# THE EFFECT OF MUSCLE AND KINEMATIC COMPLEXITY ON FEASIBLE FORCES AND MUSCLE ACTIVATIONS IN A MODEL OF THE HUMAN LEG

A Thesis
Presented to
The Academic Faculty

by

Daniel Michael Smith

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Mechanical Engineering in the
George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology
December 2016

# THE EFFECT OF MUSCLE AND KINEMATIC COMPLEXITY ON FEASIBLE FORCES AND MUSCLE ACTIVATIONS IN A MODEL OF THE HUMAN LEG

Approved by:

Dr. Lena Ting, Advisor
Coulter Department of Biomedical Engineering
Emory University & Georgia Institute of Technology

Dr. Jun Ueda
Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Dr. Harvey Lipkin
Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: 9 December 2016

To my wife, Kelly and our precious twins, Isabella & Gabriela

# ACKNOWLEDGEMENTS

As I reflect on the completion of this thesis, how grateful I feel for each individual who has contributed to this work in the slightest. I feel indebted in the best of ways to so many, but I would like to take this space to thank a few in particular.

First, thank you to my parents who sacrificed and worked so hard to raise their family in righteousness and love. Who I am is inseparable from their influence. Thank you for giving me the opportunities and encouragement to stretch myself and look outwards. My life is guided by the principles you taught. In particular, you taught me the value of and a love for knowledge, both explicitly and by example, that drives me still.

I would like to thank Steven Charles, my undergraduate research advisor and mentor. You first opened my eyes to the field of Neuromechanics and set me on this path. Your guidance and the care you have for others is inspirational. Thank you for opening so many doors.

To all the lab mates I have worked with, thank you for your support, feedback, encouragement, and all the fun times. In particular, thank you Hongchul Sohn for helping me as I began graduate school, for teaching me about modeling in neuromechanics, and passing on the knowledge and work from our modeling forbearers both within and without the lab.

To my advisor, Lena Ting, thank you for teaching me more than you will ever know. Thank you for giving me the opportunity to work on this project. You helped me develop as an engineer, a scientist, and a person. You have expanded my view of the study of human motor control, and helped me see science in its entirety in a new light.

Your experience and insight keep righting my course. I am forever grateful to your dedication to helping me succeed.

To my wife, Kelly, thank you for being with and supporting me every step of the way. I love you eternally. I would not be here without you. You motivate and guide me, and help me up every time that I fall. I treasure our time together, however limited by graduate school. This adventure is so much more fun with you by my side. Thank you for bearing our two angelic daughters, Isabella Elise and Gabriela Rose. The joy they bring to our family knows no bounds. Thank you, you two, as well.

To my Eternal Father, I express gratitude for my life and my chance to serve others for Thee. It is only through Thy guidance that this work ever began. There have been so many forks in the road that led me to this place, and I know without Thee I would not have had the wisdom nor the power to arrive here today. The grace of Thy Son is boundless supports and empowers me. Thank Thee for granting me the chance to serve Thy children in this way.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

Page

# LIST OF SYMBOLS AND ABBREVIATIONS

| | |
|---|---|
| CNS | Central Nervous System: the brain and spinal cord |
| FFS | Feasible Force Set: the set of all endpoint forces that are biomechanically feasible as determined by the anatomy, specified posture, and muscle parameters |
| FMAR | Feasible Muscle Activation Ranges: the range of activation levels a muscle can have and still maintain a certain specified mechanical output (e.g. force), allowing all other muscle activations to vary as necessary. |
| maxF-FMAR | Maximum Force-Feasible Muscle Activation Range: the FMAR for a specified muscle at the maximum force magnitude for a specified force direction |
| SML | Single Muscle Loss: excluding a single muscle from a model to represent the clinical condition where a muscle is injured or removed and to show how mechanical outputs may or may not change |
| SML-FFS | Single Muscle Loss Feasible Force Set: the feasible force set after SML is applied |
| gSML | General Single Muscle Loss: the total effect that SML can have on a given mechanical output if all muscle models are removed in turn |

## SUMMARY

Muscles move and stabilize the body. Motor deficits, regardless of their origin, decrease control of and/or coordination of muscles leading to reduced quality of life and independence. While rehabilitation interventions can improve motor control, the ability to improve upon these treatments is limited by an incomplete understanding of how muscle activity is coordinated. There is a debate over the degree that muscle coordination is determined by biomechanical versus neural constraints, which may have important implications for the type of rehabilitation an individual will receive. Musculoskeletal models can be used to study muscle redundancy, a key piece of the debate, but model complexity is inconsistent across the literature. This study aims to determine the effect of model complexity on redundancy and to identify which types of models should be used when studying muscle coordination.

One challenge for understanding muscle coordination is muscle redundancy, the presence of more muscles than necessary to perform a given task. Patterns of muscle activity may be dictated by biomechanics, that is the structure of the body and the mechanical laws that govern movement, or they may be controlled by neural strategies, and the degree to which muscle activity is determined by neural selection or by biomechanics is unresolved. Muscle redundancy allows for variations in the muscle activation patterns, but it is unknown how much variation this redundancy allows.

Contradictory results are found in the literature for quantifying redundancy: some studies suggest a large role for the nervous system in muscle coordination due to sizeable feasible variation, while others suggest biomechanics largely determine muscle coordination. Computational musculoskeletal models are useful tools for studying biomechanics and redundancy, but across the literature the level of detail in the models is inconsistent. We hypothesized that the contradictory results are due to different numbers of muscles and degrees-of-freedom (DoFs) in the models, and that models with more realistic numbers of muscle and DoFs allow for more variability in muscle coordination.

We tested our hypothesis by examining the role of individual muscles for a given task. A redundant muscle shares function with another muscle, and the more shared function the greater the redundancy. For the purposes of this study, we quantified a muscle's redundancy for a task by measuring the robustness of static force production to the loss of that muscle's function.

To examine the effect of model complexity on muscle redundancy, we systematically varied both the number of muscles and kinematic DoFs of a musculoskeletal model and tested the significance of individual muscles in each model by looking at 1) the sensitivity/robustness of static force production to single muscle loss via the set of biomechanically feasible forces (feasible force set, FFS) and 2) the feasible ranges of muscle activations (feasible muscle activation ranges, FMARs) for all maximum force in the sagittal plane.

We demonstrated that results from simplified models do not generalize to systems with more muscles and DoFs, and that the more detailed models suggest very few muscles are constrained by biomechanics. The effect of losing a single muscle on the FFS decreased as the number of independently-controlled muscles increased, indicating higher redundancy. FFSs in models with more DoFs were more sensitive to the loss of individual muscles than models with fewer and planar DoFs, but this effect was negligible when the complete set of muscles from the standard model were included. We also showed muscle activity is often unconstrained even at maximum forces; most muscles exhibited wide FMARs at maximum force in many or most force directions. Only a few muscles (the hip-knee biarticular muscles) were completely constrained for all maximum sagittal plane force directions. Further, we showed that the effects of complexity in muscles and DoFs observed in these cases are general for any musculoskeletal system.

When evaluating whether a musculoskeletal model is well-suited to study muscle redundancy, researchers should include in their considerations how well the number of

muscles in the model accurately represents the redundancy of what is being modeled, as well as the ratio of muscles to joints.

An understanding of the degree to which muscle activity is determined by biomechanics and/or by neural selection has significant implications for rehabilitation. Low levels of biomechanical constraints suggest many different neural strategies or compensations are feasible, indicating rehabilitation efforts should focus on training muscle coordination.

# CHAPTER 1 – INTRODUCTION

The musculoskeletal system is anatomically redundant (Bernstein, 1967), but the implications of redundancy for muscle coordination in movement is unresolved. Multiple joints create redundant sets of joint angles to bring an end effector to a certain location, the redundant musculature allows for a spectrum of muscle activation patterns that can create the necessary joint torques for a given task, and multiple neurons exciting each individual muscle mean the same muscle force can be produced by different patterns of neural activity. How a single muscle activation pattern is selected from among the range of redundant feasible solutions is a central question in the field of neuromechanics, the study of human control of movement from both neural and biomechanical perspectives.

## 1.1 Redundancy and musculoskeletal models

Modeling is a helpful tool for addressing questions about biomechanical redundancy. Biomechanical and neuromechanical studies often use musculoskeletal models that approximate anatomy by means of rigid bodies that represent bones, defined relationships between the bones with one or more degrees-of-freedom (DoF) to represent joints, and linear actuators to represent muscles. Models allow access to all state variables and parameters (e.g. muscle force, muscle activity, fiber length), and because muscles and/or joints can be added or removed in a model in ways not experimentally feasible, they provide a unique opportunity to study redundancy.

Optimization to find a single muscle activation pattern is a typical modeling approach to address the redundant set of muscle activation patterns that can produce the same whole-limb mechanical output, but there is often significant variation between experimental results and the optimal solution the model predicts. A solution that meets all biomechanical constraints and is optimal with respect to one or more optimization criteria may at times capture major features of experimentally observed behavior, but these results are highly task-specific and require a cost function that may not capture all the

intricacies of neural control (Crowninshield & Brand, 1981; Sohn & Ting, 2016; Thelen & Anderson, 2006). Experimentally observed and computationally optimized muscle activation patterns often differ significantly (Buchanan & Shreeve, 1996; Herzog & Leonard, 1991; van der Krogt, Delp, & Schwartz, 2012).

The variations between optimal and experimentally-observed muscle activation patterns, two implicitly-feasible solutions for the same mechanical output, show that redundancy in muscle activation does exist but cannot express how much redundancy exists. A single optimal solution cannot determine the limits of feasible variability in muscle activity indicated by the observed variations, giving no indication of how representative the optimal solution is for all feasible solutions (Simpson, Sohn, Allen, & Ting, 2015; Sohn, McKay, & Ting, 2013). Two distinct models could have the same optimal solution and very different amounts of redundancy.

Explicitly determining the bounds on feasible muscle activity would provide insight into the motor control of people with motor impairments where more variability is expected due to compensation strategies unlikely to be predicted by optimization. It is typical for individuals with motor impairments to adopt compensatory strategies to accommodate their deficits, and compensation depends on musculoskeletal redundancy. If optimization methods fail to predict muscle activity in healthy subjects, they will likely be worse at predicting the muscle activity in individuals with motor impairments.

### 1.2 Identifying biomechanical constraints and quantifying redundancy

Recent studies have begun to address redundancy by identifying the biomechanical constraints on muscle activity but results have been contradictory. Some findings suggest that biomechanics highly constrain redundancy (Kutch & Valero-Cuevas, 2011; Valero-Cuevas, Zajac, & Burgar, 1998) while others indicate that there is sufficient musculoskeletal redundancy such that biomechanics cannot completely determine observed behavior (Martelli, Calvetti, Somersalo, & Viceconti, 2015; Martelli,

Calvetti, Somersalo, Viceconti, & Taddei, 2013; McKay, Burkholder, & Ting, 2007; McKay & Ting, 2008; Sohn et al., 2013).

Redundancy can be quantified by looking at individual muscle's contribution to the system's mechanical output and how much of their capacity to produce joint torque is shared with other muscles. A muscle is redundant to the degree that its contribution to the limb's mechanical output and/or joint torque can be created by another muscle.

One method for expressing the biomechanical limits is calculating the set of forces that feasibly can be produced at a limb's endpoint (feasible force set, FFS) (Valero-Cuevas, 2000; Valero-Cuevas et al., 1998). Building on the methods of using computational geometry to map muscle activation patterns into limb mechanical outputs (Kuo & Zajac, 1993),Valero-Cuevas developed a method for mapping muscle activation pattern to endpoint force in a model and characterizing the set of all forces that were feasible (Valero-Cuevas, 2000). FFSs have now been studied in a human index-finger, both experimentally in a cadaver finger and with a model (Valero-Cuevas, 2000), in a human leg (Gruben, Lopez-Ortiz, & Schmidt, 2003), and in models of a simplified human leg (Kutch & Valero-Cuevas, 2011) and in a detailed cat hindlimb (McKay et al., 2007; McKay & Ting, 2008). FFS methods have not yet been applied to a human model with the current standard musculoskeletal model (Delp et al., 2007) with high levels of detail and complexity in the definitions of the DoFs and musculature.

Muscle redundancy has been directly tested in FFSs by examining the robustness of a FFS to loss of individual muscles (Kutch & Valero-Cuevas, 2011), but the method of computational geometry that was used is limited in applicability. Computational geometry provides an exact analytical solution, but the method is limited because the time required to find a solution increases exponentially with the number of variables. No more than fourteen variables can be feasibly used which in turn limits the complexity of the systems that can be studied. This may be appropriate for systems with few muscles, such as a finger model (4 DoF, 7 Muscles) (Valero-Cuevas et al., 1998), but a large

numbers of muscles from a human leg model were either removed or grouped in order to accommodate computational geometry (Kutch & Valero-Cuevas, 2011). The results from both the cadaveric finger and leg model used by Kutch and Valero-Cuevas (2011) demonstrated a general lack of robustness to single muscle loss and suggesting a low level of redundancy, but the simplification of the leg model may have compromised the generalizability of this claim.

Our lab has developed a heuristic method for calculating a FFS that is not limited by the number of independent variables and therefore allows for models with much greater level of detail (McKay et al., 2007; McKay & Ting, 2008).

Biomechanical limits on muscle activity can also be expressed as the feasible muscle activation range (FMAR) for an individual muscle for a given mechanical output, e.g. endpoint force, walking dynamics (Kutch & Valero-Cuevas, 2011; Simpson et al., 2015; Sohn et al., 2013). Again, results are contradictory, with finger muscle activity highly constrained at 50% maximum force (Kutch & Valero-Cuevas, 2011) while leg muscle activity is largely unconstrained in human and cat limbs, in both static and dynamic tasks (Simpson et al., 2015; Sohn et al., 2013). Other investigators have also developed methods for expressing feasible variability (Martelli et al., 2015; Martelli et al., 2013), but do not find the explicit bounds on muscle activity.

### 1.3 Motivation and Summary

No prior studies have directly tested the effect of model complexity on musculoskeletal redundancy. To fill this gap, we chose a model and calculated FFSs and FMARS while systematically varying the complexity of that model. We hypothesized that the contradictory results in the literature resulted from different levels of complexity in the models used, and that models with realistic numbers of muscles and joints will demonstrate wide variability in muscle coordination. For the purposes of this study, we quantified a muscle's redundancy for a task by measuring the robustness of endpoint

force to the loss of that muscle's function (single muscle loss, SML) (Kutch & Valero-Cuevas, 2011).

Redundancy may not be consistent across different species and appendages, so we chose to focus on a human leg model because of the leg's relevance to locomotion and balance and because of the diversity of leg models in the literature. To examine the effect of the number of independently-controlled muscles on redundancy, we removed the same muscles as Kutch and Valero-Cuevas (2011) and compared results with the intact model. To examine the effect of grouping muscles on redundancy, we grouped the same muscles as Kutch and Valero-Cuevas (2011) and compared results with both the intact model and model with muscles removed. To examine the effect of the model's kinematic structure on redundancy, we compared results from a sagittal plane version of the model with the results from the three-dimensional, 7 DoF model.

# CHAPTER 2 – METHODS

We applied two methods for evaluating the redundancy of a single muscle in force production using a detailed musculoskeletal model of the human leg. Our goal was to test the generality of the results from the simplified model used by Kutch and Valero-Cuevas (2011) in models with different levels of complexity to test if model complexity affects muscle redundancy. Musculoskeletal redundancy is affected by the number of muscles or DoFs in the model. We tested the hypothesis that models with more muscles and DoFs will be more redundant via six human leg models with different sets of muscles and DoFs, ranging from a simplified, planar model similar to Kutch & Valero's (2011) with 14 muscles and 3 DoFs to a state-of-the-art, 3D model with 43 muscles and 7 DoF that is typically used in biomechanical analysis. By systematically varying the number of muscles and DoFs in the model, we tested the effect of model complexity on muscle redundancy in static force production.

To evaluate the effects of model complexity on muscle redundancy, we defined muscle redundancy as the extent to which the feasible mechanical output of the system remained unchanged after losing that muscle's function. If a muscle is redundant, by definition it shares some function with one or more other muscles. If a redundant muscle's function was lost or impaired, the output associated with its function would not be affected or would only be partially affected.

We quantified the change in mechanical output by computing feasible force sets and feasible muscle activation ranges. First, we calculated the set of biomechanically feasible forces (feasible force set, FFS) and quantified the change in size of the sagittal plane FFS due to single muscle loss, and second, we calculated the ranges in which individual muscle activations can vary while still producing the same endpoint force (feasible muscle activation ranges, FMARs) and applied the method to the maximum forces in each direction from the intact FFS.

## 2.1 Musculoskeletal models

We used one leg from a generic OpenSim model of the human torso and lower extremities (gait2392_simbody.osim) as the most complex model in the study. OpenSim is an open-source software for modeling the musculoskeletal system and for simulating dynamic movements (Delp et al., 2007). This leg model is a detailed, three-dimensional musculoskeletal model composed of rigid bodies representing bones with a total of seven DoFs (3 at the hip, 1 at the knee, 2 at the ankle; 1 at the metatarsophalangeal (MTP) joint) and 43 muscles/muscle compartments (Table 2.1) modeled by line-paths with lengths dependent on the DoFs. A posture was selected that approximates one used in a previous study that included both experimental and model-based FFSs (Gruben et al. 2003) (hip flexion: 0.83849, hip adduction: 0, hip rotation: 0, knee angle: -0.91717, ankle angle: -0.58346, subtalar angle: 0, MTP angle: 0, all angles in radians). The pelvis is fixed in space, and the endpoint, which is defined as the MTP joint, is pinned to the ground via a gimbal joint.

