111

CHAPTER 1

INTRODUCTION

1.1. Motivation

Numerical simulations play an important role in solving complex engineering problems and have the potential to revolutionize medical decision-making and treatment strategies. Musculoskeletal conditions alone cost the U.S. economy over \$849 billion per year (7.7% of the U.S. Gross Domestic Product) and place great demands on healthcare systems worldwide [1]. Study and treatment of these conditions could greatly benefit from combined software tools that offer better insight into neuromuscular biomechanics, and predictive capabilities for optimal surgical and rehabilitation treatment planning.

In recent years, there have been many technological advances in gait-analysis technologies and devices. These technologies range from various motion capture systems that record human body positions [2-5], to force plates and pressure sensors that measure ground reaction forces [6, 7], and to different electromyography instruments [8] that detect muscle activity during various movements. These systems provide mainly a quantitative description of human movements; however, the experiments alone do not explain how muscles work together or how they individually contribute to certain motion or even how the control of these muscles can influence the resulting motion. Muscle-actuated simulations and bio-inspired control systems can potentially give insights into the complex interaction between neural commands, musculoskeletal geometry, and the resulting functional movements. Combining clinical movement analysis and simulation-based approaches may lead to a better understanding of biomechanical consequences of different human movement disorders and its treatments.

1.2. Simulation Paradigm

Forward dynamics simulations of human movement require consideration of the many individual elements involved in the control of human motion. These individual elements include neural transformation, musculoskeletal geometry and multi-joint dynamics (Figure 1.1). First, a neural command is needed to excite muscles. The electrical potentials of muscles can then be recorded with electromyography (EMG). Next, muscles produce forces based on length and velocity properties of the muscle and tendon. The muscle forces are related to joint moments through musculoskeletal geometry (e.g. muscles' moment arms, joint types and joint locations). Moments are used to determine joint accelerations and contact forces by solving the multibody dynamics equations. The resulting human movements, called states, are then derived by double integration of the joint accelerations. Finally, human movements feed back through sensory organs onto underlying neural commands that fine-tune the movement by altering the muscle length, velocity, and moment arms.

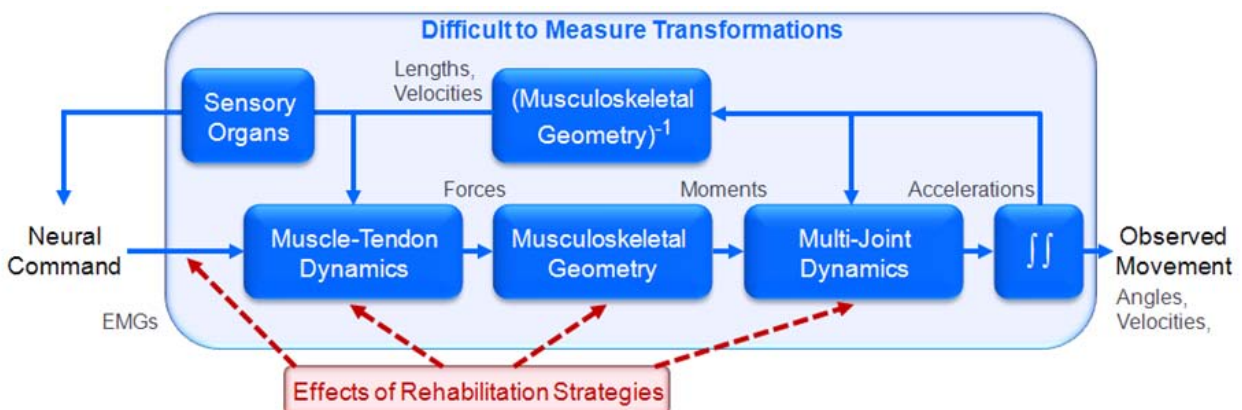


Figure 1. 1: Diagram of the individual elements between neural command and movement. Rehabilitation strategies alter many elements and the effects are not easily measured.

Many of these elements have been investigated individually through different experiments. These studies utilized a variety of experimental tools such as electrical stimulation of muscle fibers [9], cadaver studies that measured muscle moment arms [10], and the MRI imaging of live human body segments [11] to investigate the relationship between muscle excitations and human movement. However, experiments alone are limited in understanding complex human movement dynamics, and simulations can complement experiments. Although a few variables responsible for movement (e.g., kinematics, ground reaction forces, and EMG) can be measured, it is extremely difficult to measure important variables such as muscle activities and joint forces for all muscles and joints involved. It is even more difficult to establish cause-and-effect relationships that give insight into muscle function.

Similar to an observed movement, a simulated movement results from the interactions among many individual elements. Patient-specific models are used to generate the musculoskeletal geometry and multi-joint dynamics considered in a simulated movement. Combining these simulations with experimental data enable cause-and-effect relationships to be identified and can give insight into the potential treatment for an abnormality or a neurological disorder. Gait and balance abnormalities commonly observed in children with cerebral palsy are typically treated by surgically altering muscle function. Unfortunately, the outcomes of such treatments are variable. Patient-specific simulations have utility in evaluating the potential efficacy of surgical correction. Simulated movement also provides estimates of important variables involved in generating the movement. A new approach using patient-specific computational models, control system designs, and optimization may be used to discover a new movement as a treatment and to predict functional outcomes using pre-treatment data. Simulation is a vital tool for multidisciplinary study of human movement control because it can

provide fundamental insights into the causes of movement and can point the way to optimal treatment recommendations.

1.3. Related Work

Computational models and simulation-based approaches have emerged as powerful tools for investigating muscle coordination and function during human movement. Biomechanical models have been developed to estimate muscular forces in the lower extremity during balance [12, 13], walking [14-17], running [18-20], cycling [21-23], jumping [24-26], kicking [27], and other physical activities [28-32]. Sophisticated muscle-actuated, dynamic simulations have been developed to address specific clinical questions. For example, studies have been conducted to assess electrical stimulation systems to restore unsupported gait to paraplegics [33], to evaluate exercise for persons with spinal cord injury [34, 35] and for patients suffering from patella-femoral pain [36], to examine the influence of foot positioning and joint compliance on ankle sprains [37, 38], and to investigate causes of stiff-knee gait [39-41]. These studies have demonstrated the potential utility of models and computer simulations for analyzing causes of abnormalities and the effects of treatments.

Given that the ability of the model to accurately represent real-life subjects affects the validity of a simulation, a model providing the closest representation of the experimental subject produces the best variable estimations and relationship identifications. There is a rapidly growing community of engineers, therapists, and scientists eager to address clinically motivated questions on medical rehabilitation. Thus, our work here promises to significantly impact more than one field.

First-generation patient-specific models [42-50] have laid the foundation for simulations to predict patient outcomes; however, patient-specific modeling and simulation technology need

to be advanced significantly to realize the full potential of personalized, simulation-based medicine [51]. Computer simulations have made a huge impact across different science and engineering disciplines [52, 53], animations designs [54], and the movie industry. For similar impacts to be made in human movement simulation and control, multidisciplinary research teams need to be formed to merge simulations and experimental measurements with higher-level control systems and robotics concepts and methodologies [55, 56]. This synergistic relationship not only produces the best possible simulation results, but also has a potential to transform the biomechanics research field

Furthermore, simulation and control of human movement is a challenging problem that requires simultaneous knowledge of human anatomy and physiology, physics, engineering, computer simulations, robotics and control system design. While biomechanists aim to record and analyze movements to enhance athletic performance or design new rehabilitation techniques, researchers in the field of robotics try to improve robot designs and control algorithms to create human like movement. Both fields can benefit from software tools that merge these methodologies to answer complex questions regarding human movement.

In the field of robotics, there has been much research aiming to emulate human motion and create human-like robots [57-60]. Experimental human motion capture was mapped to different models using inverse dynamics approaches [61, 62], but these simulations were limited in predicting and synthesizing new motions. Inspired by human behaviors, goal-driven controllers were introduced to control task oriented dynamic problems in humanoid robots [63, 64] and synthesize human motion [65, 66]. Validating these controllers requires the use of complex physiological musculoskeletal models. Recent studies have shown successful applications of task-based controllers to create realistic human motion [66, 67]. In this research,

the task-based controller was used to control complex upper body and lower body musculoskeletal models to investigate hypotheses regarding human movement and balance control.

1.4. Specific Aims

In this research, we created a platform to combine musculoskeletal modeling, forward dynamic simulation, optimization techniques, and neuromuscular control system design to test simulation-based hypotheses related to balance recovery and movement control. Balance is among the most challenging tasks for patients with neuromuscular disorders such as children with cerebral palsy [68-70]. Using the platform, we hope to develop scientific guidelines for understanding and improving balance recovery in patients with neuromuscular disorders.

Stiff-knee gait is a troublesome neuromuscular disorder among children with cerebral palsy. Patients experiencing stiff-knee gait typically adopt energy-inefficient movements due to over-activity of lower extremity muscles and are prone to falling and tripping. Currently, corrective procedures are based on qualitative observations during clinical examination, gait analysis, and the experience of clinicians. Unfortunately, successful outcomes of such procedures are variable. Until there is a scientific basis for making treatment decisions, outcomes are not likely to improve. Combining clinical movement analysis and simulation-based approaches may lead to a better understanding of biomechanical consequences of stiff-knee gait and its treatments.

The first objective of this study was to develop a simulation and control platform to test simulation-based hypotheses related to balance recovery and movement control. The platform then was used to investigate balance recovery in children with stiff-knee gait, to explore the functional consequences of a commonly used surgical intervention, and to examine how

predictive simulations that synthesize normal healthy movement can identify the biomechanical factors that may affect postural responses. Next, the balance recovery simulations were synthesized and validated using methodologies from biomechanics, robotics, control system design, optimizations and patient-specific modeling. To accomplish these ends, the following specific aims were addressed:

Aim 1 designed a platform for dynamic simulation and control of movement based on OpenSim and MATLAB. We combined the rapid model-based design, control systems and powerful numerical methods strengths of MATLAB/Simulink with the simulation and human movement dynamics strengths of OpenSim by developing a new interface between the two software tools. OpenSim is integrated with Simulink using the MATLAB S-function mechanism, and the interface is demonstrated using both open-loop and closed-loop control systems. While the open-loop system uses MATLAB/Simulink to separately reproduce the OpenSim Forward Dynamics Tool, the closed-loop system adds the unique feature of feedback control to OpenSim, which is necessary for most human movement simulations. The new interface presented here not only integrates the OpenSim and MATLAB/Simulink software tools, but also allows neuroscientists, physiologists, biomechanists, and physical therapists to adapt and generate new solutions as treatments for musculoskeletal conditions. To achieve this aim, we:

- 1) Developed a MATLAB® S-function (system-function) based on an OpenSim model as a Simulink® block written in C++ and compiled as a MEX-file using the MATLAB® mex utility.
 - a. Began with an annotated S-function template containing skeleton implementations of all the required and optional callback methods that an S-function can implement
 - b. Updated the S-function to include OpenSim header files

- c. Implemented parameter handling methods (e.g., mdlCheckParameters verifies new parameter settings whenever parameters change or are re-evaluated during a simulation) to operate on the desired OpenSim model.
 - d. Implemented configuration and execution methods (e.g., mdlInitializeSizes, mdlStart, mdlOutputs, mdlDerivatives, and mdlTerminate) to operate on the desired OpenSim model.
 - e. Defined required S-function trailer (e.g., identifying MEX-function interface mechanism to Simulink®).
- 2) Created two Simulink® model examples that load and execute the OpenSim-based Sfunction developed in step 1.
 - a. Open loop control model.
 - b. Closed loop control model.
 - 3) Posted all C++ source code, Simulink® model examples resulting from steps 1&2 on SimTK.org within a public SimTK.org project dedicated to the interface between OpenSim and MATLAB®/Simulink®.

Aim 2 designed a simulation and control pipeline using the OpenSim/MATALB platform for predictive patient-specific neuromuscular systems.

The next step was to design a pipeline for simulation and control of patient-specific neuromuscular systems. Human movement requires the coordination of many muscles, and the transformations between neural control and purposeful movement are highly complex and involve many individual elements. Many of these individual elements have been characterized through experiments. However, considering experiments alone limits the understanding of movement dynamics to an understanding of only the individual aspects of motion. Simulations complement experiments by helping us to understand how these individual aspects work together to produce motion. Although a few variables responsible for movement such as kinematics, ground reaction forces, and electromyography be measured, it is particularly difficult to measure important variables such as individual muscle activities and actual joint forces. Predictive

modeling and simulations provide tools to investigate these limitations. A new treatment strategy combining physical examination, movement analysis, and simulation-based approaches may lead to a better understanding of human movement control. To achieve this aim, we implemented the following pipeline:

- 1) Created musculoskeletal models using OpenSim software for each specific application.
- 2) Scaled generic models based on specific subjects' anthropometric data
- 3) Performed inverse kinematics analysis to determine model kinematics that best match the experimentally measured kinematics.
- 4) Ran Forward Dynamics simulation to estimate states and generate motion due to muscle excitations.
- 5) Designed different controllers to perform predictive dynamic simulations
 - a. Low-level spinal stretch-reflex feedback controller
 - b. High-level task based controller to track kinematics

Aim 3 utilized the simulation and control pipeline to investigate the role of rectus femoris muscle in balance recovery in children with cerebral palsy. Stiff knee gait is a prevalent and troublesome movement disorder among children with cerebral palsy (CP), whereby peak knee flexion is diminished during the swing phase of gait due to over-activity of lower extremity muscles such as Rectus Femoris. Rectus femoris transfer surgery, a common treatment for stiff-knee gait, reattaches the distal tendon of this muscle from patella to the Sartorius insertion on the tibia. As a biarticular muscle, rectus femoris may play a unique role in motor control and has unrecognized benefits for maintaining balance. This study uses musculoskeletal modeling, forward dynamics simulation, optimization techniques, and neuromuscular control system design to investigate the role of the rectus femoris muscle on balance recovery following support-surface translations in children with CP following rectus femoris transfer surgery. We hypothesized that due to the unique role of rectus femoris muscle as a biarticular muscle acting

as a hip flexor and a knee extensor, pre-surgical simulations would be more stable (greater disturbance rejection ability) than post-surgical simulations. To achieve this aim, we:

- 1) Created patient specific musculoskeletal models of children with spastic CP before and after rectus femoris transfer surgery adopting either a mild or moderate crouched posture;
- 2) Designed a biologically-inspired, closed-loop control system for balance recovery
- 3) Generated forward dynamic simulations to examine balance recovery following support-surface translations
- 4) Evaluated stability margins using the extrapolated center of mass position and minimum time-to-boundary of the base of support.

Aim 4 combined techniques from biomechanics, robotics, control system design, optimizations and patient-specific modeling to synthesize balance recovery. A scientific framework developed in aims 1-3 is needed in combination with experimental data to validate the simulation results and synthesize balance recovery data and predict movement. Experimental balance recovery data of two healthy adults was used to investigate underlying neural strategies for balance control. A higher-level task-based controller was implemented to track the relationship between the primary and secondary tasks defined using the experimental balance recovery kinematics data and to synthesize new motions. Kinematics and muscle activation patterns were then compared to the experimental balance recovery data. These simulations enable a more rational basis to be developed for evaluating balance recovery and deciding on treatment planning in patients with neurological disorders. To achieve this aim, we:

- 1) Collected and analyzed the experimental balance recovery data of two healthy adults.
- 2) Created patient-specific models and ran kinematics analyses and processed EMGs to enable a reference experimental balance data for our simulations.
- 3) Utilized a high-level controller to estimate muscle excitations to synthesize experimental balance recovery data using forward dynamics simulations.

- a) Defined an initial task force to keep the extrapolated CoM under the base of support and maintain balance
 - b) Defined secondary tasks to move the feet and take step accordingly to avoid falling in cases when extrapolated CoM is outside the base of support.
- 4) Validated our simulated results with the experimental data of healthy adults by defining surrogate response surfaces that represent the patient's primary tasks as function of defined subtasks.

Together, these aims determine the roles of musculoskeletal modeling and balance control in patients with neurological disorders and direct future research for designing more effective treatment planning. The mechanisms behind human movement and balance control are exceptionally complex; yet, simulation gives insight into the complex interaction between muscle excitations, musculoskeletal geometry and observed multi-joint movements. Combining clinical gait analysis and simulation-based approaches may lead to a better understanding of the biomechanical causes of and treatments for movement disorders.

CHAPTER 2

A PLATFORM FOR DYNAMIC SIMULATION AND CONTROL OF MOVEMENT BASED ON OPENSIM AND MATLAB

2.1. Introduction

Numerical simulations play an important role in solving complex engineering problems and have the potential to revolutionize medical decision making and treatment strategies. Musculoskeletal conditions alone cost the U.S. economy over \$849 billion per year (7.7% of the U.S. Gross Domestic Product) and place great demands on healthcare systems worldwide [71]. Modeling and simulation software allows biomechanists to view models, edit muscles, generate simulations, and analyze biomechanical systems. Separately, mathematical computing software allows engineers and scientists to develop algorithms, analyze data, visualize results, and perform numerical computations. The study and treatment of musculoskeletal conditions could greatly benefit from combined software tools that offer a better understanding of neuromuscular biomechanics, and predictive capabilities for optimal surgical and rehabilitation treatment planning.

Two general categories of commercial software packages (engineering and musculoskeletal) have been used for modeling and simulation of biomechanical systems. These systems have been studied using commercial engineering software packages such as ANSYS [72], CATIA [73], ADAMS [74, 75], SD-FAST [76], Working Model [77] and MATLAB [78]. However, it is difficult to use built-in mechanical and electrical elements in these packages to model biomechanical systems. To address this issue, commercial musculoskeletal software packages such as SIMM (MusculoGraphics, Inc.), Visual 3-D (C-Motion, Inc.), and AnyBody (AnyBody Technology) were developed to study different biomechanical systems such as lower

limbs [79, 80] , upper limbs [81, 82], and the cervical and lumbar spine [83, 84]. Unfortunately, these packages do not share model standards or support data exchange as the engineering packages do. Moreover, neither engineering nor musculoskeletal commercial software packages provide access to, or customization of source code, which makes it difficult for researchers to extend software capabilities.

OpenSim [85], a free and open-source (e.g. non-commercial, peer-produced and publicly accessible API header files) modeling and simulation platform, was developed as an extension to the commercial musculoskeletal software package SIMM [79, 86, 87] to create a custom software package for modeling and simulating biomechanical systems. This package has computational tools offering built-in tracking algorithms (e.g., computed muscle control), actuators (e.g., muscle models), and analyses (e.g., muscle-induced accelerations). Its graphical user interface provides a suite of high-level tools for viewing models, editing muscles, generating muscle-actuated simulations, and plotting results. OpenSim has been used to model musculoskeletal movements such as walking [88-90] and running [20, 91] and has been used to investigate biomechanical consequences of surgical treatment options [42, 92, 93]. Although users can extend OpenSim by writing their own plug-ins in C++, this software lacks the robust design and control components which are needed for real-time changes to input controls (e.g., excitations driving torque or muscle actuators), external forces, and initial conditions associated with each simulation.

The MATLAB/Simulink (Math Works, Inc.) package, a powerful mathematical computing and control software, is a natural choice to complement the OpenSim software. Whereas MATLAB provides a programming environment for algorithm development and numerical computation, Simulink extends MATLAB with a graphical environment for rapid

model-based design, control, and simulation of complex dynamic systems. Despite its widespread use, this powerful package has limited resources for simulations of biomechanical systems and could benefit from an interface to the OpenSim musculoskeletal software package.

Currently, there is no freely available, open-source computational tool providing an interface between software packages for modeling and simulating biomechanical systems with robust design and control. Others developed tools (MMS) to convert models from a commercial musculoskeletal software package (SIMM) into Simulink blocks[94] or coupled commercial musculoskeletal tools (AnyBody) with a commercial engineering software package (ANSYS) [95]. Alternatively, some linked their non-commercial simulation software (MSMS) with MATLAB [96]; however, their package uses SimMechanics toolbox for simulating musculoskeletal dynamics. Our goal was to develop and disseminate a free interface (based on MATLAB's well-documented S-Function) between OpenSim and MATLAB/Simulink that combines their relevant strengths such as rapid model-based design, control systems, numerical simulation, and human movement dynamics. The OpenSim and MATLAB/Simulink integrated platform contributes to the overall understanding of human movement and has the potential to improve surgical and rehabilitation treatment planning.

2.2. Methods

We start by defining the basic elements of a forward dynamics simulation of a musculoskeletal model and describe how it works in OpenSim. We then explain the basic concepts and algorithms underlying the new S-function (system function) interface between OpenSim and MATLAB/Simulink. Finally, we demonstrate its application in both open-loop and closed-loop control systems using a model of a human arm.

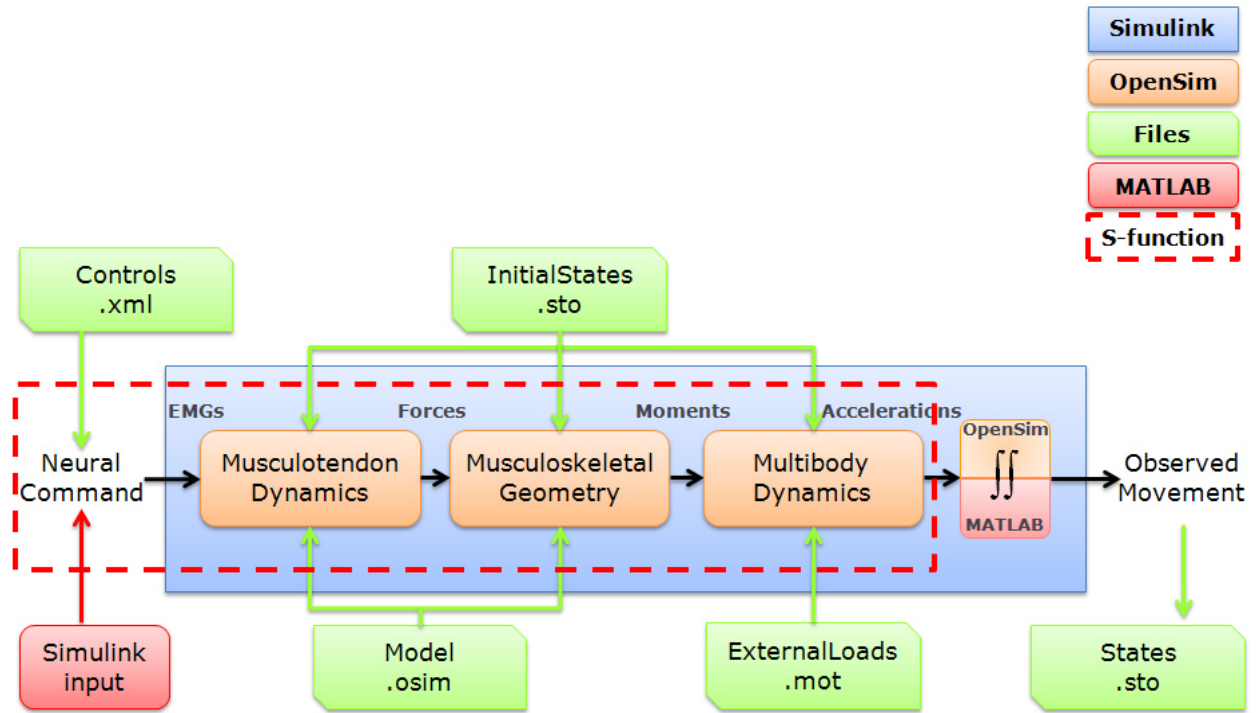


Figure 2. 1: Flowchart illustrating components of forward dynamics in OpenSim and MATLAB/Simulink. Simulink (large blue rectangle) is used instead of the OpenSim Forward Dynamics Tool. The OpenSim transformations (orange rectangles) between a neural command and state derivatives (e.g., joint accelerations) involve musculotendon dynamics, musculoskeletal geometry, and multibody dynamics. Input files (green rectangles) are required by OpenSim for the model, controls (if applicable), initial states, and external loads (if applicable). The Simulink input signal (if applicable) and MATLAB transformation (red rounded rectangles) between joint accelerations and observed movement involves numerical integration. The interface can use MATLAB integrators instead of OpenSim integrators for the forward dynamics. The new S-function interface (red dotted rectangle) takes a controls file or Simulink signal and computes state derivatives subsequently integrated by MATLAB.

2.2.1. Forward Dynamics Simulation of a Musculoskeletal Model

A forward dynamics simulation is the integration of the differential equations that define the dynamics of a musculoskeletal model [97]. In forward dynamics, a mathematical model of the system describes how model states (e.g. joint positions and velocities, muscle activations and fiber lengths) change (due to applied neural excitations driving muscles, forces, and/or torque actuators) according to Newton's second law (1):

$$\ddot{q} = [M(q)]^{-1} \{ \tau + b(q, \dot{q}) + g(q) + F \} \quad (1)$$

where $\ddot{q}$ represents the model's generalized coordinate accelerations vector due to joint torques τ (multiplication of the muscle moment arms matrix by the muscle forces vector), Coriolis and centrifugal forces vector, $b(q, \dot{q})$, as a function of coordinates q , and their velocities $\dot{q}$, gravity vector $g(q)$, vector of generalized forces applied to the model F , and the inverse of the mass matrix $[M(q)]^{-1}$.

The Forward Dynamics Tool in OpenSim (Figure 2.1, orange rectangles) uses a neural command from an input controls file (e.g., controls.xml) to generate an output states file (e.g., states.sto). Potential model inputs are excitations driving muscles, forces, and/or torque actuators. Using musculotendon dynamics, including activation-contraction and muscle-tendon dynamics, the forces actuating the model are computed; subsequently, using musculoskeletal geometry, the joint moments are computed. In addition to the internal loads, external loads such as ground reaction forces and moments are applied to the model for certain movements (e.g. human gait). Accelerations are then found with Eq. (1) as all parameters on the right hand side of the equation are known. OpenSim uses a 5th-order Runge-Kutta-Feldberg integrator (Runge-Kutta-Merson integrator in newer versions) to numerically solve the differential equations that define the

dynamics of a musculoskeletal model and compute the output states of a model including joint positions and velocities (observed movement).

2.2.2. *Interface between OpenSim and MATLAB/Simulink*

The OpenSim and MATLAB/Simulink interface (Figure 2.2.a) is developed to combine relevant strengths of each software package and enable a user to perform musculoskeletal simulations directly in a model-based design and control system environment. The new interface uses the well-documented S-function API to interact with the Simulink engine. An S-function is a computer language description of a Simulink block written in MATLAB, FORTRAN, C, or C++ language (we chose C++ to have easy access to all OpenSim C++ functions and dynamically linked libraries) and compiled as MEX-files. This dynamically linked subroutine can be automatically loaded and executed by MATLAB/Simulink. This interaction is very similar to the interaction that takes place between the engine and built-in Simulink blocks.

Similar to the execution of built-in Simulink blocks, the execution of the S-function interface proceeds in a number of stages (Figure 2.2.b & 2.2.c). First, the Simulink model is initialized. Prior to the first simulation loop, the Simulink engine initializes the S-function's block characteristics by creating SimStruct (simulation structure) containing information such as the number and dimensions of input and output ports. In the next step, the S-function's parameters are processed, which includes creating and setting up an OpenSim model, incorporating the block's sample time, and allocating the required memory. Then, the simulation loop begins and a series of simulation steps begin. During each simulation step, the Simulink engine calls S-function methods that compute the OpenSim model's states, state derivatives, and outputs for the current simulation time. In the next step, the initial conditions (i.e., OpenSim model states) for the integration are set, any defined outputs are calculated, and the states are updated, which

synchronizes the OpenSim model and Simulink block states. The S-function then uses OpenSim methods to calculate the state derivatives necessary to solve (i.e., integrate) for the motion (i.e., states) of the OpenSim model. The Simulink engine repeats previous output and derivative stages at minor time steps to compute the integrated states for the S-function. The integration continues until the solver reaches the desired accuracy for the computed states over the simulation time.

The OpenSim and MATLAB/Simulink interface requires the same inputs as the OpenSim Forward Dynamics Tool such as an OpenSim model file, controls file, initial states file, and external loads file (Figure 2.1, green rectangles). In lieu of a controls file, the user is able to use a Simulink signal, MATLAB workspace structure, or a data file. The S-function uses OpenSim's underlying Simbody dynamics engine and MATLAB integrators to generate the forward dynamics simulation of an OpenSim model (Figure 2.1, red dotted rectangle). The user can easily change Simulink configuration parameters (e.g., integration solver, error tolerance, etc.) to suit the requirements of the problem at hand.

2.2.3. *Open-loop Model Application*

A generic open-loop Simulink model that loads and executes the OpenSim-based S-function was created (Figure 2.3.a). The Forward Dynamics Tool in OpenSim and the S-function block without feedback control in Simulink are both considered open-loop systems. For an open-loop forward dynamics simulation, controls (e.g., excitations driving torque or muscle actuators) are the inputs to the multibody system (e.g., musculoskeletal system) and states (e.g., motion) are the outputs. The output states are the system's response to the input controls. The S-function interface works with any OpenSim model by defining block parameters for a particular simulation (Figure 2.3.b).

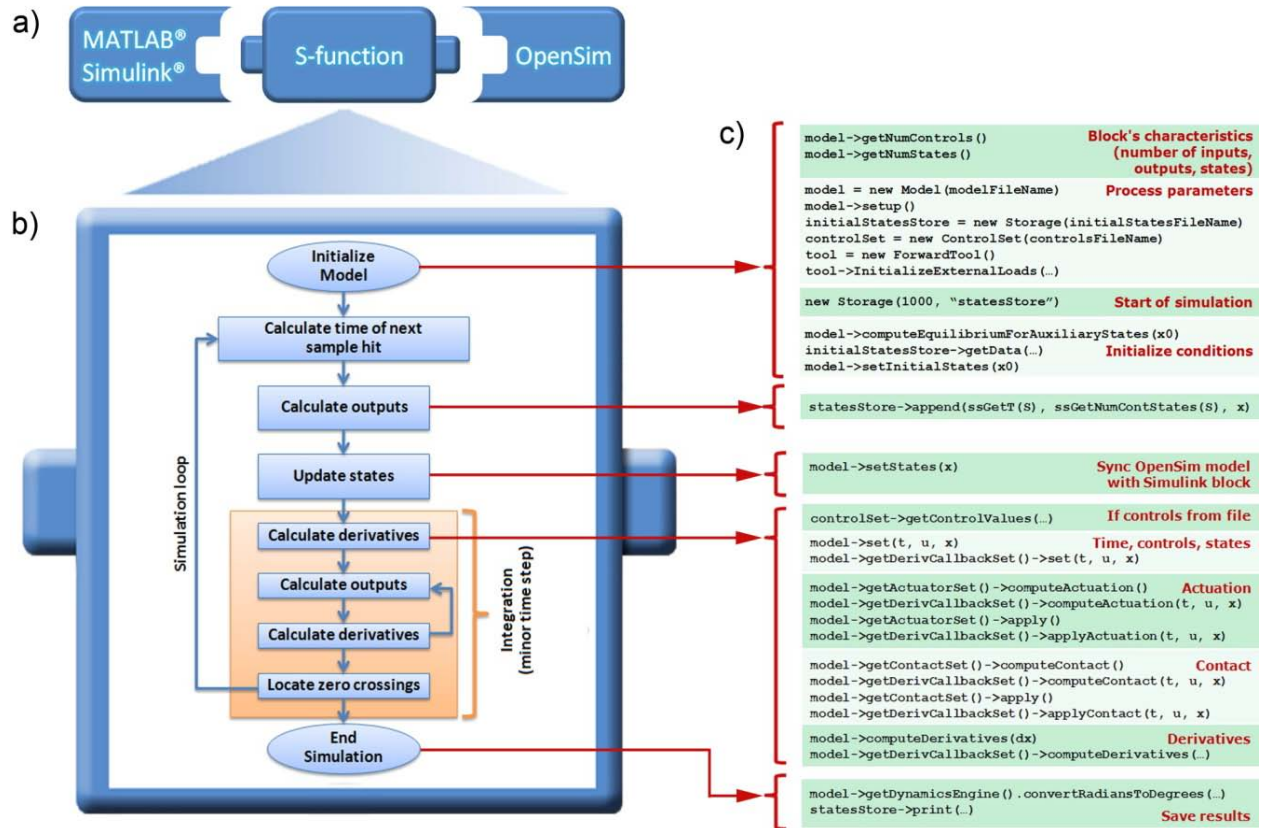


Figure 2. 2: Overview of the new S-function interface. (a) The interface links the rapid model-based design and control systems strengths of MATLAB/Simulink with the numerical simulation and human movement dynamics strengths of OpenSim. (b) Flowchart illustrating stages for each Simulink simulation. (c) OpenSim v1.9 (higher versions not shown) methods called during each simulation stage.

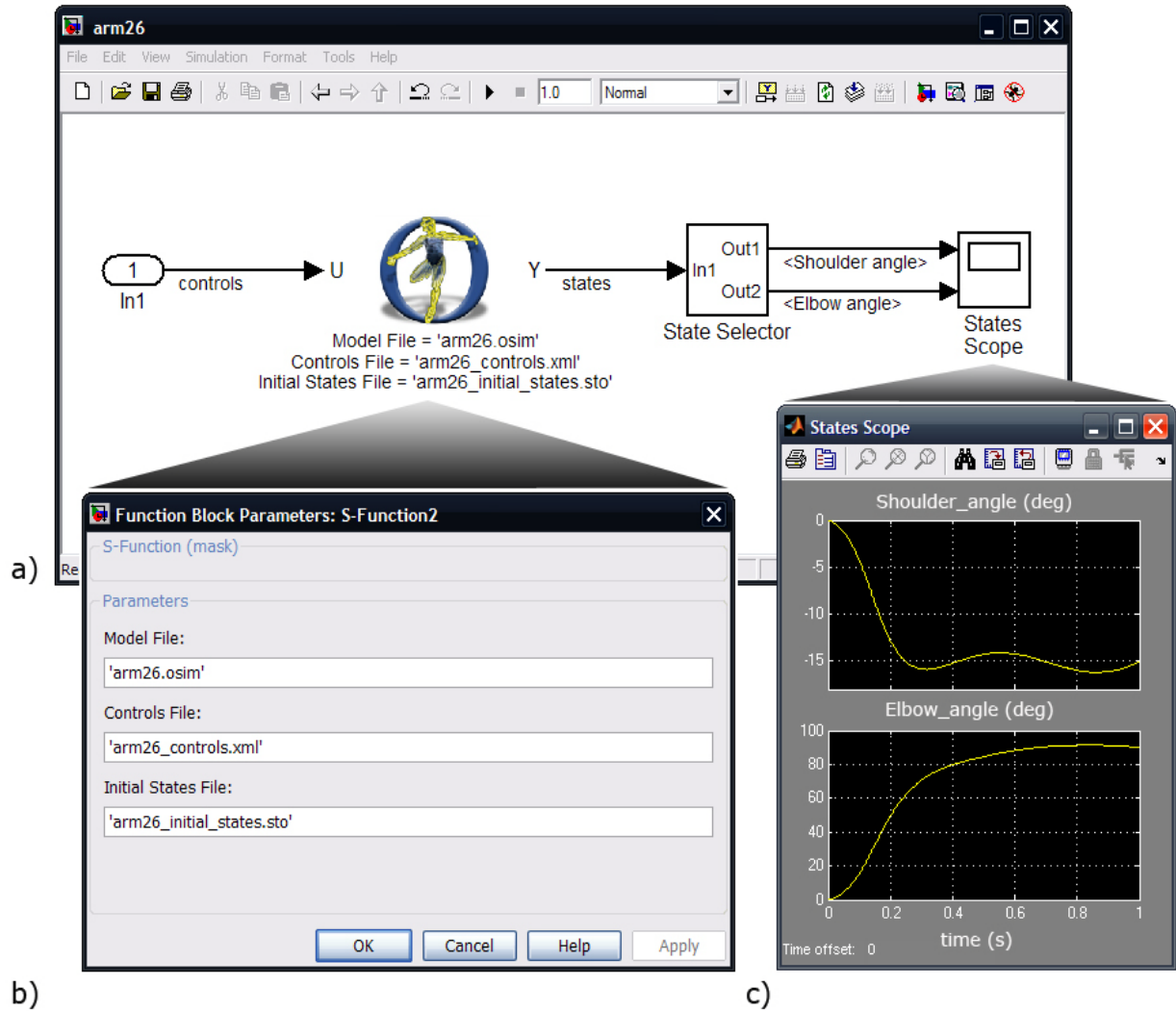


Figure 2. 3: Overview of an open-loop model application for arm flexion. (a) The Simulink graphical editor window shows input controls (U), the new S-function interface (middle), and the output states (Y). (b) The custom Simulink block parameter dialog box shows every input parameter required (or optional) by the OpenSim Forward Dynamics Tool to perform a simulation. (c) The output states may be displayed using a State Selector and Scope within the graphical editor or they may be loaded as a motion file in OpenSim to visualize the simulated movement.

