Dynamic Neuromechanical Sets for Locomotion

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Aravind Sundararajan

December 2020

Copyright © by Aravind Sundararajan, 2020

All Rights Reserved.

Acknowledgments

Learning new paradigms is hard! This dissertation would be impossible without the guidance of my advisor, Dr. Jeffrey A. Reinbolt. Lots and lots of two hours research discussions and one-on-one lessons weaponized this chemistry-focused undergrad into an unparalleled comprehension of dynamics and musculoskeletal modeling. This research was built upon the works of Dr. Misagh B. Mansouri and Dr. Nicolas A. Vivaldi who investigated the application of operational space control and task prioritization to musculoskeletal modeling and together performed some of the pioneering works in MATLAB-OpenSim interfacing beyond the scope of batch analysis. Thank you to the members of my dissertation committee. Dr. Caleb Rucker offered a unique perspective with expertise in continuum robotics, which is only a few inches away from the analysis of biological systems. Dr. Eric Espinoza-Wade was supportive in helping me understand experimental design and statistical analysis, and hammering out the clinical relevance of the developed tools. Dr. Cyril Jon Donnelly was indispensable in helping me probe the motivation behind the design of feasible sets analysis as extensions to conventional biomechanics and the relationship of feasible sets analysis to neural control itself. Thank you to Ashley Rice who has acted as my second brain for several years. Thank you to my close friend John Murray who was invaluable in my growth as a programmer and as a person. Finally, thank you to my family. They were supportive too.

Abstract

Most biological systems employ multiple redundant actuators, which is a complicated problem of controls and analysis. Unless assumptions about how the brain and body work together, and assumptions about how the body prioritizes tasks are applied, it is not possible to find the actuator controls. The purpose of this research is to develop and apply computational tools to the analysis of arbitrary musculoskeletal models that employ redundant actuators. Instead of relying primarily on optimization frameworks and numerical methods or task prioritization schemes used typically in biomechanics to find a singular solution for actuator controls, tools for feasible sets analysis are instead developed to explore the boundaries of possible actuator controls. Previously in the literature, feasible sets analysis has been used to analyze models assuming static poses. Here, tools that explore the feasible sets of actuator controls over the course of a dynamic task are developed and applied to various models of humanoid movement. The cost-function agnostic methods of analysis developed in this work run parallel and in concert with other methods of analysis such as principal component analysis, muscle synergies theory and task prioritization. Researchers and healthcare professionals may potentially gain greater insights into decision-making during behavioral tasks by layering these other tools on top of feasible sets analysis.

Table of Contents

1 Introduction **1**

1.1 Motivation 2

1.2 Objectives and Approach 2

1.3 Chapter Summary 3

2 Literature Review **5**

2.1 Principles of Biomechanics 5

2.1.1 Physics: Statics, Dynamics, Multibody Systems 5

2.1.2 Principles of Robotics and Bio-Inspired Systems 6

2.1.3 Experimental Data to Subject Specific Model 7

2.2 Optimization and Sampling Frameworks 13

2.2.1 Linear Programming (LP) 15

2.2.2 Quadratic (QP), Conic (CP) and Nonlinear Programming (NLP) . . 16

2.2.3 Vertex Enumeration 17

2.2.4 Markov Chain Monte Carlo Methods 18

2.2.5 SLAM: Simultaneous Localization and Mapping 18

2.3 Neural Control 20

2.4 Feasible Sets Analysis 30

2.4.1 Feasible Activation Space (FAS) 31

2.5 Biological Need 32

2.5.1 Gait and Balance Disorders 33

3 A Naive Approach: Pseudo-Static Analysis of End-Effector Forces **34**
3.1 Chapter Background . . . 34
3.2 Study: Pseudostatic Analysis . . . 46
3.2.1 Case Study Motivation . . . 46
3.2.2 Methods . . . 48
3.2.3 Statistical Analysis . . . 54
3.2.4 Results . . . 57
3.2.5 Discussion . . . 58
3.3 Chapter Summary . . . 59

4 Dynamical Considerations and Projection Operators **60**
4.1 Chapter Background . . . 60
4.1.1 Linear Operators . . . 60
4.1.2 Manipulator Robotics Application . . . 61
4.2 Study: Feasible Accelerations . . . 65
4.2.1 Case Study Motivation . . . 65
4.2.2 Methods . . . 65
4.2.3 Results . . . 71
4.2.4 Discussion . . . 75
4.3 Tools . . . 75
4.4 Chapter Summary . . . 77

5 Computing Feasible Controls **78**
5.1 Chapter Background . . . 78
5.1.1 Geometry . . . 78
5.1.2 Muscle Redundancy . . . 83
5.2 Study: Computing Feasible Controls . . . 85
5.2.1 Case Study Motivation . . . 85
5.2.2 Methods . . . 86
5.2.3 Results . . . 88
5.2.4 Discussion . . . 91

5.3 Tools 91
5.4 Chapter Summary 92

6 Constraining Feasible Controls by Experimental Joint Forces 93
6.1 Chapter Background 93
6.1.1 Procedurally Constructing Constraints 93
6.1.2 Muscle Lines of Action 95
6.2 Study: Constraining Controls by Joint Loads 97
6.2.1 Case Study Motivation 97
6.2.2 Methods 97
6.2.3 Results 98
6.2.4 Discussion 104
6.3 Tools 104
6.4 Chapter Summary 105

7 Windowing for Controls: Feasible Activation Space Trajectories 106
7.1 Chapter Background 106
7.1.1 Challenges 106
7.1.2 First Order Muscle Activation Dynamics 107
7.1.3 Application of the Bounds on Muscle Activation to CG 111
7.1.4 Minimal Activation 111
7.1.5 ”Centers” of FAS 111
7.1.6 Hit-and-Run (HAR) 114
7.1.7 Dikin Walk (DW) 117
7.1.8 Modified Dikin with Multivariate Skew Normal 119
7.2 Study: Feasible Activation Space Trajectories (FAST) 123
7.2.1 Case Study Motivation 123
7.2.2 Methods 123
7.2.3 Results 124
7.2.4 Discussion 130
7.3 Tools 131

7.4 Chapter Summary 132

8 Concluding Remarks **133**

8.1 Future Work 134

8.1.1 The Functional Approach and Variational Calculus 134

8.1.2 Trajectory 136

Bibliography **138**

Appendices **152**

A Algorithms 153

Vita **156**

List of Figures

2.1 Modeling begins with data acquisition, and *in silico* models can only ever be as good as the acquired data allows. 8

2.2 Inputs and outputs of the static optimization procedure featured in OpenSim from the simTK confluence site [88]. 11

2.3 Controls system featured in OpenSim CMC from the simTK confluence site [88]. 12

2.4 the simple quadratic cone $z^2 \geq \sum a_i^2$. 16

2.5 As a roomba cleans a room, it makes a map of the space by sampling points in $\mathbb{O}$ [ben]. 19

2.6 The electrical signal of the action potential of the motor neuron is transformed into a chemical signal at the neuromuscular junction by way of receptors like nicotinic acetylcholine receptor (nAChRs) and Ca^{++}-gated channels. 21

2.7 A muscle sarcomere produces force by binding crossbridges between actin and myosin filaments. Titin forms an elastic element within the muscle fibers themselves. Many sarcomeres together in parallel and series form a muscle. . 23

2.8 Muscles attach to bones via tendons. Tendons are modeled as elastic elements in Hill type models of the MTU. The total length of the MTU is purely a function of the geometry of the relevant bodies, but the individual muscle fiber length and tendon slack lengths are functions of the muscle state along with orientation of the bodies. 24

2.9 From De Sapio 2008 [28]. A muscle's active force is a function of the velocity and length of the muscle fiber, and the muscle activation a. The passive component of the MTU's force is just a function of the fiber length. 26

2.10 Spatiotemporal tunneling through sets of feasible controls over time. 29

3.1 Applied joint moment (blue) can be mapped to the end effector force (red) using the kinematic Jacobian, J. In the static domain, this output force is literally the resultant force due thanks to the propagation of moment along the end effector. In a dynamic domain, this vector represents an operational space command $\vec{F}$ (red) that reflects on the acceleration of end effector in operational space. 37

3.2 A simple 1 DOF model with 2 muscle-tendon actuators, highlighting the muscle redundancy problem. 38

3.3 The projection of a convex set onto some of its dimensions is also convex. This is the simple parallel projection, but there are many kinds of projection operators. Skew, scale, translate, rotate, orthographic, perspective. These geometric concepts form the basis of computer graphics. 40

3.4 A simple 2 DOF model with 4 muscle-tendon actuators for exploring the feasible map problem. 41

3.5 A convex set in $\mathbb{R}^2$ (left) and nonconvex set in $\mathbb{R}^2$ (right). 44

3.6 Minkowski Sum of a set of 3 generator vectors (green). The boundaries of $\bigoplus \boldsymbol{F}$ can be found by sorting the vectors about their polar angle from the origin, positive y-axis, and summing them in order. View the appendix for giftwrapping and minkowski sum algorithms. 45

3.7 Depicting the muscle moment arm $\boldsymbol{R}$ and the relationship to the output force at the end effector $\boldsymbol{F}$. 49

3.8 23 DOF 92 muscle model available in OpenSim. Scaled to reflect subject anthropomentry and using subject-specific kinematic states. $\mathbb{F}$ are highlighted for left and right legs. 50

3.9 Comparing braking and propulsion feasible forces. This convention may not be intuitive, but these sets are the spaces of $\mathbb{O}$ commands in the static domain. Action forces or commands that brake are necessarily anterior to the center of pressure or COM of the calcaneus while propulsion is necessarily posterior. 52

3.10 differences in $\mathbb{F}$ at a single frame during swing phase, highlighting the influence of $\boldsymbol{l_m}$ and $\boldsymbol{l_m} + \boldsymbol{v_m}$ consideration. 53
3.11 Total, propulsive and braking volume as the sum of p$\mathbb{F}$ and b$\mathbb{F}$ over progression of gait. Different muscle physiological considerations and speeds are plotted with different colors. Simple main effects analysis revealed significant differences across considerations in mid stance and in mid swing. 55

4.1 $\mathbb{O}$ control for robots or musculoskeletal systems is all about dictating the end effector trajectories $\boldsymbol{\mathcal{X}}$ in $\mathbb{O}$ through command vectors or tasks and manipulating $\boldsymbol{\mathcal{Q}}$ in the null-space of $\mathbb{C}$. Also, for floating base robots and biological systems, it involves how the actuated DOF can be used to induce changes to the virtual DOF. 61
4.2 Highlighting the 6 muscles of the 2 DOF 6 muscle model (Figures 4.3, 4.6). . 66
4.3 Progression of task for the 2 DOF 6 muscles model (Figure 4.6). 67
4.4 Diagram of the muscles of the Gait1018 model used to find feasible activation space ($\boldsymbol{\mathcal{V}}$-**FAS**) bounds (Figure 4.8). Biarticular muscles are between primary colors. 68
4.5 Progression of gait for the Gait1018 and Gait2354 models featured in Figures 4.8 and 5.6. 69
4.6 Feasible activations of the 6 muscle arm model with OpenSim's CMC solution plotted in red. These results are described in detail in Chapter 5. 71
4.7 Feasible accelerations of the Arm26 model hand mapped from feasible activations (Figure 4.6). 72
4.8 Plotting the upper and lower bounds of feasible activations computed by finding the upper and lower bounds for each muscle from the vertices of the feasible activation set over progression of gait. OpenSim's CMC solution is plotted in red. These results are discussed in detail in Chapter 5. 73
4.9 Vertices of $\mathbb{A} \mid \boldsymbol{\Gamma_{task}}$ (Figure 4.8) over each percent of gait are used to map to joint moments derived from ID. Only F_x,F_y,M_z are nonzero as the model is planar. 74

4.10 Software flowchart. 75

5.1 Four equivalent representations of the unit cube in $\mathbb{R}^3$. 80

5.2 Four equivalent representations of a convex polygon in $\mathbb{R}^2$. Note that the row in $\boldsymbol{\mathcal{H}}$**-FAS** $[1 \ \ -1 \ \ 0]$ is redundant as can be seen in the graph and may be removed from the $\boldsymbol{\mathcal{H}}$**-FAS** matrix. 81

5.3 A simple 1 DOF model with 2 muscle-tendon actuators, highlighting the muscle redundancy problem. There is more than 1 solution for actuator controls that can satisfy a trajectory through operational space. Activation space here is the unit square. 82

5.4 Diagram of the muscles of the Gait2354 model used to find $\boldsymbol{\mathcal{V}}$**-FAS** bounds (Figure 5.6). Biarticular muscles are between primary colors. 87

5.5 Revisiting Chapter 3. From the pseudostatic analysis, it is possible to investigate how muscle model complexity influences the bounds on FAS. . . . 89

5.6 $\boldsymbol{\mathcal{V}}$**-FAS** of the 24 leg muscles over the course of the gait cycle with the OpenSim CMC solution in red. 90

5.7 Software flowchart. 92

6.1 A typical multibody tree of a gait focused musculoskeletal model. The path to compute left knee loads is depicted in red. The linkage between the ground and the pelvis reflects the virtual DOFs that describe the MBS position and orientation relative to the $\mathbb{O}$ origin. 94

6.2 Joint relating the connection between body1 and body2 as a dashed line, and the muscle (red) applying tension to bodies (blue). 95

6.3 Similarly, $\boldsymbol{\mathcal{V}}$**-FAS** for an upper extremity model performing an elbow flexion with the hand supinated. Joint loads were determined using OpenSim JRA. 99

6.4 Diagram of the muscles of the Gait1018 model used to find $\boldsymbol{\mathcal{V}}$**-FAS** bounds in Figure 6.5 highlighting the most constrained muscles in bold. Biarticular muscles are between primary colors. 100

6.5 $\boldsymbol{\mathcal{V}}$**-FAS** for the right leg constrained by the JCF. JRA determined joint contact forces of the right knee only, which are functions of the muscle states derived from CMC. 101
6.6 Diagram of the muscles of the Gait2354 model used to find $\boldsymbol{\mathcal{V}}$**-FAS** bounds (Figure 6.7) highlighting the most constrained muscles in bold. Biarticular muscles are between primary colors. 102
6.7 Feasible JCF used to constrain the $\boldsymbol{\mathcal{V}}$**-FAS** using the joint contact force expression. 103
6.8 Chapter 6 software components. 105

7.1 Figure from Thelen2003 [96] (left) and Millard2012 [71] (right). Both of these models are very similar equilibrium-type muscle models with the main difference is that Millard2012 has additional parameters and toggles, such as the ability to ignore tendon compliance. 108
7.2 highlighting the a_{cc}, the center of the largest hypersphere that fits within $\boldsymbol{\mathcal{H}}$**-FAS**. 112
7.3 a_{cc} can often be a bad estimate of the center of $\boldsymbol{\mathcal{H}}$**-FAS**, particularly when the polytope is "thin". 113
7.4 Start at an interior point and find the λ to each halfspace. 115
7.5 Pick a unitary direction from the standard Gaussian $\boldsymbol{g}$. 116
7.6 Iteratively select points that lay within $\boldsymbol{\mathcal{V}}$**-FAS**. 117
7.7 the log barrier of the "home plate"-shaped $\boldsymbol{\mathcal{H}}$**-FAS** shown many times previously: $z = \sum_{i=1}^{m} \log(b_i - A_i^T a)$ 118
7.8 Two views of the bivariate standard Gaussian. 120
7.9 A multivariate skew normal with $\alpha = 10$ for both a_1 and a_2. 121
7.10 The euclidean distances from the upper and lower bounds to a_{t-1} is asymmetrical so the expected value sampling from the standard normal favors activation over deactivation. It is possible to trivially estimate the quartiles as functions of the euclidean distance from a_{t-1} to a_{lb} and a_{ub} and apply this method in a correction (Equation 7.20). 122

7.11 A flowchart of the FAST analysis is offered. Parameters can be tuned (Table 7.1) in order to find families of solutions which more closely match CMC (Figure 7.12), or they can be tuned to model abnormal control. 123
7.12 An investigation of how different base skew parameters on proposal distributions influence the output trajectories. 10 sample trajectories were drawn for each level. All the FAS trajectories depicted in the plot satisfy the ID determined constraints to within tolerance and additionally satisfy the first order $\boldsymbol{\mathcal{A}}$ dynamics determined window at each time point. 125
7.13 100 representative samples from FAS. FAST samples lay within the feasible bounds dictated in Chapter 5. 126
7.14 50 representative samples from FAST analysis 2354. Again, FAST samples lay within the feasible bounds dictated in Chapter 5. 127
7.15 FAST analysis of the Arm26 model, feasible activations are selected from the joint force constrained FAS. 128
7.16 FAST analysis of the gait1018 model using the joint loading constraint developed in Chapter 6, feasible activations are selected from the joint force constrained FAS. 129
7.17 Chapter 7 software components. 130

8.1 Simple 3D two body model. 134

Chapter 1

Introduction

Stand up. Can you tell me how you the reader controlled your muscles to perform this task? If another person stands up, can you tell me how they controlled their muscles to perform this task? How do the brain and body work together to decide what muscles to use in order to perform complex behavioral tasks? When computer scientists design artificial intelligence (AI) that can play games like Chess or Go, they use algorithms like mini-max where every possible board state is known or a subset of all board states is used to select a solution with the highest possibility of success. Similarly, the mammalian brain evolved a centralized location of motor planning known as the motor cortex which works in conjunction with aspects of the Central Nervous System (CNS) and the Peripheral Nervous System (PNS) to produce real-time solutions for purposeful movement; however, the exact relationship between the brain, the body, the muscles and output movement is not yet fully understood. Instead of making presuppositions about how the nervous system and body work together, this dissertation is an exercise in finding the chessboard where muscle control happens. As opposed to conventional techniques of using minimization problems to arrive at a singular optimized set of actuator controls, tools that explore families of all possible actuator controls that satisfy specific dynamic tasks were designed.

1.1 Motivation

In the near future, advances in computational capability will allow us to dictate the boundaries of control for complex humanoid systems in real time. The investigation of tools that can be used for the ad hoc analysis of musculoskeletal models has impact far beyond clinical motion analysis of humans; however, it is a stepping stone to a new functional control paradigms where the set of all possible controls is the game board, and aspects of the central and peripheral nervous system are players. This dissertation will explore methods of feasible sets analysis of musculoskeletal systems, with implications for novel real-time control of redundant manipulator systems.

1.2 Objectives and Approach

The purpose of this research is to design a framework for obtaining sets of feasible muscle activations based on the limitations posed by the dynamic task, the anthropometry, and the muscle parameters. The forward problem deals with forming matrices of configuration space or operational space parameters from sets of feasible muscle controls. The inverse problem involves constructing feasible sets of possible muscle activations from sets of constraints that describe configuration or operational space parameters in either the static or dynamic domains. To these aims, approaches from biomechanics, robotics and probability theory are synthesized into one comprehensive research product.

By using subject-specific models and a new algorithmic framework, it will lay a foundation rooted in gait analysis of humanoid systems, which can also be quickly adapted to include other complex behavioral tasks like running, jumping, reaching or stepping response, and also other arbitrary musculoskeletal models.

While *in silico* approaches to biomechanical problems give us incredible insights into control and the forces at work, true understanding lies in the application of obtained knowledge to the physical systems.

To facilitate rapid clinical and research applications, along with fostering a collaborative environment, this project will be hosted freely on GitHub. Although this research focuses on

the motion analysis of healthy populations, these activities will certainly impact other areas, including the design of wearable robotics, the modeling and mitigation of sports injury, and the testing of existing rehabilitation strategies *in silico* among others.

This work is novel as it investigates complex dynamic tasks like running or walking instead of previously studied static poses.

Additionally, it features a new probabilistic computed control algorithm which avoids the use of complex quadratic optimization or controllers.

1.3 Chapter Summary

This dissertation is organized as such.

In Chapter 2, fundamental concepts critical to understanding principles of biomechanical modeling are explored. Beginning with classical mechanics and then expanding to multibody dynamics. Then, more complex topics like Inverse Kinematics (IK) and Inverse Dynamics (ID) are explained. Optimization, vertex enumeration and Markov chain Monte Carlo methods, and their relevance to this work is explained. Finally, expanding to muscle models and the estimation of muscle forces over time.

In Chapter 3, a naive analysis of muscle end effector forces assuming static pose over each frame of a kinematic task is explored and explained as a valid and useful method of analysis for arbitrary tasks of various models.

In Chapter 4, the model of end-effector forces presented in Chapter 3 is expanded by including the body inertial forces and modifying the kinematic Jacobian such that it is dynamically consistent. Besides possible endpoint forces, set of muscle controls are shown to be mappable to sets of any operational space parameters by way of projection operators.

In Chapter 5, the inverse problem from Chapter 4 is evaluated. Generalized forces in configuration space are mapped to sets of possible muscle activations. The benefits and challenges of vertex enumeration over linear programming to the analysis of biological systems is also explained.

In Chapter 6, the framework from Chapter 5 is expanded by developing a method of applying an experimentally determined constraint on feasible activation sets: the joint

loading either synthesized *in silico* using OpenSim, or collected in the lab using subjects with instrumented joints, such as the subjects of the Grand Challenge data sets available on SimTK [40].

Further, in Chapter 7, muscle first order activation dynamics are used to constrain the sets of possible muscle controls available in each delta time. A Monte Carlo Markov chain method called Feasible Activation Space Trajectories (FAST) was developed.

An in-depth explanation of the tools and paradigms developed for this dissertation is provided at the end of each Chapter where relevant.

Finally, concluding remarks, limitations, and future work are summarized in Chapter 8.

Chapter 2

Literature Review

2.1 Principles of Biomechanics

2.1.1 Physics: Statics, Dynamics, Multibody Systems

The behavior of physical systems can be described in terms of the motion of particles as functions of time. To this aim, Newton offered three fundamental laws of motion. To start, a particle has a mass, a property that reflects on a body's resistance to changes in motion. Particles also have positions in Cartesian space (real world), an instantaneous change in position called a velocity, and a change in velocity called an acceleration. Forces are interactions such that when they are unopposed, change the motion of an object. Newton's first law or the law of inertia states that if the sum of the forces acting on an object is zero, then its velocity is also zero. Newton's second law states that changes to the momentum of a mass is directly proportional to the applied force, or equally that $\boldsymbol{F} = \boldsymbol{ma}$. Finally, Newton's third law, or the action-reaction law, states that all the forces between two masses exist in equal magnitude and opposite direction.

Colloquially, every action has an equal and opposite reaction. All three of these laws lay the foundations of classical mechanics. A moment is the product of the displacement to some point and a physical parameter at that point. [1] A coupled moment of force is a joint moment

[1] Most commonly moments in this dissertation are discussed with relation to multibody dynamics, but in chapter 7, a method of moments approach is used to estimate the skewness parameters of a multivariate skew normal distribution.

and is the product between a force and a distance known as a moment arm to that joint, or that $\boldsymbol{\tau} = \boldsymbol{r} \times \boldsymbol{F}$. Statics is the study and analysis of bodies at rest, or assumed to be at rest where Newton's second law is used to say that the sum of forces and moments acting on an object is 0. Dynamics is the study of bodies in motion: the sum of forces and moments acting on a body is equivalent to the product of the mass and acceleration of that body. Similarly, multibody dynamics is the study of systems of rigid bodies where externally applied forces can be summed to match the accelerations of those bodies. Generally, the motions of sets of bodies or particles as systems or models can be described by a set of independent parameters called generalized coordinates. Systems operate within the real world, or operational space $\mathbb{O}$, and their motion can be described by these generalized coordinates $\boldsymbol{Q}$ in coordinate space $\mathbb{C}$. The first time derivatives of generalized coordinates are called generalized speeds. From Lagrangian mechanics, one may derive sets of forces and moments known as generalized forces using the applied forces on a model and an equation that reflects the map between generalized coordinates and forces through computations of virtual work. While moments perform work over an angular distance, work itself is the product of a displacement and a force. Virtual work is the work that a force produces due to a virtual or infinitesimally small displacement. All of these aspects of classical mechanics and their consequences are used throughout this research.

2.1.2 Principles of Robotics and Bio-Inspired Systems

While robots ideally have one actuator or several coupled per degree of freedom (a motor or linear actuator), biological systems usually employ redundant actuators; they have many more muscles than degrees of freedom [18, 38]. Specifically, humanoid movement is complex as it requires the coordination of many muscles to produce a smooth, purposeful movement; however, the map from the neural commands to purposeful movement is not yet fully understood. There are many ways to approach a simple behavioral task as lifting a mug from a kinematics perspective. Similarly, there are many muscle activation patterns that are capable of producing the same joint configuration, and consequently, the same effect at a predetermined endpoint. Popular in biomechanical modeling, researchers rely on optimization frameworks that minimize an arbitrary cost function to arrive at a single solution for muscle

activations or forces that is either dynamically or statically consistent [35]. Even if researchers can solve for the system dynamics or kinematics uniquely, the actuator controls necessary to produce those motions cannot be solved uniquely in conjunction with the deterministic components of the generalized forces, such as the gravitational, centrifugal or Coriolis contributions. Trivially, there are a lot of ways one can use his or her biceps and triceps to hold the elbow in a static configuration. The Jacobian is a map between the generalized coordinates of a multi-body system and the Cartesian coordinates of the $\mathbb{O}$. However, there is also a minimum kinetic energy pseudoinverse of the Jacobian that can account for aspects of the Coriolis, centrifugal and gravitational effects called the dynamically consistent Jacobian inverse $\boldsymbol{J}^{+}$. $\boldsymbol{J}^{+}$ is found by weighting the Jacobian by the inverse mass matrix or the inertial matrix of the bodies of the system. For a set of generalized coordinates $\boldsymbol{\mathcal{Q}}$ in configuration or coordinate space $\mathbb{C}$, the generalized coordinate forces, $\boldsymbol{\Gamma}$, for an arbitrary system, can be constructed as follows:

$$M(\boldsymbol{\mathcal{Q}})\ddot{\boldsymbol{\mathcal{Q}}} + C(\boldsymbol{\mathcal{Q}}, \dot{\boldsymbol{\mathcal{Q}}})\dot{\boldsymbol{\mathcal{Q}}} + G(\boldsymbol{\mathcal{Q}}) = \boldsymbol{\Gamma} \tag{2.1}$$

Where $\boldsymbol{M}$ is the system mass matrix, $\boldsymbol{C}$ is a vector containing the contributions of centrifugal and Coriolis effects, and $\boldsymbol{G}$ is a vector of the contributions of gravitational effects. Aspects of humanoid movement like the dynamically consistent support constrained Jacobian, $\bar{\boldsymbol{J}}_{\boldsymbol{s}}$ have been implemented in the control of bio-inspired robotics, but has largely been unused in the analysis of models of human movement [68, 85].

2.1.3 Experimental Data to Subject Specific Model

The objective of biomechanical modeling is to construct physics-based simulations of behavioral tasks by finding the motion of a subject using camera data, and subsequently using Equation 2.1 to find the dynamics. Simulations provide an alternative or complementary method for evaluating complicated systems as compared to traditional experimentation. The primary strengths of simulations include the high volume of trials that can be completed in a short amount of time, as well as the diminished costs. Clinicians and researchers conventionally use data-tracking simulations of biological systems; however, ideally they

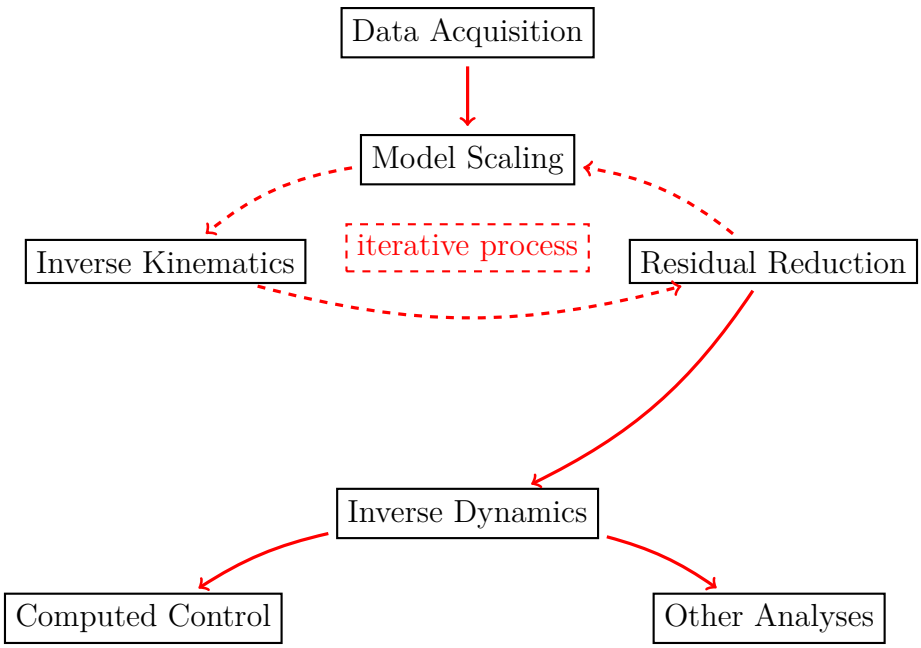


Figure 2.1: Modeling begins with data acquisition, and *in silico* models can only ever be as good as the acquired data allows.

both desire predictive simulations [120]. The first approach involves minimizing the error between the position of simulated points on the human body and their corresponding points from motion tracking data (Figure 2.1). The second approach requires the definition of some output parameter and the minimization of a performance variable, such as energy expenditure, to achieve the desired results. Chapter 7 of this dissertation deals with making predictive simulations that identify muscle activations. Both of these types of simulations have been used to help analyze and optimize the mechanics of sports and clinical rehabilitation, but these algorithms are generalized tools that are used in many fields from video game development to forensics. In particular, the clinical application of musculoskeletal modeling is challenging. Clinicians aim at understanding and/or preventing injuries, but have to use "fuzzy" data collected with sub-optimal tools like surface EMG or motion capture markers. However, simulation techniques allows researchers to gain insights into parameters that are necessarily impossible to explore through experimentation [77, 78, 79]. Some tertiary parameters cannot be easily collected such as muscle forces and joint moments

or loads; however they are essential to understanding how biological systems move in the world. Subject specific simulations of movement by tracking motion capture data has proven to be a powerful tool in determining these quantities.

Scaling

Model scaling is a procedure to scale the physical dimensions, masses and inertial properties of the bodies that compose a multibody system using subject specific measurements such that the error between the experimental data markers and the computer model markers is minimized. As the total length of the muscle-tendon unit is purely a function of the geometry, tools like OpenSim [30, 31] will scale muscle tendon unit (MTU) lengths to reflect the change in the dimensions of bodies that compose a multibody system. [2]

Inverse Kinematics (IK)

IK is the process of recovering the motion or kinematics of a subject in $\mathbb{O}$ from motion capture or camera data and mapping it to the configuration space $\mathbb{C}$ of the generalized coordinates of a model *in silico*. IK is a procedure to back out an estimation of the trajectory in $\mathbb{O}$ using experimental markers, and map it to a trajectory in $\mathbb{C}$.

Residual Reduction (RRA)

Models are often incomplete, either by design or by failure of the data acquisition. Besides techniques like adding additional reserve actuators to account for discrepancies in the recorded external loads and the dynamics of the model, RRA, scaling and IK are used iteratively to reduce the marker errors between the experimental marker data recorded through camera systems like Vicon and the virtual markers placed on a model of the subject *in silico*.

[2] It is also possible to improve the model fit by scaling the muscle parameters like the peak isometric force ($\boldsymbol{F^0}$) (which OpenSim's scaling tool will not change) using laboratory measurements of force or to scale model parameters using the cross-sectional geometry of the muscles obtained via fluoroscopy; however, these procedures are not explored in this work. They should be in future works.

Inverse Dynamics (ID)

ID is the process of identifying the generalized forces (Equation 2.1) that occur at each time frame along a model's trajectory in $\mathbb{C}$ as functions of the inertial and mass properties of the individual bodies and any externally applied forces that compose the model. ID generalized forces $\boldsymbol{\Gamma_{task}}$ are not the same as joint contact forces and this distinction is made important in Chapter 6. $\boldsymbol{\Gamma_{task}}$ output from ID are the joint torques necessary to move the model along the model kinematics in a forward dynamic simulation, whereas the joint contact forces are functions of the muscle activation. To constrain controls by the joint contact force is a more stringent constraint than to just constrain by the joint moment $\boldsymbol{\Gamma_{task}}$.

Computing Muscle Activations

The typical approach to studying multi-joint biomechanical systems involves the use of complex physical modeling with sophisticated softwares such as AnyBody or OpenSim [19, 31, 43]. In this dissertation, tools are designed using the OpenSim application programming interface (API) coupled with MATLAB or Python along with robust I/O in C++ which offer unparalleled flexibility for users and user-developers, but the paradigms employed here can easily be adapted to any choice of musculoskeletal modeling platform [58, 67, 68]. OpenSim is a powerful open-source multi-platform, multi-user space that allows users to design and simulate models human movement and musculoskeletal dynamics. OpenSim is notable because it possesses an algorithm for computing a possible set of muscle activations called computed muscle control (CMC). Computed Control is the process of back calculating the actuator controls or activations necessary to drive the model in a forward dynamic simulation along the trajectory specified by IK. Depending on the model complexity, computed control can involve determining the muscle fiber lengths and muscle activations by starting from an equilibrated model, or it can involve just muscle activations, ignoring the abstraction between muscle active and passive elements, and allowing users to perform an additional post hoc analysis to gain insight into tendon parameters, passive forces, and muscle properties.

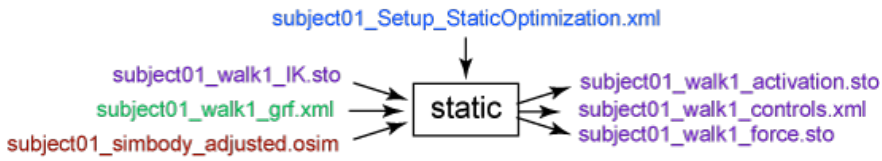


Figure 2.2: Inputs and outputs of the static optimization procedure featured in OpenSim from the simTK confluence site [88].

Static Optimization (SO) SO (Figure 2.2) uses $\boldsymbol{\Gamma_{task}}$ to solve a linear program (LP) to find a valid set of controls. In SO, the constraints (Equations 2.2, 2.3) are similar to the structures used in Chapter 3.

$$\sum_{i=1}^{m}(\boldsymbol{a_{m\times 1}} \odot \boldsymbol{F^0_{m\times 1}})\boldsymbol{R_{m\times c}} = \boldsymbol{\Gamma_{task}} \tag{2.2}$$

Additionally, SO can be expanded to concern with more complex muscle parameters:

$$\sum_{i=1}^{m}(\boldsymbol{a_{m\times 1}} \odot f(\boldsymbol{F^0}, \boldsymbol{l^M}, \boldsymbol{v^M}))\boldsymbol{R_{m\times c}} = \boldsymbol{\Gamma_{task}} \tag{2.3}$$

The key distinction between the techniques used in this dissertation and SO is that SO uses $\boldsymbol{\Gamma_{task}}$ to decompose the joint moment by each muscle contribution according to a performance criteria (Equation 2.4) whereas this dissertation avoids optimizing controls.

SO uses the objective function:

$$\min \quad \sum_{i=1}^{m} \boldsymbol{a_{m\times 1}}^P \tag{2.4}$$

Where each a_i is a muscle activation which is minimized according to an arbitrary constant P. By contrast, in Chapter 5, FAS computes the vertex enumeration of activation space boundaries using LRS which performs a pivoting algorithm known as reverse search [10].

Computed Muscle Control (CMC) CMC is another optimization-based muscle coordination strategy generator which works by determining the activations necessary to tend

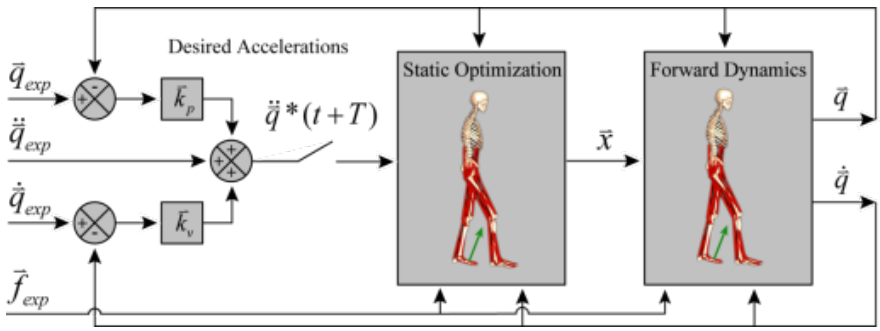


Figure 2.3: Controls system featured in OpenSim CMC from the simTK confluence site [88].

generalized coordinates in the current discrete time of a dynamic musculoskeletal model configurational space toward a desired acceleration command [97].

CMC finds the muscle activations using a combination of SO and PID control law:

$$\ddot{\boldsymbol{Q}}^{*}(t+\delta t) = \ddot{\boldsymbol{Q}}_{\boldsymbol{exp}}(t+\delta t) + \boldsymbol{k_v}[\dot{\boldsymbol{Q}}_{\boldsymbol{exp}}(t) - \dot{\boldsymbol{Q}}(t)] + \boldsymbol{k_p}[\boldsymbol{Q}_{\boldsymbol{exp}}(t) - \boldsymbol{Q}(t)] \tag{2.5}$$

CMC is performing an ad hoc forward dynamic-type analysis (Figure 2.3) where it tries to drive the model in the direction of $\ddot{\boldsymbol{Q}}^{*}$ and minimize the error between the simulation and the experimental data derived generalized coordinates, speeds and accelerations. As per Equation 2.5, CMC identifies the necessary changes to the actuator states needed to produce $\ddot{\boldsymbol{Q}}^{*}$. CMC's SO procedure features two possible objectives known as the fast (Equation 2.7) and slow (Equation 2.6) targets.

The slow target:

$$\min \quad \sum_{i=1}^{m} \boldsymbol{a}^2 + \sum_{j=1}^{c} \boldsymbol{w}(\ddot{\boldsymbol{Q}}^{*} - \ddot{\boldsymbol{Q}})^2 \tag{2.6}$$

The fast target:

$$\begin{gathered} \min \textstyle\sum_{i=1}^{m} \boldsymbol{a}^2 \\ \text{s.t.} \quad \forall j \in c, |\ddot{\boldsymbol{Q}}^{*} - \ddot{\boldsymbol{Q}}| \leq \epsilon \end{gathered} \tag{2.7}$$

The slow target features a computationally expensive optimization while the fast target avoids this issue by allowing a tolerance on $\ddot{\boldsymbol{\mathcal{Q}}}$.

In Chapter 7, a novel computed control method called Feasible Activation Space Trajectories (FAST) is developed that trades the use of controllers and objective functions for sampling from Gaussian distributions. FAST is similar to SO as it returns sets of muscle activations without returning fiber lengths information; however, FAST returns many possible trajectories that satisfy the first order activation dynamics and the ID constraint and similar to SO, the tendon analysis or muscle analysis can be performed post hoc. If the information about the model kinematics ($\ddot{\boldsymbol{\mathcal{Q}}}, \dot{\boldsymbol{\mathcal{Q}}}, \boldsymbol{\mathcal{Q}}$) and the applied externals, and by extension the $\boldsymbol{\Gamma_{task}}$ are known, FAST relies on sampling from a probability distribution that maps to locations within feasible activation space.

2.2 Optimization and Sampling Frameworks

It can help to understand the underlying mechanisms of CMC, SO, and the tools of this dissertation. While mathematicians like Gauss [41] and Newton offered iterative methods of arriving at an optimal solution for a problem, the concept of linear and nonlinear programming, and formal mathematical optimization were products of necessity of the Second World War. Optimization was born out of the adage: "America faced two chief problems in WWII: Atlantic and Pacific." How do Americans ship 15 metric tons of M3 Stuart across the ocean to counter the German offensive in the most efficient way possible? Mathematicians like Dantzig realized that systems of inequalities had spaces or sets of solutions, but it took the brilliance of people like Dantzig to realize that military commands could be mapped mathematically as objective functions. At the same time that science is waking up to nuclear energy, mathematics is formalizing optimization. After the development of the simplex algorithm by Dantzig in 1947 [25], there was a Cambrian explosion of various methods of solving optimization problems that have largely been forgotten due to their inferiority with the simplex and dual simplex methods. Methods that iterate on the interior of feasible space are often called interior-point methods. On the contrary, Simplex methods travel on the edges of feasible space. Duality, often credited to Dantzig, Tucker, Kuhn, and

Gale, or Von Neumann, is the beautiful relationship of linear programs with systems of the form $Ax \leq b$ where for a primal problem (Equation 2.8) with an objective function C and inequalities $b \geq A$:

$$\min \begin{bmatrix} C & \\ A & b \end{bmatrix} \tag{2.8}$$

That there is a dual problem (Equation 2.9) where the constraints from the primal become the variables of the dual problem, and variables of the primal similarly become constraints of the dual:

$$\max \begin{bmatrix} b & \\ A^T & C \end{bmatrix} \tag{2.9}$$

Additionally, The objective direction becomes inverted: minimization of the primal problem maps to a maximization of the dual. Duality states that if the primal or dual has a solution, then the other also has a solution and that the optimal values for both problems are equal. These ideas are exploited in this research, primarily in Chapter 7.

As for the role of optimization in biomechanics [48, 49], it is likely that the CNS and the inherent mechanical properties of the musculoskeletal system work in conjunction to select feasible sets of muscle activations, and that the CNS is probably doing much more than simple cost minimization. Prior research has shown that least squares minimization of the sum of muscle activations while accounting for residual boundary errors, or objective functions that minimize metabolic or energy expenditures have been very effective in developing simulations with high biomechanical fidelity that are sometimes able to closely mimic the EMG after signal processing [97, 98]; however, it is much more plausible that the motor cortex coordinates muscles not based on energy expenditure, but based on many elements one of which can be that optimization; However, this aspect of neuromechanics is hotly debated: see Section 2.3.

2.2.1 Linear Programming (LP)

Also known as linear optimization, LP is a method of solving linear systems subject to a cost function that sets a direction and level, navigate a feasible space and find a minimized value.

In LP, the typical form of such an inequality is:

$$\begin{gathered} Ax \leq b \\ \text{s.t.} \\ \forall x_i \in x, x_i \geq 0 \\ \forall b_i \in b, b_i \geq 0 \end{gathered} \tag{2.10}$$

Note that the optimization variables and constraints are strictly positive in the canonical form; however, there exist tools that one can use to deal with negative values.

In Equation 2.10, the expression

$$Ax \leq b$$

Is dual to the expression

$$-Ax \geq -b$$

And the expression:

$$Ax = b$$

Is dual to the joint expression

$$\begin{gathered} Ax \leq b \\ -Ax \leq -b \end{gathered}$$

Similarly, for the objective itself:$\max x$ is dual to $\min -x$.

The inequality $Ax \leq b$ can be transformed into an equality statement by introducing a slack variable: $Ax + s_i = b$. In the cases where $Ax \geq b$, artificial variables can be introduced:

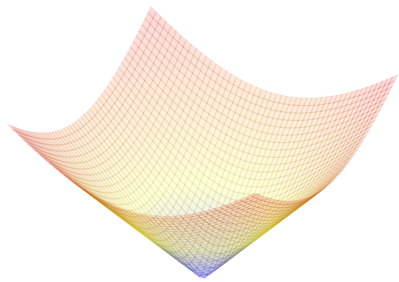


Figure 2.4: the simple quadratic cone $z^2 \geq \sum a_i^2$.

$Ax - s_i + v_i = b$. The concepts of introducing slack and artificial variables to transform a set of inequalities into a matrix that can be row operated upon are core concepts in LP. [3]

The application of LP to solving kinematics and dynamics problems is well-founded. Principle of least action, energy minimization, etc are all good presuppositions about how mechanical system function. The problem comes from when attempting to use optimization to gain insights into biological control. Again, it is possible for people to flex biceps and triceps to perform an elbow flexion, or to relax biceps and triceps to perform the exact same elbow flexion, the only difference being tertiary parameters that are not captured in the kinematics like joint forces or fiber stiffness: parameters that cannot be captured in the laboratory setting without fluoroscopy or other forms of instrumentation.

2.2.2 Quadratic (QP), Conic (CP) and Nonlinear Programming (NLP)

QP is a logical extension of LP to linearly constrained quadratic systems and can be considered a subset of general nonlinear optimization. Similarly, CP is an optimization over the intersection of the reals and a convex cone.

[3] The idea of injecting artificial and slack variables is the fundamental aspect of solving simplex methods, which is not necessary per se in this research, but is related to the double description and reverse search methods used in the Computational Geometry/vertex enumeration problem. One may also think about these variables as the "residuals" of the system. LP,CG, and later in Chapter 7, Markov chain Monte Carlo (MCMC) methods are all very intrinsically linked by the nature of interior point methods and simplex methods.