The muscle models in used in OpenSim are of the structure presented by Zajac (1989) with a single input representing the collective excitation of the muscle and a single output: muscle-tendon force. The relationship between the input excitation and output muscle force is also a function of the state of the muscle, in particular the length and velocity of the muscle fibers. The kinematics of the muscle fibers are determined by posture (quantified by joint angles, $\vec{q}$) and estimated muscle-tendon-unit parameters, including optimal fiber length, tendon slack length, and pennation angle. The maximum active force a muscle fiber can produce depends on its length (Gordon, Huxley, & Julian, 1966). Optimal fiber length is the length at which the fiber can produce the greatest force. At fiber lengths longer or shorter than the optimal fiber length the force producing capacity is decreased. Each muscle-tendon unit is modeled as a line-path whose end points, and via points if applicable, are defined relative to two or more bones, such that the muscle-tendon unit length changes with changes in the DoF(s) it crosses. The current

fiber length is determined by the current muscle-tendon unit length, the resting length of the tendons, and the pennation angle, i.e. the angle between the line of action of the muscle fibers and the muscle-tendon unit line-path. Fiber lengths are then normalized with respect to the optimal fiber length to simplify the relationship and calculations. The force-velocity relationship of muscle fibers (Fenn & Marsh, 1935; Hill, 1938) was not pertinent to this study because the task studied was static.

### 2.1.1 Systematically varying model complexity

For comparison to the leg model FFSs from Kutch & Valero-Cuevas 2011, a simplified version of the complex OpenSim model was created with three key simplifications: only three sagittal plane joints were included (flexion/extension in the hip, knee, and ankle), a reduced set of the muscle models were used (26 of 43, Table 2.1), and most (two-thirds) of the muscles in the reduced set were made into groups by constraining their activations to be the same, leaving only 14 independent muscle activation control variables (Table 2.1).

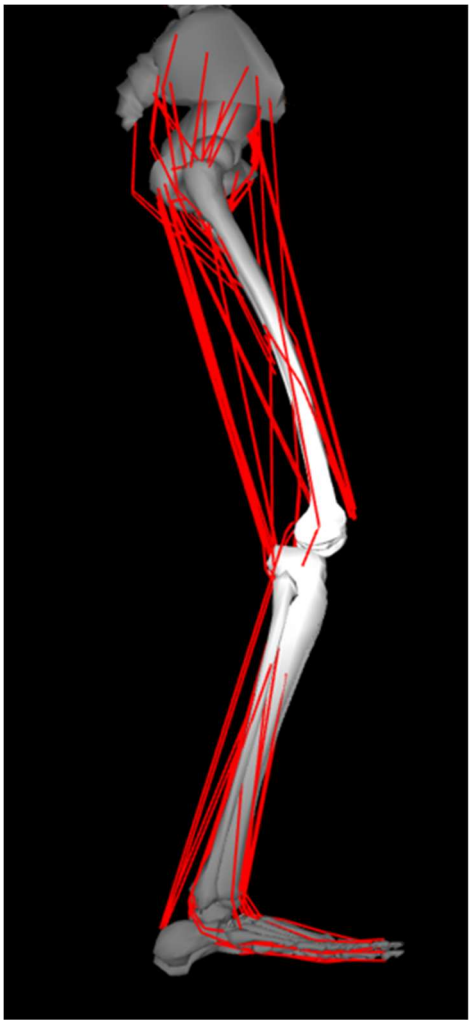


**Figure 2.1 – Anatomy & musculature of the OpenSim gait2392 lower limb model, Sagittal plane view**
Each leg has 43 muscle models, 7 DoFs, and 6 rigid bodies. Only one leg was considered for this study.

**Table 2.1 – Muscles from the OpenSim gait2392 model and organization of muscle model complexity**
Muscles in the OpenSim gait2392 musculoskeletal model and their abbreviations. The three columns of abbreviations represent the low, intermediate, and high levels of muscle model complexity (i.e. number of independent muscles). The Lo-Muscle model has a reduced set of muscles and many muscles are grouped together to act as a single muscle, that is, controlled by a single activation variable. The Int-Muscle model has the same reduced set of muscles as the Lo-Muscle model, but all muscles have independent control. The Hi-Muscle model has the complete set of muscles and all muscles have independent control.

| Muscle Name | Abbreviation | | | | | |
|---|---|---|---|---|---|---|
| | "Lo-Muscle" Model | | "Int-Muscle" Model | | "Hi-Muscle" Model | |
| | 14 Muscles | | 26 Muscles | | 43 Muscles | |
| Gluteus medius anterior | 1 | GMedMin | 1 | GMED1 | 1 | GMED1 |
| Gluteus medius middle | 2 | | 2 | GMED2 | 2 | GMED2 |
| Gluteus medius posterior | 3 | | 3 | GMED3 | 3 | GMED3 |
| Gluteus minimus anterior | 4 | | 4 | GMIN1 | 4 | GMIN1 |
| Gluteus minimus middle | 5 | | 5 | GMIN2 | 5 | GMIN2 |
| Gluteus minimus posterior | 6 | | 6 | GMIN3 | 6 | GMIN3 |
| Semimembranosus | 7 | HAM | 7 | SEMIMEM | 7 | SEMIMEM |
| Semitendinosus | 8 | | 8 | SEMITEN | 8 | SEMITEN |
| Biceps femoris long head | 9 | | 9 | BFLH | 9 | BFLH |
| Biceps femoris short head | 10 | BFSH | 10 | BFSH | 10 | BFSH |
| Sartorius | | | | | 11 | SAR |
| Adductor longus | 12 | ADDL | 12 | ADDL | 12 | ADDL |
| Adductor brevis | | | | | 13 | ADDBREV |
| Adductor magnus superior | | | | | 14 | ADDMAG1 |
| Adductor magnus middle | | | | | 15 | ADDMAG2 |
| Adductor magnus inferior | | | | | 16 | ADDMAG3 |
| Tensor fascia latae | 17 | TFL | 17 | TFL | 17 | TFL |
| Pectineus | | | | | 18 | PECT |
| Gracilis | | | | | 19 | GRAC |
| Gluteus maximus superior | 20 | GMax | 20 | GMAX1 | 20 | GMAX1 |
| Gluteus maximus middle | 21 | | 21 | GMAX2 | 21 | GMAX2 |
| Gluteus maximus inferior | 22 | | 22 | GMAX3 | 22 | GMAX3 |
| Iliacus | 23 | ILIAC | 23 | ILIAC | 23 | ILIAC |
| Psoas | | | | | 24 | PSOAS |
| Quadratus femoris | | | | | 25 | QUADF |
| Gemellus | | | | | 26 | GEM |
| Piriformis | | | | | 27 | PIRI |
| Rectus femoris | 28 | RF | 28 | RF | 28 | RF |
| Vastus medialis | 29 | VAS | 29 | VM | 29 | VM |
| Vastus intermedius | 30 | | 30 | VI | 30 | VI |
| Vastus lateralis | 31 | | 31 | VL | 31 | VL |
| Gastrocnemius medial head | 32 | GAS | 32 | MEDGAS | 32 | MEDGAS |
| Gastrocnemius lateral head | 33 | | 33 | LATGAS | 33 | LATGAS |
| Soleus | 34 | SOL | 34 | SOL | 34 | SOL |
| Tibialis posterior | 35 | TP | 35 | TP | 35 | TP |
| Flexor digitorum longus | | | | | 36 | FLEXD |
| Flexor hallucis longus | | | | | 37 | FLEXH |
| Tibialis anterior | 38 | TA | 38 | TA | 38 | TA |
| Peroneus brevis | 39 | PBREV | 39 | PBREV | 39 | PBREV |
| Peroneus longus | | | | | 40 | PLONG |
| Peroneus tertius | | | | | 41 | PTERT |
| Extensor digitorum longus | | | | | 42 | EXTD |
| Extensor hallucis longus | | | | | 43 | EXTH |

To study the effects of each of these key differences, six models were created with different levels of detail and completeness in musculature and DoFs. Three (low, intermediate, and high) levels of musculature completeness and detail were used: a reduced set of muscles with grouped control (Lo-Muscle), a reduced set of muscles with independent control (Int-Muscle), the complete set of muscles with independent control (Hi-Muscle) (Table 1). Two levels of DoF complexity were used: three, sagittal plane joints (Lo-DoF) and all seven multi-planar DoFs (Hi-DoF).

We also created an alternative intermediate muscle model (alt-Int-Muscle) that included all the muscle models but still had muscle groups for completeness in testing the effects of muscle grouping and number of muscles. The results from alt-Int-Muscle did not affect the general message of the results or the general effects of muscles and DoFs, and therefore were included only in Appendix A.

### 2.1.2 Equations of motion

The complete set of equations of motion for the leg model come from the generalized equations of motion for a musculoskeletal limb with any number, n, of muscles (nMusc) or DoFs (nDoF). The generalized torque-space equations of motion can be expressed in matrix form as

$$\boldsymbol{M}(\vec{q})\ddot{\vec{q}} = \boldsymbol{R}(\vec{q})\left[\boldsymbol{F}^{M}_{act}(\vec{q},\dot{\vec{q}})\vec{a} + \vec{F}^{M}_{pas}(\vec{q})\right] + \vec{V}(\vec{q},\dot{\vec{q}}) + \vec{G}(\vec{q}) + \boldsymbol{J}^{T}(\vec{q})\overrightarrow{W}_{end} \qquad (1)$$

where both sides of the equation are equal to $\vec{\tau}$, the vector of joint torques, and

- $\boldsymbol{M}(\vec{q})$ is the (nDoF x nDoF) posture dependent, symmetric mass-inertia matrix.
- $\vec{q}, \dot{\vec{q}},$ and $\ddot{\vec{q}}$ are the (nDoF x 1) vectors of generalized coordinates of the model and its first and second time derivatives, respectively, a.k.a. the vector of joint angles, velocities, and accelerations.
- $\boldsymbol{R}(\vec{q})$ is the (nDoF x nMusc) matrix of joint-angle-dependent moment arms of each muscle relative to the joint or joints they articulate.

- $\boldsymbol{F}_{act}^{M}(\vec{q}, \dot{\vec{q}})$ is a (nMusc x nMusc) diagonal matrix of the maximum active muscle force each muscle can produce for the given muscle fiber length and velocity, as determined by the joint angles. By definition, the maximum static force a muscle can produce is when the fiber is at the optimal fiber length.
- $\vec{a}$ is the (nMusc x 1) muscle activation vector representing the normalized activation level (from 0 to 1) of each muscle, neglecting whether activation comes from motor unit recruitment or increased motor neuron firing rate.
- $\vec{F}_{pas}^{M}(\vec{q})$ is the (nMusc x 1) vector of passive muscle forces that arise when a muscle's fibers are stretched beyond the optimal fiber length, and is therefore highly dependent on joint angles.
- $\vec{V}(\vec{q}, \dot{\vec{q}})$ is the (nDoF x 1) vector of terms that include joint angular velocity, $\dot{q}_i$, e.g. Coriolis force.
- $\vec{G}(\vec{q})$ is the (nDoF x 1) vector of joint torques created by gravity acting on each body segment.
- $\boldsymbol{J}$ is the (nDoF x 6) Jacobian matrix which maps the end-point wrench (with an element for each of the 6 spatial DoFs) into the nDoF resultant joint torques.
- $\overrightarrow{W_{end}}$ is an (6x1) externally applied "wrench," a.k.a. force-moment pair, at the endpoint of the limb of the form $\left[F_x, F_y, F_z, M_x, M_y, M_z\right]^T$.

Simplified Equations of Motion

For our specific application of static force production, key simplifications to the equations of motion were made resulting in a linear mapping from muscle activation to joint torques. We set $\vec{G} = 0$ because we are interested in the feasible forces independent of gravity, and $\dot{\vec{q}} = \ddot{\vec{q}} = 0$ because the task is static. The passive muscle forces were neglected because the posture had no extreme joint angles where these forces play a large role. Finally, the endpoint moment was constrained to zero because endpoint moments

can only be created when the endpoint is fixed to ground, an atypical condition for most tasks. Limiting the endpoint moment to zero effectively turned the endpoint wrench, $\vec{W}_{end}$, into an endpoint force, $\vec{F}_{end}$, where $\vec{F}_{end} = [F_x, F_y, F_z, 0, 0, 0]^T$. We also redefined the direction of $\vec{F}_{end}$ as the force produced by the limb, rather than externally applied to it ($\vec{F}_{end,new} = -\vec{F}_{end,old}$). By making these changes, and removing the dependencies on $\vec{q}$ for visual clarity, we are left with

$$\boldsymbol{J}^T \vec{F}_{end} = \boldsymbol{R}\, \boldsymbol{F}^M_{act}\, \vec{a}. \tag{2}$$

*Model Parameters*

The values in the matrices in the equations of motion, or the values used to calculate them, were found using OpenSim (Delp et al., 2007) and Neuromechanic (Bunderson, Bingham, Hongchul Sohn, Ting, & Burkholder, 2012). We calculated the active muscle force matrix, $\boldsymbol{F}^M_{act}$, using the maximum isometric forces, the active force-length relationship, and the pennation angles at the given posture, which we calculated using a selection of muscle parameters from the OpenSim body file. We extracted from the model the muscle tendon unit length, maximum isometric force, optimal fiber length, tendon slack length, and the pennation angle at optimum fiber length for each of the muscles. We then calculated the fiber length for each muscle at this posture (Zajac 1989) and the normalized fiber length, which we fed into the active force-length relationship used by Thelen (2003). The elements of moment arm matrix, $\boldsymbol{R}$, i.e. the posture-dependent moment arms of each muscle with respect to each DoF were calculated by OpenSim and communicated via an application programming interface (API) to MATLAB. To calculate the Jacobian, $\boldsymbol{J}$, we used a converted version of the OpenSim leg model for the neuromusculoskeletal modeling platform Neuromechanic which has a function for explicitly calculating the Jacobian for any specified endpoint with respect to the pelvis.

## 2.2 Single muscle loss feasible force sets

### 2.2.1 Feasible force sets

A FFS is the set of all biomechanically feasible endpoint forces that can be produced by varying each individual muscle's activity for a given anatomy, posture, and set of muscle parameters (Valero-Cuevas et al. 1998, Valero-Cuevas 2000). We took methods previously developed in our lab (McKay et al., 2007; McKay & Ting, 2008) and adapted them to accommodate a generalized neuromechanical model with any number of DoFs and muscles without needing a pseudoinverse, and applied it to a model of a human, at all the levels of complexity described in section 2.1.1 Systematically varying model complexity. We calculated our FFSs using our simplified equations of motion for each model complexity (eq. 2), allowing independent control of all muscles, i.e. allowing each element of the excitation vector $\vec{e}$ to vary independently. FFSs by definition are three-dimensional, but this study only looked at the sagittal plane FFS for direct comparability with previous work.