To demonstrate the open-loop characteristics and compare how well the S-function in MATLAB/Simulink agrees with the Forward Dynamics Tool in OpenSim, simple human arm model with 2 degrees of freedom (shoulder elevation and elbow flexion) and 6 muscle-tendon actuators (triceps brachii long head, triceps brachii lateral head, triceps brachii medial head, biceps brachii long head, biceps brachii short head, and brachialis) was utilized to simulate an elbow flexion movement during 1 second simulation. The model's states were integrated in MATLAB using a variable step size, non-stiff, ode45 (Dormand-Prince) integrator. The error tolerance for the MATLAB integrator was the default value of $1e^{-3}$. The error tolerance of OpenSim's integrator was set to $1e^{-3}$ to make it as accurate as the MATLAB integrator. The motions resulting from the OpenSim and Simulink simulations of elbow flexion were directly compared.

2.2.4. Closed-loop Model Application

A generic closed-loop Simulink model that loads and executes the OpenSim-based S-function was created (Figure 2.4). The open-loop Simulink model from the previous section is limited to using predefined input controls that cannot be changed; however, many human movement applications require closed-loop control systems that update input controls to generate a desired output. We extended the open-loop model by adding a proportional-integral-derivative (PID) controller with feedback.

To demonstrate the closed-loop characteristics, a human arm model balancing a pole was implemented in OpenSim. The pole was modeled in OpenSim as a cylinder with mass of 10 kg, height of 30 cm and base radius of 2.5 cm. This biomechanical system had three degrees of freedom (shoulder elevation, elbow flexion and pole rotation), 6 muscle-tendon actuators and a constraint on forearm (using OpenSim point on line constraint function) to keep the hand

connection point with the pole joint to move along the horizontal direction. A PID controller was used to balance a pole above the hand by adjusting the required control for the mentioned arm muscles. We normalized the controller output based on each muscle's maximum isometric force and designed the muscle's gain matrix. We then multiplied the muscle's gain matrix by the controller output to generate the realistic control signal for individual muscles. The controller gains were tuned using the classic Ziegler–Nichols method [98]. The desired pole angle (measured from vertical) was 0° . It was used to compute a pole angle error and, subsequently, a control correction. The initial controls, computed from static optimization, were used to maintain the arm position before a control correction was necessary. The combination of control correction and initial controls is analogous (but not identical) to the human neural command associated with the arm balancing a pole. Random force disturbances were added to the pole to cause an imbalance and the pole angle error was observed.

2.3. Results

The new interface between OpenSim and MATLAB/Simulink allowed rapid model-based design and numerical simulation of human movement using both open-loop (Figure 2.3 & 2.5) and closed-loop (Figure 2.4 & 2.6) control systems.

For the open-loop case, the Simulink generated shoulder angle matched the OpenSim generated angle within a 0.03° root mean square (RMS) difference (Figure 2.5b). The RMS difference for the elbow flexion angle was 0.06° (Figure 2.5c). Using the OpenSim integrator error tolerance of $1e-3$, the computation time was 7.1 seconds with the graphical user interface (GUI) and 3.2 seconds without the GUI (computational speed was assessed on a 3.2 GHz Intel® Xeon® workstation with 3.00 GB of RAM). However, it took 3.9 seconds to generate the same

forward dynamics simulation of the arm model using the new S-function and the MATLAB integrator.

For the closed-loop case, the PID controller successfully rejected random force disturbances ranging between ± 30 N (Figure 2.6b) and balanced the pole with a maximum pole angle error from vertical of 1.19° (Figure 2.6c). Using the MATLAB integrator with the same error tolerance as the open-loop case, the computation time for the closed-loop system was 5.9 seconds.

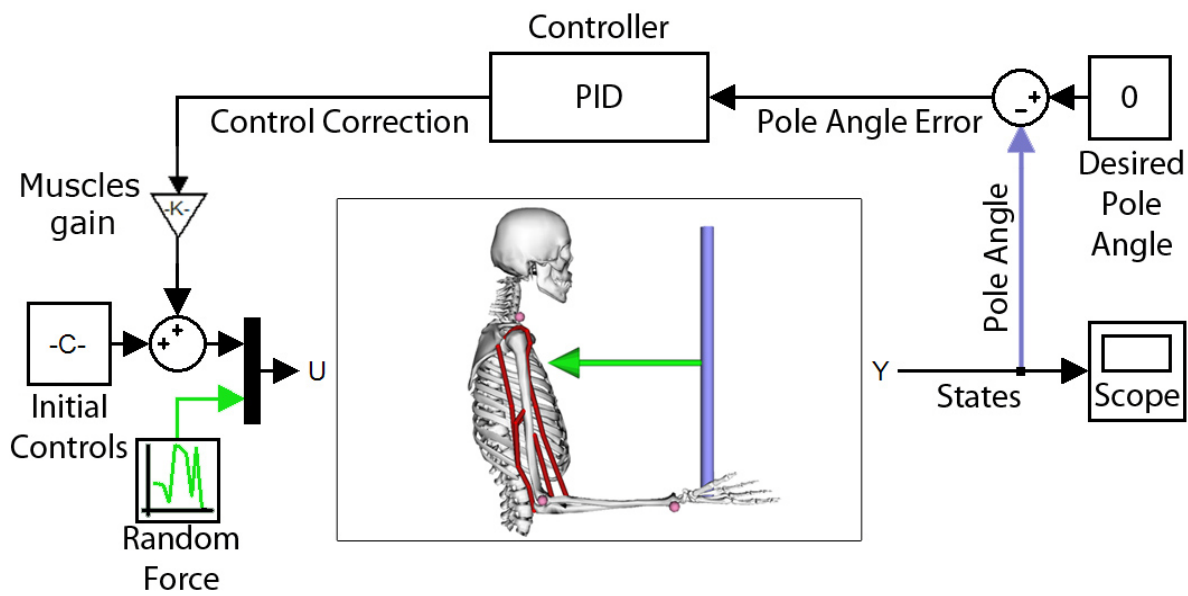


Figure 2. 4: Example closed-loop Simulink model extending the open-loop case with control of a human arm balancing a pole. The desired pole angle (measured from vertical) was zero and used to compute a pole angle error. The proportional-integral-derivative (PID) controller, along with the muscle gain matrix was used to compute control correction signals for each of the six muscles to balance the pole, despite random force disturbances exerted on the pole.

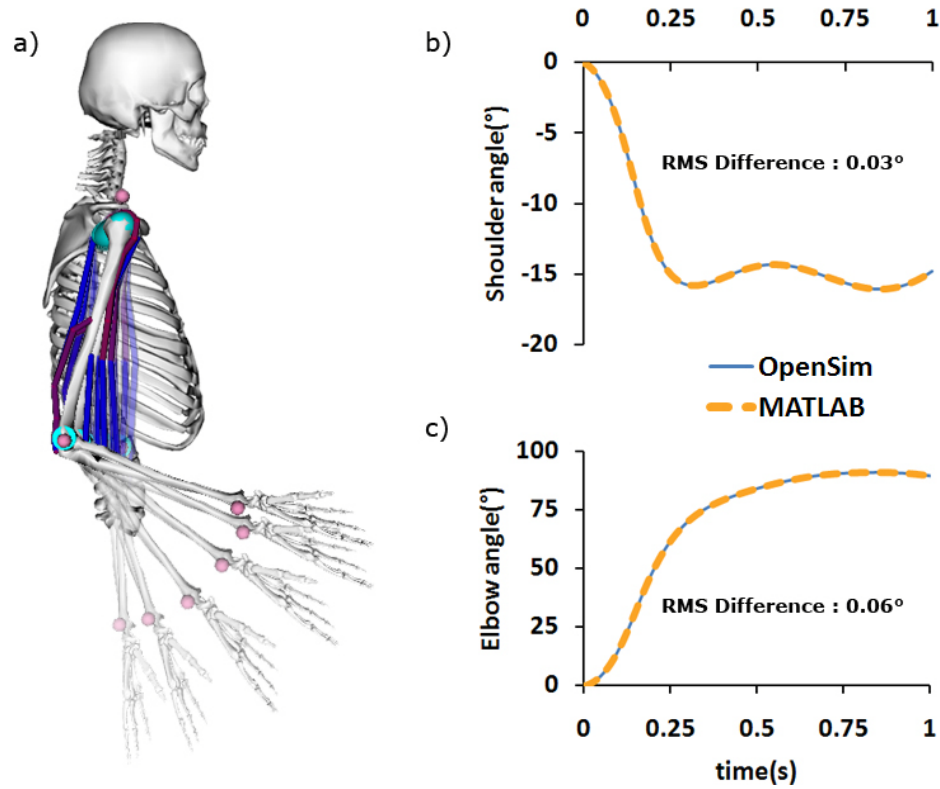


Figure 2. 5: Open-loop simulation of elbow flexion and differences between movements generated by OpenSim and Simulink simulations. (a) A simple human arm model with 2 degrees of freedom and 6 muscle-tendon actuators (six time frame series of a one second elbow flexion movement shown). (b) Shoulder angles and (c) elbow angles for the OpenSim and Simulink simulations.

2.4. Discussion

Rehabilitation and treatment of musculoskeletal conditions could greatly benefit from integrated computational tools to better understand of human movement and plan optimal treatments. Our goal was to develop an interface between OpenSim and MATLAB/Simulink that combines their relevant strengths. The new S-function interface combines the robust design, powerful math, and control system strengths of MATLAB with the numerical simulation and

human movement dynamics strengths of OpenSim. This integrated platform has promise for better understanding movement control and the potential to improve treatment planning.

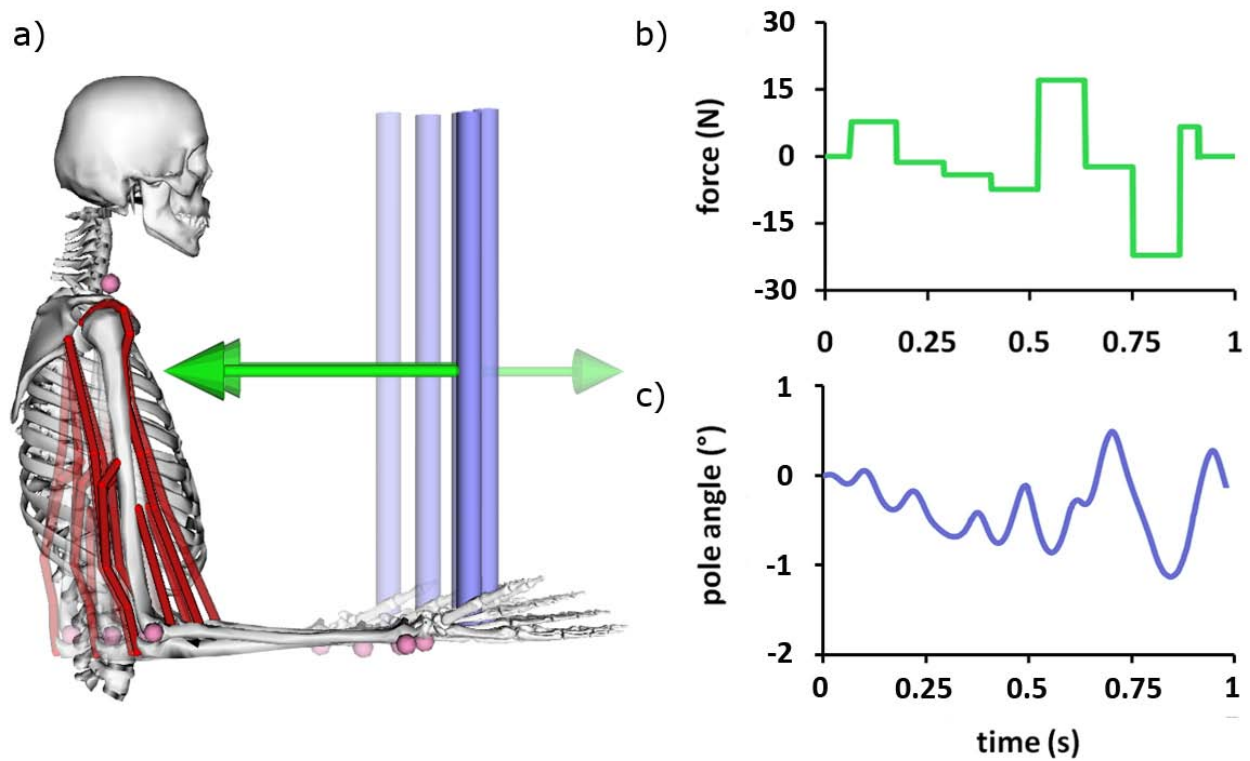


Figure 2. 6: Closed-loop simulation of balancing a pole despite random force disturbances. (a) A PID controlled human arm model balancing a pole (five time frame series from 0.25s to 0.55s shown). (b) Random force disturbances were exerted on the pole and (c) the pole angle measured from vertical remained small.

The RMS difference on the order of $1e-2^\circ$ between OpenSim and Simulink simulated joint angles is reasonable given the differences between the two integrators. Whereas OpenSim uses a 5th-order Runge-Kutta-Feldberg integrator with the error tolerance of $1e-3$, we used the MATLAB variable step size, ode45 (Dormand-Prince) integrator with the default error tolerance

value of $1e-3$. By tightening the error tolerance for each integrator, the RMS difference between MATLAB and OpenSim joint angles can be significantly reduced; unfortunately, this change increases computation time significantly. This time increase becomes even more significant for large, complicated OpenSim models such as the OpenSim gait2354 model used for walking movements. However, a new integrator in OpenSim (v2.2 and higher) uses an error controlled and robust Runge-Kutta-Merson algorithm from SimTK. This integrator is comparable to MATLAB's integrators and is proving to be even faster for stiff problems such as contact modeling. We have ongoing projects employing this new OpenSim integrator.

The S-function and control systems software development in this study had several limitations. First, the compatible versions of OpenSim (v2.3.1), MATLAB (v7.13.0), and Simulink (v7.7) were based on those available at the time of the S-function interface development. Thus, generating the newer versions of the interface may be needed as future versions of these software packages may improve computational efficiency and add new features. However, the current version of the interface, which is compatible with all versions of OpenSim (v1.9-v2.3.1), seems to be sufficient as it contains all of the previous OpenSim features plus the new contact modeling capability. Unlike the biomechanical contact models which can be idealized into joints or constraints in older versions of OpenSim, the new contact model arises from deformations of the compliant materials from which biological systems are composed (e.g. Hertz contact model and Elastic Foundation model) [99]. Second, the Forward Dynamics Tool is the only OpenSim tool extended by the interface and other tools (e.g., Inverse Dynamics, Static Optimization, and Computed Muscle Control) are beyond the scope of this study. With the exception of Computed Muscle Control, OpenSim tools do not require feedback of model states to compute input controls. Third, the PID control was one of the simplest approaches among the

many available closed-loop control systems for the balancing the pole. However, by applying the muscle gain in our closed-loop controller which was designed based on the muscle physiology and maximum isometric force, we made the control signal or muscle excitation to act analogous (but not identical) to the human neural command system. Fourth, despite the popularity and robustness of Ziegler-Nichols PID tuning method, the users should be careful in picking proportional controller gain which can venture into unstable regions and cause the system to become out of control. Although the controller development is not a focus of this study, the new interface allows custom, complex controllers to be developed, tested, and refined. The arm model was intended as a simple application for both open-loop and closed-loop control systems to demonstrate the developed S-function interface; however, complex dynamic and control systems are necessary for clinical problems.

Despite these limitations, the interface allows users to access any OpenSim model within MATLAB/Simulink and perform forward dynamics simulations using the Simbody™ dynamics engine and MATLAB integrators. There is no need for users to recompile the S-function each time they perform a simulation, change models, or switch integrators. The user is able to easily change the integrator within Simulink to suit the problem's stiffness or desired error tolerance. The interface's design is based on OpenSim menus and terminology, which make it easier for users to input the required files as S-function parameters. Moreover, it provides a single interface for all OpenSim-required input data files such as controls, initial states, and external forces. This input data can be provided by a Simulink signal, a MATLAB workspace variable, or a data file. This flexibility adds the unique ability of real-time changes to controls for OpenSim models, which has otherwise been unavailable to OpenSim users to this point and is necessary for feedback control systems to study musculoskeletal conditions. For example, insights into the

potential treatment for gait abnormalities may be investigated. Gait abnormalities commonly observed in children with cerebral palsy are typically treated by surgically altering muscle functions. Unfortunately, this treatment strategy does not consistently result in improved outcomes. Patient-specific simulations have utility to determine the potential efficacy of surgical correction.

Although our interface shares some similarities with other interfaces, this work is fundamentally different from previous work focused on functional electrical stimulation [94] and finite element analysis [95]. Other interfaces are built on commercial dynamics engines such as SD/Fast, and employ commercial musculoskeletal software such as SIMM or AnyBody. In our case, we build on the freely available Simbody dynamics engine and employ the freely available OpenSim software. Unlike other interfaces that require additional steps to compile files containing the equations of motion for the simulation, the generated interface does not require additional compilation steps. The users also should not confuse the developed interface with OpenSim's built-in capability of running the Forward Dynamics Tool using MATLAB command line feature, which only runs OpenSim tools from the command line, without any graphical representation. On the contrary, our tool uses OpenSim's dynamic engine and MATLAB's integrators and combines these in a Simulink block to add the potential of applying it in closed loop control systems.

The potential to use OpenSim and MATLAB/Simulink to study and perhaps improve treatments for musculoskeletal conditions is exciting and valuable. This project not only integrates software tools, but also allows integration of neuroscientists, physiologists, biomechanics, and physical therapists to adopt, adapt, and generate new solutions for musculoskeletal conditions. The entire source codes, Simulink model examples, and user

documentation related to this work is available on a SimTK.org project dedicated to the interface (<https://simtk.org/home/opensim-matlab>).

CHAPTER 3

SIMULATION AND CONTROL PIPELINE FOR PREDICTIVE PATIENT-SPECIFIC NEUROMUSCULAR SYSTEMS

The next step in creating a patient-specific model is to design a pipeline for simulation and control of patient-specific neuromuscular systems. Human movement requires the coordination of many muscles, and the transformations between neural control and purposeful movements are highly complex and involve many individual elements. Many of these individual elements have been characterized through experiments. However, experiments alone limit understanding of movement dynamics, simulations complement experiments. Although a few variables responsible for movement (e.g., kinematics, ground reaction forces, and EMG) can be measured, it is extremely difficult to measure important variables such as muscle activities and joint forces. Predictive modeling and simulation provide tools to investigate these limitations. A new treatment strategy combining physical examination, movement analysis, and simulation-based approaches may lead to a better understanding of human movement control. In order to achieve these goals, we used the following pipeline for the patient specific modeling, simulations and control of musculoskeletal models.

- 1) Create musculoskeletal models using OpenSim software for the specific application
- 2) Scale generic models based on specific subjects' anthropometric data
- 3) Perform the inverse kinematics analysis to determine model kinematics that best match the experimentally measured kinematics.
- 4) Run forward dynamics simulation to estimate states and generate motion due to muscle excitations.
- 5) Design controllers to perform predictive dynamic simulations

3.1. Create Musculoskeletal Models

All the musculoskeletal models were developed using the OpenSim software. OpenSim is available through SimTK.org website. SimTK is the Simbios Biosimulation Toolkit, a collection of open source software tools supporting high-performance physics based simulation of biological structures and includes the OpenSim API. Simbody, SimTK's multibody dynamics engine, is the underlying dynamic engines for OpenSim. Figure 3.1 [100] shows OpenSim's computational and functional layers. OpenSim relies on the computational infrastructure provided by Simbody, particularly for creating and solving the multibody dynamics System. OpenSim [85], a free and open-source (e.g. non-commercial, peer-produced and publicly accessible API header files) modeling and simulation platform, was developed as an extension to the commercial musculoskeletal software package SIMM [79, 86, 87] to create a custom software package for modeling and simulating biomechanical systems.

Using OpenSim generic model libraries, we can modify any generic model and add more complexities such as adding new degrees of freedom, defining custom joints (e.g. pin, ball, slider, pin-in-slot custom joints), introducing foot-ground contact models (e.g. Hunt and Crossley, Elastic Foundation), adding actuators to the models (e.g. torque actuators, muscle actuators). All these model parts maybe needed depending on complexity of the analyzed movement and to what extent these details are required to answer your research questions. We started our simulations in chapter 2 with a simple upper-body model with 2 degrees of freedom and 6 muscle actuators to test the closed-loop forward dynamics simulations. In the next chapter, we used the lower extremity model with 23 degrees of freedom and 92 muscle actuators to simulate balance recovery in children with cerebral palsy.

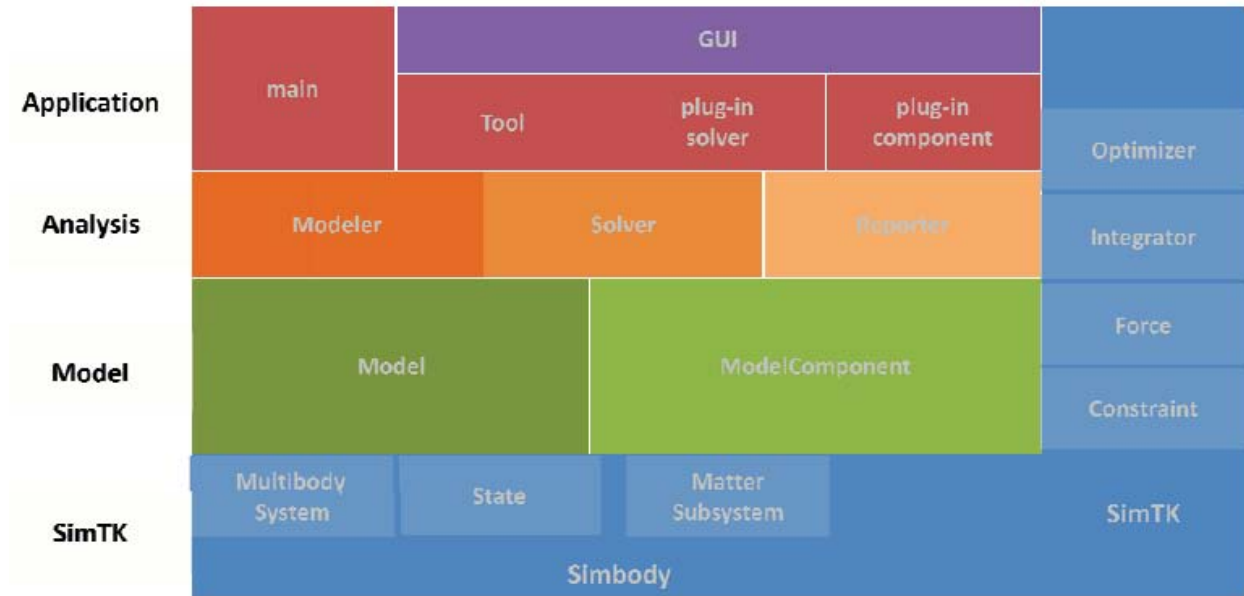


Figure 3. 1: Different OpenSim layers. The base layer is the computational layer provided by Simbody (in blue). Simbody components (e.g. integrators, optimizers) are available to all levels within OpenSim API. The next layer is modeling layer (green). The modeling layer is followed by analyses layer (orange). The top layer is the application layer (red) which is consumer of OpenSim API to generate applications such as OpenSim GUI or dynamic libraries. (Image Courtesy: Seth et al., 2011).

3.2. Scale Generic Models using Subjects-Specific Data

The next step in creating a subject-specific model is scaling a generic model. The objective of scaling is to create a subject-specific model that has the same body properties (e.g. body mass and inertia, body length and height) as the experimental subject. Using the OpenSim scale tool the body mass, inertia and length can be adjusted to represent an individual subject. OpenSim let the user to define the scaling setup. The models can be scaled individually using experimental kinematics data that can be derived from variety of motion capture systems (e.g.

reflective marker data). To reduce the error between the model and experimental markers, static pose is used to calculate the average distance between the experimental markers and the corresponding model markers. Manual iterations may be needed to get good marker positioning on a model that matches the experimental data.

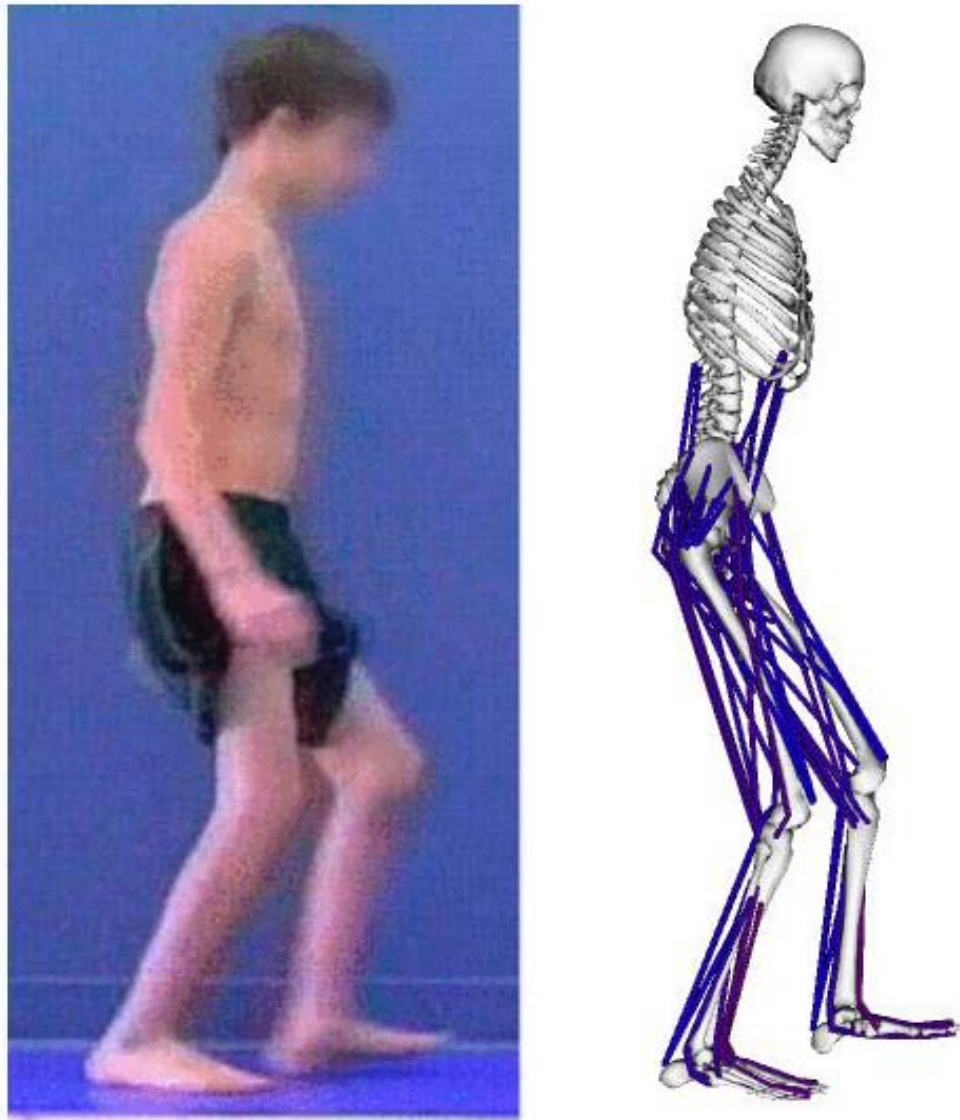


Figure 3. 2: Scaling a generic musculoskeletal model based on the experimental patient-specific data. Models are scaled to represent the length and mass of each segment of the experimental data.

3.3. Inverse Kinematic Analyses

This step is only necessary when the experimental marker data is available and is irrelevant in cases where the output motion is synthesized or predicted due to a set of muscle excitations or joint torque actuators. Synthesized motions can be compared to the experimental motions using the inverse kinematics tools to validate the simulated motions.

The inverse kinematics tool in OpenSim steps through experimental data at each time step (e.g. balance recovery kinematics data) and places the model in a pose that "best matches" the experimental marker and the corresponding coordinate data for that time step. The "best match" is the minimum of the sum of squared errors of markers and coordinates.

Inverse kinematics derives the joint angles from the experimental marker data. A weighted least square algorithm is used to decrease these errors between the experimental and model markers $(X_i^{exp}, X_i(q))$ and generalized coordinates (q_j^{exp}, q_j) , where w_i and ω_j are the user defined weightings.

$$\min_q \left[\sum_{i \text{ markers}} w_i \|X_i^{exp} - X_i(q)\|^2 + \sum_{j \text{ unprescribed coords}} \omega_j (q_j^{exp} - q_j)^2 \right]$$
$$q_j = q_j^{exp} \text{ for all prescribed coordinates } j$$

The result of this step is model joint angles and positions and velocities that accurately match the experimental movement.

3.4. Forward Dynamic Analyses

As discussed thoroughly in the previous chapter, the Forward Dynamics Tool in OpenSim uses a neural command from an input controls file (e.g., controls.xml) to generate an output states

file (e.g., states.sto). Potential model inputs are muscles excitations, joint torque actuators and 35 external forces. Using the musculotendon dynamics (activation-contraction and muscle-tendon dynamics), the forces actuating the model are computed; subsequently, using musculoskeletal geometry, the joint moments are computed. In addition to the internal loads, external loads such as ground reaction forces and moments are needed to calculate proper joint torques.

We extended the OpenSim forward dynamics by adding the real-time feedback from output states to the musculoskeletal model input controls. The input controls are modified through a controller to match a desired movement. Different control algorithms are needed to be implemented based on the complexity of the desired movements..

3.5. Controller Design to Perform Predictive Dynamic Simulations

The forward dynamics tool in OpenSim is limited to using predefined input controls (muscle excitations or joint torque actuators), in a form of a control.xml file that cannot be changed during the simulation to accommodate movement outcomes; however, many human movement applications require a controller that updates input control signals to generate desired output motions. By interfacing OpenSim with Matlab/Simulink, we have access to input controls and output states at each integration time-step. This interface is greatly beneficial to control systems development for various neuromuscular applications and otherwise was not available in commercial biomechanical modeling and simulations software packages.

A complex network of neurons controls human movement. The human central nervous system (CNS) forms the foundation of this complex movement by exciting different muscles. Although signals from the CNS can be categorized into supraspinal and spinal signals with different roles in human, there is not a clear separation of roles for each part of the CNS in

movement control. We designed controllers to represent parts of both spinal and supraspinal system in our simulations to study different control strategies in human locomotion.

3.5.1. *Low-level Spinal Stretch-reflex Controller*

Spinal reflexes associate sensory information about legged mechanics into the muscle activations via alpha motor neurons, and bypass central inputs. Studies showed that positive force feedback of leg extensor muscles play an important role in load-bearing [101], and compliant leg behavior during stance gait [102], suggesting importance of a muscle reflex in explaining important mechanical behavior of the system. Moreover, muscle stretch-reflex models that encodes principles of legged mechanics showed successful in producing human walking dynamics and muscle activities [103].

We implemented a low-level spinal controller based on the stretch-reflex concept (muscle spindles and Golgi-tendon organ (GTO) [104]) for our simulations of balance control. Muscle spindles provide CNS with information about length and contraction velocity of muscles. The GTO provides CNS with information about forces in muscle-tendon complex (MTC) and helps to stabilize posture. Together, these afferent mechanisms provide a simple feedback estimate of muscle-tendon length, velocity, and force for controlling postural responses.

$$Excitation(t) = k_p \left[l_{MTC_{ref}}(t) - l_{CE}(\Delta t) - l_{SE}(\Delta t) \right] + k_d [-V_{CE}(\Delta t)].$$

$l_{MTC_{ref}}$ is the reference muscle-tendon complex (MTC) length; l_{CE} is the contractile element (CE), or the muscle fiber length; l_{SE} is the series elastic (SE), or tendon, slack length; V_{CE} is the CE velocity; Δt is the monosynaptic latency (25ms) [105]; k_p and k_d are the constant position and velocity gains for the stretch-reflex.

A stretch-reflex was developed for our musculoskeletal models to reject support-surface perturbations and balance our models. We started with a simple inverted pendulum model with

one degree of freedom (DOF), and two muscle-tendon actuators, to test the stretch-reflex controller (Figure 3.3). We ran one-second forward dynamic simulations using the closed-loop stretch-reflex controller described above to simulate an inverted pendulum balance control. Initial velocity (-2rad/s) was introduced to the center of mass of the pendulum to perturb the initial motion. The resulting motion, muscle fiber lengths and activations are plotted in Figure 3.4, 3.5 and 3.6

In the following chapter, we applied the stretch-reflex controller to answer questions regarding balance recovery in children with cerebral palsy after support-surface perturbations.

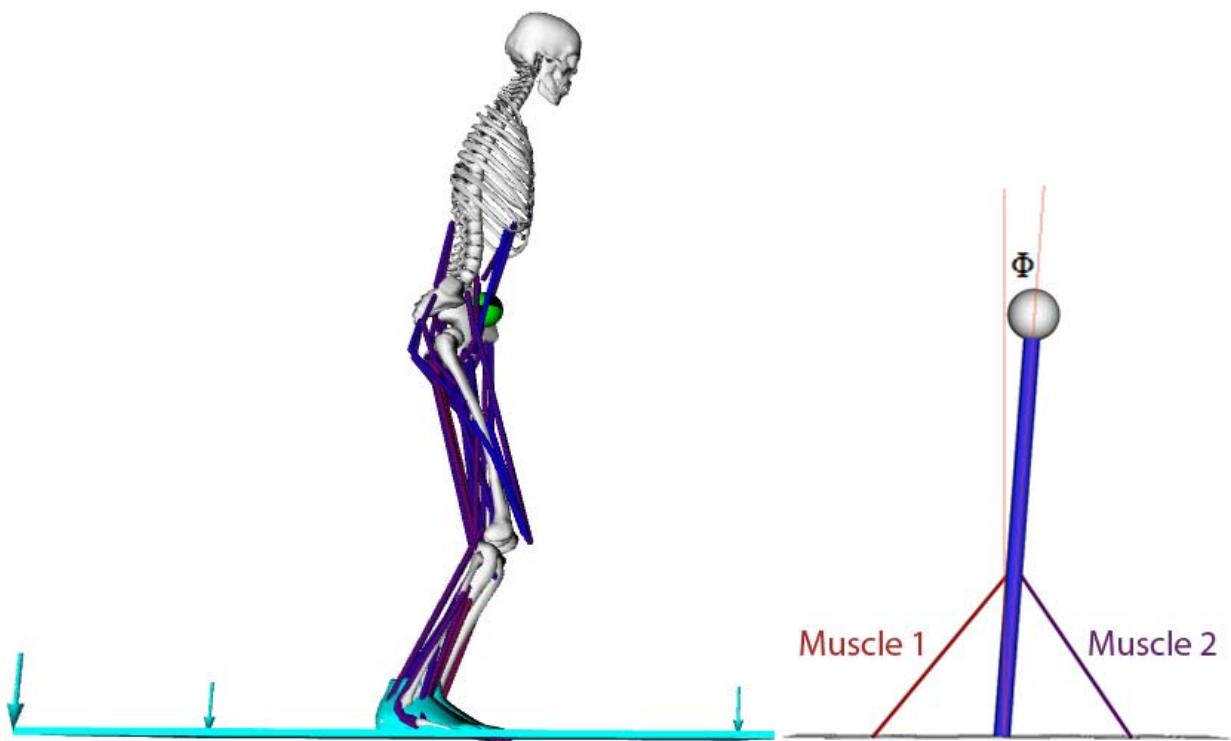


Figure 3. 3: Over simplified model of human balance control using a simple inverted pendulum with two muscle-tendon actuators was used to validate a more complex control problem such as human posture balance with 23 DOF and 92 muscle-tendon actuators.