Cones (Figure 2.4) are just special subsets of vector space where any positive linear combinations of the elements will also result in an element of the cone. Conic programming is a nice tool for optimization because it allows mathematicians[4], to optimize nonlinear objectives and constraints by describing the program in the domain of the cone. The following program is used in Chapter 7 for activations a_i and constraints of the form $b - Aa \geq 0$:

$$\max_{a} \quad \sum_{i=1}^{m} \log\left(b_i - A_i^T a\right) \tag{2.11}$$

Equation 2.11 can be modeled as a cone program:

$$\max_{\alpha}(b_i - A_i a, 1, \alpha_i) \in \mathbb{K}_{exp}$$

Where $\mathbb{K}_{exp}$ is the exponential cone described by $a_1 \geq a_2 e^{\frac{a_3}{a_2}}$ where $\alpha \geq e^x \Leftrightarrow (\alpha, 1, a) \in \mathbb{K}_{exp}$ is a section of $\mathbb{K}_{exp}$.

MATLAB users should be familiar with optimization tools such as fmincon, a gradient-based nonlinear optimization problem solver that is frequently used in the literature.

NLP solvers are nice for finding optimal solutions of nonlinear systems like the map from muscle parameters (fiber length l_m, fiber velocity v_m, activation a) to muscle forces f_m. NLP suffers from the same problem as LP since a parameter gets optimized, but it does not necessarily reflect the black box that defines neural control. CP is however used in Chapter 7 to find the analytical center of the boundaries of feasible activations.

2.2.3 Vertex Enumeration

Vertex enumeration involves solving every exterior point of an LP problem without explicitly selecting a cost function, and in that way, vertex enumeration through computational geometry (CG) is cost-agnostic, yet, extremely computationally costly. Currently, some of the best tools available for performing vertex enumeration of linear systems are LRS and MPLRS [9, 10, 8, 13, 14]. LRS uses an algorithm called reverse search to navigate on the edges of feasible space [12]. This is very intrinsically related to the simplex method for LP

[4] And Wall St. quants.

where pivots on the simplex tableau navigate along edges of feasible space [11]. There is a drought in the biomechanists toolbox for feasible space solvers [16].

2.2.4 Markov Chain Monte Carlo Methods

In probability theory, Markov Chain Monte Carlo (MCMC) methods are just classes of algorithms that deal with sampling from various types of distributions. "Central" points within a linear system are of particular importance in MCMC, because starting a Markov chain at a more central position generally reduces the mixing time or the number of iterations until the sample distribution is close to the distribution from which proposals for future steps are drawn. The hit-and-run (HAR) algorithm [2] explored in Chapter 7 arrives at the steady state distribution in at most $\mathcal{O}(d^2\gamma_\kappa^2)$ as a function of the number of rows of a matrix defining the linear system, d, and the matrix condition number γ_κ. In Chapter 3, some strategies like Tikhonov regularization are briefly touched to account for singular configurations during a quasi-static analysis; however, this strategy is not good for multibody systems control in the dynamic domain. It is very typical to deal with systems that cross singular configurations like the full extension of the elbow or the knee which causes blow-ups to the mechanical advantage and equivalently, the matrix condition. The Dikin walk later explored in Chapter 7 arrives at the steady state distribution within $\mathcal{O}(dm)$ [22] from a warm-start. Note that the Dikin walk is condition number invariant or similarly, that it is affine-invariant. Linear transformation or conditioning applied to the activation space has no effect on the mixing time of the Dikin walk. Warmness is just a measure of a markov chain for how close the initial distribution is to the proposal distribution where future steps are drawn from. As the number of samples taken from the proposal distribution approaches inf, the sample distribution approaches the shape of the proposal distribution.

2.2.5 SLAM: Simultaneous Localization and Mapping

SLAM is an MCMC-based technology that runs roomba®. As a robot moves in $\mathbb{O}$ (Figure 2.5), the SLAM problem is to estimate the $\mathbb{O}$ parameters $\boldsymbol{\mathcal{X}}$ that define the state of the robot and to iteratively update a map which describes the environment particularly in situations

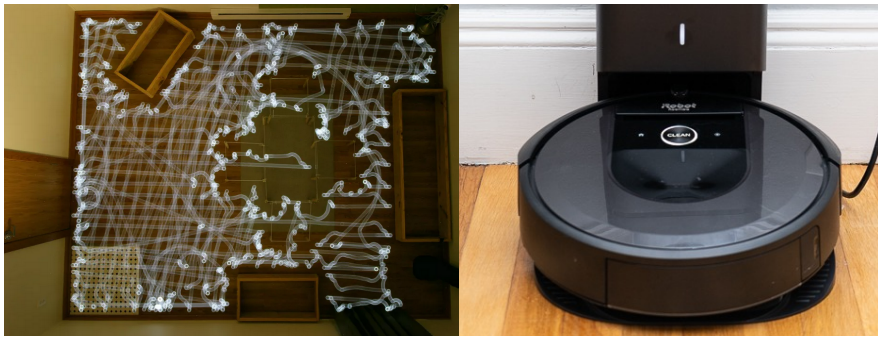


Figure 2.5: As a roomba cleans a room, it makes a map of the space by sampling points in $\mathbb{O}$ [ben].

where the true state or the true boundaries of the space cannot be known. There are many approaches to solving the SLAM problem which generally involves some method of MCMC [15].

Kalman Filter Also known as linear quadratic estimation, the kalman filter is an approach to solving the SLAM problem and involves two key steps [111]. First, in the prediction step, a kalman filter estimates what the current position of the robot is in $\mathbb{O}$ along with the uncertainty of these estimates. Second, the observations of the robot's actual state are used to weigh updates to the future frame guess. This process is iterative. Imagine a spacecraft in a far off solar system with NASA scientists controlling the craft on Earth. Kalman filters are considered a type of Hidden Markov Model as the only information related to $\mathbb{O}$ parameters $\boldsymbol{\mathcal{X}}$ is data gathered from the craft's sensors returns and that information is the only data which can be used to perform course corrections. By contrast, in Chapter 7, a novel computed control algorithm called Feasible Activation Space Trajectories (FAST) exploits that the state space that describes the set of all possible muscle activations available at each discrete time is wholly defined already based on the relationship between the muscles and the $\boldsymbol{\Gamma_{task}}$ necessary to drive the model along the kinematic trajectory and definitionally constructed: as if the roomba already knew what the room looked like.

2.3 Neural Control

This research deals with finding the landscape where the brain and body work together to produce purposeful movement. To generate a purposeful movement, first, a neural command is given to excite certain muscles, the electrical potential of which can be recorded with electromyography (EMG) [116]. Second, muscle-tendon dynamics based on length (l_m) and velocity (v_m) properties of the muscle and tendon produce muscle forces. Third, musculoskeletal geometry defines the location of joints, the direction of muscle forces, and muscle moment arms to produce joint moments. Lastly, given these moments, multi-joint dynamics determines accelerations and reactions producing the movement observed. How the human body moves affects subsequent neural commands that adjust the movement, and further affects the length and velocity of each muscle-tendon, the direction of muscle forces and moment arms, and the resulting dynamics of the multi-body system.

Central Nervous System (CNS)

Purposeful movement emerges from the synthesis of the sensory data of the surroundings and neural commands issues by the CNS and primarily the motor cortex. The CNS consists primarily of the brain and spinal cord. It accumulates the sensory data of experiences like sights and proprioception (awareness of a person's own body in the world) and uses it in some capacity to dictate the signals to send to the peripheral nervous system.

Peripheral Nervous System (PNS)

The PNS can be subdivided into the somatic and autonomic nervous systems. The autonomic nervous system (ANS) largely governs the involuntary biological responses: Things like heart rate, digestion, and the fight-or-flight response. Regarding musculoskeletal control, the somatic nervous system (SNS) is more important to consider in models as it governs voluntary movement of skeletal muscles (muscles that attach to bones via tendons and govern purposeful movements) While the ANS governs involuntary motor control, it does work in conjunction with the SNS. The stove feels hot and sensory neurons relay sense data, the ANS induces changes to organs in acute stress response or in prediction to pain, and the

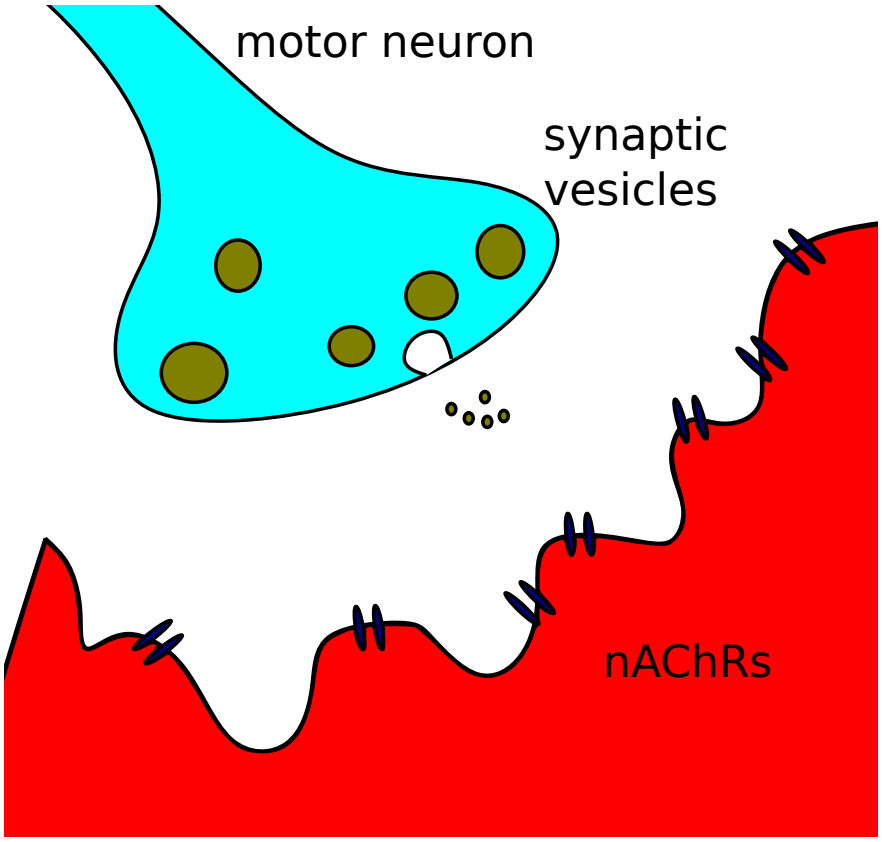


Figure 2.6: The electrical signal of the action potential of the motor neuron is transformed into a chemical signal at the neuromuscular junction by way of receptors like nicotinic acetylcholine receptor (nAChRs) and Ca^{++}-gated channels.

SNS participates via reflexes making the hand move away. Then other pathways work to return the body to homeostasis.

Unfortunately, while these overarching pathways are known, the exact relationship between the CNS, PNS, and body is not yet fully understood.

Nerves

Neurons are the fundamental unit of a nerve and are specialized cells within the body that carry electrical impulses. The nervous system consists primarily of three basic types of neurons each with different function: sensory neurons, motor neurons, and interneurons [42]. Sensory neurons carry impulses from tissues and organs to the CNS. Interneurons

connect various cells within the CNS. Motor neurons relay impulses from the CNS to effector cells. Nerve signals are just electrical signals called action potentials that propagate along a neuron's dendrites. Specifically the motion of Na^+ and K^+ during the action potential induces the release of neurotransmitters at synapses that induces further signal transduction.

Modeling Excitation and Activation

Muscles are composed of units called motor units. These motor units are coordinated together via depolarization or action potentials to produce force; however, muscles cannot simply produce forces instantaneously [45]. The transduction of action potentials along motor neurons and the subsequent formation of actin-myosin cross bridges within muscle myofibrils produces the forces whose reactions can be observed in the lab (Figure 2.6) [96]. This accumulation of Ca^{++} ions that produces the cross bridge formation is referred to as muscle activation. The firing of action potentials of the motor neurons is known as excitation. The relationship between muscle activation and excitation is often modeled as a first order dynamical system [5]. Despite musculoskeletal systems displaying high levels of redundancy, there is a gap in the knowledge as to how or if this redundancy is 'considered' by the CNS when the physiological parameters like the muscle strength are altered by disorder or disease [80]. Previous studies have shown any particular surgical or therapeutic approach may not improve function in all individuals equally and particular surgical or therapeutic approaches that aid one patient, may harm another [56, 57]. Because the joint configurations can be produced by a wide combination of muscle activations, which can be represented as FAS to achieve the same functional tasks, it is important to develop subject-specific treatment methods. Gait impairments which reduce an individual's gait symmetry or weaken muscles, change the volume of FAS. Patient-specific modeling and simulation along with predictive tools using feedback control have been previously used to design new metrics that aim to improve surgical and rehabilitation outcomes [68]. Ultimately, no studies have laid out a standardized and individualized metric that would quantitatively describe exactly how a doctor should modify the musculoskeletal system to achieve normal function which this paper seeks to lay out the groundwork to perform. Also, there have not been parallels

[5] This relationship is further explored in Chapter 7.

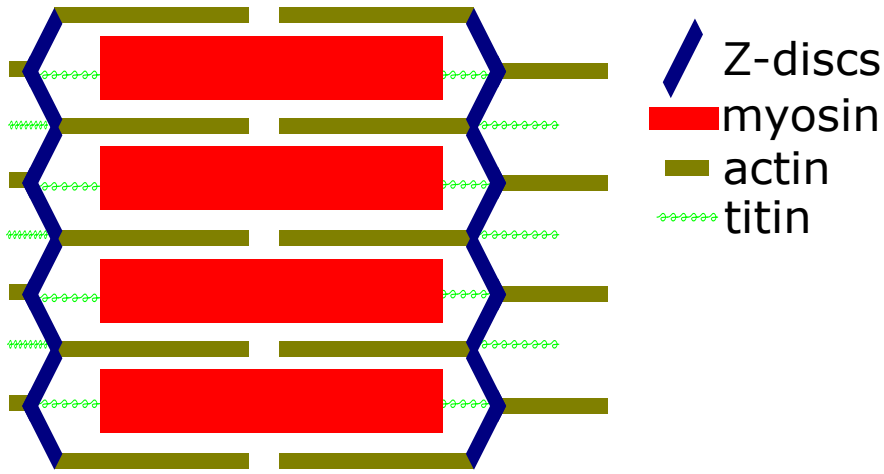


Figure 2.7: A muscle sarcomere produces force by binding crossbridges between actin and myosin filaments. Titin forms an elastic element within the muscle fibers themselves. Many sarcomeres together in parallel and series form a muscle.

drawn between the feasible neural command space, which is distinct from FAS, and the measured EMG, making it difficult to investigate the neural origin of muscle coordination [107, 108, 110].

Thelen2003 [96], and Winters1995 [115]:

$$\frac{da}{dt} = \frac{u - a}{\tau(a, u)} \tag{2.12}$$

Muscles and Muscle Models

Generally when researchers devise models that map the parameters of a muscle to the output force, they have to consider these various active and passive elements (Figure 2.7). Experiment characterized muscle moment arms, or the relationship between muscle force and the joint moment it subsequently produces [117]. Musculoskeletal models (Figure 2.8) rely on muscle tendon actuators to apply tension to bodies along lines of action. Typically, researchers use Hill-type models (Equation 2.12) to simulate first order activation dynamics [5, 6, 7].

Generally, when researchers are making a mathematical model of biological phenomena, they want to design something that is robust enough to accurately reflect the biological reality

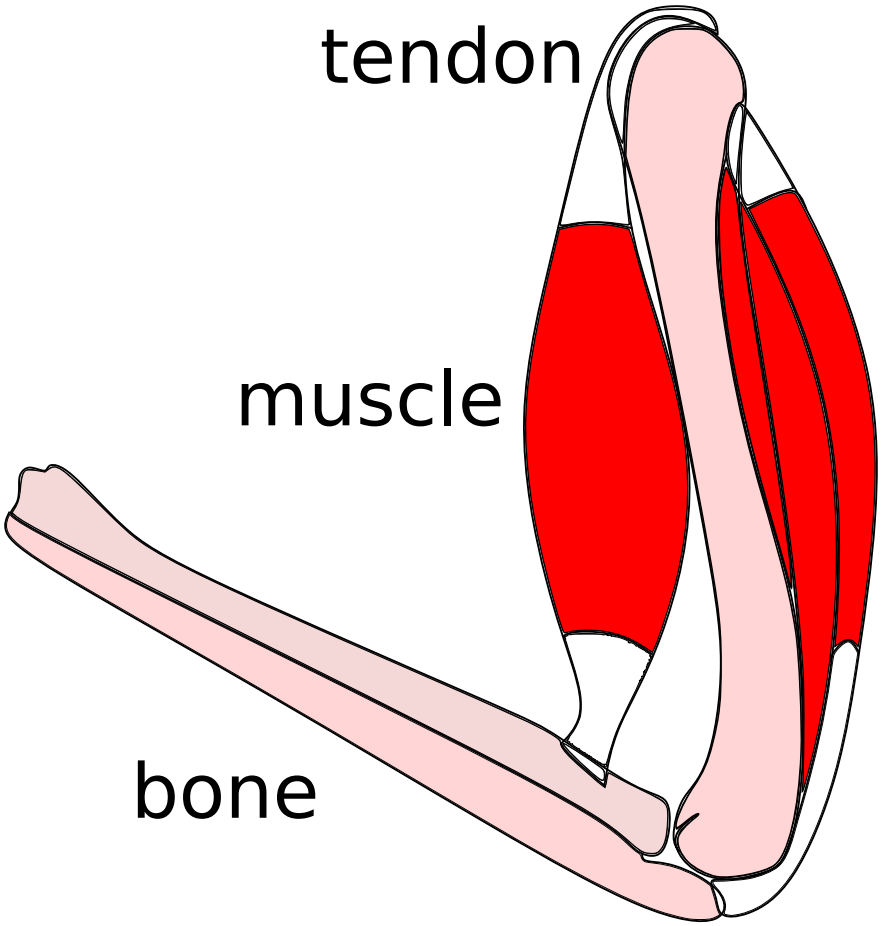


Figure 2.8: Muscles attach to bones via tendons. Tendons are modeled as elastic elements in Hill type models of the MTU. The total length of the MTU is purely a function of the geometry of the relevant bodies, but the individual muscle fiber length and tendon slack lengths are functions of the muscle state along with orientation of the bodies.

under various test cases, without compromising too much on computational speed. The abstraction of muscle excitations from motor neurons (Figure 2.6) to the muscle activations (reflection of Ca^{2+} ions in muscle cells) is often represented as a 1st order differential equation; however, surface EMG (sEMG) collected in the lab itself is subject to a rigorous process of normalization, rectification and application of signal filter. Muscles in models like Millard2012 [71] or Thelen2003 [96] have two states: fiber length (l_m), also known specifically as the length of the muscle of the muscle-tendon unit and not the total length of the MTU[6], and the before described muscle activation level, a.

Muscles operate on a force length velocity curve (Figure 2.9) from which two gains , c_l and c_v can be extracted which reflect on the changes to force due to the normalized length of muscles and the velocity of muscle fibers [5, 6]. By assuming an inextensible tendon during a particular instantaneous time, muscle force can be computed by:

$$\boldsymbol{f_m = f_0(a\mathrm{f}^L(l_m)\mathrm{f}^V(v_m) + \mathrm{f}^{PE}(l_m))\cos\alpha} \tag{2.13}$$

Briefly, some competing theories of recruitment strategies are explored.

Muscle Synergies Theory

Muscle coordination strategies that attempt to reflect the biological reality as opposed to the optimal possibility need to consider how the CNS might choose which muscles to use when constructing a muscle coordination algorithm to determine realistic feasible activations [57, 118]. One hypothesis is that motor units are organized at the neural level instead of mechanical and that the CNS decides which muscles to use for actively coordinated movement based on muscle synergies [70, 95, 102, 103]. In the muscle synergy hypothesis, the brain organizes muscles at the neural level into groups according to their function [23]. Instead of the CNS and PNS controlling individual muscles, it is thought that the nervous system controls muscle groups by way of the agonist muscles. Typically, musculoskeletal systems are highly redundant in that there are generally more muscles than there are kinematic degrees of freedom or joints and the muscle synergy hypothesis is an answer to the apparent paradoxes

[6] which is exactly determined as a function of the geometry

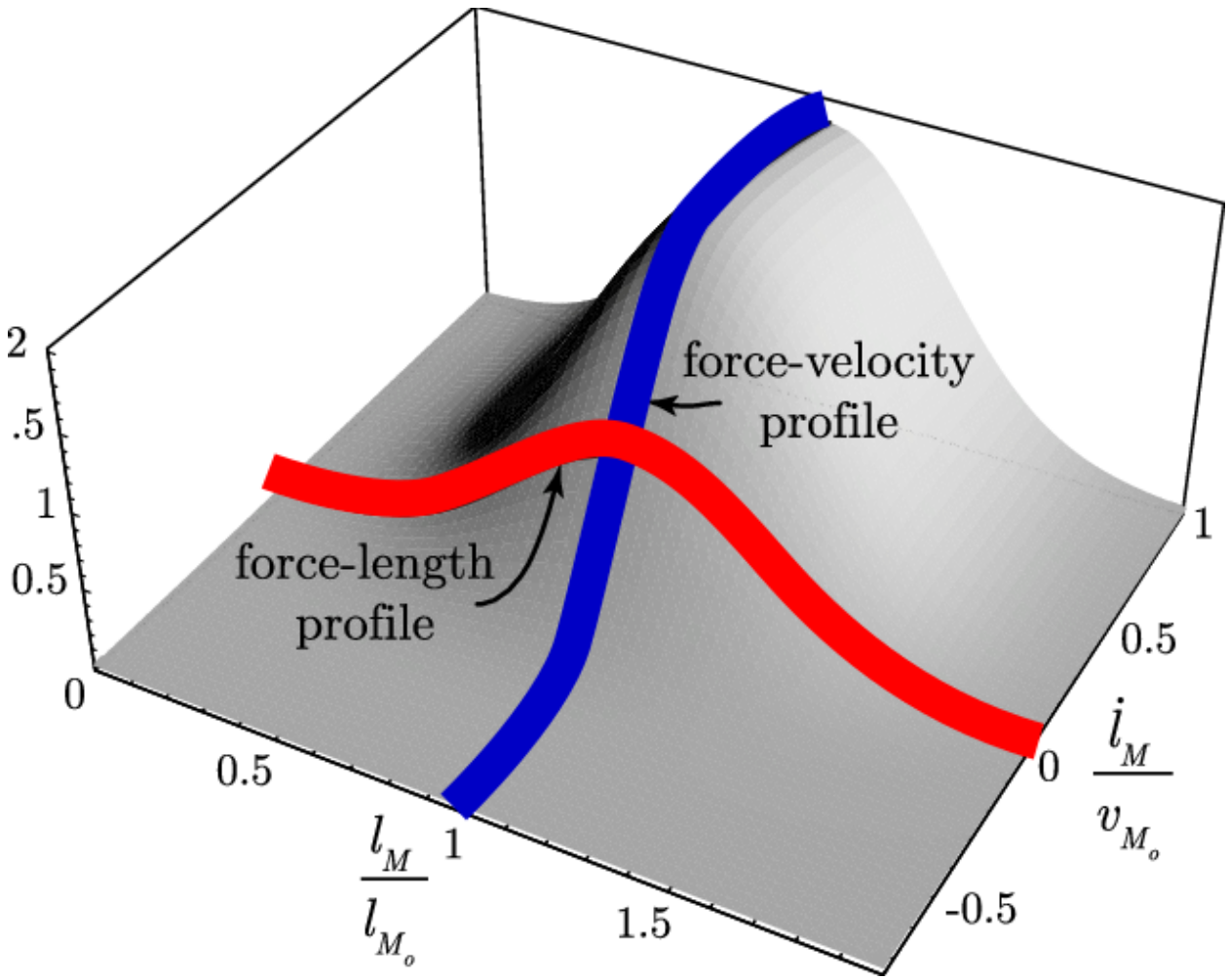


Figure 2.9: From De Sapio 2008 [28]. A muscle's active force is a function of the velocity and length of the muscle fiber, and the muscle activation a. The passive component of the MTU's force is just a function of the fiber length.

of control imparted by muscle redundancy. Despite the body being highly redundant at the mechanical level, both neural and physiological attributes work in conjunction to produce purposeful movement; however the creation of synergies that can mimic the kinematics or EMG does not imply that CNS is using muscle synergies. Muscle synergies analysis is similar to principal components analysis. Of course if the number of components is sufficiently large enough, it is possible to describe the system closely: it is a mathematical principle that always works. A gap in the knowledge exists in the exact roles that each aspect of the nervous system play in force selection. The efforts developed in this project will help to shed light on the muscle synergy hypothesis and whether muscle coordination is at the neural or at the muscular level, but ultimately it will shed light on how the motor cortex deals with muscle redundancy.

Cost-Minimization Theory (CMT)

Cost minimization theory of neural control assumes the *a priori* presupposition that the nervous system is attempting to minimize a parameter like energy or tension in the MTUs over the course of a behavioral task [105, 106].

Task Prioritization A typical approach used in the control of bio-inspired systems or continuum robotics is the use of a primary task such that the end effector moves towards the desired goal with an additional secondary task of minimizing a parameter as in CMT. Task prioritization is a popular control paradigm stemming in humanoid robotics with the presupposition that the mind has a hierarchical structure of tasks that which it satisfies over the course of a kinematic task and that behavioral modifications mid-task are simply reorientation of weightings on specific tasks in the prioritization scheme. Tasks can encompass a wide assortment of basal behaviors like ”keep the head above the chest” which matches observed reality and runs contrary to using solely CMT [27, 84, 85].

Feasibility Theory

A synthesis of the biomechanist's toolbox and feasible sets analysis arrives with feasibility theory: a conceptual and computational framework for understanding and describing feasible

activation spaces [24]. This framework is novel in that it poses as an umbrella for several of the before-described theories on neuromuscular control that all attempt to characterize the high dimensional space that bounds possible controls. Feasibility Theory has been demonstrated in feasible activation space of the seven muscles of the index finger producing static fingertip force consists of a 3-dimensional polytope in 7-dimensional space; however, this paradigm is applicable to all tendon-driven systems [52].

The introduction of feasibility Theory is particularly impactful, since it tackles the challenge of describing the nature of high-dimensionality of feasible activation spaces. This dissertation adapts paradigms conceived in feasibility theory and constructs tools that allow for the feasible sets analysis of dynamic tasks. Via spatiotemporal tunneling (Figure 2.10) as theorized in [24], each frame of the kinematic task is assumed to be in dynamic equilibrium where the inertial forces can be solved uniquely while the muscles' contributions are state-dependent and described as an n-dimensional space. Each individual frame volume can be overlapped or connected via the shortest euclidean distance between frames which can give insights into control.

Previous research in feasible sets analysis of musculoskeletal system has largely focus on models assuming static poses or under the assumption of statics over the course of a kinematic task. By contrast, this dissertation provides expansions on feasible sets analysis for the exploration of arbitrary dynamic tasks of arbitrary models. In feasible sets analysis of manipulator systems, the muscle moment arms, peak isometric forces and the kinematic Jacobian based on the generalized coordinates to construct feasible spaces. These feasible spaces can be constructed using computational geometry and describe explicitly what is allowed in terms of muscle activations, joint moments, and end effector generalized forces (forces and moments). Feasible sets analysis provides us with a rigorous framework for static task analysis that can be built up to study sets of neural commands for dynamic tasks. The bounds of muscle activation for tasks like gait have been previously described in the literature [87, 89]; However, instead of robustly defining the activation space through CG, researchers used LP to find the upper and lower bounds on feasible controls [90], which are both lower fidelity data sets and essentially useless for constructing controls paradigms, but does have utility for identifying muscles that are necessary for a specific task. Prior static analyses of

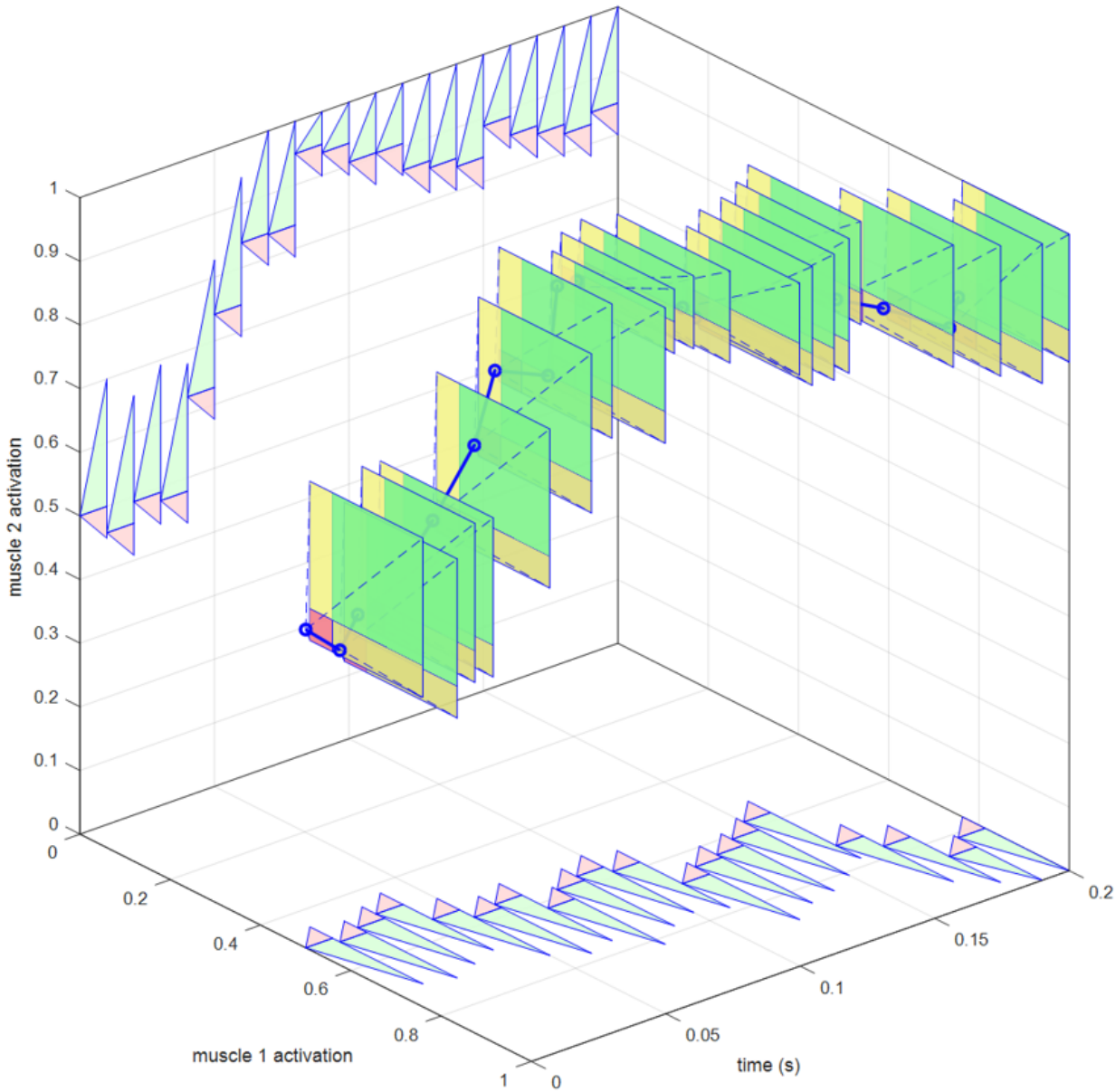


Figure 2.10: Spatiotemporal tunneling through sets of feasible controls over time.

redundant manipulator systems highlight without any assumptions about control very few if any muscles are necessary to perform static tasks. While this assumption is incorrect, it can provide some insight into dynamic tasks and feasible areas are robust regardless of the assumption.

Current research is moving towards the synthesis of the robotics tools for dynamical analyses and feasible sets analysis to develop dynamically consistent sets of acceleration-based parameters instead of singular optimized or minimized solutions that don't always reflect the biological reality. Conveniently, moving from moments to end effector forces is a linear map so operations like $\bar{\boldsymbol{J}}_{\boldsymbol{s}}^{\boldsymbol{T}} \setminus \boldsymbol{\Gamma}_{\boldsymbol{task}} = \boldsymbol{F}$ can be performed to find the dynamically consistent generalized forces $\boldsymbol{F}$. However, the inverse operation is complicated because an infinitely large set of feasible activations yield the same end effector generalized force. To sidestep the issue posed by having multiple competing theories of neural control that can not be immediately validated feasible sets analysis is a higher order abstraction that make no assumptions about neural control and runs in parallel with any of these theories. Many researchers are moving away from methods that perform cost function-based analysis, and are shifting towards a cost-agnostic analysis.

2.4 Feasible Sets Analysis

Feasible sets analysis through LP, QP, CP, or CG are popular methods of problem solving that have been applied to many fields. A feasibility analysis can be considered a borrowed term from finance. These tools are cost-agnostic methods of analysis that find families of solutions subject to some specified boundaries. At the most elementary level, what are the different ways one can spend 10 USD on apples and oranges, where oranges ($\mathcal{O}$) are 2.00 and apples ($\mathcal{A}$) are 1.00 USD?

Three nontrivial boundary conditions become immediately obvious (Equations 2.14).

Spend some money on $\mathcal{A}$, spend some money on $\mathcal{O}$, or spend some money on $\mathcal{O}$ and $\mathcal{A}$.

$$
\begin{aligned}
0 \leq 2\mathcal{O} \leq 10 \\
0 \leq 1\mathcal{A} \leq 10 \\
2\mathcal{O} + 1\mathcal{A} \leq 10
\end{aligned}
\tag{2.14}
$$

This method of analysis can be quickly adapted to any redundant system including the analysis of muscle controls in musculoskeletal modeling. Ex.:What are the different ways I can activate muscle $1(m_1)$ with moment arm r_1 , and muscle $2(m_2)$ with r_2 such that my joint moment is 10 Nm?

A mathematical framework for the analysis of tendon driven limb systems which accounts for muscle redundancy is feasible sets analysis [89]. Feasible sets analysis meshes well with existing understanding of biomechanics and modeling paradigms [87]. In this methodology, each of the muscles in a multi-joint system can be described as having a peak isometric force (F_0) obtained from cadaver experimentation, and an activation with range $[0.0, 1.0]$. Activation is an abstraction from muscle excitation that reflects the true signal from the CNS and instead maps on the accumulation of ions.[7] The set of all possible muscle activations under some set of constraints is defined as the Feasible Activation Space (FAS). While feasible sets analysis has been used in the study of static poses and particularly in the study of simplistic tendon driven limb models such as 3 joint finger models, this dissertation investigates feasible sets over time by considering dynamics and muscle physiology beyond [108]. Designing a framework that allows us to investigate the feasible mechanical outputs and the feasible activations that produce those mechanical outputs over time through the duration of a behavioral task, will allow us to both gain fundamental insights into the neural control of those behavioral tasks and to design metrics that help improve rehabilitation and surgical strategies.

2.4.1 Feasible Activation Space (FAS)

Despite musculoskeletal systems displaying high levels of redundancy, there is a gap in the knowledge as to how or if this redundancy is 'considered' by the CNS when the physiological

[7] This relation is explored more in Chapter 7 beyond the above section regarding Thelen and Winters muscle models.

parameters like the muscle strength or l^m are altered by disorder or disease [20, 91, 99]. Previous studies have shown any particular surgical or therapeutic approach may not improve function in all individuals equally and particular surgical or therapeutic approaches that aid one patient, may harm another [56, 57]. Because the joint configurations can be produced by a wide combination of muscle activations, which can be represented as FAS to achieve the same functional tasks, it is important to develop subject-specific treatment methods. Gait impairments which reduce an individual's gait symmetry or weaken muscles, change the volume of FAS. Patient-specific modeling and simulation along with predictive tools using feedback control have been previously used to design new metrics that aim to improve surgical and rehabilitation outcomes [68]. Ultimately, no studies have laid out a standardized and individualized metric that would quantitatively describe exactly how a doctor should modify the musculoskeletal system to achieve normal function which this paper seeks to lay out the groundwork to perform. Also, there have not been parallels drawn between the feasible neural command space, which is separate from FAS, and the measured EMG, making it difficult to investigate the neural origin of muscle coordination [107, 108, 110].

2.5 Biological Need

This dissertation developed tools using healthy subject data sets; however, there is a clinical need for tools that can provide insights into neuromusculoskeletal disorders at various levels. While these tools and paradigms are tested using data from healthy populations or synthesized *in silico*, all the tools in this dissertation can be applied to the study of various patient populations. In Chapter 6, a procedure using the feasible sets analysis of a subject with instrumented knee is shown: having clinical relevance to the analysis of osteoarthritis (OA). Generally, clinicians use a relatively small number of metrics to determine treatments. Along with being constrained to a small subset of all possible dynamically or kinematically related parameters, clinicians can only ever interpret the quantities that they can directly measure and quantities that may be extremely important for understanding purposeful movement are either impossible to measure or very challenging to measure. Physics-based modeling attempts to fill that gap by offering researchers and clinicians access to parameters

that are necessarily impossible to measure in the laboratory setting such as the joint loading or muscle fiber lengths. Additionally, this dissertation offers researchers interested in neural networks based on CMC or SO output, a platform to magnify the size or variability of their training sets [8].

2.5.1 Gait and Balance Disorders

Every year, over 27,000 people in the US die from falls alone and brain-related injury due to falls has only increased over the past 10 years. By conservative estimates, the CDC detects approximately 35% of older adults suffer from some sort of abnormal gait. There are 47.5 million adults in the U.S. who have a disability and roughly 795,000 people experience a disabling stroke each year. As the aging population grows each year, the number of individuals with gait impairments will only grow, suggesting that there is a great clinical desire for effective treatments. Related to gait disorders, Stroke is the leading cause of long-term disability in the United States [100]. Therefore, it is of the utmost importance to design frameworks that permit the rapid evaluation and rehabilitation of abnormal or asymmetric gait to improve the lives of the impaired or elderly. The origin of muscle coordination whether it be neural or biomechanical, is highly debated so tools that explore the families of activations without applying assumptions about control are valuable. This dissertation offers structures for conceiving effective subject-specific gait retraining metrics and methodologies. Feasible sets analysis rests in parallel to existing biomechanics paradigms like SO, CMC, RRA, muscle synergies analysis and task prioritization.

[8] invaluable in clinical motion analysis where subject data sets are very typically $n < 10$

Chapter 3

A Naive Approach: Pseudo-Static Analysis of End-Effector Forces

3.1 Chapter Background

This chapter focuses on an analysis of force generating capacity at the end effector over the course of a kinematic task. Naive refers to the false premise of a multibody system remaining static on each discrete time of a kinematic task. Occam's Razor tells us to avoid multiplying parameters without necessity: simpler models that still accurately reflect reality are valuable. When researchers model phenomena, they always weigh between computational complexity and accuracy of insights. In this way, electrical engineers can meaningfully use Kirchhoff's Circuit Laws (KVL) to select resistive elements in circuits while KVL is meaningless to physicists who should only be concerned with Maxwell's equations for dealing with flux over vector fields and electromotive forces. Newtonian Mechanics itself is an approximated model of reality. It fails to account for planetary motion offered in Kepler's Laws, and yet no one will force mechanical engineers to solve dynamics problems with quantum mechanics. For slow behavioral tasks like gait or elderly reaching, it is reasonable to assume statics because the centrifugal and Coriolis force contributions will be low. Paradigms that make false premises, such as SO which ignores the muscle parallel elastic element and tendon extensibility, can offer accurate controls approximations for some tasks of some models [117, 119, 121, 122, 124].

Summary of Variables and Terms

For some arbitrary actuated multibody system (MBS):

DOF	degrees of freedom
$\mathbb{R}$	the set of real numbers or sometimes the real field
$\mathbb{R}^{\boldsymbol{n}}$	n-dimensional Euclidean space
$\subset, \in, \mid, \exists, \forall$	subset,element of, such that, there exists, for all
t	time
Δt	discrete time delta
$\boldsymbol{X} \to \boldsymbol{Y}$	Map set $\boldsymbol{X}$ to set $\boldsymbol{Y}$
$x \mapsto f(x)$	Map element x to element $f(x)$
$f : \boldsymbol{X} \to \boldsymbol{Y}, x \mapsto f(x)$	function definition
MTU	muscle tendon unit actuator
m	number of MTU in MBS
c	number of generalized coordinates in MBS
j	number of joints in MBS
b	number of bodies in MBS
$\mathcal{M}$	set of m MTU in MBS
$\mathcal{B}$	set of b bodies in MBS
$\mathcal{J}$	set of j joints in MBS
$\mathbb{C}$	coordinate vector space ($\mathbb{C} \subseteq \mathbb{R}^{\boldsymbol{c}}$)
$\mathbb{O}$	operational vector space ($\mathbb{O} \subseteq \mathbb{R}^{\boldsymbol{6}}$)
$\mathbb{A}$	activation vector space ($\mathbb{A} \subset \mathbb{R}^{\boldsymbol{m}} \mid \mathbb{A} = [0,1]^m$)
$\boldsymbol{\mathcal{Q}}$	vector of c generalized coordinates of MBS, ($\boldsymbol{\mathcal{Q}} \in \mathbb{C}$)
$\boldsymbol{\mathcal{X}}$	vector of 6 parameters defining positions and orientations, ($\boldsymbol{\mathcal{X}} \in \mathbb{O}$)

$\boldsymbol{\mathcal{A}}$	vector of m MTU activations, ($\boldsymbol{\mathcal{A}} \in \mathbb{A}$)
$\boldsymbol{F}$	vector of m MTU forces, ($\boldsymbol{F} \in \mathbb{R}^{\boldsymbol{m}}$)
$\boldsymbol{l_m}$	vector of m MTU fiber lengths, ($\boldsymbol{l_m} \in \mathbb{R}^{\boldsymbol{m}}$)
$\boldsymbol{v_m}$	vector of m MTU fiber velocities, ($\boldsymbol{v_m} \in \mathbb{R}^{\boldsymbol{m}}$)
$\boldsymbol{l}$	vector of m MTU total lengths, ($\boldsymbol{l} \in \mathbb{R}^{\boldsymbol{m}}$)
$\boldsymbol{R_{c\times m}}$	muscle moment arms matrix of MBS
$\boldsymbol{J_{6\times c}}$	kinematic Jacobian matrix of MBS
$\boldsymbol{J^{+}}$	Moore-Penrose pseudoinverse of $\boldsymbol{J}$
$\boldsymbol{J^{-T}}$	Inverse Transpose of $\boldsymbol{J}$
$\dot{\boldsymbol{X}}$	first time derivatives of $\boldsymbol{X}, \dot{}: \boldsymbol{X} \to \dot{\boldsymbol{X}}, x \mapsto \dot{x} = \frac{\partial x}{\partial t}$
$\boldsymbol{\Gamma}$	set of c joint moments of MBS, ($\boldsymbol{\Gamma} \in \mathbb{C} \mid \forall c_i \in \boldsymbol{\mathcal{C}}\ \exists\, \Gamma \in \boldsymbol{\Gamma}$)
$\boldsymbol{\Gamma}_{c\times m}$	moments matrix of MBS
$\boldsymbol{F}$	vector of generalized forces, $\boldsymbol{F} \in \mathbb{O} \mid \boldsymbol{F} = (\boldsymbol{v}, \boldsymbol{\omega}) = (F_x, F_y, F_z, M_x, M_y, M_z)$
$\boldsymbol{F}_{6\times m}$	endpoint force matrix of MBS
$\oplus$	Minkowski Sum
$\mathbb{F}$	endpoint force space, $\oplus\boldsymbol{F}$

Sets are abstract collections of concrete or abstract objects called elements. Vectors are elements of a vector space, and a vector space is just a set that holds vectors. Vector spaces are special sets where vector addition and scalar multiplication are defined. [1] In Euclidean spaces, vectors are elements with magnitudes and directions that map to tuples of parameters in Cartesian space. Similarly, a matrix is a tuple of tuples. This research features rigorous formulations dealing with sets, vectors, vector spaces, and matrices.

What does static force generating capacity along a kinematic task even represent? Forces are interactions between a body and its environment or between two bodies that induces a change to the motion of said body. Without contact, there is no force, unless, of course, considering influences of gravity, electromagnetism or nuclear forces. The concept of an end

[1] Technically, vector spaces are defined over things called fields which are just sets where $+, -, \times, \div$ are defined and act as if using these operators on real numbers. There won't be any proofs dealing with fields in this research, but the comprehension can be useful to understand how sets relate to spaces, vectors and matrices.