To calculate a FFS directly, the Jacobian would need to be inverted such that

$$\vec{F}_{end} = \boldsymbol{J^{-T}RF}^{M}_{act}\vec{a}, \tag{3}$$

which can be only be directly computed if the Jacobian is invertible. In practice, this means the model must have either 6 DoFs for 3D applications or 3 DoFs for planar applications. For other applications, when the Jacobian is not a square matrix, $\vec{F}_{end}$ can only be isolated by computing a pseudoinverse for the Jacobian (McKay & Ting, 2008, 2012) but pseudoinverses are not unique and use optimization to select one matrix from among the solution space, potentially reducing the range of endpoint force vectors reachable by the muscle activation vector which would introduce a false deterministic relationship between those vectors. To solve the general case of any number of DoFs without using a pseudoinverse, we converted the equations of motion from force space into torque space,

$$\vec{\tau}_{act} \equiv \boldsymbol{R}\boldsymbol{F}_{act}^{M}\vec{a}\,, \tag{4}$$

where $\vec{\tau}_{act}$ is the vector of actual joint torques created by the active muscle forces, which, by equation (2), is equivalent to the resulting joint torques for a specified endpoint force

$$\vec{\tau}_{act} = \boldsymbol{J}^{T}\vec{F}_{end}. \tag{5}$$

To find a single maximum feasible force in a desired direction, a single unit force vector, $\hat{F}_{des}$, is defined and multiplied by the transposed Jacobian to create a desired joint torque vector,

$$\vec{\tau}_{des} \equiv \boldsymbol{J}^{T}\hat{F}_{des}. \tag{6}$$

In order to find the maximum feasible joint torque in that direction, we used numerical optimization to minimize the following cost function

$$\max_{\vec{a}} \; |\vec{\tau}_{act} \cdot \vec{\tau}_{des}| \;\; s.t. \; \vec{\tau}_{act} \,||\, \vec{\tau}_{des}, \tag{7}$$

with $\vec{a}$ (used to calculate $\vec{\tau}_{act}$) as the optimization variable, and then repeated the method with 360 equally spaced $\hat{F}_{des}$ vectors to sample the sagittal plane. The linearity constraint was necessary to ensure consistent sampling of FFS. Linearity constraints can be implemented mathematically in several ways, but there are limitations of certain methods. The cross-product is commonly used to determine or constrain collinearity,

$$\vec{\tau}_{des} \times \vec{\tau}_{act} = \vec{0}, \tag{8}$$

but this is only possible when nDoF is 3 or 7 because the cross product that maintains the basic properties including orthogonality can only be defined for 3 and 7 dimensional vectors (Massey, 1983). To develop a general method for implementing collinearity of two vectors of any equal dimension, we expressed collinearity as

$$\vec{\tau}_{act} = k\,\vec{\tau}_{des}, \tag{9}$$

where $k$ is a scalar. Because $\vec{\tau}_{des}$ is specified prior to the optimization, we used its elements to create a constraint that can be implemented using linear programming for any nDoF. By inverting each element of $\vec{\tau}_{des}$ and multiplying it by $\vec{\tau}_{act}$, the solution is a vector where each element is $k$. However, $k$ cannot be explicitly included in the constraint equation because its value depends on the direction of $\vec{\tau}_{des}$ and is unknown before the optimization. The explicit reference to the value of k can be removed from the constraint by pre-multiplying the vector of $k$'s by a matrix to calculate the difference between each of the elements and constraining it to zero.

$$\left[\begin{array}{c|cccc} 1 & -1 & 0 & \cdots & 0 \\ 1 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & -1 \end{array}\right] \begin{bmatrix} 1/\tau_{des,1} & 0 & \cdots & 0 \\ 0 & 1/\tau_{des,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1/\tau_{des,nDoF} \end{bmatrix} \vec{\tau}_{act} = \vec{0} \quad (10)$$

When any element $\tau_{des,i}$ was equal to zero, the collinearity constraint only demands that $\tau_{act,i} = 0$ because $0 = k * 0$. This simplification was implemented by replacing the $i^{th}$ row of the linearity constraint matrix with the $i^{th}$ row of the identity matrix of the appropriate dimension. This allowed us to avoid the mathematical impossibility of dividing by zero.

Each FFS was defined as the convex polygon that contained all directions of $\vec{F}_{end}^{MAX}$ as determined by the *convhull.m* function in MATLAB (McKay & Ting, 2008), where $\vec{F}_{end}^{MAX}$ was found by multiplying the unit force direction by the ratio of the norms of the actual torque vector and the desired torque vector,

$$\vec{F}_{end}^{MAX} = \|\vec{\tau}_{act}\| / \|\vec{\tau}_{des}\| \, \hat{F}_{des}, \quad (11)$$

for all 360 sets of $\hat{F}_{des}$, $\vec{\tau}_{des}$, and $\vec{\tau}_{act}$.

### 2.2.2 Single muscle loss

We tested the muscular redundancy of the models by calculating FFSs and quantifying the change in FFS area due to removing a single muscle from the muscle set

(single muscle loss, SML) and repeating the process for all muscles. Kutch and Valero-Cuevas (2011) previously tested the effects of muscle dysfunction on a FFS in a simplified model of the human leg but were limited by their method of computational geometry. Computational geometry cannot not be feasibly applied to models with more than 14 free variables due to the exponential increase in computation time required by systems, thereby excluding the standard OpenSim models.

We implemented SML by constraining each muscle's activation level one-at-a-time to be zero and recalculating the FFS. The resulting FFSs from models with SML (SML-FFS) represent feasible variability in muscle activity by showing all forces where a given muscle in unnecessary.

To determine the effect of SML on a FFS, the FFSs from an intact model and the SML-FFSs were compared in terms of robustness and sensitivity. Robustness of the FFS to SML was defined as the percent area that was unaffected by loss of that muscle, i.e. the overlapping areas between the intact-FFS and the SML-FFS (Figure 3.1, green areas). Sensitivity of the FFS to SML, a.k.a. the effect the loss of a single muscle has on the FFS, was defined as the percent area lost after loss of that muscle, i.e. the non-overlapping areas of the intact- and SML-FFSs (Figure 3.1, blue areas). The sum of the sensitivity and the robustness of a FFS to single muscle loss is, by definition, 100% ($sensitivity + robustness = 1$). An increase in muscle redundancy and feasible variability for muscle coordination would be indicated by an increase in FFS robustness to SML and a decrease in FFS sensitivity to SML.

Robustness of the FFS to general single muscle loss (gSML) was defined as the percent area of the FFS unaffected by the loss of any single muscle, considered one at a time, i.e. the intersection of all the SML-FFSs which are equal in number to the number of muscles in the intact model (nMusc).

When considering different complexity of models, it was important to consider the changing size of the total FFS area along with its changing sensitivity and robustness. The sagittal plane FFS areas were calculated, were normalized to the Lo-Muscle/Lo-DoF FFS area to aid with intuition, and then compared between models.

### 2.3 Feasible muscle activation ranges at maximum force

Using the maximum forces from the FFS of the intact model (no muscle loss), we identified the lower and upper bounds on each muscle's activity at maximum force as a function of direction to test the redundancy of muscles in a maximal task. A FMAR is the range of activation levels a muscle can have and still maintain a certain specified mechanical output (e.g. force), letting all other muscle activations vary as necessary (Kutch & Valero-Cuevas 2011, Sohn et al. 2013, Simpson et al. 2015). FMARs can be determined (FMAR width = 0), undetermined (FMAR width > 0), or unconstrained (FMAR width = 1). This study looked at the FMARs at maximum force magnitude (maxF-FMARs) in all directions in the sagittal plane.

We substituted into equation (2) the maximum endpoint forces calculated in the intact FFS

$$\boldsymbol{J}^T \vec{F}_{end}^{MAX} = \boldsymbol{R}\,\boldsymbol{F}_{act}^{M}\,\vec{a}, \qquad (12)$$

leaving only $\vec{a}$ as a free variable, and used linear programming to find the upper and lower bounds on each individual muscle's activation (Sohn et al. 2013). The optimization technique found the minimum and then the maximum values of each element of $\vec{a}$, one at a time, letting the remaining nMusc - 1 elements of $\vec{a}$ vary as necessary, that satisfied equation **Error! Reference source not found.**),

$$\min_{\vec{a}} a_i \quad \text{s.t.} \quad \boldsymbol{J}^T \vec{F}_{end}^{MAX} = \boldsymbol{R}\,\boldsymbol{F}_{act}^{M}\,\vec{a}. \qquad (13)$$

$$\min_{\vec{a}} -a_i \quad \text{s.t.} \quad \boldsymbol{J}^T \vec{F}_{end}^{MAX} = \boldsymbol{R}\,\boldsymbol{F}_{act}^{M}\,\vec{a}. \qquad (14)$$

for all $i$ from 1 to nMusc

and repeated the method for all directions of $\vec{F}_{end}^{MAX}$.

# CHAPTER 3 – RESULTS

## 3.1 The robustness of static force production to general single muscle loss increases as model complexity increases

Intact FFSs were qualitatively similar to previous reports in both humans (Gruben et al. 2003, Kutch & Valero-Cuevas 2011) and in animals (McKay et al., 2007) in that they were roughly elliptical with the axis approximately in line with the axis of the limb and the peak forces were directed distally from the endpoint (Figure 3.1, blue areas). The area of the FFS in the Lo-Muscle/Lo-DoF model was $8.35\times10^5$ $N^2$. To facilitate comparison, the area of the FFSs were normalized to the area of FFS in the Lo-Muscle/Lo-DoF, the leg model most similar to the one used by Kutch and Valero-Cuevas (2011). In general, FFS area was greater in models with more independent muscles (Figure 3.1, left to right) and was smaller in models with more DoFs (Figure 3.1, top to bottom).

FFS robustness to general single muscle loss (gSML) increased as the number of independent muscles increased (Figure 3.1, green areas). gSML is defined as the percent area of the FFS robust to the loss of any individual muscle and was highly dependent on the number and grouping of muscles in the model. The robustness of the FFS with Lo-Muscle/Lo-DoF complexity increased significantly when muscles were ungrouped (Lo-Muscle/Lo-DoF: 7.2%, Int-Muscle/Lo-DoF: 40.6%, Figure 3.1A&B) with very little increase in FFS area (1.029, normalized to the area of the Lo-Muscle/Lo-DoF FFS area). The percent of the area robust to gSML increased further when the remainder of the muscles were included (Hi-Muscle/Lo-DoF: 52.3%, Figure 3.1C) despite the substantially increased total FFS area (1.521 normalized).

Models with more DoFs had less robust FFSs to gSML, but this effect is mitigated by the increasing complexity of the sets of muscles (Figure 3.1). For all sets of muscles, the robustness of the FFS to gSML was less in models with Hi-DoF than in

those with Lo-DoF (compare Figure 3.1ABC with Figure 3.1DEF), but decrease in FFS robustness to gSML was less in models with more independent muscles (percent decrease in robust area in Lo-DoF vs. Hi-DoF in i) Lo-Muscle: 90.3%, Figure 3.1A&D, ii) Int-Muscle: 35.7%, Figure 3.1B&E, and iii) Hi-Muscle: 6.1%, Figure 3.1C&F). The FFSs were approximately 50% robust to gSML in both models with Hi-Muscle, regardless of the number of DoFs included in the model.

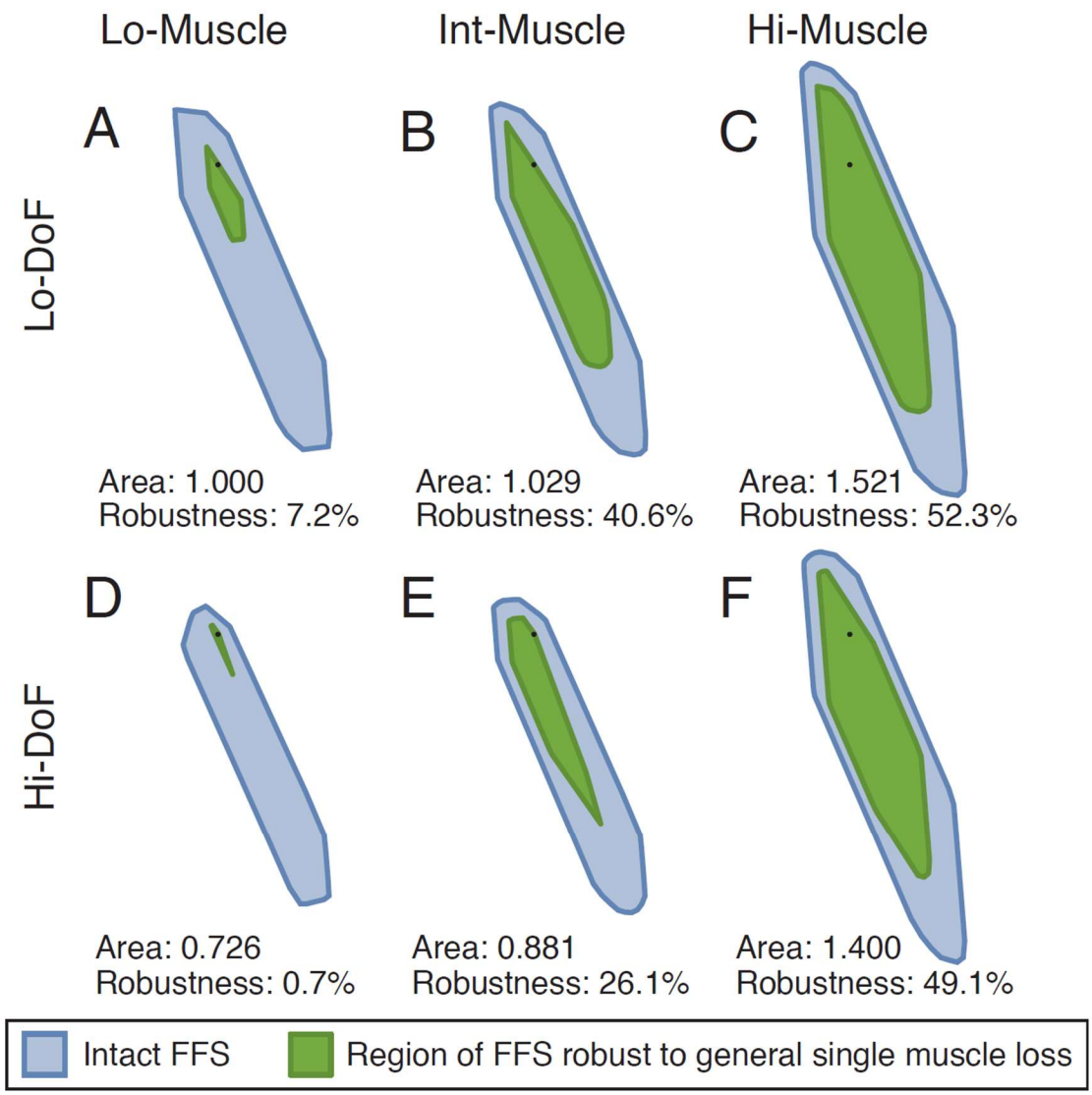



**Figure 3.1 – Intact feasible force sets and their robustness to general single muscle loss in leg models that vary in complexity and size of sets of muscles and degrees of freedom.** Intact FFSs (blue regions) and robust areas of FFS to gSML (green regions) in six different leg models. The robust region to gSML is defined as the area where no individual muscle is necessary. The FFSs in the top row were created with models all of which had three planar DoFs (Lo-DoF), but different sets of muscles. The models used to create these FFSs were (A) Lo-Muscle/Lo-DoF, a model created to replicate Kutch and Valero-Cuevas (2011) with 14 independent muscles and 3 planar DoFs, (B) Int-Muscle/Lo-DoF, a model with ungrouped muscles from Lo-Muscle resulting in 26 independent muscles, (C) Hi-Muscle/Lo-DoF, a model with the complete and ungrouped set of muscles from OpenSim model gait2392. Figures (D), (E), and (F), follow the same pattern of muscle models as (A), (B), and (C), but use the Hi-DoF model with seven, 3D DoFs.

### 3.2 The sensitivity of static force production to single muscle loss decreases as model complexity increases

FFS sensitivity to SML was greater in the most simplified model (Lo-Muscle/Lo-DoF) than in the OpenSim model (Hi-Muscle/Hi-DoF). FFS sensitivity to SML was calculated for each muscle, and the muscles were ordered within each model from low sensitivity (min: 0.0% in Lo-, Int-, and Hi-Muscle/Lo-DoF and Int- and Hi-Muscle/Hi-DoF, max: 0.3% in Lo-Muscle/Hi-DoF) to high sensitivity (min: 25.7% in Hi-Muscle/Lo-DoF, max: 74.5% in Lo-Muscle/Hi-DoF) (Figure 3.2). The sensitivity values corresponding to each muscle (Figure 3.2, blue bars) represent the percent of the area of the FFS affected by the loss of that muscle (Figure 3.2, blue areas in subset figures). The distributions of the sensitivities to SML were quantified by dividing the distributions into quartiles and recording the values of sensitivity associated with each quartile. These values demonstrated a lower FFS sensitivity to SML in Hi-Muscle/Hi-DoF compared to Lo-Muscle/Lo-DoF. The value of the 75$^{th}$ percentile in the Hi-Muscle/Hi-DoF model was 7.7% which means only 25% of muscles in that model influenced the FFS by 7.7% or more with a maximum effect of 25.8%. Conversely, 50% of the muscles in the Lo-Muscle/Lo-DoF model had an effect than 9.0% with a maximum of 68.1%.

As the number of independent muscle models increased, the sensitivity values associated with each quartile of FFS sensitivity to SML decreased (Figure 3.2, left to right), except when the 0th percentile was already at 0.0% sensitivity. The maximum FFS sensitivity to SML (100th percentile) decreased as the number of independent muscles increased, both by ungrouping muscles (from Lo-Muscle to Int-Muscle: 68.1% to 27.7% in Lo-DoF, Figure 3.2A&B, and 74.5% to 41.6% in Hi-DoF, Figure 3.2D&E) and by including the complete set of muscles (Hi-Muscle: 25.7% in Lo-DoF, Figure 3.2C, and 25.8% in Hi-DoF, Figure 3.2F). The same decreasing trend was followed in the 75th, 50th, and 25th percentiles. Of note, the 75th percentile in the Hi-Muscle/Hi-DoF model

was 7.7% while the 75th percentile in the Lo-Muscle/Lo-DoF model was 23.6%. The minimum FFS sensitivity to SML (0th percentile) was 0.0% in all models, except Lo-Muscle/Hi-DoF: 0.3%.