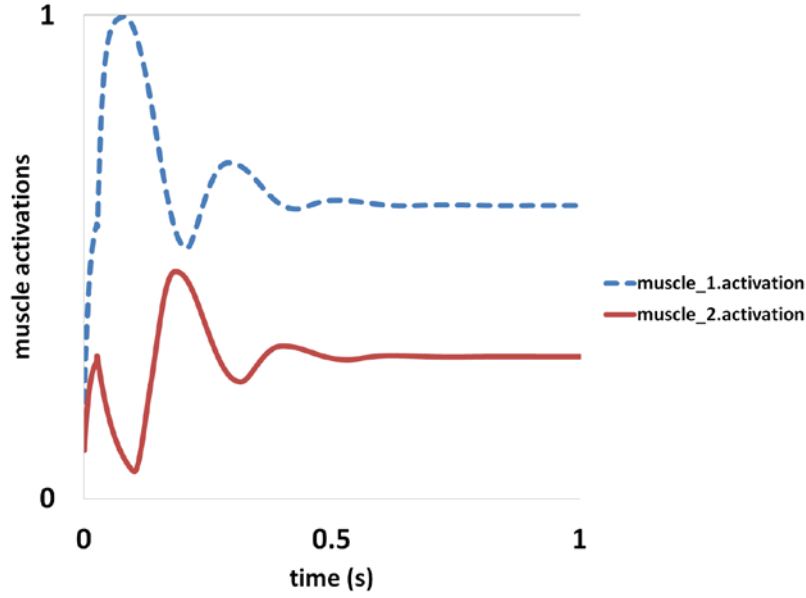


Figure 3. 4: Muscles' activations during a one second forward dynamic simulation with stretch-reflex controller due to the initial pendulum velocity as a perturbation to the system.

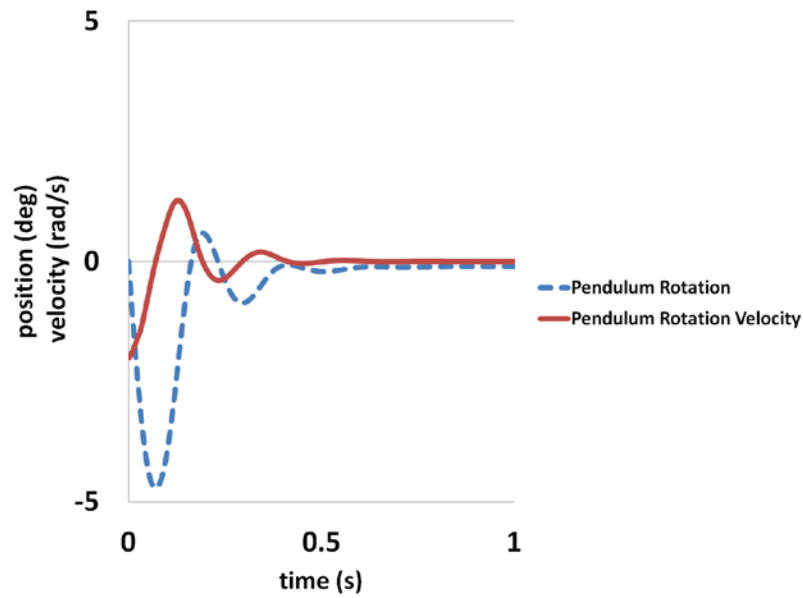


Figure 3. 5: The inverted pendulum rotation angles (Φ) and rotation velocity during a one second forward dynamic simulation with stretch-reflex controller due to the initial pendulum velocity as a perturbation to the system.

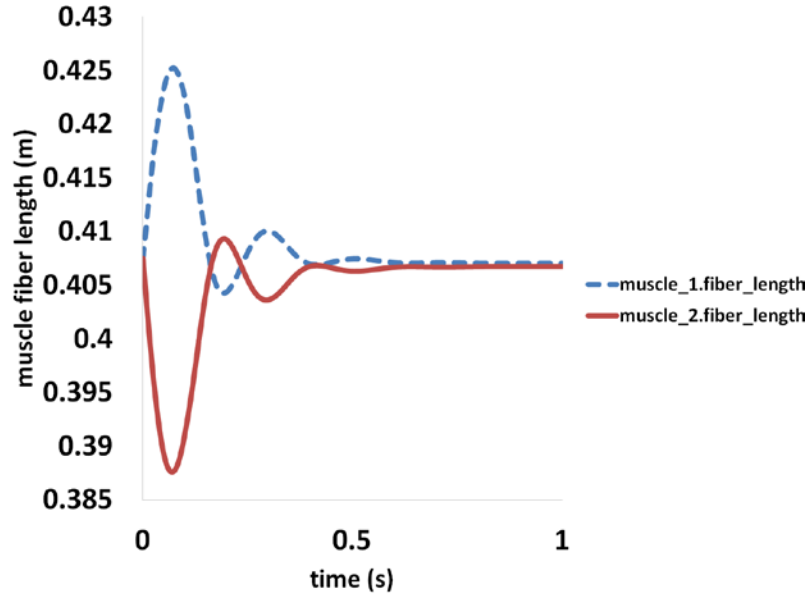


Figure 3. 6: Muscles' fiber length during a one second forward dynamic simulation with stretch-reflex controller due to the initial pendulum velocity as a perturbation to the system.

3.5.2. *High-level Controller*

The low-level stretch-reflex controller is a good starting point for designing a controller that rejects disturbances and balances our models; however, advanced control signals are needed to fully represent complex human movement and different parts of human CNS. The high-level control signals from the brain are believed to be a part of the CNS that is responsible for tracking specific tasks by integration of multiple sensory inputs such as vestibular, visual, and proprioceptive signals. Our controllers do not include all of the sensory organs that provide sensory information for human brain due to the overwhelming complexity of such models. Alternatively, our controllers utilize accurate musculoskeletal states (e.g. joint angles and velocities, and muscle fiber lengths and fiber velocities) by solving the system dynamics along with the neuromuscular feedback to locate the whole-body center of mass in the ground

reference frame at each simulation time-step. This information was used as primary sensory information that is required to control a specific task.

In the following chapters, we first introduce how we can utilize low-level spinal controllers to answer questions regarding balance recovery in children with cerebral palsy (chapter 4). We then build upon our low-level spinal controller using frequently applied task-based control methodologies in robotics by defining high-level task commands (chapters 5 and 6). High-level task commands can range from a task to maintain the whole-body center of mass over the base of support to tasks that control the upper-body and swing leg positions and orientations in space. Introducing and controlling a model using these task-level commands not only reduces the computational costs of simulations, it enables predictive simulation of movement where experimental data is not available.

CHAPTER 4

INFLUENCE OF THE BIARTICULAR RECTUS FEMORIS MUSCLE ON BALANCE RECOVERY IN CHILDREN WITH CEREBRAL PALSY

4.1. Introduction

Stiff-knee gait is one of the most prevalent and troublesome movement abnormalities among children with cerebral palsy (CP) and its symptoms are treatable with surgery involving the rectus femoris muscle. There are approximately 764,000 individuals affected by CP in the United States with about 80% of them are affected by spasticity [106]. Patients experiencing stiff-knee gait typically adopt energy-inefficient movements to compensate for reduced toe-clearance and avoid tripping or falling due to spastic, over active lower-extremity muscles. Stiff-knee gait is often accompanied by excessive knee flexion during stance (crouch gait) [107]. Increase in crouch angle during stance results in altered gravitational moments during initial swing, and consequently reduces the knee flexion during swing phase of gait (stiff-knee gait) [107]. The degree of crouch also play a role in balance control [108]. Over-activity of the rectus femoris is attributed as a primary cause of stiff-knee gait [109]. Various surgical procedures such as rectus femoris transfer surgery and hamstrings lengthening can treat CP symptoms by altering the function of problematic muscles including the rectus femoris [110, 111]. Rectus femoris transfer surgery is one such procedure, which aims to convert the muscle's knee extension moment to a knee flexion moment. Unfortunately, outcomes following surgery are met with variable success [112, 113].

Rectus femoris is a biarticular muscle, acting as both a hip flexor and knee extensor, and may play a unique role in maintaining balance during dynamic tasks [114, 115]. Some have suggested biarticular muscles act as energy transfer straps across joints [116]. Furthermore,

others have suggested that biarticular muscles play a unique role in motor control and are among the first muscles affected in persons with CP [117]. As a biarticular muscle, rectus femoris may be important in maintaining balance and may have deleterious effects when transferred to become a knee flexor.

Recommendations for rectus femoris transfer surgeries are generally based on physical examinations along with clinical movement analysis and surgical treatments; the biomechanical consequences and outcomes following rectus femoris transfer surgery are rarely considered. Physical examination of a single passive joint motion does not address coordinated multi-joint movements. Clinical movement analysis characterizes the motion of limb segments, but not the individual muscle contributions causing this motion. Muscle-actuated simulations and bio-inspired control systems potentially give insights into the complex interaction between neural commands, musculoskeletal geometry, and resulting functional movements. Combining clinical movement analysis and simulation-based approaches may lead to a better understanding of biomechanical consequences of stiff-knee gait and its treatments.

The objective of this study was to determine the effects of simulated rectus femoris transfer surgeries on the balance recovery of children with CP under two different crouched postures. We hypothesized that the stability of pre-surgical simulations would be different (i.e., have better or worse balance recovery) than post-surgical simulations of unilateral or bilateral rectus femoris transfers. We tested this hypothesis by comparing the minimum distance from the extrapolated whole body center of mass ($\text{CoM}_{\text{extrp}}$) to the base of support (BoS) boundary as well as the minimum time for the $\text{CoM}_{\text{extrp}}$ to reach the BoS boundary for each simulation [118]. We also used these muscle-actuated simulations of balance recovery to determine whether moderate crouched postures are more stable than mild crouched postures.

4.2. Methods

We determined the influence of the biarticular rectus femoris muscle on balance recovery in children with CP by performing the following four steps: 1) creating patient specific musculoskeletal models of children with spastic CP before and after rectus femoris transfer surgery adopting either a mild or moderate crouched posture; 2) designing a biologically-inspired, closed-loop control system for balance recovery; 3) generating forward dynamic simulations to examine balance recovery following support-surface translations; and 4) evaluating stability margins using the extrapolated center of mass position and minimum time-to-boundary of the base of support.

4.2.1. *Musculoskeletal Models*

Three-dimensional musculoskeletal models with 92 muscles and 23 degrees of freedom were constructed in OpenSim [119]. The Hill-type muscle-tendon model was used as the basis for each muscle in the simulations [120]. The foot-ground contact geometry were based on 3D scans of cadaver feet [121] and ground reaction forces were modeled using elastic foundation mesh-based contact [119]. The models were scaled to represent the size of children with spastic CP [122]. To simulate rectus femoris transfer surgery, we modified the pre-surgical models to create post-surgical ones by reattaching the distal tendon of the rectus femoris from the patella (Figure 4.1a) to the insertion of the Sartorius on the tibia (Figure 4.1b). The tendon slack length of the transferred muscle was scaled using anthropometric mass and heights of the subjects to ensure the muscle fibers operated near their pre-surgical length ranges [109]. The tendon attachments on bones and muscle via-points are defined based on the MRI images of subjects with CP [123] and modeled using OpenSim via points. The unilateral model represented a simulated transfer on the left limb only, while the bilateral model represented a transfer on both

limbs. The crouched postures of the models were selected based on average kinematics data in our subject pool consisting of children with cerebral palsy adopting mild (ankle, knee and hip angle of 13.8° , 20.9° and 17.5° , respectively) and moderate (ankle, knee and hip angle of 20.6° , 37.3° and 28.1°) crouch gait. For each model to maintain balance during forward dynamic simulations, a control system design was necessary.

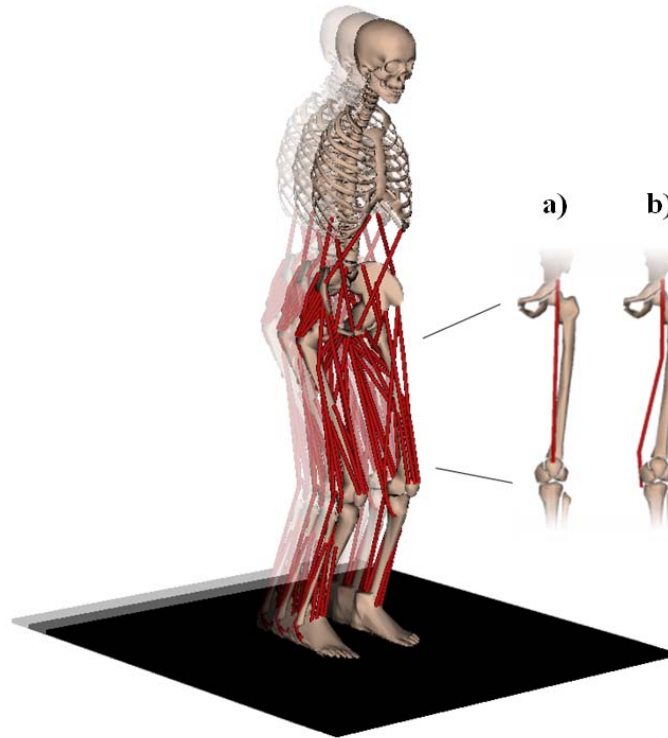


Figure 4. 1: Three-dimensional, 23 degree-of-freedom musculoskeletal model with 92 muscle-tendon actuators (shown in red) recovering balance on a support surface translating 7.5 cm over 0.55 seconds with a maximum velocity of 18 cm/s. The musculoskeletal model was used along with a biologically-inspired controller to create forward dynamic simulations of children with mild and moderate crouch to test our hypothesis regarding balance recovery following rectus femoris transfer surgery. Biarticular locations for the rectus femoris muscle are shown as (a) the pre-surgical attachment to patella and (b) the post-surgical transfer to the insertion of the sartorius on the tibia.

4.2.2. *Biologically-inspired Controller*

Each musculoskeletal model used a biologically-inspired controller to track the experimental CoM displacements similar to the Central Nervous System (CNS) combination of high-level supraspinal and low-level spinal signals in controlling human balance. The high-level controller utilized computed muscle control (CMC) [97] to calculate muscle excitations (peripheral nerve action potential, which initiates the muscles excitation-to-activation process signals) to maintain a static posture in spite of the forces of gravity.

We utilized OpenSim CMC to calculate muscle excitations that drive a dynamic musculoskeletal model to track a set of desired kinematics. The CMC algorithm utilizes a combination of feedback control, static optimization and forward dynamics to estimate muscles excitations required for tracking a movement [97]. The forward dynamics portion uses muscle excitations calculated from static optimization to drive a model that replicates the experimental joint motion. The feedback controller ensures the muscle excitations generate a simulated movement that tracks the experimental one. The low-level controller utilized a stretch-reflex based on a combination of muscle spindles and Golgi-tendon organ (GTO) [104]. The muscle spindle provides the CNS with information about length and contraction velocity of muscles. The GTO provides the CNS with information about forces in muscle-tendon complex (MTC) and helps to stabilize posture. Together, these afferent mechanisms provide a simple feedback estimate of muscle-tendon length, velocity, and force for controlling postural responses (Figure 4.2).

$$Excitation(t) = CMC(t) + k_p [l_{MTCref}(t) - l_{CE}(\Delta t) - l_{SE}(\Delta t)] + k_d [-V_{CE}(\Delta t)]$$

l_{MTCref} is the reference MTC length; l_{CE} is the contractile element (CE), or the muscle fiber length; l_{SE} is the series elastic element (SE), or tendon, slack length; V_{CE} is the CE velocity;

$\Delta t = t - 0.025$, is the time minus monosynaptic latency (25ms) [124]; k_p and k_d are the position and velocity gains for the stretch-reflex.

Stretch-reflex gains were chosen to represent subjects with non-spastic, typically developing (children without CP) and spastic CP responses to stretch. The gains were determined using a non-linear least squares optimization in MATLAB (lsqnonlin) minimizing the CoM position and velocity errors between simulated and experimental CoM displacements [125] for typically developing subjects. The optimization was performed over different time periods (0.55s, 1s, 2s and 2.8s); however, simulation time increase in optimization did not have a major effect on resulting gains after the first 55s. In addition, we tracked hip, knee and ankle joints in our optimization cost function to avoid unrealistic joint motions. An upper- and lower-boundaries for the design variables were used to avoid unrealistic gains. Spasticity was simulated by multiplying the non-spastic position and velocity gains by a factor of 1.1 (10% increase in muscle gains) to increase the sensitivity of the controller [126].

4.2.3. Forward Dynamics Simulations

The musculoskeletal models and biologically-inspired controller were used to create forward dynamic simulations of balance recovery to test our hypothesis regarding stability of children with CP before and after transfer surgery as well as comparing different crouched postures (Figure 4.1). We used an OpenSim/MATLAB interface [127] to generate simulations of balance recovery following support-surface translations. The translation was defined using a generalized cross-validation spline function prescribing the position of the support surface as a function of time based on clinical studies [70]. The support surface was translated 7.5 cm in the anterior and posterior directions with a peak velocity of 18 cm/s, which took 0.55s to complete. A total of 21 forward dynamic simulations were performed for combinations of non-spastic,

spastic, mild crouch, moderate crouch, pre-surgical, and post-surgical models (Tables 4.1 and 4.2). Each 4-second simulation involved the 0.55s support-surface translation and 3.45s of balance recovery to record the stability margins and potential fall due to oscillations.

4.2.4. *Dynamic Stability Margins*

We quantified our balance recovery data using the dynamic stability margins formulations described by Hof et al., [118]. A spatial stability margin ($b = BoS_{min} + CoM_{pos} + \frac{CoM_{vel}}{\sqrt{g/l}}$) is defined as the minimum distance between extrapolated center of mass (CoM_{extrp}) and the BoS boundary, and a temporal stability margin ($TtB = \frac{b}{CoM_{vel}}$) is defined as the minimum time-to-boundary for the CoM_{extrp} to reach the BoS boundary (Figure 4.3) [118]. The CoM_{extrp} is the CoM displacement extrapolated in the direction of its velocity. Larger b_{min} values indicate the CoM_{extrp} is farther from the BoS boundary and the simulated condition is more stable than other conditions with smaller b_{min} values. Likewise, larger TtB_{min} values indicate more time is necessary for the CoM_{extrp} to reach the BoS boundary and the condition is more stable than others. The subjects BoS was 20 cm. We evaluated our hypothesis regarding the stability of children with CP under different surgical conditions by comparing the changes in TtB_{min} values, which are functions of b_{min} and CoM_{extrp} velocity. A two-tailed, paired t -test at the 0.05 significance level was performed against the alternative hypothesis that the stability of pre-surgical simulations is different from post-surgical ones (for both anterior/posterior support-surface translations, for both mild/moderate crouch posture and for both unilateral/bilateral surgical cases). We analyzed the differences in b_{min} and TtB_{min} stability margins by comparing pre-surgical simulations to post-surgical ones following anterior and posterior support-surface translations; in addition, we analyzed differences between mild and moderate crouched postures.

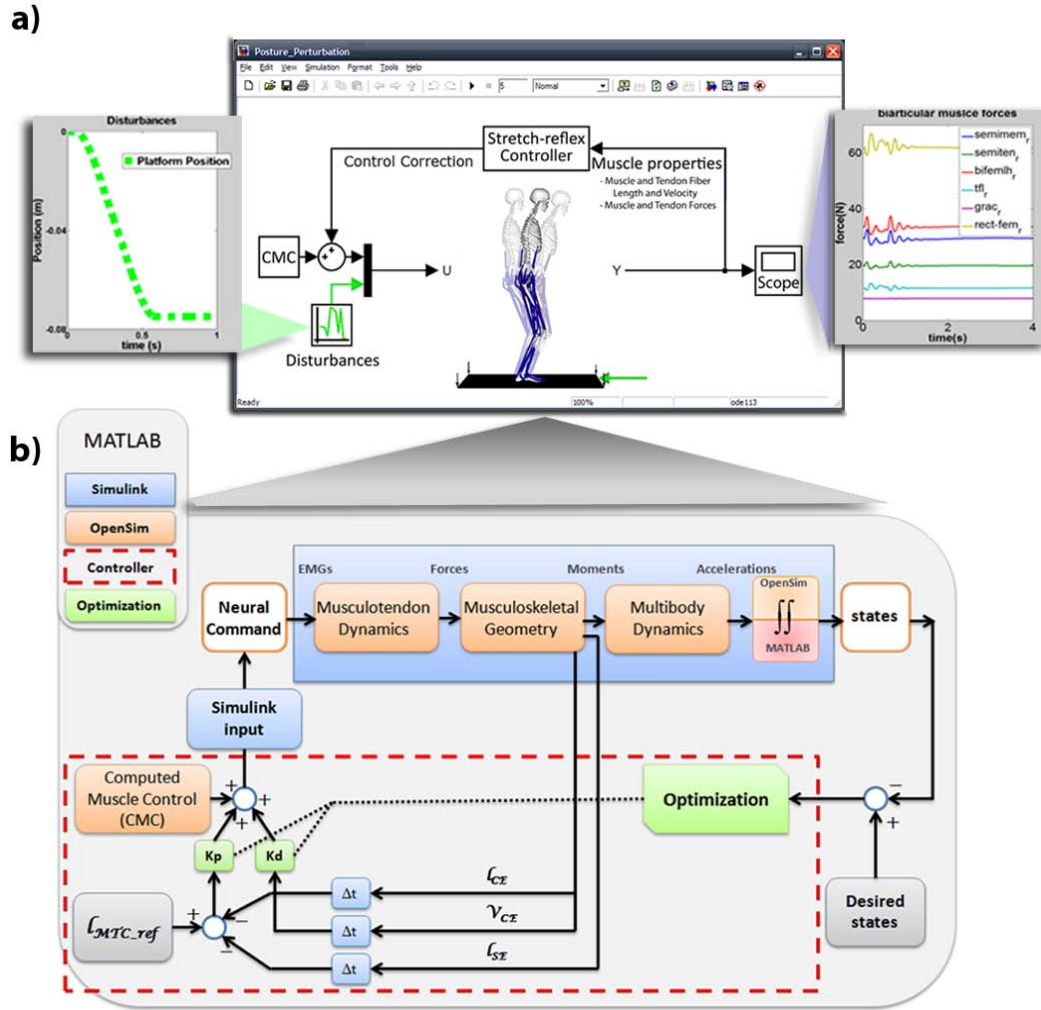


Figure 4. 2: Schematic of (a) the closed-loop forward dynamic simulation in Simulink and (b) the underlying algorithms. The inputs to the musculoskeletal model, or neural command, were calculated based on the low-level (stretch-reflex controller) and high-level (computed muscle control) feedback control system. Disturbances (e.g., support-surface translations) may also be introduced to the system as inputs. The outputs of the dynamic simulation are the model's states (i.e., joint angles and velocities, muscle activations and fiber lengths). The stretch-reflex controller gains, k_p and k_d , were selected by a non-linear least squares optimization minimizing position and velocity errors between simulated and experimental center of mass displacements.

4.3. Results

The balance recovery of all pre-surgical simulations of children with CP ($b_{\min} = 2.31+1.12\text{cm}$, $TtB_{\min} = 0.19+0.10\text{s}$) was significantly different ($p=0.022$), on average, than post-surgical ones ($b_{\min} = -4.94+11.38\text{cm}$, $TtB_{\min} = -0.09+0.32\text{s}$). The negative values indicate that models are falling outside the base of support. Following posterior support-surface translations, all pre-surgical simulations ($b_{\min} = 3.09+0.95\text{cm}$, $TtB_{\min} = 0.27+0.09\text{s}$) were significantly more stable ($p=0.002$), on average, than post-surgical ones ($b_{\min} = -11.79 +15.10\text{cm}$, $TtB_{\min} = -0.34+0.23\text{s}$) (Table 4.1). Following anterior support-surface translations, all pre-surgical simulations ($b_{\min} = 1.52+0.61\text{cm}$, $TtB_{\min} = 0.12+0.04\text{s}$) were not significantly different ($p>0.05$), on average, from post-surgical ones ($b_{\min} = 1.89+0.82\text{cm}$, $TtB_{\min} = 0.15+0.06\text{s}$). The pre-surgical simulations maintained balance following both anterior and posterior support-surface translations (Figure 4.4). On the contrary, the post-surgical simulations (unilateral and bilateral transfers) were not able to recover balance following posterior support-surface translations (Figure 4.4, bottom row).

Balance recovery was also influenced by crouched posture (Figure 4.4, columns). Following anterior support-surface translations, the moderate crouch simulations ($b_{\min} = 2.36+0.42\text{cm}$, $TtB_{\min} = 0.18+0.03\text{s}$) were more stable than the mild crouch simulations ($b_{\min} = 1.18+0.27\text{cm}$, $TtB_{\min} = 0.095+0.02\text{s}$) (Table 4.1). Following posterior support-surface translations, the stability of moderate crouch simulations ($b_{\min} = -0.78+3.54\text{cm}$, $TtB_{\min} = -0.05+0.27\text{s}$) was not significantly different, compared with mild crouch simulations ($b_{\min} = -12.87+19.18\text{cm}$, $TtB_{\min} = -0.22+0.48\text{s}$).

Alternatively, Table 4.2 shows the quantitative values of CoM_{extrp} , CoM_{vel} and center of pressure (CoP) which was used to calculate stability across simulations of balance recovery.

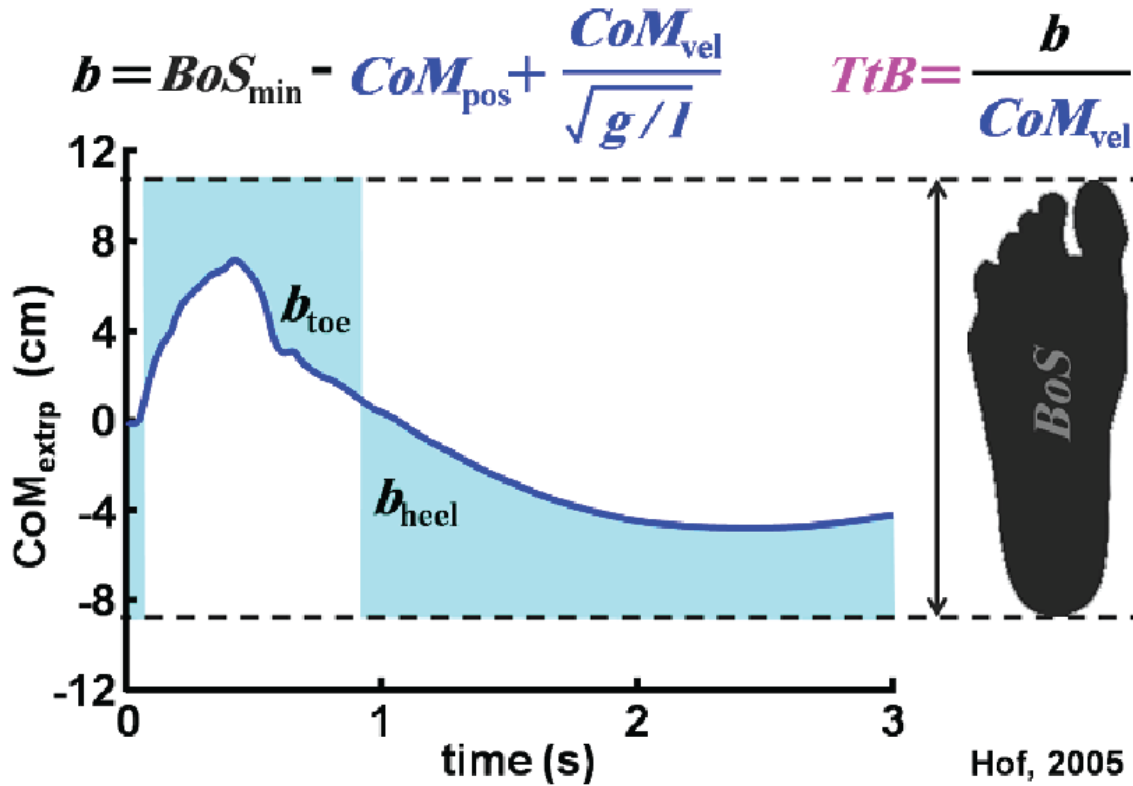


Figure 4. 3: The extrapolated center of mass of a pre-surgical model following posterior support-surface translation as a function of simulation time. CoM_{extrp} was chosen over CoM and center of pressure (CoP) in defining the stability margins because models may be moving toward a fall that occurs after the simulation time despite the CoP being located within the BoS throughout the simulation. The functions b and TtB are defined based on Hof et al., 2005 and the minimum values of these functions were used as an indicator of how stable the simulations are at each time step.

Table 4.1: Comparison across simulations of balance recovery defined by a spatial stability margin, $b_{\min}$, which was the minimum distance between extrapolated center of mass ($\text{CoM}_{\text{extrp}}$) and the BoS boundary, and a temporal stability margin, $\text{TtB}_{\min}$, which was the minimum time-to-boundary for the $\text{CoM}_{\text{extrp}}$ to reach the BoS boundary. The negative values on the table show the $\text{CoM}_{\text{extrp}}$ is outside of BoS and the model is indeed falling. Additional information about these simulations of balance recovery is available as supplementary material (Table 4.2)

Crouched posture	Support-surface translation	Pre-surgery		Post-surgery			
		$b_{\min}$ (cm)	$\text{TtB}_{\min}$ (s)	Unilateral		Bilateral	
				$b_{\min}$ (cm)	$\text{TtB}_{\min}$ (s)	$b_{\min}$ (cm)	$\text{TtB}_{\min}$ (s)
Mild	Anterior	1.09	0.09	0.98	0.08	1.49	0.12
	Posterior	3.76	0.33	-33.86	-0.55	-8.52	-0.44
Moderate	Anterior	1.96	0.15	2.32	0.19	2.80	0.21
	Posterior	2.42	0.21	-0.20	-0.03	-4.58	-0.33

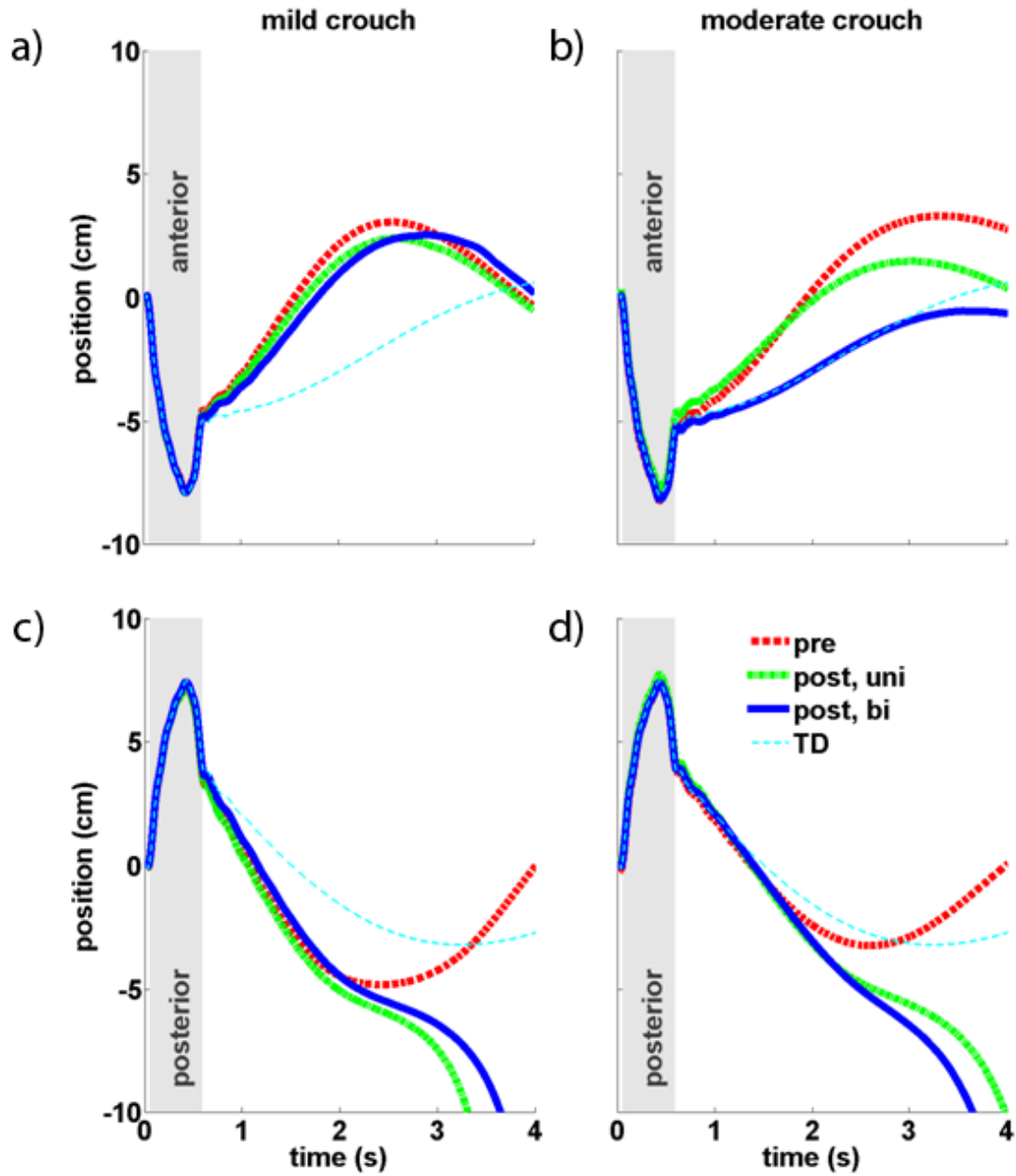


Figure 4. 4: Extrapolated center of mass displacements relative to the support-surface during anterior (a & b) and posterior (c & d) translations (*shaded regions*) for simulations before (pre) and after (post, unilateral and bilateral) rectus femoris tendon transfer for mild (a & c) and moderate (b & d) crouched postures. The displacements for simulations of the typically developing (TD) model without a spastic stretch-reflex are shown as well.

Table 4.2: Quantitative values of CoM_{extrp} , CoM_{vel} and CoP were used to calculate differences of stability across simulations of balance recovery.

Crouched posture	Support-surface translation	Pre-surgery			Post-surgery					
					Unilateral			Bilateral		
		CoM_{extrp} (cm)	CoM_{vel} (cm/s)	CoP (cm)	CoM_{extrp} (cm)	CoM_{vel} (cm/s)	CoP (cm)	CoM_{extrp} (cm)	CoM_{vel} (cm/s)	CoP (cm)
Mild	Anterior	3.93	5.83	8.04	3.98	5.91	8.79	3.82	5.28	7.88
	Posterior	3.33	5.64	6.56	10.86	13.83	8.41	6.33	6.93	7.14
Moderate	Anterior	4.04	5.18	5.15	3.69	5.12	5.24	4.42	4.62	5.35
	Posterior	2.91	4.87	3.22	4.82	5.48	4.66	6.55	6.41	5.16

4.4. Discussion

Rectus femoris transfer surgery is a common procedure for treating stiff-knee gait, converting the function of this the biarticular muscle from a knee extensor to a knee flexor, which may influence an individual's ability to maintain balance. Several studies have suggested that biarticular muscles act as energy transfer straps across joints, are among the first muscles affected in persons with CP, and play a unique role in motor control [114, 116, 117]. The objective of this study was to determine the how the biarticular rectus femoris muscle influences the balance recovery of children with CP. Our results confirm that changing the rectus femoris muscle's function as a knee extensor to that of a knee flexor had a negative effect on balance recovery during simulations following different perturbations. Pre-surgical simulations were significantly more stable than post-surgical ones during posterior support-surface translations. Balance recovery was also influenced by crouched posture. The moderate crouched postures were more stable than mild crouched postures; however, this difference was only significant following anterior support-surface translations.

The significant differences in balance recovery observed in pre- and post-surgical simulations and during anterior versus posterior support-surface translations can be explained by following biomechanical concepts. For anterior support-surface translations, the joint motions to maintain balance require flexing the knee to compensate for an extension induced by the anterior motion of support surface. Although our results showed that both pre- and post-surgical simulations maintained balance, the transferred rectus femoris muscle enhanced the ability of the knee flexor muscle group during balance recovery. For posterior support-surface translations, the inability to maintain balance in post-surgical simulations (both unilateral and bilateral transfers) was apparent. Joint motions to maintain balance require the simulation to extend the knee to

compensate for a flexion induced by the posterior motion of support surface. In addition, dynamic coupling play a major role in human movement dynamics. Zajac and Gordon [128] showed due to dynamic coupling, the biarticular muscles can induce accelerations in direction opposite to the joint moment they generate. The transfer surgery eliminated the role of biarticular rectus femoris as a knee extensor, which resulted in a negative effect on balance recovery following support-surface translations. Furthermore, Clark [129] showed that negative effects on balance were minimal in absence of other monoarticular knee extensor such as vastus medialis. The simulated results show the rectus femoris, along with other knee extensors play a role modulating the knee kinematics to maintain balance. The transfer surgery changes the role of rectus femoris, which adversely affects the balance recovery following support-surface perturbations.