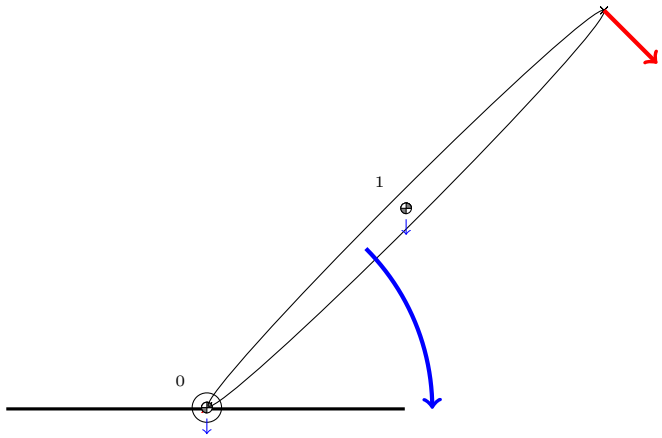


Figure 3.1: Applied joint moment (blue) can be mapped to the end effector force (red) using the kinematic Jacobian, J. In the static domain, this output force is literally the resultant force due thanks to the propagation of moment along the end effector. In a dynamic domain, this vector represents an operational space command $\vec{F}$ (red) that reflects on the acceleration of end effector in operational space.

effector force without contact, reflects more closely to the robotics concepts of $\mathbb{O}$ command vector $\vec{F}$ than generalized forces that actually exist in $\mathbb{O}$ like contact forces (Figure 3.1). It maps to the vector of end effector accelerations $\ddot{\boldsymbol{\mathcal{X}}}$ in $\mathbb{O}$ while ignoring centrifugal and Coriolis contributions to that force. $\mathbb{O}$ commands $\boldsymbol{F}$ at the end effectors of kinematic chains of MBS in $\mathbb{O}$ map to sets of moment commands $\boldsymbol{\Gamma}$ in coordinate space $\mathbb{C}$, and map to sets of muscle activation commands $\boldsymbol{\mathcal{A}}$ in activation space $\mathbb{A}$.

Sets of $\boldsymbol{F}$ map to sets of end effector $\ddot{\boldsymbol{\mathcal{X}}}$ based on the current configuration of the generalized coordinates $\boldsymbol{\mathcal{Q}}$. The possible forces that the end effectors may apply and the ways that an organism may accelerate its end effectors are directly related to the capacity of biological systems and floating base robots to position the center of mass (COM) and perform complex behaviors like balance control. To produce changes to virtual generalized coordinates that describe the relative location of the MBS in $\mathbb{O}$, MBS must use their end effectors to move their base node or COM or otherwise be influenced by external loads $\boldsymbol{F_{ext}}$.

$\boldsymbol{F}$ is a vector-valued function of the model pose $\boldsymbol{\mathcal{Q}}$, and the model actuator states $\boldsymbol{\mathcal{A}}, \boldsymbol{l_m}$, and $\boldsymbol{v_m}$. These vectors can each be constructed from the muscle set by the functional:

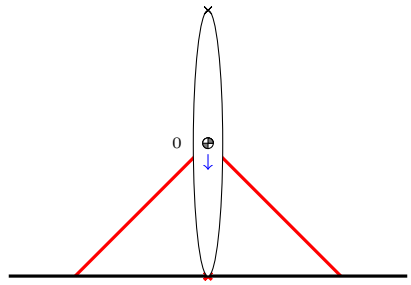


Figure 3.2: A simple 1 DOF model with 2 muscle-tendon actuators, highlighting the muscle redundancy problem.

$$f : \mathcal{M} \rightarrow z, m \mapsto z_m \mid \forall z \in \boldsymbol{\mathcal{Z}} = \{\boldsymbol{l_m}, \boldsymbol{v_m}, \boldsymbol{\mathcal{A}}, \text{or any other muscle parameters...}\} \tag{3.1}$$

Similarly, vectors of the generalized coordinates can be constructed from the set of coordinates (Equation 3.1). Remember from Chapter 2 that the product of forces and distances are a moment known as Γ that does work over an angular distance about a joint. There exists a relationship that maps the moments about model joints assuming a static pose to the force at the end of a kinematic chain for MBS.

For MBS with a set of generalized coordinates $\boldsymbol{\mathcal{Q}}$ and a set of MTU $\mathcal{M}$, consider the simplest non-trivial case (Figure 3.2). Here, one joint is actuated by 2 MTU. Each muscle contribution to static or dynamically consistent generalized coordinate force $\boldsymbol{\Gamma}$ cannot be known without a parameter that is extremely difficult or impossible to measure without invasive instrumentation like the muscle fiber stiffness. For simplicity, assume that an MTU's applied force can vary between 0 and the peak isometric force F_m^0, $\forall m \in \mathcal{M} \; \exists \; F_m^0 \in \mathbb{R}$. Expansions of the calculation of muscle force using more complicated models is later explored. Using a gain called the activation a such that

$$a \in \boldsymbol{\mathcal{A}} \in \mathbb{A} \subset \mathbb{R}^{\boldsymbol{m}} \mid \mathbb{A} = [0, 1]^m \tag{3.2}$$

Equation 3.2 for muscle force can be expressed as Equation 3.3:

$$F : \boldsymbol{\mathcal{A}} \rightarrow \boldsymbol{F},\ a_m \mapsto F(a_m) \mid [F(a_m) = F_m^0 a_m] \quad [2] \tag{3.3}$$

The relationship between a muscle's force (Equation 3.3) and the applied moment about joints is derived from the definition of moments: applied force times a lever arm about said joint and this relationship (Equation 3.4).

$$(\forall c \in \mathcal{C})(\forall m \in \mathcal{M})\boldsymbol{\Gamma}_{c,m} = \boldsymbol{R}_{c,m}\boldsymbol{F}_m\boldsymbol{\mathcal{A}}_m \in \mathbb{R} \tag{3.4}$$

For a system of c generalized coordinates and m muscles, matrices $\boldsymbol{R_{c\times m}}$ can be constructed that maps $\boldsymbol{F}$ to the moment contribution of each muscle about each coordinate using the moment arm of each muscle about each coordinate. For convience, Equation 3.4 can be evaluated for each muscle in a MBS (Equation 3.5):

$$\boldsymbol{\Gamma}_{c\times m} = \boldsymbol{R_{c\times m}} \odot \boldsymbol{F} \tag{3.5}$$

The operator $\odot$ is an elementwise multiplication with $\boldsymbol{F}$ on each row of $\boldsymbol{R}$, $\boldsymbol{F}_{n,m} = \boldsymbol{R}_{n,m}\boldsymbol{F}_m$. It will be prudent to note that the moment arms matrix $\boldsymbol{R}_{n\times m}$ can be considered a linear operator that projects $\boldsymbol{F}$ from $\mathbb{R}^{\boldsymbol{m}}$ into $\mathbb{C}$ as $\boldsymbol{\Gamma}$.

Projections are operators such that $f : \boldsymbol{X} \rightarrow \boldsymbol{Y}, x \mapsto f(x)$ and f may map one or more elements of $\boldsymbol{X}$ to the same element of $\boldsymbol{Y}$ (Figure 3.3).

Additionally, for more complicated models of muscle force, $\boldsymbol{F}$ maps muscle controls $\boldsymbol{\mathcal{A}}$ and muscle fiber $\boldsymbol{l_m}$, $\boldsymbol{v_m}$ and tendon properties) to muscle forces contribution. $\boldsymbol{R_{c\times m}}$ reflects the geometric relationship mapping the change in system configuration, $\dot{\boldsymbol{\mathcal{Q}}}$, to the change in the total length of the MTU, $\dot{\boldsymbol{l}}$ (Equation 3.6).

$$\dot{\boldsymbol{l}} = \boldsymbol{R_{c\times m}}\dot{\boldsymbol{\mathcal{Q}}} \tag{3.6}$$

In figure 3.1, the system Jacobian $\boldsymbol{J_{6nt\times c}}$ of a kinematic chain was described as an operator that maps between the time derivatives of the generalized coordinates $\dot{\boldsymbol{\mathcal{Q}}}$ and the time

[2] This equation is a simplified expression for muscle force.

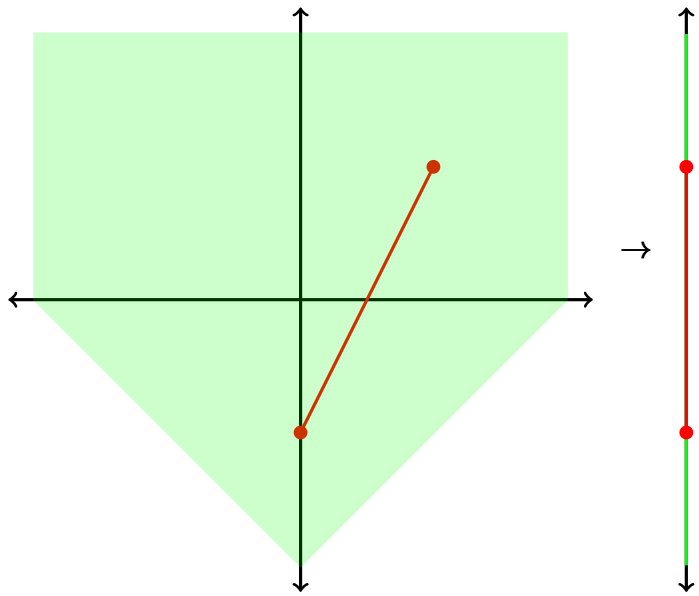


Figure 3.3: The projection of a convex set onto some of its dimensions is also convex. This is the simple parallel projection, but there are many kinds of projection operators. Skew, scale, translate, rotate, orthographic, perspective. These geometric concepts form the basis of computer graphics.

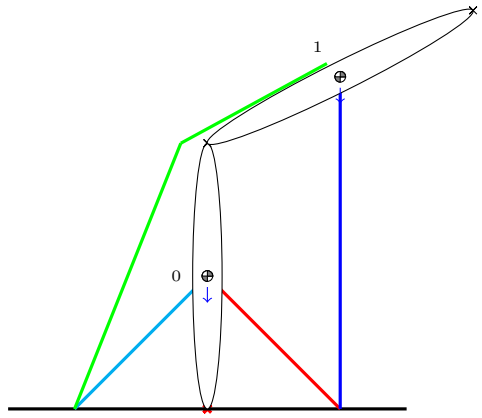


Figure 3.4: A simple 2 DOF model with 4 muscle-tendon actuators for exploring the feasible map problem.

derivatives of the $\mathbb{O}$ parameters $\dot{\boldsymbol{\mathcal{X}}}$. This relationship is derived (Equation 3.7). For a set of $\mathbb{O}$ speeds $\dot{\boldsymbol{\mathcal{X}}}$ and a set of generalized speeds $\dot{\boldsymbol{\mathcal{Q}}}$:

$$
\begin{aligned}
J &: \dot{\boldsymbol{\mathcal{Q}}} \rightarrow \dot{\boldsymbol{\mathcal{X}}} \\
J(\boldsymbol{\mathcal{Q}}) &= \frac{\partial \boldsymbol{\mathcal{X}}}{\partial \boldsymbol{\mathcal{Q}}} \\
J(\boldsymbol{\mathcal{Q}}) = \frac{\frac{\partial \boldsymbol{\mathcal{X}}}{\partial t}}{\frac{\partial \boldsymbol{\mathcal{Q}}}{\partial t}} &\rightarrow \frac{\partial \boldsymbol{\mathcal{X}}}{\partial t} = J(\boldsymbol{\mathcal{Q}}) \frac{\partial \boldsymbol{\mathcal{Q}}}{\partial t} \\
\dot{\boldsymbol{\mathcal{X}}} &= J(\boldsymbol{\mathcal{Q}}) \dot{\boldsymbol{\mathcal{Q}}}
\end{aligned}
\tag{3.7}
$$

$\dot{\boldsymbol{\mathcal{X}}}$ parameters live in $\mathbb{O}$: reflecting the 3 parameters of translation and 3 of orientation, or similarly 6 possible force and moment, in $\mathbb{O}$.

In this 2-link planar manipulator with 4 muscles (Figure 3.4):

It is possible to construct the sets of muscle contributions to static force (Equation 3.8).

$$\boldsymbol{\mathcal{A}} = \{a_1, \quad a_2, \quad a_3, \quad a_4\}$$

$$\boldsymbol{F} = F(\boldsymbol{\mathcal{A}}) = \{F(a_1), \quad F(a_2), \quad F(a_3), \quad F(a_4)\}$$

$$\boldsymbol{R_{2\times4}} = \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} & R_{1,4} \\ R_{2,1} & R_{2,2} & R_{2,3} & R_{2,4} \end{bmatrix}$$

$$\boldsymbol{\Gamma_{2\times4}} = \begin{bmatrix} \Gamma_{1,1} & \Gamma_{1,2} & \Gamma_{1,3} & \Gamma_{1,4} \\ \Gamma_{2,1} & \Gamma_{2,2} & \Gamma_{2,3} & \Gamma_{2,4} \end{bmatrix} = \boldsymbol{R_{2\times4}} \odot \boldsymbol{F} =$$
$$\begin{bmatrix} F(a_1)R_{1,1} & F(a_2)R_{1,2} & F(a_3)R_{1,3} & F(a_4)R_{1,4} \\ F(a_1)R_{2,1} & F(a_2)R_{2,2} & F(a_3)R_{2,3} & F(a_4)R_{2,4} \end{bmatrix}$$

$$\boldsymbol{J_{6\times2}} = \begin{bmatrix} J_{1,1} & J_{1,2} \\ J_{2,1} & J_{2,2} \\ J_{3,1} & J_{3,2} \\ J_{4,1} & J_{4,2} \\ J_{5,1} & J_{5,2} \\ J_{6,1} & J_{6,2} \end{bmatrix} = \begin{bmatrix} J_{x,1} & J_{x,2} \\ J_{y,1} & J_{y,2} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \tag{3.8}$$

To map from joint moments to end effector forces, take the inverse transpose of the Jacobian, $\boldsymbol{J^{-T}}$ to satisfy the expression $\boldsymbol{F_{6\times m}} = \boldsymbol{J^{-T}\Gamma_{c\times m}}$ [69]. Because $\boldsymbol{J_{6\times c}}$, in this example, is not square, there is no unique inverse of $\boldsymbol{J_{6\times c}}$ that satisfies $\boldsymbol{I_{c\times c}} = \boldsymbol{J^{-1}} \times \boldsymbol{J_{6\times c}}$. For a given trajectory of the end effector in $\mathbb{O}$, oftentimes the locations of joints are under-determined, a problem known as kinematic redundancy. To alleviate these issues, a generalization of

the matrix inverse operation known as the Moore-Penrose pseudo-inverse of $\boldsymbol{J}$, $\boldsymbol{J^+}$, which is the minimum least-squares pseudo-inverse of $\boldsymbol{J}$ such that $\boldsymbol{J^+ = (J^T J)^{-1} J^T}$, is used. There are actually many approaches to deal with the under-determined nature of the map from angular velocities to endpoint linear velocities that roboticists use when designing closed form solutions or performing forward kinematics, which include techniques like damping, Jacobian transpose, and Tikhonov regularization. In robotics, the idea of controlling in the under-determined location is known as controlling in the null space. Some biomechanics researchers believe all movement disorders in biological systems manifest in this nullspace, of course, sidestepping questions of if the nervous system thinks about controlling the system in this way. In Chapter 4, the dynamically consistent inverse of $\boldsymbol{J}$, $\boldsymbol{\bar{J}}$ substituting the inertia matrix $\boldsymbol{M^{-1}_{b\times b}}$ for the nullspace is utilized to gain insights into dynamically consistent downstream parameters in the dynamic as opposed to static domain (Equation 3.9).

$$\boldsymbol{F_{6\times 4}} = \boldsymbol{J^{-T}_{6\times 2}\Gamma_{2\times 4}} \tag{3.9}$$

The columns of $\boldsymbol{F_{6\times m}}$ are each muscle's maximal contribution to generalized forces at the end effector.

Instead of the simplistic map F in the previous example, generally, muscle-tendon actuators in modeling platforms such as OpenSim use models that involve passive and active elements of muscles explored in Chapter 2: muscle fiber length $\boldsymbol{l_m}$, muscle fiber velocity $\boldsymbol{v_m}$, and the muscle activations $\boldsymbol{\mathcal{A}}$, pennation angle $\boldsymbol{\alpha}$ and tendon properties.

To circle back to discussions on the trade-offs between modeling complexity and accuracy, a case study is designed to explore the significance of parameters $\boldsymbol{l_m}$ or $\boldsymbol{v_m}$ to the end effector force during human gait and 3 possible models are derived. To avoid nonlinear operators, assume that the tendon is inextensible (isovelocity/ no velocity) for each discrete time frame of a kinematic task:

1. Peaks isometric model considers only $\boldsymbol{F^0}$
2. Fiber length model considers $\boldsymbol{l_m}$ and $\boldsymbol{F^0}$

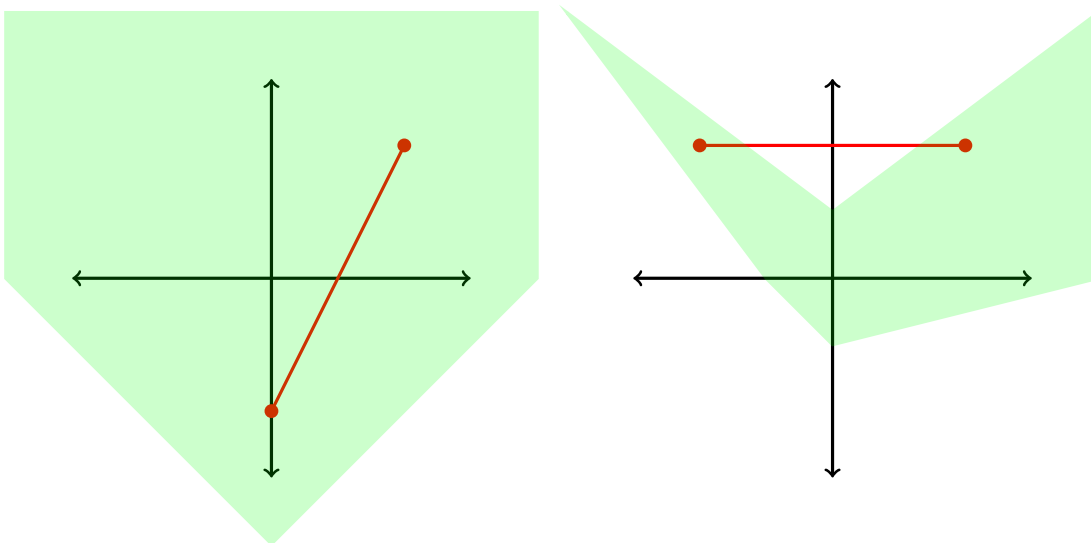


Figure 3.5: A convex set in $\mathbb{R}^2$ (left) and nonconvex set in $\mathbb{R}^2$ (right).

3. Fiber velocity model considers $\boldsymbol{l_m}$, $\boldsymbol{v_m}$, and $\boldsymbol{F^0}$ [3]

Using each of these three models, assuming a static configuration, one may map $\mathbb{A} \to \mathbb{O}$. The activations hypercube or $\mathbb{A} \subset \mathbb{R}^m$ forms a connected and convex $\mathbb{R}^m$ subspace such that $\mathbb{A} = [0, 1]^m \subset \mathbb{R}^m$ (Figure 3.5). $\mathbb{A}$ is convex which implies every point along a line between any two interior points is also an interior point of $\mathbb{A}$. In $\mathbb{R}^2$, this hypercube is a unit square, and in $\mathbb{R}^3$, this hypercube is a unit cube, and so on for higher dimensions.

The nice thing about convex sets is projections of convex sets into only some of their dimensions will also always be convex sets; even multi-stepped maps like $\mathbb{A} \to \mathbb{R}^m \to \mathbb{C} \to \mathbb{O}$. Additionally, even the projection of a convex set to a higher dimension remains convex: this property is how homogeneous coordinates offered by Möbius can be used to translate an image in $\mathbb{R}^m$.

The inverse problem, backing out the possible activations from dynamic $\mathbb{O}$ or $\mathbb{C}$ constraints, is explored in Chapters 5 and 6. This relationship of using the projection

[3] Expressions of inextensible fiber force also involves scalars for passive contribution and is included in the force calculation during procedures like OpenSim's CMC and SO. While it is possible to separately obtain the $\boldsymbol{l_m}$ and $\boldsymbol{v_m}$ multipliers as individual scalars during runtime in an ad hoc simulation using OpenSim and the IK results, l_m and v_m are really coupled by definition. How the muscle moves on the F-V curve depends on the starting and ending positions in the F-L curve, so it is not appropriate to make a model that considers v_m, but ignores l_m. To understand, see the F-L-V surface from Chapter 2 again.

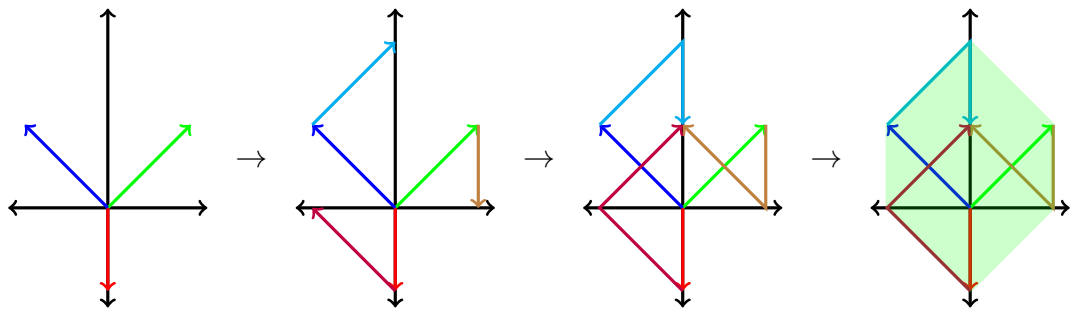


Figure 3.6: Minkowski Sum of a set of 3 generator vectors (green). The boundaries of $\bigoplus \boldsymbol{F}$ can be found by sorting the vectors about their polar angle from the origin, positive y-axis, and summing them in order. View the appendix for giftwrapping and minkowski sum algorithms.

operators that map from controls to downstream parameters will become more relevant in Chapter 4.

If $\mathbb{A}$ is convex, then the superset of all possible $\boldsymbol{F}$ mapped from elements of $\mathbb{A}$, $\mathbb{F} \subseteq \mathbb{O}$, is also convex where some vertices or extreme points of $\mathbb{A}$ map to vertices of $\mathbb{F}$. One may project $\boldsymbol{\mathcal{A}}$ to only the forces or only the moments at the end effector that propagate from the applied moments and this set will also still be convex. For an $\mathbb{A}$, the lower bound of $\mathbb{F}$ can be $\boldsymbol{F_{6\times m}}^{lb} = \boldsymbol{J^{-T}R_{c\times m}} \odot F(\{0,\ldots,0\})_{1\times m}$ and the upper bound can be $\boldsymbol{F_{6\times m}}^{ub} = \boldsymbol{J^{-T}_{6\times c}R_{c\times m}} \odot F(\{1,\ldots,1\})_{1\times m}$. Columns of $\boldsymbol{F_{ub}}$ are known as generators and reflect the maximal contribution of each muscle to the end effector force. The generators can be used to reconstruct the vertices of $\boldsymbol{F}$ using an operation called Minkowski Sum ($\bigoplus$). $\bigoplus : \boldsymbol{\mathcal{A}} \bigoplus \boldsymbol{B} = \{a+b \mid a \in \boldsymbol{\mathcal{A}}, b \in \boldsymbol{B}\}$. Matrix operations relating $\boldsymbol{\mathcal{A}} \rightarrow \boldsymbol{F}$ can be used to construct generators of $\mathbb{F}$ from generators of $\mathbb{A}$ which are then used to reconstruct boundaries of $\mathbb{F}$ by $\bigoplus \mathbb{F}$. The vertices of $\mathbb{F}$ can be found using algorithms known as convex hull methods. There are many algorithms for computing the Minkowski Sum (Figure 3.6) and subsequent convex hull some of which are highlighted in the appendix (Algorithms 4, 5).

3.2 Study: Pseudostatic Analysis

A case study is presented that explores defining $\mathbb{A}$, constructing the map $\boldsymbol{\mathcal{A}} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{\Gamma} \rightarrow \boldsymbol{F}$, and finding $\bigoplus \mathbb{F}$ for each Δt of a discrete domain kinematic trajectory of the gait cycle in post hoc analysis with OpenSim. This case study is a sensitivity analysis to see how $\boldsymbol{l_m}$, $\boldsymbol{v_m}$, and $\boldsymbol{\mathcal{Q}}$ for subjects walking at each of 4 self-selected walking speeds while assuming inextensible muscle tendons, influences different features that can be extracted from $\bigoplus \mathbb{F}$, namely capacity to propel and capacity to brake: critical aspects of human gait. Additionally, this study serves as an exploration of the mechanical advantage offered by interactions of the posture and muscle parameters.

3.2.1 Case Study Motivation

Humans move in their environment by actuating muscles whose forces transmit along kinematic chains to generate reactions against the ground, and in turn, propel their COM. Gait is achieved by modulating the force output of the legs using muscles and orienting the COM; however, the mapping from muscle forces to end point forces necessary to achieve gait is a complex process, involving many nonlinear components. Understanding this relationship can prove useful in the design of control systems for gait and clinical gait analysis. This pseudostatic analysis of force generating capability of the human legs over the course of the gait cycle identified significant differences in braking and propulsive capacity in different phases of gait. For a group of 7 subjects each walking at 4 self-selected walking speeds, significant differences in the capacity to generate forces, accelerate the end effectors, and in turn accelerate the COM in different parts of the gait cycle. Here, the contributions to the ranges of force generating capacity of the limbs in the anterior-posterior directions due muscle physiological parameters over the gait cycle was examined. No significant effects of self selected gait speed on endpoint force space ($\mathbb{F}$). Significant effects of muscle fiber velocity ($\boldsymbol{v_m}$) on $\mathbb{F}$ were found.

The movement of biological systems begins with the mapping of desired controls from the motor cortex which deal chiefly with purposeful movement in the CNS and propagates action potentials along the nerves of the SNS. The electrical signal is then transformed

into a chemical signal at the neuromuscular junction which in turn produces purposeful movement by way of muscle activations. The relationship between the muscle activation and the neurophysiology's excitation is well understood, however the commands from the motor cortex itself and how the motor cortex works in conjunction with the other parts of the nervous system and the physiology itself is not yet understood. This "black box" problem of the controls that produce movement is confounded by the nature of the multi-level redundancy of musculoskeletal system. Biological systems typically use many muscle tendon (MTU) actuators that apply tension to bodies about the same joint or across multiple joints.

By contrast, it can be ideal to control robots by manipulating each prismatic or revolute joint using a single linear actuator or motor as it reduces complexity of control [72]. The actuator redundancy of biological systems proves a complex problem in the analysis of the nervous system's role in the control of complex behavioral tasks. Additionally, biological systems usually have many more joints than degrees of freedom (DOF). This coupled dynamic and kinematic redundancy imposes a complex problem in the analysis of human movement and control of humanoid systems [92, 93, 94]. For specific tasks or motions, there are several sets of feasible controls that can achieve said motion: activations $\boldsymbol{\mathcal{A}}$, muscle fiber lengths $\boldsymbol{l_m}$, and muscle fiber velocities $\boldsymbol{v_m}$. To account for the under-determined problem, researchers use tools like optimization frameworks or make presuppositions about how the motor cortex controls the body. Because of how the movement of limbs subsequently induces changes in the inertial and gravitational forces, it subsequently affects the neural command to adjust these movements, and it affects the length and velocity of each muscle-tendon, the direction of muscle forces and moment arms, and the resulting dynamics of the multi-body system. However, as no two individuals may necessarily utilize the same control scheme even for the same trajectory in $\mathbb{O}$, minimization techniques such as energy reduction or power maximization are not necessarily or solely what the nervous system uses for control [106]. Researchers are opting instead to investigate solution spaces in their entirety [93]. The nature of the influence of specific muscles to support and progression [74, 73, 124] as well as the influence of postural changes to a limb's force generating capacity are well observed [44, 55, 62].

This work proposes an application of a computational framework and tools that can be used in the three-dimensional analysis of gait by describing a lower limb's mechanical advantage in terms of output forces assuming a static pose over gait progression. To quantify limb capability and to arrive at feasible sets of end effector forces in this framework, the set containing all possible combinations of activations of muscles-tendon actuators (MTU), $\mathbb{A}$ is mapped to a set of all possible end effector forces $\boldsymbol{F} \in \mathbb{O}$ under static assumptions for a specific model pose [109]. This study was motivated by other studies that did not investigate the influences of muscle physiological parameters such as the normalized fiber length ($\boldsymbol{l_m}$), fiber velocity ($\boldsymbol{v_m}$) and the postural changes introduced by modifying the gait speed in investigating neural control through optimization frameworks [87, 89]. Feasible end effector forces through a novel pseudo-static analysis that incorporated aspects of robotics, computational geometry, and musculoskeletal modeling are explored.

The set of feasible end effector forces can be denoted as a space ($\mathbb{F}$) which bounds the possible muscle contributions to the end effector force. Bounds on $\mathbb{F}$ considering muscle physiology for complex 3D musculoskeletal gait models were found. It was hypothesized that there was a synergistic relationship between gait speed and increasing muscle model complexity.

3.2.2 Methods

Experimental gait analysis data available on SimTK was analyzed from 7 subjects (7-18 yrs.) walking over ground at four self-selected walking speeds: extra slow, slow, free, and fast [63]. A complex 23 DOF 92 MTU gait model along with inverse kinematics (IK) to compute joint kinematics for each gait trial was used.

For each subject at each walking speed, CMC was used to obtain $\boldsymbol{l_m}$ and $\boldsymbol{v_m}$. $\mathbb{F}$ was determined for each frame of motion using the columns of

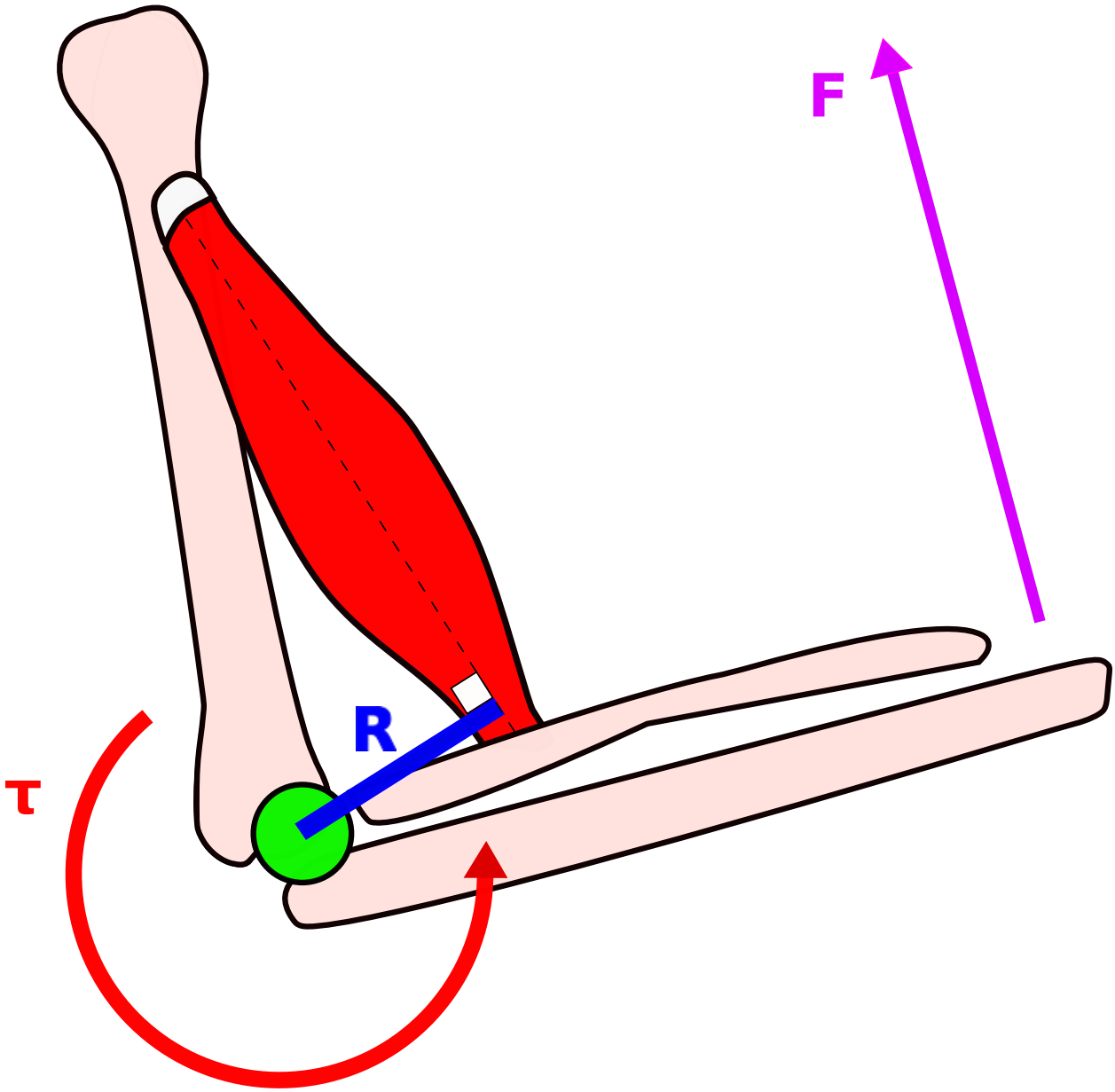


Figure 3.7: Depicting the muscle moment arm $\boldsymbol{R}$ and the relationship to the output force at the end effector $\boldsymbol{F}$.

Figure 3.8: 23 DOF 92 muscle model available in OpenSim. Scaled to reflect subject anthropomentry and using subject-specific kinematic states. $\mathbb{F}$ are highlighted for left and right legs.

$$\boldsymbol{F_{6\times m}} = \begin{bmatrix} M_x^1 & \dots & M_x^m \\ M_y^1 & \dots & M_y^m \\ M_z^1 & \dots & M_z^m \\ F_x^1 & \dots & F_x^m \\ F_y^1 & \dots & F_y^m \\ F_Z^1 & \dots & F_z^m \end{bmatrix} = \boldsymbol{J_{6\times c}^{-T} R_{c\times m}} \odot \boldsymbol{F_{1\times m}} \tag{3.10}$$

Where, for a limb with c generalized coordinates and m muscles, $\boldsymbol{J^{-T}}$ is the least-squares damped inverse transpose Jacobian which maps joint moments to end effector forces and moments. $\boldsymbol{R_{c\times m}}$ (Figure 3.7) is the muscle moment arms matrix which maps muscle forces to joint moments and $\boldsymbol{F}$ is the muscle strength (mapping maximal muscle activations to peak isometric muscle forces). For each model and kinematic data set, a post hoc simulation is performed using the output state of CMC and the subject-specific model. The simulations' models and states, comprising generalized coordinates, generalized velocities and the muscle-tendon actuation fiber lengths and activations, were used to investigate the effects of muscle fiber length $\boldsymbol{l_m}$ and muscle fiber velocity $\boldsymbol{v_m}$ on $\mathbb{F}$ for 3 cases of varying muscle physiology inclusion (Figure 3.8).

All muscles operate on a force-length-velocity surface and the position of a muscle on this surface varies over the gait cycle [5, 6, 7, 45], and to investigate the nonlinear mapping of applied muscle forces to $\mathbb{F}$, scalar mappings from a muscle's peak isometric force to its position on the F-L-V curve can be obtained at run-time using the MATLAB API for OpenSim. Conveniently, the Millard2012 muscle model used in Gait2392 is designed such that the force of a muscle is explicitly a function of the velocity component and the fiber length component as scalars derived from the muscle F-L-V surface [71]:

From Equation 2.13 where the terms $f^l(\boldsymbol{l_m})$ and $f^v(\boldsymbol{v_m})$ reflect the position of the muscle on the force-length and force-velocity curves respectively. The term $f^{PE}(\boldsymbol{l_m})$ is the force of the muscle due to the passive element, which is purely a function of the length of the muscle unit assuming an inextensible tendon.

For no muscle physiology (case 1), $f^l(\boldsymbol{l_m})$ and $f^v(\boldsymbol{v_m})$ terms can be ignored such that the muscle is assumed to be operating exclusively at the peak of the F-L-V surface. At each time,

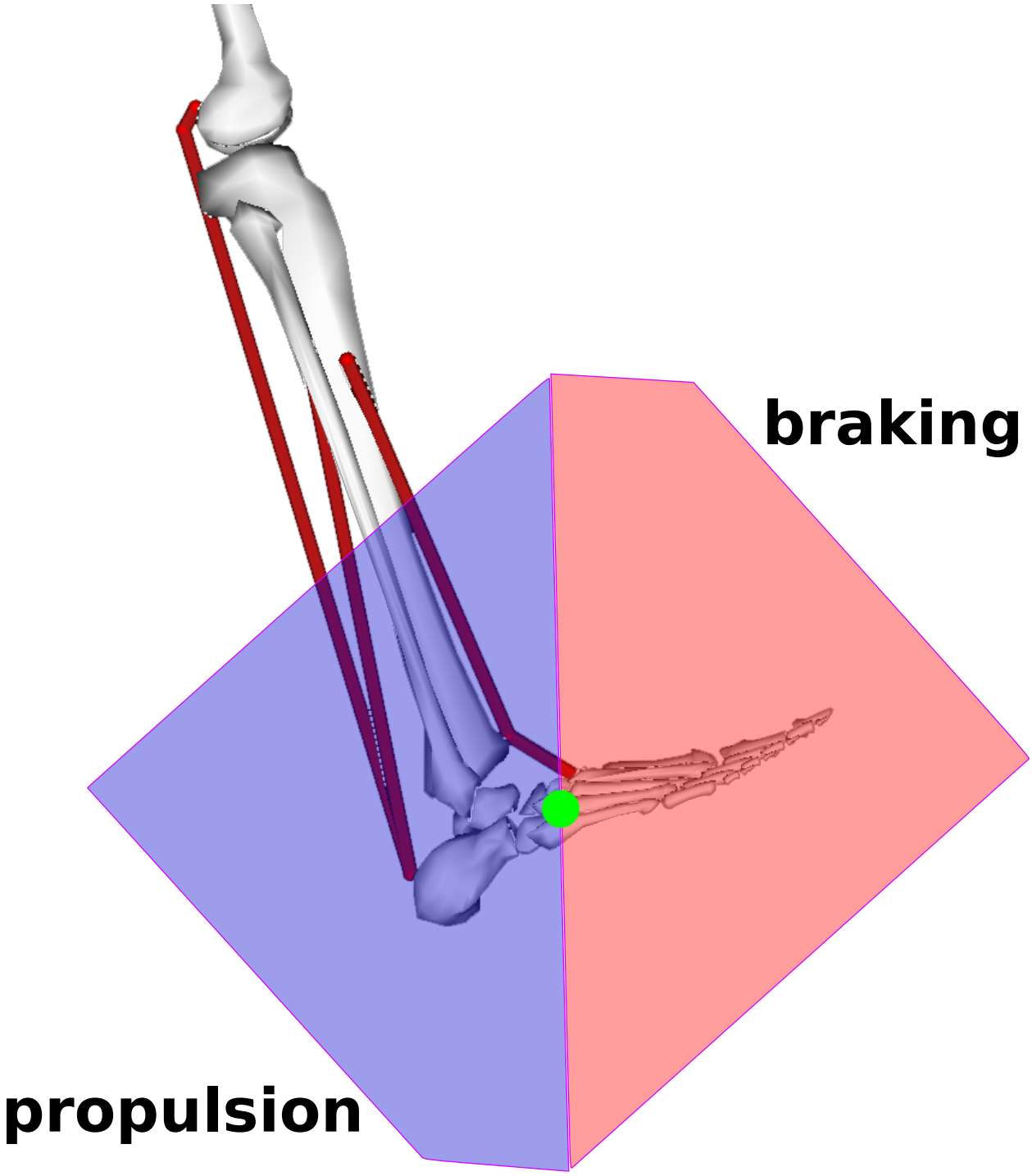


Figure 3.9: Comparing braking and propulsion feasible forces. This convention may not be intuitive, but these sets are the spaces of $\mathbb{O}$ commands in the static domain. Action forces or commands that brake are necessarily anterior to the center of pressure or COM of the calcaneus while propulsion is necessarily posterior.

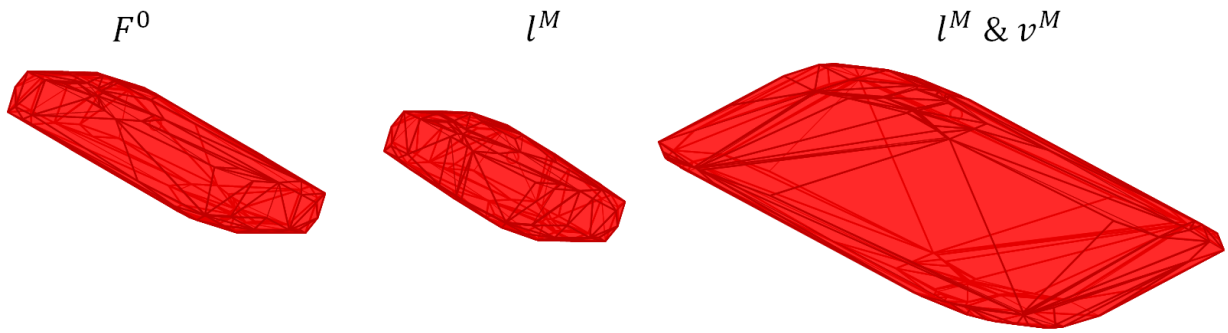


Figure 3.10: differences in $\mathbb{F}$ at a single frame during swing phase, highlighting the influence of $\boldsymbol{l_m}$ and $\boldsymbol{l_m} + \boldsymbol{v_m}$ consideration.

each muscle's peak isometric force is obtained along with $\boldsymbol{R}$ and $\boldsymbol{J^{-T}}$. Muscle force set $\boldsymbol{F}$ is computed as a function of the pennation angle that the muscle fibers form along the tendon and the peak isometric force. The end effector forces (Equation 3.10) due to the applied muscle forces can then be computed as a function of $\boldsymbol{F_0}$ and $\boldsymbol{J^{-T}}$ and $\boldsymbol{R}$. The feasible forces are computed as per algorithm 1. The muscle fiber lengths and muscle moment arms change over the gait cycle. Also, muscle fiber lengths are not purely a function of the geometry of the underlying system, as the muscle models used in the gait model consist of both a muscle and tendon.

For $\boldsymbol{l_m}$ effects (case 2), the model is again updated to reflect CMC states for each percent of gait, but the feasible forces are now computed using $f^l(\boldsymbol{l_m})$. This force-length multiplier unique to the muscle's current state can be found in OpenSim as per algorithm 2. This force-length multiplier reflects the muscles position on the force-length curve and changes with state.

For both $\boldsymbol{l_m}$ and $\boldsymbol{v_m}$ effects (case 3), the model is updated to reflect CMC states, but the feasible force is now computed using both the force-length multiplier and force-velocity multiplier, which reflects the muscle's position on the force-length-velocity surface as per algorithm 3. In all cases, the matrix of muscle moment arms about each joint was computed

and used to compute the joint moments (Equation 3.11):

$$\boldsymbol{\Gamma}_{\mathbf{6\times c}} = \boldsymbol{R}_{\boldsymbol{c\times m}}\boldsymbol{F} \tag{3.11}$$

$\mathbb{F}$ is the vector space which contains all possible end effector forces for a given model pose mapped from all possible $\boldsymbol{\mathcal{A}}$ in $\mathbb{A}$ and is computed by solving a vertex enumeration problem using the Minkowski sums of non-convex sets which is a complex problem [39]. This is very powerful, as the volume or boundaries of $\mathbb{F}$ can be used as a metric of mechanical advantage and quantify if subjects can adequately perform support or propulsion with respect to gait or balance recovery in healthy subjects, but it is a complex function of many inputs.

Solutions assuming static consistency can have practical utility particularly in the analysis of gait [3]. $\mathbb{F}$ is computed using the end effector force contributions of each muscle within the columns of the wrench matrix, $\boldsymbol{F}$ (Equation 3.10). For each volume, the components of $\mathbb{F}$ volume that maps the muscles' capacity to apply braking and propulsive forces (Figure 3.9) as the lower limb's capability to apply braking and propulsive forces are critical to the acceleration of the COM during gait.

Braking (b$\mathbb{F}$) and propulsive (p$\mathbb{F}$) volumes can be extracted from full $\mathbb{F}$ by using a delaunay triangulation slicing algorithm to bifurcate $\mathbb{F}$ into a positive and negative sections along the medial-lateral direction and recomputing the convex hull through QHull or a similar algorithm.

3.2.3 Statistical Analysis

Braking and Propulsive $\mathbb{F}$ can then be compared by using 3 separate repeated measures ANOVA in SPSS through the GLM procedure to verify statistical differences between different levels of muscle physiological inclusion and different levels of self-selected walking speed and to identify the sources of interaction effects.