FFSs were more sensitive to SML in models with Hi-DoF than with Lo-DoF, (compare Figure 3.2ABC with Figure 3.2DEF), but this effect was counteracted by the number of independent muscles. The sensitivity values associated with each quartile was higher in Hi-DoF than in Lo-DoF models when the set of muscles was held constant, and Lo-Muscle/Hi-DoF was the model most sensitive to SML (Figure 3.2D, $0^{th}$, $25^{th}$, $50^{th}$, $75^{th}$, and $100^{th}$ percentiles: 0.3%, 6.4%, 18.7%, 49.8%, and 74.5%). However, the distributions of FFS sensitivity to SML in Hi-Muscle/Lo-DoF and Hi-Muscle/Hi-DoF were almost identical despite changing DoF complexity (the differences in the sensitivity values of the quartiles between the Hi-Muscle/Lo-DoF and Hi-Muscle/Hi-DoF models was ≤0.6%, Figure 3.2C&F).

Some muscles had a drastically different effect on the FFSs of Lo-Muscle/Hi-DoF and Int-Muscle/Hi-DoFs than on other models. In all Lo-DoF models, the FFSs were least sensitive to the loss of PBREV (muscle #39, Table 2.1) (0.0% sensitive, Figure 3.2ABC) and in the current standard model (Hi-Muscle/Hi-DoF) it was in the bottom five of 43 muscles (0.2% sensitive, Hi-Muscle/Hi-DoF, Figure 3.2F). However, in Lo-Muscle/Hi-DoF and Int-Muscle/Hi-DoF, both the absolute and relative sensitivity of the FFS to PBREV were very high. In Lo-Muscle/Hi-DoF, the FFS was forth most sensitive to PBREV (49.8% sensitive, Figure 3.2D), and in Int-Muscle/Hi-DoF, PBREV was the muscle that the FFS was most sensitive to losing (41.6%, Figure 3.2E).

The individual muscles put into groups by (Kutch and Valero-Cuevas (2011)) (Lo-Muscle) were among the muscles which had the greatest effect on the FFSs in the muscle models with more independent muscles. The 3 muscles that had the largest effect on the Lo-Muscle/Lo-DoF FFS, i.e. the top 21%, are all grouped muscles made from a total of 9 muscles in the Int- and Hi-Muscle models. These 9 muscles, when considered

individually, are in the top 17 muscles (top 40%) that had the largest effect on the Hi-Muscle/Hi-DoF FFS. Four of the five and six of the eight muscles to which the Hi-Muscle/Hi-DoF FFS is most sensitive are also from those nine individual muscles (Figure 3.2A&F).

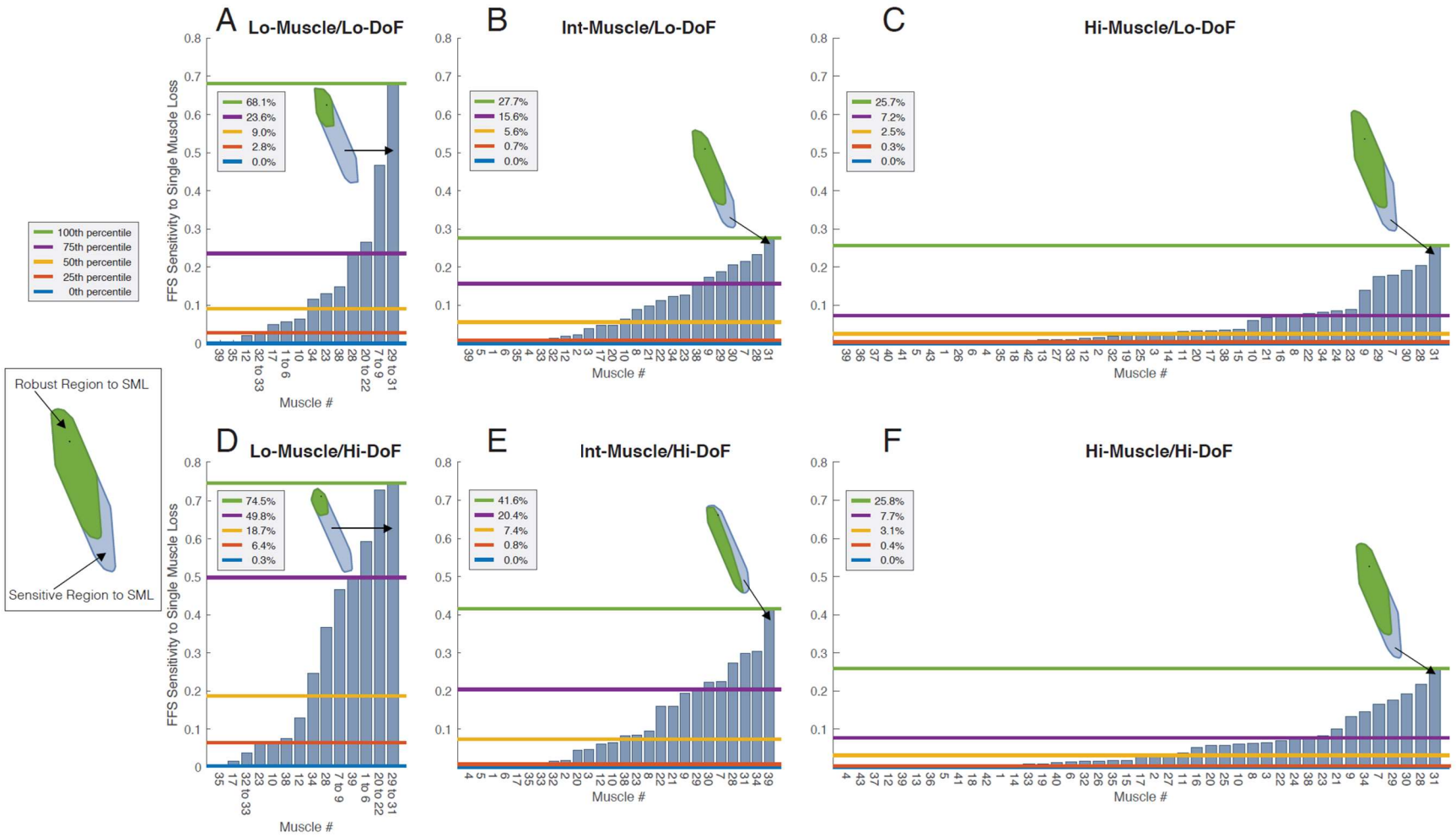



**Figure 3.2 – The distribution of FFS sensitivity to SML in six leg models that vary in complexity and size of sets of muscles and degrees of freedom y**
The same models and organization of models as used in Figure 3.1 are used. The FFS sensitivity (measured in percent) to all muscles in a given model are presented (blue bar graphs). FFS sensitivity to a muscle is defined as the percent reduction of the FFS due to loss of that muscle, as shown by the blue regions in the inset SML-FFSs in each section of the figure, all of which show the SML-FFS for the most sensitive muscle in their respective models. Each of the 5 quartiles (0%, 25%, 50%, 75%, and 100%) are labeled and color coded. The muscles are labeled with their index as defined in Table 2.1.

### 3.3 Grouping muscles increased FFS sensitivity to single muscle loss

FFS sensitivity to loss of the grouped muscles in Lo-Muscle (selected muscles shown in Figure 3.3, first column, blue areas) was roughly the same as the FFS sensitivity to loss of the corresponding group of individual muscles in Int-Muscle (Figure 3.3, second column, blue areas) but much greater than the FFS sensitivity to loss of the corresponding muscles individually (Figure 3.3,). The Lo-Muscle/Lo-DoF FFS was highly sensitive to loss of Vasti (Figure 3.3A, 68.1 %), but when compared with the Int-Muscle/Lo-DoF FSS its sensitivity visually and quantitatively matched the sensitivity to VM, VI, and VL as "triple muscle loss" (Figure 3.3B, 67.3%). rather than FFS sensitivity to SML for the three vastus muscles individually (VM: 18.9%, VI: 20.6%, VL: 27.7%, Figure 3.3C) which had, on average, a 67.1% reduced effect on the FFS. The same pattern was found in the other grouped muscles from Lo-Muscle (HAM: Figure 3.3DE&F, all others: not pictured). The sensitivity of the Lo-Muscle/Lo-DoF FFS to HAM matched the sensitivity of the Int-Muscle/Lo-DoF FFS to loss of SEMIMEM, SEMITEN, and BFLH (46.6% and 46.6% sensitive, Figure 3.3D&E) but not sensitivity to SML in those three muscles (SEMIMEM: 21.3%, SEMITEN: 8.8%, BFLH: 17.3% sensitive, Figure 3.3F). The sensitivity of the Lo-Muscle/Lo-DoF FFS to the remaining three muscle groups, GMedMin, GMax, and Gastroc, was 5.7%, 26.6%, and 2.8%, and when these groups of muscles were independently controlled in the Int-Muscle/Lo-DoF FFS, their influence was even less (not pictured).

The sensitivity of the Lo-Muscle/Lo-DoF FFS to grouped muscles was approximately equal to the sum of Int-Muscle/Lo-DoF FFS sensitivity to the individual ungrouped muscles (compare Figure 3.3A with C, and D with F). The difference was likely accounted for by slight differences in both total FFS area and relative function of individual muscles between the two models as demonstrated by comparing the slight changes in the FFS sensitivity to SML in a non-grouped muscle between the two models (e.g. Soleus, muscle #34: 14.9% vs 15.6% sensitive, Figure 3.2A&B).

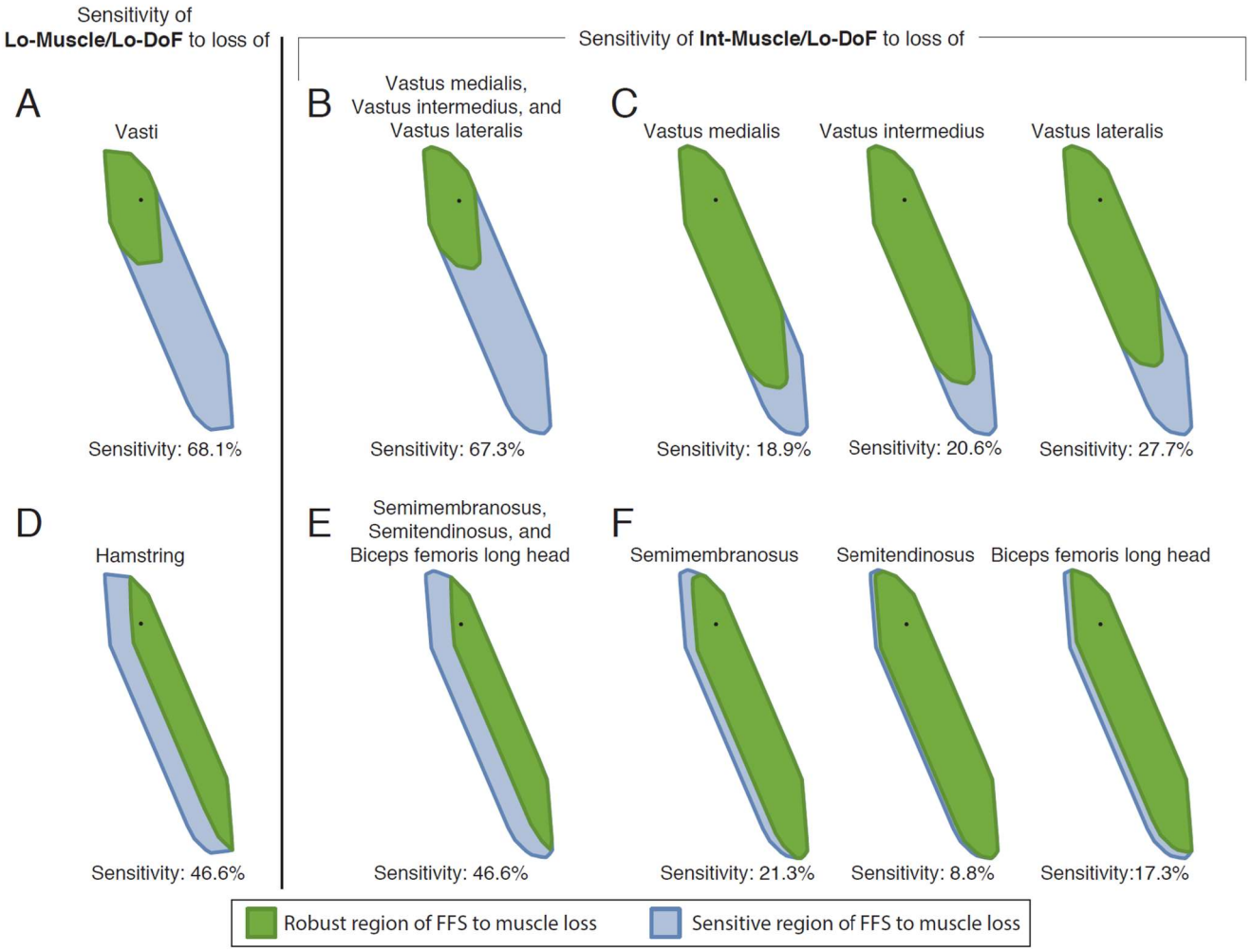



**Figure 3.3 – FFS sensitivity to SML of two selected muscles in both Lo-Muscle and Int-Muscle muscle sets**
A comparison of FFS sensitivities to (A) and (D) SML in two muscles, VAS & HAM from Lo-Muscle formed from muscle groups, (B) and (E) loss of the entire group of corresponding independent muscles in Int-Muscle, (C) and (F) SML in each of the corresponding muscles in Int-Muscle. The portion of the intact FFS (blue) visible beneath the SML-FFS (green) is the sensitive region of the FFS to muscle loss, grouped or single.

### 3.4 Variability in feasible patterns of muscle coordination exists at maximum force magnitudes

The maxF-FMARs for all directions were plotted on a polar plot (Figure 3.4). The inner black circles represent an activation value of 0 and the outer black circles represent an activation value of 1. The maxF-FMAR for each direction in the sagittal plane is shown in the corresponding radial line in the plots in Figure 3.4, e.g. the FMAR for the maximum force in the anterior direction is shown on the horizontal-right radius between the black circles. A solid green line indicates a determined maxF-FMAR, while a green shaded area indicates an undetermined maxF-FMAR or potentially an unconstrained maxF-FMAR if the green shaded area covers the entire area between the inner and outer black circles.

Undetermined maxF-FMARs were found in most muscles for many force directions indicating feasible variation in muscle patterns at maximum force (Figure 3.4). maxF-FMARs were largely insensitive to which set of muscles were included in the Lo-DoF models. The only changes in model complexity with considerable effects were when some of the grouped muscle models were ungrouped (comparing Lo-Muscle with Int-Muscle) and when the full set of DoFs were included prior to the inclusion of the full set of muscles (Lo-Muscle/Hi-DoF and Int-Muscle/Hi-DoF).

The directions in which maxF-FMARs where determined (FMAR width = 0) or undetermined (FMAR width >0), did not significantly differ in models of different complexity (Figure 3.4). The percentage of the FFS with undetermined maxF-FMARs in at least one muscle was high in all models and only slightly increased due to increased model complexity (Lo-Muscle/Lo-DoF: 85.8%, Hi-Muscle/Hi-DoF: 89.4%). Most undetermined maxF-FMARs (width >0) were also unconstrained (width=1). Most muscles had many force directions in which their maxF-FMARs were undetermined, and the average percentage of directions with undetermined maxF-FMARs across muscles

was not substantially affected by model complexity (Lo-Muscle/Lo-DoF: 43.9%, Int-Muscle/Lo-DoF: 37.5%, Hi-Muscle/Lo-DoF: 45.9%, Hi-Muscle/Hi-DoF: 43.4%). Only a few muscles, the hip-knee biarticular muscles, were fully determined at maximum sagittal plane force directions (Lo-Muscle: 3 of 14 muscles, Int-Muscle: 5 of 26 muscles, Hi-Muscle: 7 of 43 muscles, in both Lo-DoF & Hi-DoF, Figure 3.4F&G show two examples).

Ungrouping muscles had a greater effect on maxF-FMAR width in undetermined directions than on whether the directions in which maxF-FMARs were determined (zero) or undetermined (non-zero). The maxF-FMARS for the Vasti (grouped) and vastus (ungrouped) muscles are a representative sample. Although Vasti had undetermined maxF-FMARs in about half of the sagittal FFS, the widths were very narrow (Lo-Muscle/Lo-DoF: force directions with unconstrained FMARs: 55.0%, average FMAR width in unconstrained directions: 0.11, Figure 3.4A, row1). Ungrouping the vastus muscles had almost no effect on which force directions had undetermined maxF-FMARs, but the widths grew to several times that of the grouped muscles (Int-Muscle/Lo-DoF: force directions with unconstrained FMARs: 55.6%, average FMAR width in unconstrained directions: VM 0.605, VI 0.573, VL 0.465, Figure 3.4A, row 2). The pattern continued when the full set of DoFs and muscles were both included (Hi-Muscle/Hi-DoF: force directions with unconstrained FMARs: 50.8%, average FMAR width in unconstrained directions: VM 0.779, VI 0.719, VL 0.601, Figure 3.4A, row 4).