These results for anterior and posterior support-surface translations also hold true for both the mild and moderate crouch simulations. The moderate crouched postures were more stable than mild crouched postures; however, this difference was only significant following anterior support-surface translations. Our simulation results for mild and moderate crouch patients in this study were similar to crouched postures from Hoang, et al. [108]. They showed several crouched postures afford the lower limb muscles the ability to generate increased ground reaction forces. Furthermore, our results in both perturbation directions of a control subject undergoing perturbation showed good conformity with the experimental healthy, upright adults from Ting [130] and Henry, et al. [125] (Figure 4.5). Results from this study highlight the importance of the rectus femoris muscle as a biarticular muscle in balance recovery. Further research is needed to determine whether mild and moderate crouch simulations possess clinically meaningful differences in the context of balance recovery.

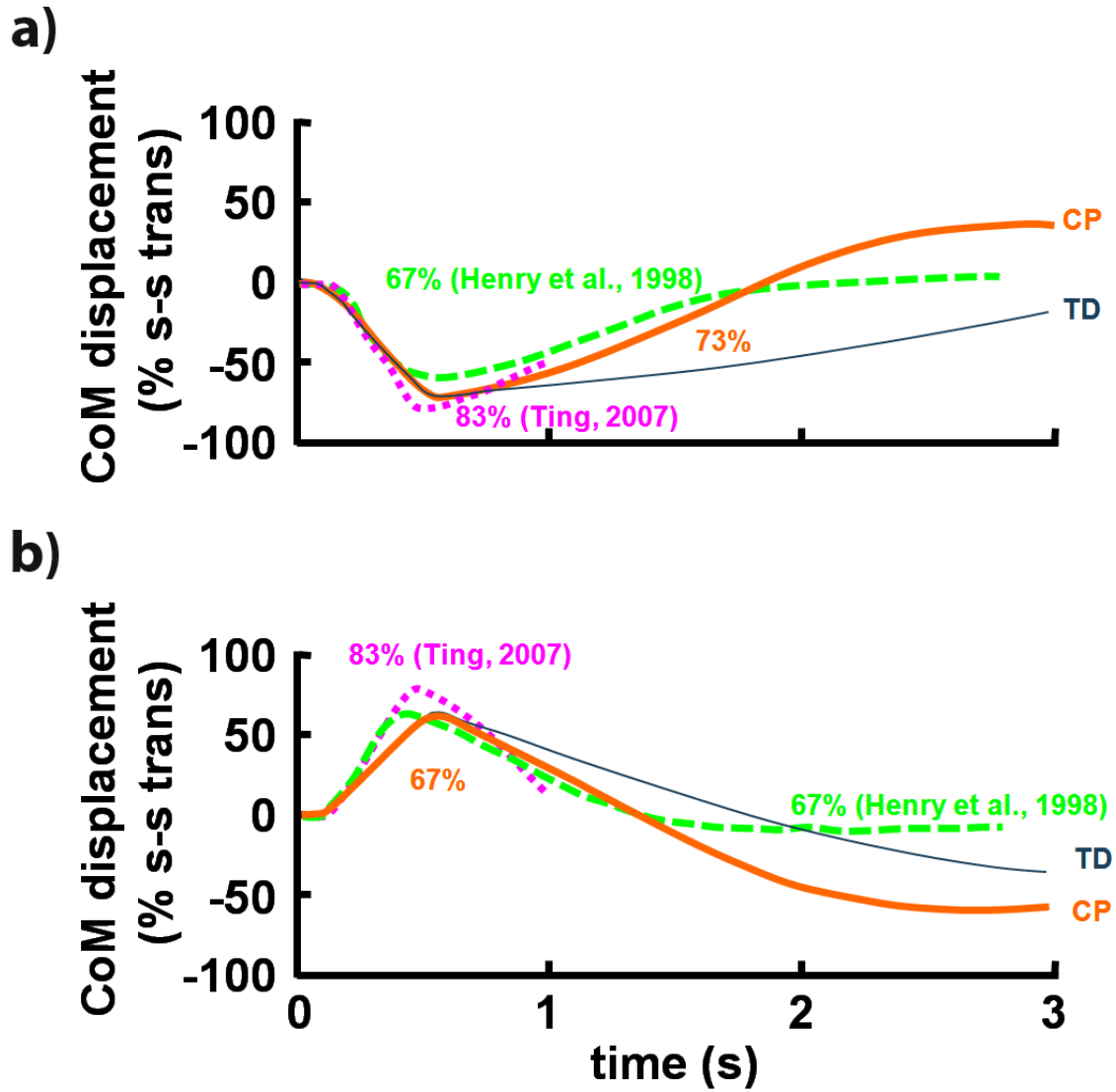


Figure 4. 5: Simulated center of mass (CoM) displacements for Typically Developing (thin solid line) and pre-surgical, spastic Cerebral Palsy (thick solid line) models with mild crouch compared with experimental CoM displacements of healthy adults with upright posture from Ting, 2007 (square dot line) and Henry, et al., 1998 (dash line) during a) anterior and b) posterior support-surface translations. To compare the balance recoveries across support-surface translation magnitudes, the CoM displacements were normalized by the support-surface translations in each study.

This study had several challenges and our findings should be interpreted within the context of modeling assumptions and capabilities to perform predictive simulations. First, we did not report our simulation results for different subject sizes. However, after uniformly scaling models to different sizes and support-surface translations, there is negligible to no differences in balance recovery due to model sizes. Moreover, Feldesman [131] showed that the ratios of femur to stature in children range between 8 to 18 had a very tight standard deviation (male [mean = 26.97, SD = 0.32] and female [mean = 27.11, SD = 0.46]) and the conclusions are unlikely to change according to Alonso, et al. [132] that found poor correlations between postural balance and the lower-limb length. Second, the balance recovery simulations used foot-ground contact modeling that depends upon parameters such as geometry, stiffness and dissipation. The stiffness and dissipation values were chosen based on material properties of skin (foot) and concrete (ground) materials. The same contact model parameters were used in each simulation, so observed differences are not the result of foot-ground differences. Third, although the controller design was not the focus of this study, the simulation was sensitive to stretch-reflex gains. Multiple optimizations seeded with random initial guesses were used to avoid local minima and determine control parameters for maintaining balance while tracking experimental whole-body CoM position and velocity [125]. Forth, we did not consider the stretch-reflex latency differences between muscles in this study. Alternatively, we focused on musculoskeletal geometry differences between muscle attachments and muscle moment arms to indicate a change in balance recovery for different postures, tendon locations, and stretch-reflex controller gains. Fifth, the difference between the experimental and simulated CoM (Figure 4.5) can be explained with the fact that the bio-inspired controller is not a full representation of the CNS including vestibular, visual, and proprioceptive sensory information; however, our controller uses

information about the current whole body CoM to adjust the stretch-reflex controller gains. In spite of this limitation, our results are consistent with the experimental balance recovery data [125, 130]. We believe these challenges do not influence the answers to the underlying question about what role the biarticular rectus femoris muscle plays in balance recovery before and after transfer surgery for subjects adopting mild and moderate crouched postures. Although the predicted stability margins may change if we made different modeling assumptions, the conclusions regarding the surgical alteration of the rectus femoris muscle's function having an influence on balance recovery are unlikely to change.

There are several opportunities for future work to address problems with validating our simulation results and further investigate biomechanical factors contributing to movement abnormalities. Concerning problems with validation, we were unable to find previous studies investigating support-surface perturbations in children with CP following rectus femoris transfer surgeries. Clinical studies [69, 70] have reported some data for this population before surgery; however, we were not able to compare our CoM displacements directly with these studies. Unfortunately, for functional tests (e.g., Gross Motor Function Measure), there is no clear indication of the link between balance recovery potential and changes in gross motor function over time. Importantly, it is unethical to separate, or decouple, spinal reflexes (as we have studied here) and supraspinal brain signals in human balance studies. Concerning further investigation, alternative treatment procedures such as distal transfer of the rectus femoris to the iliotibial band for stiff-knee gait and Achilles tendon lengthening for equinus gait may contribute additional insight into the influence of other biarticular muscles in postural control [92]. By using multidirectional (e.g., mediolateral, diagonal, rotational) support-surface perturbations as well as

ones with varying magnitudes and velocities, relationships between various biarticular muscles and different directions may be established.

The musculoskeletal modeling, neuromuscular control system design, and forward dynamic simulation in this study identified that distal transfer of the rectus femoris muscle to the insertion of the sartorius may change control of balance provided by the muscle path for patients with spastic CP. The pre-surgical simulations maintained balance following both anterior and posterior support-surface translations while post-surgical ones did not maintain balance following posterior translations. Balance recovery was also influenced by crouched postures where moderate crouch was better than mild crouch in helping recover from anterior support-surface translations but not necessarily posterior translations. These results indicate that rectus femoris plays an important role in postural response to support-surface translations. This study provides a foundation for the identification of other biomechanical factors contributing to movement abnormalities and potentially improve surgical and rehabilitation treatments for patients with neurological disorders.

CHAPTER 5

TASK-BASED CONTROL OF HUMAN MOVEMENT

5.1. Introduction

Motivated by the robotics research field, the synthesis of human motion control using task-level control commands can be addressed by merging approaches from biomechanics and robotics. A task-based controller using an operational space approach has been investigated in robotics field for many years [65, 133, 134]. Integration of task-based control with complex musculoskeletal models has a very promising outlook in biomechanics and rehabilitation fields. In this chapter, we utilize the operational space methodologies in muscle-driven simulations. In addition, we used the OpenSim/MATLAB interface that provides interaction between musculoskeletal models and task-level control systems via feedback control algorithms, which previously were not available.

In previous chapters, we discussed different algorithms to calculate joint torques and muscle excitations that generate motion. We discussed the computed muscle control (CMC) and how it is computationally expensive while it tracks every single joint and estimates muscle forces that produce that motion using static optimizations. In addition, we showed that a stretch-reflex controller was successful in rejecting the disturbances and in balancing our models of children with cerebral palsy. However, the stretch-reflex controller is limited when it comes to tracking more complex experimental kinematic data due to lack of the higher-level control signals required for complex human movement.

A task-based controller is a model-based controller that requires complete information about the musculoskeletal system dynamics through system Jacobian. By choosing a physiological task, this controller can model certain aspects of central nervous system (CNS)

behavior. Using the feedback from dynamic system states (positions and velocities) in physics-based simulations provides a robust, real-time interaction that is missing in other approaches such as feed-forward optimization approaches.

In this chapter, we formulated the basics of task-level control inside our s-function interfaces. In addition, we showed its application in a simple arm model balancing a pole example.

5.2. Musculoskeletal Dynamics

A task-based controller is a model-based controller, which utilizes precise model of multibody dynamics through the system Jacobian. A general multibody dynamical system can be described by a set of generalized coordinates, $q \in \mathbb{R}^n$ and the following system of n equations,

$$\tau = M(q)\ddot{q} + c(q, \dot{q}) + g(q),$$

where $\tau \in \mathbb{R}^n$ is the vector of generalized actuator forces (e.g. joint torques), $M(q) \in \mathbb{R}^{n \times n}$ is the system mass matrix, $c(q, \dot{q}) \in \mathbb{R}^n$ is the vector of centrifugal and Coriolis terms, and $g(q) \in \mathbb{R}^n$ is the vector of gravitational terms.

5.3. Task-based Control Formulation

In robotics, the operational space formulation [63] was first introduced to relate robot task space and force. This work utilizes operational space control strategies for controlling of complex musculoskeletal systems. A task can be defined as a desired movement of a body that is a function of generalized coordinates. Given the models described before, the task coordinates are $x = x(q)$, the task Jacobian can then be defined as:

$$J(q) = \frac{\partial x}{\partial q} \in \mathbb{R}^{m \times n}$$

The dynamic behavior in task space is calculated by projecting the Cartesian multi-body dynamics into the task space using a dynamically consistent inverse of the Jacobian.

$$\bar{J} = M^{-1} J^T (J M^{-1} J^T)^{-1}$$

The generalized actuator τ can be written as, $J^T f, f \in \mathfrak{R}^m$ is the task or operational space force. We can map the equation of motion onto the operational space using the following:

$$M\ddot{q} + b + g = \tau \xrightarrow{J^T} f = \Lambda\ddot{x} + \mu + p$$

where, the operational space mass matrix,

$$\Lambda(q) = (J M^{-1} J^T)^{-1},$$

the operational space centrifugal and Coriolis vector,

$$\mu(q, \dot{q}) = \Lambda J M^{-1} b - \Lambda \dot{J} \dot{q},$$

the operational space gravity vector,

$$p(q) = \Lambda J M^{-1} g.$$

The dimension of task is always less than or equal to the operating space. When this dimension is strictly less than the dimension of the operating space, the system is called redundant with respect to the task. In this case the null space of the task is linked with the posture space of the motion [133] that complements the task term through the following decomposition equation.

$$\tau = \tau_{task} + \tau_{posture} = J^T f + (I - J^T \bar{J}^T) \tau_o = J^T f + N^T \tau_o,$$

$$\tau = J^T (\Lambda\ddot{x} + \mu + p) + N^T \tau_o$$

where, τ_o , is an arbitrary generalized force and

$$N(q)^T = I - J^T \Lambda J M^{-1} = I - J^T \bar{J}^T.$$

Furthermore, a control law needs to be incorporated in the model. We used PID control law to create the control response f^* as the input to our system:

$$f^* = K_p(x - x_o) + K_v\dot{x},$$

and

$$f = \hat{A}f^* + \hat{\mu} + \hat{p}$$

where $\hat{\cdot}$ represents estimates of operational space dynamic properties. Placing this into the equation of motion, we obtain:

$$\tau = J^T(\hat{A}f^* + \hat{\mu} + \hat{p}) + \hat{N}^T\tau_o$$

To solve for muscle forces, we used static optimization to minimize muscle activations squared, $\min\|\tilde{a}\|^2$, that satisfy the inequality condition of $0 \leq a_i \leq 1$.

Figure 5.1 shows the task-level algorithm that uses states as inputs and output the neural excitations required for performing a task such as balance control. A task-level neuromuscular feedback controller includes (a) Task-level command input states. (b) Task space feedback linearization using estimation of the dynamic parameters. (c) Task space feedback loop for tracking task space commands. (d) Posture space control input. (e) Static optimization of muscle activations. (f) Activation feedback loop for tracking optimization solution. (g) Output as neural excitations. Optimization algorithm can be used to tune the gains for tracking controller.

A task-level neuromuscular feedback controller enables predictive simulations of human movement by defining the task commands that represents human actions. Next, it is a simple example of a task level command that is used to show application of closed-loop task-based controller in balancing a pole while moving the hand in desired positions in space.

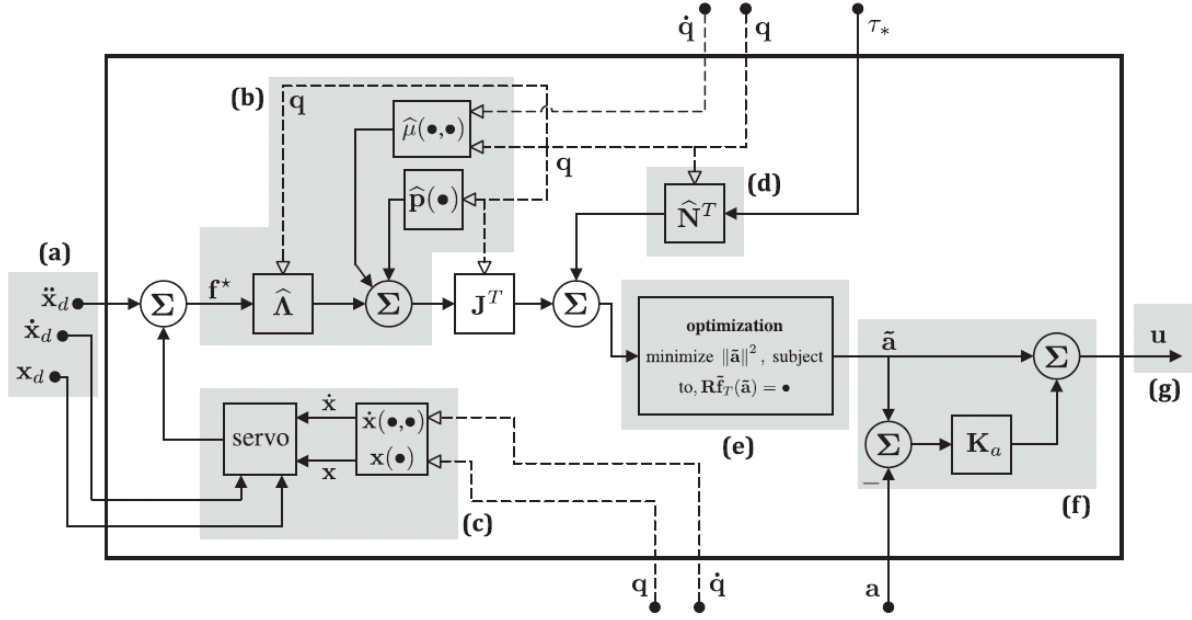


Figure 5. 1: A task-level neuromuscular feedback controller (a) Task-level command input states. (b) Task space feedback linearization using estimation of the dynamic parameters. (c) Task space feedback loop for tracking task space commands. (d) Posture space control input. (e) Static optimization of muscle activations. (f) Activation feedback loop for tracking optimization solution. (g) Output as neural excitations. (Image courtesy: Goldfarb, 2014).

5.4. Task-based Control of Movement (Arm-Pole Control Example)

Using OpenSim/MATLAB interface (Figure 5.2), we implement the tasked-based controller on a human arm with 3 degrees of freedom including the pole and 6 muscle actuators (Figure 5.3). Task command inputs (hand positions and pole rotation) were specified. The hand was driven to three locations inside the working space of the arm (pink spheres) while balancing a pole (to keep the pole from falling). There was no actuator on the hand-pole joint. The pole is modeled as a cylinder with distributed mass of 1kg and height of 0.25m, and is connected to the hand through a pin joint with 1 degree of freedom. Disturbance is introduced to the pole as a

force vector to the pole center of mass at a certain simulation time (green arrow). The static optimization was implemented inside of the S-function interface using the IPOPT optimizer to minimize the muscles activations-squared and to solve for muscle controls.

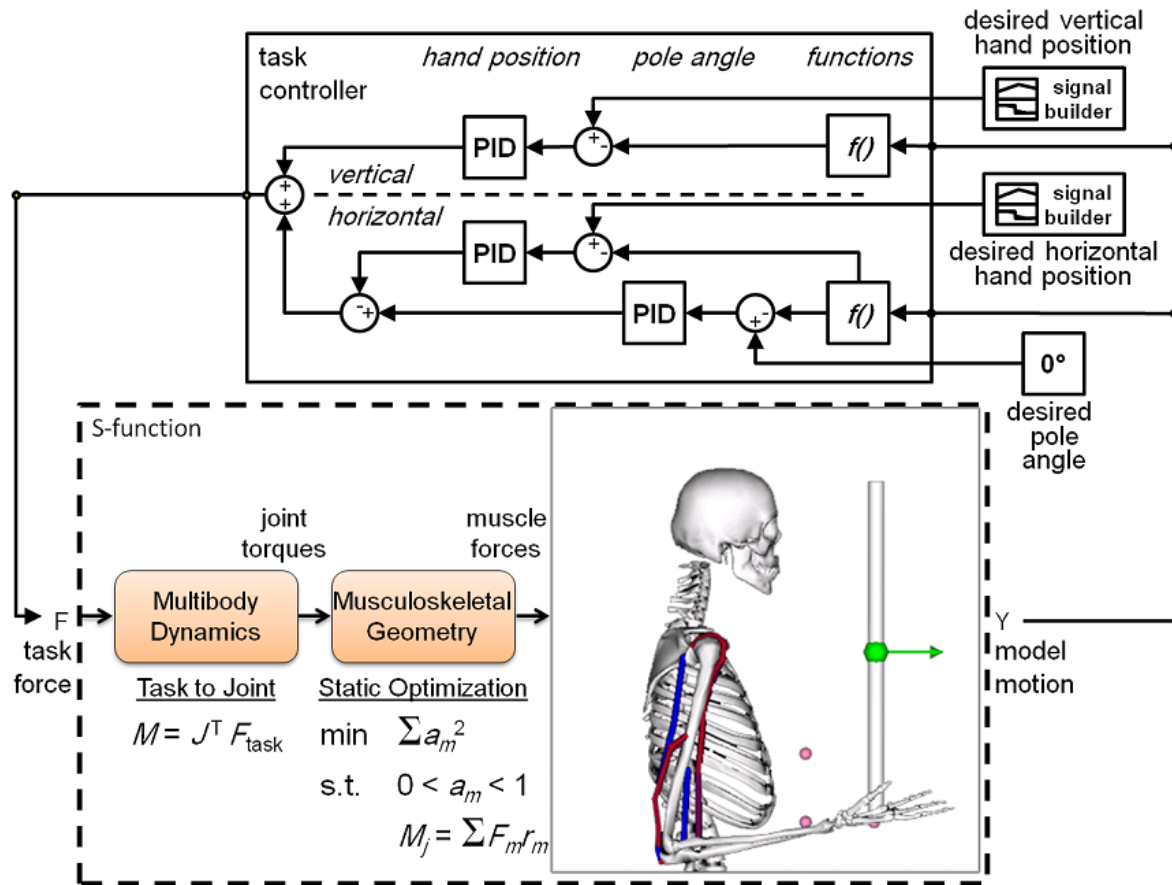


Figure 5. 2: Schematic model of the task-based neuromuscular controller implemented through the OpenSim/MATLAB interface. The interface (dotted lines in a) calls the OpenSim arm model and then the closed-loop task force is calculated as an input to the interface. The task force then was used to calculate the actuation force (joint torques) using multi-body dynamics and later the muscle excitations using static optimization. All these steps are implemented inside the interface by calling OpenSim libraries.

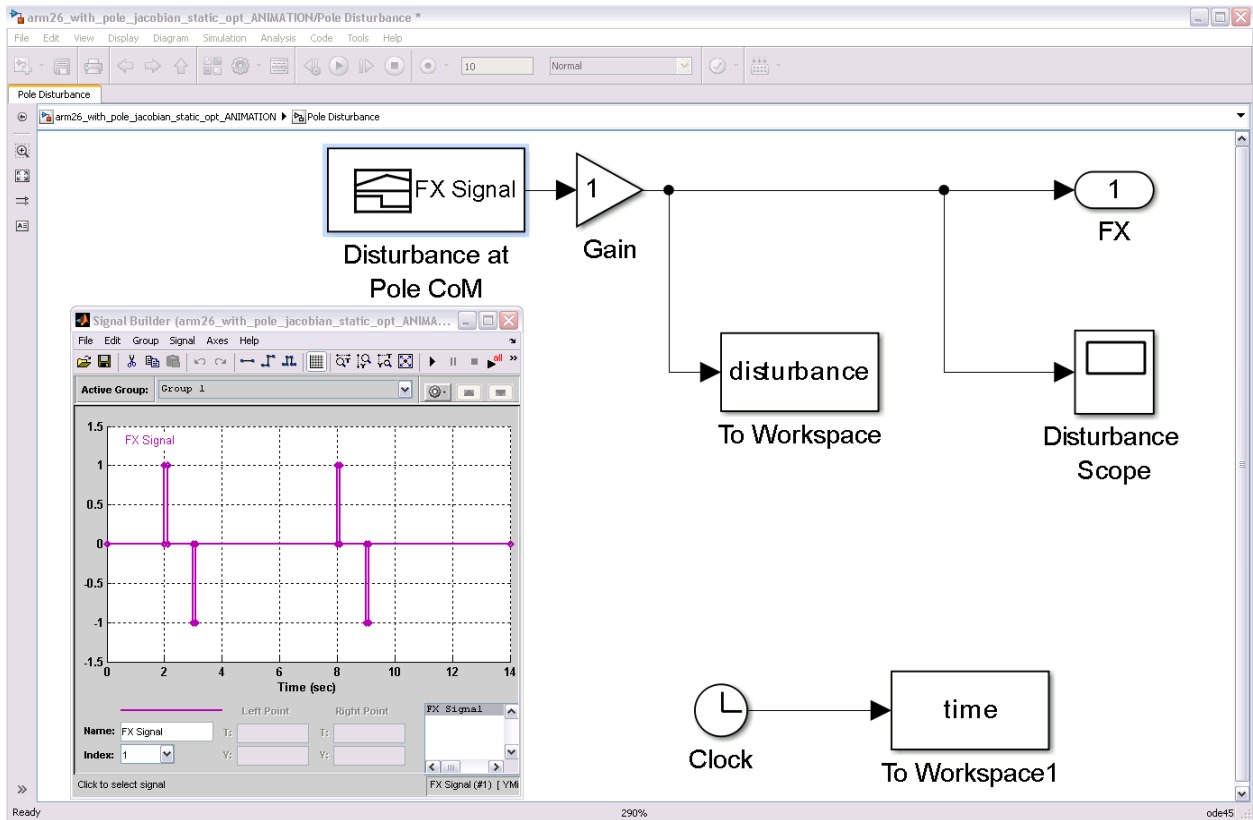


Figure 5. 3: Disturbance was introduced to the center of mass of the pole to test the controller robustness in presence of external perturbation.

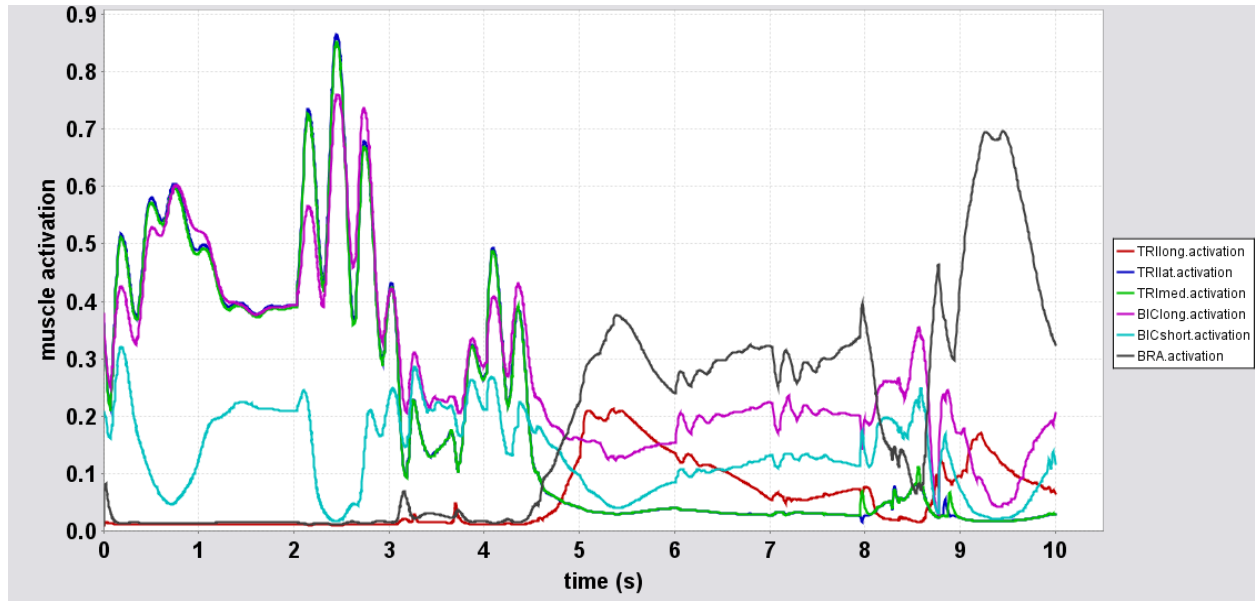


Figure 5. 4: The resultant muscle activations calculated through the task-based controller implemented during 10 second simulations of hand position control and pole balancing tasks.

CHAPTER 6

PRIORITIZED TASK-BASED CONTROL OF MOVEMENT WITH SUPPORTING CONTACTS USING OPENSIM AND MATLAB

In this chapter, we used task prioritization to synthesize whole-body human balance. In contrary to most robotic manipulators which use open chain (serial kinematics) formulations, and similar to humanoid robots, our models uses free-floating formulation which permits the system to detach from the ground [135]. Gravity forces exerted on our musculoskeletal models push the models to the ground. Foot contact modeling was used to generate the ground reaction forces (GRF) required for moving the model in different directions. Furthermore, motion of our musculoskeletal system is not only defined by body and joint positions and velocities in a body reference frame but also by relative positions and orientations with respect to the moving (inertia) reference frame.

Multiple operational tasks are required for controlling a complex human movement. In robotics, the operational space formulation [63] was first introduced to relate robot task space and force. This work utilizes operational space control strategies to implement control algorithms for complex musculoskeletal models. Task priority formulations have been researched at the inverse kinematics level [136-138] and later at the dynamics level [63, 135, 139]. In this chapter, we go through implementation of prioritized task-based control formulation for free-floating support-consistent musculoskeletal models. Free-floating control of robots was first introduced in Cartesian space by Arai and Tachi [1]. The formulation of free-floating Jacobian was extended to include dynamic interaction of generalized forces of the base segment ([22], [7] and [17]). Later, Sentis and Khatib [135] introduced formulation of free-floating dynamics in an operational space. In this chapter, we implemented prioritized, multi-task, support-consistent

control of human balance in operational space using related robotics' literatures [67, 135, 139, 140].

6.1. Musculoskeletal Model

We started with a generic OpenSim model with 23 DOF and 92 muscle actuators. We used OpenSim's five Hunt and Crossley contact spheres under each foot to model the foot-ground contact modeling [15]. The models had 10 body segments; subtalar (2x), talus (2x), tibia (2x), femur (2x), pelvis and torso. The subtalar, ankle and knee joints were modeled as revolute joints while hip and lumbar joints were modeled as ball joints. The pelvis body (free-floating base) has 6 degrees of freedom in space (3 translational and 3 rotational).

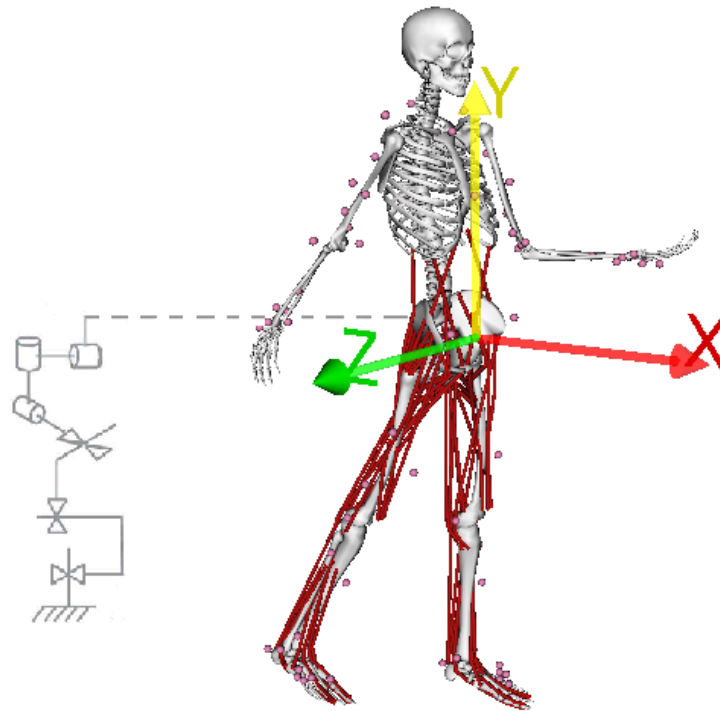


Figure 6. 1: Free-floating musculoskeletal model. Pelvis body (free-floating base) has 6 degrees of freedom in space (3 translational and 3 rotational).

6.2. Free-floating Dynamic Formulation in Operational Space

The free-floating human dynamics under supporting contact can be presented as follow:

$$M_{sys} \begin{pmatrix} \dot{v}_b \\ \ddot{q} \end{pmatrix} + c_{sys} + g_{sys} + J_s^T F_r = \begin{pmatrix} 0 \\ \Gamma \end{pmatrix}$$

where M_{sys} is $n \times n$ mass matrix, b_{sys} is the $n \times 1$ Coriolis/centripetal and g_{sys} is $n \times 1$ gravity component, $\dot{v}_b$ is a 6×1 vector of base body (pelvis) accelerations (three translational and three rotational), $\ddot{q}$ is the $(n-6) \times 1$ vector of joint accelerations, J_s^T is the transpose of the Jacobian of all the supporting links and F_r is the summation of all the foot-ground contact forces and moments and Γ is the $(n-6) \times 1$ vector of joint torques (n is number of model's coordinates and 6 is number of base (pelvis) degrees of freedoms).

The resultant foot-ground contact forces are related to the forces and moments acting of the model's coordinates, Γ_s through the following equation:

$$\Gamma_s = J_s^T F_r \in R^{(n)}$$

The support Jacobian, J_s is consisted of the base Jacobian $J_{s(b)}$ and the $J_{s(r)}$ and $J_{s(l)}$. The foot support constraint relates the base joint displacements to the feet displacements.

The Simbody's frame Jacobian was used to formulate this relationship. The frame Jacobian calculates generalized forces Γ , resulting from a set of spatial task forces in the task frame fixed on the body. Each task force was measured and expressed in the ground frame. The rotational part of a Jacobian is the same for any frame fixed to the same body, so the frame's origin point need to be specified on the body. The system Jacobian is a Frame Jacobian such that the task frames are the same as body frames [99].

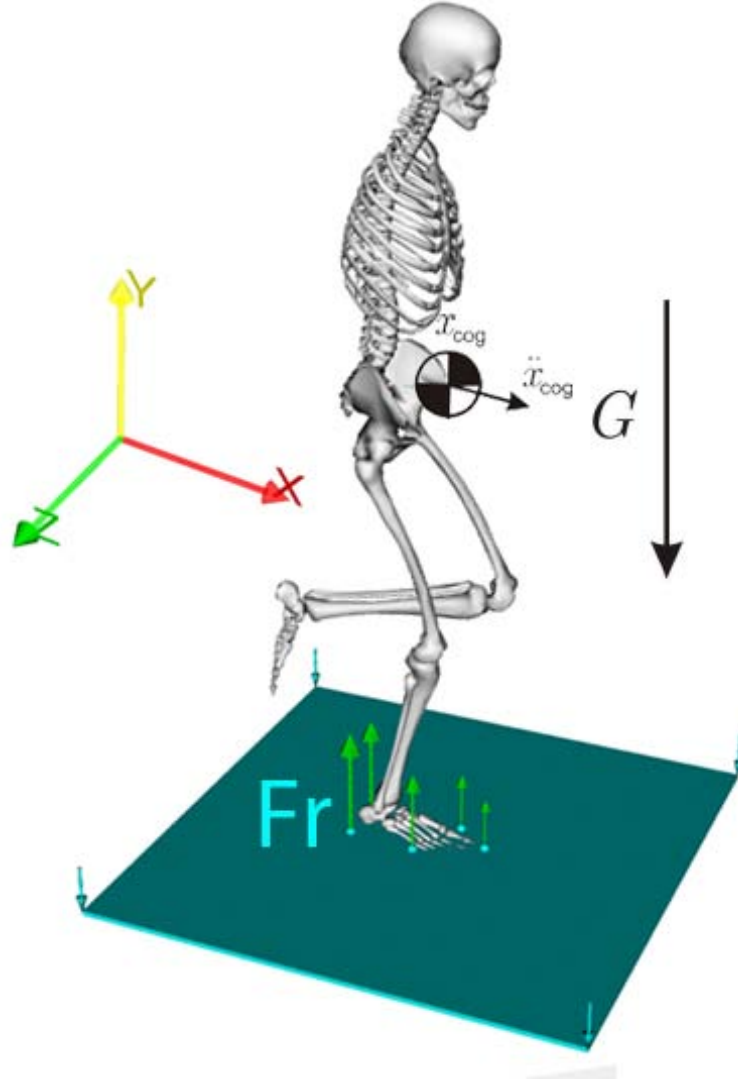


Figure 6. 2: Resultant ground reaction forces on the foot are due to gravity and CoG accelerations.