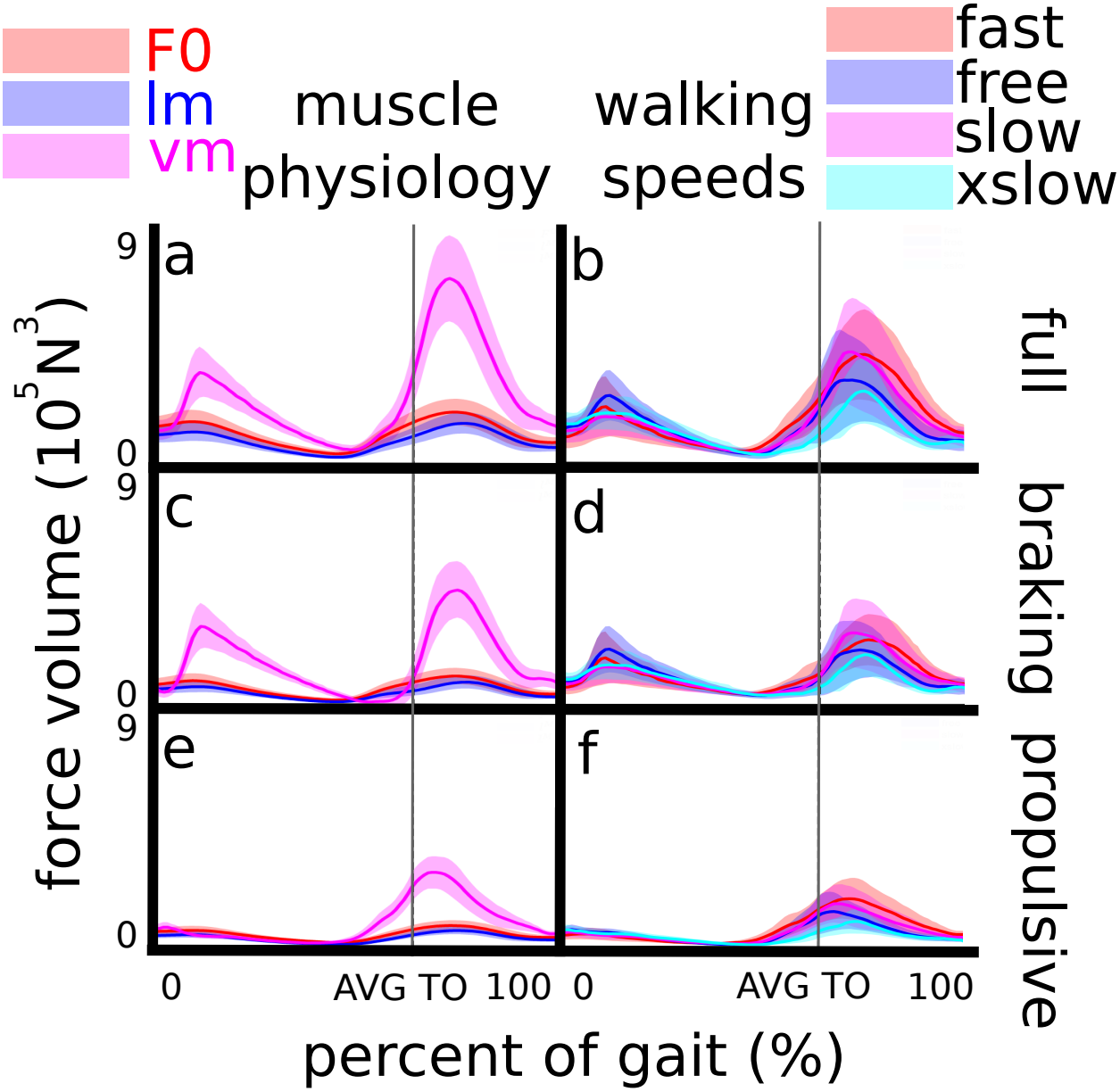


Figure 3.11: Total, propulsive and braking volume as the sum of p$\mathbf{F}$ and b$\mathbf{F}$ over progression of gait. Different muscle physiological considerations and speeds are plotted with different colors. Simple main effects analysis revealed significant differences across considerations in mid stance and in mid swing.

Table 3.1: Full, propulsive, and braking force volumes (N^3)*: The mean difference is significant at the 0.05 level. b: adjustment for multiple comparison: bonferroni.

consid.	consid.	mean diff	std. error	$sig.^b$	lb	ub
Full						
1	2	28814.6*	3776.3	0.001	16400.0	41229.3
	3	-137297.9*	23951.3	0.004	-216036.7	-58559.2
2	1	-28814.7*	3776.3	0.001	-41229.3	-16400.1
	3	-166112.6*	27359.0	0.003	-256054.0	-76171.2
3	1	137297.9*	23951.3	0.004	58559.2	216036.7
	2	166112.6*	27359.0	0.003	76171.2	256054.0
propulsive						
1	2	15136.6*	2136.0	0.001	8114.7	22158.6
	3	-95367.2*	19099.2	0.007	-158154.9	-32579.4
2	1	-15136.6*	2136.0	0.001	-22158.6	-8114.7
	3	-110503.8*	20766.6	0.005	-178773.1	-42234.5
3	1	95367.2*	19099.2	0.007	32579.4	158154.9
	2	110503.8*	20766.6	0.005	42234.5	178773.1
braking						
1	2	13678.0*	1720.2	0.001	8022.8	19333.3
	3	41930.8*	6043.8	0.001	-61799.6	-22061.9
2	1	-13678.0*	1720.2	0.001	-19333.3	-8022.8
	3	-55608.8*	7575.1	0.001	-80511.5	-30706.0
3	1	41930.8*	6043.8	0.001	22061.9	61799.6
	2	55608.8*	7575.1	0.001	30706.1	80511.5

3.2.4 Results

p$\mathbb{F}$ and b$\mathbb{F}$ were extracted from $\mathbb{F}$ over progression of gait for each subject walking at each of the 4 self-selected walking speeds. Significant differences were found for different physiological considerations (Figure 3.10).

Full volumes $\mathbb{F}$

Mauchly's test of sphericity identified that physiological consideration violated assumptions of sphericity ($\chi^2(2) = 22.263, p < .0005$) so a Greenhouse-Geisser correction was applied in subsequent data analyses. Different physiological considerations showed statistically significant (Table 3.1) differences in $\mathbb{F}$ ($F(6.035, 1.006) = 35.373, p = .001$). Also, a significant interaction term between consideration and time was observed ($F(12.250, 2.042) = 19.978, p \leq .0005$). No significant effect of gait speed on $\mathbb{F}$ was observed ($F(18, 3) = 2.007, p = .193$) (Figure 3.11b).

Propulsive volumes p$\mathbb{F}$

Mauchly's test of sphericity identified that physiological consideration violated assumptions of sphericity ($\chi^2(2) = 16.641, p < .0005$) so a Greenhouse-Geisser correction was applied in subsequent data analyses. Different physiological considerations showed statistically significant differences in $\mathbb{F}$ ($F(6.11, 1.018) = 52.004, p \leq .005$). Also, a significant interaction term between consideration and time was observed ($F(10.964, 1.827) = 34.047, p \leq .005$). No significant effect of gait speed on $\mathbb{F}$ was observed ($F(18, 3) = 3.358, p = .042$) (Figure 3.11d).

Braking volumes b$\mathbb{F}$

Mauchly's test of sphericity identified that physiological consideration violated assumptions of sphericity ($\chi^2(2) = 21.630, p < .0005$) so a Greenhouse-Geisser correction was applied in subsequent data analyses. Different physiological considerations showed statistically significant differences in $\mathbb{F}$ ($F(6.11, 1.018) = 26.899, p = .002$). Also, a significant interaction term between consideration and time was observed ($F(11.872, 1.979) = 15.326, p = .001$).

No significant effect of gait speed on $\mathbb{F}$ was observed ($F(18, 3) = 1.322, p = .301$) (Figure 3.11f).

3.2.5 Discussion

The objective of this study was to explore the advantage afforded by muscle physiological and postural differences to progression and braking during healthy gait at different self-selected walking speeds. Several key differences in the capacities of the musculoskeletal system to generate propulsive and braking forces under different physiological considerations were identified. b$\mathbb{F}$ was maximized near midswing and early stance while p$\mathbb{F}$ was maximized just after toe off. Prior research has found that there are also potentially other limiting factors to the feasible end effector forces [52, 55]. $\mathbb{F}$ is a nonlinear function of several components, derived from the musculoskeletal geometry during a specific pose in the gait cycle and the muscle states derived using OpenSim's CMC algorithm.

Postural differences induced by varying gait speed were much less significant than the inclusion of $\boldsymbol{l_m}$ or $\boldsymbol{v_m}$ effects, implying that the differences in postures at varying self-selected speed are much less important than the differences in gravitational, centrifugal and Coriolis contributions at varying gait speed. The multiplicative effect of joint orientation and muscle physiological considerations is a well observed phenomenon in literature [4]. Previously established investigations of muscle physiology and gait speed or posture showed that there were significant influences to downstream parameters like ground reaction forces at different speeds/postures [7, 46, 73, 75, 104] and also prior research has shown postural differences to be significant factors in gait efficiency [76, 61, 114]; however, these postural changes were not as influential in this analysis. Observations of decreased muscle output at faster speeds coincide with prior observations [75].

There are several key limitations in this method of analysis. First, the muscle forces estimated with these dynamic simulations may not accurately reflect the forces generated by individual subjects as some parameters of the model, such as the muscle max isometric forces are not scaled [36, 81]. Second, the $\mathbb{F}$ is only computed assuming a static pose ignoring gravitational forces and explores a space of command vectors capable of maintaining static

equilibrium, while gait itself is a dynamic task. Further investigation should be done to see if faster or slower modes of movement produce significant differences in speed.

Future work should include contributions of inertial and gravitational forces to $\mathbb{F}$ and particularly explore the feasible space of forces capable of exploring support consistent dynamic reaction force space. Also, future work should investigate if faster modes of movement induce significant differences in $\mathbb{F}$. The true superset of $\boldsymbol{\mathcal{A}}$ capable of generating the dynamically consistent subspace in $\mathbb{F}$ is a subspace of $\mathbb{A}$ [87]. Additionally, there should be consideration of the centrifugal and Coriolis contributions to loading [47]. The inclusion of muscle fiber length effects consistently reduced $\mathbb{F}$ over the gait cycle, while the inclusion of fiber length and fiber velocity effects either increased or decreased $\mathbb{F}$ relative to peak isometric depending on the phase of gait.

3.3 Chapter Summary

From this sensitivity analysis of the model pose over progression of gait at different gait speeds, gait speed was not actually significant, but the fiber length and fiber velocity considerations were significant to the contribution of feasible end effector forces over the gait cycle. So, when performing feasible sets analysis in Chapters 4, 5, 6 and 7, fiber length and fiber velocity multipliers will be included in the calculations. Postural differences induced by increasing gait speed did not produce the significant changes to end effector force; however, Chapter 4 will show that bounding activations by the joint moment constraints, then projection operators can be constructed that map the muscle activations to a dynamically consistent muscle-dependent $\mathbb{O}$ parameters, and under the dynamic domain, $\ddot{\boldsymbol{\mathcal{X}}}, \dot{\boldsymbol{\mathcal{X}}}$, and $\dot{\boldsymbol{\mathcal{Q}}}$ do influence the force generating capacity. In Chapter 5, the influence of muscle physiology on the bounds of dynamically consistent bounds of activation space is revisited.

Chapter 4

Dynamical Considerations and Projection Operators

4.1 Chapter Background

The procedure from Chapter 3 is expanded, exploring how to map known subsets of $\mathbb{A}$ to subsets of an arbitrary lower dimensional space for arbitrary MBS. To find supersets of $\boldsymbol{\mathcal{A}}$ that satisfy joint moment requirements determined using ID for a specific kinematic task before exploring the projection from $\mathbb{A}$ to $\mathbb{O}$ using vertex enumeration, view Chapter 5 which produces $\mathbb{A}$ constrained by ID joint moments $\boldsymbol{\Gamma_{task}}$, $\mathbb{A} \mid \boldsymbol{\Gamma_{task}}$. High throughput tools that transform high dimensional $\mathbb{A}$ using a known projection operator and a case study that shows mapping $\mathbb{A} \mid \boldsymbol{\Gamma_{task}}$ to supersets of dynamically consistent $\ddot{\boldsymbol{\mathcal{X}}}$ and $\boldsymbol{F_{task}}$ are explored.

4.1.1 Linear Operators

This research focuses on the maps between muscle controls and downstream parameters, and how observations of downstream parameters constrains muscle controls. Following and extrapolating on the example from Chapter 3, the map from muscle activation to end effector force is defined (Equation 4.1):

$$\boldsymbol{\mathcal{A}} \rightarrow \boldsymbol{F} \rightarrow \boldsymbol{\Gamma} \rightarrow \boldsymbol{F} \tag{4.1}$$

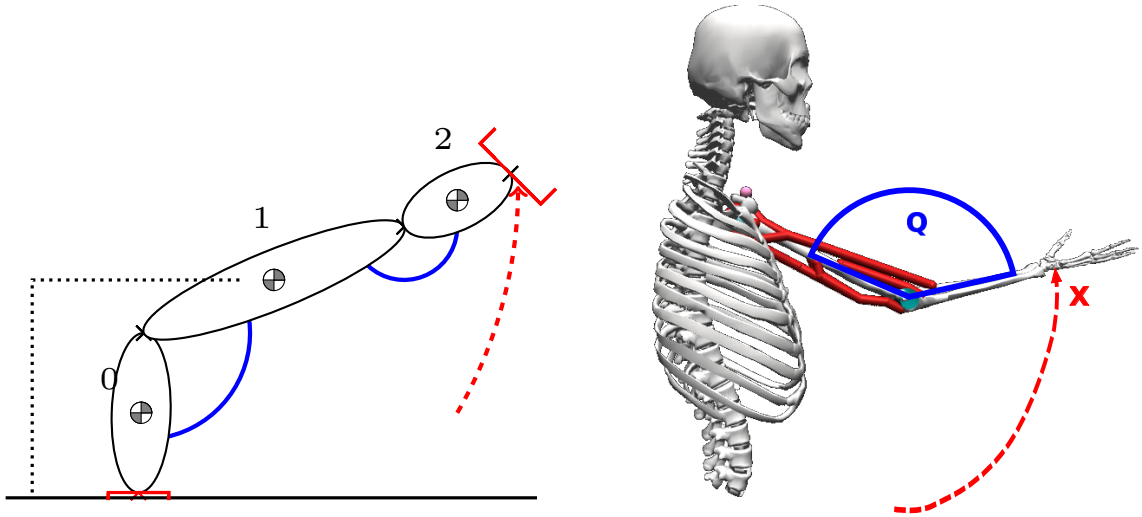


Figure 4.1: $\mathbb{O}$ control for robots or musculoskeletal systems is all about dictating the end effector trajectories $\boldsymbol{\mathcal{X}}$ in $\mathbb{O}$ through command vectors or tasks and manipulating $\boldsymbol{\mathcal{Q}}$ in the null-space of $\mathbb{C}$. Also, for floating base robots and biological systems, it involves how the actuated DOF can be used to induce changes to the virtual DOF.

Where $\boldsymbol{J^{-T}}$ is the projection operator that maps the muscle contributions to joint moments to muscle contributions to end effector forces in the static domain; however, for dynamically consistent values of $\boldsymbol{F}$, $\boldsymbol{J^{-T}}$ will not be sufficient as centrifugal and Coriolis contributions to the task need to be accounted for. To this aim, some tools in $\mathbb{O}$ control (Figure 4.1) and manipulator robotics can be adapted for the analysis of MBS in feasible sets analysis [27, 28, 53, 101].

4.1.2 Manipulator Robotics Application

The relationship between ID and $\mathbb{O}$ formulation is explored. The relevance of $\mathbb{O}$ formulation to the control of musculoskeletal systems is also explored. The muscle force contribution to coordinate generalized forces in $\mathbb{C}$ is computed in Chapter 5.

For robotic manipulators, there are two paradigms for control: either control in $\mathbb{O}$ or $\mathbb{C}$. Systems can be controlled by moments commands or generalized coordinate forces, or they can be controlled by tasks in $\mathbb{O}$ that push end effectors to follow desired kinematic trajectories of the end effector in $\mathbb{O}$. For people who design controls systems, this paradigm

is a control law that maps $\boldsymbol{\mathcal{X}}, \dot{\boldsymbol{\mathcal{X}}}, \ddot{\boldsymbol{\mathcal{X}}}$ to $\boldsymbol{\mathcal{Q}}, \dot{\boldsymbol{\mathcal{Q}}}$. $\mathbb{O}$ formulation from [53] shows that the map between the moment commands $\boldsymbol{\Gamma}$ and the physical constraint in $\mathbb{O}$ is uniquely determined from extrapolations of principles from D'Alembert's Principle or dually Gauss' Principle (Equation 4.2): [1]

$$\Lambda(\boldsymbol{\mathcal{X}})\ddot{\boldsymbol{\mathcal{X}}} + \mu(\boldsymbol{\mathcal{X}}, \dot{\boldsymbol{\mathcal{X}}}) + p(\boldsymbol{\mathcal{X}}) = \boldsymbol{F} \tag{4.2}$$

Where $\boldsymbol{\mathcal{X}} = (x_1 \dots x_n)$ are the independent parameters of the system in $\mathbb{O}$ and $\boldsymbol{\Lambda}$ is the kinetic energy matrix. $\boldsymbol{\mu}$ represents the contributions to the centrifugal and Coriolis forces on bodies of the multibody system [26, 54]. Finally, $\boldsymbol{p}$ is the force due to gravity on bodies and $\boldsymbol{F}$ is the external forces. The muscles of the model must account for the moments besides contributions of the externally applied loads. Equation 4.3 will look familiar to those with experience in ID in OpenSim:

$$M(\boldsymbol{\mathcal{Q}})\ddot{\boldsymbol{\mathcal{Q}}} + C(\boldsymbol{\mathcal{Q}}, \dot{\boldsymbol{\mathcal{Q}}})\dot{\boldsymbol{\mathcal{Q}}} + G(\boldsymbol{\mathcal{Q}}) = \boldsymbol{\Gamma} \tag{4.3}$$

Where $\boldsymbol{\mathcal{Q}}$ is the vector of generalized coordinates of the model, $\boldsymbol{M_{b \times b}}$ is the system Mass Matrix, $\boldsymbol{C}$ is the vector of the Coriolis and centrifugal forces, $\boldsymbol{G}$ is the vector of gravitational forces, and $\boldsymbol{\Gamma}$ is the vector of generalized forces in $\mathbb{C}$.

The map between $\boldsymbol{\mathcal{Q}}$ of MBS to $\mathbb{O}$ is the kinematic Jacobian ($\dot{\boldsymbol{\mathcal{X}}} = J(\boldsymbol{\mathcal{Q}})\dot{\boldsymbol{\mathcal{Q}}}$) and, similarly, the map from $\boldsymbol{\Lambda}$ in $\mathbb{O}$ formulation to $\boldsymbol{M}$ in the $\mathbb{C}$ equations of motion can be formulated (Equation 4.4):

$$M(\boldsymbol{\mathcal{Q}}) = J^T(\boldsymbol{\mathcal{Q}})\Lambda(\boldsymbol{\mathcal{X}})J(\boldsymbol{\mathcal{Q}}) \tag{4.4}$$

$\boldsymbol{M}$ is really the kinetic energy weighted pseudo-inverse of the system Jacobian and $\boldsymbol{M}$ is in fact the mass matrix. The difference being $\boldsymbol{M}$ operates for $\mathbb{C}$ and $\boldsymbol{\Lambda}$ operates for $\mathbb{O}$.

The joint space generalized forces can be mapped along a given set of frame tasks to $\mathbb{O}$ (Equation 4.5):

[1] These concepts are natural laws: abstractions of the principle of least action. Minimize the Lagrangian integral to identify the true trajectory of particles from the space of all possible trajectories.

$$\boldsymbol{\Gamma_{task}} = J(\boldsymbol{\mathcal{Q}})^T \boldsymbol{F_{task}} \tag{4.5}$$

For redundant systems, Equation 4.5 is expanded to Equation 4.6,

$$\boldsymbol{F_{task}} = \bar{J}(\boldsymbol{\mathcal{Q}})\boldsymbol{\Gamma_{task}} \tag{4.6}$$

Where $\bar{\boldsymbol{J}}$ (Equation 4.6) is the dynamically consistent Jacobian (Equation 4.7):

$$\bar{\boldsymbol{J}} = \boldsymbol{M^{-1}J^T(J_tM^{-1}J^T)^{-1}} \tag{4.7}$$

Similarly, for a vector of muscle forces $\boldsymbol{F}$,

$$\boldsymbol{\Gamma} = L(\boldsymbol{\mathcal{Q}})^T \boldsymbol{F} \tag{4.8}$$

Where $L(\boldsymbol{\mathcal{Q}})$ (Equation 4.8) is the "muscle Jacobian" [29], which maps the configuration of the model in $\mathbb{C}$ to the total length of the MTU length (Equations 4.9, 4.10).

$$\dot{\boldsymbol{l}} = L(\boldsymbol{\mathcal{Q}})\dot{\boldsymbol{\mathcal{Q}}} \tag{4.9}$$

Specifically, this relation between joint moment and muscle force (Equation 4.9) is the muscle moment arms matrix $\boldsymbol{R_{c\times m}}$ [33, 93] from Chapter 3, so:

$$\dot{\boldsymbol{l}} = R(\boldsymbol{\mathcal{Q}})\dot{\boldsymbol{\mathcal{Q}}} \tag{4.10}$$

These equations can be combined to arrive an equation of feasible accelerations of the task (Equation 4.11):

$$\ddot{\boldsymbol{\mathcal{X}}} = J(\boldsymbol{\mathcal{Q}})M(\boldsymbol{\mathcal{Q}})^{-1}(\boldsymbol{\Gamma} - \boldsymbol{J_{loc}^T F_{ext}}) \tag{4.11}$$

Where $\boldsymbol{J_{loc}}$ would be the Jacobian from base node to the location on body of the externally applied loads. From Chapter 3, there are expressions (Equation 4.12) for mapping the model joint toques and the muscle activation level in conjunction with Equation 4.11:

$$\ddot{\boldsymbol{\mathcal{X}}} = J(\boldsymbol{\mathcal{Q}})M(\boldsymbol{\mathcal{Q}})^{-1}(R(\boldsymbol{\mathcal{Q}}) \odot \boldsymbol{F^0 l_m v_m} \mathbb{A} cos(\boldsymbol{\alpha}) - \boldsymbol{J_{loc}^T F_{ext}}) \tag{4.12}$$

Note, there are maps from sets of feasible activations of the model to sets of dynamically consistent end-effector forces.

Another tool would be to project $\mathbb{A}$ itself into the null-space (Equations 4.13, 4.14) of the task to filter out activations that do not contribute to the task.

$$\boldsymbol{\mathcal{N}_{c\times c}} = (\boldsymbol{I_{c\times c}} - \boldsymbol{\bar{J}_{c\times 6} J_{6\times c}}) \tag{4.13}$$

To construct a projector that maps $\mathbb{A} \rightarrow \mathbb{A}_{\boldsymbol{\mathcal{N}}}$:

$$\boldsymbol{\mathcal{N}_{m\times m}} = \boldsymbol{R^T_{m\times c} \mathcal{N}_{c\times c} R_{c\times m}} \tag{4.14}$$

MTU have both active and passive components. To deal with the dichotomy of active and passive components depending on a selected model of muscle force, homogeneous coordinates are leveraged (Equation 4.15):

1. Construct a projector reflecting the $\boldsymbol{\mathcal{A}}$-dependent components of force
2. Project to a dimension $n+1$
3. Return to n and translate by the $\boldsymbol{\mathcal{A}}$-independent components of force

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P^{a-\text{dependent}}_{n\times m}} & 0 \\ \boldsymbol{P^{a-\text{independent}}_{1\times m}} & 1 \end{bmatrix} \tag{4.15}$$

$\boldsymbol{P^{active}_{1\times m}}$ contains the $\boldsymbol{\mathcal{A}}$-dependent contributions to the downstream parameter decomposed by muscle, and $\boldsymbol{P^{a-\text{independent}}_{1\times m}}$ contains all the $\boldsymbol{\mathcal{A}}$-independent contributions. The idea of lifting into a higher dimension to solve the problem is a tool used by LRS, the vertex enumeration solver used in Chapter 5. What is particularly challenging about mapping from activation vectors to some downstream parameter for which the projection operator can be constructed? For a robust gait model of 25 muscles on each leg, $\mathbb{A} \mid$ *some constraint* could potentially have hundreds upon hundreds of thousands of vertices in $\mathbb{R}^{92}$, and so for any chance of clinical

relevance or to navigate in $\mathbb{A}$ as part of a control paradigm, there needs to be implementations of rigorous methods for performing projection of vector spaces that deal with data sets on the orders of GB or even TB in space.

For generalization to arbitrary linear projections for some tasks, command line tools were developed for mapping every set of activation in a supplied file containing the activation sets using an input projection operator. The code developed in this dissertation uses OpenMPI and MPLRS, and allows the end user to run high dimensional problems with supercomputers and clusters such as Titan available in Oak Ridge National Laboratory.

4.2 Study: Feasible Accelerations

4.2.1 Case Study Motivation

A case study is offered that constructs $\mathbb{A} \mid \boldsymbol{\Gamma_{task}}$ and maps to sets of $\ddot{\boldsymbol{\mathcal{X}}}$ for two models freely available with OpenSim for testing, tutorials, and proof-of-concepts. The static domain tools from Chapter 3 are extrapolated to an induced acceleration analysis produced using methods from Chapter 5. For a 2 DOF 6 muscle upper-extremity model, $\mathbb{A} \mid \boldsymbol{\Gamma_{task}}$ is mapped to muscle contributions to $\mathbb{O}$ accelerations. Also, for a 10 DOF 18 muscle gait,$\mathbb{A} \mid \boldsymbol{\Gamma_{task}}$ is mapped back to joint moments to highlight parity with the ID solution.

4.2.2 Methods

The Arm26 model (Figure 4.2) freely available with OpenSim is used to map feasible activations to induced accelerations over the course of a simple elbow flexion task (Figure 4.3). CMC determined muscle excitations and resulting model states for a forward dynamic simulation. $\mathbb{A} \mid \boldsymbol{\Gamma_{task}}$ was computed using the joint moments derived from OpenSim's ID procedure. Alternatively, $\boldsymbol{\Gamma_{task}}$ can be calculated from the $\boldsymbol{F_{ext}}$ at run-time in an ad hoc analysis; however, this calculation was not performed in this study.

Feasible activations at each discrete time of the elbow flexion task were computed using MPLRS as described in Chapter 5.

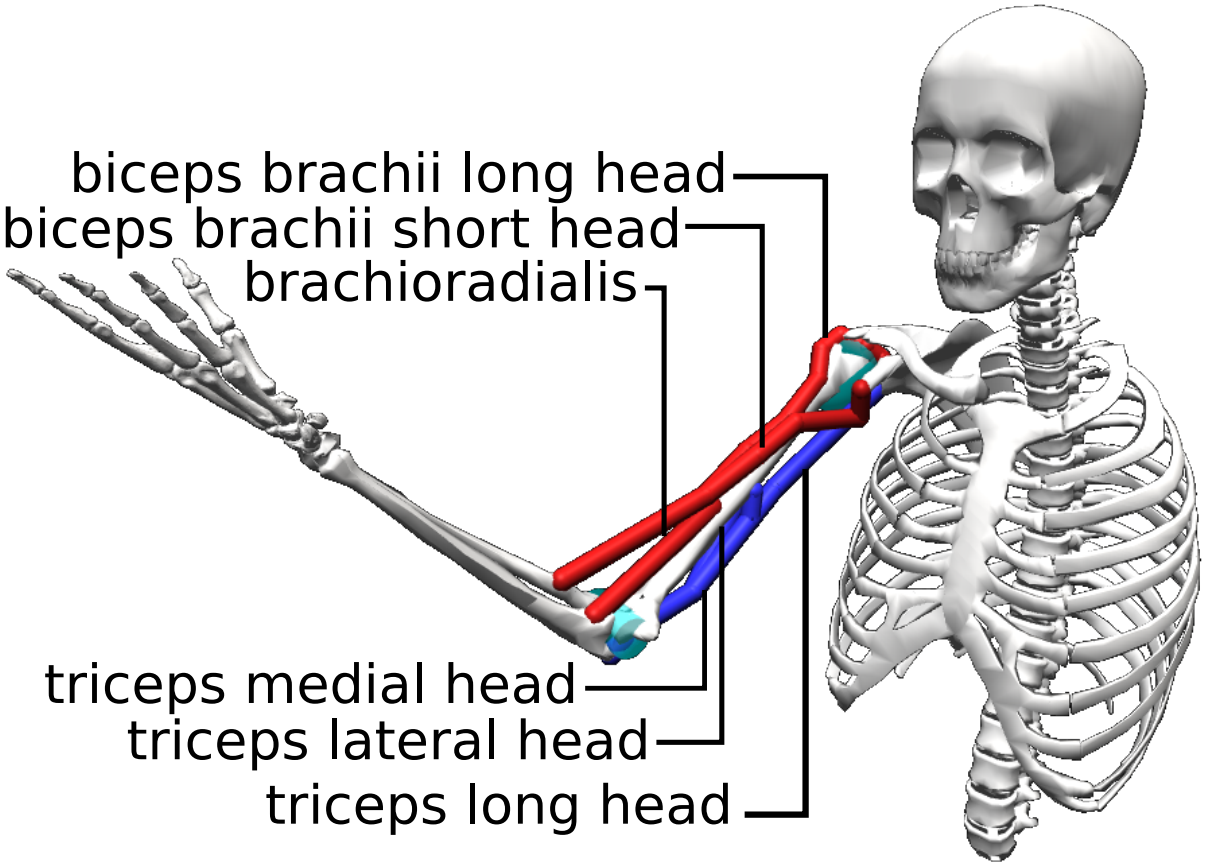


Figure 4.2: Highlighting the 6 muscles of the 2 DOF 6 muscle model (Figures 4.3, 4.6).

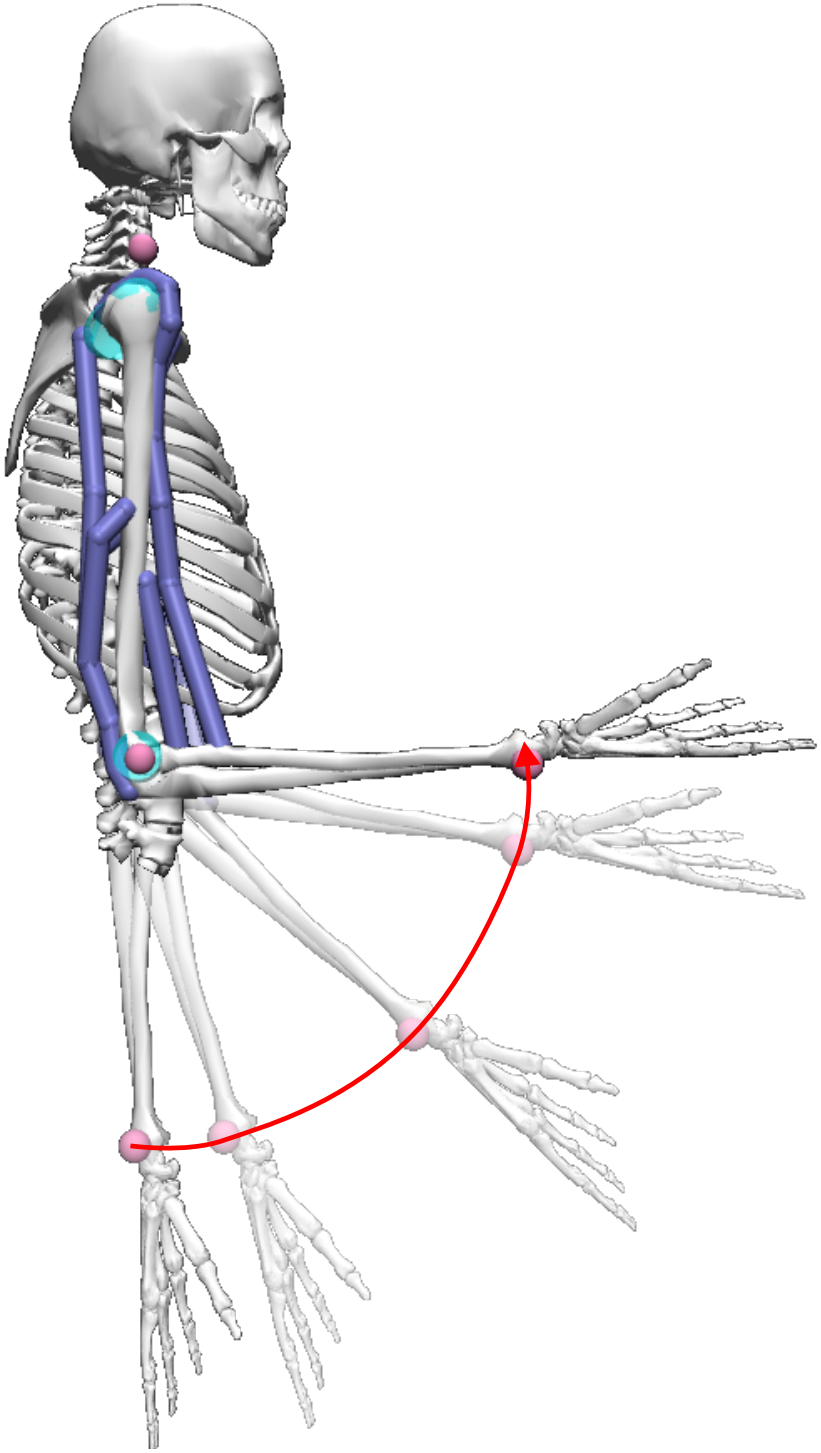


Figure 4.3: Progression of task for the 2 DOF 6 muscles model (Figure 4.6).

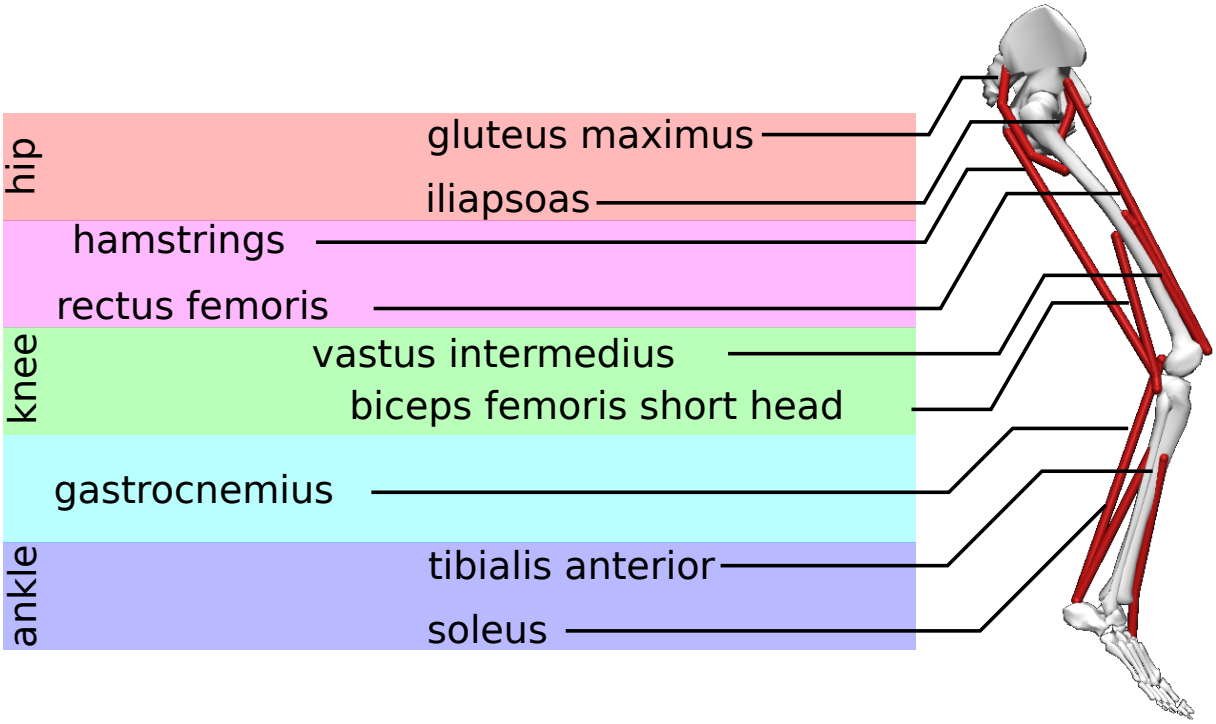


Figure 4.4: Diagram of the muscles of the Gait1018 model used to find feasible activation space (**$\mathcal{V}$-FAS**) bounds (Figure 4.8). Biarticular muscles are between primary colors.

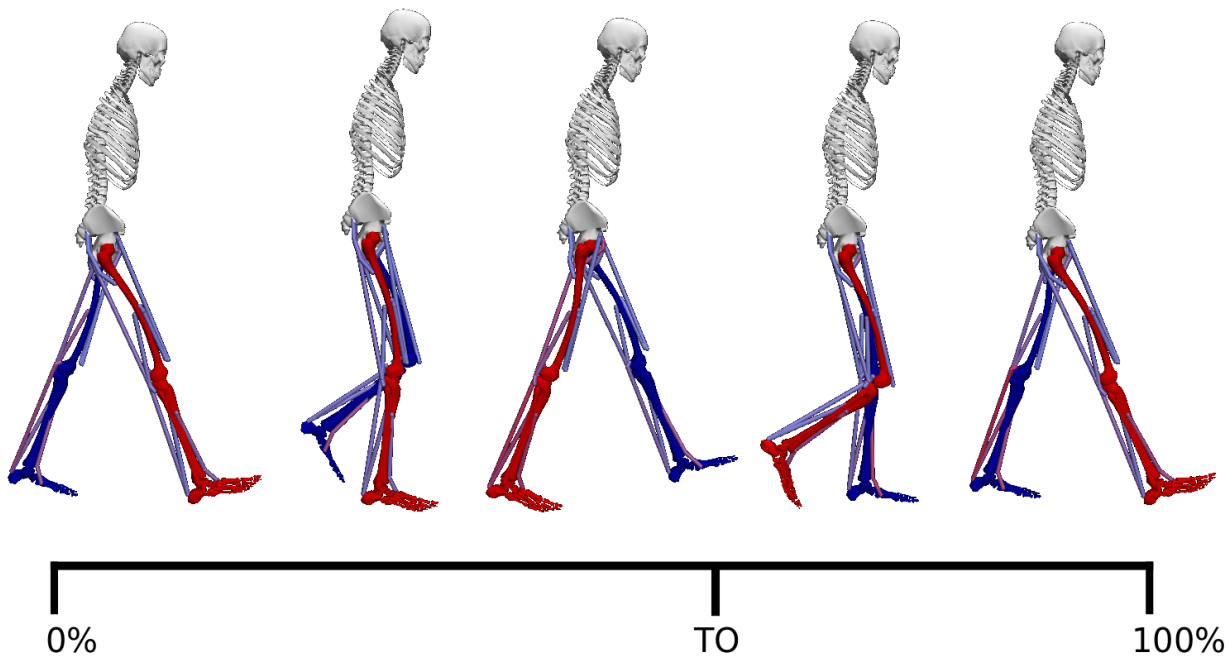


Figure 4.5: Progression of gait for the Gait1018 and Gait2354 models featured in Figures 4.8 and 5.6.

Similarly, a planar 10 DOF 18 muscle gait model (Figure 4.4) freely available with OpenSim was also used to map feasible activations back to joint moments over the course of the gait cycle (Figure 4.5). The model was scaled to the anthropometry of the subject(75 kg, 1.8 m). IK determined model kinematics matching experimental marker data of the subject walking at 1.2 m/s on a treadmill. RRA adjusted model kinematics and inertial properties to minimize dynamic inconsistencies between the model dynamics and experimental ground reaction forces. Muscle maximal isometric force, optimal fiber length, tendon slack length, pennation angle at the optimal fiber length, maximal eccentric force, parallel muscle fiber stiffness, and active force-length-velocity scale factor and passive force-length scale factor were extracted from each muscle over the course of the ad hoc simulation.

$\boldsymbol{\Gamma_{task}}$ (Equation 2.1) is decomposed into active muscle contributions and passive contributions (Equation 4.16) and replacing the moment expression with the muscle force and moment arm expressions (Equations 4.17, 4.18):

$$\boldsymbol{\Gamma_{task} - \Gamma_{passive} = \Gamma_{active}} \tag{4.16}$$

$$\boldsymbol{\Gamma_{task} - R_{c\times m} F^{m}_{passive} = R_{c\times m} F^{m}_{active}} \tag{4.17}$$

$$\boldsymbol{\Gamma_{task} - R_{c\times m} F^{m}_{passive} = R_{c\times m} F^{m}_{0} l_m v_m a} \cos\alpha \tag{4.18}$$

Boundaries of $\boldsymbol{\mathcal{V}}$**-FAS** over each frame of the gait cycle were then computed by vertex enumeration using MPLRS (Chapter 5). Boundaries of activations that satisfy the joint moments over each frame of the elbow flexion of Arm26 model and the gait cycle for the Gait1018 model were then computed by vertex enumeration using MPLRS with procedures later described in Chapter 5. Each vertex, representing a boundary of possible combinations activations that satisfied the joint moment constraints, was then mapped using the projection operator (Equation 4.19).

$$\boldsymbol{\bar{J} = M^{-1} J_t^T (J_t M^{-1} J_t^T)^{-1}} \tag{4.19}$$

Note, to map the ID moments or dually the muscle moment contributions space that satisfies the kinematic requirements of the task. Using Equation 4.2 and rearranging (Equations 4.20, 4.21):

$$M(\boldsymbol{\mathcal{Q}}) = J^T(\boldsymbol{\mathcal{Q}})\Lambda(\boldsymbol{\mathcal{X}})J(\boldsymbol{\mathcal{Q}}) \tag{4.20}$$

$$\boldsymbol{F_t} = \boldsymbol{\bar{J}_t^T \Gamma_t} \tag{4.21}$$

Then Equation 4.22 for feasible accelerations is defined [33]:

$$\boldsymbol{\ddot{\mathcal{X}} = J_t^T M^{-1} \Gamma} \tag{4.22}$$

Each vertex of $\boldsymbol{\mathbb{A} \mid \Gamma_{task}}$, representing a boundary of possible $\boldsymbol{\mathcal{A}}$ that satisfied the joint moment constraints, was then mapped using the projection operator (Equation 4.22).

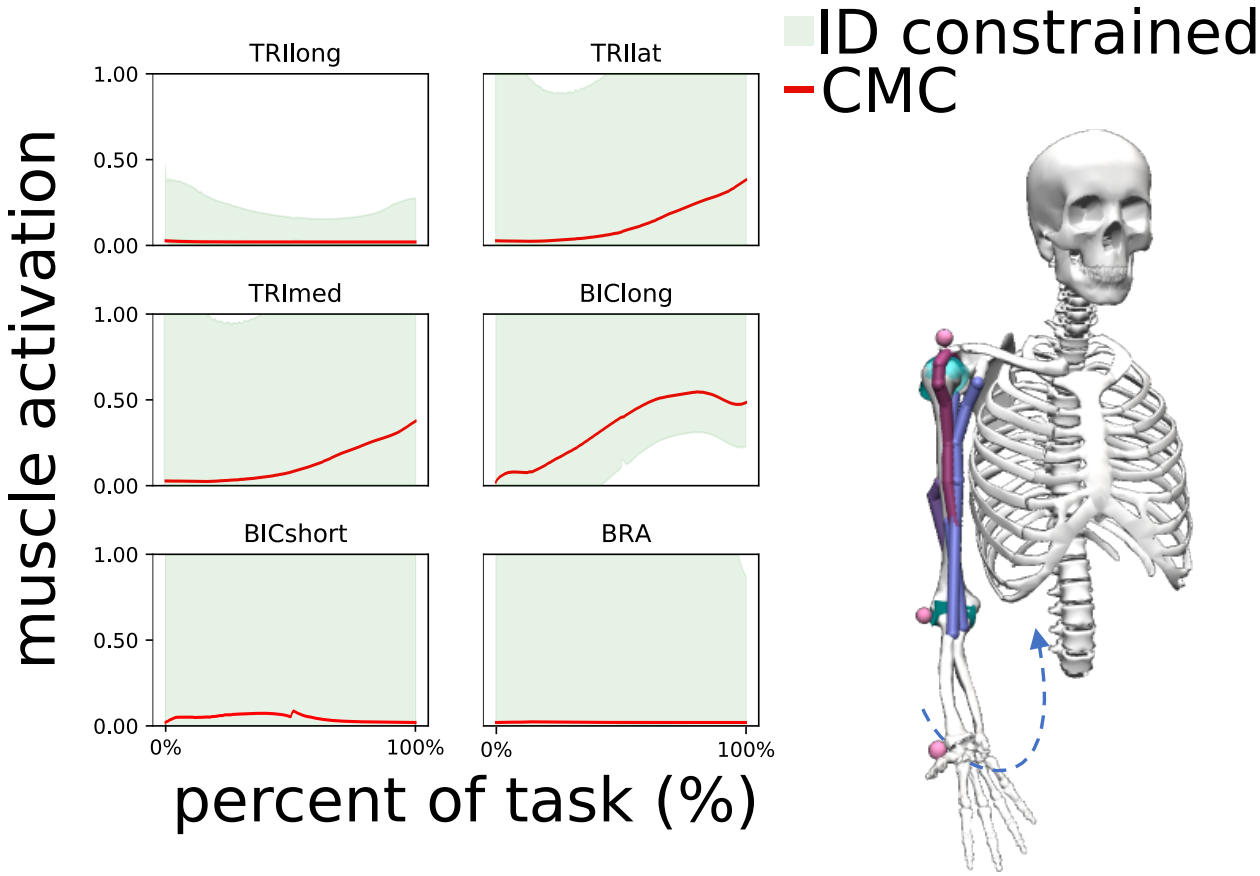


Figure 4.6: Feasible activations of the 6 muscle arm model with OpenSim's CMC solution plotted in red. These results are described in detail in Chapter 5.