The Hi-DoF model when paired with the Lo-Muscle and Int-Muscle models resulted in highly-constrained, unrealistic patterns in the maxF-FMARs (Figure 3.4, rows 4 and 5). For example, ADDL, a hip adductor with a primary action outside the sagittal plane, has wide and mainly unconstrained maxFMARs for many directions particularly posterior directions (left on the polar plot) in most models, but in Lo-Muscle/Hi-DoF, the ADDL maxF-FMARs are determined for all force directions (Figure 3.4 row 4). ADDL in Lo-Muscle/Hi-DoF also exhibits unrealistic behavior in an inferior-anterior force

aligned for the maximum forces of the FFS (see Figure 3.1) by claiming ADDL is constrained to have 100% activation for a force directed only a few degrees away on either side from forces where it is constrained to have 0% activation. The average percentage of directions with undetermined maxF-FMARs across muscles was much less in Lo-Muscle/Hi-DoF and Int-Muscle/Hi-DoF than in other models (Lo-Muscle/Hi-DoF: 29.7%, Int-Muscle/Hi-DoF: 31.5%, all others above 50%).


**Figure 3.4 – MaxF-FMARs for 7 selected muscles across model complexity**
The maxF-FMARs in 7 selected muscles models (columns) in six models of varying complexity (rows). Each radial line between the two black circles corresponds to the same direction in the sagittal plane (right: anterior, up: superior). The inner black circle represents a feasible activation of zero and the outer black circle represents a muscle activation of one or "maximum". The maximum and minimum feasible activations for the maximum force in each direction in the sagittal plane are plotted in green on the corresponding radius of the polar plot. The space between them, if any, is filled in (green) to demonstrate the width of the FMAR for a maxF in that direction.

# CHAPTER 4 – DISCUSSION

## 4.1 Effects of modeling complexity on musculoskeletal redundancy

Our results demonstrate that musculoskeletal redundancy is highly sensitive to model complexity. The standard model (Delp et al., 2007) was highly redundant and demonstrated substantial variability in feasible muscle activity in both maximal and submaximal forces, while the most simplified models' muscle activity were highly constrained by biomechanics.

### 4.1.1 Comparing the model specific effects of varying joint and muscle complexity

Reducing the number of independently-controlled muscles, whether by removing muscles or grouping them, artificially reduces the redundancy of the system. Removing muscles reduces both the overall strength of the model (i.e., size of the FFS) and robustness to single muscle loss. Removing muscles removed 180° or more of the force directions from the robust regions of the FFS (in the anterior-superior direction), indicating that one or more muscles that contribute to force in that direction were removed in this simplification, artificially decreasing the robustness of those forces to single muscle loss.

Grouping muscles in the model drastically reduced the robustness of the FFSs. Not only does this artificially reduce muscle redundancy, but it does not represent a biomechanical constraint but rather a neural constraint since grouping muscles is implemented by constraining that they all are controlled by a single activation value.

In contrast to the redundancy-reducing simplification in the muscles, reducing the joints or DoFs of the model artificially increased the robustness of the leg to SML. Locking joints to create planar models misrepresents muscles' function, even for planar tasks. One of the most impactful simplifications in the planar model was modeling the hip as a single DoF rather than a ball-and-socket joint with three orthogonal DoFs. Fixing

two of the three axes at hip over-simplified and changed the function of the hip muscles, all of which have some effect about all three DoFs. For example, while tensor fascia latae (TFL) has a significant moment arm for hip flexion and therefore plays a role in sagittal plane forces, clinically we observe TFL mostly participate in hip abduction, a DoF locked by in the planar model.

### 4.1.2 Guidelines for model selection with regards to redundancy

First, including a more complete set of muscles should be prioritized over a complete set of DoFs when a model cannot incorporate both. While the models with the 3D set of joints had smaller, less robust FFSs, including the full set of independent muscles from the OpenSim model mitigated this effect. The results from the Hi-Muscle/Lo-DoF model were significantly more similar to those from the standard Hi-Muscle/Hi-DoF model than either of the other Hi-DoF models. The FMARs for models with high complexity in DoFs but simplified musculatures showed unrealistic, over-constrained results (Figure 3.4, rows 4 and 5). However, based on the heuristic methods presented, we do not foresee a need to choose between including muscles or DoFs.

Second, model redundancy can be estimated with the ratio of muscle to joints: a muscles:DoF ratio of close to 2 will likely be highly constrained by biomechanics. Since muscles only pull, for each joint to be fully actuated it needs two opposing muscles. Consider the Lo-Muscle/Hi-DoF model, which has a muscle-to-joints ratio of exactly 2: the model is 99.3% sensitive to gSML (Figure 3.1) and half of the muscles affect 20% or more of the FFS (Figure 3.2). Additionally, the index-finger model from the literature has a ratio of 1.75 and is also highly constrained biomechanically (Kutch & Valero-Cuevas, 2011; Valero-Cuevas, 2000; Valero-Cuevas et al., 1998). However, this metric should only be used as an initial estimate, mainly to rule out models that are too simple, for it be confounded by planar vs 3D models, depending on the task. 3D models with lower ratios may be more robust to gSML for a planar task than planar models with higher ratios

(compare Lo-Muscle/Lo-DoF, ratio of 4.7, with Int-Muscle/Hi-DoF, ratio of 3.7, in Figure 3.1) and models with drastically different ratios may have essentially equal robustness to gSML for a planar task if one model is planar and the other is 3D (compare Hi-Muscle/Lo-DoF, ratio of 14.3, with Hi-Muscle/Hi-DoF, ratio of 6.1, in Figure 3.1 and 3.2).

### 4.2 Generalized effects of joints and muscles on force production

The effect of model complexity in joints and muscles on static force production shown here generalizes to all musculoskeletal models in all postures: in all cases, removing a joint will either increase the FFS or leave it unchanged, while removing a muscle will decrease the FFS or leave it unchanged.

Locking a joint will either increase the maximum force in a given direction or leave it unchanged. In a static system, the relative magnitude of all joint torques is determined by the endpoint force direction and the moment arm between each joint and the endpoint, and that ratio of all the joint torques will remain consistent as it is scaled by changes in endpoint force magnitude. Also, every joint has a maximum torque value that can be produced about it in both directions, determined by the strength and moment arms of the muscles that cross it. The maximum endpoint force in a given direction will always correspond to one joint (possibly more) reaching its maximum torque capacity. If the joint limiting that force was locked and therefore able to provide infinite torque, the maximum force could then increase until a different joint reached its maximum. However, if a joint that was not the limiting joint for the force was locked, there would be no change in the maximum force magnitude in that direction. Inversely, if a different joint was created or unlocked, depending on the joint torque capacity and the moment arm, the force maximum could only either decrease or remain the same.

Removing a muscle will decrease the maximum force in a given direction or leave it unchanged. Muscle activity produces joint torque in this model, which omits passive

muscle force or ligaments. If a muscle is removed from a joint that was the limiting factor in determining maximum force in a given direction, the torque capacity of that joint would then be decreased, reducing the magnitude of the maximum force value. Adding a muscle to that joint could result in an increase of force magnitude depending on the direction of the torque that new muscle could produce. Changing the muscles about joints that are not limiting factors for the maximum force in a given direction would change redundancy, but no the magnitude of the maximum force.

Intersecting joints and muscles that cross more than one joint create more complexity in these results, but the general principles related to joint torque capacity and endpoint force magnitude remain unchanged.

### 4.3 Implications for Redundancy

Our results show that the standard model is highly redundant, implying that biomechanics are insufficient to determine muscle activity in static force production. When the biomechanics of the task leave ample room for variability in muscle activation, the CNS may select muscle activation patterns from within the FMARs based on other criteria such as generalizability, stability, or resistance to fatigue (Bunderson, Burkholder, & Ting, 2008; Loeb, 2000; Sohn & Ting, 2016).

While the current standard musculoskeletal model does not capture all of the complexities of the human leg muscles and joints, it captures more of the redundancy of an actual human leg compared to a planar model with a reduced set of muscles. This model approaches the complexity of the human leg and is sufficient to establish our claims of model generality and redundancy. Even with modeling errors, the model demonstrates that biomechanics are insufficient to determine muscle activity. If the model fidelity was increased, this finding would be even stronger: redundancy and the resultant insufficiency of biomechanics to determine muscle activation patterns would increase.

Different limbs or appendages in the body may have different amounts of redundancy and therefore are could be more or less susceptible to impairment or possibly controlled differently by the CNS. Evidence suggests that index fingers are highly biomechanically constrained, but the finger is not representative of the rest of the body in terms of numbers of muscles and joints. The muscles that actuate the fingers are all in either the palm or the forearm with generally long tendons. Having non-self-contained muscles lends itself to cadaveric studies of the finger muscles because the muscles are accessible from outside the finger, but this is not a common feature in the body. The ratio of muscles-to-joints is much higher in larger limbs, and limbs have many self-contained muscles.

Musculoskeletal redundancy does not mean that some muscles are unnecessary, but suggests that the degree of multi-functionality of bodies requires a musculature that is redundant at many single-task levels.

## 4.4 Limitations and future work

### 4.4.1 Planar vs. three-dimensional force directions

To be able to compare planar and three-dimensional models, the current study used only sagittal plane forces. The planar models could not produce forces outside the sagittal plane, which further detracts from their generality. The detailed models can produce FFSs in outside of the sagittal plane.

Future work may create a better representation of muscle redundancy by investigating forces in all dimensions. In that case, we predict that muscles' variability will shift, that muscles that had no feasible variability in muscle activity at maximum force in all sagittal force directions would begin to have feasible variability outside the sagittal plane, and that some unconstrained muscles in the sagittal plane will become constrained, indicating that no muscle is completely constrained in all force production,

even at maximum force. Similar trade-offs would likely happen in FFS sensitivity to SML.

### 4.4.2 Dynamic force production and changing postures

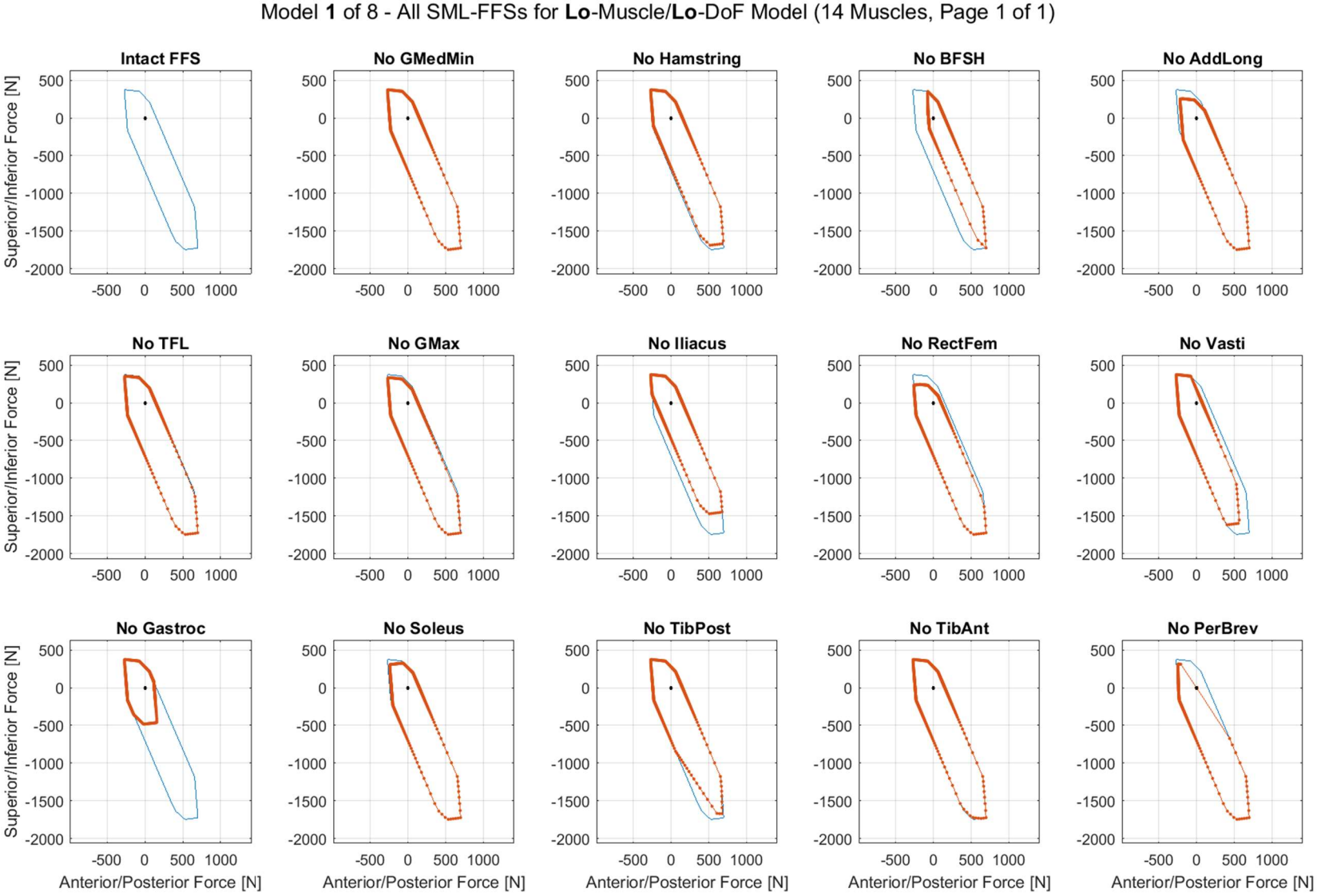
Dynamic tasks may apply more biomechanical constraints on muscle activity variation than seen with static force tasks. However, wide FMARs have been demonstrated in human gait (Simpson et al., 2015), but this redundancy could be further tested by applying SML or for more maximal tasks.

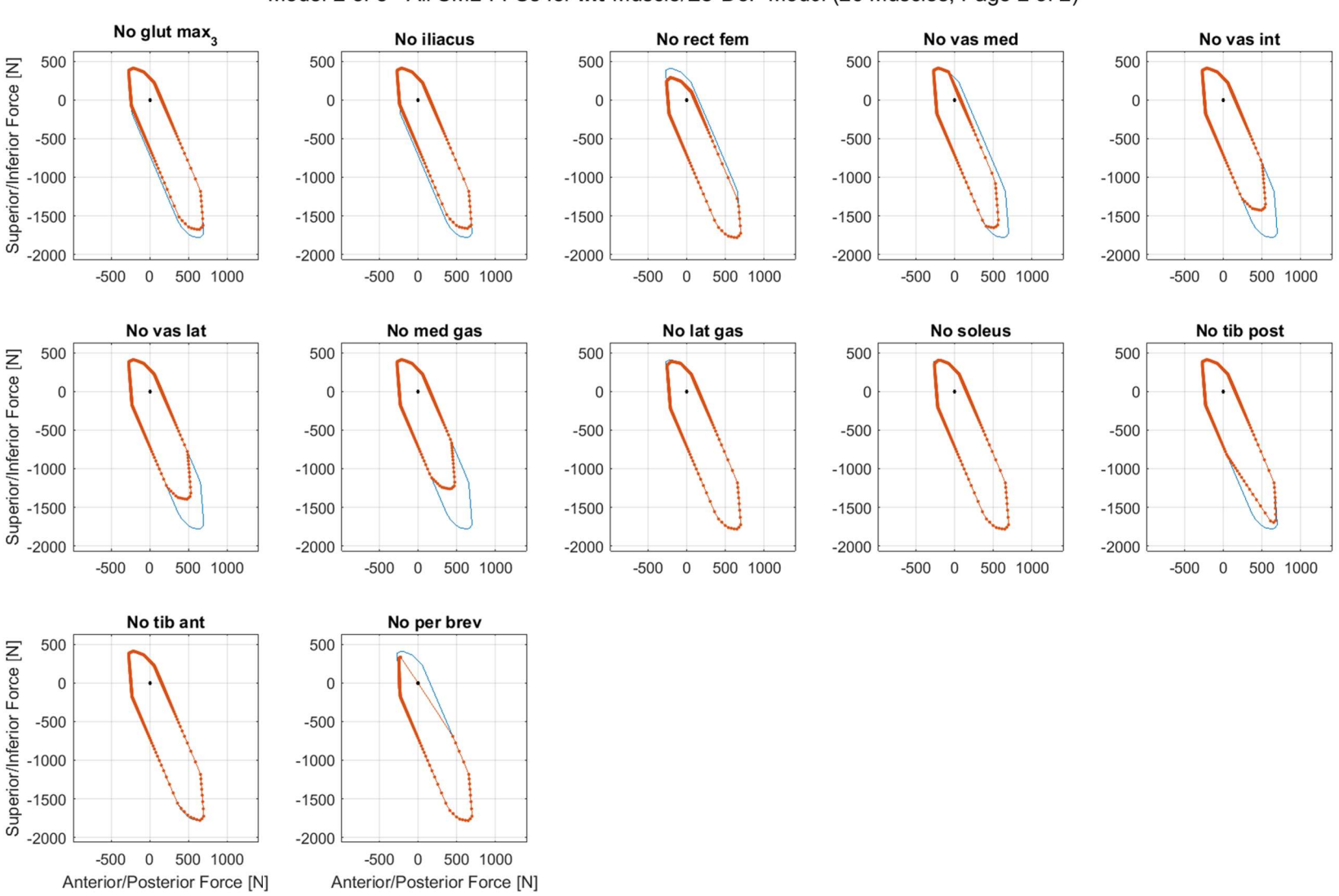
In static force production, the relationship between joint torque and muscle activations was linear (see equation (2)). A dynamic task will not have the same linear mapping (see equation (1)) which will provide new challenges. Furthermore, many of the model parameters are posture specific, and any dynamic task considered in more than one time step will require updating these parameters.