6.2.1. *Supporting Contact Dynamic Formulation in Operational Space*

The foot-ground contact was modeled in OpenSim using the Hunt and Crossley dissipation model [141] and the Stribek friction model. We used five contact points to represent the foot-ground interaction. For each contact element, a force composed of stiffness, dissipation, and friction was produced [99]:

$$F_r = f_{stiffness} + f_{dissipation} + f_{friction}$$

The contacts were implemented in OpenSim model files as a force set. The following code shows the sample .xml script and the contact parameters (stiffness, dissipation, static, dynamic and viscous friction, and transition velocity) that were used in our simulations [25]:

```
<HuntCrossleyForce name="contact_force">
  <!--Material properties.-->
  <HuntCrossleyForce::ContactParametersSet name="contact_parameters">
    <objects>
      <HuntCrossleyForce::ContactParameters>
        <!--Names of geometry objects affected by these parameters.-->
        <geometry>ground_contact toe_r midfoot_medial_r
          midfoot_lateral_r heel_medial_r heel_lateral_r</geometry>
        <stiffness>20000000</stiffness>
        <dissipation>0.95</dissipation>
        <static_friction>0.9</static_friction>
        <dynamic_friction>0.9</dynamic_friction>
        <viscous_friction>0.6</viscous_friction>
      </HuntCrossleyForce::ContactParameters>
    </objects>
    <groups />
  </HuntCrossleyForce::ContactParametersSet>
  <!--Slip velocity (creep) at which peak static friction occurs.-->
  <transition_velocity>0.9</transition_velocity>
</HuntCrossleyForce>
```

Figure 6. 2: Foot-ground contact modeled as Hunt Crossley Forces inside the OpenSim’s model file. Five contact points (two on each side of the heel and two on each side of the metatarsal bones and one on the tip of the toes) were defined according to Anderson and Pandy, 1999, to represent foot-ground contact.

Foot-ground contact forces were transformed into generalized forces that actuate the joints. Balance was achieved through controlling internal forces to keep the foot on ground with no

relative movement with respect to the ground contact point. As a result, relative velocities at the contact points are equal to zero ($v_b=0$ and $\dot{v}_b=0$).

The actuation matrix $S_q=[0_{(n-6) \times 6} \quad I_{(n-6) \times (n-6)}]$ was defined to select the model's active coordinates and the base joint. The actuation matrix was equaled to the right hand side of the equation, $\begin{pmatrix} 0_{6 \times 1} \\ \Gamma \end{pmatrix} = S^T \Gamma$, plugging it into the equation of motion and using the constraint $\dot{v}_b = 0$, we have:

$$F_r = \bar{J}_s^T S^T \Gamma - \bar{J}_s^T (c+g) + A_s \dot{J}_s \begin{pmatrix} v_b \\ \dot{q} \end{pmatrix}.$$

where,

$$A_s = (J_s M^{-1} J_s^T)^{-1},$$

is the inertia matrix at the supporting links, and

$$\bar{J}_s = M^{-1} J_s^T A_s,$$

is a dynamically consistent generalized inverse of J_s . Based on (Khatib 1987), the dynamically consistent null space of a supporting contact J_s can be written as:

$$N_s = I - \bar{J}_s J_s,$$

and defines a generalized space of motion that has no force coupling effects on the supporting links.

6.2.2. Kinematic Constraints in Operational Space

The base position in our musculoskeletal model can be derived from the position of actuated joints alone while our system is constrained by the supporting ground [142]. This means that the base and joints velocities are not arbitrary but that they are defined in the null space of the support Jacobian. Constrained joint velocities can be derived as:

$$\dot{q}^* = S N_s \begin{pmatrix} v_b \\ \dot{q} \end{pmatrix}$$

The matrix SN_s is a full rank in the single support stance and so the $\dot{q}^*$ can take arbitrary values.

The support-consistent generalized inverse of SN_s that satisfies the constraint $v_b = 0$ can be derived from the following equation:

$$\overline{SN_s} = M^{-1} (SN_s)^T (\phi^*)^+,$$

where, during the single support stance (SN_s is full rank) the pseudo-inverse(+) becomes an inverse and,

$$\phi^* = SN_s M^{-1} (SN_s)^T,$$

is the constrained projection of our model inertia matrix inverse, M^{-1} .

Using above equations, the support consistent reduced Jacobian (J^*) that shows the dependency of the base velocity on other joints velocities can be defined as:

$$J^* = J \overline{SN_s}.$$

6.2.3. Operational Space Control of CoG

Balance was achieved by defining a task force for controlling the CoG horizontal and vertical positions. Assuming the support constraint maintained foot-ground contact, the CoG position and its Jacobian can be defined as:

$$x = \frac{1}{M} \sum_{i=1}^n m_i x_{cog(i)}, \quad J = \frac{1}{M} \sum_{i=1}^n m_i J_{cog(i)},$$

where M is the total musculoskeletal body mass, m is the mass of the i^{th} body, and $x_{cog(i)}$ is the center of mass of the i^{th} body, and $J_{cog(i)}$ is the Jacobian of the same body. Using the support consistent reduced Jacobian for CoG is:

$$J_{cog}^* = J_{cog} \overline{SN_s},$$

and therefore, the support consistent full Jacobian can be derived as:

$$J_{cog|s} = JN_s$$

where the subscript cog|s shows that the task Jacobian (J_{cog}) is projected onto the support consistent constraints space. The dynamically consistent generalized inverse of the Jacobian is then defined as:

$$\bar{J}_{cog|s} = M^{-1} J_{cog|s}^T \Lambda_{cog|s}$$

where

$$\Lambda_{cog|s} = (J_{cog|s} M^{-1} J_{cog|s}^T)^{-1},$$

is a task space inertia matrix and is used to form the task space equation of motion:

$$\Lambda_{cog|s} \ddot{x} + \mu_{cog|s} + p_{cog|s} + F_c = \bar{J}_{cog|s}^T (SN_s)^T \Gamma,$$

where $\mu_{cog|s}$ is the Coriolis/centrifugal and $p_{cog|s}$ represents the gravitational terms in the task space of COG under supporting constraints.

$$\mu_{cog|s} = \bar{J}_{cog|s}^T b - (\Lambda_{cog|s} \dot{J}_{cog|s} + \bar{J}_{cog|s}^T J_s^T \Lambda_s \dot{J}_s) \begin{pmatrix} v_b \\ \dot{q} \end{pmatrix}$$

$$p_{cog|s} = \bar{J}_{cog|s}^T g.$$

The operational space control torque vector is then formed as:

$$\Gamma = J_{cog}^* F_{cog},$$

where F is a vector of CoG task forces that is formed as follow:

$$F_{cog} = \Lambda_{cog|s} a^{ref} + \mu_{cog|s} + p_{cog|s},$$

and a^{ref} is a reference CoG acceleration vector.

6.3. Tasks prioritization in Operational Space

In this section we explain control algorithms for multiple operational tasks using prioritized control strategies. This allows controlling of complex human movement while ensuring that critical tasks (e.g. CoG control) are accomplished without the interference of lower priority tasks (hand movement, swing leg position and torso orientation).

Task prioritization, multi-task control of robots have been studied in the robotics field for many years [138, 139, 143]. Prioritized multi-Task control, which considering the supporting constraints, was introduced into the operational space later [142]. We implemented and extended the work by Sentis, 2007, in this section.

Multi-task control can be used to simplify the syntheses of human movement by identifying and sequencing individual tasks that explain the more complex motion. For example, in balance control of a musculoskeletal model, the first task can be defined to control the CoG in three directions (x, y, z), while keeping the CoG over the base of support boundaries (region of stability). The lower priority subtasks, can be then defined (e.g. swing leg movement, torso orientation). Based on the complexity of the desired movement, other tasks may need to be defined. In generalized form, we can write the torque structure that controls N arbitrary tasks as:

$$\Gamma = \Gamma_1 + \Gamma_{2|prec(2)} + \Gamma_{3|prec(3)} + \Gamma_{N|prec(N)} = \sum_{k=1}^N \Gamma_{k|prec(k)}$$

where, k-th task operates in the null space of all higher priority tasks and

$$\Gamma_{k|prec(k)} = N_{prec(k)}^T \Gamma_k,$$

and $N_{prec(k)}$ is the null space of all higher priority tasks to the kth task.

$$N_{prec(k)} = I - \sum_{i=1}^{k-1} \bar{J}_{i|p(i)} J_{i|p(i)}$$

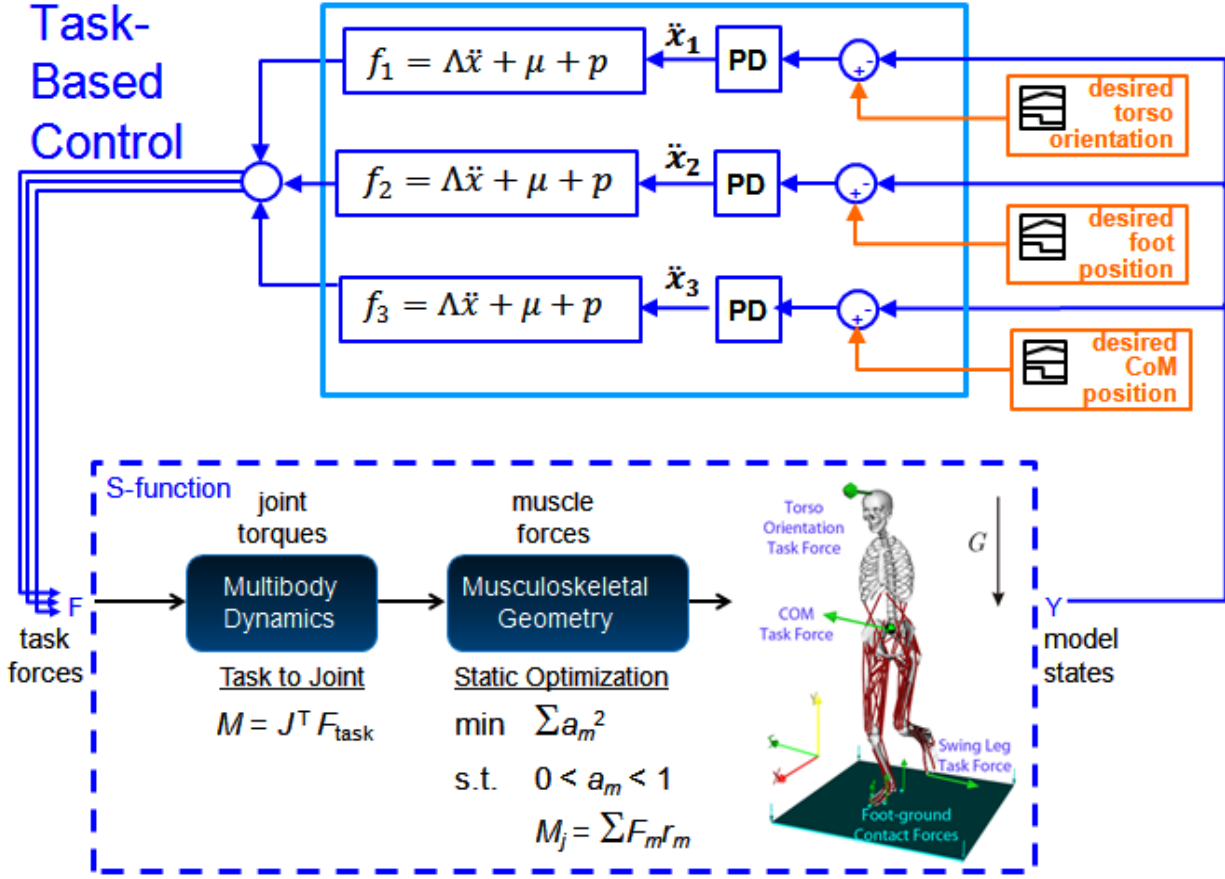


Figure 6. 3: Schematic of the task-based neuromuscular controller implemented through OpenSim/MATLAB interface in the Simulink environment. The interface (dotted lines in a) calls the OpenSim 2392 gait model, initial controls, initial states files, and perform a forward dynamics simulation with a closed-loop task-based feedback control in operational space using the defined task forces required for performing a sequence of tasks. The task force is used to calculate the actuation force (joint torques) through multi-body system Jacobian transpose. Muscle excitations are calculated using static optimization.

Based on the torque structure formula above, we can derive a general operational space equation for the torques:

$$\Gamma = J_1^{*T} F_1 + J_{2l}^{*T} F_{2l} + \dots + J_{N|prec(N)}^{*T} F_{N|prec(N)},$$

where, the prioritized Jacobian related to the k^{th} task is :

$$J_{k|prec(k)}^* = J_k^* N_{prec(k)}^*,$$

and J_k^* is the constrained Jacobian and $N_{prec(k)}^*$ is the prioritized null space of all tasks preceding the k^{th} one.

To solve for muscle forces, we used static optimization to minimize muscle activations squared, $\min \|\tilde{a}\|^2$ that satisfy the inequality condition of $0 \leq a_i \leq 1$.

6.4. Task-based Control of Movement (Whole-body Control Example)

We implemented the prioritized multi-tasks controller using the described musculoskeletal model with 23 DOF and 92 muscle-actuators in OpenSim and MATLAB interface (Figure 6.5, 6.6). We defined 3 different tasks for this specific example. The task's order is shown in the following table.

Table 6.1: Prioritization of primitive tasks required for human balance

Task Primitive	Priority Level
CoG Balance	1
Swing Leg	2
Torso Orientation	3

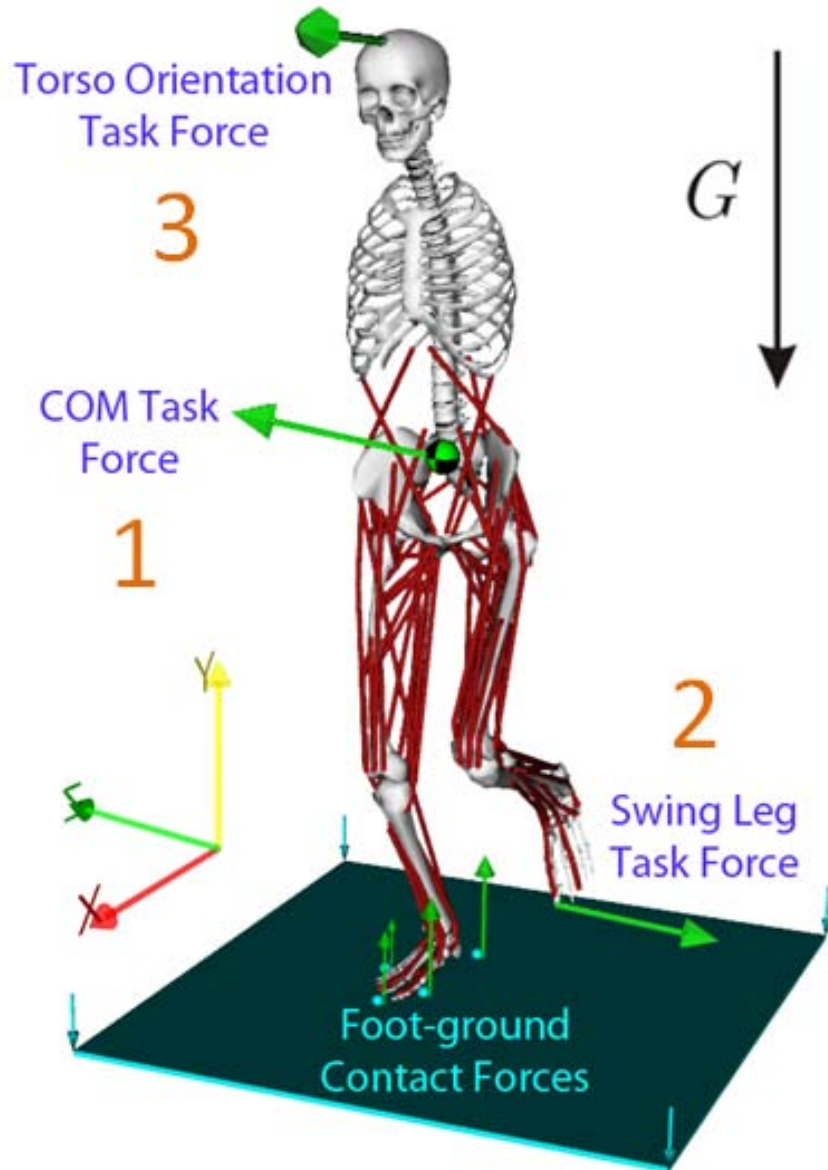


Figure 6. 4: Different tasks defined for the single-leg balance example. All these tasks defined within the supporting contact constraint. Three task force vectors were defined. The first task was exerted on the whole body center of mass. The second task force was exerted on the swing leg foot and the third task force was exerted on our model head. The arrows on the foot-ground contacts are not the task forces. They show the foot-ground contact forces due to gravitational forces. The resultant ground reaction forces are summation of all those five contact forces and are applied to the center of pressure of the model.

The primary task was to keep the models balance on one foot while moving from point 1 to point 4 in numeric order (Figure 6.7a, pink spheres and Figure 6.8). The CoG control task was defined within the supporting contact constraint. This means that the supporting leg's foot-ground contact was enforced during these tasks. However, the individual contact points can still leave the ground depending on the defined task. The highest priority task (CoG) is then defined by keeping the whole-body CoG projection over the base of support midpoint (middle point between heel and toes) as the equilibrium point for balance. The task force F_{cog} (derived priviously) was exerted on the CoG of our model and was tuned based on the acceleration error between the desired CoG and the actual simulated CoG ($a^{ref} = \ddot{x}$). The second task was defined on the swing leg (left leg) to keep the foot in a fixed position in space while keeping the model from falling and performing the main task. The second task is implemented in the null space of tasks prior to this task as the following equations so that it does not interfere with the higher priority tasks.

$$\Gamma = J_1^{*T} F_1 + J_{2|1}^{*T} F_{2|1},$$

$$J_{2|1}^* = J_2^* N_1^*,$$

where subscript (1) shows the first task (CoG) and subscript (2) is the second task (swing leg).

We defined the third task as upper body (torso) orientations in the ground reference frame. We defined the desired torso orientation with zero rotations in the x, y, and z coordinates to keep the torso upright. The compound torque then can be written as:

$$\Gamma = J_1^{*T} F_1 + J_{2|1}^{*T} F_{2|1} + J_{3|2}^{*T} F_{3|2},$$

$$J_{2|1}^* = J_2^* N_1^*,$$

$$J_{3|2}^* = J_3^* N_2^*,$$

where subscript (1) shows the first task (CoG) and subscript (2) is the second task (swing leg) and the subscript (3) is the third task (torso orientation). The individual right leg (stance leg) torques (N.m) during an 8-second simulation required for performing the mentioned tasks in this example is shown in Figure 6.7b.

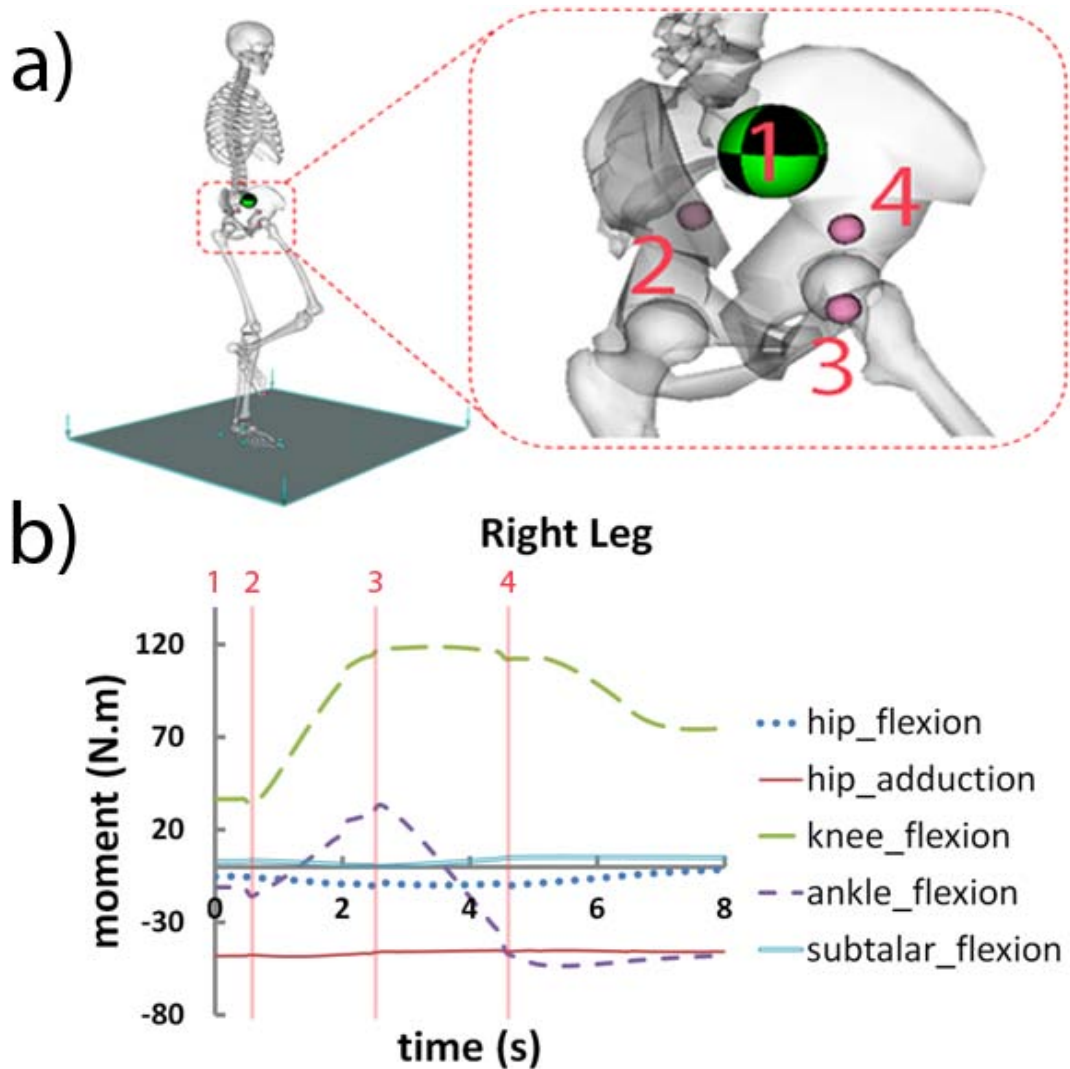


Figure 6. 5: a) The desired whole-body CoM trajectory defined for the musculoskeletal system through the points (1-4). The model is suppose to go through these point while maintaining balance on one foot. b) Right leg (stance leg) joint torques required to perform the assigned task.

The static optimization was implemented inside the S-function interface using the IPOPT optimizer to minimize the muscle activations squared, $\min \sum a_m^2$ and to solve for muscle controls that satisfy the inequality condition of $0 \leq a_m \leq 1$.

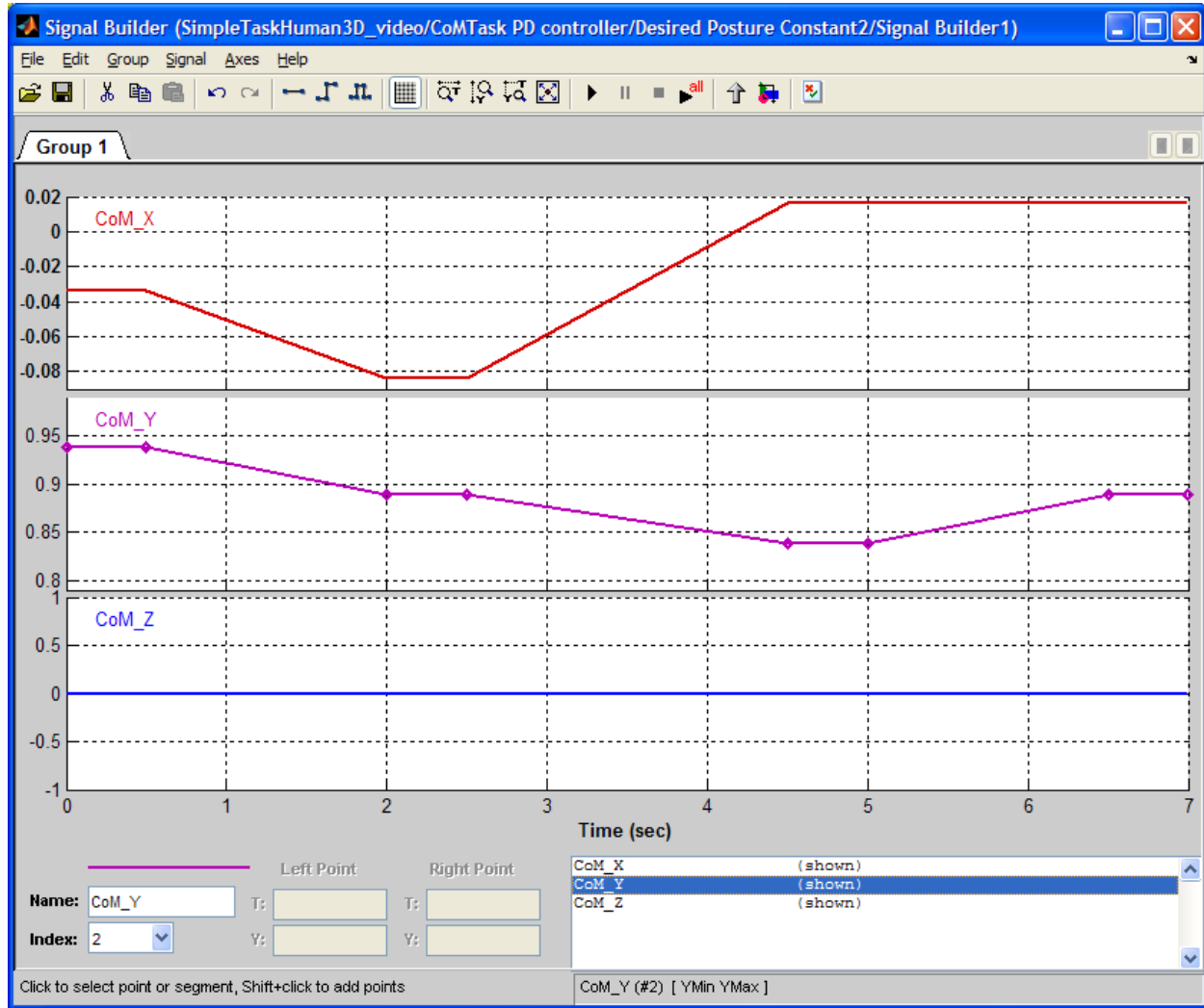


Figure 6. 6: The reference whole-body CoM path (x, y and z), was defined in our balance control example inside Simulink.

CHAPTER 7

SYNTHESIS OF SUBJECT-SPECIFIC TASK-LEVEL MOTIONS FOR PREDICTIVE SIMULATIONS OF BALANCE RECOVERY

A scientific framework developed in the previous chapters is needed in combination with experimental data to validate the simulation results and synthesize balance recovery data and predict functional outcomes. Here, experimental balance recovery data from normal healthy adults were used to investigate underlying neural strategies for balance control.

The prioritized task-based control algorithm developed in previous chapter, can not only be used as a controller to track specific tasks or end-effector positions but more importantly can be used to track a relationship between different bodies' movement. Series of these relationships can be used to define the underlying neural strategies that form a motion. By defining these relationships, one can use task-based controllers to synthesize new motions. In this chapter, we first discussed the experimental balance data as a reference to validate our simulations. We then introduced algorithms using surrogate model surfaces to relate the positions of stance leg, swing leg and torso to changes in whole-body center of mass. We then described how we imported and utilized subject-specific data in our OpenSim/MATLAB platform to implement the predictive simulations of human balance. Lastly, we described how we tested these relationships with experimental kinematics data from two different subjects' balance recovery data.

These predictive simulations enable a more rational basis for development of treatment planning and balance recovery in patients with neurological disorders.

7.1. Experimental Balance Recovery Setup

The balance recovery data was collected by our collaborators at University of Western Australia and Liverpool John Moores and served as the basis for validation of the results of our task-based control simulations of balance recovery. The state of the art CAREN (Computer Assisted Rehabilitation Environment) system (MotekMedical, Amsterdam, The Netherlands) was used to record the kinematics and muscle activities of the two healthy young adult subjects. CAREN [144] is consisted of a software driven six degrees of freedom movable platform with a complete motion analysis system including a Vicon Nexus system with sixteen infrared cameras, force platform and electromyography (Figure 7.1). Two young healthy adults, a female (subject 1; age 25; height 1.72 m; mass 68 kg) and a male (subject 2; age 25; height 1.79 m; mass 84.5 kg) were recruited for the balance experiment. Data including, 3D marker trajectories, ground reaction forces (GRF) (Kinematic sample rate, 250 Hz, GRF 2,000 Hz) and surface electromyography (EMG 2,000 Hz) of 8 muscles were collected for each subject. Muscles studied included Gluteus Maximus (GMAX), Gluteus Medius (GMED) Rectus Femoris (RF), Vastus Lateralis (VL), Vastus Medialis (VM), Bicep Femoris (BF), Semitendinosus (ST), and Medial Gastrocnemius (GASMED).

The CAREN platform was translated in anterior-posterior directions for 6cm and 12cm with maximum viable working speed (40cm/s). The subjects stand on the platform with a single leg. During single leg-support, the contra-lateral leg was lifted a minimum of 10 cm from the floor (90 degree knee flexion). Perturbation was triggered randomly between 1 and 3 seconds after the trial has been initiated. Instruction to participants for all trials was the same. Participants were not given a familiarization session. Participants were not allowed to use their hands for balance (arms crossed on the chest).



Figure 7. 1: CAREN System (MotekMedical), single-leg support setup.

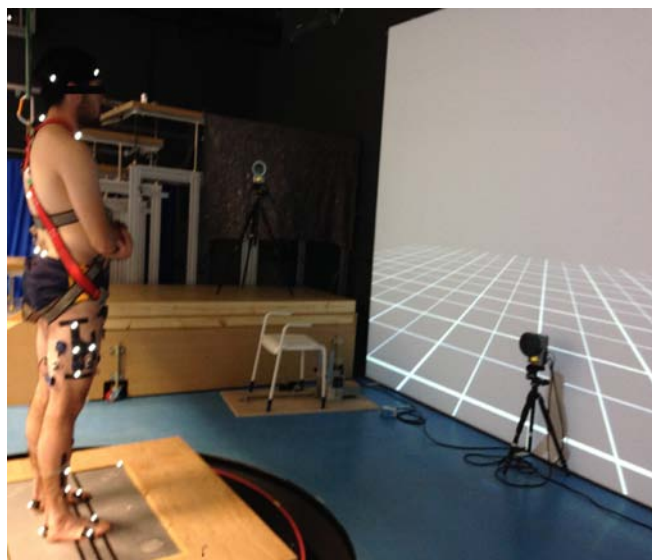


Figure 7. 2: Static pose for the second subject shows placement of reflective markers and the force plate setup on the CAREN platform.

7.2. Processing and Analyzing Experimental Balance Recovery Data

Synthesizing the motion based on experimental data is a multi-step process. The experimental data first need to be converted and imported into the OpenSim software. After scaling the models and running inverse kinematics analyses, the experimental data is used as a reference for our controller design.

Multiple steps were taken to process the experimental kinematic data, muscle activities and ground reaction forces. The output files from the Vicon Nexus system that contained the reflective marker trajectory data were in a C3D format. The C3D format is a binary file format and it has been used in Biomechanics and Gait Analysis frequently to record 3D motion data. In order to import the C3D files into the OpenSim, some pre-processing needs to be done. We first used the freely available b-tk (biomechanics toolkit) to read the C3D files in MATLAB workspace. The b-tk toolkit converts C3D files to data structures in MATLAB that contain all the information about the markers and forces. Next, the marker's data need to be first filtered and then transformed to the OpenSim global coordinate convention and then be written in a file format readable by OpenSim (.trc). The .trc (Track Row Column) file format was created by Motion Analysis Corporation to specify the positions of markers placed on a subject at different times during a motion capture trial. Similarly, for forces that are recorded from the force plates, they need to be transformed from the lab to OpenSim global coordinate and then filtered. Additionally, the center of pressure was calculated and then those forces were associated with the bodies in the model files. Last, the forces were written into the OpenSim motion file (.mot). The experimental data lab coordinates were rotated to match OpenSim coordinate convention (x- anterior/posterior, y- superior/inferior and z- medial/lateral).

After preprocessing the C3D files and importing the marker data and ground reaction forces into the OpenSim, we used this information to first scale our models using the scale tool and then obtain the experimental motion using the inverse kinematics tool in OpenSim. Please see chapter 3 for clarification about the scale tool and the inverse kinematics tools. The inverse kinematics tool derives the joint angles from the experimental marker data that accurately match the experimental movement. In order to get each body's center of mass position and velocity, we ran the Body Kinematics Analyses in OpenSim using the motion obtained from inverse kinematics. The Body Kinematics outputs linear and rotational position, velocities and accelerations of each body during the entire simulation. Body kinematics was used in the following section to derive the relationship between the positions of the stance leg, the swing leg and the torso and their contribution to changes in whole-body center of mass.

7.3. Simulation Pipeline for Synthesizing Subject-specific Movement

It is well known that human movement involves closed-loop control coordinating inputs with desired outputs [12, 145]. Feedback from physics-based simulations provides a real-time interaction that is otherwise missing in feedforward approaches. Simulations can complement experiments to help uncover principles of coordinated and uncoordinated movements.

Our objective was to merge approaches from biomechanics and robotics into a platform to advance the study of human movement control and to improve outcome prediction. In addition, we constructed surrogate response surfaces approximate functions of defined task forces using subject specific data. This is specifically important in order to perform predictive simulations. The experimental motion capture data was used to develop surrogate response surfaces approximating the task-level motion of a specific subject. We used surrogate response surface model to fit quadratic functions to our experimental body kinematic data. The surrogate

response model captures the purpose of the fit instead of the actual data for prediction when the data is computationally expensive and/or noisy. A separate quadratic response surface (*multicolored polynomial surface*) was created for each desired subtask (e.g., swing foot position, torso orientation) as a function of a primary task (e.g., center of mass position). Each quadratic response surface requires the solution of unknown polynomial coefficients (b) from the following equation:

$$A = b_0 + \sum_{j=1}^n b_j x_j + \sum_{j=1}^n b_{jj} x_j^2 + \sum_{i \neq j}^n b_{ij} x_i x_j$$

Where, n is number of design variables, x is design variables.

We related the position of the center of mass with respect to the base of support (V_1)'s components in the x (anterior-posterior) and z (medial-lateral) directions (design variables) to position of the left foot (V_2) and the torso (V_3) with respect to the whole-body center of mass (Figure 7.3). This means that:

$$V_3(x, y, z) = f(V_1(x, z), b_{3, coeffs}(x, y, z))$$

and

$$V_2(x, y, z) = f(V_1(x, z), b_{2, coeffs}(x, y, z))$$

After forming surrogate models, we combined the simulated motion (*blue line*) with surrogate models of the experimental motion capture data (*CAREN image, far top right*) to begin forming the closed-loop control systems that are required in many human movement applications (Figure 7.4).

Experimentally measured tasks for each subject were used as data points (*green open circles near surface*) within the bounds of the motion captured. The response surface polynomial coefficients were then determined via a linear least squares fit in MATLAB. Using the response surfaces as surrogates for the subject's specific movement coordination, desired tasks (*black*

line) were computed as data points (*blue filled circles*) on these surfaces as functions of the primary task. The response surface approach allows synthesis of a new, feasible subject specific motion without experimentally measuring that particular motion (e.g., prediction of post-treatment functional outcome from pre-treatment motion capture).

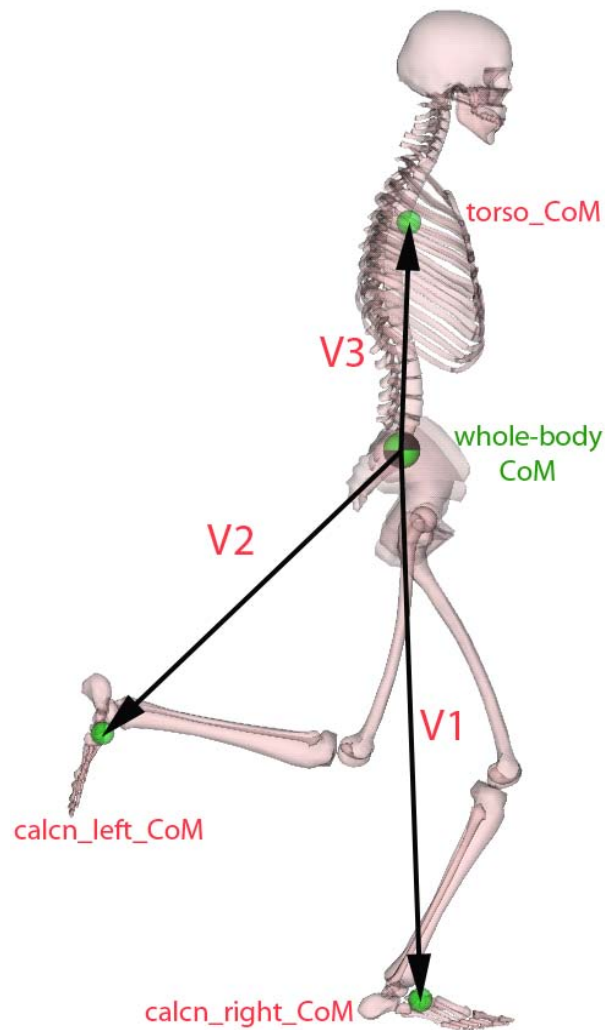


Figure 7. 3: Task-based control of balance simulations flowchart using subject-specific data.