The boundaries of the space can be determined using a convex hull algorithm like 3D Jarvis or gift-wrapping to find just the sets of possible dynamically consistent end effector forces or moments, which represents the projection of the 6-Dimensional wrench space onto just the 3 translational dimensions of Cartesian space.

Boundaries of feasible activations for the model over each frame of the gait cycle were then computed by vertex enumeration using MPLRS.

4.2.3 Results

Every $\boldsymbol{\mathcal{A}}$ in feasible activations of the 6 muscles of the Arm26 model (Figure 4.6) for a simple flexion task were mapped to one set of feasible accelerations (Figure 4.7).

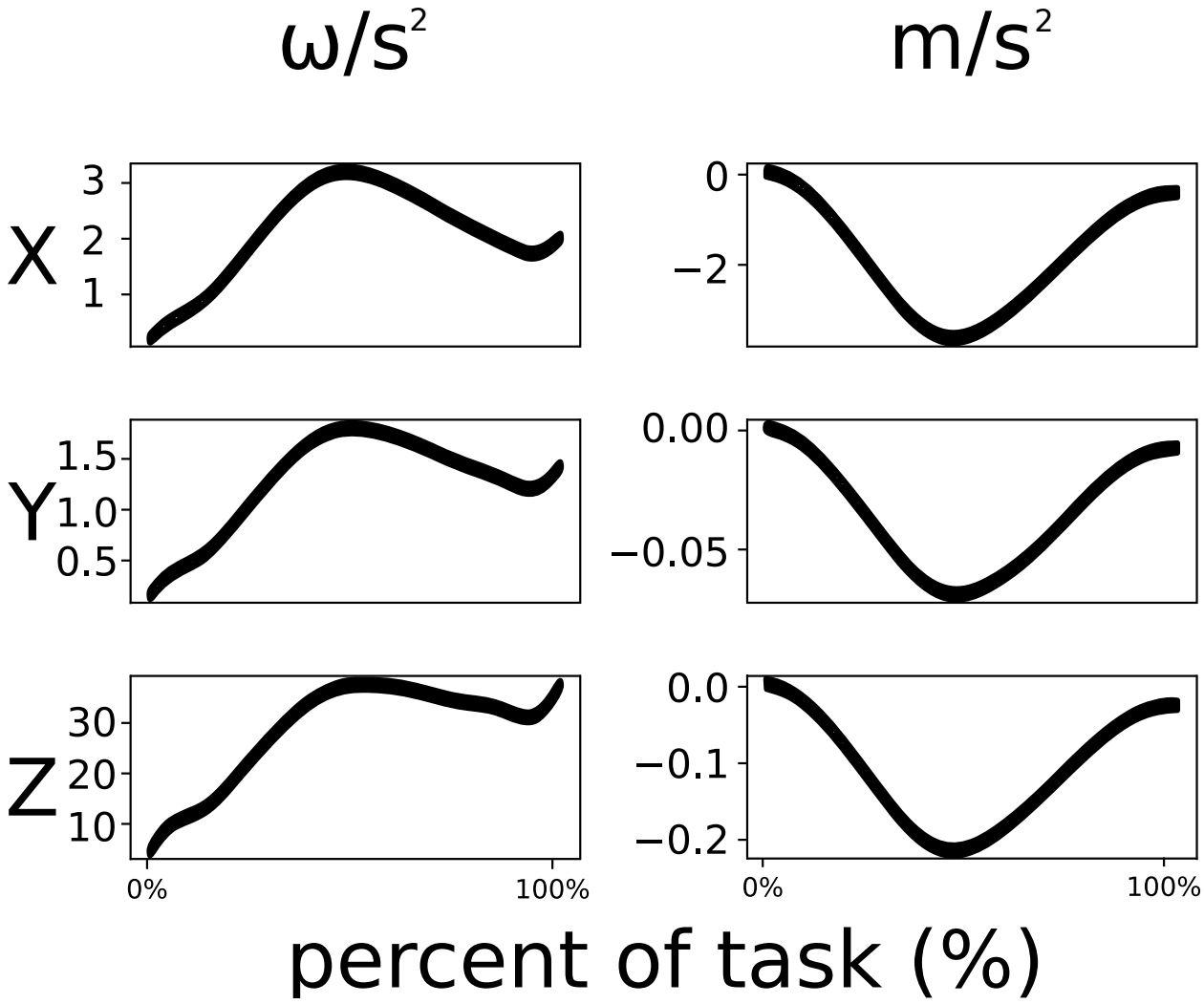


Figure 4.7: Feasible accelerations of the Arm26 model hand mapped from feasible activations (Figure 4.6).

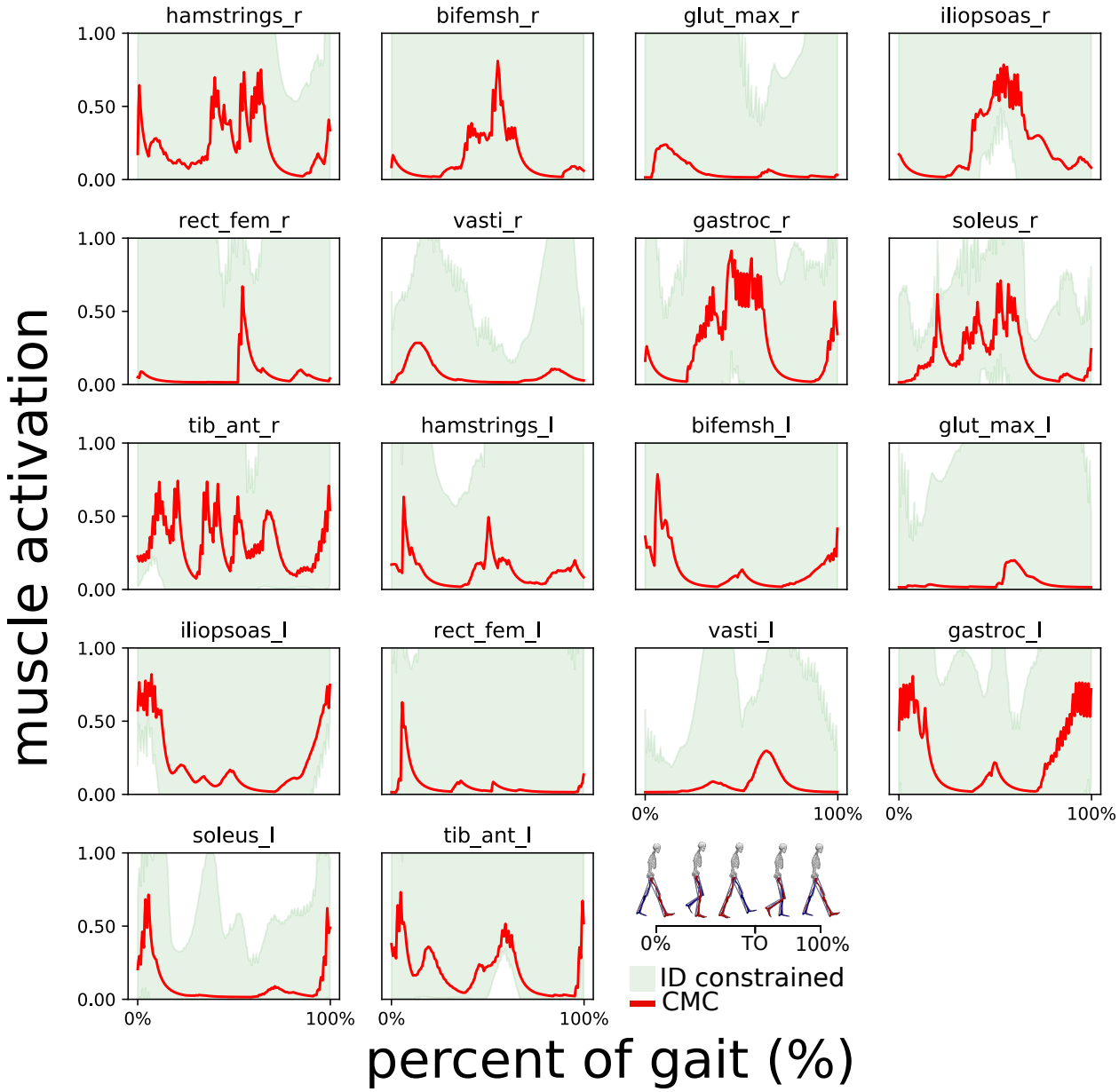


Figure 4.8: Plotting the upper and lower bounds of feasible activations computed by finding the upper and lower bounds for each muscle from the vertices of the feasible activation set over progression of gait. OpenSim's CMC solution is plotted in red. These results are discussed in detail in Chapter 5.

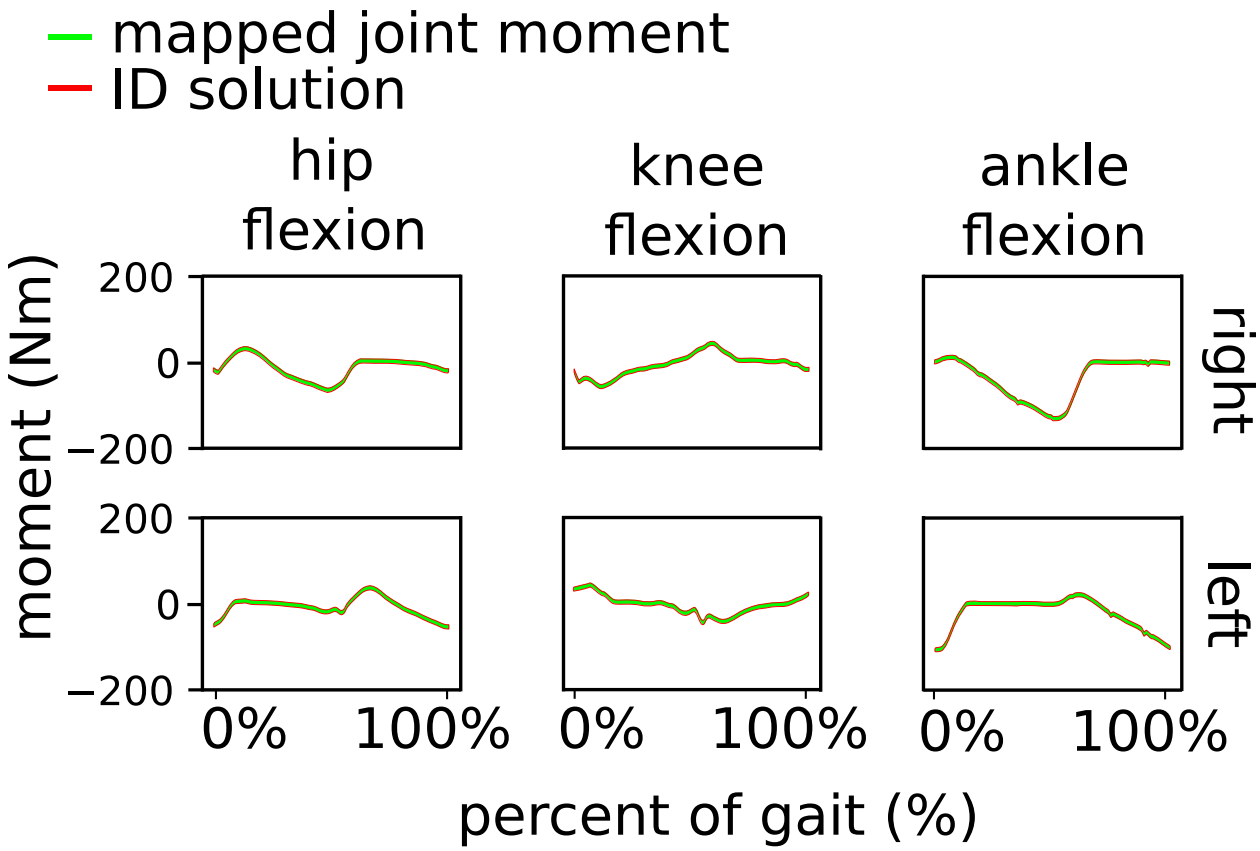


Figure 4.9: Vertices of $\mathbb{A} \mid \boldsymbol{\Gamma}_{\boldsymbol{task}}$ (Figure 4.8) over each percent of gait are used to map to joint moments derived from ID. Only F_x,F_y,M_z are nonzero as the model is planar.

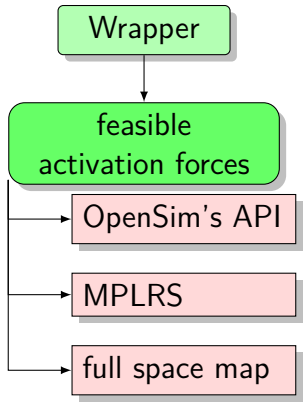


Figure 4.10: Software flowchart.

Similarly, every $\boldsymbol{\mathcal{A}}$ in feasible activations (Figure 4.8) of the 18 muscles of the Gait1018 model (Figure 4.4) over the course of the gait cycle (Figure 4.5) were mapped to one set of joint moments (Figure 4.9). Here is an expansion of feasible sets analysis of the muscle contribution to the dynamic domain. This is a case study to highlight mapping a dynamically consistent feasible activation set to an arbitrary downstream parameter set such as $\mathbb{O}$ acceleration $\ddot{\boldsymbol{\mathcal{X}}}$ or joint moments $\boldsymbol{\Gamma}$.

4.2.4 Discussion

The dynamically consistent accelerations are obtained by mapping the activation space for each discrete time frame of the elbow flexion task. Conceptually, the superset of activations that map to one set of generalized coordinate forces should also map to one set of end effector accelerations. The bounds of $\boldsymbol{\mathcal{V}}$**-FAS** (Figure 4.8) are for the planar Gait1018 model, but later in Chapter 5, a significantly more complex 23 DOF 54 muscles model is analyzed.

4.3 Tools

For advanced developers, an brief exploration of the code and paradigms used in Section 4.2 are outlined in this section (Figure 4.10).

feasibleActivationForces This is the front-end python 3 package for computing feasible force spaces. Construct a full activations hypercube or constrained activations hypercube and map it using a procedurally generated map to $\mathbb{O}$. MPLRS is used to solve vertex enumeration problems in parallel. Full-Space-Map takes an input .map file containing a projection matrix and an input activation polytope in .CSV format, and performs the operation $\boldsymbol{\mathcal{P}} \odot \boldsymbol{\mathcal{A}}$ and writes to a .force file where $\boldsymbol{\mathcal{A}}$ is each vertices of the input polytope in $\mathbb{A}$. One may trivially parallelize this code by splitting the input CSV into multiple parts and running mapper on more than one thread.

findBounds From a supplied feasible activation set for a specific time frame, compute the upper and lower bounds of muscle activations for each muscle. This is essentially the map from the vertex enumeration method of the activations H-rep to the minimization and maximization of the activations H-rep as the LP problem. This is specifically NOT the minimization or maximization of the sum of activations, but it is the LP minimization or maximization such that the objective function is $f = a_i$ for every muscle $i \in m$. Understand that the minimization of the sum total of muscle activations typical in tools like OpenSim static optimization is a particular solution between the bounds determined by findBounds.

findMean From a supplied $\mathbb{A}$, compute the mean of each muscle activation. Trivially, it can be shown that the centroid of a feasible activation space is the sum of the vertices divided by the number of vertices. Since activation is a convex subspace of activation space, it also follows that the average of the vertices must also be a viable solution that satisfies the kinematic and dynamic constraints. This is also known as the vertex center of the space.

mapper From a supplied $\mathbb{A}$ and $\boldsymbol{H}$, for every individual feasible activation set $\boldsymbol{\mathcal{A}}$ in $\mathbb{A}$, perform the matrix multiplication $\boldsymbol{\mathcal{P}} \odot \boldsymbol{\mathcal{A}}$. A typical use case would be to map feasible muscle controls to muscle moment contribution to joints using the moment arms matrix $\vec{R}$.

(fsm)full_space_map Fsm is a command line driver that calls mapper. Arguments are the path to the activation set and the path to the projection operator which are both saved as CSV.

convert Convert is a command line driver that converts files from the cdd/ MPLRS format to CSV and removes headers/footers. This is a convenience method for researchers that transforms the vertex rational string format of MPLRS/LRS to CSV. This is similar to the command line driver rat2float.c supplied with LRS.

4.4 Chapter Summary

The transmission of controls and muscle parameters to any dynamically dependent parameter in $\mathbb{O}$ or $\mathbb{C}$ allows for characterizations of a model's ability in the environment, and by proxy, for a subject-specific musculoskeletal model, allows for characterizations of a subject's capability for a given task. High fidelity insights into a subject's location relative to the family of possible solutions, $\mathbb{A}$ for a given task. A trajectory in $\mathbb{A}$ that moves outside the polytope bounds dictated by the task, necessitates a change in the kinematics. In Chapter 5, the inverse problem, identifying the space of all possible combinations of activations that produce a specified $\mathbb{O}$ or $\mathbb{C}$ constraint, is formulated and performed. These tools are not further used in this dissertation, but served as a sanity test to ensure that the API was being called appropriately and to show that it was possible to back calculate the downstream parameters if the activations space was obtained. In any case, the tools of this chapter may be useful in feasible induced accelerations analysis or explorations of the nullspace of the task.

Chapter 5

Computing Feasible Controls

5.1 Chapter Background

Chapters 3 and 4 rigorously defined some principles of finding the feasible force space comparable to forces found through induced accelerations analysis. Now, instead the inverse problem is explored. Computing activations from end effector forces or joint moments is complicated as it involves CG if researchers desire to characterize the full space. Researchers might think that CG is unneeded because they frequently use optimization frameworks to minimize a particular parameter like energy or muscle tension to obtain an optimized set of controls that can drive a model in the forward dynamic sense, but these frameworks obligate making assumptions about the black box that is neural control. Feasible sets analysis through CG, while computationally expensive, is superior to optimization methods as it facilitates platforms for machine learning and allows researchers to explore the parameter space constrained by a dynamic task.

5.1.1 Geometry

While biomechanists and roboticists might think that visual comprehension of the geometric concepts might be dull, a visual intuition about $\mathcal{V}$**-FAS** is ultimately necessary to understand how to turn an LP into a CG problem and later how to exploit the LP as a MCMC

method: the computational tool used to probe a $\boldsymbol{\mathcal{V}}$**-FAS** interior instead of optimizing a specific objective function f.

n-Dimensional euclidean spaces, $\mathbb{R}^n$, are abstract space with tuples where n independent parameters may be defined. For example, in $\mathbb{R}^3$, points are of the form (x, y, z) for 3 arbitrary parameters x,y,z. $\mathcal{S} \subset \mathbb{R}^n$ is a subspace $\mathcal{S}$ of $\mathbb{R}^n$ such that $\mathcal{S}$ is a subset of $\mathbb{R}^n$. A hyperplane is an $(n-1)$ subspace of $\mathbb{R}^n$. In $\mathbb{R}^2$, these objects are lines. In $\mathbb{R}^3$, these objects are planes. This concept can be extrapolated for higher dimensions. Every $(n-1)$ hyperplane bisects $\mathbb{R}^n$ into two n-Dimensional subspaces of $\mathbb{R}^n$ called halfspaces. A halfspace may be written symbolically as an inequality of the form:

$$c_1 a_1 + c_2 a_2 + \cdots + c_{n-1} a_{n-1} \geq b_i \tag{5.1}$$

Where a_i are activations for each muscle, c_i are some coefficient, and b_i are constraints.

Inequalities $Ax \leq b$ are equivalent to $b - Ax \geq 0$ and matrices known as halfspace representations $\boldsymbol{\mathcal{H}}$ are (Equation 5.2):

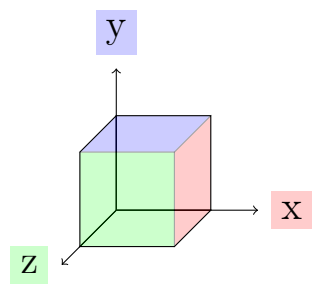
$$H = \begin{bmatrix} b & | & -A \end{bmatrix} \tag{5.2}$$

In Chapter 3, the concept of convex sets was explored. Similarly, in geometry, there is a concept of convex polytopes. In 2D, these objects are polygons (Figure 3.5), and in 3D, these objects are polyhedra (Figures 5.1, 5.2). These polytopes are the simplest geometric objects with flat faces of arbitrary dimension, and are usually just defined in terms of two possible representations: $\boldsymbol{\mathcal{V}}$-polytopes and $\boldsymbol{\mathcal{H}}$-polytopes. $\boldsymbol{\mathcal{V}}$**-FAS**, or FAS-polytopes defined by their vertex representation, consist of the set of vertices, the boundaries of **FAS** (Figure 5.1). By contrast, $\boldsymbol{\mathcal{H}}$**-FAS** consist of the set of inequalities of the form $Aa \leq b$ that define FAS. From $\boldsymbol{\mathcal{H}}$**-FAS**, it is possible to construct the boundaries of $\boldsymbol{\mathcal{V}}$**-FAS** by using the intersections of several halfspaces.

Maximizing or minimizing the sum of activations only gives you a single vertex or interior point of an entire $\mathbb{R}^n$ $\boldsymbol{\mathcal{V}}$**-FAS**. The set of inequalities that form $\boldsymbol{\mathcal{H}}$**-FAS**, and the geometric representation of the intersections of several halfspaces are both critical to understanding how to navigate within $\boldsymbol{\mathcal{V}}$**-FAS**, a task tackled in Chapter 7.

$$0 \leq \mathrm{x} \leq 1$$
$$0 \leq \mathrm{y} \leq 1$$
$$0 \leq \mathrm{z} \leq 1$$

$\mathcal{G}$-**FAS** =



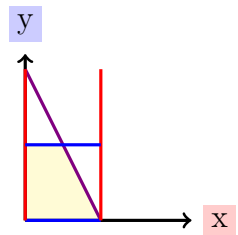
$\mathcal{V}$-**FAS** =
$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$\mathcal{H}$-**FAS** =
$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}$$

Figure 5.1: Four equivalent representations of the unit cube in $\mathbb{R}^3$.

$$
\begin{aligned}
0 \leq \mathrm{y} &\leq 1 \\
0 \leq \mathrm{x} &\leq 1 \\
\mathrm{y} &\leq 2 - 2\,\mathrm{x}
\end{aligned}
$$

$\mathcal{G}$-**FAS** =



$$\mathcal{V}\text{-}\mathbf{FAS} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ .5 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\mathcal{H}\text{-}\mathbf{FAS} = \begin{bmatrix} -2 & 2 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

Figure 5.2: Four equivalent representations of a convex polygon in $\mathbb{R}^2$. Note that the row in $\mathcal{H}$-**FAS** $[1 \;\; -1 \;\; 0]$ is redundant as can be seen in the graph and may be removed from the $\mathcal{H}$-**FAS** matrix.

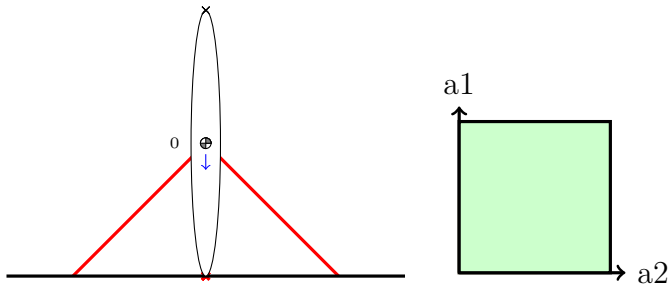


Figure 5.3: A simple 1 DOF model with 2 muscle-tendon actuators, highlighting the muscle redundancy problem. There is more than 1 solution for actuator controls that can satisfy a trajectory through operational space. Activation space here is the unit square.

Thinking about systems in terms of $\mathcal{H}$-**FAS** has a unique challenge. While $\mathcal{V}$-**FAS** generally do not have redundancies unless dealing with floating-point tolerances or the output of a Minkowski sum algorithm as used in Chapter 3, two linear systems may have different $\mathcal{H}$-**FAS**, but the same $\mathcal{V}$-**FAS** (Figure 5.2). $\mathcal{H}$-**FAS** that cannot be reduced further by removing inequalities are called minimal and inequalities that are removed are called redundant. Thankfully, thinking about feasible space in terms of $\mathcal{H}$-**FAS** instead of $\mathcal{V}$-**FAS** has nice properties, even if forced to consider redundancy. For example, an intersection operation of two $\mathcal{H}$-**FAS** is trivial: just concatenate the $\mathcal{H}$-matrices, whereas intersection of two $\mathcal{V}$-**FAS** is hard. Later in Chapter 7, redundant rows in $\mathcal{H}$-**FAS** are shown to influence central points of FAS.

5.1.2 Muscle Redundancy

$$H_{\text{bounds}} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & -1 & 0 & \dots & 0 & 0 \\ 1 & 0 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & -1 & 0 \\ 1 & 0 & 0 & \dots & 0 & -1 \end{bmatrix} \tag{5.3}$$

$\boldsymbol{\mathcal{V}}$**-FAS** is defined as the set of possible muscle-tendon actuator controls that can satisfy the constraints identified through ID, or such that the maximal active component of muscle moments can be scaled to match the ID solution accounting for the muscle passive component of muscle moments and applied external generalized forces as per Equation 5.4.

$$\boldsymbol{\Gamma_{task}} = F_m^{amf}\vec{a}\boldsymbol{R_{m\times c}} + F_m^{pmf}\boldsymbol{R_{m\times c}} - \tau_{ext} \tag{5.4}$$

Or restructuring Equation 5.4 into the halfspace format in Equation 5.5 such that:

$$\boldsymbol{\Gamma_{task}} - F_m^{pmf}\boldsymbol{R_{m\times c}} + \tau_{ext} - F_m^{amf}\vec{a}\boldsymbol{R_{m\times c}} = 0 \tag{5.5}$$

$\boldsymbol{\mathcal{H}}$**-FAS** of an arbitrary musculoskeletal model at a particular time frame t_{sim} can be constructed by appending the $\boldsymbol{\Gamma_{task}}$ expressions (Equation 5.6) to the $\boldsymbol{\mathcal{H}}$**-FAS** (Equation 5.7) of the unit hypercube in $\mathbb{R}^m$:

$$\boldsymbol{\mathcal{H}}\textbf{-FAS} = \begin{bmatrix} -\boldsymbol{\Gamma_{task}} + \sum F_m^{pmf}\boldsymbol{R_{m\times c}} + \tau_{ext} + \mathcal{E} & F_{m_1}^{amf}\boldsymbol{R_{1\times c}}a_1 & \dots & F_{m_m}^{amf}\boldsymbol{R_{1\times c}}a_m \\ \boldsymbol{\Gamma_{task}} - \sum F_m^{pmf}\boldsymbol{R_{m\times c}} - \tau_{ext} + \mathcal{E} & -F_{m_1}^{amf}\boldsymbol{R_{1\times c}}a_1 & \dots & -F_{m_m}^{amf}\boldsymbol{R_{1\times c}}a_m \end{bmatrix} \tag{5.6}$$

$$\boldsymbol{\mathcal{H}}\textbf{-FAS} = \begin{bmatrix} -\boldsymbol{\Gamma_{task}} + \sum F_m^{pmf}\boldsymbol{R_{m\times c}} + \tau_{ext} + \mathcal{E} & F_{m_1}^{amf}\boldsymbol{R_{1\times c}}a_1 & \dots & F_{m_m}^{amf}\boldsymbol{R_{1\times c}}a_m \\ \boldsymbol{\Gamma_{task}} - \sum F_m^{pmf}\boldsymbol{R_{m\times c}} - \tau_{ext} + \mathcal{E} & -F_{m_1}^{amf}\boldsymbol{R_{1\times c}}a_1 & \dots & -F_{m_m}^{amf}\boldsymbol{R_{1\times c}}a_m \\ 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 \\ 1 & -1 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & -1 \end{bmatrix} \tag{5.7}$$

Alternatively, moments expressions are used mapped to end-effector forces or D'Alembert's principle is used to derive expressions relating the inertial properties of bodies to dynamically consistent accelerations. This method can have benefits as it reduces the number of constraints as the set of coordinates of the model is usually much greater than 6 (the maximum number of possible positional and orientational dimensions in Cartesian space). The term $\mathcal{E}$ is a tolerance to the equality constraint to reduce stringency. Remember Chapter 4 showed some examples of taking an existing $\boldsymbol{\mathcal{V}}$**-FAS** and if it is possible to accurately construct the projection operator, then it is possible to map the dynamically consistent feasible forces to some downstream parameter. Muscles might work together, changing their stiffness, increasing the contact force, but not necessarily changing the joint moments required to drive the model. Again, one may flex biceps and triceps in different ways, yet complete the same dynamic task, but different joint contact forces. Further, filtering $\boldsymbol{\mathcal{V}}$**-FAS** using an experimental or model-derived joint contact force is shown in Chapter 6.

Touched in Chapter 2, if only the upper and lower bounds of each muscle over the course of a dynamic task is desired, an equivalent problem is the following LP:

$$
\begin{aligned}
&\min a_m \\
&\text{s.t.} 0 \leq m \leq 1 \\
&\tau = \sum_{i=0}^{m} H a_i \\
\forall m \qquad &\text{AND} \\
&\max m_i \\
&\text{s.t.} 0 \leq m \leq 1 \\
&\tau = \sum_{i=0}^{m} H a_i
\end{aligned}
\tag{5.8}
$$

However, this optimization analysis (Equation 5.8) has marginal utility other than to identify muscles that are necessary to perform a specific task. By contrast, the CG offers us exactly how muscles are coupled to perform the task. How must the other muscles compensate when one muscle is minimally or maximally activated is immediately observable in feasible sets analysis. This is an extremely rich data set that is possible to generate on each discrete time frame of the gait cycle and is conducive to analysis through MCMC methods or neural networks. Any interior point of an activation space can be expressed as the linear combination of the vertices of $\boldsymbol{\mathcal{V}}$**-FAS**. A case study is offered where $\boldsymbol{\mathcal{V}}$**-FAS** is constrained by the ID moments. The results from this example should be familiar, as they were used in Chapter 4 along with linear operators to map dynamically consistent activations to interesting downstream parameters.

5.2 Study: Computing Feasible Controls

5.2.1 Case Study Motivation

Feasible sets analysis can be used to investigate the boundaries of control for arbitrary models performing arbitrary tasks. The entirety of the space that defines the boundaries of control, $\boldsymbol{\mathcal{V}}$**-FAS**, can be calculated at each discrete time of a kinematic trajectory using the inverse dynamics moments $\boldsymbol{\Gamma}_{\textbf{task}}$. Some case studies are offered using feasible sets analysis with OpenSim's freely available Arm26, Gait1018, and Gait2354 models, and the CG tool

MPLRS [13]. Additionally, a reflection of Chapter 3 is offered where the feasible control for one subject using a planar model is used to highlight the differences in $\boldsymbol{\mathcal{V}}$**-FAS** under different levels of muscle model complexity.

5.2.2 Methods

First, a 2 DOF 6 muscles arm model (Figure 4.2) freely available with OpenSim was used to identify feasible muscle activations that satisfy the ID-determined constraints over the course of a simple elbow flexion task (Figure 4.3).

Second, A planar gait model with 10 degrees of freedom and 18 muscles available in OpenSim was used to explore the influence of muscle physiological consideration on the boundaries of $\boldsymbol{\mathcal{V}}$**-FAS** over progression of gait. The model was scaled to the anthropometry of the subject (75 kg, 1.8 m). IK determined model kinematics matching experimental marker data of the subject walking at 1.2 m/s on a treadmill. RRA adjusted model kinematics and inertial properties to minimize dynamic inconsistencies between the model dynamics and experimental ground reaction forces. CMC determined muscle excitations and resulting model states for a forward dynamic simulation.

Additionally, using data collected as part of [50], the freely available Gait2354 model (Figures 5.4) packaged with OpenSim [86] was scaled to the anthropometry of the subject (1.8 m, 75.16 kg). IK procedure was used to determine the model kinematics matching experimental marker data walking on treadmill at a self selected speed. RRA was used to minimize dynamic inconsistencies in the model dynamics and experimental ground reaction forces. CMC determined muscle excitations and resulting model states for a forward dynamic simulation.

For each of the 3 models described, for each discrete time of the kinematic task, the boundaries of activation space are computed by constructing $\boldsymbol{\mathcal{H}}$**-FAS** (Equation 5.7) and performing CG using MPLRS [13].

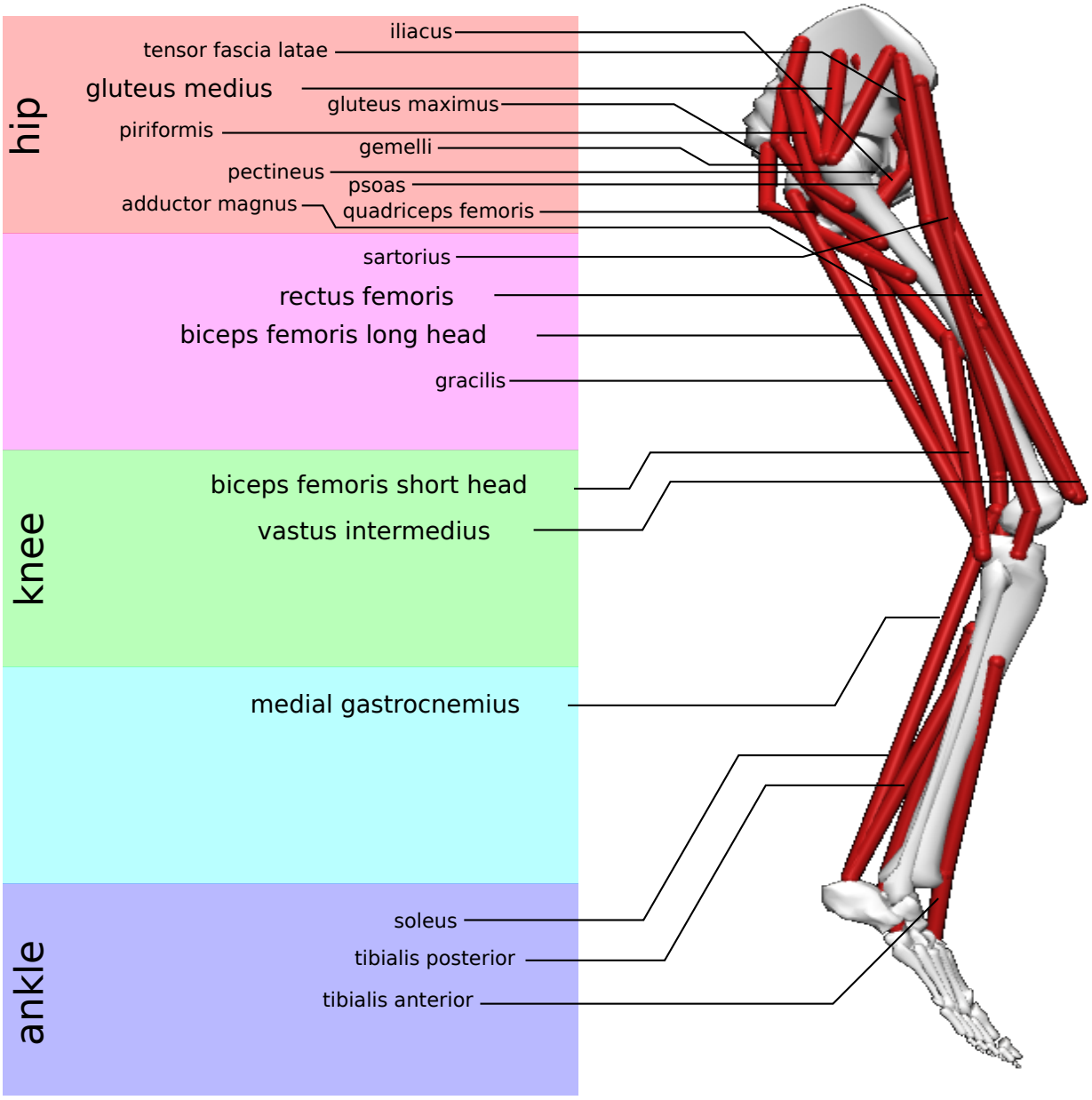


Figure 5.4: Diagram of the muscles of the Gait2354 model used to find **$\mathcal{V}$-FAS** bounds (Figure 5.6). Biarticular muscles are between primary colors.

5.2.3 Results

The ID-constrained $\boldsymbol{\mathcal{V}}$**-FAS** bounds were found for the simple Arm26 model. The nonzero lower bounds of biceps brachii long head (Figure 4.6) imply that for this model, this muscle is necessary: it is impossible to perform the elbow flexion without using biceps brachii long head.

Similarly, the planar Gait1018 model (Figure 4.4) was used to find feasible muscle activations over the course of the gait cycle. These results agree well with the OpenSim CMC solution (Figure 4.8). Necessary muscles can be observed from nonzero lower bounds: particularly iliopsoas near the right toe off is necessary for performing the hip flexion to prepare the leg for swing phase. Also, tibialis anterior plays a critical role in the ankle plantar-flexion during the heelstrike. Note that the specific kinematic trajectories analyzed in this feasible sets analysis show that iliopsoas and tibialis anterior are not just critical, but absolutely necessary to performing these specific behavioral tasks, it may be possible to modify the kinematics in such a way that still produces gait without needing these two muscles.

To revisit the static parameter analysis from Chapter 3 (Figure 5.5), $\boldsymbol{\mathcal{V}}$**-FAS** was different under different physiological considerations. $\boldsymbol{v_m}$ considerations increased the necessity of tib_ant during ankle dorsiflexion near heel strike at 0%. Interestingly, no muscles became less necessary when including $\boldsymbol{v_m}$ over progression of gait as would be indicated by a lowered lower bound, therefore including the muscle parameters seems to constrain feasible controls, but further analysis with more complicated multi-joint models should be explored.

In the analysis of the Gait2354 (Figure 5.6), very few muscles were found to be necessary as indicated by nonzero lower bounds on muscle activation. Nonzero lower activation bounds indicated that only glut_med1 during toe off and tib_ant during heel strike were deemed necessary muscles for gait, and other muscles were able to account for the functions of these two muscles. Interestingly, increasing the model complexity from Gait1018 to Gait2354 reduced the necessity of muscles. These results align with previous works that used linear optimization to identify bounds of muscle activation over the course of the gait cycle [87]; however, these papers used a more complex model featuring 92 muscles instead of the 54

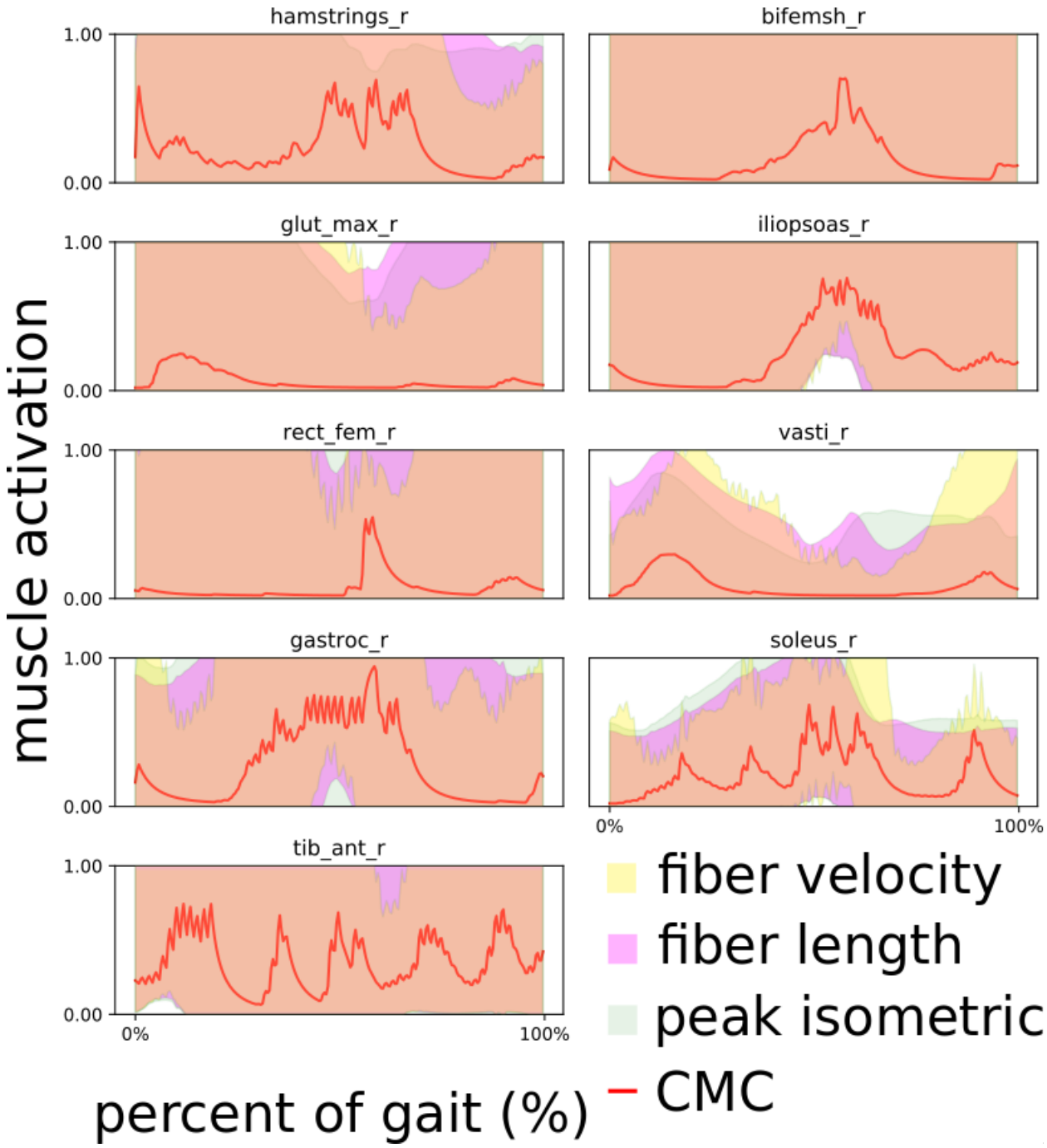


Figure 5.5: Revisiting Chapter 3. From the pseudostatic analysis, it is possible to investigate how muscle model complexity influences the bounds on FAS.