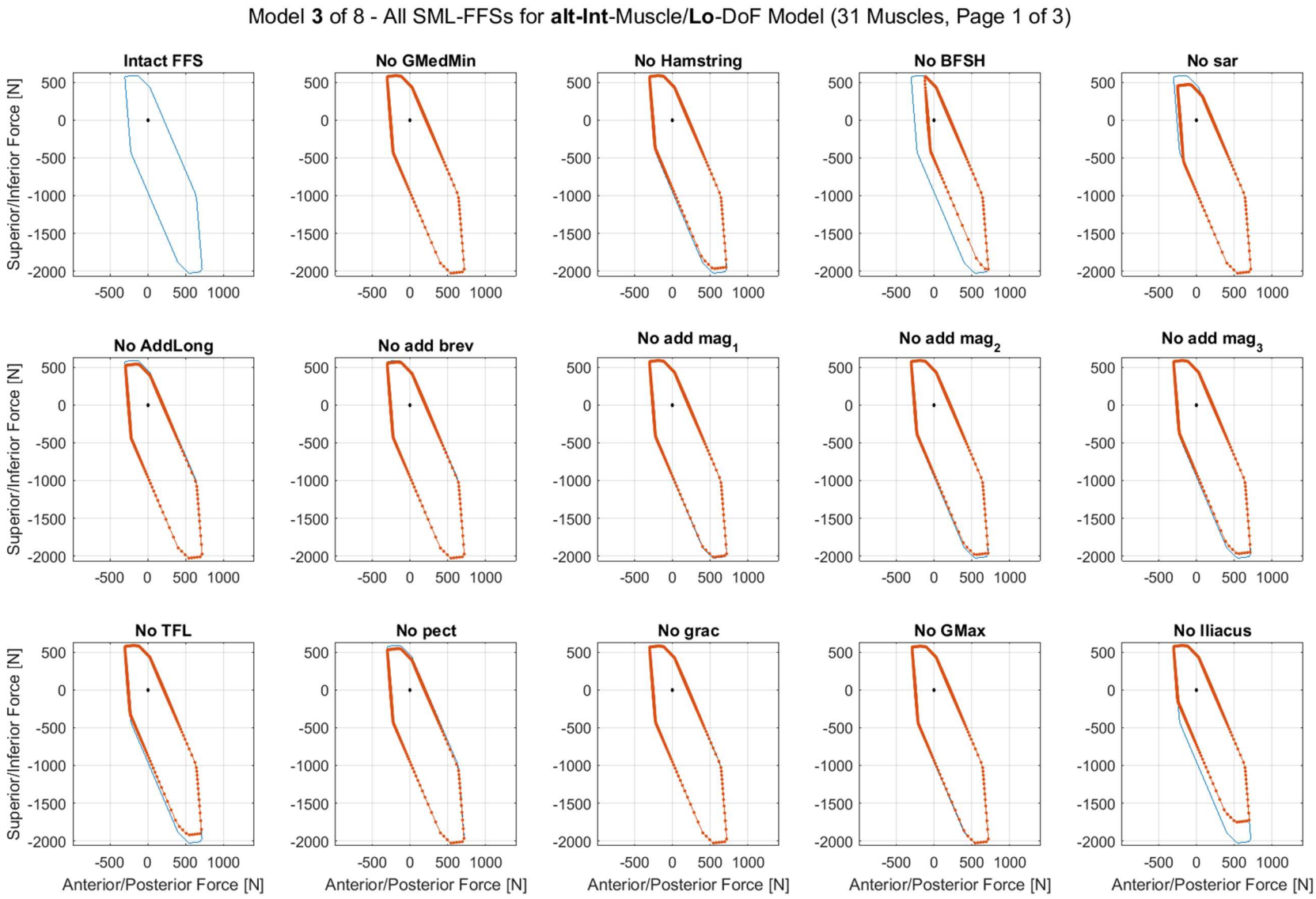
There are close relationships between FMARs and FFSs, particularly SML-FFSs that remain to be explored to determine which metrics best capture the inherent muscle redundancy (or lack thereof) of musculoskeletal systems. The sensitive area of a FFS to a single muscle is all the force directions and magnitudes where the FMAR does not include zero, i.e. the muscle is necessary. There is a direct comparison between FMAR and FFS, because FMARs also can demonstrate which force directions or magnitudes make specific muscles necessary. These are two different perspectives on the same force-activation space that remains to be thoroughly investigated.

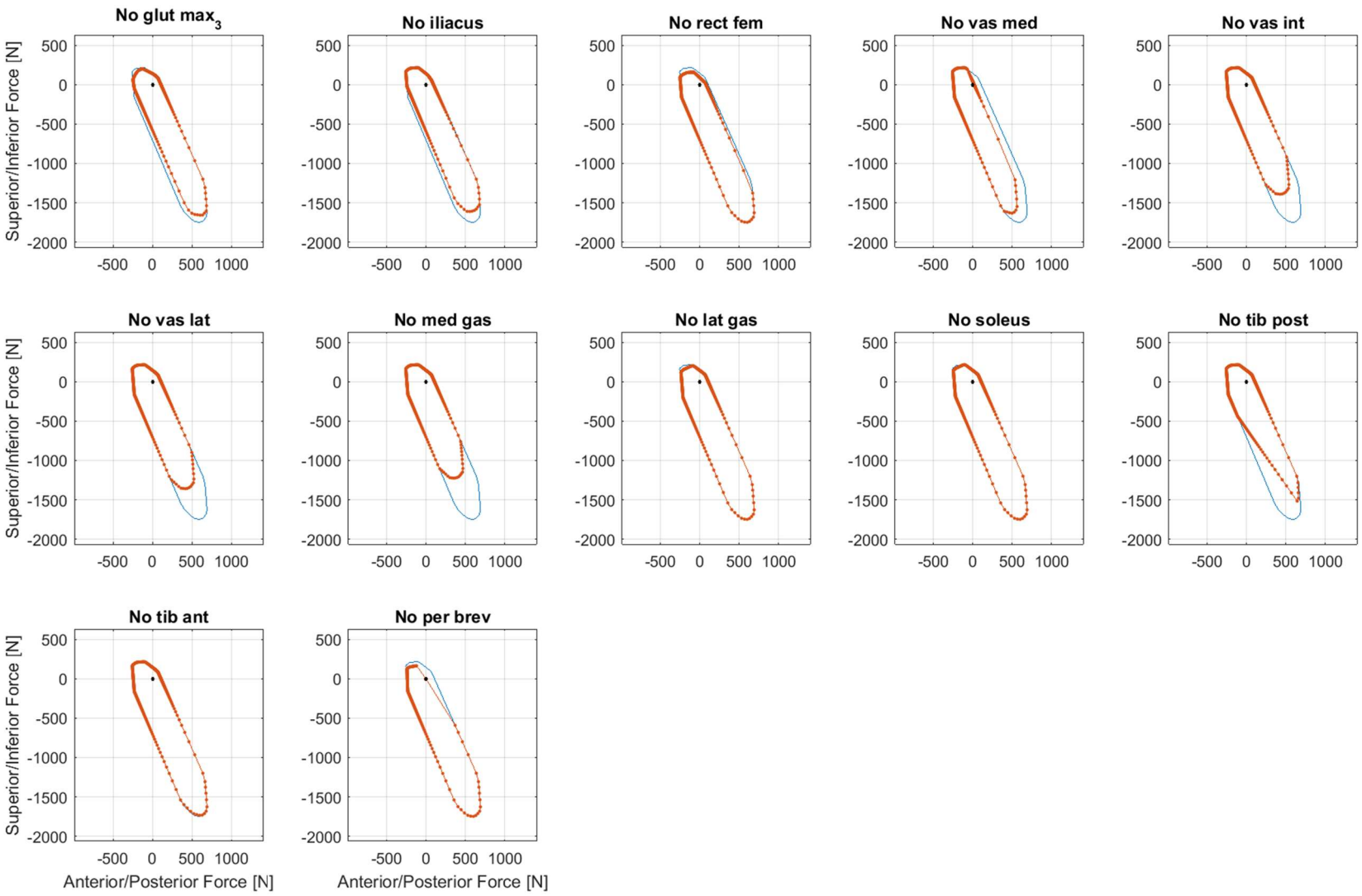
# APPENDIX A

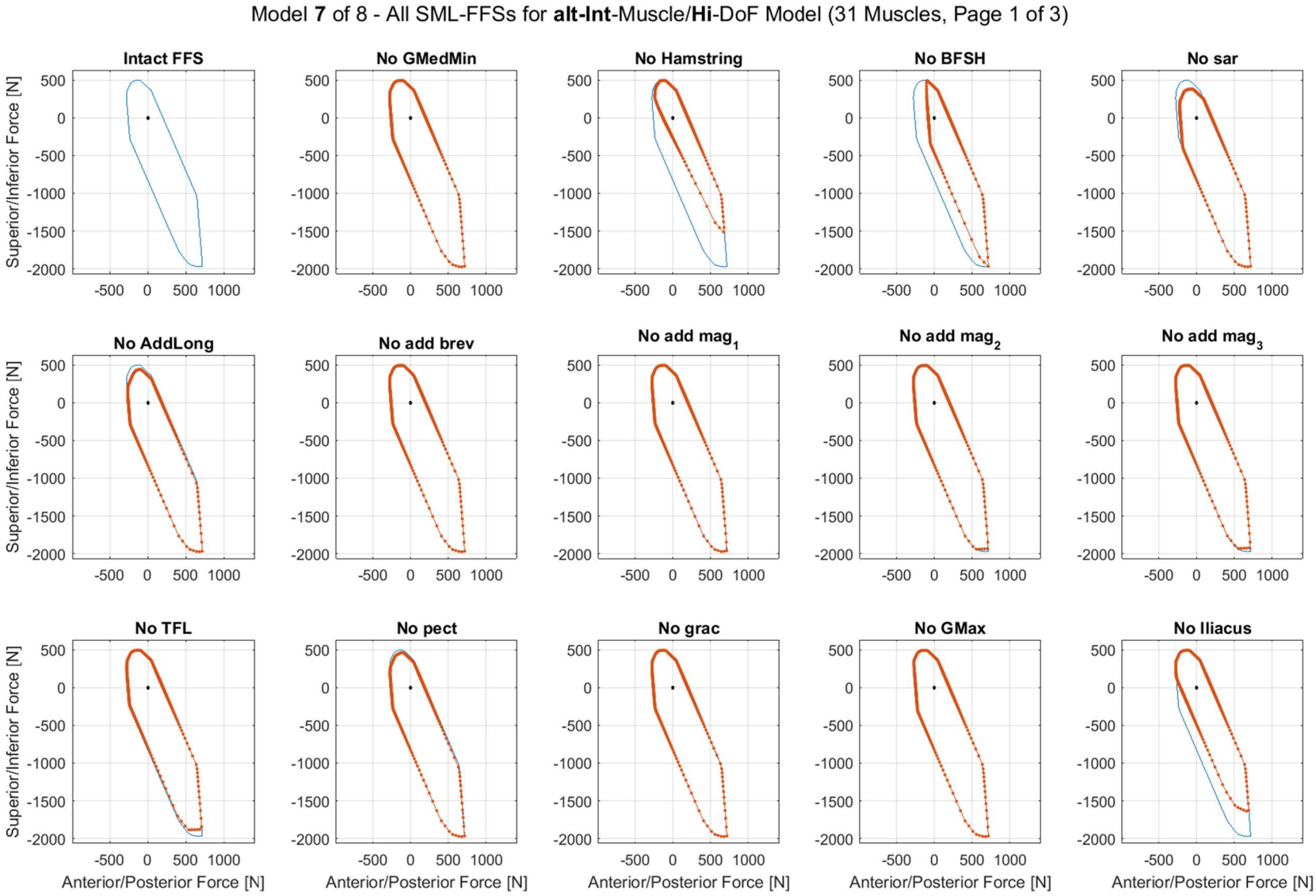
Single Muscle Loss Feasible Force Sets (SML-FFSs) across all muscles in all models



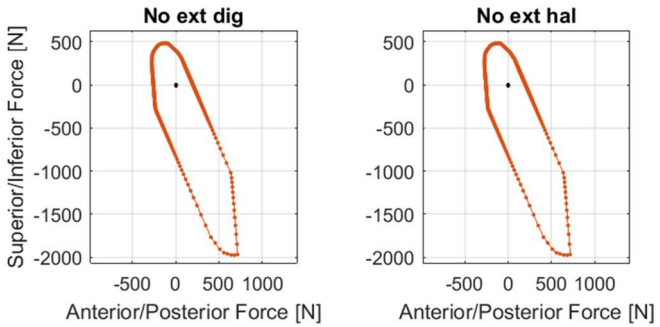








Model **7** of 8 - All SML-FFSs for **alt-Int**-Muscle/**Hi**-DoF Model (31 Muscles, Page 3 of 3)



# APPENDIX B

Maximum Force Feasible Muscle Activation Ranges (maxF-FMARs)
across all muscles in all models

# APPENDIX C

MATLAB Function for creating Feasible Force Sets

**Contents**



```
function [ACT,area,err,fval,maxFend,maxt] = ...
    funcCalcFFS_20160119(filename_saveAllData,whichLeg,filename_definitions,...
    indexMuscDysf,W,whichJoints,whichPlanes,n,filename_Muscles,filename_Jacobian)
```

```
% This script can create a FFS for an intact model of the human leg, and
% one with a simulated amputation. It is based on code written by Hongchul
% Sohn in September 2014, located in Ting Lab's archives in the folder.
%
% For this data, the right leg is in mid extension phase and the left leg
% is in mid flexion phase. Right/ext left/flex will be used interchangably.
%
%
% Necessary functions:
% - funcBuildLinearModel
% - funcCrossMatrix
% - funcDirConst
% - funcExtractMuscleParameters
%
% Necessary models:
%   Both an ".osim" and a ".nmcb" version of the same model. The ".osim"
%   must be located in the same folder. The code here will use that model
%   to extract the muscle parameters. The ".nmcb" file can be located
%   anywhere. It is simply used to calculate the Jacobian (i.e. no problems
%   with muscle path definitions in the model matter. Whew). I typically
%   include a ".txt" with the complete Jacobian (6x27) in the folder, and
%   then I select only the columns that align with the DoFs of the model.
%
% Other necessary files:
% - "Definitions_Muscles&Coords.mat" To clean up the file, I saved these
%   long strings into a ".mat" file that needs to be loaded to run
% - "Jacobian_raw.mat" - which has the 6x27 Jacobian inside it.
%
% EDIT: 19 Oct 2015. Set up cleaner to be run for individual FFSs.
% EDIT: 17 Nov 2015. Made the code more modular, and grouped up high the
%           variables I'll want as inputs when I change this to a function.
% EDIT: 12 Jan 2016. I'm changing how the collinearity constraint is
%                    implemented which will let me use LINPROG rather than
%                    fmincon to run the optimization. Woo!
% EDIT: 19 Jan 2016. I had been implementing the "legacy" variable
```

```
%                   whichMuscles incorrectly in the updated script, so I
%                   removed it, and am now using "indexMuscDysf" to
%                   implement the muscle dysfunction.
%                   The script also now outputs the most interesting
%                   variables that might want to be considered as a whole
%                   for all muscle dysfunction (and will aid in computing
%                   the robust area of the FFS).
% Danny Smith, 1 Oct 2015, Last Updated: 19 Jan 2016
tic
```

**Get Muscle Parameters: either from OpenSim model, or load in saved parameters**

```
% The variable 'filename_Muscles' will either be an OpenSim body file
% (.osim), which means this code will call OpenSim to calculate the muscle
% parameters for this model in this posture, or the variable will be a .mat
% file, which will have the muscle parameters saved from a previous data
% collection.
[~,~,MuscleExtension] = fileparts(filename_Muscles); % .osim or .mat?
if strcmp(MuscleExtension,'.osim')
    % Select the muscles and joints/coordinate names to draw from the model
    nMusc1 = size(W,2);
    load(filename_definitions,'muscles*','coordsR','coordsL')
    if strcmp(whichLeg,'right')
        muscles = musclesR;
        coords  = coordsR;
    elseif strcmp(whichLeg,'left')
        muscles = musclesL;
        coords  = coordsL;
    end
    clear musclesL_* musclesR_* coordsL coordsR % Only let the good variables stay

    % Run the function to find the muscle parameters
    [R,MTL,Fmax,L0,TSL,a0] = funcExtractMuscleParameters(filename_Muscles,coords,muscles);
    % Save the muscle parameters
    save([filename_saveAllData,'_MuscleData.mat'], 'R','MTL','Fmax','L0','TSL','a0');

elseif strcmp(MuscleExtension,'.mat')
    load(filename_Muscles)
else
    display('ERROR: Incorrect file type/file extension for muscle data.')
end
```

```
Not enough input arguments.

Error in funcCalcFFS_20160119 (line 56)
[~,~,MuscleExtension] = fileparts(filename_Muscles); % .osim or .mat?
```