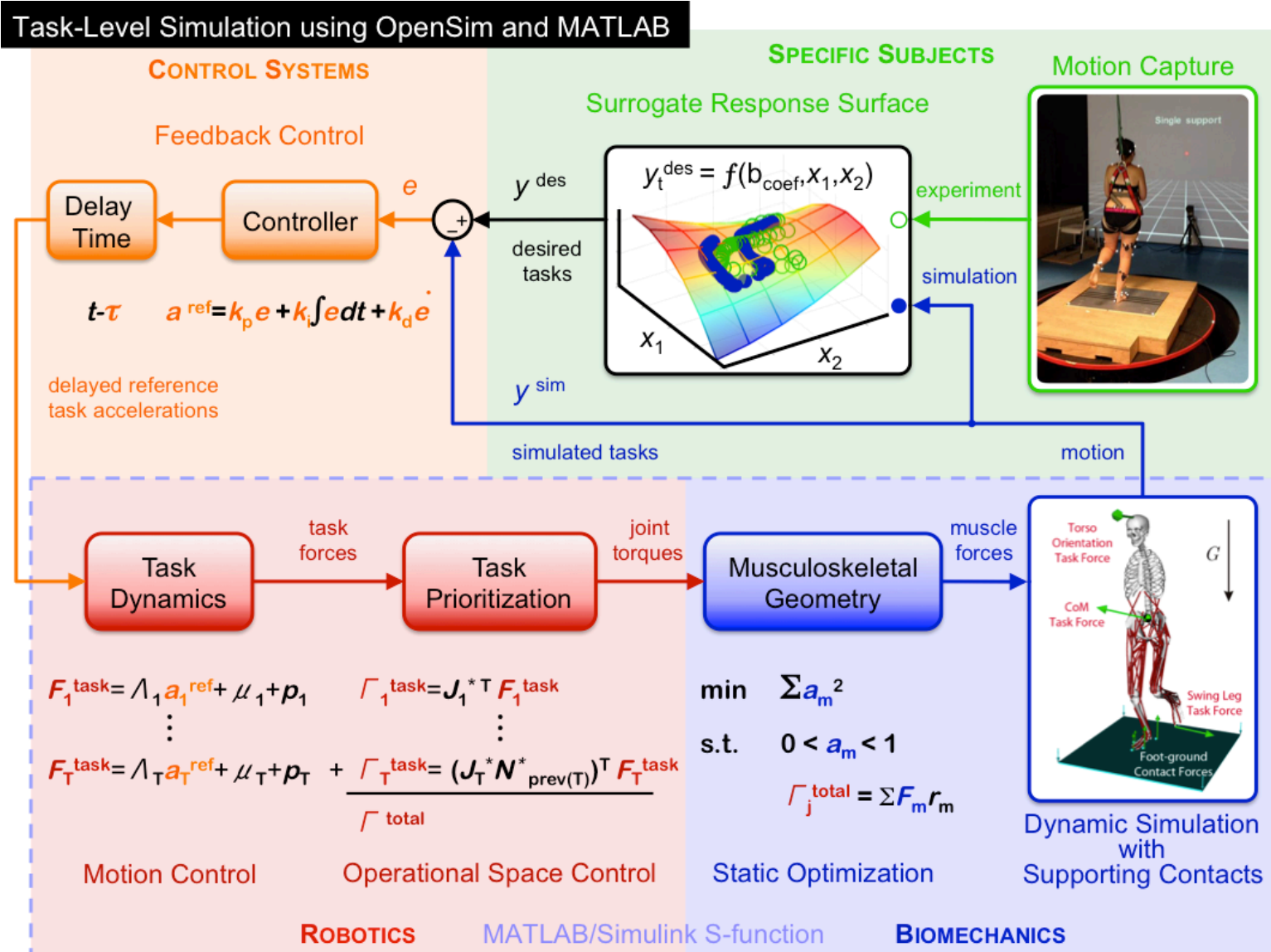


Figure 7. 4: Task-based control of balance synthesis flowchart using subject-specific data.

In Figure 7.4, the bottom right (*blue*), we generated dynamic simulations by combining the rapid model-based design, control systems, and powerful numerical method strengths of MATLAB/Simulink with the simulation and human movement dynamics strengths of OpenSim by developing a new interface between the two software tools (Chapter 2). OpenSim was integrated with Simulink using the MATLAB S-function (system function) mechanism (*purple dashed outlined box*). An S-function is a computer language description of a Simulink block written in MATLAB, FORTRAN, C, or C++ language and compiled as a MEX-file, a dynamically linked subroutine automatically loaded and executed by MATLAB/Simulink, similar to the interaction that takes place between the Simulink engine and built-in Simulink blocks. A dynamic simulation with supporting contacts (*musculoskeletal model image, far bottom right*) may be generated using biomechanics-based approaches. The musculotendon dynamics compute the muscle forces actuating the model from input controls; next, using musculoskeletal geometry, the joint moments are computed; then, using multibody dynamics, accelerations and other state derivatives are computed. Finally, state derivatives are numerically integrated to determine the model's new states, including the simulated motion (*blue line*).

In the top left (*orange*), we used differences between the desired (*black line*) and simulated (*blue line*) tasks to compute task errors. These task errors were used as inputs to proportional-integral-derivative (PID) controllers to continue the feedback loop for updating the simulated motion (*blue line*). The differences between the experimental (*green open circles used to develop response surfaces*) and simulated (*blue filled circles computed from response surfaces*) tasks were minimized by tuning the PID controller gains to a specific subject. A motion control strategy was implemented with the PID control law determining a control policy in acceleration space. The desired task accelerations were delayed in time by a constant ($\tau =$

60ms) [146] representing the neuromechanical time lag for neural processing and control correction command transmission. These delayed reference accelerations (*orange line*) include control policies for all desired task points.

In the bottom left (*red*), we used the delayed reference task accelerations (*orange line*) to generate predictive, whole-body movements using robotics-based prioritized, multi-task approaches for a support-consistent formulation maintaining the foot-ground contact. The task forces necessary for linear control of task accelerations are determined by task dynamics mapping the joint space equations of motion into operational space. Each task force is comprised of the task acceleration multiplied with its task space inertia matrix under supporting constraints, plus the velocity and gravity compensation terms for each task. The joint torques necessary for linear control of task forces are determined by multiplying these forces by the associated support-consistent reduced Jacobians and priority-level-determined, dynamically-consistent null space matrices for each task. As an example application, we defined multiple prioritized tasks to balance a musculoskeletal model on a single supporting limb in contact with the ground. The first priority task was to keep the whole-body center of mass over the base of support (foot). In addition, the lower priority subtasks were the swing foot and torso positions relative to the whole-body center of mass. A compound torque formulation was used to control each prioritized task simultaneously.

Returning to the bottom right (*blue*), we used the joint torques computed from the prioritized operational space motion control to drive a musculoskeletal dynamic simulation with supporting contacts. Static optimization is used to solve the “distribution” problem arising from the fact there are more unknown muscle forces than the number of joint torques. The sum of muscle activations squared is minimized subject to the lower (0) and upper (1) bounds on

activation and the constraint that the net muscle moments, generated by multiplying muscles forces by their moment arms, must equal the total task joint torques. The muscle forces generate joint accelerations and other state derivatives that are numerically integrated to simulate the subject-specific, whole-body coordinated movement (*blue line*). The simulations started at 1.3s and ended at 3s. The time span was selected to include all the perturbations and recovery time while neglect the experimental data before the perturbations started to reduce computational costs of the simulations.

7.4. Results

We analyzed two experimental trials from the CAREN platform of our subjects during anterior (subject 1, female) and posterior (subject 2, male) support surfaces translations [144]. We compared the experimental joint kinematics and muscle activities with our synthesized muscle driven simulations. The quadratic surrogate response surfaces for both subjects are illustrated in the following figures 7.5, 7.6, 7.7 and 7.8. These figures show the relationship between vectors V_2 (left calcaneus) and V_3 (torso) with projection of V_1 in the x and z directions for both the experimental and the simulated surrogate trials.

The kinematic differences between the simulated and experimental data are shown in Figure 7.9-15. Figure 7.9 is a qualitative illustration of the last frame of simulations. It compares the joint angles resulting from the experiments and the simulations for each subject. The quantitative root mean square values for subject one's right hip flexion angle (3.4°), right hip adduction angle (0.93°), right knee angles (4.98°), left hip flexion angle (6.5°), left hip adduction angle (3.37°), left knee angles (9.58°) is shown in figure 7.10. The quantitative root mean square values for subject two's right hip flexion angle (6.08°), right hip adduction angle (2.16°), right knee angles (4.53°), left hip flexion angle (6.77°), left hip adduction angle (4.85°), left knee

angles (5.86°) is shown in figure 7.12. Figure 7.11 and figure 7.13 show the quantitative kinematics for the 6 DOF pelvis joints (three rotational and three translational) for both subjects. Similarly, figures 7.14 and 7.15 show CoM displacement in both anterior/posterior directions from the experiment (green, solid) and the corresponding simulation (blue, dashed) for both subjects. The simulated whole-body CoM displacements were, on average, within 0.7 cm (female) and 1.3 cm (male) of experimental ones.

Simulated muscle activities were also compared with the experimental EMG data. Our simulations were based on 92 muscle-actuators. We only compared the 8 muscles (Biceps Femoris, Medial Gastrocnemius, Gluteus Maximus, Gluteus Medius, Rectus Femoris, Semitendinosus, Vastus Lateralis and Vastus Medialis) for which the experimental EMG data was available. The raw EMGs (Figure 7.16-7.19, first rows) were first rectified (Figure 7.16-7.19, second rows) and were then filtered (Figure 7.16-7.19, third rows) using a zero-lag Butterworth filter (high pass of 30 Hz and low pass of 500 Hz, cut-off frequencies) in MATLAB. The rectified and filtered raw EMGs then were compared with the corresponding simulated muscle activations derived from static optimization (Figure 7.16-7.19, fourth rows). Model muscle activations matched experimental EMG in timing of peak muscle excitations. Differences between simulated and experimental balance recovery were similar in both anterior and posterior translation directions of the CAREN base.

7.5. Discussion & Conclusion

We merged approaches from biomechanics, control theories and robotics for task-level simulation of subject-specific balance recovery. OpenSim/MATLAB interface was used as a main platform for merging closed-loop forward dynamics simulations with state-of-the-art operational space controller using task prioritization, support-consistent contact modeling and

free-floating dynamic formulations for predictive simulations of human movement. To synthesize predictive subject-specific simulations, we used surrogate models of experimental kinematics to form a relationship between our prioritized tasks. Instead of tracking a specific trajectory (joint coordinates or body kinematics), we used experimental body kinematics data to find relationships between bodies' end-effector movements and their contribution to whole-body center of mass using surrogate response surfaces. The response surface approach allows synthesis of novel subject-specific movements without experimentally measuring that particular motion (e.g., prediction of post-treatment functional outcome from pre-treatment motion capture).

Our results showed good conformity between the experimental kinematics and the synthesized motions. In spite of reported kinematic differences at individual joints, the overall motions (whole-body center of mass movements) were analogous to the experiments. In addition, the estimated muscle activations closely matched the experimental EMG in both directions (anterior and posterior translations). These results confirm how a complex subject-specific movement can be reconstructed by sequencing multiple simple tasks such as keeping whole-body center of mass over the base of support, controlling torso orientation in ground frame and controlling swing leg to achieve a desired task.

The results of this study should be interpreted within the context of modeling assumptions and capabilities to perform predictive simulations. First, synthesizing kinematics using the quadratic surrogate response model is only one way to represent the relationship between the defined tasks. Higher-order models can be designed to represent various movements; however, the quadratic model was sufficient to represent our experimental balance recovery data. Second, single surrogate response surface may not be a full representation of each

subject's motor control model. Creating multiple surrogate surfaces to represent subject's movements in multi-directional perturbations and various different movements can be used to form a more comprehensive motor control model. Third, despite kinematic differences between the synthesized motion and the experimental balance recovery data, the muscle activities to generate those motions were similar in both peak timing and magnitude for both subjects. Fourth, the balance recovery simulations used foot-ground contact modeling that depends upon parameters such as geometry, stiffness and dissipation. The stiffness and dissipation values were chosen based on material properties of skin (foot) and concrete (ground) materials. The same contact model parameters were used in each simulation, so observed differences are not the result of foot-ground differences. Despite these challenges, our conclusions regarding predictive simulation of human balance recovery will remain valid.

Directions for future work include defining different surrogate models based on individual subject movement pattern data to better understand and predict human movement that comprise the human motor control strategies. The combination of biomechanics and robotics approaches for closed-loop control simulations allowed synthesis of subject-specific task-level motions. The surrogate response surfaces allowed synthesis of new, coordinated motions without additional experimental motion capture data. The optimal PID gains allowed control over simulated tasks for each subject. The operational space control allowed multiple prioritized tasks to balance on a single supporting limb in contact with the ground. The four areas of the closed loop (specific subjects, control systems, robotics, and biomechanics) offer numerous directions for future work to advance the study of human balance control and subject-specific outcome predictions.

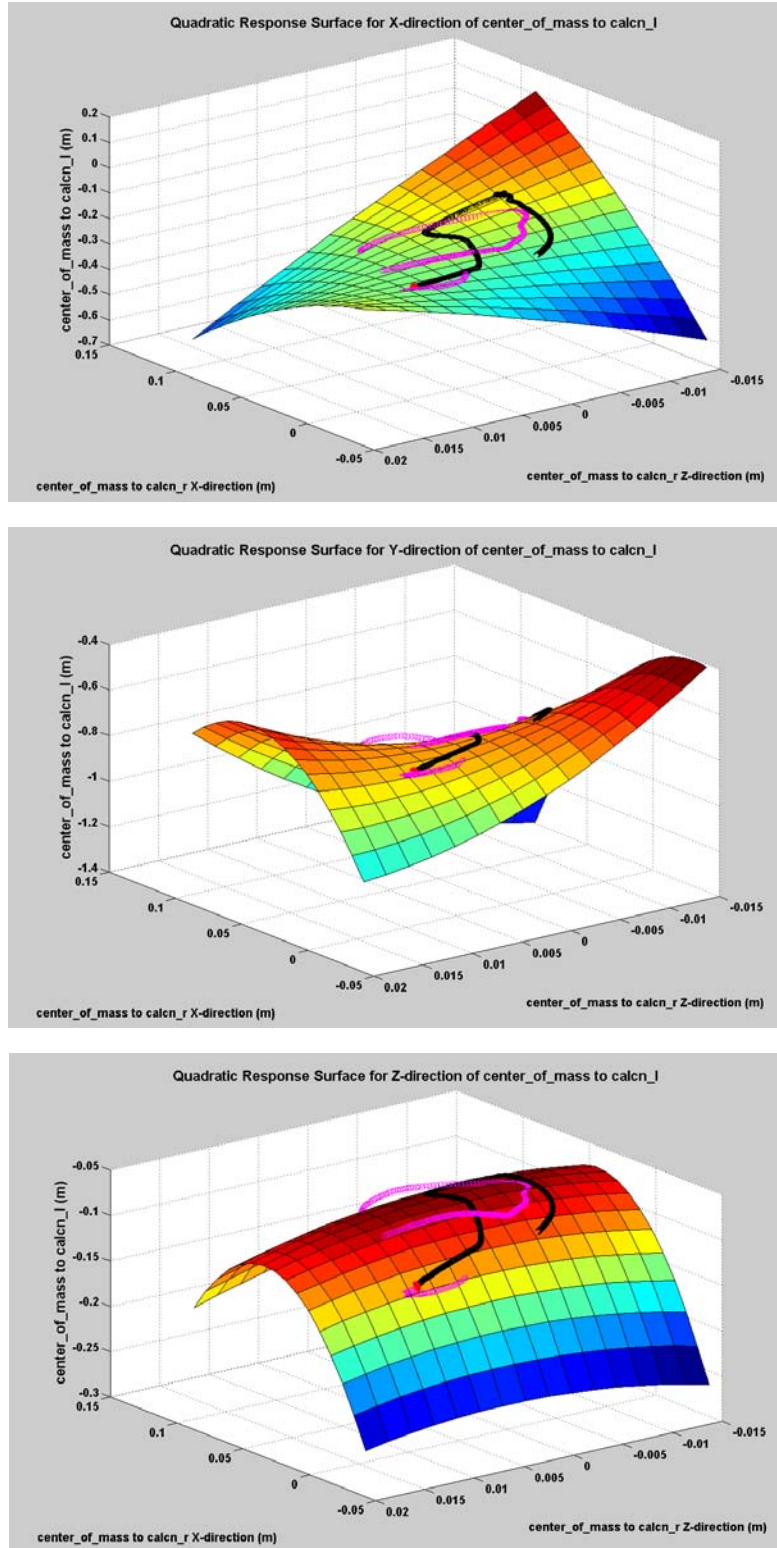


Figure 7. 5: Quadratic response surfaces for Subject 1's CoM to left calcaneus (V_2) in x, y and z direction over projection of V1 (experimental (magenta) and simulated (black)).

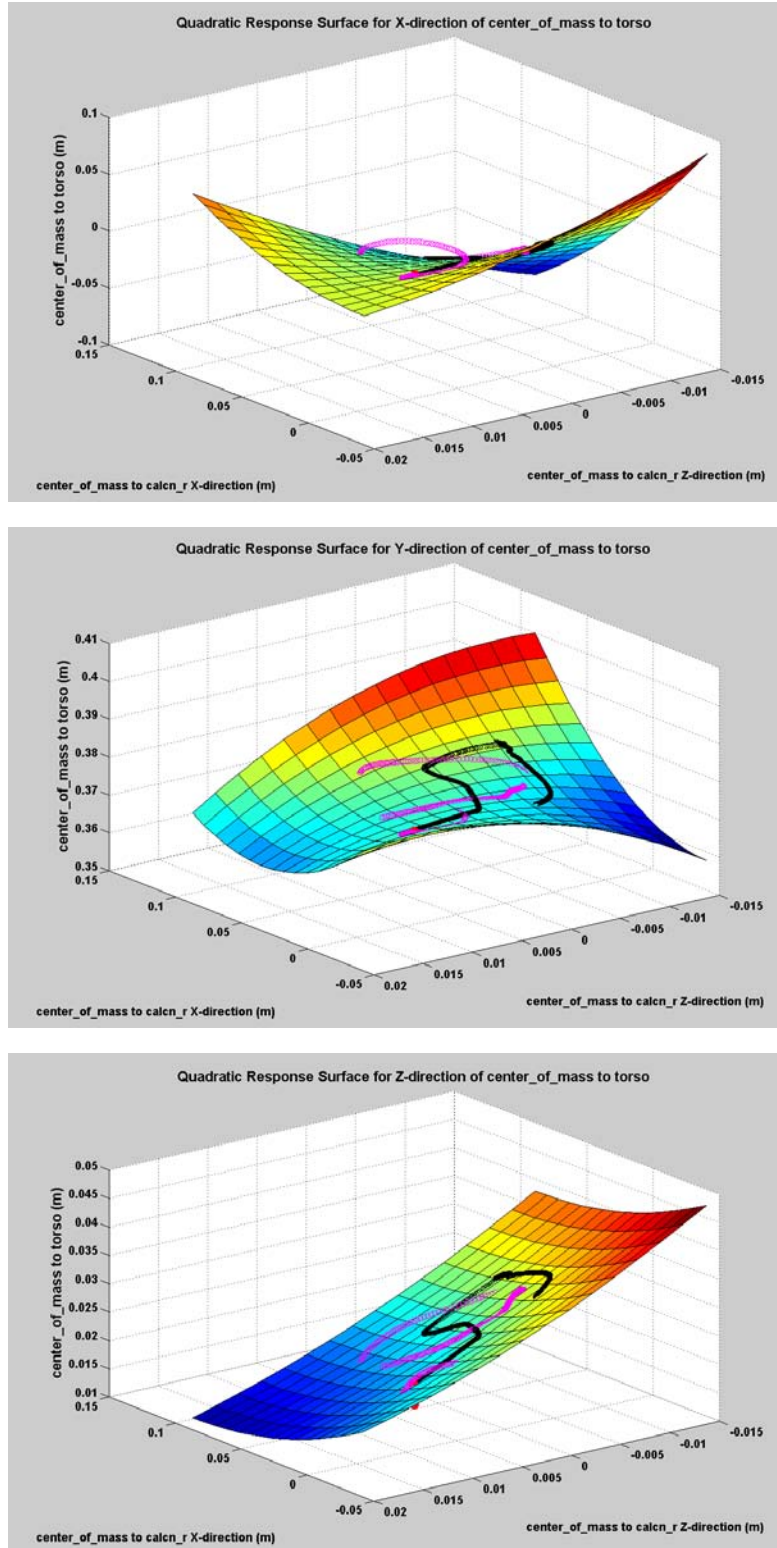


Figure 7. 6: Quadratic responses surface for Subject 1's CoM to torso (V_3) in x, y and z direction over projection of V1 (experimental (magenta) and simulated (black)).

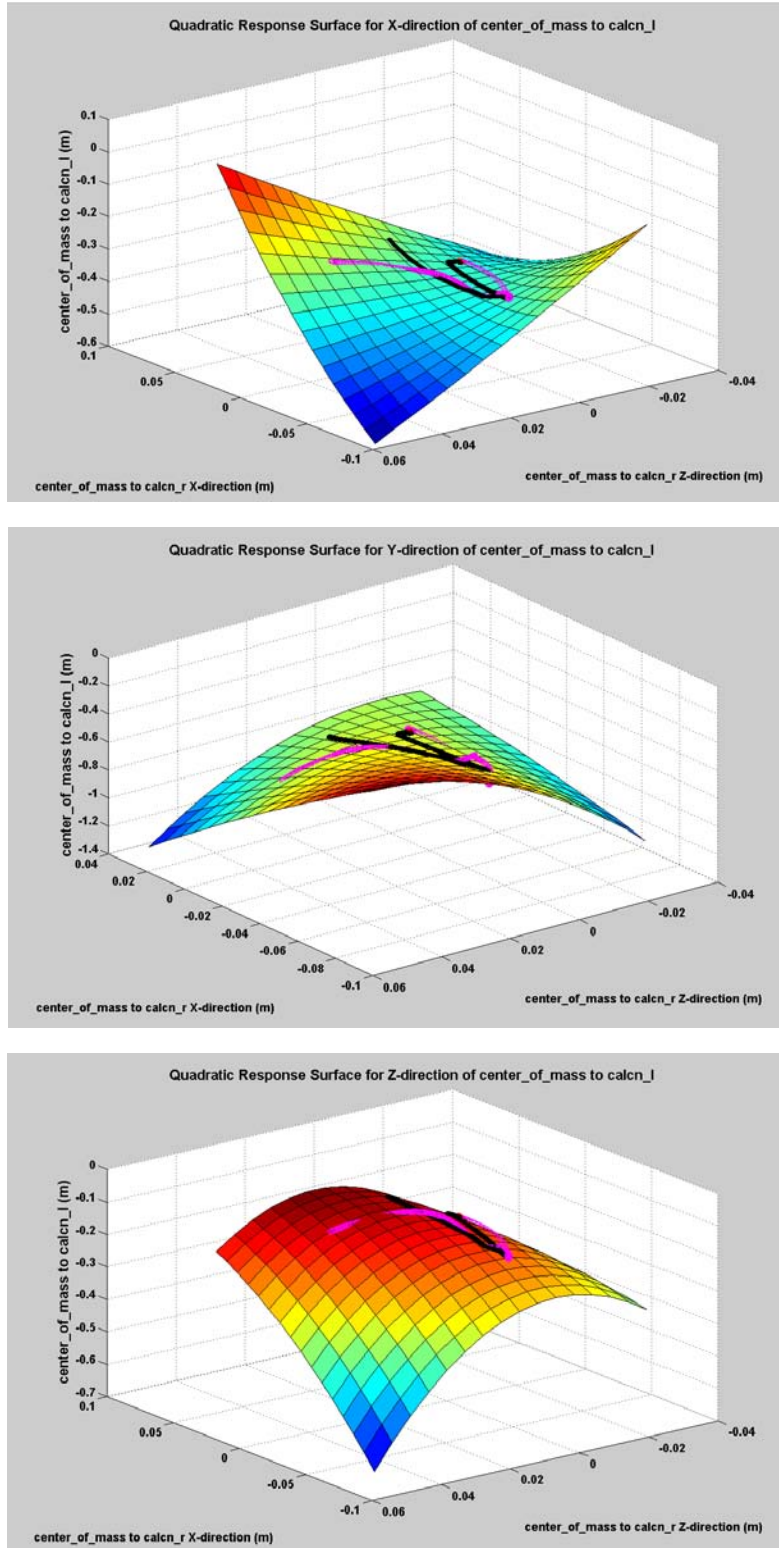


Figure 7. 7: Quadratic response surfaces for Subject 2's CoM to left calcaneus (V₂) in x, y and z direction over projection of V1 (experimental (magenta) and simulated (black)).

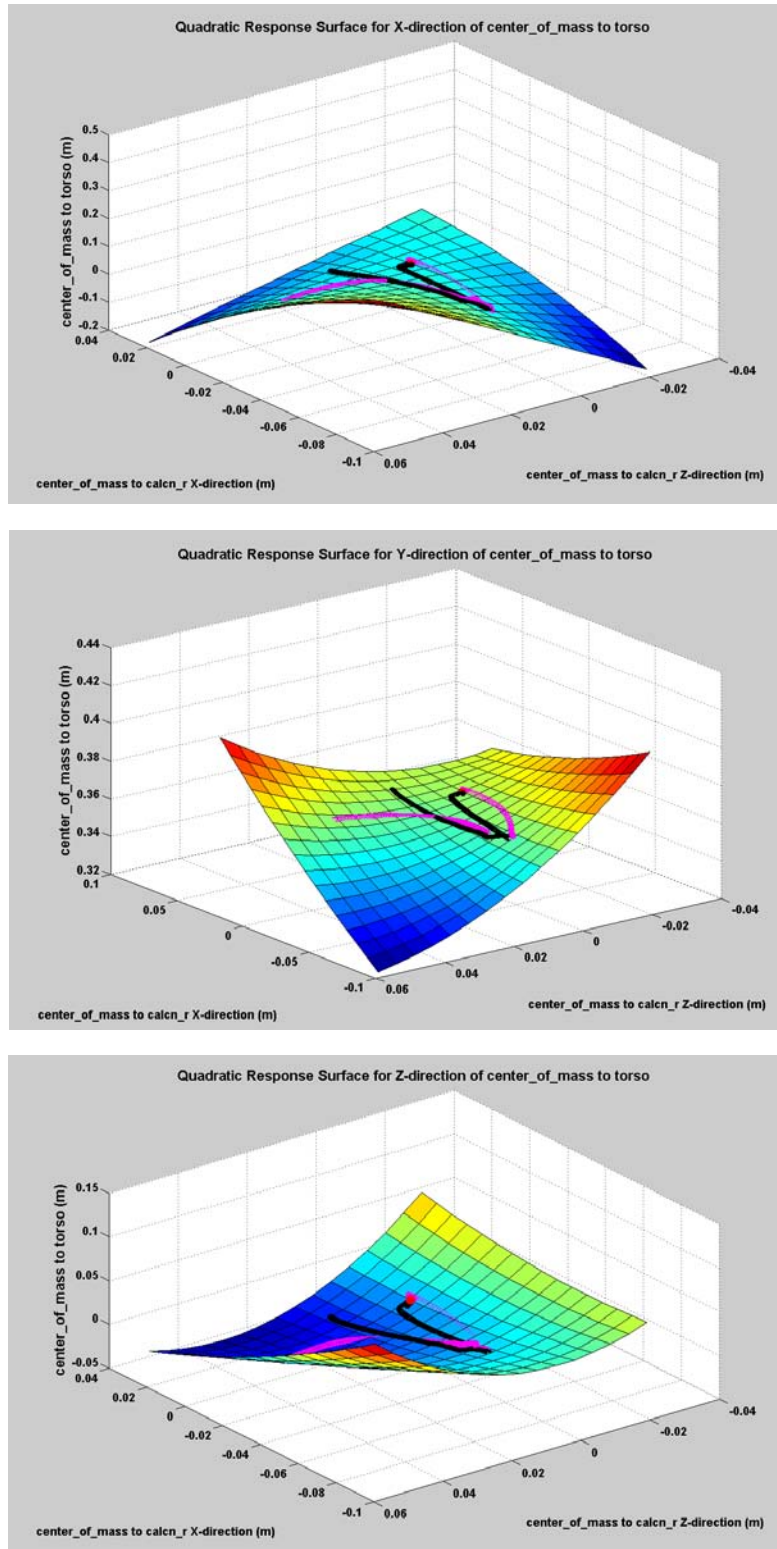


Figure 7. 8: Quadratic response surfaces for Subject 2's CoM to left torso (V_3) in x, y and z direction over projection of V_1 (experimental (magenta) and simulated (black)).

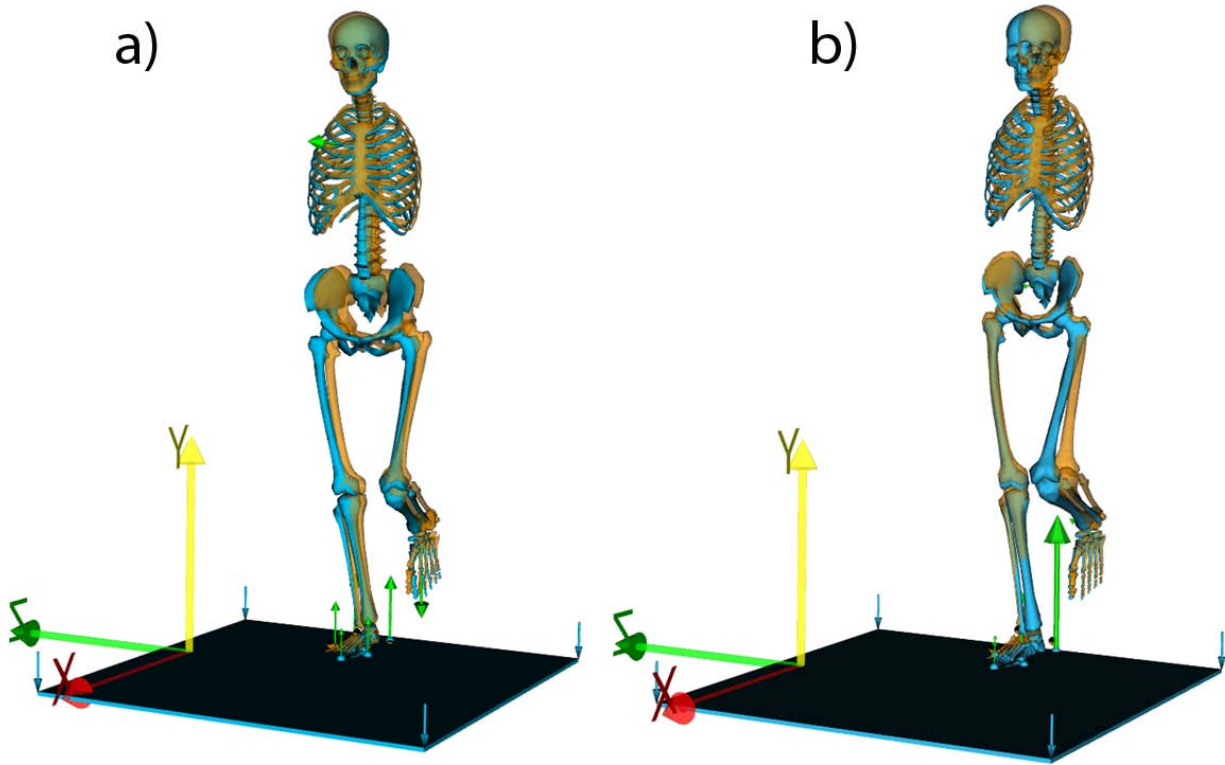


Figure 7. 9: kinematic differences between experimental (orange) and simulated movement (blue) for a) subject 1 and b) subject 2. This figure is constructed from the last frame of the simulations to qualitatively show how well the experiments and synthesized motion match.

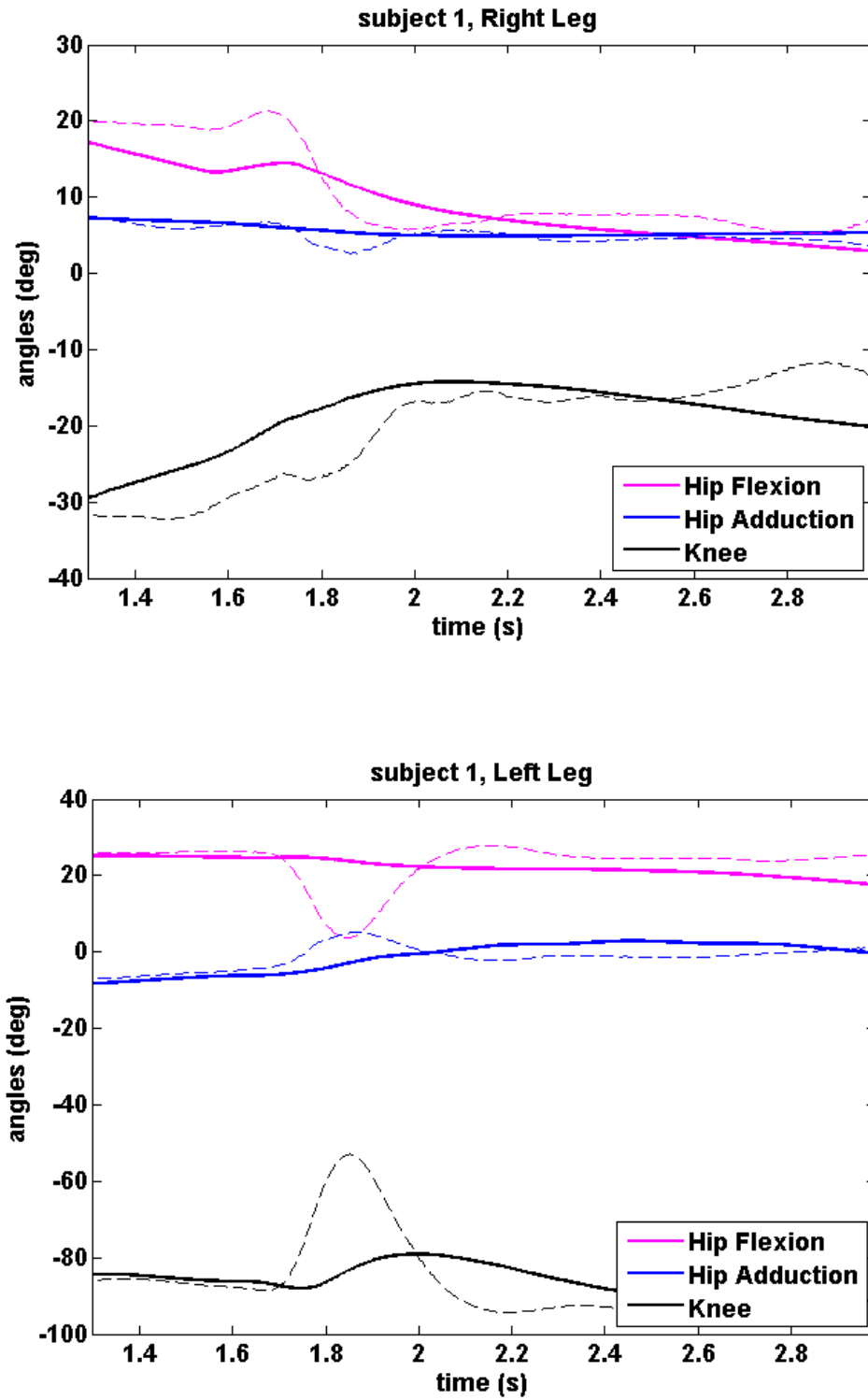


Figure 7. 10: Subject 1's right leg and left leg hip and knee joints kinematics (experimental (dash) and synthesized (solid)).