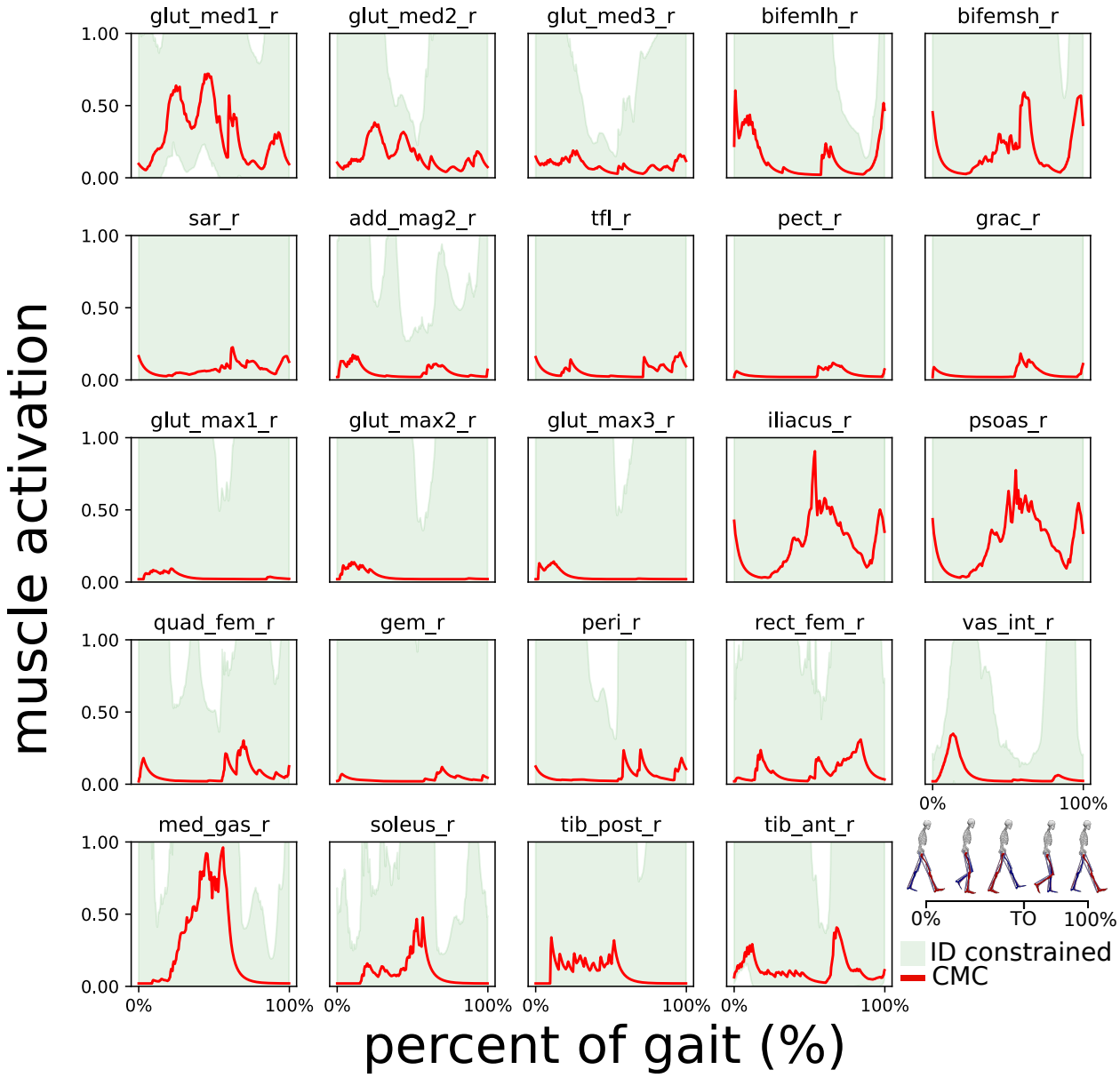


Figure 5.6: $\mathcal{V}$-FAS of the 24 leg muscles over the course of the gait cycle with the OpenSim CMC solution in red.

muscle model analyzed here. CG is limited to within 40 muscles even with sophistication such as trivial parallelization or running the analysis on supercomputers/clusters due to the computational complexity of CG.

5.2.4 Discussion

In each of the 3 models investigated, ID-constrained $\boldsymbol{\mathcal{V}}$**-FAS** (Figures 4.6, 4.8, and 5.6) bounds the OpenSim CMC solution and highlights that there is a wide range of possible combinations of actuator controls instead of the optimal solutions found through CMC or SO.

Key differences in the feasible activation ranges were identified under different muscle physiological consideration (Figure 5.5). For some muscles like iliopsoas, $\boldsymbol{\mathcal{V}}$**-FAS** bounds under the $\boldsymbol{v_m}$ consideration were constrained from the peak isometric case, whereas for some muscles like tibialis anterior, the inclusion of $\boldsymbol{v_m}$ considerations increased the lower bound of feasible activations and increased muscle necessity. Near toe off, $\boldsymbol{v_m}$ consideration constrained $\boldsymbol{\mathcal{V}}$**-FAS** for muscles (glut_max, vastus lateralis, iliopsoas,rectus femoris, gastrocnemeus), but increased $\boldsymbol{\mathcal{V}}$**-FAS** bounds for soleus and hamstrings. These results mirror concerns discussed previously in the literature [7, 117] and highlight the importance of the muscle physiology being incorporated in the muscle model to obtain more accurate controls approximations.

Feasible sets analysis has promise in the design of MCMC methods, yet remains computationally expensive; however, it may be possible to reduce computational complexity by grouping muscles by function or to perform a principal component analysis to identify muscle synergies and performing feasible sets on a reduced dimension muscle synergy set instead of a large muscle set.

5.3 Tools

For advanced developers, an in-depth exploration of the mathematics and code and paradigms used are outlined in this section (Figure 5.7).

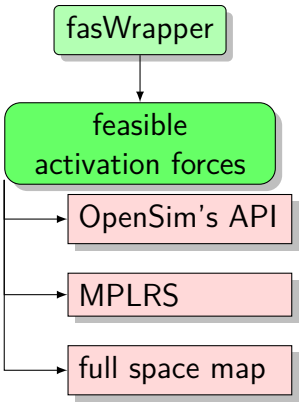


Figure 5.7: Software flowchart.

feasibleActivation

Feasible activation sets analysis such that the activations $\mathcal{H}$representation is only bounded by the ID constraints

5.4 Chapter Summary

This basic framework of constructing $\mathcal{H}$-**FAS** to fully define the parameter space at each discrete time over the gait cycle is the basic framework that is expanded in Chapters 6 and 7. While Chapter 6 continues on the path of using $\mathcal{H}$-**FAS** to CG to concretely define activations $\mathcal{V}$-**FAS** for each discrete time of the dynamic task, Chapter 7 instead uses $\mathcal{H}$-**FAS** to navigate the interior of feasible space using probability distributions.

Chapter 6

Constraining Feasible Controls by Experimental Joint Forces

6.1 Chapter Background

It is possible to take the methodologies developed in Chapter 5 and find the $\mathcal{V}$-**FAS** of the muscles and constrain by an experimentally recorded or *in silico* derived muscle-dependent parameter [32, 60]. In this chapter, the joint contact forces (JCF) derived using OpenSim Joint Reaction Analysis (JRA) is used to constrain feasible controls; however, the method of constructing $\mathcal{H}$-**FAS** over each discrete time of a kinematic task by using a procedurally generated analytical solution may be extrapolated to any muscle dependent parameter. Also, the method of using JRA to constrain controls can be extrapolated to using data recorded with an instrumented knee.

6.1.1 Procedurally Constructing Constraints

The topology of the multibody tree (Figure 6.1) is important as it is a graph containing the information about how forces need to be summed to obtain downstream parameters. The connection from ground to pelvis represents the virtual DOFs relating the base node pelvis to the position and orientation of the entire system in the global cartesian frame. It is possible to programmatically define the path from relevant leaves of the multibody graph to the

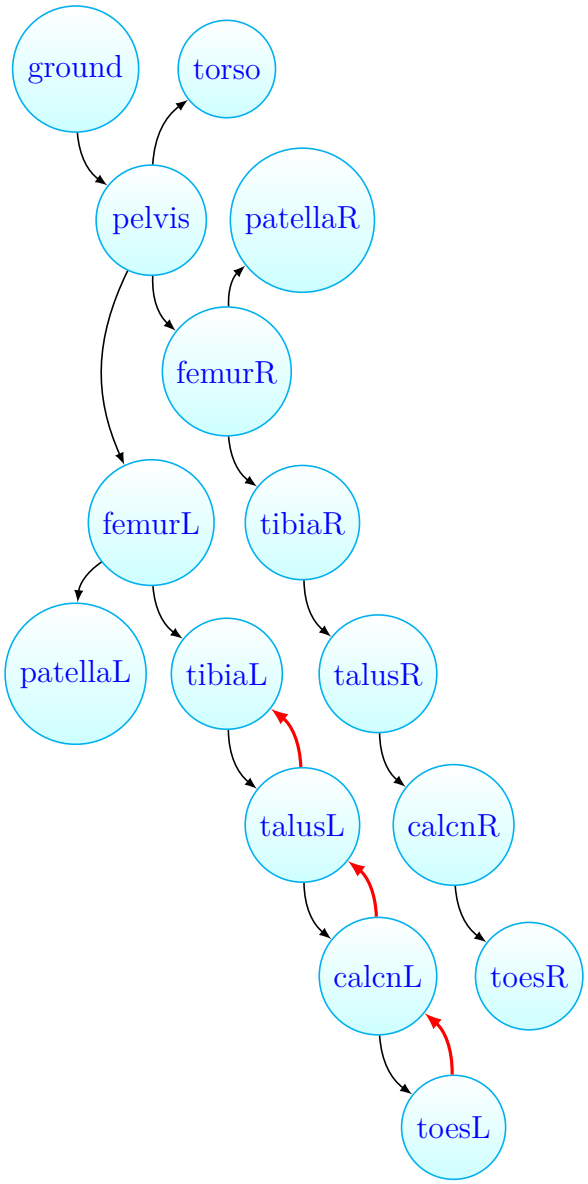


Figure 6.1: A typical multibody tree of a gait focused musculoskeletal model. The path to compute left knee loads is depicted in red. The linkage between the ground and the pelvis reflects the virtual DOFs that describe the MBS position and orientation relative to the $\mathbb{O}$ origin.

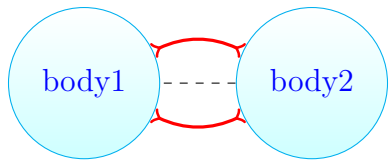


Figure 6.2: Joint relating the connection between body1 and body2 as a dashed line, and the muscle (red) applying tension to bodies (blue).

important body of a joint desired to be constrained, by constructing a directed graph based on the bodies that compose each joint. OpenSim is a fantastic tool for musculoskeletal modeling; however, it does not offer analytical expressions for the JCF, so methods of estimating the loads have to be implemented to construct FAS constraints that map to JCF [32, 59, 64, 123]. Similarly, there exist plugins that solve for the JCF, yet no solutions exist that procedurally construct the analytical expressions. [1] A limitation of this approach is the inability to deal with closed-loop kinematic chains; however, future work should be able to address this limitation.

Generally, when solving for JCF, one approach is to work up the kinematic chain from the location of the applied external loads and considering the linear acceleration of bodies at t_{sim} [4, 37, 113, 118, 124].

6.1.2 Muscle Lines of Action

Muscles apply tension (Figure 6.2) in directions from points on the bodies where their tendons insert to the bone (Figure 2.8) [32]. In OpenSim, it is trivial to find the geometry path of a muscle and the line of action of the applied on-body force.

To relate the experimentally recorded JCF to the model $\boldsymbol{\mathcal{A}}$:

$$\begin{aligned} F_j = \; & \vec{F_{m_{am_{f_{b_1}}}}} + \cdots + \vec{F_{m_{am_{f_{b_b}}}}} + \\ & \vec{F_{m_{pm_{f_{b_1}}}}} + \cdots + \vec{F_{m_{pm_{f_{b_b}}}}} + \\ & F_{ext_{b_1}} + \cdots + F_{ext_{b_b}} + \\ & \vec{a_{b_1}} m_{b_1} + \cdots + \vec{a_{b_b}} m_{b_b} \end{aligned} \tag{6.1}$$

[1] These tools instead rely on Simbody to do that hard work.

By rearranging Equation 6.1 and concatenating this matrix to the Equations 5.3 and 5.6:

$$RHS = \sum \left[F_j - \vec{F_{m_{pmf_b}}}_{\sum 1 \to b} - \vec{F_{ext_b}}_{\sum 1 \to b} \right]$$

$$LHS = \vec{F_{m_{amf_b}}}_{\sum 1 \to b}$$

$$H = \begin{bmatrix} -\Gamma_{task} + \\ \sum(F_m^{pmf} R_{m \times c}) \\ +\tau_{ext} + \mathcal{E} & F_{m_1}^{amf} R_{1 \times c} a_1 & \dots & F_{m_{m-1}}^{amf} R_{1 \times c} a_{m-1} & F_{m_m}^{amf} R_{1 \times c} a_m \\ \Gamma_{task} - \\ \sum(F_m^{pmf} R_{m \times c}) \\ -\tau_{ext} + \mathcal{E} & -F_{m_1}^{amf} R_{1 \times c} a_1 & \dots & -F_{m_{m-1}}^{amf} R_{1 \times c} a_{m-1} & -F_{m_m}^{amf} R_{1 \times c} a_m \\ -RHS + \mathcal{E}_2 & LHS_{a_1} & \dots & LHS_{a_{m-1}} & LHS_{a_m} \\ RHS + \mathcal{E}_2 & -LHS_{a_1} & \dots & -LHS_{a_{m-1}} & -LHS_{a_m} \\ 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \\ 1 & -1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & \dots & -1 & 0 \\ 1 & 0 & \dots & 0 & -1 \end{bmatrix} \tag{6.2}$$

6.2 Study: Constraining Controls by Joint Loads

6.2.1 Case Study Motivation

Biomechanics researchers or clinicians generally rely on optimization to find sets of muscle-tendon actuator controls capable of driving a musculoskeletal model in forward dynamic simulations to gain insights into neural control. Similarly, when humans design robots, they want each degree of freedom or generalized coordinate to be controlled by a single motor or linear actuator. By contrast, biological systems employ many redundant muscles that overlap in function which is a complicated controls problem. To address this issue, researchers compute $\boldsymbol{\mathcal{A}}$ by minimizing parameters like energy or muscle tension. $\boldsymbol{\mathcal{V}}$**-FAS** is the set of all possible $\boldsymbol{\mathcal{A}}$ that can satisfy the ID and IK constraints for a specific discrete time. The joint contact forces are a function of both the inertial properties of the bodies and also the geometry and physiology of the muscles that span the joints [17]. Instead of relying on optimization, an analytical approximation of joint contact forces is used to constrain $\boldsymbol{\mathcal{V}}$**-FAS** found using CG.

6.2.2 Methods

It is possible to construct a linear equation of the form (Equation 5.1) that maps the active muscle force contribution to a parameter like generalized force. Coefficients of such equations can be combined to form the halfspace representation ($\boldsymbol{\mathcal{H}}$) of a convex polytope. Rewriting Equation 6.2 in a more concise form (Equation 6.3):

$$H = \begin{bmatrix} -\Gamma_{task} & \tau_1 & \tau_2 & \dots & \tau_m \\ -R_{exp} & F_1 & F_2 & \dots & F_m \\ \Gamma_{task} & -\tau_1 & -\tau_2 & \dots & -\tau_m \\ R_{exp} & -F_1 & -F_2 & \dots & -F_m \\ 0 & a_1 & 0 & \dots & 0 \\ 0 & 0 & a_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_m \\ 1 & -a_1 & 0 & \dots & 0 \\ 1 & 0 & -a_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & -a_m \end{bmatrix} \tag{6.3}$$

$\mathcal{V}$-FAS is found by CG on **$\mathcal{H}$-FAS** using an algorithm such as reverse search developed by Avis and Fukuda [10]. While computationally expensive relative to optimization, CG produces extremely rich data sets that describe the boundaries of the possible combinations of every muscle and avoids pitfalls like finding only local minima.

6.2.3 Results

The joint-constrained **$\mathcal{V}$-FAS** bounds were found for the simple arm26 model (Figure 6.3) freely available with OpenSim where the elbow loading obtained using JRA is used to constrain **$\mathcal{V}$-FAS**. JCF-constrained **$\mathcal{V}$-FAS** bounds the OpenSim CMC solution. Perhaps not surprisingly, biceps brachii long head is important for elbow flexion and becomes more necessary when constraining by the JCF.

Similarly, the planar Gait1018 model (Figure 6.4) provided with OpenSim was used to generate *in silico* joint contact forces of the knee over the course of the gait cycle. OpenSim CMC and JRA were used to generate experimental joint forces. These results agree well with the OpenSim CMC solution (Figure 6.5). Hamstrings and gastrocnemius can be significantly constrained while other muscles have much higher space for variability.

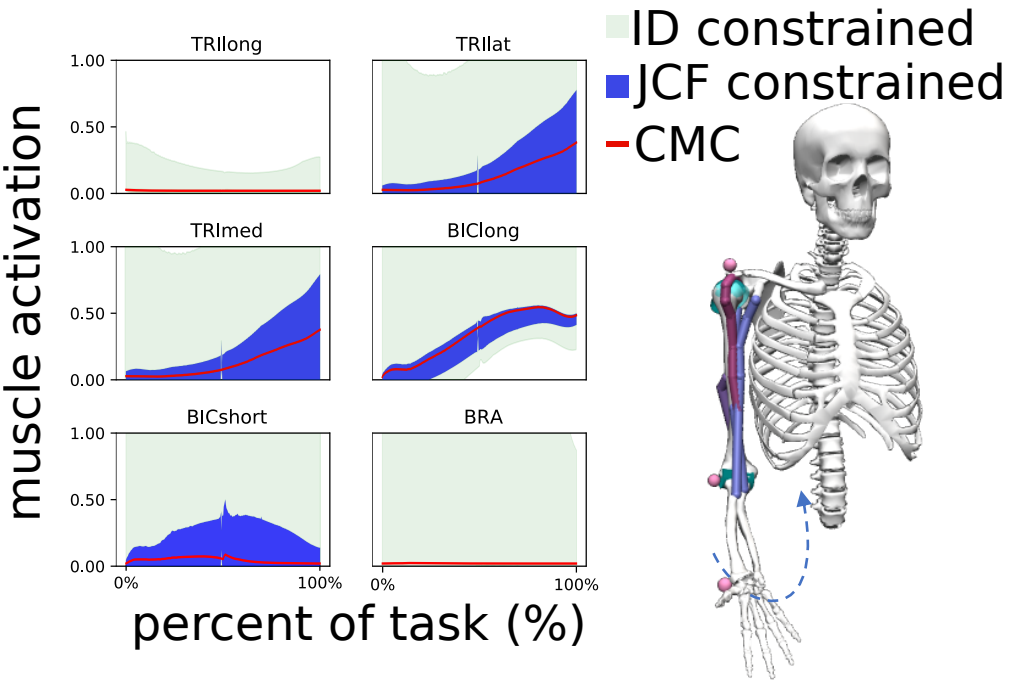


Figure 6.3: Similarly, $\mathcal{V}$**-FAS** for an upper extremity model performing an elbow flexion with the hand supinated. Joint loads were determined using OpenSim JRA.

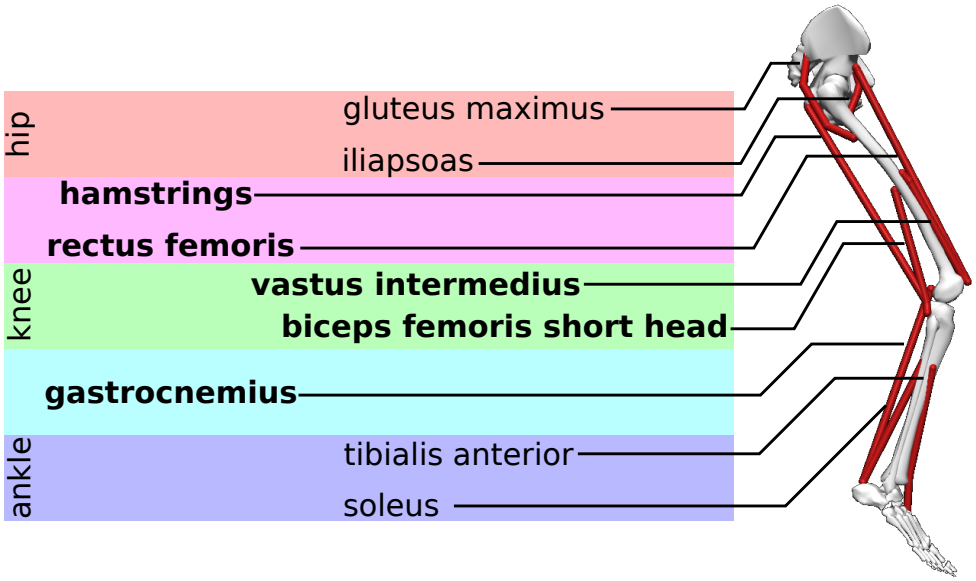


Figure 6.4: Diagram of the muscles of the Gait1018 model used to find $\boldsymbol{\mathcal{V}}$**-FAS** bounds in Figure 6.5 highlighting the most constrained muscles in bold. Biarticular muscles are between primary colors.

The previous Gait1018 analysis is applied to the more complex Gait2354 model (Figure 6.6) available freely with OpenSim. The boundaries of $\boldsymbol{\mathcal{V}}$**-FAS** is constrained by the knee loading. JCF-constrained $\boldsymbol{\mathcal{V}}$**-FAS** bounds the CMC solution as in the previous two examples (Figure 6.7). Feasible $\boldsymbol{\mathcal{A}}$ for medial gastroc, rectus femoris, vastus intermedius and biceps femoris long head is very significantly constrained relative to other muscles, highlighting the relative importance of these muscles in the generation of knee loads. These results differ slightly from the Gait1018 results (Figure 6.4) as models with more muscles have more room to manipulate the muscles to produce the desired forces. This analysis was performed without any grouping of muscles by function or muscle synergies, so these aspects of control are not reflected, but may be an interesting avenue for future work. Feasible sets analysis captures the CMC solution using the joint reactions, but highlights that there is a range of possible controls.

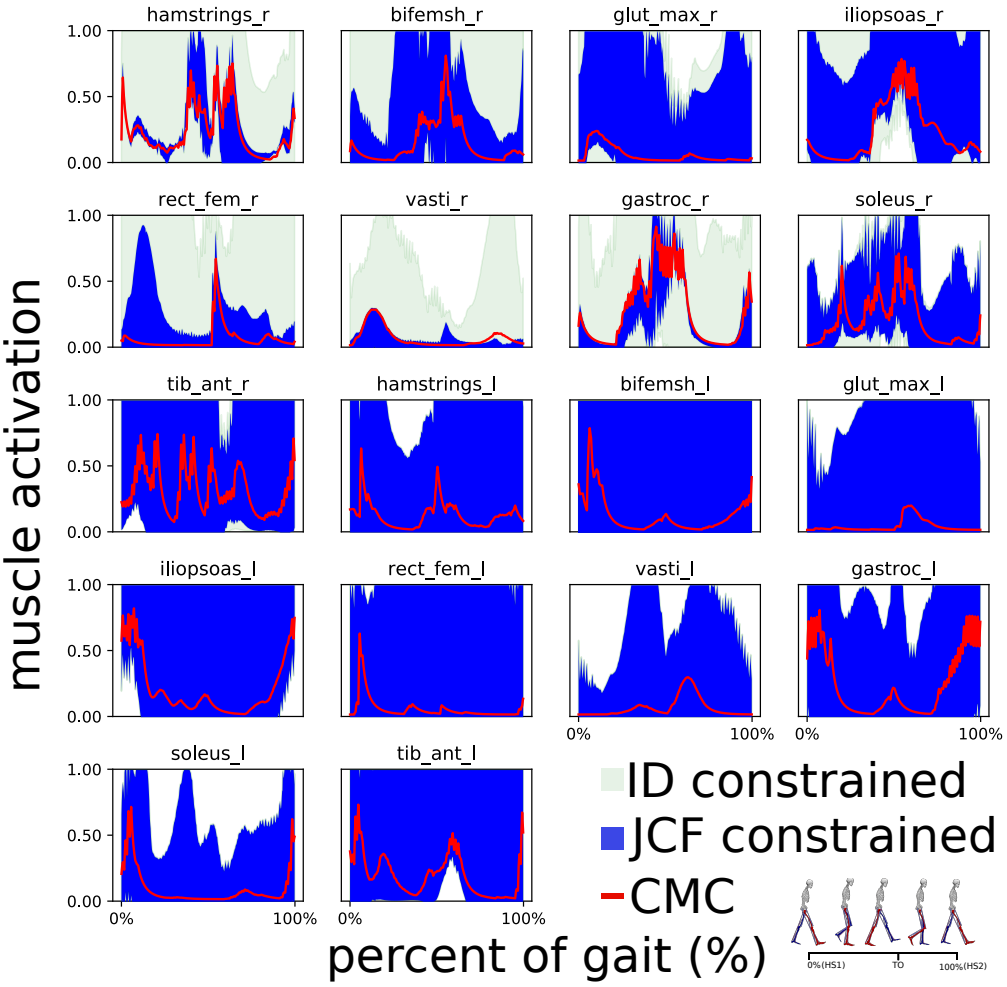


Figure 6.5: $\mathcal{V}$-FAS for the right leg constrained by the JCF. JRA determined joint contact forces of the right knee only, which are functions of the muscle states derived from CMC.

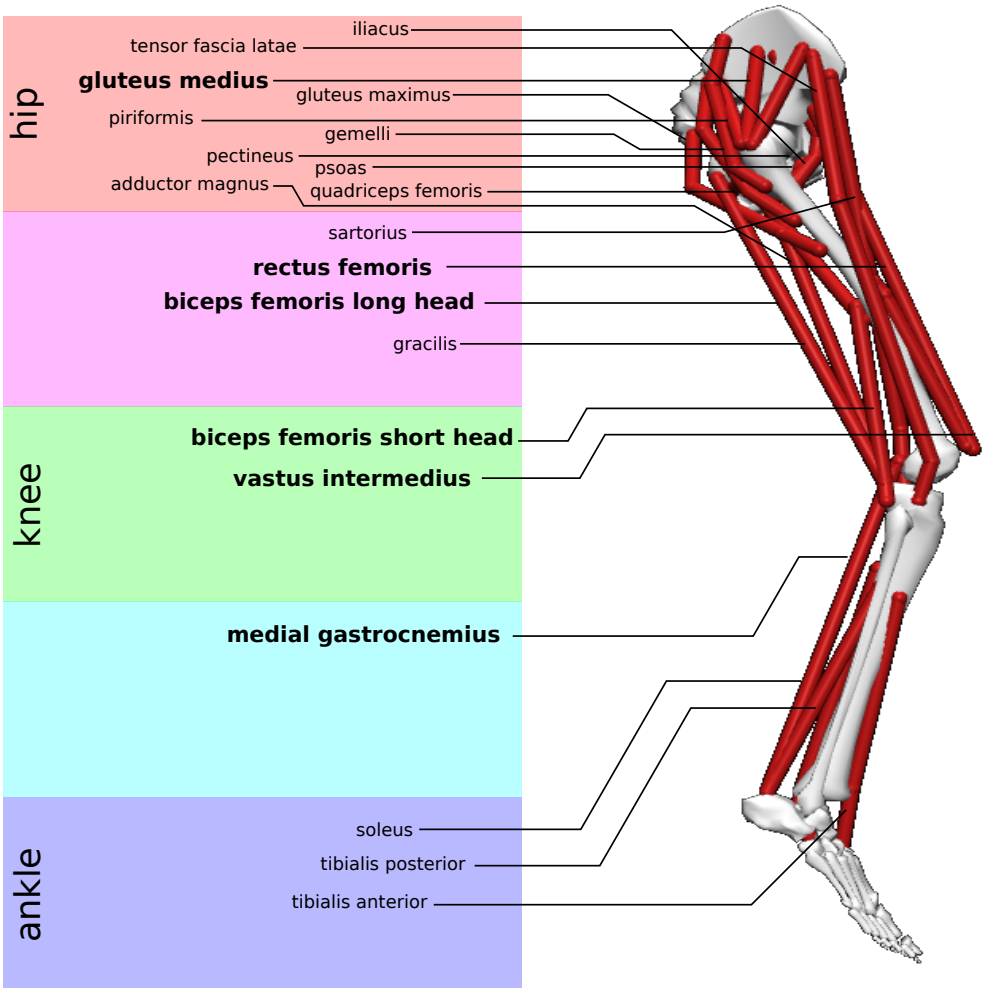


Figure 6.6: Diagram of the muscles of the Gait2354 model used to find $\mathcal{V}$**-FAS** bounds (Figure 6.7) highlighting the most constrained muscles in bold. Biarticular muscles are between primary colors.

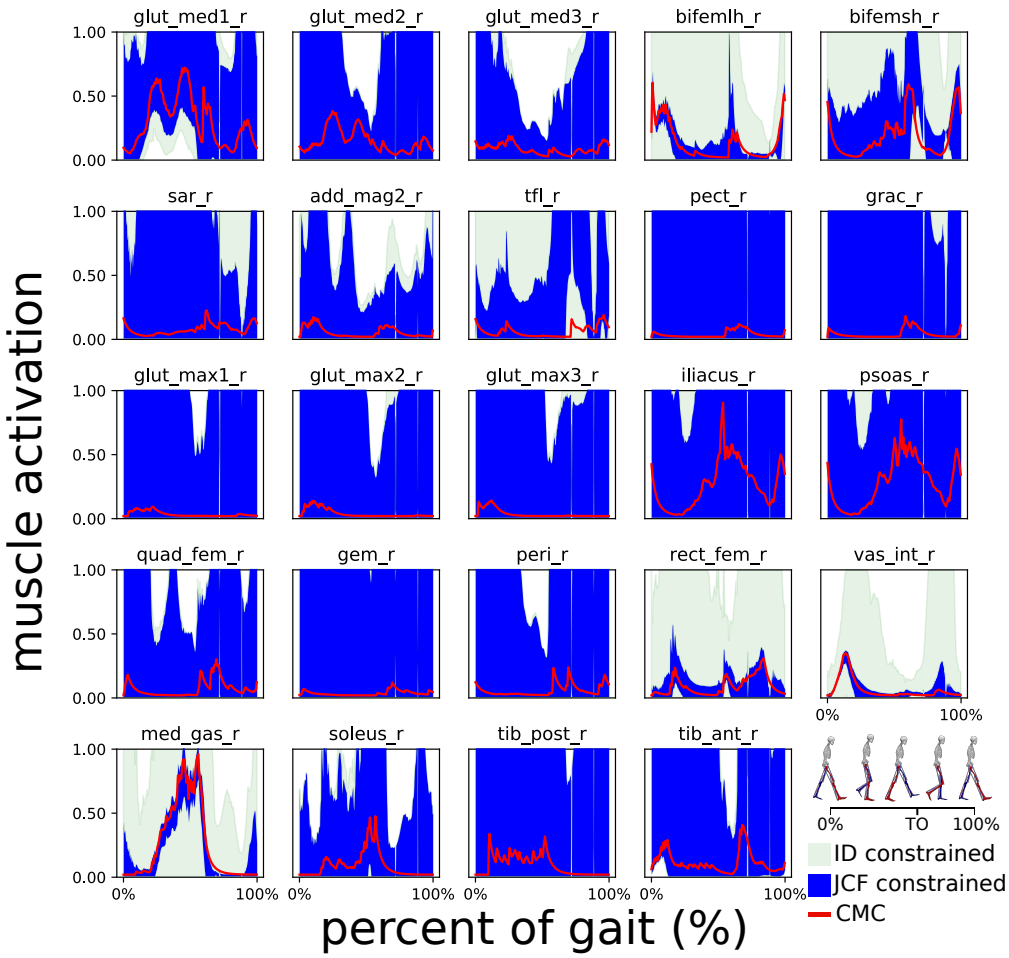


Figure 6.7: Feasible JCF used to constrain the $\mathcal{V}$**-FAS** using the joint contact force expression.

Using the frameworks developed from previous chapters, additional constraints are added by appending them to $\mathcal{H}$-**FAS**. The procedure for constructing equations that describe the relationships between muscle $\boldsymbol{\mathcal{A}}$ and the interstitial forces at a selected joint are described.

6.2.4 Discussion

Filtering feasible activation by the experimental JCF constrains $\mathcal{V}$-**FAS** over the gait cycle. Analytical approximations of joint reactions using muscle lines of action and the body inertial properties were determined for each frame of an ad hoc simulation at run-time.

Muscles apply tensions to their attached bodies along a path known as the line of action which is a function of the geometry and muscle routing, and changes over time as the model performs a kinematic task. These analyses were performed using CMC states; however, the IK solution can also just as easily be used similar to the SO procedure. These analytical approximation of JCF can be obtained at run-time trivially for models that do not have closed loop kinematic chains. Future works should investigate implementing different kinds of joints and various types of actuators that were not investigated in this analysis. JCF constrained FAS bounds OpenSim's CMC solution and highlights that there is great variability of possible $\boldsymbol{\mathcal{A}}$ that may satisfy the JCF requirements of the motion. These results have clinical relevance for the analysis of osteoarthritis in subjects with instrumented joints, and can be used to compare filtered EMG with joint-constrained $\mathcal{V}$-**FAS** bounds and vertices. Additionally, these methods can be easily expanded to investigate other muscle-dependent constraints.

6.3 Tools

For advanced developers, an in-depth exploration of the mathematics and code and paradigms used are outlined in this section (Figure 6.8).

feasibleActivationConstrained.py FAS analysis such that the $\boldsymbol{\mathcal{A}}$ $\mathcal{H}$-representation is bounded by an experimental JCF and the inverse dynamics solution.

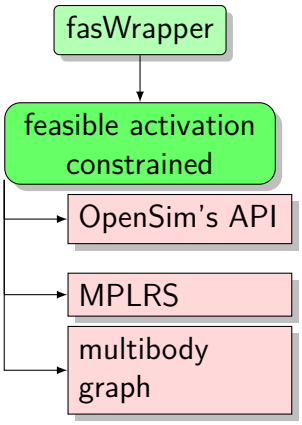


Figure 6.8: Chapter 6 software components.

6.4 Chapter Summary

Procedurally constructed **$\mathcal{H}$-FAS** using any muscle dependent downstream parameters such as JCF constrain FAS significantly and also bounds CMC's solutions within filtered feasible space. Navigating a multibody tree down from the base node (in gait models, this node will be the pelvis) to the location of applied loads and navigating up to where the JCF are applied is employed in feasibleActivationsConstrained.py.

Chapter 7

Windowing for Controls: Feasible Activation Space Trajectories

7.1 Chapter Background

In previous chapters FAS were found that satisfied the ID or JCF constraints necessary to drive the forward dynamic simulation. In this chapter, a high-throughput MCMC approach to computing $\boldsymbol{\mathcal{H}}$-**FAS** over time using only the first order $\boldsymbol{\mathcal{A}}$ dynamics is explored. FAS Trajectories (FAST) is an MCMC method of computed muscle control that exploits the fundamental aspects of convex $\boldsymbol{\mathcal{H}}$-**FAS** as explored in the previous chapters of this research.

7.1.1 Challenges

The CG problems explored in Chapters 5 and 6 produce large high fidelity data sets that researchers can use to construct MCMC by way of arbitrary linear combinations of the $\boldsymbol{\mathcal{A}}$ vertices, but these methods are constrained by the nature of the CG problem. Problems of sufficient dimension (a sufficient number of muscles) have disk space and computational time requirements that exceed the bounds of human existence itself. Until there are breakthroughs in $P = NP$ or quantum computation, it is not really possible to make tools that researchers and clinicians can use with complex models of many muscles. Instead, some conic optimization and MCMC methods are exploited that are designed specifically for the

analysis of polytopes: exactly the objects constructed in Chapters 5 and 6. Chiefly, there are two key problems:

1. how to move inside one $\boldsymbol{\mathcal{H}}$-**FAS** without finding the equivalent $\boldsymbol{\mathcal{V}}$-**FAS**
2. how to move between two $\boldsymbol{\mathcal{H}}$-**FAS** for different discrete times

Definitionally, unless in statics or assuming the exact same model pose with the same applied external loads, $\boldsymbol{\mathcal{A}}$ that produce one $\boldsymbol{\Gamma_{task}}$ for a specific discrete time are never equal to $\boldsymbol{\mathcal{A}}$ that produce another $\boldsymbol{\Gamma_{task}}$ for a different discrete time. This relationship is why looking at the upper and lower bounds of the FAS is so deceiving. Just because the upper and lower bounds plots shown in Chapters 4, 5 and 6 at two distinct time points are the same, it does not imply that combinations of $\boldsymbol{\mathcal{A}}$ that produced a particular movement are the same at all, unless of course, the $\boldsymbol{\mathcal{Q}}, \dot{\boldsymbol{\mathcal{Q}}}, \ddot{\boldsymbol{\mathcal{Q}}}$ and $\boldsymbol{\mathcal{F}_{ext}}$ are exactly the same.

7.1.2 First Order Muscle Activation Dynamics

Before exploring MCMC, a review of first order muscle $\boldsymbol{\mathcal{A}}$ dynamics as outlined in dynamical models written in Winters, and again in Thelen, is in order (Figure 7.1), and then the upper and lower bounds of FAS for a future time frame in a discrete time analysis is derived. A muscle's force comprises potentially three components. An active muscle force is a function of the $\boldsymbol{l_m}$ of a muscle, $\boldsymbol{v_m}$ of a muscle, and the muscle's $\boldsymbol{\mathcal{A}}$ level. The rate of firing neural excitations from the brain are mapped to this excitation level, which reflect the effective Calcium ion level within muscle cells. For OpenSim models, it is typical to use either Thelen or Millard muscle models, and so, this discussion is pertinent to this research. Muscles apply tension to bones along tendons, which attach to bones. Additionally, a muscle also has a passive force which is a function exclusively of the muscle $\boldsymbol{l_m}$s. While important for musculoskeletal modeling, for slow behavioral tasks or the analysis of rigid tendon driven robotic manipulators, the dichotomy between active and passive muscle force can be ignored. The entirety of the length of MTU is a function of the coordinates of the model; however, the individual lengths of the muscle and the tendon are each a function of the $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{v_m}$. For this reason, controllers for muscle tendon actuators in OpenSim rely on using both the muscle $\boldsymbol{\mathcal{A}}$ level and the muscle $\boldsymbol{l_m}$.

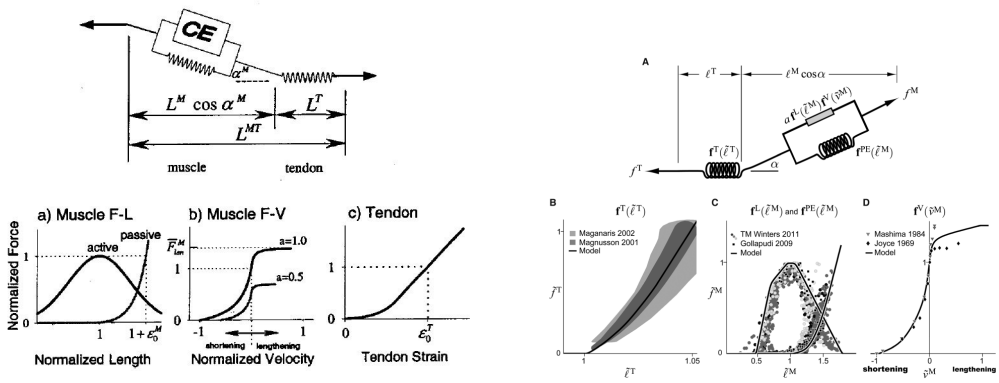


Figure 7.1: Figure from Thelen2003 [96] (left) and Millard2012 [71] (right). Both of these models are very similar equilibrium-type muscle models with the main difference is that Millard2012 has additional parameters and toggles, such as the ability to ignore tendon compliance.

For the Thelen muscle model, muscle force is defined in Equation 3.9.

From Equation 2.13, $\mathrm{f}^L(\boldsymbol{l_m})$ is the gain representing the position of the muscle on the F-L curve, and $\mathrm{f}^V(\boldsymbol{v_m})$ is the position of the muscle on the F-L-V surface and $\mathrm{f}^{PE}(\boldsymbol{l_m})$ is the position of the muscle on the passive force length curve.

These equations of muscle forces should be familiar, but here the first order muscle $\boldsymbol{\mathcal{A}}$ dynamics are analyzed.

$$\frac{da}{dt} = \frac{u - a}{\tau(a, u)} \tag{7.1}$$

$$\tau(a, u) = \begin{matrix} \tau_{act}(0.5 + 1.5a) & : u > a \\ \tau_{deact}/(0.5 + 1.5a) & : u \leq a \end{matrix} \tag{7.2}$$

Where u is the excitation, which reflects the map to the muscle activation a, $u \rightarrow a$ occurs via the accumulation of Calcium ions in the muscle. Upper and lower bounds on $\boldsymbol{\mathcal{A}}$ in the next discrete time frame are now derived.

Let $u_{lb} = 0$ and $u_{ub} = 1$ be the upper bound and lower bounds on muscle excitation. Muscles can be maximally or minimally excited by substituting these bounds to Equation 7.2.

Lower Bound:

$$\tau(a,0) = \begin{matrix} \tau_{act}(0.5+1.5a) & : 0 > a \\ \tau_{deact}/(0.5+1.5a) & : 0 \leq a \end{matrix} \tag{7.3}$$

$0 < a$ so from Equation 7.3, it is possible to derive (Equation 7.4):

$$\tau(a,0) = \tau_{deact}/(0.5+1.5a) \tag{7.4}$$

Upper Bound:

$$\tau(a,1) = \begin{matrix} \tau_{act}(0.5+1.5a) & : 1 > a \\ \tau_{deact}/(0.5+1.5a) & : 1 \leq a \end{matrix} \tag{7.5}$$

Take the first conditional of Equation 7.5 to be always true (Equation 7.6),

$$\tau(a,1) = \tau_{act}(0.5+1.5a) \tag{7.6}$$

Now to compute upper and lower bounds on $\boldsymbol{\mathcal{A}}$, a_{lb} and a_{ub} from the Equation 7.1:

Let activations vector $\boldsymbol{a}$ be constructed from the set $\boldsymbol{\mathcal{A}}$ and let $\boldsymbol{u}$ be the vector of neural excitations for each muscle and let Δt be the frame time for a simulation of n discrete time frames:

Lower Bound:

Let $\boldsymbol{u} = \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix}$ and let $\boldsymbol{a} = \begin{bmatrix} a_1 & a_2 & \dots & a_m \end{bmatrix}$.

$$\frac{\boldsymbol{da}}{\boldsymbol{dt}}_{lb} = \frac{\boldsymbol{u}-\boldsymbol{a}}{\tau(\boldsymbol{a},\boldsymbol{u})} = \frac{-\boldsymbol{a}(0.5+1.5\boldsymbol{a})}{\tau_{deact}} = \frac{(-0.5\boldsymbol{a}-1.5\boldsymbol{a}^2)}{\tau_{deact}} \tag{7.7}$$

Equation 7.7 involves finite differences to relate back to activations so:

$$\boldsymbol{a}_{lb} = \boldsymbol{a} + \Delta t \frac{\boldsymbol{da}}{\boldsymbol{dt}}_{lb} = \boldsymbol{a} + \Delta t \frac{(-0.5\boldsymbol{a}-1.5\boldsymbol{a}^2)}{\tau_{deact}} \tag{7.8}$$

Upper Bound:

Let $\boldsymbol{u} = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}$ and let $\boldsymbol{a} = \begin{bmatrix} a_1 & a_2 & \dots & a_m \end{bmatrix}$.

$$\frac{\boldsymbol{da}}{\boldsymbol{dt}}_{ub} = \frac{\boldsymbol{u} - \boldsymbol{a}}{\tau(\boldsymbol{a}, \boldsymbol{u})} = (\boldsymbol{u} - \boldsymbol{a})\tau_{act}(0.5 + 1.5\boldsymbol{a}) = \tau_{act}(.5\boldsymbol{u} + \boldsymbol{a} - 1.5\boldsymbol{a}^2) \tag{7.9}$$

Equation 7.9 can be written as Equation 7.10, similarly to Equation 7.8:

$$\boldsymbol{a}_{ub} = \boldsymbol{a} + \Delta t \frac{\boldsymbol{da}}{\boldsymbol{dt}}_{ub} = \boldsymbol{a} + \Delta t \tau_{act}(.5\boldsymbol{u} + \boldsymbol{a} - 1.5\boldsymbol{a}^2) \tag{7.10}$$

The beautiful thing about working with halfspaces is that hypercube bounds can be changed really easily. Remember the halfspace representation from Chapter 5 where at the end of $\boldsymbol{\mathcal{H}}$**-FAS** is the matrix (Equation 5.3).

This halfspace representation represents a unit hypercube in $\mathbb{R}^n$ for a system of n muscles. To change $\boldsymbol{\mathcal{H}}$**-FAS** to reflect the first order dynamics muscle $\boldsymbol{\mathcal{A}}$ bounds:

$$H_{\Delta a \Delta t} = \begin{bmatrix} -a_{lb}^1 & 1 & 0 & \dots & 0 \\ -a_{lb}^2 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{lb}^m & 0 & 0 & \dots & 1 \\ a_{ub}^1 & -1 & 0 & \dots & 0 \\ a_{ub}^2 & 0 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{ub}^m & 0 & 0 & \dots & -1 \end{bmatrix} \tag{7.11}$$

The upper and lower bounds of the future time frame (Equations 7.7, 7.9) and this new $H_{\Delta a \Delta t}$ (Equation 7.11) is appended to the ID constraint H or similarly any other procedurally constructed muscle dependent constraint. Ideally there would be some analytical/ functional approach for determining the feasible muscle $\boldsymbol{\mathcal{A}}$. Definitionally, the sets of $\boldsymbol{\mathcal{A}}$ that satisfy the ID constraint for one time frame can never equal the sets of feasible $\boldsymbol{\mathcal{A}}$ that satisfy the ID constraints for another time frame unless the ID constraint itself is equal between time frames and the model assumes the same pose.