**Build the Action matrix**

```
Action = funcBuildLinearModel(R,MTL,Fmax,L0,TSL,a0);

Action = Action*W;
```

### Load and process the Jacobian from Neuromechanic

This part isn't all run in NMC. I use the same model converted from an .osim file to a .nmcb file. There are some errors with how the muscles are converted, but the skeleton is fine which is all I need for the Jacobian.

```
load(filename_Jacobian,'Jacobian_raw')

J_R = Jacobian_raw(:,[7:9,12:15]); %#ok<NODEF>
J_L = Jacobian_raw(:,[7:9,12:15]+9);

if     strcmp(whichLeg,'right')
    J = J_R;
elseif strcmp(whichLeg,'left')
    J = J_L;
end
clear J_*
% Which vectors
if ~strcmp(whichJoints,'all')
    % Include only the muscles whose indices are specified by the vector whichMuscles
    nDoF = length(whichJoints);
    J = J(:,whichJoints);
    Action = Action(whichJoints,:);
else
    nDoF = size(Action,1);
end
```

### Create Unit Vectors for Force Directions

Whether we want a FFS in the sagittal plane, or in all 3D directions, we create a different set of unit vectors that define the directions in which we'll sample the FFS.

```
if strcmp(whichPlanes,'all')
    nDir  = (n+1)^2; % n is an input variable to the function
    [Z,Y,X] = sphere(n);
    Xmodel=reshape(X,1,nDir); Ymodel=reshape(Y,1,nDir);Zmodel=reshape(Z,1,nDir);
    Fdir = [Ymodel;Xmodel;Zmodel]; % I switched X &  Y to sample larger forces more
    % Rotate
    angle = 20;
    Fdir = [cosd(angle), -sind(angle), 0; sind(angle), cosd(angle), 0; 0, 0, 1]*Fdir;
    Fdir = [Fdir;zeros(3,(n+1)^2)];
elseif strcmp(whichPlanes,'sagittal')
    nDir = n; % n is an input variable to the function
    [X,Y,Z] = cylinder(1,n); % cylinder(radius, # points to define the circle)
    X = X(1,1:n);
    Y = Y(1,1:n);
    Z = Z(1,1:n);% Filters out just the points I want
    % Fdir = [X;Y;Z;zeros(3,n)]; % The XYZ coord.s for each of the directions of the force ve
ctors

    % Fdir is technically a wrench, and we need to specify it as such because
    % the Jacobian has all 6 spatial DoFs. Here, we specify it as zero, but it
    % whatever you choose, when solving in torque space, the endpoint torque
    % must be specified
    Fdir = [X;Y;Z;zeros(3,n)];
end
```

```
% plot(Fdir(1,:),Fdir(2,:),'.');
% axis equal
```

**Initialize variables & prepare for the optimization**

```
nMusc = size(Action,2);

% Do Upper/Lower Bounds AND Implement the muscle dysfunction
lb = zeros(nMusc,1);  % lower bounds
ub = ones(nMusc,1);   % upper bounds
if isempty(indexMuscDysf)==0 % When we want no Musc Dysf, this var is empty
    ub(indexMuscDysf) = 0; % Set the upper bound to zero for the "dysfunctioning" muscle
end

% options = optimoptions(
% options.MaxFunEvals = 5000;
options.Display = 'off';

% For fmincon, we need an initial guess. Now that we've returned to
% linprog, we don't need it anymore
% X0 = ones(nMusc,1);

% Initialize
ACT  = zeros(nMusc,nDir); % Vector of muscle activations in each direction
fval = zeros(nDir,1); % Final Value of Optimization
err  = zeros(nDir,1); % Error Flags

% % % % % % % % % % %% % % % % % % % % % % % % % % % % % % % % % % % parpool;
%
% [zeroJoint,zeroColumn] = ind2sub(size(J'*Fdir),find(J'*Fdir==0));
% whichZero =
% singular_index = min(abs(J'*Fdir)) == 0; % Finds all columns with a Zero in it
% non_singular_directions = 1:nDir;
% non_singular_directions(singular_index) = []; % Removes those indices from the FFS.
```

**Find the Muscles Activations corresponding to Maximum Force is each Direction (the Optimization Section)**

```
for dir = 1:nDir
    Fdes = Fdir(:,dir); % Select one force direction
    t_des = J'*Fdes; % Convert the force direction into torque space
    f = -t_des'*Action; %Since the optimization is f*(the activation vector),this is part of
the dot-product we're optimizing

    % Implementing the collinearity constraint
    vectorDiffMatrix = [ones(nDoF-1,1),-eye(nDoF-1)];
    % Switch the first two columns of the vector difference matrix if the
    % first element is zero.
    if t_des(1) == 0
        vectorDiffMatrix(:,[1 2]) = vectorDiffMatrix(:,[2 1]);
    end
    torqueInv = diag(1./t_des);
    Aeq = vectorDiffMatrix *torqueInv;
```

```
    [i,j] = ind2sub(size(Aeq),find(Aeq == -Inf | Aeq == Inf)); % Find everyone we divided by
zero
    if length(i)>1 % and then fix the places we divided by zero
        for k = 1:length(i)
        Aeq(   :,j(k)) = 0;
        Aeq(i(k),   :) = 0;
        Aeq(i(k),j(k)) = 1;
        end
    else
        Aeq(:,j) = 0;
        Aeq(i,:) = 0;
        Aeq(i,j) = 1;
    end
    Aeq = Aeq*Action;
    beq = zeros(nDoF-1,1); %end of collinearity constraint

    % HERE we run the actual optimization!!!
    [ACT(:,dir),fval(dir),err(dir)] = linprog(f,[],[],Aeq,beq,lb,ub,[],options);

end

% % % % % % % % % % % % % % % % % % % % % % % % % %% % % % % % % % delete(gcp);
% delete(gcp);

% initial condition can be zeros or ones
% %
```

**Concert the Muscle Activations for the Max Forces into the actual Torque and Force Values**

```
maxt = Action*ACT;
maxFend = zeros(3,nDir);
for dir = 1:nDir
maxFend(1:3,dir)=norm(maxt(:,dir))/norm(J'*Fdir(:,dir))*Fdir(1:3,dir);
end

whichModel = ['d',num2str(nDoF),'m',num2str(nMusc)]; %#ok<NASGU>  % This variable can identif
y these results later.

% Process the results a little, separating out the force directions and
% calculating the convex hull, including the area of the FFS
Fx = maxFend(1,:);
Fy = maxFend(2,:);
[index,area] = convhull(Fx,Fy); %#ok<ASGLU>
```

**Save all the data**

```
save(filename_saveAllData);

% %% Display
endtime = toc;
% disp(['This script ran for ',num2str(endtime),' seconds or ',num2str(endtime/60),' minutes.
']);

% display(['All Data saved in ',filename_saveAllData])
```

# APPENDIX D

MATLAB Function for creating Feasible Muscle Activation Ranges

## Contents



```
function func_CalcFMAR(filenameFFS)
```

```
% This function calculates and saves the FMARs for a precalculated FFS.
% Input: A string containing the name of the .mat where the FFS is saved.
% 2 March 2016, Danny Smith
```

## Load in FFS Data

```
load(filenameFFS,...
    'Action','nMusc','maxt','n')
% Action (nDoF x nMusc)
% nMusc (scalar)
% maxt (nDoF x n)
% n

nForceLevels = 3; % Number of force magnitudes at which to find the FMAR (including 0% and 10
0%)
a = linspace(0,1,nForceLevels);

% Initialize
Cmax  = zeros(nMusc,nForceLevels,nMusc,n);
Cmin  = zeros(nMusc,nForceLevels,nMusc,n);
c_max = zeros(nMusc,nForceLevels,n);
c_min = zeros(nMusc,nForceLevels,n);

% Inputs into linprog(f,[],[],Aeq,beq,LB,UB,[],options)
% beq in the for-loop
Aeq = Action;
f    = zeros(nMusc,1);
LB  = zeros(nMusc,1); % Lower Bound: All zeros
UB  =  ones(nMusc,1); % Upper Bound: All ones
options = struct('Display','none'); % Keeps it from displaying "done" every time through the
for-loop

for i = 1:n
```

```
    % F_end_levels = F_end*a;
    jointTorque_levels = maxt(:,i)*a;
```

## Find the range for each of the m muscles

```
    for m = 1:(nMusc)
```

### Optimize the Excitation Vectors for Maximum Forces in Specified Directions

```
        f(m) = 1; % Tell linprog which muscle to find the max/min for
        for j = 1:nForceLevels
            beq = jointTorque_levels(:,j);
            [Cmax(:,j,m,i)] = linprog(-f,[],[],Aeq,beq,LB,UB,[],options);
            [Cmin(:,j,m,i)] = linprog( f,[],[],Aeq,beq,LB,UB,[],options);
            c_max(m,j,i) = Cmax(m,j,m,i);
            c_min(m,j,i) = Cmin(m,j,m,i);
        end
        f(m) = 0; % Clear the selected muscle
```

```
    end
```

```
end
clear i m j   % Clear the for loop variables

saveFilename = regexprep(filenameFFS,'_FFS_','_FMAR_');
save(saveFilename)
% % % %% Plot FMARs
% % % figure('units','normalized','outerposition',[0 0 1 1]);
% % % subplotRows = 3;
% % % subplotCols = 5;
% % % subplotEntries = subplotRows*subplotCols;
% % %
% % % for m = 1:subplotEntries
% % %     subplot(subplotRows,subplotCols,m)
% % %     hold on
% % %     fill([a fliplr(a)],[c_min(m,:), fliplr(c_max(m,:))],[100 150 220]./256)
% % %     plot(a,c_max(m,:),'ks-')
% % %     plot(a,c_min(m,:),'kd-')
% % %     plot([1 1],[0 1],'k')
% % %     title(muscle_names{m});
% % %     axis([0 multiplier 0 1]);
% % %     m_last_fig1 = m;
% % % end
% % % suptitle('FMARs for 43 Muscles/Muscle Compartments of a Planar Leg Model in the directi
on of its Peak Feasible Extensor Force (1/3) - UNfeasible force magnitudes')
% % %
```

```
Not enough input arguments.

Error in func_CalcFMAR (line 7)
load(filenameFFS,...
```

*Published with MATLAB® R2015b*

# APPENDIX E

Matrix Values

```
Jacobian_raw =

  Columns 1 through 11

   1.0000         0         0    0.8758         0    0.0765    0.8097   -0.0052   -0.0047    0.6686   -0.7436
        0    1.0000         0    0.2750   -0.0765         0    0.3457    0.0047   -0.0052    0.7436    0.6686
        0         0    1.0000         0   -0.8758   -0.2750         0   -0.7985    0.3710         0         0
        0         0         0         0    1.0000         0         0    0.6686   -0.7436         0         0
        0         0         0         0         0    1.0000         0    0.7436    0.6686         0         0
        0         0         0    1.0000         0         0    1.0000         0         0         0         0

  Columns 12 through 22

   0.5425    0.1126   -0.0212         0         0         0         0         0         0         0         0
   0.0432    0.0745   -0.0080         0         0         0         0         0         0         0         0
        0    0.0262   -0.1097         0         0         0         0         0         0         0         0
        0   -0.1183    0.9814   -0.4015         0         0         0         0         0         0         0
        0   -0.1652    0.0190    0.2739         0         0         0         0         0         0         0
   1.0000    0.9791   -0.1910    0.8740         0         0         0         0         0         0         0

  Columns 23 through 27

        0         0         0         0         0
        0         0         0         0         0
        0         0         0         0         0
        0         0         0         0         0
        0         0         0         0         0
        0         0         0         0         0
```

```
R =

  Columns 1 through 11

   0.0012   -0.0121   -0.0206    0.0057    0.0022   -0.0040   -0.0556   -0.0694   -0.0628         0    0.0803
  -0.0219   -0.0303   -0.0352   -0.0174   -0.0248   -0.0295    0.0123    0.0231    0.0122         0   -0.0381
   0.0465    0.0248    0.0045    0.0430    0.0345    0.0217    0.0011    0.0006   -0.0059         0   -0.0092
        0         0         0         0         0         0   -0.0404   -0.0483   -0.0324   -0.0338   -0.0145
        0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0

  Columns 12 through 22

   0.0133   -0.0098   -0.0353   -0.0498   -0.0621    0.0692    0.0138   -0.0406   -0.0287   -0.0379   -0.0613
   0.0779    0.0651    0.0613    0.0569    0.0342   -0.0401    0.0444    0.0571   -0.0274   -0.0188    0.0252
   0.0033   -0.0019   -0.0122   -0.0036    0.0049    0.0297   -0.0023   -0.0032    0.0051   -0.0058   -0.0152
        0         0         0         0         0    0.0074         0   -0.0386         0         0         0
        0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0         0

  Columns 23 through 33

   0.0433    0.0428   -0.0287   -0.0065   -0.0103    0.0537         0         0         0         0         0
   0.0059    0.0112    0.0140   -0.0134   -0.0292   -0.0135         0         0         0         0         0
   0.0047    0.0036   -0.0441   -0.0293   -0.0048   -0.0013         0         0         0         0         0
        0         0         0         0         0    0.0571    0.0520    0.0536    0.0526   -0.0181   -0.0184
        0         0         0         0         0         0         0         0         0   -0.0448   -0.0461
        0         0         0         0         0         0         0         0         0   -0.0005    0.0026
        0         0         0         0         0         0         0         0         0         0         0

  Columns 34 through 43

        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
  -0.0470   -0.0141   -0.0143   -0.0195    0.0332   -0.0071   -0.0112    0.0231    0.0323    0.0333
   0.0018    0.0213    0.0213    0.0181    0.0151   -0.0250   -0.0253   -0.0257   -0.0111    0.0066
        0         0   -0.0065   -0.0058         0         0         0         0    0.0056    0.0066
```

```
a0 =

  Columns 1 through 11

    0.1396         0    0.3316    0.1745         0    0.0175    0.2618    0.0873         0    0.4014         0

  Columns 12 through 22

    0.1047         0    0.0873    0.0524    0.0873    0.0524         0    0.0524    0.0873         0    0.0873

  Columns 23 through 33

    0.1222    0.1396         0         0    0.1745    0.0873    0.0873    0.0524    0.0873    0.2967    0.1396

  Columns 34 through 43

    0.4363    0.2094    0.1222    0.1745    0.0873    0.0873    0.1745    0.2269    0.1396    0.1047

>> Fmax

Fmax =

  Columns 1 through 9

         819         573         653         270         285         323        1288         410         896

  Columns 10 through 18

         804         156         627         429         381         343         488         233         266

  Columns 19 through 27

         162         573         819         552        1073        1113         381         164         444

  Columns 28 through 36

        1169        1294        1365        1871        1558         683        3549        1588         310

  Columns 37 through 43

         322         905         435         943         180         512         162

>> L0

L0 =

  Columns 1 through 11

    0.0535    0.0845    0.0646    0.0680    0.0560    0.0380    0.0800    0.2010    0.1090    0.1730    0.5200

  Columns 12 through 22

    0.1380    0.1330    0.0870    0.1210    0.1310    0.0950    0.1000    0.3520    0.1420    0.1470    0.1440

  Columns 23 through 33

    0.1000    0.1000    0.0540    0.0240    0.0260    0.1140    0.0890    0.0870    0.0840    0.0600    0.0640

  Columns 34 through 43

    0.0500    0.0310    0.0340    0.0430    0.0980    0.0500    0.0490    0.0790    0.1020    0.1110
```

```
MTL =

  Columns 1 through 11

    0.1270    0.1464    0.1293    0.0809    0.0843    0.0926    0.4264    0.4783    0.4430    0.2186    0.5074

  Columns 12 through 22

    0.1928    0.1350    0.1325    0.2223    0.3880    0.5013    0.0869    0.4304    0.2320    0.2495    0.2937

  Columns 23 through 33

    0.1711    0.2258    0.0816    0.0667    0.1450    0.4123    0.2121    0.2226    0.2403    0.4038    0.3997

  Columns 34 through 43

    0.2637    0.3378    0.4259    0.4097    0.3260    0.2073    0.3899    0.1931    0.4626    0.4315

TSL =

  Columns 1 through 11

    0.0780    0.0530    0.0530    0.0160    0.0260    0.0510    0.3590    0.2555    0.3260    0.0890    0.1000

  Columns 12 through 22

    0.1100    0.0200    0.0600    0.1200    0.2490    0.4250    0.0330    0.1260    0.1250    0.1270    0.1450

  Columns 23 through 33

    0.1000    0.1600    0.0240    0.0390    0.1150    0.3100    0.1260    0.1360    0.1570    0.3900    0.3800

  Columns 34 through 43

    0.2500    0.3100    0.4000    0.3800    0.2230    0.1610    0.3450    0.1000    0.3450    0.3050
```