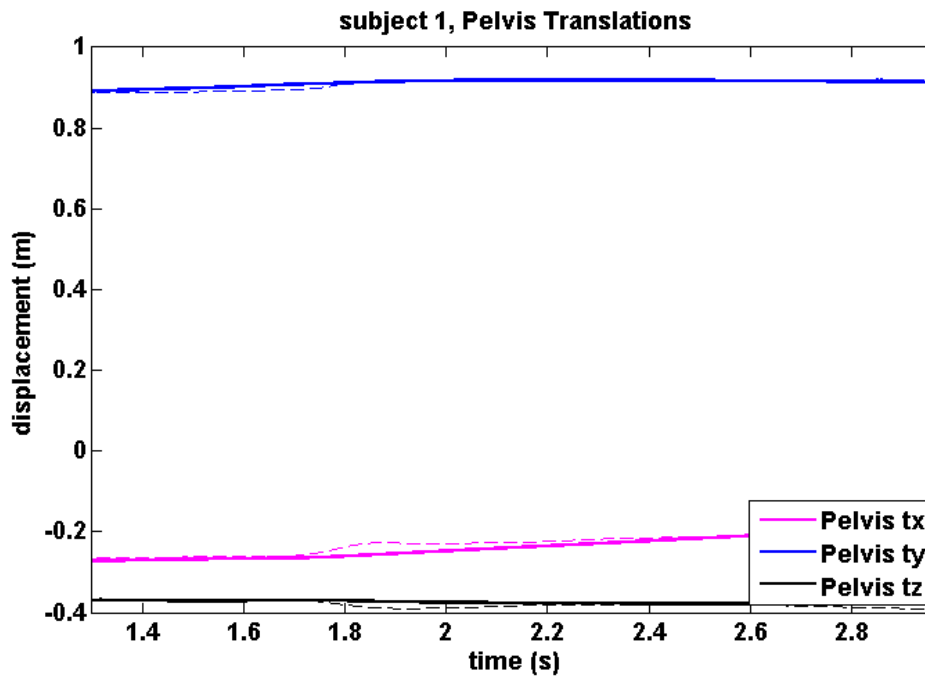
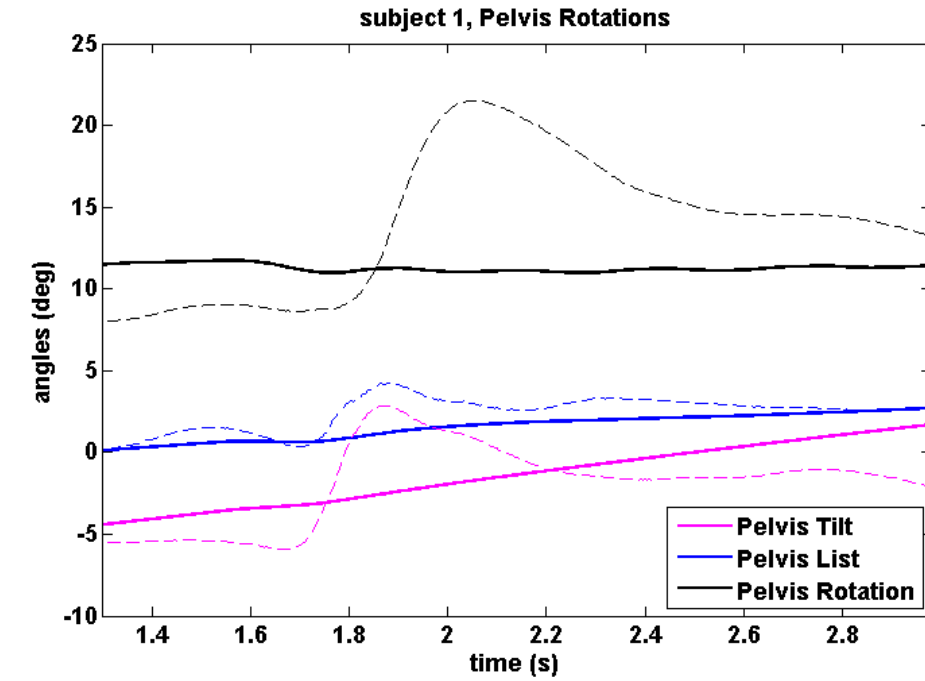


Figure 7. 11: Subject 1's pelvis rotations and translation joint kinematics (experimental (dash) and synthesized (solid)).

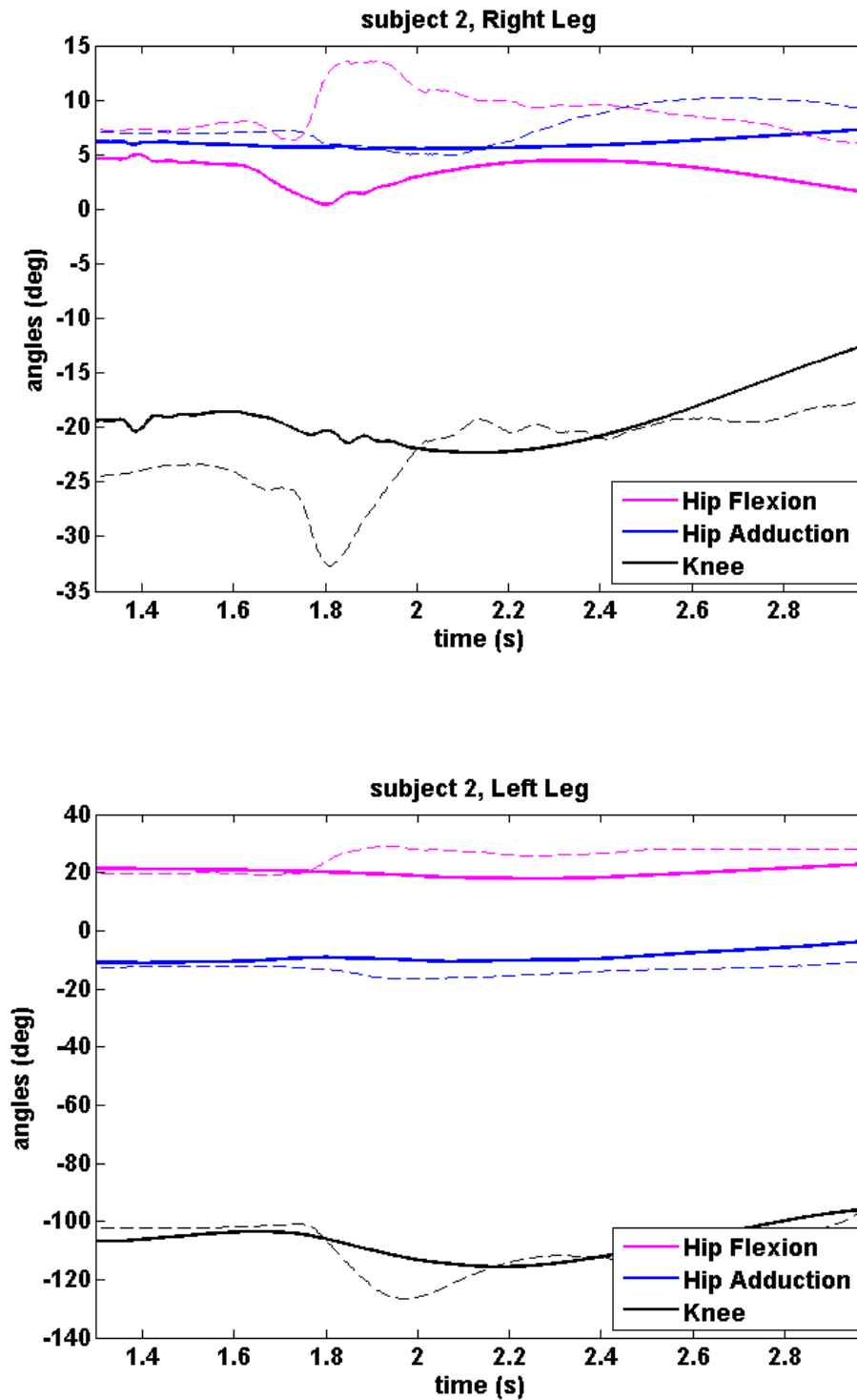


Figure 7. 12: Subject 2's right leg and left leg hip and knee joints kinematics (experimental (dash) and synthesized (solid)).

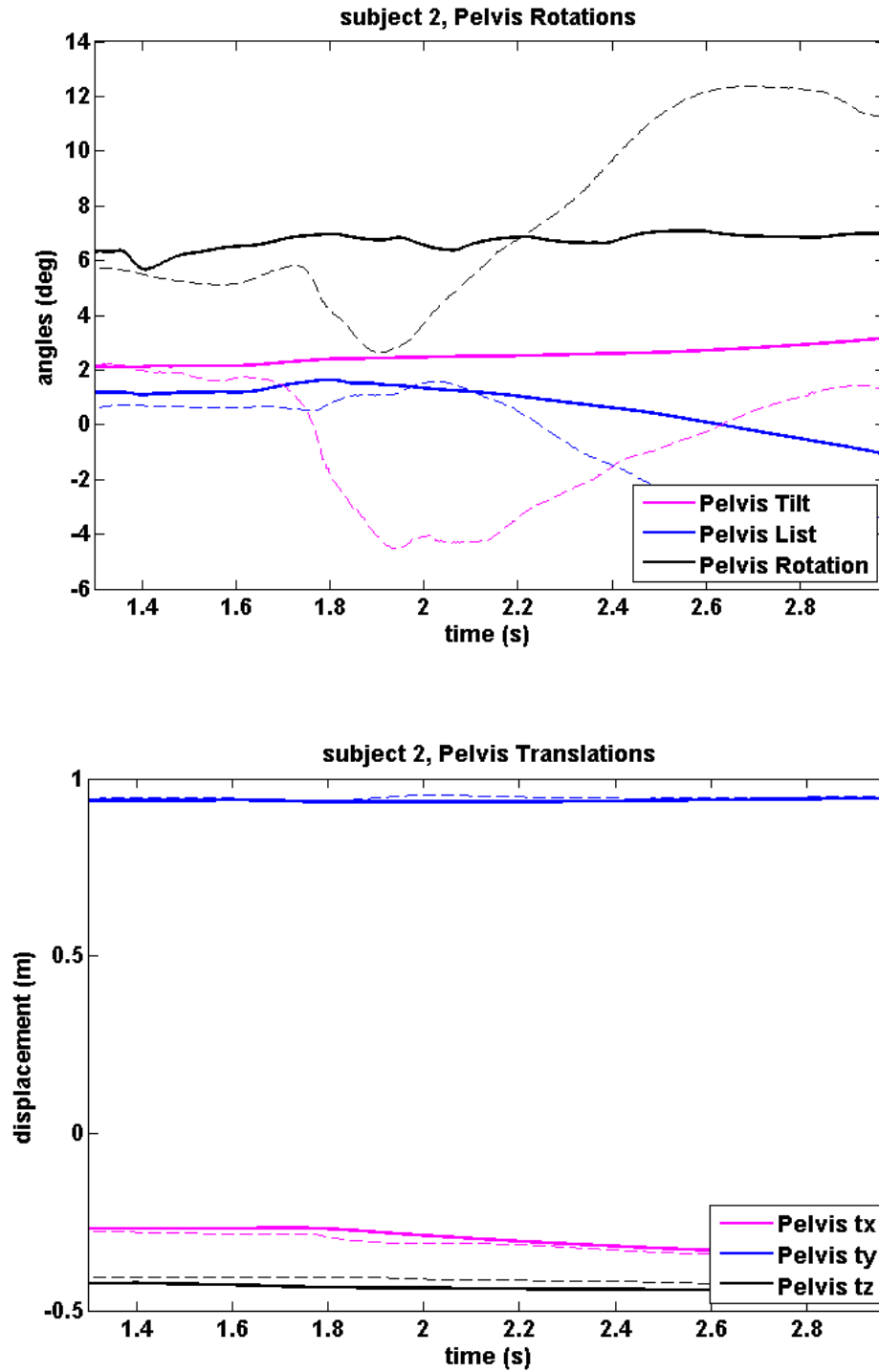


Figure 7. 13: Subject 2's pelvis rotations and translation joint kinematics (experimental (dash) and synthesized (solid)).

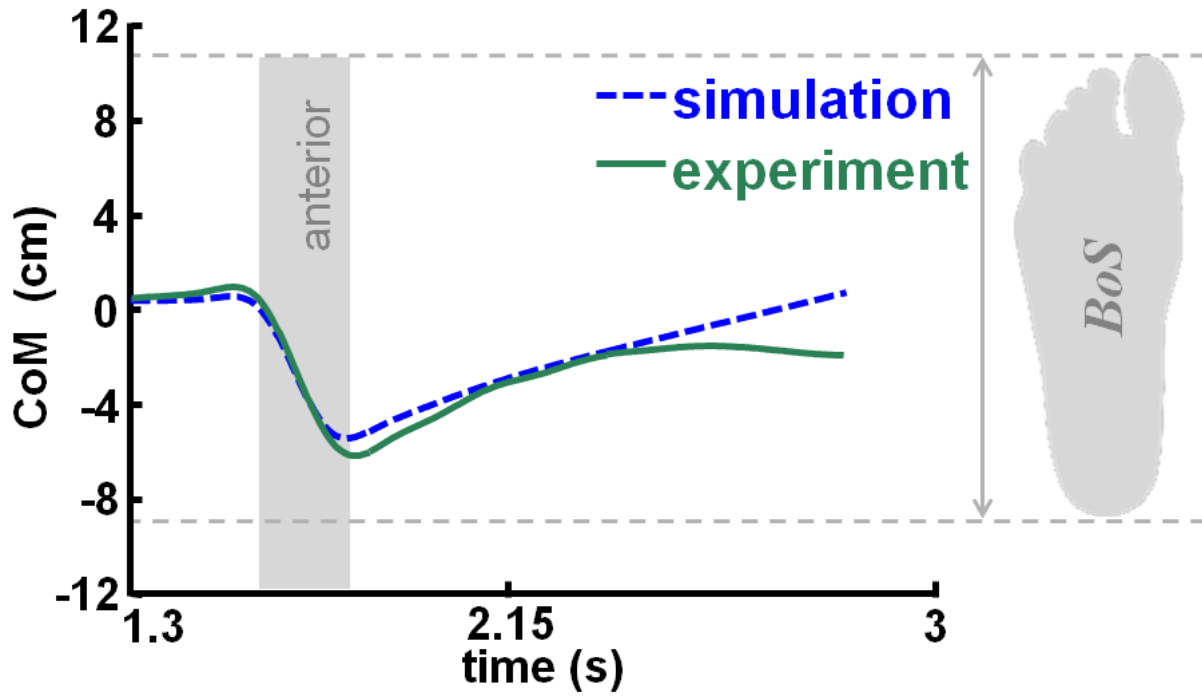


Figure 7. 14: Subject 1's CoM displacement in anterior-posterior direction from the experiment (green) and the corresponding simulation (blue).

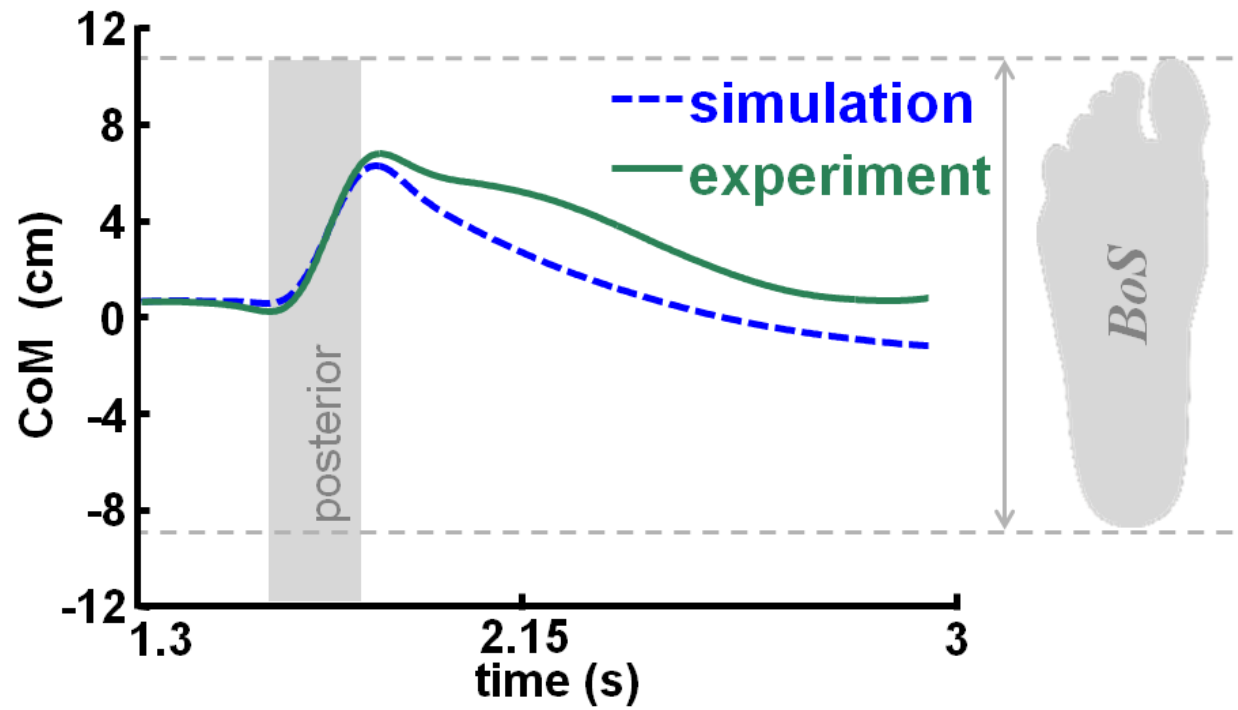


Figure 7. 15: Subject 2's CoM displacement in anterior-posterior direction from the experiment (green) and the corresponding simulation (blue).

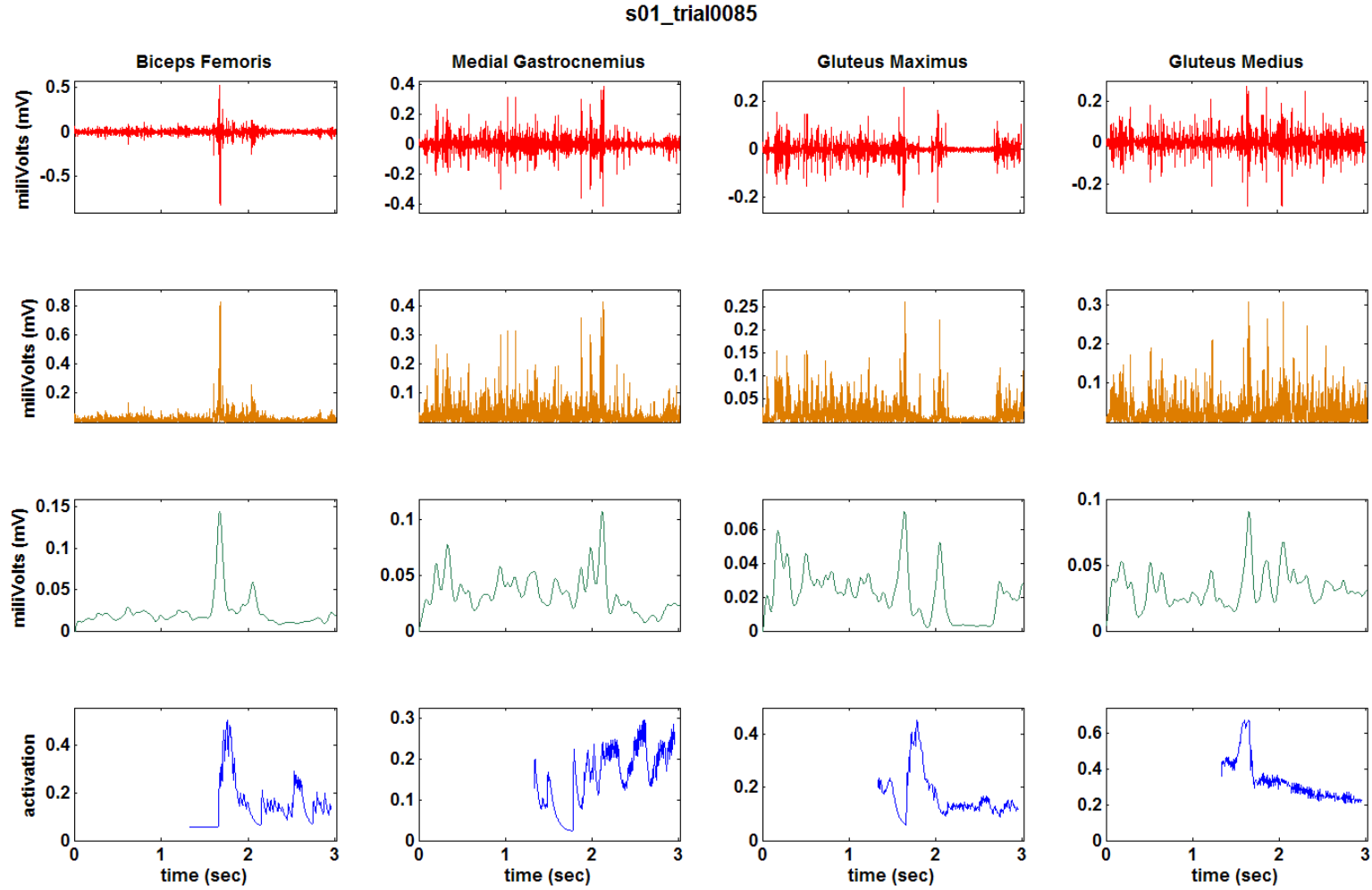


Figure 7. 16: Experimental (first three rows) EMG versus the simulated (last row) muscle activities (Biceps Femoris, Medial Gastrocnemius, Gluteus Maximus and Gluteus Medius) for subject 1. Experimental raw EMGs (first row, red), rectified EMGs (second row, orange), experimental filtered EMGs (third row, green) and simulated muscle activations (last row, blue).

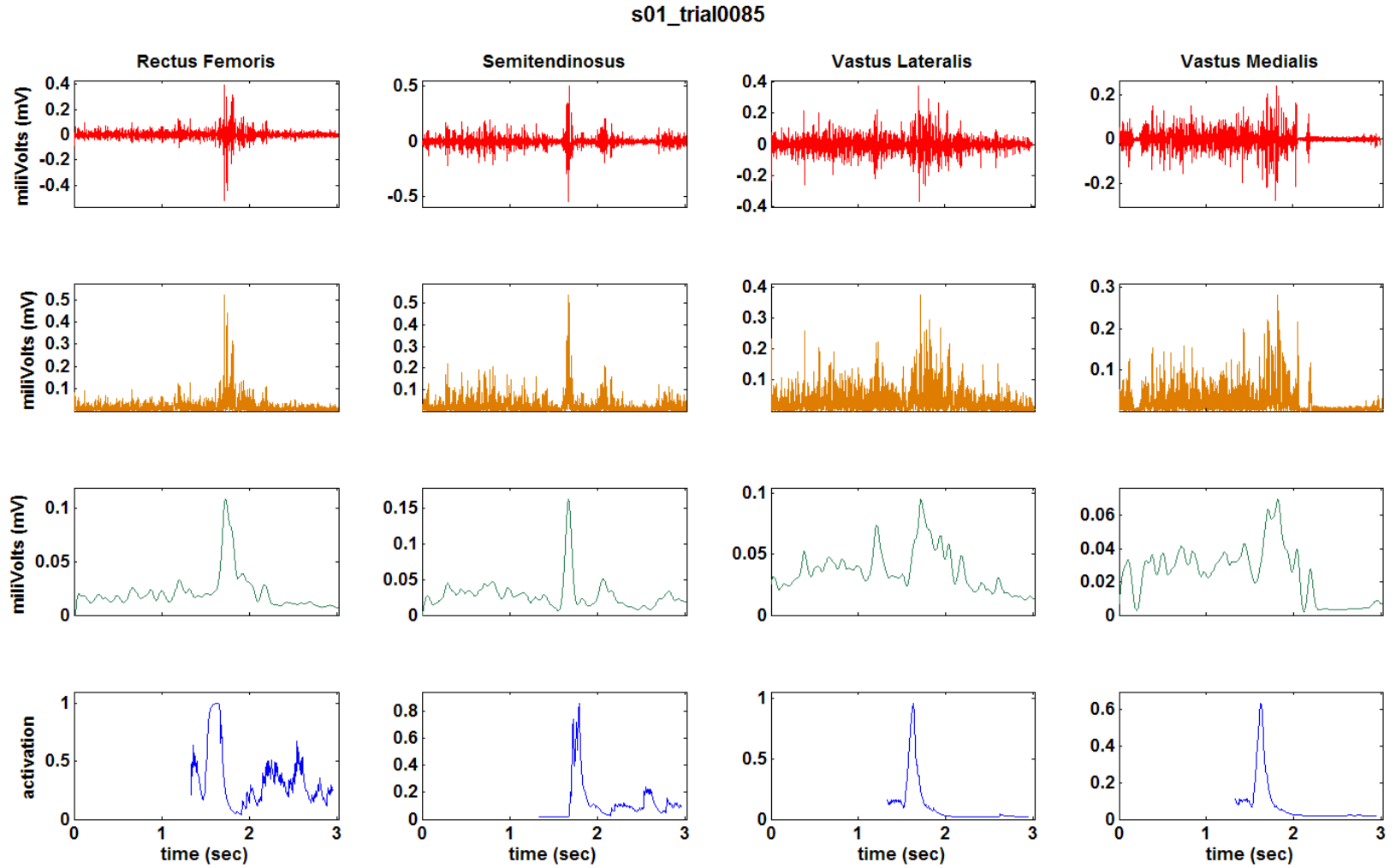


Figure 7. 17: Experimental (first three rows) EMG versus the simulated (last row) muscle activities (Rectus Femoris, Semitendinosus, Vastus Lateralis, Vastus Medialis) for subject 1. Experimental raw EMGs (first row, red), Experimental rectified EMGs (second row, orange), experimental filtered EMGs (third row, green) and simulated muscle activations (last row, blue).

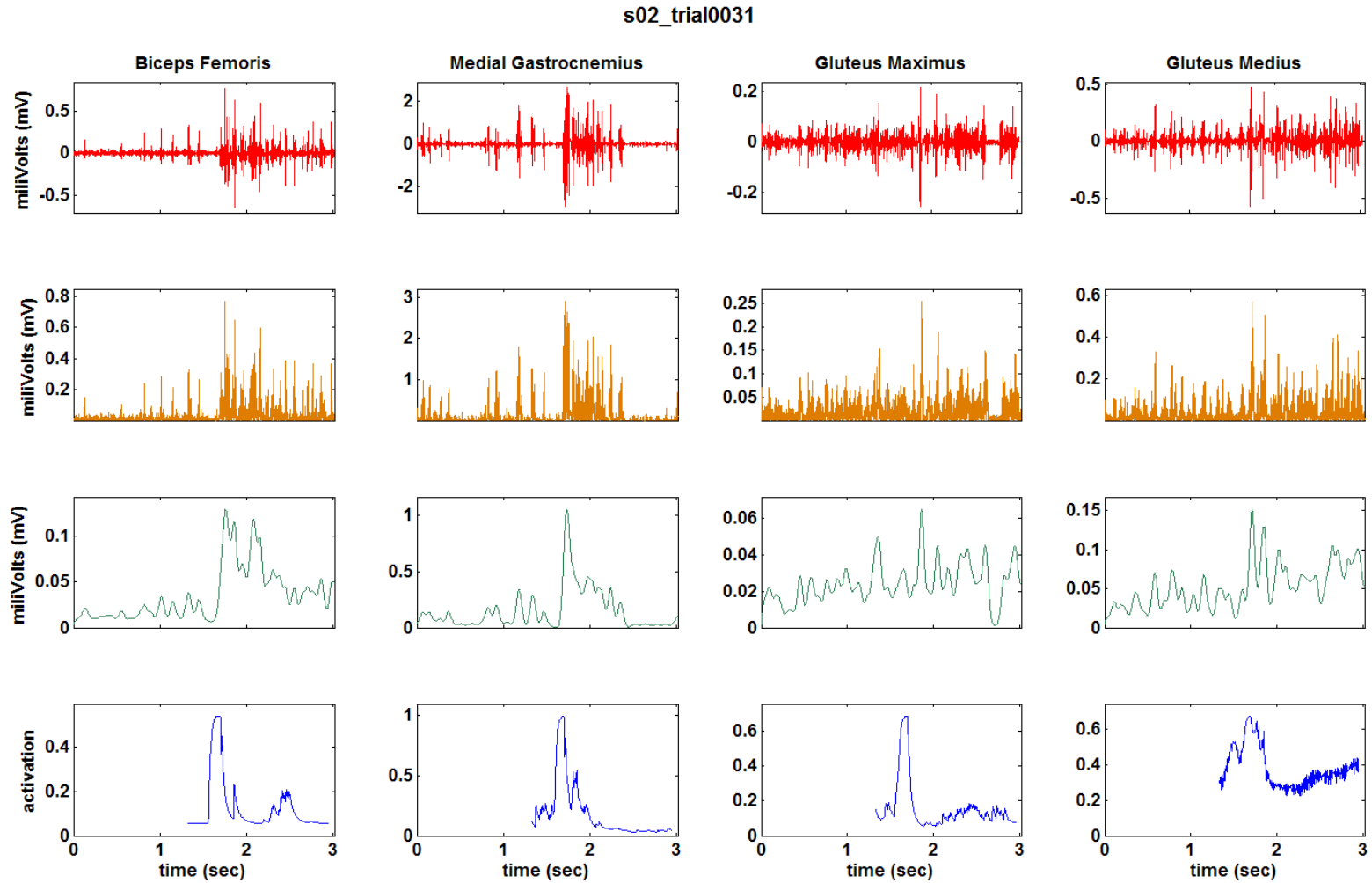


Figure 7. 18: Experimental (first three rows) EMG versus the simulated (last row) muscle activities (Biceps Femoris, Medial Gastrocnemius, Gluteus Maximus and Gluteus Medius) for subject 2. Experimental raw EMGs (first row, red), rectified EMGs (second row, orange), experimental filtered EMGs (third row, green) and simulated muscle activations (last row, blue).

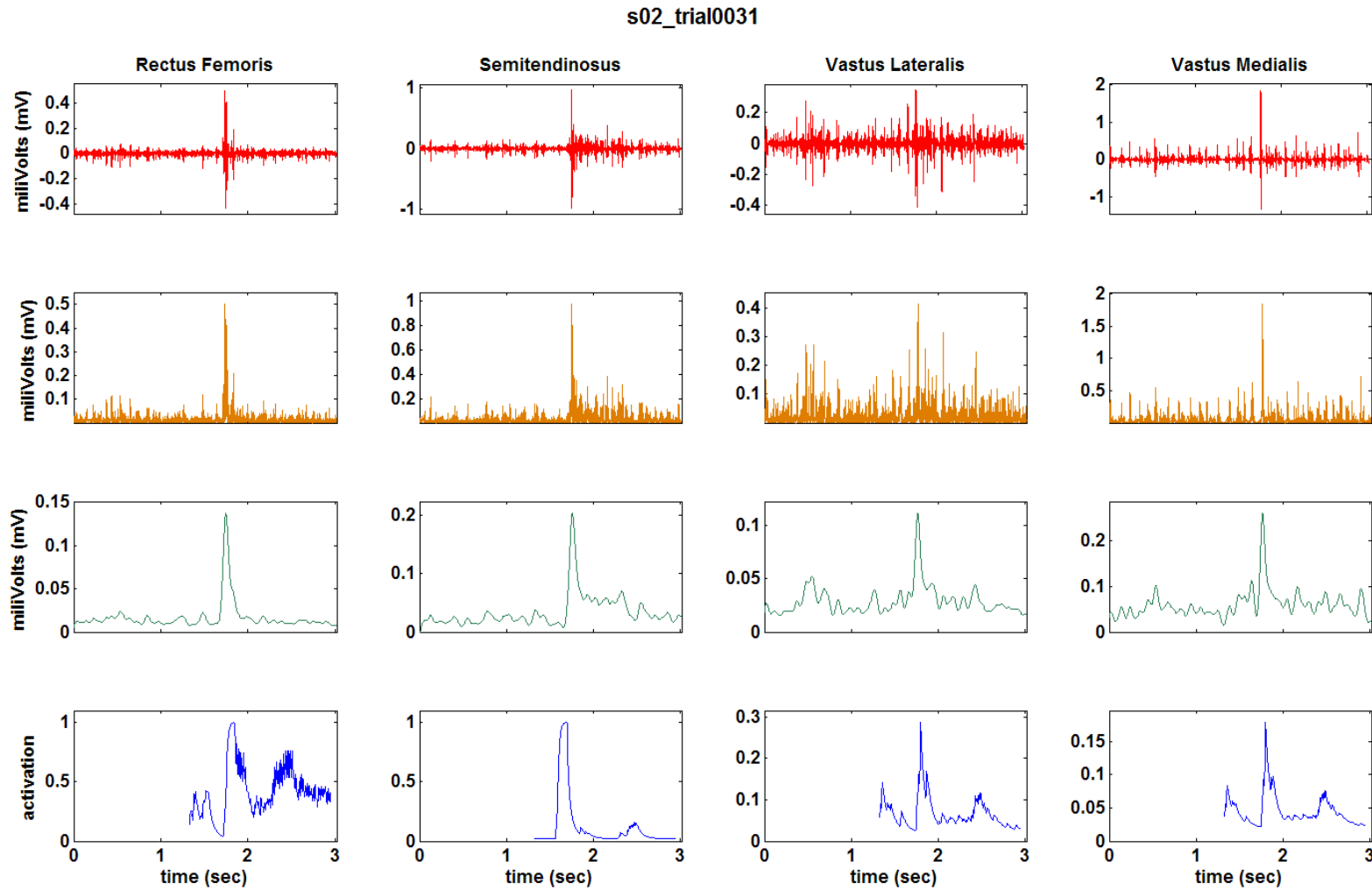


Figure 7. 19: Experimental (first three rows) EMG versus the simulated (last row) muscle activities (Rectus Femoris, Semitendinosus, Vastus Lateralis, Vastus Medialis) for subject 2. Experimental raw EMGs (first row, red), Experimental rectified EMGs (second row, orange), experimental filtered EMGs (third row, green) and simulated muscle activations (last row, blue).

CHAPTER 8

CONCLUDING REMARKS

8.1. Research Innovation

Treatment recommendations and rehabilitation techniques for many biomechanical conditions are generally based on qualitative observations of the patient's gait, experimental gait analyses, clinical examinations, and intuition of the clinical team. Even though experimental approaches have advanced our understanding of biomechanics, they are limited in predicting functional outcome and identifying cause-effect relationships (i.e. muscle contributions to movement) between muscle forces and joint biomechanics. Muscle-actuated simulations and biologically-inspired controller design gives insight into the complex interaction between muscle excitations, musculoskeletal geometry and observed multi-joint movements.

The interface developed in this study, combines numerical simulation and human movement dynamics strengths of OpenSim with the robust design, powerful math, and control systems strengths of MATLAB. This integrated platform has promise for helping researches gain a better understanding of movement control and has the potential to improve treatment planning. This interface also enables us to add closed-loop forward dynamics simulation that was not previously available through other software packages. The open-loop model is limited to using predefined input controls that cannot be changed; however, many human movement applications require closed-loop control systems that update input controls to generate a desired output.

In addition, complex patient-specific musculoskeletal model with 23 degrees of freedom and 92 muscle actuators was used for our musculoskeletal modeling. This provides a more realistic modeling and simulation tool to analyze human movement. Different contact modeling approaches (e.g. elastic foundation and Hunt-Crossley) were used for modeling foot-ground

contacts and generating ground reaction forces required for predictive forward dynamic simulations. This allows us to calculate reaction forces and predict motions in applications where experimental data is not available. Furthermore, this work combined high-level supraspinal signals and low-level spinal signals to test hypotheses regarding human movement and balance control. The biologically-inspired controller acts similarly to the central nervous system (CNS) in controlling human balance.

This research creates a basis for modeling and analyzing important biomechanical conditions such as balance and control in humans with neuromuscular control deficits by merging the clinical and experimental expertise of physicians, physical therapists and kinesiologists with the dynamic modeling, computer simulation and control systems designs know-how of engineers. In addition, subject-specific models were developed on freely available open-source software. Using a freely available software implies that the discoveries can be readily shared, investigated and discussed with other researchers to promote dialogue amongst clinical and computational disciplines in order help advance the field toward the end goal of understanding the underlying mechanisms behind balance and movement disorders.

8.2. Summary

In this research, we created a platform to combine musculoskeletal modeling, forward dynamic simulation, optimization techniques, and neuromuscular control. This integrated platform has promise for better understanding movement control and the potential to improve treatment planning.

Our platform extended the OpenSim forward dynamics tool by adding the real-time feedback loop from output states to the musculoskeletal model input controls. This closed-loop feedback controller is required in many human movement applications. Building upon our

platform, we then developed low-level spinal (stretch-reflex) and high-level (task-level) controllers to test simulation-based hypotheses related to balance recovery and movement control.