7.1.3 Application of the Bounds on Muscle Activation to CG

Previously in the literature, the Hit-and-Run (HAR) algorithm has been proposed as a tool that can be used to sample $\boldsymbol{\mathcal{H}}$**-FAS**; however, HAR suffers in the dynamic case as ID-constrained $\boldsymbol{\mathcal{H}}$**-FAS** are thin and HAR often gets trapped locally within the $\boldsymbol{\mathcal{V}}$**-FAS** without rescaling operations [24].

Concepts of polytope centers, HAR, and a polytope-focused MCMC method, the Dikin Walk (DW), are explored as methods of tunneling for feasible $\boldsymbol{\mathcal{A}}$ and getting insights into the structure of feasible space.

7.1.4 Minimal Activation

Minimizing $\|a\|$ is similar to approaches used in SO and CMC, but both SO and CMC use a quadratic objective instead of the simple linear objective. $\boldsymbol{\mathcal{V}}$**-FAS** encompasses a wide assortment of possible solutions that satisfy the dynamic task; however, approaches like CMC arrive at singular solutions with stringent constraints on how the objective can be changed, making it challenging or even impossible to construct models of impaired control. [1]

7.1.5 "Centers" of FAS

The fortunate aspect of working with convex sets is that the mean of the vertices of the polytope is an interior point. [2]

Axiom 1. *The vertex center of a convex* $\boldsymbol{\mathcal{H}}$***-FAS*** *is the average of the vertices of that polytope. If* v *is the number of vertices of the* $\boldsymbol{\mathcal{V}}$ *of an equivalent* $\boldsymbol{\mathcal{H}}$***-FAS****:*

$$a_{vc} = \frac{1}{v}\sum_{i=0}^{v} a_i$$

Axiom 1 is the least useful to us as it obligates us to perform CG from the $\boldsymbol{\mathcal{H}}$**-FAS** from Chapters 5 and 6. Clinicians and researchers want fast methods, and they cannot wait hours

[1] Perhaps definitionally, impaired control is not optimal.

[2] If this relationship is not immediately obvious, just average the vertices of a square, or the ends of a line segment or the corners of a cube.

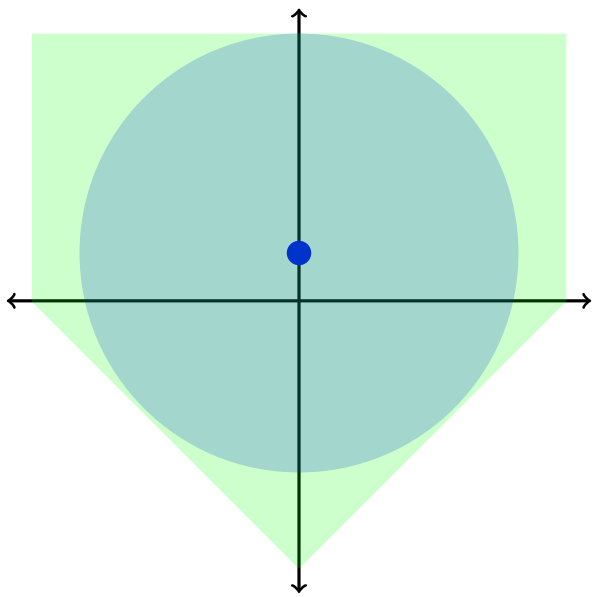


Figure 7.2: highlighting the a_{cc}, the center of the largest hypersphere that fits within $\boldsymbol{\mathcal{H}}$**-FAS**.

to perform CG. Additionally, the vertex center need not be the optimal starting position. These ideas of centers are explored in Section 2.2.4.

To that aim, there is another definition of a "central point" that lies within $\boldsymbol{\mathcal{V}}$**-FAS**: the Chebyshev center a_{cc}. Every $\boldsymbol{\mathcal{H}}$**-FAS** has at least one largest possible hypersphere that may lay within it. It has a radius r and has a feasible location a_{cc}.

Axiom 2. *a_{cc} of $\boldsymbol{\mathcal{H}}$-FAS is the point that satisfies the following LP: Inject an additional variable r which is the length of the radius of the largest euclidean ball that fits within the $\boldsymbol{\mathcal{H}}$-FAS.*

$$\max r$$

$$s.t. \quad b_i - A_i x - \|A_i\| r \geq 0$$

Axiom 2 has more utility than the vertex center (Figure 7.2) since it can be computed in reasonable time using LP, but the $\boldsymbol{\mathcal{H}}$**-FAS** used in feasible muscle sets analysis are "thin": relaxed from a stringent equality constraint only by a factor ϵ, so these $\boldsymbol{\mathcal{H}}$**-FAS** technically have many possible solutions that satisfy definitionally for a_{cc}.

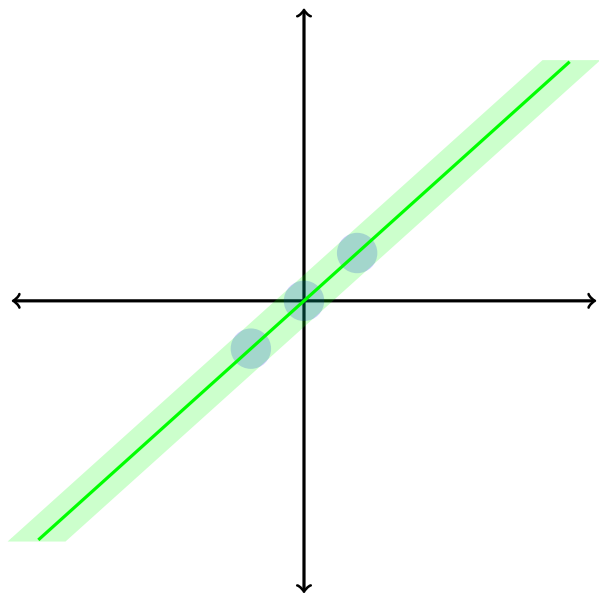


Figure 7.3: a_{cc} can often be a bad estimate of the center of $\boldsymbol{\mathcal{H}}$**-FAS**, particularly when the polytope is "thin".

It is not uncommon to compute the a_{cc} of a thin $\boldsymbol{\mathcal{H}}$**-FAS** only to arrive at a relative extremum within $\boldsymbol{\mathcal{V}}$**-FAS** as opposed to warm starts (Figure 7.3).

For each $\boldsymbol{\mathcal{H}}$**-FAS**, there is a unique analytical center a_{ac}. a_{ac} is distinct from the vertex center described in Axiom 1. One $\boldsymbol{\mathcal{V}}$**-FAS** may have many equivalent $\boldsymbol{\mathcal{H}}$**-FAS** due to constraint redundancy. a_{ac} is sensitive to redundancy of the constraints applied to the system. Therefore, many $\boldsymbol{\mathcal{H}}$**-FAS** that have the same $\boldsymbol{\mathcal{V}}$**-FAS** representation, and thus, the same vertex center, can have different a_{ac}. In statics and dynamics, this distinction between the analytical and vertex centers is akin to the distinction between the geometric centroid and the center of mass of a body. The a_{ac} of $\boldsymbol{\mathcal{H}}$**-FAS** can be found by performing a maximization on what is known as the logarithmic potential function P_x.

Axiom 3. *a_{ac} of $\boldsymbol{\mathcal{H}}$-FAS is the point that satisfies the following CP:*

$$\max_{a} \quad \sum_{i=1}^{m} \log\left(b_i - A_i^T a\right)$$

Similarly a weighted a_{ac} can be found for a given set of weights w_i:

$$\max_{a} \quad \sum_{i=1}^{m} w_i \log\left(b_i - A_i^T a\right)$$

The log-sum expression in Axiom 3 is a structure that proves useful later in sampling $\boldsymbol{\mathcal{H}}$-**FAS**. This expression is known as a type of barrier function that maps the matrix $\boldsymbol{\mathcal{H}}$-**FAS** into a single function.

Lemma 7.1. *a_{ac} can be approximated by newton's iterative method [82] with the newton direction as*

$$\begin{aligned} & \delta_{nt} = (A^T S^{-2} A)^{-1} A^T y \\ s.t. \quad & S = diag(\frac{1}{y}) \\ & y_i = b_i - A_i a_i \end{aligned}$$

Lemma 7.2. *When sampling from an $\boldsymbol{\mathcal{H}}$-**FAS** with a uniform distribution, for $\mathcal{N} \to \infty$, the mean of sample points approaches the a_{ac}.*

If the CG for an $\boldsymbol{\mathcal{H}}$-**FAS** is already performed, from Lemma 7.1, to investigate the most likely $\boldsymbol{\mathcal{A}}$, a fair central point to use is a_{vc} of the $\boldsymbol{\mathcal{H}}$-**FAS**.

Lemma 7.3. *For $\mathcal{N} \to \infty$, the samples points distribution approaches the distribution that gains vector $\boldsymbol{g}$ was selected from.*

Checking if $\boldsymbol{p}$ is an interior point of $\boldsymbol{\mathcal{H}}$-**FAS** is trivial: just verify if every inequality of $\boldsymbol{\mathcal{H}}$-**FAS** is satisfied for the point $\boldsymbol{p}$. If any number inequalities fail to hold, then p cannot be within this $\boldsymbol{\mathcal{H}}$-**FAS**.

7.1.6 Hit-and-Run (HAR)

In the static domain, where there is a wide range on the constraint, HAR is a fantastic tool for sampling $\boldsymbol{\mathcal{V}}$-**FAS**. Remember that the halfspace representation of $\boldsymbol{\mathcal{H}}$-**FAS** is just the

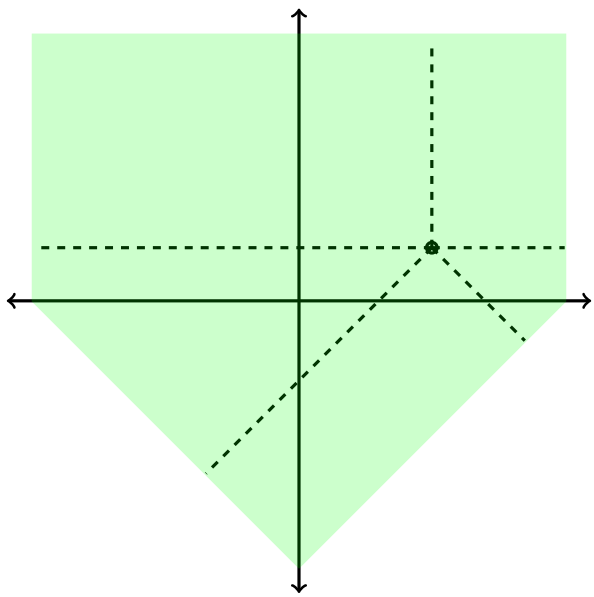


Figure 7.4: Start at an interior point and find the λ to each halfspace.

intersection of several halfspaces instead of the vertices and that a halfspace is a subspace of $\mathbb{R}^n$ created when the space is bisected $\mathbb{R}^n$ by a hyperplane [2, 65, 66]. For some arbitrary point $\boldsymbol{p}$ within FAS, a vector of the shortest euclidean distances from this point $\boldsymbol{p}$, to each of the hyperplanes that construct an $\boldsymbol{\mathcal{H}}$**-FAS** is constructed. The points on the halfspaces that form FAS from $\boldsymbol{p}$ along these shortest euclidean distances may or may not necessarily be within the $\boldsymbol{\mathcal{H}}$**-FAS**. It is also possible to project these along an arbitrary direction to arrive at the shortest euclidean distances along the current direction to arrive at a given halfspace, known as λ_i for each A_i. A negative euclidean norm signifies that the halfspace is in the opposite direction from the supplied unitary direction vector.

Each halfspace is dual to the normal vectors from that halfspace, and there exist normal vector from the halfspaces that cross the point $\boldsymbol{p}$ (Figure 7.4).

For a given hyper-spherical unitary direction, it is possible to find the distances λ_i from the point $\boldsymbol{p}$ to each of the halfspaces that comprise A (Figure 7.5). The shortest positive and negative λ can be used to select a new point along the unitary direction that definitionally must lie within $\boldsymbol{\mathcal{V}}$**-FAS** (Figure 7.6). This method is repeated as necessary. As per Lemmas

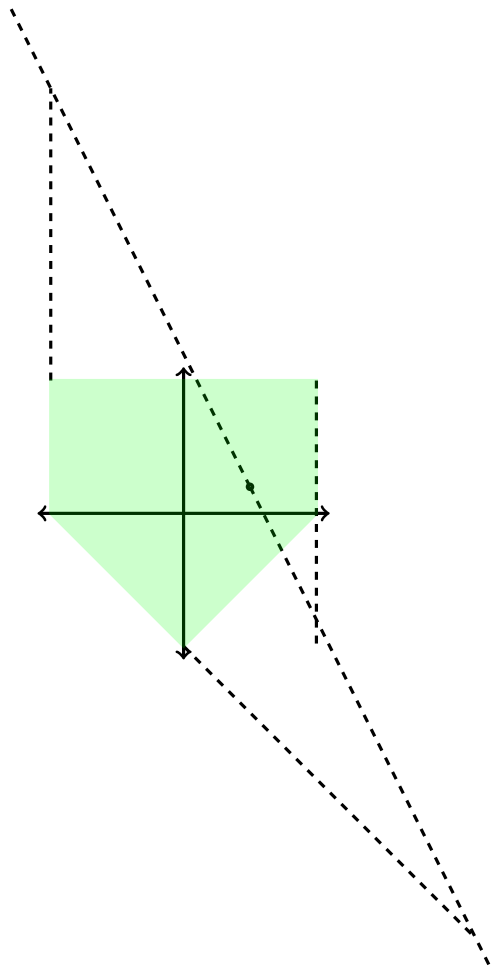


Figure 7.5: Pick a unitary direction from the standard Gaussian $\boldsymbol{g}$.

7.2 and 7.3, enough sample points on $\boldsymbol{\mathcal{V}}$**-FAS** interiors will lead the sample distribution toward the sampling distribution.

For some gains vector $\boldsymbol{g}$ and a set of vectors $\boldsymbol{v}_{aux}$ following the shortest euclidean distance from an arbitrary point $\boldsymbol{p}$. In the case that $\boldsymbol{\mathcal{H}}$**-FAS** is not non-redundant, an additional step is performed to check if a point $\boldsymbol{p} + \boldsymbol{g} \odot \boldsymbol{v}$ is within $\boldsymbol{\mathcal{H}}$**-FAS**, and if it is outside, then do not update $\boldsymbol{p}$. By simply filtering the upper and lower bounds on λ to the smallest positive and smallest negative, it is possible to solve this issue; however, ideally, it would be beneficial to perform a redundancy removal on the $\boldsymbol{\mathcal{H}}$**-FAS** to remove redundant inequalities. In the dynamic case, HAR (Algorithm 6) suffers because the iteration gets trapped locally as it hits the sides of $\boldsymbol{\mathcal{V}}$**-FAS**. There are scaling methods for dealing with this issue, but instead an

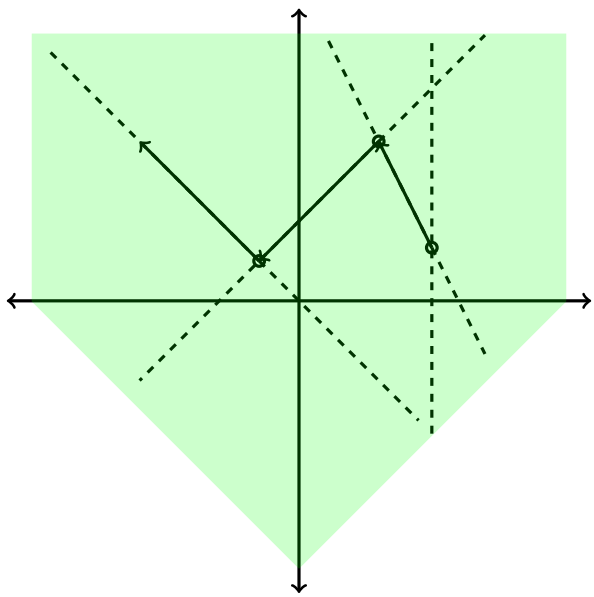


Figure 7.6: Iteratively select points that lay within $\mathcal{V}$**-FAS**.

adaptation of an MCMC method that is specialized for dealing with polytopes as opposed to purely linear systems was developed: the Dikin walk.

7.1.7 Dikin Walk (DW)

In the iterative Newtons approximation for the analytical center, $(A^T S^{-2} A)^{-1}$ is an inverse Hessian matrix. As the Jacobian is a matrix of first order partials, the Hessian is a matrix of second order partials. The Hessian matrix (Equation 7.12) is just the Jacobian of the gradient of a function: $H = J(\nabla f)$.

$$H = \begin{bmatrix} \frac{\delta^2 f}{\delta a_1^2} & \frac{\delta^2 f}{\delta a_1 \delta a_2} & \cdots & \frac{\delta^2 f}{\delta a_1 \delta a_m} \\ \frac{\delta^2 f}{\delta a_2 \delta a_1} & \frac{\delta^2 f}{\delta a_2^2} & \cdots & \frac{\delta^2 f}{\delta a_2 \delta a_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta^2 f}{\delta a_m \delta a_1} & \frac{\delta^2 f}{\delta a_m \delta a_2} & \cdots & \frac{\delta^2 f}{\delta a_m^2} \end{bmatrix} \tag{7.12}$$

Additionally, the concept of barrier functions were touched on in this chapter when solving for the analytical center through the conic maximization of the log potential, using

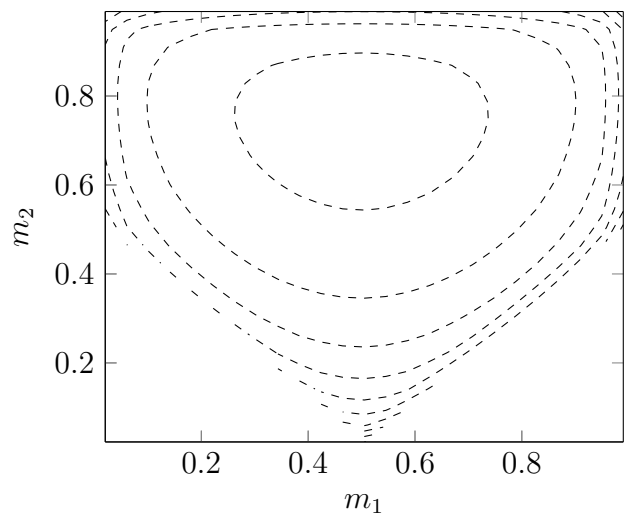


Figure 7.7: the log barrier of the "home plate"-shaped $\boldsymbol{\mathcal{H}}$**-FAS** shown many times previously: $z = \sum_{i=1}^{m} \log\left(b_i - A_i^T a\right)$

the logical extension of the Lemma 7.2 (Figure 7.7), a complicated system of many linear constraints equations can be formulated, visualized, and optimized as a single function.

In the thin $\boldsymbol{\mathcal{V}}$**-FAS** produced in the dynamically constrained problem and the joint contact force constrained problem, the HAR procedure tends to get trapped locally as it bounces on the edges of a thin $\boldsymbol{\mathcal{V}}$**-FAS** and succumbs under ill-conditioned Jacobian matrix for $\mathbb{O}$ constraints: an unfortunate event that happens extremely often for biological systems.

The concept of the Chebyshev center and the largest possible euclidean ball that may fit within a $\boldsymbol{\mathcal{H}}$**-FAS** can be extrapolated to ellipsoids called John ellipsoids.

A special John ellipsoid can be found by computing the Hessian of the log barrier function used in the conic maximization: it is possible to arrive at an expression for an ellipsoid known as the Dikin ellipsoid. The Dikin ellipsoid is the central feature of the Dikin walk [34]: to iteratively compute these and to select a new sample from the ellipsoid interior to use in the computation of the future frame Hessian. While the HAR algorithm is dependent on the condition number of $\boldsymbol{\mathcal{H}}$**-FAS**, the Dikin walk is affine-invariant: the mixing time and the selection of sufficiently different samples do not depend on features typical in multibody dynamics like kinematic singularities [21]. Fortunately, this Hessian is symmetric and positive-definite, which simplifies inverse calculation.

In the standard Dikin walk, the future frame u is called a proposal vector which is chosen at random from a uniform distribution centered on the current $\boldsymbol{\mathcal{A}}$, a, and where $D_a = \nabla^2 \mathcal{F}(a)$ is the Hessian of the log barrier (Equation 7.13):

$$\{u \in \mathbb{R}^m | (u-a)^T D_a (u-a) \leq R\} \tag{7.13}$$

There is a Dikin walk variant where instead of the uniform distribution, the proposal vector is chosen from a Gaussian with the covariance determined from the Dikin ellipsoid, which was has been shown to produce proposals with a high probability of landing in $\boldsymbol{\mathcal{V}}$**-FAS** [83, 51]. In the next section, a novel variant of the Dikin walk where proposals are drawn from a multivariate skew normal instead of the Gaussian distribution to preferably weight deactivation future states is explored.

7.1.8 Modified Dikin with Multivariate Skew Normal

An unfortunate consequence of using the first order $\boldsymbol{\mathcal{A}}$ dynamics to find a_{ub} and a_{lb} is that if the standard Gaussian is used to sample $\boldsymbol{\mathcal{H}}$**-FAS**, $\boldsymbol{\mathcal{A}}$ will tend up toward maximal excitation under typical physiological selection of muscle $\tau_{act} = .01$ and $\tau_{deact} = .04$. In order to generate curves that are similar to CMC, a modification of the Dikin walk is offered so that the proposal vectors are drawn instead of a Gaussian distribution (Figure 7.8), a skewed multivariate normal distribution that prefers proposal vectors that tend toward more equally weighted activation/deactivation from the previous discrete time. And these two features, that the future frame $\boldsymbol{\mathcal{A}}$ are bounded by the upper and lower bounds from first order $\boldsymbol{\mathcal{A}}$ dynamics expressions explored earlier in this chapter and Dikin walk proposals are drawn from the multivariate skew normal are enough to produce CMC-like plots with variance determined from the skew factors (Figure 7.9). The skew Gaussian allows for modifying the likelihood of activation or deactivation relative to the boundaries of FAS; however, there is no closed form solution for the shape parameter α. A proof of how the skewness is estimated and then used to compute a maximum likelihood-based estimate of α is provided.

The multivariate skew normal can be written in the form (Equations 7.14, 7.15, 7.16):

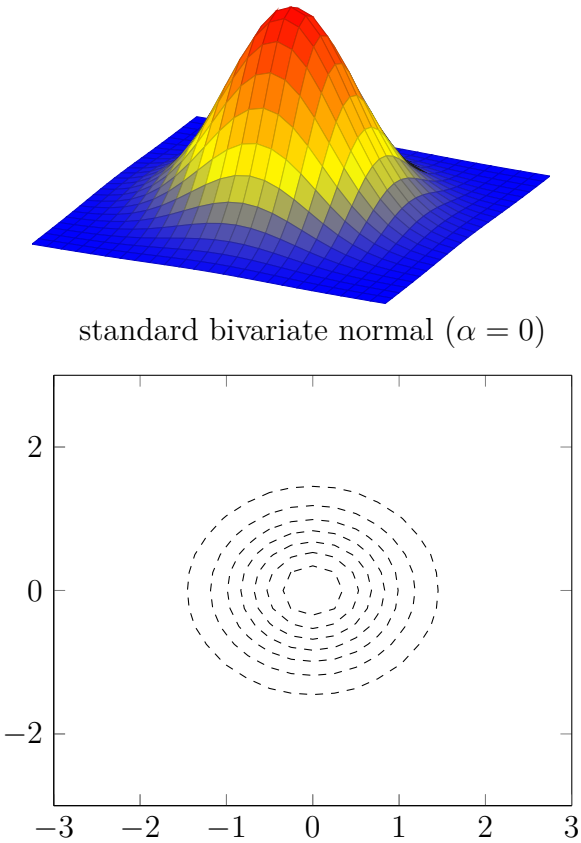


Figure 7.8: Two views of the bivariate standard Gaussian.

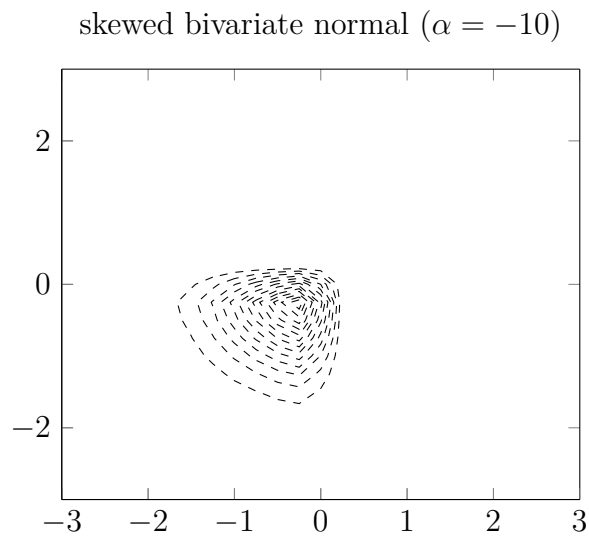


Figure 7.9: A multivariate skew normal with $\alpha = 10$ for both a_1 and a_2.

$$f(a) = 2\phi(a)\Phi(\alpha a) \tag{7.14}$$

Where $\Phi(a)$ is the cumulative distribution function:

$$\Phi(a) = \frac{1}{2}(1 + \text{erf}(\frac{\alpha a}{\sqrt{2}})) \tag{7.15}$$

And $\phi(a)$ is the probability density function:

$$\phi(a) = \frac{1}{\sqrt{2\pi}} e^{-\frac{a^2}{2}} \tag{7.16}$$

$\text{erf}(a)$ is also known as the Gaussian error function [112].

The Bowley skewness of a distribution is a function of the quartiles (Equation 7.17):

$$\gamma = \frac{Q_3 + Q_1 - 2Q_2}{Q_3 - Q_1} \tag{7.17}$$

The quartiles from the median a_{t-1} are selected (Equation 7.18):

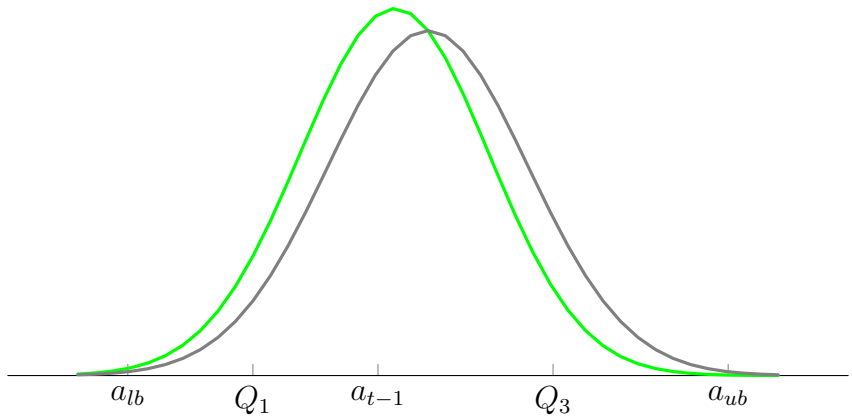


Figure 7.10: The euclidean distances from the upper and lower bounds to a_{t-1} is asymmetrical so the expected value sampling from the standard normal favors activation over deactivation. It is possible to trivially estimate the quartiles as functions of the euclidean distance from a_{t-1} to a_{lb} and a_{ub} and apply this method in a correction (Equation 7.20).

$$\begin{aligned} Q_1 &= a_{t-1} + \tfrac{.67\Delta t \tau_{deact}}{2} \\ Q_2 &= a_{t-1} \\ Q_3 &= a_{t-1} + \tfrac{.67\Delta t \tau_{act}}{2} \end{aligned} \tag{7.18}$$

In Equation 7.18, .67 is the z-statistic for Q_1 and the sample probability distribution quartiles are estimated on each side. FAST employs Equation 7.19 to estimate the shape factor α via the method of moments (Figure 7.10). Using the Bowley skewness estimate, it is possible to compute α by inverting the skewness function:

$$|\delta| = \sqrt{\frac{\pi}{2} \frac{|\gamma|^{\frac{2}{3}}}{|\gamma|^{\frac{2}{3}} + (\frac{4-\pi}{2})^{\frac{2}{3}}}} \tag{7.19}$$

and finally:

$$\hat{\alpha} = \frac{\delta}{\sqrt{1-\delta^2}} \tag{7.20}$$

Note that the domain of $|\delta|$ (Equation 7.20) is between -1 and 1 so near extreme values for $\boldsymbol{\mathcal{A}}$, poor skewness might be computed, but generally $\boldsymbol{\mathcal{A}}$ that tend to hug 0.0 or 1.0 indicate a poor model fit by way of muscle weakness or high reserves, so this estimate should be fine

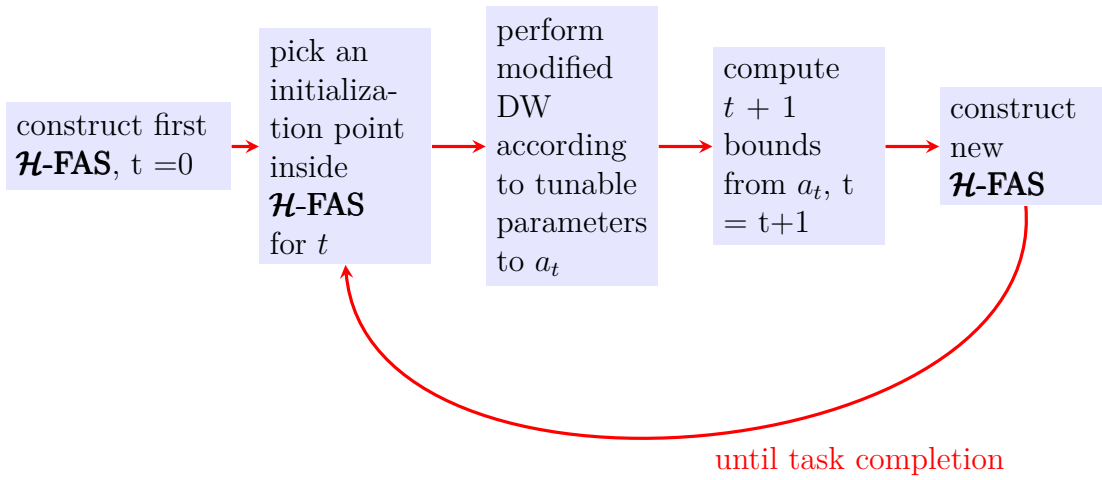


Figure 7.11: A flowchart of the FAST analysis is offered. Parameters can be tuned (Table 7.1) in order to find families of solutions which more closely match CMC (Figure 7.12), or they can be tuned to model abnormal control.

for the purposes of computed control. Now some FAST analysis is performed on the 3 models previously seen through this dissertation.

7.2 Study: Feasible Activation Space Trajectories (FAST)

7.2.1 Case Study Motivation

FAST is a probabilistic alternative to CMC or SO. Instead of minimizing an objective function, FAST leverages the $\mathcal{H}$-FAS which wholly defines the relationship between $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\Gamma_{task}}$, and exploits the geometry of $\mathcal{H}$-FAS without performing the vertex enumeration.

7.2.2 Methods

FAST optimizes a CP in each discrete time frame of a kinematic task to solve for a_{ac} within FAS as one possible choice of initialization parameters; however, several choices of starting positions may be used which will influence the starting distributions of output curves (Table 7.1). Then, a DW variant is performed using a multivariate skew normal based on the bounds

Table 7.1: List of the tunable parameters featured in the FAST analysis.

parameter	**influence**
initialization	influences starting distributions (SO, CMC, a_{ac},a_{cc},a_{min},a_{max})
dikin walk radius r	solution variation from initialization
dikin walk # steps	solution variation from initialization,increases computational time
shape parameter α	soft objective, likelihood of cocontraction or deactivation

determined from first order $\boldsymbol{\mathcal{A}}$ dynamics and the last time frame activation selection as per algorithm 7. Additionally, a high level flowchart of the FAST analysis is offered (Figure 7.11).

For each frame of the kinematic task, $\boldsymbol{\mathcal{H}}$**-FAS** is formed and then used to compute the Hessian $\mathcal{D}_a = \nabla \mathcal{F}_a$.

Then a new point is selected from the ellipsoid $\{u \in R^d | (u-a)^T \mathcal{D}_a (u-a) \leq R\}$according to a skew normal Gaussian $\boldsymbol{g}(z)$ where the skew parameter is estimated using Bowley's skewness estimate. First, a parameter exploration of the influence of α on the output trajectories is shown. Then a case study is provided that finds $\boldsymbol{\mathcal{H}}$**-FAS** for each discrete time of a kinematic trajectory for gait which is then constrained using the FAST procedure for a 10 DOF 18 muscles gait model freely available and packaged with OpenSim. Similarly, $\boldsymbol{\mathcal{H}}$**-FAS** is constrained for a more complicated 23 DOF 54 muscles gait model also packaged with OpenSim. Finally, $\boldsymbol{\mathcal{H}}$**-FAS** constrained along with the joint loading constraint is used to constrain FAS trajectories.

7.2.3 Results

The parameter α can be considered a "soft" objective (Figure 7.12), it essentially increases the likelihood of observing samples with co-contraction of muscles during FAST.

Additionally, increased negative skew decreases the likelihood of co-contraction and tends toward the optimal CMC/SO solution.

FAST analysis of the Gait1018 model (Figure 7.13) produced curves that match well with CMC for some muscles, particularly gastrocnemius and vastus. These results indicate

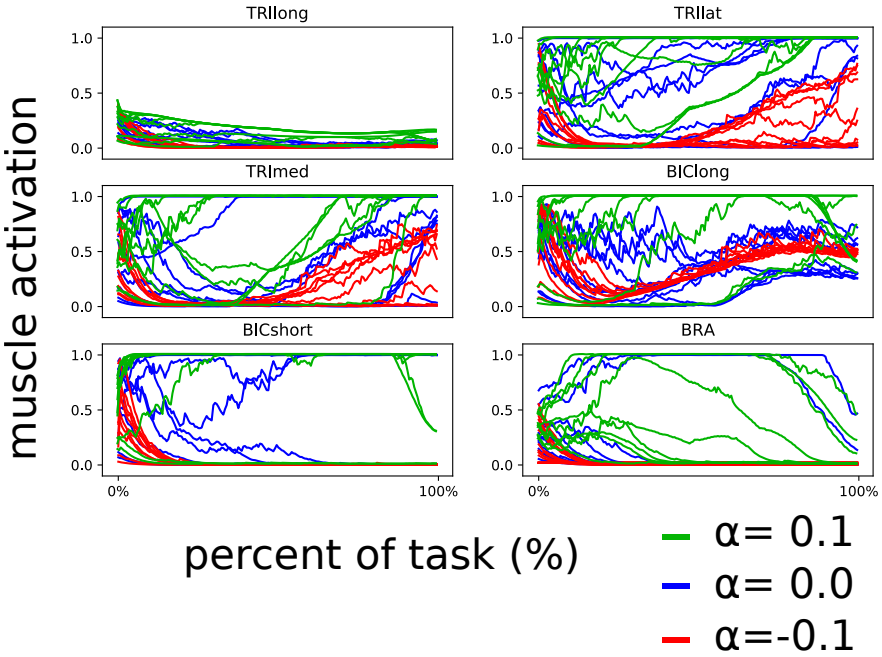


Figure 7.12: An investigation of how different base skew parameters on proposal distributions influence the output trajectories. 10 sample trajectories were drawn for each level. All the FAS trajectories depicted in the plot satisfy the ID determined constraints to within tolerance and additionally satisfy the first order $\boldsymbol{\mathcal{A}}$ dynamics determined window at each time point.

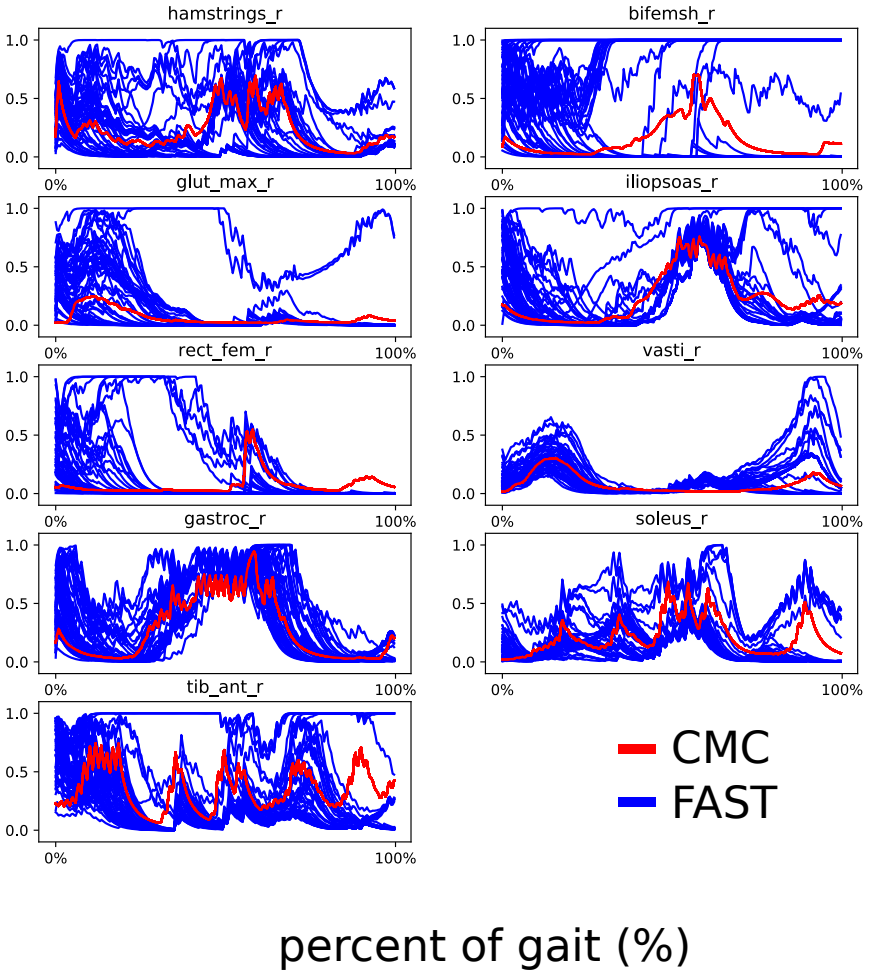


Figure 7.13: 100 representative samples from FAS. FAST samples lay within the feasible bounds dictated in Chapter 5.

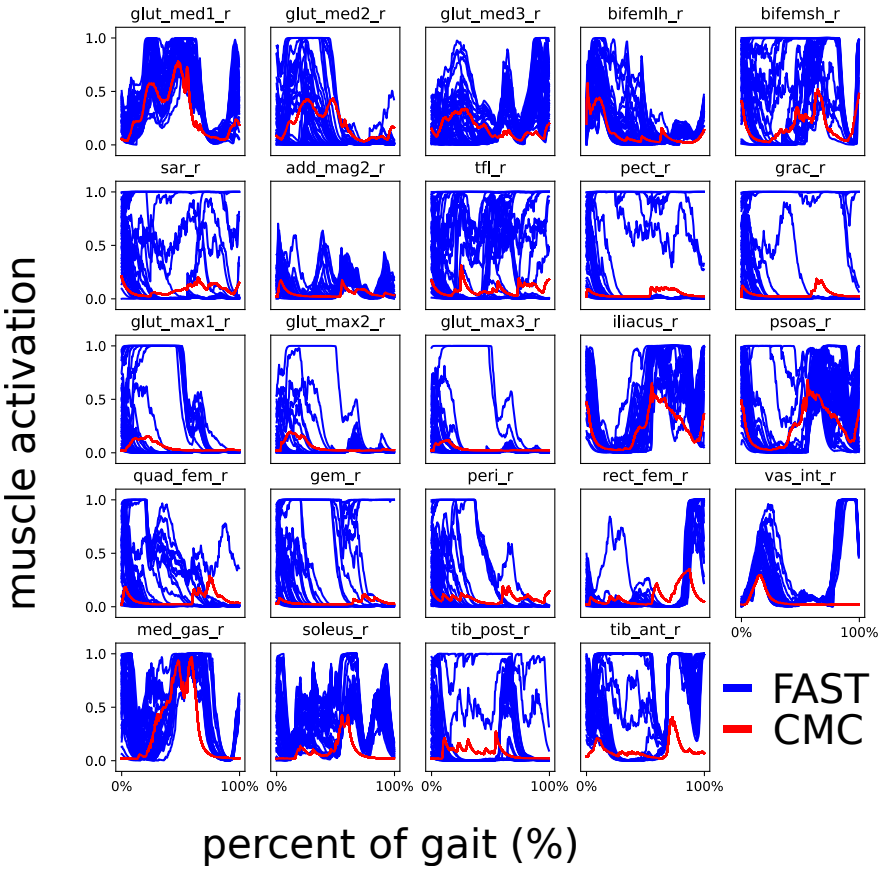


Figure 7.14: 50 representative samples from FAST analysis 2354. Again, FAST samples lay within the feasible bounds dictated in Chapter 5.

that the structure of feasible control space is much more organized than previously depicted when the muscle activation dynamics are used to muscle activations together in time [108].

Moving to the more complicated Gait2354 model, some muscles align very well with the CMC solution (Figure 7.14). Noting the high activation of vastus intermedius after toe off, future work should investigate modifying the skew normal distribution using a scaling method such that muscles are weighted by some combination of moment arms and muscle peak isometric forces in order to preferably weight other knee extensors.

By tuning the parameters (Table 7.1), it should be possible to find the families of solutions that lay closer to CMC's choices for controls; however, the intention of FAST is not to be a Las Vegas method of finding the CMC solution, but to provide variability in output solutions

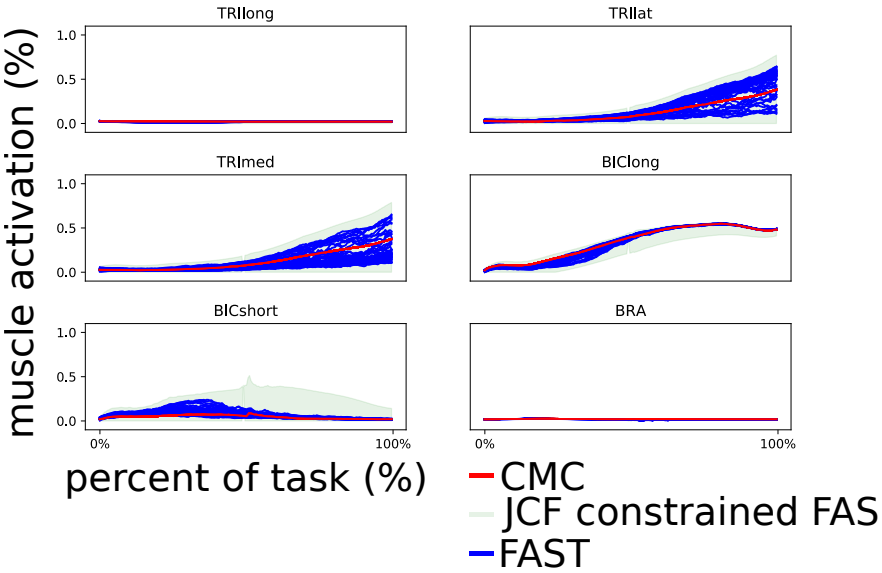


Figure 7.15: FAST analysis of the Arm26 model, feasible activations are selected from the joint force constrained FAS.

along with tunable parameters in order to model abnormal control in a way that CMC and SO can not.

Instead of the ID constraint, it is also possible to use the analytical elbow contact forces developed in Chapter 6 (Figure 7.15). Feasible activations for biceps brachii long follow more closely to the CMC solution thanks to constraining activations by the joint contact forces derived by joint reaction analysis. Constraining by the joint loading constrains FAST solutions to more closely match the CMC solution. Some muscles, most notably triceps long head, brachioradialis and biceps long head, tended to produce activation curves that matched very closely to the CMC solution.

FAST analysis of the Gait1018 model constrained by the JCF (Figure 7.16) was also performed. All FAST trajectories are bounded by the upper and lower bounds determined using the methodologies from Chapters 5 and 6. Using only a Gaussian controls process along with the procedurally generated constraint, it is possible to extract sets of controls that are similar to CMC with variation. FAST is extremely powerful as every single FAST analysis output definitionally satisfies the ID requirements of the task without the use of controllers.