# APPENDIX F

Section of the Neuromechanic Body File used to Calculate the Jacobian

```
<NeuromechanicFile>
  <Bodies>
    <RigidBody Name="pelvis">
      <FrameLocation Parent="ground">0. 0. 0.</FrameLocation>
      <Mass>1.1777E1</Mass>
      <Inertia>1.028E-1 8.71E-2 5.79E-2 0. 0. 0.</Inertia>
      <CenterOfMass>-7.07E-2 0. 0.</CenterOfMass>
      <OSIMjointLOC>0 0 0</OSIMjointLOC>
      <DegreeOfFreedom Name="pelvis_tilt" Type="rotation">
        <Locked>false</Locked>
        <Axis>0. 0. 1.</Axis>
        <State Type="Kinematic">0. 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
      <DegreeOfFreedom Name="pelvis_list" Type="rotation">
        <Locked>false</Locked>
        <Axis>1. 0. 0.</Axis>
        <State Type="Kinematic">0. 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
      <DegreeOfFreedom Name="pelvis_rotation" Type="rotation">
        <Locked>false</Locked>
        <Axis>0. 1. 0.</Axis>
        <State Type="Kinematic">0. 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
      <DegreeOfFreedom Name="pelvis_tx" Type="translation">
        <Locked>false</Locked>
        <Axis>1. 0. 0.</Axis>
        <State Type="Kinematic">7.096141956912E-22 0.</State>
        <Range>-2.E1 2.E1</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
      <DegreeOfFreedom Name="pelvis_ty" Type="translation">
        <Locked>false</Locked>
        <Axis>0. 1. 0.</Axis>
        <State Type="Kinematic">0. 0.</State>
        <Range>-2.E1 2.E1</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
      <DegreeOfFreedom Name="pelvis_tz" Type="translation">
        <Locked>false</Locked>
        <Axis>0. 0. 1.</Axis>
        <State Type="Kinematic">7.096141956911E-22 0.</State>
        <Range>-2.E1 2.E1</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
    </RigidBody>
    <RigidBody Name="femur_r">
      <FrameLocation Parent="pelvis">-7.07E-2 -6.61E-2 8.35E-2</FrameLocation>
```

```
      <Mass>9.3014</Mass>
      <Inertia>1.339E-1 3.51E-2 1.412E-1 0. 0. 0.</Inertia>
      <CenterOfMass>0. -1.7E-1 0.</CenterOfMass>
      <OSIMjointLOC>0 0 0</OSIMjointLOC>
      <DegreeOfFreedom Name="hip_flexion_r" Type="rotation">
        <Locked>false</Locked>
        <Axis>0. 0. 1.</Axis>
        <State Type="Kinematic">0.83849 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
      <DegreeOfFreedom Name="hip_adduction_r" Type="rotation">
        <Locked>false</Locked>
        <Axis>1. 0. 0.</Axis>
        <State Type="Kinematic">0. 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
      <DegreeOfFreedom Name="hip_rotation_r" Type="rotation">
        <Locked>false</Locked>
        <Axis>0. 1. 0.</Axis>
        <State Type="Kinematic">0. 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
    </RigidBody>
    <RigidBody Name="tibia_r">
      <FrameLocation Parent="femur_r">0. 0. 0.</FrameLocation>
      <Mass>3.7075</Mass>
      <Inertia>5.04E-2 5.1E-3 5.11E-2 0. 0. 0.</Inertia>
      <CenterOfMass>0. -1.867E-1 0.</CenterOfMass>
      <OSIMjointLOC>0 0 0</OSIMjointLOC>
      <DegreeOfFreedom Name="knee_angle_r" Type="rotation">
        <Locked>false</Locked>
        <Axis>0. 0. 1.</Axis>
        <State Type="Kinematic">-0.91717 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
      <DegreeOfFreedom Name="dofC_knee_angle_r0" Type="translation">
        <Locked>false</Locked>
        <Axis>1. 0. 0.</Axis>
        <State Type="Kinematic">-4.508592094335E-3 0.</State>
        <Range>-2.E1 2.E1</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
      <DegreeOfFreedom Name="dofC_knee_angle_r1" Type="translation">
        <Locked>false</Locked>
        <Axis>0. 1. 0.</Axis>
        <State Type="Kinematic">-3.958166714949E-1 0.</State>
        <Range>-2.E1 2.E1</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
```

```
    </RigidBody>
    <RigidBody Name="talus_r">
      <FrameLocation Parent="tibia_r">0. -4.3E-1 0.</FrameLocation>
      <Mass>1.E-1</Mass>
      <Inertia>1.E-3 1.E-3 1.E-3 0. 0. 0.</Inertia>
      <CenterOfMass>0. 0. 0.</CenterOfMass>
      <OSIMjointLOC>0 0 0</OSIMjointLOC>
      <DegreeOfFreedom Name="ankle_angle_r" Type="rotation">
        <Locked>false</Locked>
        <Axis>-1.050135495115E-1 -1.740224491904E-1 9.791263154449E-1</Axis>
        <State Type="Kinematic">-0.58346 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
    </RigidBody>
    <RigidBody Name="calcn_r">
      <FrameLocation Parent="talus_r">-4.877E-2 -4.195E-2 7.92E-3</FrameLocation>
      <Mass>1.25</Mass>
      <Inertia>1.4E-3 3.9E-3 4.1E-3 0. 0. 0.</Inertia>
      <CenterOfMass>1.E-1 3.E-2 0.</CenterOfMass>
      <OSIMjointLOC>0 0 0</OSIMjointLOC>
      <DegreeOfFreedom Name="subtalar_angle_r" Type="rotation">
        <Locked>false</Locked>
        <Axis>7.871796068877E-1 6.04747457609E-1 -1.209494895218E-1</Axis>
        <State Type="Kinematic">0. 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
    </RigidBody>
    <RigidBody Name="toes_r">
      <FrameLocation Parent="calcn_r">1.788E-1 -2.E-3 1.08E-3</FrameLocation>
      <Mass>2.166E-1</Mass>
      <Inertia>1.E-4 1.99E-4 1.E-4 0. 0. 0.</Inertia>
      <CenterOfMass>3.46E-2 6.E-3 -1.75E-2</CenterOfMass>
      <OSIMjointLOC>0 0 0</OSIMjointLOC>
      <DegreeOfFreedom Name="mtp_angle_r" Type="rotation">
        <Locked>false</Locked>
        <Axis>-5.809543982451E-1 0. 8.139361075414E-1</Axis>
        <State Type="Kinematic">0. 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
    </RigidBody>
    <RigidBody Name="femur_l">
      <FrameLocation Parent="pelvis">-7.07E-2 -6.61E-2 -8.35E-2</FrameLocation>
      <Mass>9.3014</Mass>
      <Inertia>1.339E-1 3.51E-2 1.412E-1 0. 0. 0.</Inertia>
      <CenterOfMass>0. -1.7E-1 0.</CenterOfMass>
      <OSIMjointLOC>0 0 0</OSIMjointLOC>
      <DegreeOfFreedom Name="hip_flexion_l" Type="rotation">
        <Locked>false</Locked>
        <Axis>0. 0. 1.</Axis>
        <State Type="Kinematic">0.875197 0.</State>
```

```
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
      <DegreeOfFreedom Name="hip_adduction_l" Type="rotation">
        <Locked>false</Locked>
        <Axis>-1. 0. 0.</Axis>
        <State Type="Kinematic">0. 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
      <DegreeOfFreedom Name="hip_rotation_l" Type="rotation">
        <Locked>false</Locked>
        <Axis>0. -1. 0.</Axis>
        <State Type="Kinematic">0. 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
    </RigidBody>
    <RigidBody Name="tibia_l">
      <FrameLocation Parent="femur_l">0. 0. 0.</FrameLocation>
      <Mass>3.7075</Mass>
      <Inertia>5.04E-2 5.1E-3 5.11E-2 0. 0. 0.</Inertia>
      <CenterOfMass>0. -1.867E-1 0.</CenterOfMass>
      <OSIMjointLOC>0 0 0</OSIMjointLOC>
      <DegreeOfFreedom Name="knee_angle_l" Type="rotation">
        <Locked>false</Locked>
        <Axis>0. 0. 1.</Axis>
        <State Type="Kinematic">-1.641733 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
      <DegreeOfFreedom Name="dofC_knee_angle_l2" Type="translation">
        <Locked>false</Locked>
        <Axis>1. 0. 0.</Axis>
        <State Type="Kinematic">-4.508592094335E-3 0.</State>
        <Range>-2.E1 2.E1</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
      <DegreeOfFreedom Name="dofC_knee_angle_l3" Type="translation">
        <Locked>false</Locked>
        <Axis>0. 1. 0.</Axis>
        <State Type="Kinematic">-3.958166714949E-1 0.</State>
        <Range>-2.E1 2.E1</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
    </RigidBody>
    <RigidBody Name="talus_l">
      <FrameLocation Parent="tibia_l">0. -4.3E-1 0.</FrameLocation>
      <Mass>1.E-1</Mass>
      <Inertia>1.E-3 1.E-3 1.E-3 0. 0. 0.</Inertia>
      <CenterOfMass>0. 0. 0.</CenterOfMass>
      <OSIMjointLOC>0 0 0</OSIMjointLOC>
      <DegreeOfFreedom Name="ankle_angle_l" Type="rotation">
```

```
        <Locked>false</Locked>
        <Axis>1.050135495115E-1 1.740224491904E-1 9.791263154449E-1</Axis>
        <State Type="Kinematic">0.417631 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
    </RigidBody>
    <RigidBody Name="calcn_l">
      <FrameLocation Parent="talus_l">-4.877E-2 -4.195E-2 -7.92E-3</FrameLocation>
      <Mass>1.25</Mass>
      <Inertia>1.4E-3 3.9E-3 4.1E-3 0. 0. 0.</Inertia>
      <CenterOfMass>1.E-1 3.E-2 0.</CenterOfMass>
      <OSIMjointLOC>0 0 0</OSIMjointLOC>
      <DegreeOfFreedom Name="subtalar_angle_l" Type="rotation">
        <Locked>false</Locked>
        <Axis>-7.871796068877E-1 -6.04747457609E-1 -1.209494895218E-1</Axis>
        <State Type="Kinematic">0. 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
    </RigidBody>
    <RigidBody Name="toes_l">
      <FrameLocation Parent="calcn_l">1.788E-1 -2.E-3 -1.08E-3</FrameLocation>
      <Mass>2.166E-1</Mass>
      <Inertia>1.E-4 1.99E-4 1.E-4 0. 0. 0.</Inertia>
      <CenterOfMass>3.46E-2 6.E-3 1.75E-2</CenterOfMass>
      <OSIMjointLOC>0 0 0</OSIMjointLOC>
      <DegreeOfFreedom Name="mtp_angle_l" Type="rotation">
        <Locked>false</Locked>
        <Axis>5.809543982451E-1 0. 8.139361075414E-1</Axis>
        <State Type="Kinematic">0. 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
    </RigidBody>
    <RigidBody Name="torso">
      <FrameLocation Parent="pelvis">-1.007E-1 8.15E-2 0.</FrameLocation>
      <Mass>3.42366E1</Mass>
      <Inertia>1.4745 7.555E-1 1.4314 0. 0. 0.</Inertia>
      <CenterOfMass>-3.E-2 3.2E-1 0.</CenterOfMass>
      <OSIMjointLOC>0 0 0</OSIMjointLOC>
      <DegreeOfFreedom Name="lumbar_extension" Type="rotation">
        <Locked>false</Locked>
        <Axis>0. 0. 1.</Axis>
        <State Type="Kinematic">0. 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
        <MotionCost>1.</MotionCost>
      </DegreeOfFreedom>
      <DegreeOfFreedom Name="lumbar_bending" Type="rotation">
        <Locked>false</Locked>
        <Axis>1. 0. 0.</Axis>
        <State Type="Kinematic">0. 0.</State>
        <Range>-6.28318530718 6.28318530718</Range>
```

# REFERENCES


Bernstein, N. (1967). *The Coordination and Regulation of Movements*. New York: Pergamon Press.

Buchanan, T. S., & Shreeve, D. A. (1996). An Evaluation of Optimization Techniques for the Prediction of Muscle Activation Patterns During Isometric Tasks. *Journal of Biomechanical Engineering, 118*(4), 565-574.

Bunderson, N. E., Bingham, J. T., Hongchul Sohn, M., Ting, L. H., & Burkholder, T. J. (2012). Neuromechanic: A computational platform for simulation and analysis of the neural control of movement. *Int j numer method biomed eng, 28*(10), 1015-1027. doi: 10.1002/cnm.2486

Bunderson, N. E., Burkholder, T. J., & Ting, L. H. (2008). Reduction of neuromuscular redundancy for postural force generation using an intrinsic stability criterion. *J Biomech*.

Crowninshield, R. D., & Brand, R. A. (1981). A physiologically based criterion of muscle force prediction in locomotion. *Journal of Biomechanics, 14*(11), 793-801.

Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., . . . Thelen, D. G. (2007). OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement. *Ieee Transactions on Biomedical Engineering, 54*(11), 1940-1950. doi: 10.1109/TBME.2007.901024

Fenn, W. O., & Marsh, B. S. (1935). Muscular force at different speeds of shortening. *The Journal of Physiology, 85*(3), 277-297.

Gordon, A. M., Huxley, A. F., & Julian, F. J. (1966). The variation in isometric tension with sarcomere length in vertebrate muscle fibres. *J Physiol, 184*(1), 170-192.

Gruben, K. G., Lopez-Ortiz, C., & Schmidt, M. W. (2003). The control of foot force during pushing efforts against a moving pedal. *Exp Brain Res, 148*(1), 50-61.

Herzog, W., & Leonard, T. R. (1991). Validation of optimization models that estimate the forces exerted by synergistic muscles. *Journal of Biomechanics, 24*(Supplement 1), 31-39.

Hill, A. V. (1938). The Heat of Shortening and the Dynamic Constants of Muscle. *Proccedings of the Royal Society of London, 126, B*(843), 136-195.

Kuo, A. D., & Zajac, F. E. (1993). A biomechanical analysis of muscle strength as a limiting factor in standing posture. *J Biomech, 26 Suppl 1*, 137-150.

Kutch, J. J., & Valero-Cuevas, F. J. (2011). Muscle redundancy does not imply robustness to muscle dysfunction. *J Biomech, 44*(7), 1264-1270. doi: 10.1016/j.jbiomech.2011.02.014

Loeb, G. E. (2000). Overcomplete musculature or underspecified tasks? *Motor Control, 4*(1), 81-83; discussion 97-116.

Martelli, S., Calvetti, D., Somersalo, E., & Viceconti, M. (2015). Stochastic modelling of muscle recruitment during activity. *Interface Focus, 5*(2), 20140094. doi: 10.1098/rsfs.2014.0094

Martelli, S., Calvetti, D., Somersalo, E., Viceconti, M., & Taddei, F. (2013). Computational tools for calculating alternative muscle force patterns during motion: a comparison of possible solutions. *J Biomech, 46*(12), 2097-2100. doi: 10.1016/j.jbiomech.2013.05.023

Massey, W. S. (1983). Cross Products of Vectors in Higher Dimensional Euclidean Spaces. *American Mathematical Monthly, 90*(10), 697-701. doi: 10.2307/2323537

McKay, J. L., Burkholder, T. J., & Ting, L. H. (2007). Biomechanical capabilities influence postural control strategies in the cat hindlimb. *J Biomech, 40*(10), 2254-2260.

McKay, J. L., & Ting, L. H. (2008). Functional muscle synergies constrain force production during postural tasks. *J Biomech, 41*(2), 299-306. doi: S0021-9290(07)00390-9 [pii] 10.1016/j.jbiomech.2007.09.012

McKay, J. L., & Ting, L. H. (2012). Optimization of Muscle Activity for Task-Level Goals Predicts Complex Changes in Limb Forces across Biomechanical Contexts. *PLoS Comput Biol, 8*(4), e1002465. doi: 10.1371/journal.pcbi.1002465

Simpson, C. S., Sohn, M. H., Allen, J. L., & Ting, L. H. (2015). Feasible muscle activation ranges based on inverse dynamics analyses of human walking. *J Biomech*. doi: 10.1016/j.jbiomech.2015.07.037

Sohn, M. H., McKay, J. L., & Ting, L. H. (2013). Defining feasible bounds on muscle activation in a redundant biomechanical task: practical implications of redundancy. *J Biomech, 46*(7), 1363-1368. doi: 10.1016/j.jbiomech.2013.01.020

Sohn, M. H., & Ting, L. H. (2016). Suboptimal Muscle Synergy Activation Patterns Generalize their Motor Function across Postures. *Front Comput Neurosci, 10*, 7. doi: 10.3389/fncom.2016.00007

Thelen, D. G., & Anderson, F. C. (2006). Using computed muscle control to generate forward dynamic simulations of human walking from experimental data. *Journal of Biomechanics, 39*(6), 1107-1115.

Valero-Cuevas, F. J. (2000). Predictive modulation of muscle coordination pattern magnitude scales fingertip force magnitude over the voluntary range. *J Neurophysiol, 83*(3), 1469-1479.

Valero-Cuevas, F. J., Zajac, F. E., & Burgar, C. G. (1998). Large index-fingertip forces are produced by subject-independent patterns of muscle excitation. *J Biomech, 31*(8), 693-703.

van der Krogt, M. M., Delp, S. L., & Schwartz, M. H. (2012). How robust is human gait to muscle weakness? *Gait Posture, 36*(1), 113-119. doi: 10.1016/j.gaitpost.2012.01.017

Zajac, F. E. (1989). Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Crit Rev Biomed Eng, 17*(4), 359-411.