Specifically, we targeted a corrective procedure (rectus femoris transfer surgery) in children experiencing stiff-knee gait with the goal of exploring how this procedure may affect these children's balance recovery. Balance is among the most challenging tasks for patients with neuromuscular disorders such as children with cerebral palsy. We combined clinical movement analysis and simulation-based approaches to understand the biomechanical consequences of this surgical procedure. Our results suggested that distal transfer of the rectus femoris muscle to the insertion of the sartorius may change control of balance provided by the muscle path for patients with spastic CP. The pre-surgical simulations maintained balance following both anterior and posterior support-surface translations while post-surgical ones did not maintain balance following posterior translations. Balance recovery was also influenced by crouched postures where moderate crouch was better than mild crouch in helping recover from anterior support-surface translations but not necessarily posterior translations. These results indicate that rectus femoris plays an important role in postural response to support-surface translations. This study provides a foundation for the identification of other biomechanical factors contributing to movement abnormalities and may potentially improve surgical and rehabilitation treatments for patients with neurological disorders.

We extended our closed-loop (high-level) controllers by merging approaches from robotics and biomechanics. We implemented a prioritized multi-task, support-consistent, task-based controller inside our simulation platform to synthesize human balance. We then validated our simulated results with the experimental data of healthy adults by defining surrogate response

surfaces that represent the patient's primary tasks (to keep their balance) as function of defined subtasks (e.g. swing leg positions or torso orientations. The combination of biomechanics and robotics approaches for closed-loop control simulations allowed synthesis of subject-specific task-level motions. The surrogate response surfaces allowed synthesis of new, coordinated motions without additional experimental motion capture data. The optimal PID gains allowed control over simulated tasks for each subject. The operational space control allowed multiple prioritized tasks to balance on a single supporting limb in contact with the ground. The four areas of the closed loop (specific subjects, control systems, robotics, and biomechanics) offer numerous directions for future work to advance the study of human balance control and subject-specific outcome predictions.

There are several opportunities for future research to build upon the results of this study. 1) a need for studies investigating support-surface perturbations in children with CP following rectus femoris transfer surgeries; 2) a proper link between functional tests for gross motor function and balance recovery; 3) immeasurable decoupling of spinal reflexes (as we have studied here) and supraspinal brain signals in human studies of balance; 4) alternative treatment procedures (e.g., Achilles tendon lengthening); and 5) multidirectional support-surface perturbations of differing amplitudes. We feel this future work may lead to uncovering principles that govern how the CNS selects appropriate muscle patterns for balance recovery.

The potential of using this platform to study, predict functional outcomes and perhaps improve treatments for musculoskeletal conditions is exciting and valuable. This project not only integrates software tools, but also allows integration of neuroscientists, physiologists, biomechanists, and physical therapists to adopt, adapt, and generate new solutions for musculoskeletal conditions.

8.3. Source Code

Our interface is implemented in freely available musculoskeletal modeling and simulation software (OpenSim), which enables these results to be shared with other research groups. In addition, all of the source codes of the interface, and Simulink model examples, related to this work are available on a SimTK.org project dedicated to the interface (<https://simtk.org/home/opensim-matlab>).

REFERENCES

- [1] T. King. (2008, January). *How great is the burden of musculoskeletal disease?* Available: <http://www.aaos.org/news/aaosnow/jan08/youraaos15.asp>
- [2] R. D. Komistek, D. A. Dennis, and M. Mahfouz, "In vivo fluoroscopic analysis of the normal human knee," *Clinical Orthopaedics and Related Research*, pp. 69-81, May 2003.
- [3] J. J. Wu, "Clinical-Application of Vicon System to Evaluate the Gait Pattern after Toe-to-Thumb Reconstruction," *Journal of Biomechanics*, vol. 20, pp. 910-910, 1987.
- [4] J. Schmid, A. Sandholm, F. Chung, D. Thalmann, H. Delingette, and N. Magnenat-Thalmann, "Musculoskeletal Simulation Model Generation from MRI Data Sets and Motion Capture Data," *Recent Advances in the 3d Physiological Human*, pp. 3-19, 2009.
- [5] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H. P. Seidel, "Motion Capture Using Joint Skeleton Tracking and Surface Estimation," *Cvpr: 2009 Ieee Conference on Computer Vision and Pattern Recognition, Vols 1-4*, pp. 1746-1753, 2009.
- [6] R. Kram, T. M. Griffin, J. M. Donelan, and Y. H. Chang, "Force treadmill for measuring vertical and horizontal ground reaction forces," *J Appl Physiol (1985)*, vol. 85, pp. 764-9, Aug 1998.
- [7] Z. P. Luo, L. J. Berglund, and K. N. An, "Validation of F-Scan pressure sensor system: A technical note," *Journal of Rehabilitation Research and Development*, vol. 35, pp. 186-191, Jun 1998.
- [8] C. J. De Luca, "The use of surface electromyography in biomechanics," *Journal of Applied Biomechanics*, vol. 13, pp. 135-163, May 1997.
- [9] G. Robertson, G. Coldwell, J. Hamill, G. Kamen, and S. Whittlesey, *Research Methods in Biomechanics*. Champaign: Human Kinetics, 2004.
- [10] W. M. Murray, S. L. Delp, and T. S. Buchanan, "Variation of muscle moment arms with elbow and forearm position," *Journal of Biomechanics*, vol. 28, pp. 513-25, May 1995.
- [11] A. S. Arnold, S. Salinas, D. J. Asakawa, and S. L. Delp, "Accuracy of muscle moment arms estimated from MRI-based musculoskeletal models of the lower extremity," *Comput Aided Surg*, vol. 5, pp. 108-19, 2000.
- [12] A. D. Kuo, "An Optimal-Control Model for Analyzing Human Postural Balance," *Ieee Transactions on Biomedical Engineering*, vol. 42, pp. 87-101, Jan 1995.
- [13] T. Mergner, C. Maurer, and R. J. Peterka, "A multisensory posture control model of human upright stance," *Neural Control of Space Coding and Action Production*, vol. 142, pp. 189-199, 2003.
- [14] D. T. Davy and M. L. Audu, "A dynamic optimization technique for predicting muscle forces in the swing phase of gait," *Journal of Biomechanics*, vol. 20, pp. 187-201, 1987.
- [15] F. C. Anderson and M. G. Pandy, "Dynamic optimization of human walking," *J Biomech Eng*, vol. 123, pp. 381-90, Oct 2001.
- [16] R. R. Neptune, F. E. Zajac, and S. A. Kautz, "Muscle force redistributes segmental power for body progression during walking," *Gait & Posture*, vol. 19, pp. 194-205, Apr 2004.
- [17] I. Jonkers, C. Stewart, and A. Spaepen, "The complementary role of the plantarflexors, hamstrings and gluteus maximus in the control of stance limb stability during gait," *Gait Posture*, vol. 17, pp. 264-72, Jun 2003.
- [18] G. K. Cole, B. M. Nigg, A. J. vandenBogert, and K. G. M. Gerritsen, "Lower extremity joint loading during impact in running," *Clinical Biomechanics*, vol. 11, pp. 181-193, Jun 1996.

- [19] K. G. Gerritsen, A. J. van den Bogert, and B. M. Nigg, "Direct dynamics simulation of the impact phase in heel-toe running," *Journal of Biomechanics*, vol. 28, pp. 661-8, Jun 1995.
- [20] S. R. Hamner, A. Seth, and S. L. Delp, "Muscle contributions to propulsion and support during running," *Journal of Biomechanics*, vol. 43, pp. 2709-16, Oct 19 2010.
- [21] R. R. Neptune and M. L. Hull, "Evaluation of performance criteria for simulation of submaximal steady-state cycling using a forward dynamic model," *J Biomech Eng*, vol. 120, pp. 334-41, Jun 1998.
- [22] A. J. K. van Soest and L. J. R. Casius, "Which factors determine the optimal pedaling rate in sprint cycling?," *Medicine and Science in Sports and Exercise*, vol. 32, pp. 1927-1934, Nov 2000.
- [23] R. Redfield and M. L. Hull, "Prediction of Pedal Forces in Bicycling Using Optimization Methods," *Journal of Biomechanics*, vol. 19, pp. 523-540, 1986.
- [24] M. G. Pandy and F. E. Zajac, "Optimal muscular coordination strategies for jumping," *Journal of Biomechanics*, vol. 24, pp. 1-10, 1991.
- [25] F. C. Anderson and M. G. Pandy, "A Dynamic Optimization Solution for Vertical Jumping in Three Dimensions," *Comput Methods Biomech Biomed Engin*, vol. 2, pp. 201-231, 1999.
- [26] M. F. Bobbert and A. J. van Soest, "Why do people jump the way they do?," *Exerc Sport Sci Rev*, vol. 29, pp. 95-102, Jul 2001.
- [27] H. Hatze, "The inverse dynamics problem of neuromuscular control," *Biological Cybernetics*, vol. 82, pp. 133-41, Feb 2000.
- [28] K. G. M. Gerritsen, W. Nachbauer, and A. J. vandenBogert, "Computer simulation of landing movement in downhill skiing: Anterior cruciate ligament injuries," *Journal of Biomechanics*, vol. 29, pp. 845-854, Jul 1996.
- [29] K. R. Kaufman, K. N. An, W. J. Litchy, and E. Y. Chao, "Physiological prediction of muscle forces--II. Application to isokinetic exercise," *Neuroscience*, vol. 40, pp. 793-804, 1991.
- [30] G. Li, K. R. Kaufman, E. Y. S. Chao, and H. E. Rubash, "Prediction of antagonistic muscle forces using inverse dynamic optimization during flexion extension of the knee," *Journal of Biomechanical Engineering-Transactions of the Asme*, vol. 121, pp. 316-322, Jun 1999.
- [31] K. B. Shelburne and M. G. Pandy, "A dynamic model of the knee and lower limb for simulating rising movements," *Comput Methods Biomech Biomed Engin*, vol. 5, pp. 149-59, Apr 2002.
- [32] R. R. Neptune, I. C. Wright, and A. J. Van den Bogert, "Muscle coordination and function during cutting movements," *Medicine and Science in Sports and Exercise*, vol. 31, pp. 294-302, Feb 1999.
- [33] G. T. Yamaguchi and F. E. Zajac, "Restoring unassisted natural gait to paraplegics via functional neuromuscular stimulation: a computer simulation study," *IEEE Trans Biomed Eng*, vol. 37, pp. 886-902, Sep 1990.
- [34] L. M. Schutte, M. M. Rodgers, and F. E. Zajac, "Improving the efficacy of electrical stimulation-induced leg cycle ergometry: an analysis based on a dynamic musculoskeletal model," *IEEE Transactions on Rehabilitation Engineering*, vol. 1, p. 17, 1993.
- [35] R. F. Kirsch, A. M. Acosta, F. C. T. van der Helm, R. J. J. Rotteveel, and L. A. Cash, "Model-based development of neuroprostheses for restoring proximal arm function,"

- Journal of Rehabilitation Research and Development*, vol. 38, pp. 619-626, Nov-Dec 2001.
- [36] R. R. Neptune and S. A. Kautz, "Knee joint loading in forward versus backward pedaling: implications for rehabilitation strategies," *Clinical Biomechanics*, vol. 15, pp. 528-535, Aug 2000.
 - [37] I. C. Wright, R. R. Neptune, A. J. van den Bogert, and B. M. Nigg, "The influence of foot positioning on ankle sprains," *Journal of Biomechanics*, vol. 33, pp. 513-519, May 2000.
 - [38] I. C. Wright, R. R. Neptune, A. J. van den Bogert, and B. M. Nigg, "The effects of ankle compliance and flexibility on ankle sprains," *Medicine and Science in Sports and Exercise*, vol. 32, pp. 260-265, Feb 2000.
 - [39] S. J. Piazza and S. L. Delp, "The influence of muscles on knee flexion during the swing phase of gait," *Journal of Biomechanics*, vol. 29, pp. 723-33, Jun 1996.
 - [40] P. G. Riley, "Medicine as a moral art: the Hippocratic philosophy of Herbert Ratner, M.D," *Linacre Q*, vol. 65, pp. 5-38, Nov 1998.
 - [41] S. R. Goldberg, F. C. Anderson, M. G. Pandy, and S. L. Delp, "Muscles that influence knee flexion velocity in double support: implications for stiff-knee gait," *Journal of Biomechanics*, vol. 37, pp. 1189-96, Aug 2004.
 - [42] J. A. Reinbolt, R. T. Haftka, T. L. Chmielewski, and B. J. Fregly, "A computational framework to predict post-treatment outcome for gait-related disorders," *Medical Engineering & Physics*, vol. 30, pp. 434-443, May 2008.
 - [43] J. A. Reinbolt, J. F. Schutte, B. J. Fregly, B. Koh, R. T. Haftka, A. D. George, *et al.*, "Determination of patient-specific multi-joint kinematic models through two-level optimization," *Journal of Biomechanics*, vol. 38, pp. 621-626, Mar 2005.
 - [44] B. I. Koh, J. A. Reinbolt, B. J. Fregly, and A. D. George, "Evaluation of parallel decomposition methods for biomechanical optimizations," *Comput Methods Biomech Biomed Engin*, vol. 7, pp. 215-25, Aug 2004.
 - [45] J. F. Schutte, B. I. Koh, J. A. Reinbolt, R. T. Haftka, A. D. George, and B. J. Fregly, "Evaluation of a particle swarm algorithm for biomechanical optimization," *Journal of Biomechanical Engineering-Transactions of the Asme*, vol. 127, pp. 465-474, Jun 2005.
 - [46] J. F. Schutte, J. A. Reinbolt, B. J. Fregly, R. T. Haftka, and A. D. George, "Parallel global optimization with the particle swarm algorithm," *International Journal for Numerical Methods in Engineering*, vol. 61, pp. 2296-2315, Dec 7 2004.
 - [47] B. J. Fregly, J. A. Reinbolt, K. L. Rooney, K. H. Mitchell, and T. L. Chmielewski, "Design of patient-specific gait modifications for knee osteoarthritis rehabilitation," *Ieee Transactions on Biomedical Engineering*, vol. 54, pp. 1687-1695, Sep 2007.
 - [48] B. J. Fregly, J. A. Reinbolt, and T. L. Chmielewski, "Evaluation of a patient-specific cost function to predict the influence of foot path on the knee adduction torque during gait," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 11, pp. 63-71, Feb 2008.
 - [49] B. I. Koh, J. A. Reinbolt, A. D. George, R. T. Haftka, and B. J. Fregly, "Limitations of parallel global optimization for large-scale human movement problems," *Medical Engineering & Physics*, vol. 31, pp. 515-521, Jun 2009.
 - [50] J. A. Reinbolt, R. T. Haftka, T. L. Chmielewski, and B. J. Fregly, "Are patient-specific joint and inertial parameters necessary for accurate inverse dynamics analyses of gait?," *Ieee Transactions on Biomedical Engineering*, vol. 54, pp. 782-793, May 2007.

- [51] J. T. Oden, T. Belytschko, J. Fish, T. J. R. Hughes, C. Johnson, D. Keyes, *et al.*, "Report of the National Science Foundation Blue Ribbon Panel on Simulation-Based Engineering Science," 2006.
- [52] F. H. Stillinger and T. A. Weber, "Computer-Simulation of Local Order in Condensed Phases of Silicon," *Physical Review B*, vol. 31, pp. 5262-5271, 1985.
- [53] C. K. Birdsall and A. B. Langdon, *Plasma physics via computer simulation*. New York: McGraw-Hill, 1985.
- [54] F. Multon, L. France, M. P. Cani-Gascuel, and G. Debunne, "Computer animation of human walking: a survey," *Journal of Visualization and Computer Animation*, vol. 10, pp. 39-54, Jan-Mar 1999.
- [55] F. Darema, "Engineering/scientific and commercial applicaitons: differences, similarities, and future evolution," in *Second Hellenic European Conference on Mathematics and Informations*, 1994, pp. 367-374.
- [56] C. Douglas and A. Deshmukh, "Dynamic Data Driven Application Systems. NSF Workshop Report.," 2000.
- [57] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of Honda humanoid robot," *1998 Ieee International Conference on Robotics and Automation, Vols 1-4*, pp. 1321-1326, 1998.
- [58] J. Yamaguchi, E. Soga, S. Inoue, and A. Takanishi, "Development of a bipedal humanoid robot - Control method of whole body cooperative dynamic biped walking," *Icra '99: Ieee International Conference on Robotics and Automation, Vols 1-4, Proceedings*, pp. 368-374, 1999.
- [59] N. S. Pollard, J. K. Hodgins, M. J. Riley, and C. G. Atkeson, "Adapting human motion for the control of a humanoid robot," *2002 Ieee International Conference on Robotics and Automation, Vols I-Iv, Proceedings*, pp. 1390-1397, 2002.
- [60] Y. Nakamura, K. Yamane, Y. Fujita, and I. Suzuki, "Somatosensory computation for man-machine interface LE from motion-capture data and musculoskeletal human model," *Ieee Transactions on Robotics*, vol. 21, pp. 58-66, Feb 2005.
- [61] A. Safonova, J. K. Hodgins, and N. S. Pollard, "Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces," *Acm Transactions on Graphics*, vol. 23, pp. 514-521, Aug 2004.
- [62] H. Ko and N. I. Badler, "Animating human locomotion with inverse dynamics," *Ieee Computer Graphics and Applications*, vol. 16, pp. 50-59, Mar 1996.
- [63] O. Khatib, "A Unified Approach for Motion and Force Control of Robot Manipulators - the Operational Space Formulation," *Ieee Journal of Robotics and Automation*, vol. 3, pp. 43-53, Feb 1987.
- [64] B. Dariush, M. Gienger, B. Jian, C. Goerick, and K. Fujimura, "Whole body humanoid control from human motion descriptors," *2008 Ieee International Conference on Robotics and Automation, Vols 1-9*, pp. 2677-2684, 2008.
- [65] O. Khatib, E. Demircan, V. De Sapió, L. Sentis, T. Besier, and S. Delp, "Robotics-based synthesis of human motion," *Journal of Physiology-Paris*, vol. 103, pp. 211-219, May-Sep 2009.
- [66] V. De Sapió, J. Warren, O. Khatib, and S. Delp, "Simulating the task-level control of human motion: a methodology and framework for implementation," *Visual Computer*, vol. 21, pp. 289-302, Jun 2005.

- [67] O. Khatib, E. Demircan, V. De Sapia, L. Sentis, T. Besier, and S. Delp, "Robotics-based synthesis of human motion," *J Physiol Paris*, vol. 103, pp. 211-9, Sep-Dec 2009.
- [68] L. M. Nashner, A. Shumwaycook, and O. Marin, "Stance Posture Control in Select Groups of Children with Cerebral-Palsy - Deficits in Sensory Organization and Muscular Coordination," *Experimental Brain Research*, vol. 49, pp. 393-409, 1983.
- [69] J. Rose, D. R. Wolff, V. K. Jones, D. A. Bloch, J. W. Oehlert, and J. G. Gamble, "Postural balance in children with cerebral palsy," *Developmental Medicine and Child Neurology*, vol. 44, pp. 58-63, Jan 2002.
- [70] M. H. Woollacott and A. Shumway-Cook, "Postural dysfunction during standing and walking in children with cerebral palsy: what are the underlying problems and what new therapies might improve balance?," *Neural Plast*, vol. 12, pp. 211-9; discussion 263-72, 2005.
- [71] J. J. Jacobs, G. B. J. Andersson, J. E. Bell, S. L. Weinstein, J. P. Dormans, S. M. Gnatz, *et al.*, "The Burden of Musculoskeletal Diseases," *Bone and Joint Decade*, 2008.
- [72] R. V. Ootoole, B. Jaramaz, A. M. Digioia, C. D. Visnic, and R. H. Reid, "Biomechanics for Preoperative Planning and Surgical Simulations in Orthopedics," *Computers in Biology and Medicine*, vol. 25, pp. 183-191, Mar 1995.
- [73] B. Kayis and P. A. Iskander, "A three-dimensional human model for the IBM/CATIA system," *Appl Ergon*, vol. 25, pp. 395-7, Dec 1994.
- [74] M. M. Adamczyk and P. E. Crago, "Simulated feedforward neural network coordination of hand grasp and wrist angle in a neuroprosthesis," *IEEE Trans Rehabil Eng*, vol. 8, pp. 297-304, Sep 2000.
- [75] M. A. Lemay and P. E. Crago, "A dynamic model for simulating movements of the elbow, forearm, an wrist," *Journal of Biomechanics*, vol. 29, pp. 1319-30, Oct 1996.
- [76] B. J. Andrews, R. Davoodi, R. Kamnik, and T. Bajd, "Control of FES in paraplegia: modeling voluntary arm forces," *Biomed Mater Eng*, vol. 8, pp. 241-51, 1998.
- [77] G. E. Loeb, I. E. Brown, and E. J. Cheng, "A hierarchical foundation for models of sensorimotor control," *Exp Brain Res*, vol. 126, pp. 1-18, May 1999.
- [78] T. M. Barker, C. Kirtley, and J. Ratanapinunchai, "Calculation of multi-segment rigid body joint dynamics using MATLAB," *Proceedings of the Institution of Mechanical Engineers Part H-Journal of Engineering in Medicine*, vol. 211, pp. 483-487, 1997.
- [79] S. L. Delp, J. P. Loan, M. G. Hoy, F. E. Zajac, E. L. Topp, and J. M. Rosen, "An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures," *IEEE Trans Biomed Eng*, vol. 37, pp. 757-67, Aug 1990.
- [80] T. M. Kepple, K. L. Siegel, and S. J. Stanhope, "Relative contributions of the lower extremity joint moments to forward progression and support during gait," *Gait & Posture*, vol. 6, pp. 1-8, Aug 1997.
- [81] K. R. Holzbaur, W. M. Murray, and S. L. Delp, "A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control," *Ann Biomed Eng*, vol. 33, pp. 829-40, Jun 2005.
- [82] M. Damsgaard, J. Rasmussen, S. T. Christensen, E. Surma, and M. de Zee, "Analysis of musculoskeletal systems in the AnyBody Modeling System," *Simulation Modelling Practice and Theory*, vol. 14, pp. 1100-1111, 2006.
- [83] A. N. Vasavada, S. Li, and S. L. Delp, "Influence of muscle morphometry and moment arms on the moment-generating capacity of human neck muscles," *Spine (Phila Pa 1976)*, vol. 23, pp. 412-22, Feb 15 1998.

- [84] M. de Zee, L. Hansen, C. Wong, J. Rasmussen, and E. B. Simonsen, "A generic detailed rigid-body lumbar spine model," *Journal of Biomechanics*, vol. 40, pp. 1219-27, 2007.
- [85] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, *et al.*, "OpenSim: open-source software to create and analyze dynamic simulations of movement," *IEEE Trans Biomed Eng*, vol. 54, pp. 1940-50, Nov 2007.
- [86] S. L. Delp and J. P. Loan, "A graphics-based software system to develop and analyze models of musculoskeletal structures," *Comput Biol Med*, vol. 25, pp. 21-34, Jan 1995.
- [87] S. L. Delp and J. P. Loan, "A computational framework for simulating and analyzing human and animal movement," *Computing in Science & Engineering*, vol. 2, pp. 46-55, 2000.
- [88] M. D. Fox and S. L. Delp, "Contributions of muscles and passive dynamics to swing initiation over a range of walking speeds," *Journal of Biomechanics*, vol. 43, pp. 1450-5, May 28 2010.
- [89] R. R. Neptune, S. A. Kautz, and F. E. Zajac, "Contributions of the individual ankle plantar flexors to support, forward progression and swing initiation during walking," *Journal of Biomechanics*, vol. 34, pp. 1387-98, Nov 2001.
- [90] D. G. Thelen and F. C. Anderson, "Using computed muscle control to generate forward dynamic simulations of human walking from experimental data," *Journal of Biomechanics*, vol. 39, pp. 1107-15, 2006.
- [91] K. Sasaki and R. R. Neptune, "Muscle mechanical work and elastic energy utilization during walking and running near the preferred gait transition speed," *Gait & Posture*, vol. 23, pp. 383-390, Apr 2006.
- [92] M. D. Fox, J. A. Reinbolt, S. Ounpuu, and S. L. Delp, "Mechanisms of improved knee flexion after rectus femoris transfer surgery," *Journal of Biomechanics*, vol. 42, pp. 614-9, Mar 26 2009.
- [93] J. A. Reinbolt, M. D. Fox, M. H. Schwartz, and S. L. Delp, "Predicting outcomes of rectus femoris transfer surgery," *Gait & Posture*, vol. 30, pp. 100-5, Jul 2009.
- [94] R. Davoodi and G. E. Loeb, "A software tool for faster development of complex models of musculoskeletal systems and sensorimotor controllers in Simulink (TM)," *Journal of Applied Biomechanics*, vol. 18, pp. 357-365, Nov 2002.
- [95] J. Rasmussen and M. Ozen, "AnyBody-ANSYS Interface: CAE Technology for the Human Body.," *CADFEM Medical*, 2007.
- [96] M. Hauschild, R. Davoodi, and G. E. Loeb, "A virtual reality environment for designing and fitting neural prosthetic limbs," *Ieee Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, pp. 9-15, Mar 2007.
- [97] D. G. Thelen, F. C. Anderson, and S. L. Delp, "Generating dynamic simulations of movement using computed muscle control," *Journal of Biomechanics*, vol. 36, pp. 321-8, Mar 2003.
- [98] J. G. ZIEGLER and N. B. NICHOLS, "Optimum Settings for Automatic Controllers," *Trans. A.S.M.E*, vol. 64, pp. 759-768., 1942.
- [99] M. A. Sherman, Seth, A., Delp, S. L. , "Simbody: multibody dynamics for biomedical research," *Procedia IUTAM, International Union of Theoretical and Applied Mechanics, Elsevier Science*, 2011.
- [100] A. Seth, M. Sherman, J. Reinbolt, and S. Delp, "OpenSim: a musculoskeletal modeling and simulation framework for in silico investigations and exchange," *procedia IUTAM*,

- International Union of Theoretical and Applied Mechanics, Elsevier Science*, vol. 2, pp. 212-232, 2011.
- [101] A. Prochazka, D. Gillard, and D. J. Bennett, "Positive force feedback control of muscles," *Journal of Neurophysiology*, vol. 77, pp. 3226-3236, Jun 1997.
 - [102] H. Geyer, A. Seyfarth, and R. Blickhan, "Positive force feedback in bouncing gaits?," *Proc Biol Sci*, vol. 270, pp. 2173-83, Oct 22 2003.
 - [103] H. Geyer and H. Herr, "A Muscle-Reflex Model That Encodes Principles of Legged Mechanics Produces Human Walking Dynamics and Muscle Activities," *Ieee Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, pp. 263-273, Jun 2010.
 - [104] D. A. Kistemaker, A. J. K. Van Soest, J. D. Wong, I. Kurtzer, and P. L. Gribble, "Control of position and movement is simplified by combined muscle spindle and Golgi tendon organ feedback," *Journal of Neurophysiology*, vol. 109, pp. 1126-1139, Feb 2013.
 - [105] P. Brodal, *The central nervous system : structure and function*. New York: Oxford University Press, 1992.
 - [106] (2012). *Facts About Cerebral Palsy*. Available: <http://www.cdc.gov/ncbddd/cp/facts.html>
 - [107] M. M. van der Krogt, D. J. J. Bregman, M. Wisse, C. A. M. Doorenbosch, J. Harlaar, and S. H. Collins, "How Crouch Gait Can Dynamically Induce Stiff-Knee Gait," *Annals of Biomedical Engineering*, vol. 38, pp. 1593-1606, Apr 2010.
 - [108] H. X. Hoang and J. A. Reinbolt, "Crouched posture maximizes ground reaction forces generated by muscles," *Gait & Posture*, Apr 25 2012.
 - [109] M. D. Fox, J. A. Reinbolt, S. Ounpuu, and S. L. Delp, "Mechanisms of improved knee flexion after rectus femoris transfer surgery," *Journal of Biomechanics*, vol. 42, pp. 614-619, Mar 26 2009.
 - [110] K. Muthusamy, A. J. Seidl, R. M. Friesen, J. J. Carollo, Z. Pan, and F. M. Chang, "Rectus femoris transfer in children with cerebral palsy: evaluation of transfer site and preoperative indicators," *J Pediatr Orthop*, vol. 28, pp. 674-8, Sep 2008.
 - [111] N. Khouri and E. Desailly, "Rectus femoris transfer in multilevel surgery: Technical details and gait outcome assessment in cerebral palsy patients," *Orthopaedics & Traumatology-Surgery & Research*, vol. 99, pp. 333-340, May 2013.
 - [112] S. A. Rethlefsen, G. Kam, T. A. Wren, and R. M. Kay, "Predictors of outcome of distal rectus femoris transfer surgery in ambulatory children with cerebral palsy," *J Pediatr Orthop B*, vol. 18, pp. 58-62, Mar 2009.
 - [113] D. H. Sutherland, M. Santi, and M. F. Abel, "Treatment of Stiff-Knee Gait in Cerebral-Palsy - a Comparison by Gait Analysis of Distal Rectus Femoris Transfer Versus Proximal Rectus Release," *Journal of Pediatric Orthopaedics*, vol. 10, pp. 433-441, Jul-Aug 1990.
 - [114] B. I. Prilutsky, "Coordination of two- and one-joint muscles: functional consequences and implications for motor control," *Motor Control*, vol. 4, pp. 1-44, Jan 2000.
 - [115] G. J. V. Schenau, C. A. Pratt, and J. M. Macpherson, "Differential Use and Control of Monoarticular and Biarticular Muscles," *Human Movement Science*, vol. 13, pp. 495-517, Aug 1994.
 - [116] D. Metaxiotis, S. Wolf, and L. Doederlein, "Conversion of biarticular to monoarticular muscles as a component of multilevel surgery in spastic diplegia," *Journal of Bone and Joint Surgery-British Volume*, vol. 86B, pp. 102-109, Jan 2004.

- [117] J. R. Gage, P. A. Deluca, and T. S. Renshaw, "Gait Analysis: Principles and Applications Emphasis on its use in Cerebral Palsy," *The Journal of Bone and Joint Surgery*, vol. 77-A, pp. 1607-1623, 1995.
- [118] A. L. Hof, M. G. J. Gazendam, and W. E. Sinke, "The condition for dynamic stability," *Journal of Biomechanics*, vol. 38, pp. 1-8, Jan 2005.
- [119] A. Seth, M. Sherman, J. A. Reinbolt, and S. L. Delp, "OpenSim: a musculoskeletal modeling and simulation framework for in silico investigations and exchange," *Iutam Symposium on Human Body Dynamics*, vol. 2, pp. 212-232, 2011.
- [120] D. G. Thelen, "Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults," *J Biomech Eng*, vol. 125, pp. 70-7, Feb 2003.
- [121] A. Erdemir, P. A. Sirimamilla, J. P. Halloran, and A. J. van den Bogert, "An Elaborate Data Set Characterizing the Mechanical Response of the Foot," *Journal of Biomechanical Engineering-Transactions of the Asme*, vol. 131, Sep 2009.
- [122] J. L. Hicks, M. H. Schwartz, A. S. Arnold, and S. L. Delp, "Crouched postures reduce the capacity of muscles to extend the hip and knee during the single-limb stance phase of gait," *Journal of Biomechanics*, vol. 41, pp. 960-7, 2008.
- [123] D. S. Asakawa, S. S. Blemker, G. T. Rab, A. Bagley, and S. L. Delp, "Three-dimensional muscle-tendon geometry after rectus femoris tendon transfer," *J Bone Joint Surg Am*, vol. 86-A, pp. 348-54, Feb 2004.
- [124] P. Brodal, *The central nervous system : structure and function*, 4th ed. New York: Oxford University Press, 2010.
- [125] S. M. Henry, J. Fung, and F. B. Horak, "Control of stance during lateral and anterior/posterior surface translations," *IEEE Trans Rehabil Eng*, vol. 6, pp. 32-42, Mar 1998.
- [126] K. Jansen, F. De Groote, W. Aerts, J. De Schutter, J. Duysens, and I. Jonkers, "Altering length and velocity feedback during a neuro-musculoskeletal simulation of normal gait contributes to hemiparetic gait characteristics," *J Neuroeng Rehabil*, vol. 11, p. 78, Apr 30 2014.
- [127] M. Mansouri and J. A. Reinbolt, "A platform for dynamic simulation and control of movement based on OpenSim and MATLAB," *Journal of Biomechanics*, vol. 45, pp. 1517-21, May 11 2012.
- [128] F. E. Zajac and M. E. Gordon, "Determining muscle's force and action in multi-articular movement," *Exerc Sport Sci Rev*, vol. 17, pp. 187-230, 1989.
- [129] A. Clark, "Biarticular Muscles Influence Postural Responses," MSc, Mechanical, Aerospace, and Biomedical Engineering, Tennessee, 2012.
- [130] L. H. Ting, "Dimensional reduction in sensorimotor systems: a framework for understanding muscle coordination of posture," *Prog Brain Res*, vol. 165, pp. 299-321, 2007.
- [131] M. R. Feldesman, "Femur/stature ratio and estimates of stature in children," *Am J Phys Anthropol*, vol. 87, pp. 447-59, Apr 1992.
- [132] A. C. Alonso, N. M. S. Luna, L. Mochizuki, F. Barbieri, S. Santos, and J. M. D. Greve, "The influence of anthropometric factors on postural balance: the relationship between body composition and posturographic measurements in young adults," *Clinics*, vol. 67, pp. 1433-1441, 2012.
- [133] O. Khatib, "Inertial Properties in Robotic Manipulation - an Object-Level Framework," *International Journal of Robotics Research*, vol. 14, pp. 19-36, Feb 1995.

- [134] B. Dariush, G. Bin Hammam, and D. Orin, "Constrained Resolved Acceleration Control for Humanoids," *Ieee/Rsj 2010 International Conference on Intelligent Robots and Systems (Iros 2010)*, 2010.
- [135] L. Sentis and O. Khatib, "Control of free-floating humanoid robots through task prioritization," *2005 IEEE International Conference on Robotics and Automation (ICRA), Vols 1-4*, pp. 1718-1723, 2005.
- [136] J. M. Hollerbach and K. C. Suh, "Redundancy Resolution of Manipulators through Torque Optimization," *Ieee Journal of Robotics and Automation*, vol. 3, pp. 308-316, Aug 1987.
- [137] A. A. Maciejewski and C. A. Klein, "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," *International Journal of Robotics Research*, vol. 4, pp. 109-117, 1985.
- [138] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-Priority Based Redundancy Control of Robot Manipulators," *International Journal of Robotics Research*, vol. 6, pp. 3-15, Sum 1987.
- [139] L. Sentis and O. Khatib, "Prioritized multi-objective dynamics and control of robots in human environments," *2004 4th Ieee/Ras International Conference on Humanoid Robots, Vols 1 and 2, Proceedings*, pp. 764-780, 2004.
- [140] L. Sentis and O. Khatib, "A whole-body control framework for humanoids operating in human environments," *2006 Ieee International Conference on Robotics and Automation (Icra), Vols 1-10*, pp. 2641-2648, 2006.
- [141] K. H. Hunt and F. R. E. Crossley, "Coefficient of Restitution Interpreted as Damping in Vibroimpact," *Journal of Applied Mechanics-Transactions of the Asme*, vol. 42, pp. 440-445, 1975.
- [142] L. Sentis, "Synthesis and Control of Whole-Body Behaviors in Humanoid Systems," Doctorate, Stanford University, 2007.
- [143] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Dynamic multi-priority control in redundant robotic systems," *Robotica*, vol. 31, pp. 1155-1167, Oct 2013.
- [144] A. Lees, J. Vanrenterghem, G. Barton, and M. Lake, "Kinematic response characteristics of the CAREN moving platform system for use in posture and balance research," *Medical Engineering & Physics*, vol. 29, pp. 629-635, Jun 2007.
- [145] J. A. Adams, "Closed-Loop Theory of Motor Learning," *Journal of Motor Behavior*, vol. 3, pp. 111-150, 1971.
- [146] T. D. Welch and L. H. Ting, "A feedback model reproduces muscle activity during human postural responses to support-surface translations," *J Neurophysiol*, vol. 99, pp. 1032-8, Feb 2008.

VITA

Misagh Mansouri was born February 16, 1984 in Esfahan, Iran. He graduated from Mobin high school, Tehran, Iran in 2002. He received his Bachelors of Science in Mechanical Engineering from the Azad University, South Tehran Campus in 2006, and his first Masters of Science in Mechatronics Engineering from Sharif University of Technology in 2009. He then moved to United States in 2009, where he received his second Masters of Science in 2014 and his Doctor of Philosophy in May 2015 both in Mechanical Engineering from the University of Tennessee, Knoxville, USA.