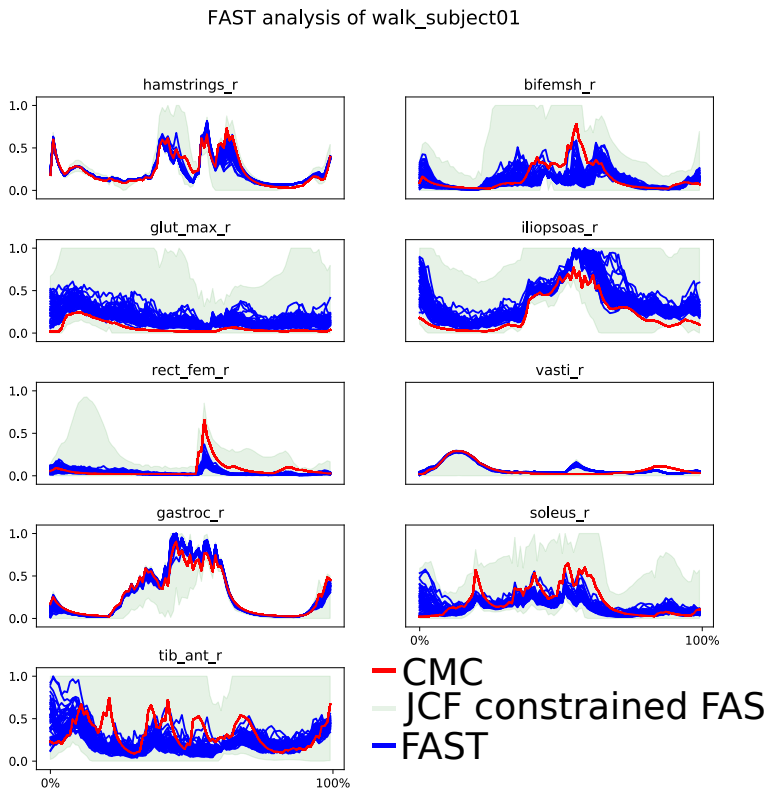


Figure 7.16: FAST analysis of the gait1018 model using the joint loading constraint developed in Chapter 6, feasible activations are selected from the joint force constrained FAS.

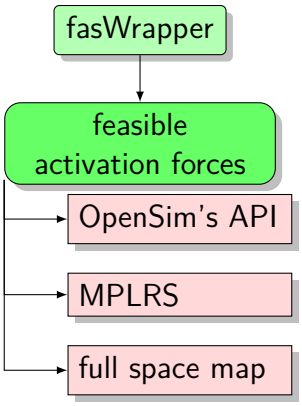


Figure 7.17: Chapter 7 software components.

7.2.4 Discussion

FAST allows researchers to make computed controls that satisfy the $\boldsymbol{\Gamma}_{\boldsymbol{task}}$ constraints similarly to SO or CMC without relying on the quadratic objective or the use of controllers, instead FAST uses a skew normal multivariate Gaussian that is weighted based on the last time frame activations. Whereas CMC or SO are unable to model neural impairment because of adherence to optimization objectives, FAST analysis allows users methods of investigating suboptimal control perceived in crouched gait, cerebral palsy, or stroke. Moving from the hard objective sense to the "soft" skew probability on steps, allows for interesting properties such as a parameter that governs the chance of co-contraction (Figure 7.12). For the models tested in this research, the FAST analysis proved up to 30 times faster than CMC, especially in lower dimensional models: while CMC of the Arm26 model is on the order of minutes, FAST completes on the order of seconds. These methods may prove invaluable for researchers performing principal components analysis or designing systems for complex machine learning tasks like gesture recognition from computed controls. Additionally, if the joint loading is known, $\boldsymbol{\mathcal{H}}$**-FAS** can be further constrained by the known loads and used to obtain solutions that are closer to CMC, but still have deviations.

7.3 Tools

For advanced developers, an in-depth exploration of the mathematics and code and paradigms used are outlined in this section (Figure 7.17). A synthesis of python, C++, and C++ wrapped within Python through SWIG was used. Researchers need extremely high throughput data processing with C++, but it helps to abstract tools through wrappers, front ends and python libraries since clinicians or even most researchers do not care about things like pointers or command line arguments.

feasibleActivationWindowing.py Generates the feasible $\boldsymbol{\mathcal{A}}$ similar to feasibleActivation, but also computes the feasible $\boldsymbol{\mathcal{A}}$ constrained to the limitations determined by the first order $\boldsymbol{\mathcal{A}}$ dynamics.

sampler From a supplied $\boldsymbol{\mathcal{H}}$**-FAS** decomposed into an A file and b file, representing the augmented matrices $[-b|A]$ where $Ax \leq b$,sampler is a command line wrapper program that loads $\boldsymbol{A}$ and $\boldsymbol{b}$ and a starting location into a polytope object (see polytope.cpp) and performs a HAR of the polytope (see hitandrun.cpp). A good idea for a starting position can be the vertex center found using findMean, Chebyshev center or analytical center of the FAS by optimization on an activation set.

stats Generate statistics of a sampler data set. Allows user to compute a confidence interval from supplied alpha, mean and standard deviations.

polytope.h From a supplied $\boldsymbol{\mathcal{H}}$**-FAS** decomposed into an A file and b file, representing the augmented matrices $[-b|A]$ where $Ax \leq b$

hitandrun.h For a polytope object, perform a hit-and-run of the interior of the convex hull. Hit-and-run also wraps the methods for conic maximization on the analytical center.

convert_file.py This is a clone of the C++ file that converts MPLRS files to CSV.

extend_ArrayDouble.py Convert SimTK::ArrayDouble to Python numpy array.

find_coordinates_in_multibody_tree_order.py Returns a model's multibody tree order as a list.

get_data_from_storage_by_label.py Load data from a storage into a series of key-value pairs that can be accessed by storage column label.

interp.py Using the Python API for OpenSim, linearly interpolate a SimTK::Storage object to n frames.

make_output_directory.py Convenience method that makes an output directory from the CWD if it doesn't exist.

multibodyGraph.py Construct a multibody graph and various tools for navigating directed graphs.

to_np_matrix.py Convert an OpenSim::Matrix into a numpy matrix.

to_python_list.py Convert arbitrary OpenSim vectors into Python list objects.

fasWrapper.py An example wrapper function that calls the various FAS methods.

7.4 Chapter Summary

Previously in the literature, HAR MCMC was theorized as a tool for sampling FAS; instead the WK MCMC is used in the development of FAST. FAST is a probabilistic computed control algorithm that uses a variant of the Dikin walk in order to sample $\mathcal{H}$**-FAS** over the course of a dynamic task. FAST can use either the IK solution or an existing CMC states trajectory in order to obtain other feasible trajectories. FAST works for arbitrary tasks of arbitrary models just like CMC and SO. Additionally, instead of just the ID constraint, FAST can be used with $\mathcal{H}$**-FAS**-representations constructed from any downstream parameter from the muscle activations at each discrete time.

Chapter 8

Concluding Remarks

This dissertation implemented methods of performing feasible sets analysis over the course of a dynamic task for arbitrary musculoskeletal models. Using the methods in Chapter 5, it is possible to compute the sets of activations that satisfy the inverse dynamics constraints of arbitrary tasks for arbitrary models.

Using these task constrained activation sets, it is possible to map them to a downstream parameter using the methods in Chapter 4. Additionally, it was shown that the inverse problem halfspace representations developed in Chapter 5 could be even further constrained by the joint loading by procedurally constructing the analytical expression of the joint loads.

Note though, that the methods developed in Chapter 6 are not isolated to joint loads. If the analytical expression can be constructed for any muscle-dependent parameter such as fiber length and then observed in the lab by way of fluoroscopy, then they can be used as constraints in the halfspace representations.

In Chapter 7, a probabilistic computed control algorithm called FAST was developed that constrains feasible muscle activation by any user constructed n-dimensional geometry that wholly defines the task and random walks to activations over progression of the task instead of performing the typical quadratic optimization.

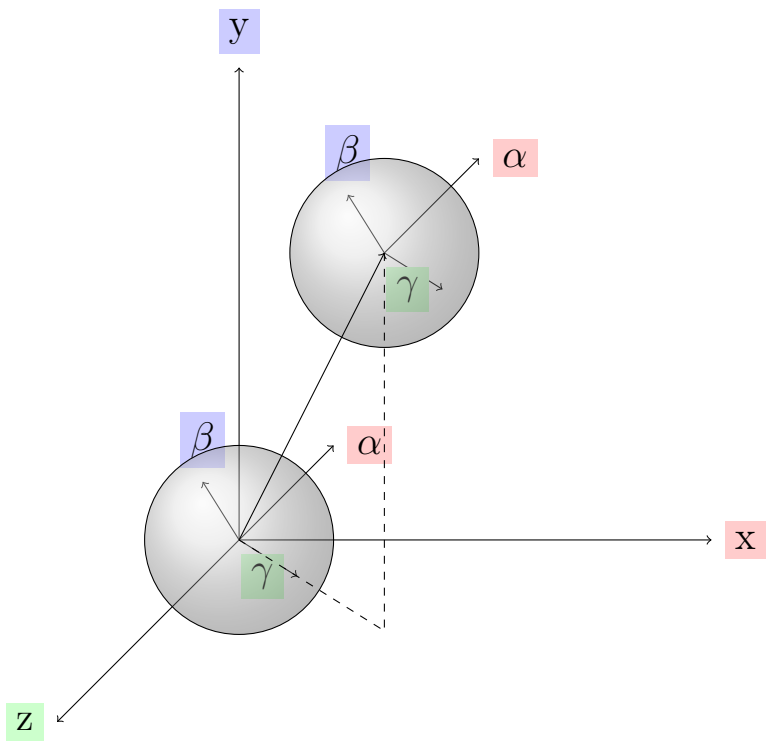


Figure 8.1: Simple 3D two body model.

8.1 Future Work

8.1.1 The Functional Approach and Variational Calculus

Be well aware that all the equations thus far deal with multibody systems existing in particular states (discrete times) of $\mathbb{C}$ trajectories: they are linear approximations for a given state of what really should be functional expressions.

Look at feasible trajectories in Chapter 7: these paths in feasible space are families of curves with functional forms that live in the non-convex space over the time domain. While functional approaches or Finite Elements analyses are popular methods in structural engineering where there is a boundary value problem with some arbitrary continuous operator and use some discretization method on the domain like Galerkin, the finite elements approach can be broadly applied to all sorts of engineering problems by the application of variational calculus, even rigid multibody dynamics (Figure 8.1). It is possible to derive a functional

expression of activations upper bounds and lower bounds with continuous time derivatives, but this exercise is left to the reader as future work. A finite elements approach to multibody dynamics:

It is possible to write the equations of motion (Equation 8.1)

$$\boldsymbol{a} = \frac{\boldsymbol{dv}}{\boldsymbol{dt}} = \frac{\boldsymbol{1}}{\boldsymbol{m}} \sum \boldsymbol{F} \tag{8.1}$$

Where $\boldsymbol{F}$ are MBS forces that live in $\mathbb{O}$.

And it is possible to write the conservation of angular momentum as (Equation 8.2):

$$\frac{\boldsymbol{d\mathcal{Q}}}{\boldsymbol{dt}} = \sum (\boldsymbol{r} \odot \boldsymbol{F})^{\boldsymbol{T}} \tag{8.2}$$

To formulate this problem as a Boundary Value Problem: where over the domain of a body, Ω_b, at the center of mass (Equation 8.3):

$$\int_{\boldsymbol{\Omega_b}} \boldsymbol{r\rho\delta\Omega_b} = \boldsymbol{0} \tag{8.3}$$

For any fixed point $\boldsymbol{p}$ in the body frame $\alpha - \beta - \gamma$ when the body has a center of mass $\boldsymbol{x}$ in the global frame $(x - y - z)$ (Equation 8.4):

$$\frac{\boldsymbol{dp}}{\boldsymbol{dt}} = \boldsymbol{0} \tag{8.4}$$

Because, these bodies are assumed rigid here (Equation 8.5):

$$\boldsymbol{\mathcal{Q}} = \int_{\boldsymbol{\Omega_b}} \boldsymbol{\rho r} \odot \boldsymbol{v\delta\Omega_b} = \int_{\boldsymbol{\Omega_b}} \boldsymbol{\rho r} \odot \frac{\boldsymbol{dr}}{\boldsymbol{dt}} \boldsymbol{\delta\Omega_b} + \int_{\boldsymbol{\Omega_b}} \boldsymbol{\rho r\delta\Omega_b} \odot \frac{\boldsymbol{dx}}{\boldsymbol{dt}} \tag{8.5}$$

But $\frac{dr}{dt} = 0$, so (Equation 8.6).

$$\boldsymbol{\mathcal{Q}} = \int_{\boldsymbol{\Omega_b}} \boldsymbol{\rho r\delta\Omega_b} \odot \frac{\boldsymbol{d\hat{x}}}{\boldsymbol{dt}} = \int_{\boldsymbol{\Omega_b}} \boldsymbol{\rho r} \odot (\boldsymbol{\omega} \odot \boldsymbol{r})\boldsymbol{\delta\Omega_b} \tag{8.6}$$

Where $\boldsymbol{\omega}$ is the angular velocity vector and note that this mapping is linear map over the domain.

It is possible to write (Equation 8.7):

$$\mathcal{Q} = \mathcal{I}\omega \tag{8.7}$$

Where the inertia matrix $\mathcal{I}$ is a function (Equation 8.8):

$$\mathcal{I} = \left[\rho r_z r_z \delta_{xy} - r_x r_y\right] \tag{8.8}$$

If unfamiliar with FEM, δ is the Dirac delta. Finally, Equation 8.6 can be rewritten (Equation 8.9):

$$\frac{dq}{dt} + \omega \odot q = \sum r \odot f \tag{8.9}$$

The tasks of incorporating applied muscle forces on Ω_b, discretization, and considering complicated multibody systems are out of the scope of this paper and are left as homework for the reader, but it should be clear that the future trajectory of this research is to design rigorous functional approaches, investigate for closed form solutions to feasible sets analysis, and discretizations over the time domain.

8.1.2 Trajectory

Future work should explore avenues for implementing the tools in real-time control of robotic manipulators. Further exploration into parameter tuning and methods of scaling the Dikin walk weights from Chapter 7 should be performed. Additionally, other probability distributions and MCMC methods should be explored as optimizations to the tools developed in this research. It would be nice to have someone write a plugin that can be imported into popular programs such as AnyBody or OpenSim. Generally these principles have been adapted for the post hoc analysis of OpenSim models; however, it would be beneficial for researchers to develop tools for the computation and visualization of feasible controls of a model at run-time from within OpenSim. The methods of Chapter 6 were only performed using the *in silico* results of OpenSim's joint reaction analysis, and a practical application using instrumented joints or some other muscle-dependent measurable parameter is in order. It is possible to use these results in tandem with muscle synergies analysis or task

prioritization schemes instead of just muscle activations. Also, enabling other researchers to magnify their computed control sets by a significant number will help facilitate further developments in biomechanics and has profound implications in clinical motion analysis.

Bibliography

[ben] https://www.cnet.com/news/how-to-choose-the-best-robot-vacuum-for-your-home-roomba-neato-ecovacs-2019/. ix, 19

[2] Andersen, H. C. and Diaconis, P. (2007). Hit and run as a unifying device. *Journal de la société française de statistique*, 148(4):5–28. 18, 115

[3] Anderson, F. C. and Pandy, M. G. (2001). Static and dynamic optimization solutions for gait are practically equivalent. *Journal of Biomechanics*, 34(2):153–161. 54

[4] Antwerp, K. W. V., Burkholder, T. J., and Ting, L. H. (2007). Inter-joint coupling effects on muscle contributions to endpoint force and acceleration in a musculoskeletal model of the cat hindlimb. *Journal of Biomechanics*, 40(16):3570–3579. 58, 95

[5] Arnold, A. S., Anderson, F. C., Pandy, M. G., and Delp, S. L. (2005). Muscular contributions to hip and knee extension during the single limb stance phase of normal gait: a framework for investigating the causes of crouch gait. *Journal of Biomechanics*, 38(11):2181–2189. 23, 25, 51

[6] Arnold, E. M. and Delp, S. L. (2011). Fibre operating lengths of human lower limb muscles during walking. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1570):1530–1539. 23, 25, 51

[7] Arnold, E. M., Hamner, S. R., Seth, A., Millard, M., and Delp, S. L. (2013). How muscle fiber lengths and velocities affect muscle force generation as humans walk and run at different speeds. *Journal of Experimental Biology*, 216(11):2150–2160. 23, 51, 58, 91

[8] Avis, D. (1998a). Computational experience with the reverse search vertex enumeration algorithm. *Optimization methods and software*, 10(2):107–124. 17

[9] Avis, D. (1998b). Living with lrs. In *Japanese Conference on Discrete and Computational Geometry*, pages 47–56. Springer. 17

[10] Avis, D. (2005). lrs homepage. 11, 17, 98

[11] Avis, D. and Fukuda, K. (1992). A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete & Computational Geometry*, 8(3):295–313. 18

[12] Avis, D. and Fukuda, K. (1996). Reverse search for enumeration. *Discrete applied mathematics*, 65(1-3):21–46. 17

[13] Avis, D. and Jordan, C. (2018). mplrs: A scalable parallel vertex/facet enumeration code. *Mathematical Programming Computation*, 10(2):267–302. 17, 86

[14] Avis, D. and Roumanis, G. (2013). A portable parallel implementation of the lrs vertex enumeration code. In *International Conference on Combinatorial Optimization and Applications*, pages 414–429. Springer. 17

[15] Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (slam): Part ii. *IEEE robotics & automation magazine*, 13(3):108–117. 19

[16] Berberich, E., Fogel, E., Halperin, D., Kerber, M., and Setter, O. (2010). Arrangements on parametric surfaces ii: Concretizations and applications. *Mathematics in Computer Science*, 4(1):67–91. 18

[17] Besier, T. F., Fredericson, M., Gold, G. E., Beaupré, G. S., and Delp, S. L. (2009). Knee muscle forces during walking and running in patellofemoral pain patients and pain-free controls. *Journal of biomechanics*, 42(7):898–905. 97

[18] Blajer, W. and Kołodziejczyk, K. (2004). A geometric approach to solving problems of control constraints: theory and a dae framework. *Multibody System Dynamics*, 11(4):343–364. 6

[19] Buchanan, T. S. and 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. 10

[20] Bunderson, N. E., Burkholder, T. J., and Ting, L. H. (2008). Reduction of neuromuscular redundancy for postural force generation using an intrinsic stability criterion. *Journal of Biomechanics*, 41(7):1537–1544. 32

[21] Chen, Y., Dwivedi, R., Wainwright, M. J., and Yu, B. (2017). Vaidya walk: A sampling algorithm based on the volumetric barrier. In *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1220–1227. IEEE. 118

[22] Chen, Y., Dwivedi, R., Wainwright, M. J., and Yu, B. (2018). Fast mcmc sampling algorithms on polytopes. *The Journal of Machine Learning Research*, 19(1):2146–2231. 18

[23] Chvatal, S. A. and Ting, L. H. (2013). Common muscle synergies for balance and walking. *Frontiers in Computational Neuroscience*, 7. 25

[24] Cohn, B. A., Szedlák, M., Gärtner, B., and Valero-Cuevas, F. J. (2018). Feasibility theory reconciles and informs alternative approaches to neuromuscular control. *Frontiers in Computational Neuroscience*, 12. 28, 111

[25] Dantzig, G. B. (1990). Origins of the simplex method. In *A history of scientific computing*, pages 141–151. 13

[26] De Sapio, V. and Khatib, O. (2005). Operational space control of multibody systems with explicit holonomic constraints. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2950–2956. IEEE. 62

[27] De Sapio, V., Khatib, O., and Delp, S. (2006). Task-level approaches for the control of constrained multibody systems. *Multibody System Dynamics*, 16(1):73–102. 27, 61

[28] De Sapio, V., Khatib, O., and Delp, S. (2008). Least action principles and their application to constrained and task-level problems in robotics and biomechanics. *Multibody System Dynamics*, 19(3):303–322. ix, 26, 61

[29] De Sapio, V., Warren, J., Khatib, O., and Delp, S. (2005). Simulating the task-level control of human motion: a methodology and framework for implementation. *The Visual Computer*, 21(5):289–302. 63

[30] Delp, S., Loan, J., Hoy, M., Zajac, F., Topp, E., and Rosen, J. (1990). An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Transactions on Biomedical Engineering*, 37(8):757–767. 9

[31] Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., Guendelman, E., and 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. 9, 10

[32] DeMers, M. S., Pal, S., and Delp, S. L. (2014). Changes in tibiofemoral forces due to variations in muscle activity during walking. *Journal of orthopaedic research*, 32(6):769–776. 93, 95

[33] Demircan, E., Besier, T. F., and Khatib, O. (2012). Muscle force transmission to operational space accelerations during elite golf swings. In *2012 IEEE International Conference on Robotics and Automation*, pages 1464–1469. 63, 70

[34] Dikin, I. (1967). Iterative solution of problems of linear and quadratic programming. In *Doklady Akademii Nauk*, volume 174, pages 747–748. Russian Academy of Sciences. 118

[35] Dorn, T. W., Wang, J. M., Hicks, J. L., and Delp, S. L. (2015). Predictive simulation generates human adaptations during loaded and inclined walking. *PloS one*, 10(4):e0121407. 7

[36] Edgerton, V. R., Smith, J. L., and Simpson, D. R. (1975). Muscle fibre type populations of human leg muscles. *The Histochemical Journal*, 7(3):259–266. 58

[37] Falisse, A., Van Rossom, S., Jonkers, I., and De Groote, F. (2016). Emg-driven optimal estimation of subject-specific hill model muscle–tendon parameters of the knee joint actuators. *IEEE Transactions on Biomedical Engineering*, 64(9):2253–2262. 95

[38] Featherstone, R. (1983). The calculation of robot dynamics using articulated-body inertias. *The international journal of robotics research*, 2(1):13–30. 6

[39] Fogel, E. and Halperin, D. (2007). Exact and efficient construction of minkowski sums of convex polyhedra with applications. *Computer-Aided Design*, 39(11):929–940. 54

[40] Fregly, B. J., Besier, T. F., Lloyd, D. G., Delp, S. L., Banks, S. A., Pandy, M. G., and D'lima, D. D. (2012). Grand challenge competition to predict in vivo knee loads. *Journal of Orthopaedic Research*, 30(4):503–513. 4

[41] Gauß, C. F. (1829). Über ein neues allgemeines grundgesetz der mechanik. *Journal für die reine und angewandte Mathematik*, 1829(4):232–235. 13

[42] Gerrig, R. J., Zimbardo, P. G., Zimbardo, P. G., Psychologue, E.-U., and Zimbardo, P. G. (2010). *Psychology and life*, volume 20. Pearson Boston. 21

[43] Gilchrist, L. A. and Winter, D. A. (1997). A multisegment computer simulation of normal human gait. *IEEE transactions on rehabilitation engineering*, 5(4):290–299. 10

[44] Gunther, M., Keppler, V., Seyfarth, A., and Blickhan, R. (2004). Human leg design: optimal axial alignment under constraints. *Journal of Mathematical Biology*, 48(6):623–646. 47

[45] Herzog, W. and Keurs, H. E. D. J. T. (1988). Force-length relation of in-vivo human rectus femoris muscles. *Pflügers Archiv*, 411(6):642–647. 22, 51

[46] Hicks, J. L., Schwartz, M. H., Arnold, A. S., and Delp, S. L. (2008). Crouched postures reduce the capacity of muscles to extend the hip and knee during the single-limb stance phase of gait. *Journal of Biomechanics*, 41(5):960–967. 58

[47] Higginson, J., Zajac, F., Neptune, R., Kautz, S., and Delp, S. (2006). Muscle contributions to support during gait in an individual with post-stroke hemiparesis. *Journal of biomechanics*, 39(10):1769–1777. 59

[48] Holmes, P., Full, R. J., Koditschek, D., and Guckenheimer, J. (2006). The dynamics of legged locomotion: Models, analyses, and challenges. *SIAM Review*, 48(2):207–304. 14

[49] Inouye, J. M. (2012). *Bio-inspired tendon-driven systems: Computational analysis, optimization, and hardware implementation*. University of Southern California. 14

[50] John, C. T., Anderson, F. C., Higginson, J. S., and Delp, S. L. (2013). Stabilisation of walking by intrinsic muscle properties revealed in a three-dimensional muscle-driven simulation. *Computer methods in biomechanics and biomedical engineering*, 16(4):451–462. 86

[51] Kannan, R. and Narayanan, H. (2012). Random walks on polytopes and an affine interior point method for linear programming. *Mathematics of Operations Research*, 37(1):1–20. 119

[52] Keenan, K. G., Santos, V. J., Venkadesan, M., and Valero-Cuevas, F. J. (2009). Maximal voluntary fingertip force production is not limited by movement speed in combined motion and force tasks. *Journal of Neuroscience*, 29(27):8784–8789. 28, 58

[53] Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53. 61, 62

[54] Khatib, O., Demircan, E., De Sapio, V., Sentis, L., Besier, T., and Delp, S. (2009). Robotics-based synthesis of human motion. *Journal of Physiology-Paris*, 103(3-5):211–219. 62

[55] Kirtley, C., Whittle, M., and Jefferson, R. (1985). Influence of walking speed on gait parameters. *Journal of Biomedical Engineering*, 7(4):282–288. 47, 58

[56] Kutch, J. J. and Valero-Cuevas, F. J. (2011). Muscle redundancy does not imply robustness to muscle dysfunction. *Journal of Biomechanics*, 44(7):1264–1270. 22, 32

[57] Kutch, J. J. and Valero-Cuevas, F. J. (2012). Challenges and new approaches to proving the existence of muscle synergies of neural origin. *PLoS Computational Biology*, 8(5). 22, 25, 32

[58] Lee, L.-F. and Umberger, B. R. (2016). Generating optimal control simulations of musculoskeletal movement using opensim and matlab. *PeerJ*, 4:e1638. 10

[59] Lerner, Z. F., Haight, D. J., DeMers, M. S., Board, W. J., and Browning, R. C. (2014). The effects of walking speed on tibiofemoral loading estimated via musculoskeletal modeling. *Journal of applied biomechanics*, 30(2):197–205. 95

[60] Lin, Y.-C., Walter, J. P., Banks, S. A., Pandy, M. G., and Fregly, B. J. (2010). Simultaneous prediction of muscle and contact forces in the knee during gait. *Journal of Biomechanics*, 43(5):945–952. 93

[61] Linden, M. L. V. D., Kerr, A. M., Hazlewood, M. E., Hillman, S. J., and Robb, J. E. (2002). Kinematic and kinetic gait characteristics of normal children walking at a range of clinically relevant speeds. *Journal of Pediatric Orthopaedics*, 22(6):800–806. 58

[62] Liu, M. Q., Anderson, F. C., Pandy, M. G., and Delp, S. L. (2006). Muscles that support the body also modulate forward progression during walking. *Journal of Biomechanics*, 39(14):2623–2630. 47

[63] Liu, M. Q., Anderson, F. C., Schwartz, M. H., and Delp, S. L. (2008). Muscle contributions to support and progression over a range of walking speeds. *Journal of Biomechanics*, 41(15):3243–3252. 48

[64] Lloyd, D. G. and Besier, T. F. (2003). An emg-driven musculoskeletal model to estimate muscle forces and knee joint moments in vivo. *Journal of biomechanics*, 36(6):765–776. 95

[65] Lovász, L. (1999). Hit-and-run mixes fast. *Mathematical Programming*, 86(3):443–461. 115

[66] Lovász, L. and Vempala, S. (2006). Hit-and-run from a corner. *SIAM Journal on Computing*, 35(4):985–1005. 115

[67] Mansouri, M. and Reinbolt, J. A. (2012). A platform for dynamic simulation and control of movement based on opensim and matlab. *Journal of biomechanics*, 45(8):1517–1521. 10

[68] Mansouri Boroujeni, M. (2015). Dynamic simulation and neuromuscular control of movement: Applications for predictive simulations of balance recovery. 7, 10, 22, 32

[69] Marjaninejad, A., Berry, J. A., and Valero-Cuevas, F. J. (2018). An analytical approach to posture-dependent muscle force and muscle activation patterns. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2068–2071. IEEE. 42

[70] Meyer, A. J., Eskinazi, I., Jackson, J. N., Rao, A. V., Patten, C., and Fregly, B. J. (2016). Muscle synergies facilitate computational prediction of subject-specific walking motions. *Frontiers in bioengineering and biotechnology*, 4:77. 25

[71] Millard, M., Uchida, T., Seth, A., and Delp, S. L. (2013). Flexing computational muscle: Modeling and simulation of musculotendon dynamics. *Journal of Biomechanical Engineering*, 135(2). xiii, 25, 51, 108

[72] Murray, R. M., Li, Z., and Sastry, S. S. (1993). *A Mathematical introduction to robotic manipulation*. Crc Press. 47

[73] Neptune, R., Kautz, S., and Zajac, F. (2001). Contributions of the individual ankle plantar flexors to support, forward progression and swing initiation during walking. *Journal of Biomechanics*, 34(11):1387–1398. 47, 58

[74] Neptune, R., Zajac, F., and Kautz, S. (2004). Muscle force redistributes segmental power for body progression during walking. *Gait and Posture*, 19(2):194–205. 47

[75] Neptune, R. R., Sasaki, K., and Kautz, S. A. (2008). The effect of walking speed on muscle function and mechanical energetics. *Gait and Posture*, 28(1):135–143. 58

[76] Nymark, J. R., Balmer, S. J., Melis, E. H., Lemaire, E. D., and Millar, S. (2005). Electromyographic and kinematic nondisabled gait differences at extremely slow overground and treadmill walking speeds. *The Journal of Rehabilitation Research and Development*, 42(4):523. 58

[77] Onyshko, S. and Winter, D. (1980). A mathematical model for the dynamics of human locomotion. *Journal of biomechanics*, 13(4):361–368. 8

[78] Pandy, M. G. (2001). Computer modeling and simulation of human movement. *Annual Review of Biomedical Engineering*, 3(1):245–273. 8

[79] Rajagopal, A., Dembia, C. L., DeMers, M. S., Delp, D. D., Hicks, J. L., and Delp, S. L. (2016). Full-body musculoskeletal model for muscle-driven simulation of human gait. *IEEE transactions on biomedical engineering*, 63(10):2068–2079. 8

[80] Ranganathan, R., Adewuyi, A., and Mussa-Ivaldi, F. A. (2013). Learning to be lazy: exploiting redundancy in a novel task to minimize movement-related effort. *Journal of Neuroscience*, 33(7):2754–2760. 22

[81] Reinbolt, J. A., Fox, M. D., Arnold, A. S., Õunpuu, S., and Delp, S. L. (2008). Importance of preswing rectus femoris activity in stiff-knee gait. *Journal of biomechanics*, 41(11):2362–2369. 58

[82] Renegar, J. (1988). A polynomial-time algorithm, based on newton's method, for linear programming. *Mathematical programming*, 40(1-3):59–93. 114

[83] Sachdeva, S. and Vishnoi, N. K. (2016). The mixing time of the dikin walk in a polytope—a simple proof. *Operations Research Letters*, 44(5):630–634. 119

[84] Sentis, L. (2007). *Synthesis and control of whole-body behaviors in humanoid systems*. Citeseer. 27

[85] Sentis, L. and Khatib, O. (2005). Control of free-floating humanoid robots through task prioritization. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1718–1723. IEEE. 7, 27

[86] Seth, A., Hicks, J. L., Uchida, T. K., Habib, A., Dembia, C. L., Dunne, J. J., Ong, C. F., DeMers, M. S., Rajagopal, A., Millard, M., et al. (2018). Opensim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLoS computational biology*, 14(7):e1006223. 86

[87] Simpson, C. S., Sohn, M. H., Allen, J. L., and Ting, L. H. (2015). Feasible muscle activation ranges based on inverse dynamics analyses of human walking. *Journal of Biomechanics*, 48(12):2990–2997. 28, 31, 48, 59, 88

[88] SimTK:Confluence (2019). Opensim documentation https://simtk-confluence.stanford.edu. ix, 11, 12

[89] Sohn, M., Mckay, J. L., and Ting, L. H. (2013). Defining feasible bounds on muscle activation in a redundant biomechanical task: practical implications of redundancy. *Journal of Biomechanics*, 46(7):1363–1368. 28, 31, 48

[90] Sohn, M. H., Smith, D. M., and Ting, L. H. (2019). Effects of kinematic complexity and number of muscles on musculoskeletal model robustness to muscle dysfunction. *Plos one*, 14(7):e0219779. 28

[91] Sreenivasa, M., Valero-Cuevas, F. J., Tresch, M., Nakamura, Y., Schouten, A. C., and Sartori, M. (2019). Neuromechanics and control of physical behavior: From experimental and computational formulations to bio-inspired technologies. *Frontiers in computational neuroscience*, 13:13. 32

[92] Stanev, D. (2018). *Biomechanical simulation of virtual physiological humans: modeling of musculoskeletal kinematic and dynamic redundancy using coordinate projection methods*. PhD thesis. 47

[93] Stanev, D. and Moustakas, K. (2019a). Modeling musculoskeletal kinematic and dynamic redundancy using null space projection. *PLOS ONE*, 14(1):1–26. 47, 63

[94] Stanev, D. and Moustakas, K. (2019b). Stiffness modulation of redundant musculoskeletal systems. *Journal of biomechanics*, 85:101–107. 47

[95] Steele, K. M., Tresch, M. C., and Perreault, E. J. (2013). The number and choice of muscles impact the results of muscle synergy analyses. *Frontiers in computational neuroscience*, 7:105. 25

[96] Thelen, D. G. (2003). Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults. *J. Biomech. Eng.*, 125(1):70–77. xiii, 22, 23, 25, 108

[97] Thelen, D. G. and 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. 12, 14

[98] Thelen, D. G., Anderson, F. C., and Delp, S. L. (2003). Generating dynamic simulations of movement using computed muscle control. *Journal of Biomechanics*, 36(3):321–328. 14

[99] Theodorou, E. and Valero-Cuevas, F. J. (2010). Optimality in neuromuscular systems. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 4510–4516. IEEE. 32

[100] Thurman, R. J., Jauch, E. C., Panagos, P. D., Reynolds, M. R., and Mocco, J. (2012). Four evolving strategies in the emergent treatment of acute ischemic stroke. *Emergency medicine practice*, 14(7):1–26. 33

[101] Todorov, E. and Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, 5(11):1226–1235. 61

[102] Tresch, M. C., Cheung, V. C., and d'Avella, A. (2006). Matrix factorization algorithms for the identification of muscle synergies: evaluation on simulated and experimental data sets. *Journal of neurophysiology*, 95(4):2199–2212. 25

[103] Tresch, M. C. and Jarc, A. (2009). The case for and against muscle synergies. *Current opinion in neurobiology*, 19(6):601–607. 25

[104] Umberger, B. R. (2010). Stance and swing phase costs in human walking. *Journal of the Royal Society Interface*, 7(50):1329–1340. 58

[105] Umberger, B. R., Gerritsen, K. G., and Martin, P. E. (2003). A model of human muscle energy expenditure. *Computer methods in biomechanics and biomedical engineering*, 6(2):99–111. 27

[106] Umberger, B. R. and Martin, P. E. (2007). Mechanical power and efficiency of level walking with different stride rates. *Journal of Experimental Biology*, 210(18):3255–3265. 27, 47

[107] Valero-Cuevas, F., Cohn, B., Szedlák, M., Fukuda, K., and Gärtner, B. (2015a). Structure of the set of feasible neural commands for complex motor tasks. In *Conference proceedings:... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, volume 2015, page 1440. NIH Public Access. 23, 32

[108] Valero-Cuevas, F., Cohn, B., Yngvason, H., and Lawrence, E. (2015b). Exploring the high-dimensional structure of muscle redundancy via subject-specific and generic musculoskeletal models. *Journal of Biomechanics*, 48(11):2887–2896. 23, 31, 32, 127

[109] Valero-Cuevas, F. J. (2016). *Fundamentals of neuromechanics*. Springer. 48

[110] Valero-Cuevas, F. J., Hoffmann, H., Kurse, M. U., Kutch, J. J., and Theodorou, E. A. (2009). Computational models for neuromuscular function. *IEEE reviews in biomedical engineering*, 2:110–135. 23, 32

[111] Welch, G., Bishop, G., et al. (1995). An introduction to the kalman filter. 19

[112] Whitehead, J. C. (2006). A comparison of contingent valuation method and random utility model estimates of the value of avoiding reductions in king mackerel bag limits. *Applied Economics*, 38(15):1725–1735. 121

[113] Winby, C. R., Lloyd, D. G., Besier, T. F., and Kirk, T. B. (2009). Muscle and external load contribution to knee joint contact loads during normal gait. *Journal of biomechanics*, 42(14):2294–2300. 95

[114] Winter, D. A. (1983). Knee flexion during stance as a determinant of inefficient walking. *Physical Therapy*, 63(3):331–333. 58

[115] Winters, J. M. (1995). An improved muscle-reflex actuator for use in large-scale neuromusculoskeletal models. *Annals of biomedical engineering*, 23(4):359–374. 23

[116] Winters, J. M. and Crago, P. E. (2012). *Biomechanics and neural control of posture and movement*. Springer Science & Business Media. 20

[117] Winters, J. M. and Stark, L. (1987). Muscle models: what is gained and what is lost by varying model complexity. *Biological cybernetics*, 55(6):403–420. 23, 34, 91

[118] Winters, J. M. and Stark, L. (1988). Estimated mechanical properties of synergistic muscles involved in movements of a variety of human joints. *Journal of biomechanics*, 21(12):1027–1041. 25, 95

[119] Winters, J. M., Woo, S. L., and Delp, I. (2012). *Multiple muscle systems: Biomechanics and movement organization*. Springer Science & Business Media. 34

[120] Xiang, Y., Arora, J. S., and Abdel-Malek, K. (2012). Hybrid predictive dynamics: a new approach to simulate human motion. *Multibody System Dynamics*, 28(3):199–224. 8

[121] Zajac, F. E. (1989). Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical reviews in biomedical engineering*, 17(4):359–411. 34

[122] Zajac, F. E. (1993). Muscle coordination of movement: a perspective. *Journal of biomechanics*, 26:109–124. 34

[123] Zajac, F. E. and Gordon, M. E. (1989). Determining muscles force and action in multi-articular movement. *Exercise and Sport Sciences Reviews*, 16. 95

[124] Zajac, F. E. and Winters, J. M. (1990). Modeling musculoskeletal movement systems: joint and body segmental dynamics, musculoskeletal actuation, and neuromuscular control. In *Multiple muscle systems*, pages 121–148. Springer. 34, 47, 95

Appendices

A Algorithms

Algorithm 1 Case 1 EFS (Peak Isometric F_0)

Input Model $\mathcal{M}$, CMC-derived set of states $\mathcal{S}_\mathcal{T}$, desired end-effector $\mathcal{B}_{ef}$
Output F_0-derived feasible force space over specific motion

```
load M
load S_T
for all s in S_T do
    load s
    for all coordinates c in M do
        for all muscles m in M do
            M_A(c, m) = compute moment arm of m about c
            F(c, m) = F_0^m cos(θ_m) where θ_m is a model state-dependant scalar.
            τ(c, m) = M_A(c, m) * F(c, m)
        end for
    end for
    compute station jacobian J of B_ef at COM
    W(s) = J^{-T} τ
    EFS_1(s) = ⊕(forces of W) where ⊕ = Minkowski sum of columns
end for
```

Algorithm 2 Case 2 EFS (Fiber Length $\tilde{l}^M$)

Input Model $\mathcal{M}$, CMC-derived set of states $\mathcal{S}_\mathcal{T}$, desired end-effector $\mathcal{B}_{ef}$
Output $\mathcal{F}_{\tilde{l}^M}$-derived feasible force space over specific motion

```
load M
load S_T
for all s in S_T do
    load s
    equilibrate muscles assuming 0 Fiber Velocity
    for all coordinates c in M do
        for all muscles m in M do
            M_A(c, m) = compute moment arm of m about c
            F_l̃M(c, m) = F_0^m c_l cos(θ_m) + F_PMF where θ_m and c_l are model state-dependant scalar.
            τ(c, m) = M_A(c, m) * F_l̃M(c, m)
        end for
    end for
    compute station jacobian J of B_ef at COM
    W(s) = J^{-T} τ
    EFS_2(s) = ⊕(forces of W)
end for
```

Algorithm 3 Case 3 EFS (Fiber Length & Fiber Velocity $\tilde{l}^M$ & $\tilde{v}^M$)

Input Model $\mathcal{M}$, CMC-derived set of states $\mathcal{S}_\mathcal{T}$, desired end-effector $\mathcal{B}_{ef}$
Output $\mathcal{F}_{\tilde{l}^M \& \tilde{v}^M}$-derived feasible force space over specific motion

```
load M
load S_T
for all s in S_T do
    load s
    for all coordinates c in M do
        for all muscles m in M do
            M_A(c, m) = compute moment arm of m about c
            F_l̃M&ṽM(c, m) = F_0^m c_l c_v cos(θ_m) + F_PMF where θ_m, c_l, and c_v are model state-dependant scalars.
            τ(c, m) = M_A(c, m) * F_l̃M&ṽM(c, m)
        end for
    end for
    compute station jacobian J of B_ef at COM
    W(s) = J^{-T} τ
    EFS_3(s) = ⊕(forces of W)
end for
```

Algorithm 4 2D EFS Computation

Input forces of $\mathcal{W}$ $\mathcal{W}_f$ of length m

Output Area $\mathcal{A}$, 2D convex hull of EFS $\mathcal{C}$

```
c=0
for i_1 = 0 to m − 1 do
    S = [0; 0; 0]
    sort columns of W_f as 2D vectors by increasing polar angle starting from W_f(:, i_1) (this can also just be presorted and circularly indexed)

    for i_2 = 0 to m − 1 do
        S += W_f(:, i_2)
        P(:, c) = S
        c+ = 1
    end for
end for
use a convex hull algorithm like 2D giftwrap of P to find A and C
```

Algorithm 5 3D EFS Computation

Input forces of $\mathcal{W}$ $\mathcal{W}_f$ of length m

Output Volume $\mathcal{V}$, 3D convex hull of EFS $\mathcal{C}$

```
c=0
for i_1 = 0 to m − 2 do
    S_1 = W_f(:, i_1)
    S_2 = [0; 0; 0]
    Pick W_f(:, i_1)
    Find the orthonormal basis of W_f(:, i_1)
    project vectors from 3D to the orthonormal basis
    sort the vectors by polar angle in the orthonormal basis
    for i_2 = 0 to m − 2 do
        S_1 += W_f(:, i_2)
        P(:, c) = S_1
        c += 1
        S_2 += W_f(:, i_2)
        P(:, c) = S_2
        c += 1
    end for
end for
use a convex hull algorithm like 3D Jarvis of P to find V and C
```

Algorithm 6 Hit and Run

Input $\mathcal{H}$ H-representation of convex polytope, initial point p_0

Output new point p

```
[b_{n×1} | A_{n×m}] = H_{n×m+1}
r = rand(m) (dist range[0,1])
current = p
for i_1 = 0 to n − 1 do
    h_s = A(i, :)
    aux = auxiliar point from p to A(i, :)
    vec_i = aux − p
end for
for i_1 = 0 to n − 1 do
    current+ = vec_i * r_i
end for
p = current
Go to 1
```

Algorithm 7 FAST

Input $\boldsymbol{\Gamma_{task}}$ for discrete time step Δt of a behavioral task and c_a and c_d

Output feasible activation space trajectory $\mathcal{A}$

1: $t = 0$
2: $\mathcal{A} = \emptyset$
3: $\forall i \in m, lb_i = 0, ub_i = 1$
4: construct $\mathcal{H}$ from $\boldsymbol{\Gamma_{task}}$ at t and ub and lb
5: find analytical center a_{ac} of $\mathcal{H}$
6: Dikin walk inside $\mathcal{H}$ from a_{ac} to new a_t
7: append a_t to $\mathcal{A}$
8: $\forall i \in m, lb_i = a_{t\,i} + \Delta t * \frac{1-a_{t\,i}}{c_a}, ub_i = a_{t\,i} + \Delta t * \frac{0-a_{t\,i}}{c_d}$
9: $t = t + 1$
10: Go to 4

Vita

Aravind Sundararajan was born January 13th, 1993 in Chennai, India. He grew up in West Windsor, NJ where he graduated from WW-P High School North. He received his Bachelor of Science in Biomedical Engineering with minor in Mathematics from Rutgers, New Brunswick, NJ in May 2015. He had a brief stint at the University of Madras in the Molecular Biology Department. He then went on to pursue biomechanics research with the Reinbolt Research Group in the Neuromuscular Biomechanics Lab at the University of Tennessee, Knoxville where he finished his Master's Degree in Biomedical Engineering in 2018, and successfully defended his Doctor of Philosophy in July 2020.